Quantifying the Notion of ‘Clumpiness’ Within Alignments Obtained From BLAST Similarity Searches

Jacky Birrell

PhD 2007
QUANTIFYING THE NOTION OF ‘CLUMPINESS’ WITHIN ALIGNMENTS OBTAINED FROM BLAST SIMILARITY SEARCHES

JACKY BIRRELL

A thesis submitted to the University of Abertay in partial fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computing and Creative Technologies

November 2007

I certify that this thesis is the true and accurate version of the thesis approved by the examiners

Signed.................................................. Date......................................

(Director of Studies)
ACKNOWLEDGEMENTS

This thesis is dedicated to the memory of my father
and all those times he kept me from quitting.

Thanks also go to my supervisors Louis and Janine for all their help and
encouragement and all my friends for helping me to celebrate my successes and
supporting me through difficulties. I'd also like to thank all my subjects for their
time in taking part in my experiments.
ABSTRACT

There are numerous methods utilised in the determination of the function of newly sequenced DNA or proteins. One such method is the use of sequence similarity searches, such as BLAST. However, due to the speed at which sequences can be produced and the ever-increasing size of the databases against which they are searched, it is becoming progressively more difficult for the scientist to carry out the necessary data analysis manually. Therefore, an automation of the analysis of the BLAST results should greatly reduce the amount of labour for the scientist and so improve the chances of accelerating research progress or indicate new fields of investigation.

An in-depth study of how the BLAST algorithm works was conducted. Also, interviews were used to determine which of the BLAST result features are of importance to the scientist in the decision of whether a particular similarity hit was of importance to their field of research and function determination. Based on this study, the feature of the clumpiness of a match’s alignment was chosen as the focus of this research. This decided, techniques into quantifying this clumpiness were studied and several possible clumpiness measures were proposed.

These measures were then tested with regard to specified criteria in order to assess their suitability as a clumpiness measure. This analysis was first conducted on synthetic data and it was found that the CUSUM measure proved to be the best according to the criteria and was chosen as the clumpiness measure for the subsequent testing. This took the form of testing the measure within real BLAST sequence analysis via the use of a prototype, which was utilised by scientists in their research. In conjugation with this, benchmark datasets containing families with distant relatives were used in order to assess the clumpiness measure’s ability to identify these distant relatives. Additional testing of the clumpiness measure was performed on a more abstract dataset of events and non-events in a one-dimensional field.
For both the prototype and the abstract testing, the results showed that the CUSUM clumpiness measure gives a good approximation of the degree of clustering of events within a one-dimensional field. In addition there is an indication that the measure will be of use in the identification of distant relatives, however, further testing is required to widen the subject base and further validate the measures suitability for assisting in the function determination of novel sequences.
CONTENTS

ACKNOWLEDGEMENTS........................................................................................................... 1
ABSTRACT........................................................................................................................................ 2
CONTENTS........................................................................................................................................ 4
TABLE OF FIGURES.................................................................................................................. 9
1  INTRODUCTION .................................................................................................................. 12
   1.1 Objectives ......................................................................................................................... 15
   1.2 Achievement of Objectives............................................................................................... 16
2  APPROACHES TO FUNCTION DETERMINATION.............................................................. 17
   2.1 Gene Prediction.................................................................................................................... 17
   2.2 Similarity Searches............................................................................................................. 18
      2.2.1 Similarity Search Contamination................................................................................ 19
      2.2.2 Volume of Similarity Search Results......................................................................... 20
      2.2.3 Orthologue/Paralogue Distinction........................................................................... 20
   2.3 Protein Domain and Family Prediction............................................................................. 21
   2.4 Structural Predictions......................................................................................................... 23
   2.5 Integrated Systems............................................................................................................. 25
      2.5.1 Graphical Display....................................................................................................... 25
      2.5.2 Automatic Annotation............................................................................................... 27
   2.6 Summary............................................................................................................................ 29
3  SIMILARITY SEARCHING WITH BLAST ........................................................................... 30
   3.1 Investigation of the BLAST Algorithm............................................................................... 30
      3.1.1 The Similarity Score................................................................................................. 30
      3.1.2 The E-Value.............................................................................................................. 34
      3.1.3 Gapped Sequence Searches..................................................................................... 34
   3.2 The BLAST Programs and Layout of the BLAST Results............................................... 35
4  BLAST ANALYSIS .................................................................................................................. 40
   4.1 The BLAST Analysis Issue................................................................................................. 40
      4.1.1 Data Overload............................................................................................................ 40
      4.1.2 Lack of Data Details.................................................................................................. 41
      4.1.3 Data Relevance......................................................................................................... 42
      4.1.4 Automation............................................................................................................... 43
   4.2 The Current Manual System.............................................................................................. 44
      4.2.1 Knowledge Elicitation............................................................................................... 44
4.2.2 Methods Used ..................................................................................... 45
4.2.3 Outcome .............................................................................................. 46
  4.2.3.1 Graphical Display ............................................................................ 47
  4.2.3.2 Description, Score and E-value ....................................................... 49
  4.2.3.3 Alignments ....................................................................................... 50
  4.2.3.4 Other Issues ..................................................................................... 51
4.2.4 Annotation Files ................................................................................... 51
4.3 Existing Approaches to Feature Automation ............................................... 52
5 THE NOTION OF CLUMPINESS .................................................................... 56
  5.1 Definition ..................................................................................................... 57
  5.2 Differentiation between Different Levels of Clumpiness and Reordering 58
  5.3 Circular Problem ...................................................................................... 60
  5.4 Clumpiness Criteria .................................................................................... 62
6 CLUMPINESS MEASURES ............................................................................. 65
  6.1 Elementary Statistical Measures ............................................................... 66
    6.1.1 Average Lengths of the Groups ........................................................ 66
    6.1.2 Largest Group Length and Proportion ................................................ 66
    6.1.3 Frequencies of the Different Sizes of the Groups ............................... 69
  6.2 Sliding Window Measures .......................................................................... 71
    6.2.1 Level of Match Intensity along the Length of the Line ...................... 71
    6.2.2 Markov Chains .................................................................................... 73
      6.2.2.1 Kth Order Markov Chains .............................................................. 75
    6.2.3 Hidden Markov Model ........................................................................ 78
      6.2.3.1 Two State HMM ............................................................................. 82
      6.2.3.2 Three State HMM ............................................................................ 83
    6.2.4 Spatial Data Analysis ........................................................................... 84
      6.2.4.1 Combined Count and Distance Analysis .......................................... 84
      6.2.4.2 Nearest Neighbours ........................................................................ 86
      6.2.4.3 Transect L-Function ........................................................................ 87
  6.3 Miscellaneous Measures ............................................................................. 90
    6.3.1 Adaptation of Chi-Squared Goodness-of-Fit Test .............................. 90
    6.3.2 CUSUM ............................................................................................... 92
  6.4 Summary ..................................................................................................... 96
7 MEASURE EVALUATION .............................................................................. 99
7.1 Methodology ........................................................................................................... 99
7.2 Simulated dataset .................................................................................................... 101
7.2.1 Markov .............................................................................................................. 101
7.2.2 HMM ............................................................................................................... 103
7.3 Subjects .................................................................................................................. 104
7.4 Results and Discussion .......................................................................................... 106
7.4.1 Correlation ........................................................................................................... 106
7.4.2 The 'Lonely Zero' ............................................................................................ 108
7.4.2.1 Additional Zeros in Line J ........................................................................... 109
7.4.2.2 Additional Test Set ...................................................................................... 111
7.4.3 'Big-Oh' Notation .............................................................................................. 116
7.4.3.1 Determining Complexity ........................................................................... 117
7.4.3.2 Complexities in programs – an example ....................................................... 118
7.4.4 Summary ............................................................................................................ 121
8 SYSTEM EVALUATION .............................................................................................. 124
8.1 Prototype Usage ..................................................................................................... 124
8.1.1 Methodology ....................................................................................................... 125
8.1.2 Results and Discussion ...................................................................................... 127
8.2 'Non-expert' Testing for Possible Measure Transferability .................................... 132
8.2.1 Methodology ...................................................................................................... 133
8.2.2 Results and Discussion ...................................................................................... 135
9 BENCHMARK TESTING <may change> ...................................................................... 140
9.1 Methodology .......................................................................................................... 140
9.1.1 Composition of dataset ...................................................................................... 140
9.1.1.1 Structure of Subset ...................................................................................... 141
9.1.1.2 Structure of BLAST results ........................................................................ 142
9.1.2 Testing methodology .......................................................................................... 143
9.2 Candidates for Dataset .......................................................................................... 144
9.2.1 Ubiquitin Conjugating Enzyme ......................................................................... 146
9.2.2 Transforming Growth Factor - Beta .................................................................. 148
9.2.3 Fibronectin ........................................................................................................ 149
9.2.4 SNF2 .................................................................................................................. 151
9.3 Results and Discussion ........................................................................................... 152
9.3.1 Positional Movement Analysis .......................................................................... 152

6
TABLE OF FIGURES

Figure 1.1. Representation of the journey from a gene, with its component promoter site, exons and introns, to the encoding protein sequence ........................................ 13
Figure 2.1. Display above results list in BLAST file ........................................ 20
Figure 2.2. Protein secondary structure a. Alpha helix, b. Beta sheet, where the sticks and balls represent the amino acid sequence, and c. secondary structures folding into protein tertiary structure, adapted from Alberts et al (Alberts et al. 2002) .... 24
Figure 2.3. Coiled coil showing the two α-helices twined together so that the hydrophobic regions are buried within the structure (Alberts et al. 2002) .......... 25
Figure 2.4. Example of graphical display showing predicted features, orange and blue bars, along a query sequence, black line, adapted from Gilbert (Gilbert 2002) ................................................................................................................. 26
Figure 3.1. The Blosum62 substitution matrix with an example of its use in calculating a similarity score ............................................................................................................. 31
Figure 3.2. Creation of word list with scores for each word when compared against itself using Blosum62 matrix (Madden 2003) .......................................................................................................................... 32
Figure 3.3. Extension of database hits within A residues .................................. 33
Figure 3.4. Example of a gapped alignment for a DNA sequence (Madden 2003) .......................................................................................................................... 35
Figure 3.5. Introductory section of the BLAST result file .................................. 37
Figure 3.6. BLAST hits distribution graph .......................................................... 37
Figure 3.7. Summaries of BLAST hits ................................................................ 38
Figure 3.8. Alignments between query sequence and matching database sequence .................................................................................................................. 38
Figure 4.1. Graph illustrating the exponential growth of sequence entries in the Genbank database from 1982 to 2004, (NCBI 2005e) ........................................................................ 41
Figure 4.2. Schematic representation of domain alignments between two sequences A and B .................................................................................................................. 43
Figure 4.3. Summary of key areas of the BLAST results used in function determination .................................................................................................................. 47
Figure 4.4. Example of sequence contamination showing short matches to vector sequence, circled. Generated by adding a short piece of vector sequence to the end of a sequence used in a BLAST search .................................................................................................................. 48
Figure 4.5. Once a sequence has been mapped onto a fully sequence genome of a related species, inferences may be made regarding neighbouring regions .......... 50
Figure 5.1. Variation in match intensity or alignment length will have affects on the distribution of matches and, therefore, clumpiness .................................................................................................................. 58
Figure 5.2. Examples of varying degrees of clumpiness .................................... 59
Figure 5.3. Reordered alignments in accordance with clumpiness, the blocks have been used for clarity .................................................................................................................. 60
Figure 5.4. Circular problem ............................................................................. 61
Figure 5.5. Incorporation of mismatches into a conserved region ..................... 63
Figure 6.1. Conversion of the similarity line to binary ........................................ 65
Figure 6.2. Example of the process of calculating the average group size .......... 66
Figure 6.3. Example of the process of calculating a. the largest group size and b. proportion the average is of this largest group .................................................................................................................. 67
Figure 6.4. Examples of various approaches to the redefinition of the largest group measure .................................................................................................................. 68
Figure 6.5. Example of frequency counts at different group sizes. A clumpy line would have higher frequencies at a higher group size than expected from a randomly distributed line .................................................................................................................. 69
Figure 6.6. Example of the process for calculating the frequency of groups compared to random ................................................................. 70
Figure 6.7. Sliding window along line framing consecutive subsections of the line from each position in the line .......................................................... 71
Figure 6.8. Comparison of two sequences with the same number of ones but different distributions ................................................................. 72
Figure 6.9. Example of the intensities measure calculating a. the number of windows with a greater intensity than the overall and b. the number of windows with a greater intensity than expected ................................................................. 73
Figure 6.10. Movements through a Markov chain between states a, b, c and d with a. all possible movements, b. restricted movement between certain states .......... 74
Figure 6.11. Systematic diagram for the 1st order Markov chain with states 1 or 0.. 75
Figure 6.12. Estimation of transition probabilities ................................................................. 75
Figure 6.13. Example of a \( k \)th order Markov chain. a. \( k \)th order Markov chain probability formula. b. Example of the probability formula using \( k = 3 \)........ 76
Figure 6.14. Examples for calculating a. the average probability for each of the 8 order Markov chains and b. the number of probabilities greater than or different from expected ................................................................. 77
Figure 6.15. Urn and ball model showing four urns each with seven coloured balls. Underneath are a sample of the associated probabilities of obtaining each of the colours from each of the urns (adapted from Rabiner (Rabiner 1989))............ 78
Figure 6.16. Three dimensional representation of the search space for the HMM parameter estimation showing two local maxima, where \( c1 \) is an observation probability and \( c\text{uuu}1 \) represents all the other probabilities ......................... 82
Figure 6.17. Representation of the two state Hidden Markov Model for producing lines of ones and zeros ................................................................. 82
Figure 6.18. Representation of the three state Hidden Markov Model for producing lines of ones and zeros ................................................................. 83
Figure 6.19. From the first position, counting the number of ones a. 3, b. 5 and c. 7 characters along ................................................................. 84
Figure 6.20. Graph showing the counts of ones at the various positions in a line against the different distances, created using the R package ......................... 85
Figure 6.21. Example of the process for the combined distance and count, showing the calculation of a. the number greater than expected and b. the number greater than half the distance plus one................................................................. 86
Figure 6.22. Example of the process of counting neighbours with a selected position underlined and its two neighbours circled ................................................................. 87
Figure 6.23. Example of how counts at distances can affect \( L(h) \) if \( h = 6 \) .............. 88
Figure 6.24. (a) Estimated K-function for a clustered spatial pattern (full line) and theoretical K-function (dashed line), (b) Estimated L-function for the same clustered spatial pattern (full line) and theoretical L-function (dashed line); data courtesy to Paul Armstrong ................................................................. 89
Figure 6.25. Example of the line being split into subsections of length 5 with a. a large clump being broken by the ‘cut’ (arrowed) and b. using multiple runs to compensate for the split ................................................................. 91
Figure 6.26. Example of the calculation of the clumpiness value using an adaptation of the Chi-squared test on a clumpy and non-clumpy line ...................... 92
Figure 6.27. Example of CUSUM detection of a mean shift ............................................. 94
Figure 6.28. Example of the CUSUM measure ................................................................. 96
Figure 7.1. Markov chain used to generate a simulated dataset showing a. schematic diagram of the Markov model and b. the probability matrix for state transitions ................................................................. 102
Figure 7.2. Progression through probabilities to generate sequences. As the two probabilities for each state total 1, the probability of the transition to state 0 in each case is the 1- probability to state 1 .......................................................... 102
Figure 7.3. Examples of the different Markov chain probabilities producing simulated sequences but with widely varying intensities .................................................. 103
Figure 7.4. Hidden Markov model used to generate a simulated dataset ............... 104
Figure 7.5. Strips used in ordered dataset creation .................................................. 105
Figure 7.6. Adding extra zero into two locations in line. Lines are truncated with tildes for illustrative purposes ................................................................. 109
Figure 7.7. CUSUM catching lonely zero ............................................................... 111
Figure 7.8. Graph showing the variation in the CUSUM value including a single zero ............................................................. 111
Figure 7.9. Hidden Markov generated set, a., with a replaced zero to create one test set, b., and a relocated zero to create another set, c., circled. For clarity, only the line with the replacements is shown and white blocks denote a zero and green a one...................................................................................................................... 113
Figure 7.10. Pseudo-code for the CUSUM measure program showing a. the main body of the program and b. the method ‘Stats’ called from the main body. ... 119
Figure 8.1. Repositioning of a BLAST result, circled form a. the 17th to b. the 11th position of the reordered list in the clumpiness report, b................................. 127
Figure 8.2. Original clumpiness report with higher clumpiness values, circled, not used in the reordering process ................................................................. 128
Figure 8.3. Section of the report page for the search using ube2ji showing sequence raised in order (circled) ........................................................................ 129
Figure 8.4. Alignments for the moved sequence and the one immediately below it 130
Figure 8.5. Comparison of BLAST e-values to Clumpiness value ordered by e-value .................................................................................................................. 132
Figure 8.6. Section of original strips illustrating confusion of white lines........... 133
Figure 8.7. Example strip used in effectiveness of clumpiness measure.............. 134
Figure 9.1. Schematic representation of a comet to illustrate the form the dataset needs to take....................................................................................................... 141
Figure 9.2. Schematic representation for the desired appearance of the BLAST result hit list ......................................................................................................... 142
Figure 9.3. Schematic representation of the desired dataset.................................. 143
Figure 9.4. Example table for the comparison of order positions ......................... 144
Figure 9.5. Subsections of a phylogenetic tree to illustrate the difference in degrees of distance with a. closely related sequences and b. distantly related sequences .................................................. 145
Figure 9.6. Process of Ubiquitin ligase complex activation, (Alberts et al. 2002). 146
Figure 9.7. Phylogenetic tree of human UBE2 proteins generated in the Phylip suite with a greyed out half ‘comet’ for illustrative purposes ......................... 147
Figure 9.8. Process of cell signalling by the TGF-β, adapted from Alberts et al (Alberts et al. 2002) ................................................................................................. 148
Figure 9.9. Modular characteristics of protein domains, (Strachen and Read 1999) 149
Figure 9.10. Representation of the change in the distribution subset sequences, grey, between a. the BLAST score order and b. the Clumpiness score order....... 155
1 INTRODUCTION

Except for a few cell types, such as red blood cells, all cells in any organism contain a copy of the organism’s genetic makeup within its nucleus. This genetic makeup is encoded in genes within the DNA. DNA is a double helix of two complementary linear sequences, which are bound together by the interaction of four bases, or nucleotides, Adenosine, Cytosine, Guanine and Thiamine (A, C, G and T), specifically, Adenosine binds with Thiamine and Cytosine binds with Guanine. As well as coding the hereditary characteristics of a particular organism, genes also encode for all the proteins and enzymes required by the cells of the organism for basic function and for the specific function of that cell, such as the β-cells in the pancreas producing insulin.

Each protein is encoded in the DNA as a gene, consisting of exons and introns, with associated promoter sites to define when and to what extent that gene should be expressed, top of Figure 1.1\(^1\). Inside the nucleus DNA is transcribed into RNA via RNA polymerase, step 1 in Figure 1.1, which attaches to the promoter site and moves stepwise along one of the strands of DNA transcribing a complimentary copy to create a preliminary mRNA. Spliceosomes, step 2 in Figure 1.1, then splice out the non-coding introns to form the mRNA, which is then exported to the ribosome for protein translation. It should be noted that in prokaryotes a simpler system is employed without the use of introns and, therefore, the splicing step. Every three nucleotides equates to a codon, the complimentary anticodon of which codes for a specific amino acid. Inside the Ribosome tRNA brings the amino acids with the correct anticodon in line with the codon on the mRNA, Figure 1.1 inset. Attached to the tRNA is the correct amino acid, blue in Figure 1.1, which then binds to each of its neighbouring amino acids, thereby creating a chain of amino acids, a protein sequence. At each step there are mechanisms to ensure accuracy, RNA polymerase in the nucleus and RNA synthetase in the ribosome.

Previously, it was very time-consuming and costly to sequence a section of DNA but improvements in technology and automation have made it is easier and faster to obtain DNA or protein sequences. Therefore, there are now numerous projects for determining the sequence of DNA and their encoded proteins for any organism, and an increasing number of complete genomes, including Human, Drosophila melanogaster and Arabidopsis thaliana, all of which can be viewed on NCBI’s Entrez Genome database, (NCBI 2005c). However, the knowledge of a gene or protein sequence does not equate to knowledge of its function and, therefore, the next step in these genomic and protein-sequencing projects is the function determination of the sequences produced (Scordis et al. 1999). The primary method to obtain the correct function for DNA or protein sequences is to conduct biochemical experimentation in
the lab (Pellegrini et al. 1999). However, it is extremely time-consuming to attempt all the different tests for all the possible functions; therefore, some prior knowledge to reduce this number of experiments would be advantageous.

There are various indicatory methods to achieve this, including gene and protein structure prediction, detection of homologies to DNA and protein sequences of known function or protein structures and the study of phylogenetics. Each method gives an indication as to the possible function or functions of an unknown DNA or protein sequence, which can then be used to direct the choice of biochemical experiments. The manual application of methods themselves can be time-consuming but it is possible to automate them and an overview of just some of the many programs and algorithms that have been produced is presented in Chapter 2. However, the detection of homologies, similarity searches, are frequently cited in the literature as a first step in function determination (Attwood 2000) and that the similarity search algorithm, BLAST, is the more commonly used (Pearson 1998). Therefore, the focus of this research is in the determination of the function of a protein sequence by homology to other protein sequences by the BLAST algorithm (Altschul et al. 1990), a detailed description of which can be found in Chapter 3.

On the other hand, as is described in Chapter 4, there are a number of issues in analysing the results from BLAST searches, of most interest here being the possibility of missing results of low similarity but high biological significance due to the vast numbers of results that BLAST can produce. Therefore, some form of automation of the BLAST analysis is required in order to reduce the workload and identify these ‘twilight zone’, (Rost 1999), BLAST hits. To this end, it is first necessary to understand the current manual process used to analyse a BLAST result file, the details of which can be found in Chapter 4, and from that it is possible to determine what form of automation is required.

This research concentrates on the quantification of the notion of ‘Clumpiness’, as defined in Chapter 5, within the alignments between a query sequence and a BLAST result hit. The hypothesis is that this quantification would highlight otherwise overlooked hits and can be used to give additional information to the scientist, assisting them in their analysis of the BLAST result file. With the form of the
automation decided, it is necessary to formulate possible solutions to meet this challenge and these are described in detail in Chapter 6. Subsequently, these solutions must be assessed as to their suitability for addressing the issue in question, the details and results of which can be found in Chapter 7.

Since the purpose of this research is the development of a form of automation to assist in the BLAST results analysis, any proposed automation algorithm also needs to be tested within actual BLAST analysis problems. That is, for the scientists to use the automation in the course of their research to ascertain whether it is actually of any use in rendering the BLAST results easier to analyse. In order to do this, the automation needs to be incorporated into some form of program, or prototype, which the scientists can use and evaluate. The design of this prototype is given in Appendix E and the results of its evaluation in Chapter 8. Finally, Chapter 9 describes and summarises the results of a benchmark test on two of the possible measures.

1.1 Objectives
The overall objective of this research is the definition and quantification of the notion of clumpiness as it relates to the field of protein function determination using the BLAST similarity search algorithm.

In order to demonstrate the value of clumpiness analysis in the field of protein function determination it was first necessary to lay down the ground work upon which the main research would be built. Therefore, the initial aims were to:

a. Gain an understanding of the current methods utilised in the process of DNA and protein function determination.

b. Obtain an understanding of the BLAST similarity search algorithm.

c. Gather and analyse data on the manual process of BLAST analysis currently undertaken.

With the foundations thoroughly laid down the research could concentrate on the main objectives of the research, that of:

1. The definition of the notion of clumpiness and the criteria describing it.
2. The exploration of possible quantification methods.
3. The verification of the quantification methods as determined by the criteria.
4. The creation and evaluation of a prototype in order to examine the ability of a clumpiness measure to assist in the determination of protein function.

1.2 Achievement of Objectives

Of the initial foundation aims, thorough literature reviews and interviews were carried out giving the necessary knowledge base, specifically in objective order:

a. Covered in Chapter 2, giving a broad coverage of the different programs available for function determination.

b. Covered in Chapter 3, giving an in-depth review of the BLAST algorithm.

c. Covered in Chapter 4, describing the methodology for elucidating the current manual BLAST result analysis and the analysis of the information obtained.

The main focus of this research is the definition, methodology and analysis of clumpiness quantification and forms the latter four aims of this research:

1. Covered in Chapter 5, detailing the definition of the notion of clumpiness and the resulting criteria against which a prospective measure could be assessed. Additionally, the circular issue of the relationship between a model and a measure of clumpiness is described and the difficulty in identifying the ideal measure when there is currently no model.

2. Covered in Chapter 6, describing a number of different approaches to the development of a clumpiness measure.

3. Covered in Chapter 7, containing the analysis of each of the measures against an initial test set as per the criteria defined by aim 1 and suggested which of these possibles would be the 'best' clumpiness measure.

4. Covered in Chapters 8 and 9, with Chapter 8 giving an analysis of the feedback from subjects using a prototype program that contained one of the clumpiness measures highlighted in Chapter 7. Chapter 9 covered further testing of the clumpiness measure with respect to its utility in assisting scientists in identifying less similar but biologically relevant sequences.
2 APPROACHES TO FUNCTION DETERMINATION

As stated in the introduction, there are a vast number of programs and algorithms that give insight into the functional roles of portions of DNA or protein sequences and this Chapter conducts a survey exploring a subset of these strategies. The various algorithms that have been developed, concentrate on different aspects of sequence analysis, often dependent on the particular interest of the developer (Frishman et al. 2001). Here, these areas have been broadly categorised under (i) gene prediction, (ii) similarity searches, (iii) protein domain and family prediction and (iv) structural predictions.

2.1 Gene Prediction

Gene prediction is the identification of functional elements within strands of DNA, such as exons, introns or promoter sites (Burge and Karlin 1997). From the perspective of this research, the most important of these is the prediction of the coding regions, or exons, allowing the discovery of the complete protein sequence for use in its function determination. There are two broad approaches to this, termed in the literature as intrinsic, or ab initio, and extrinsic searches. Intrinsic involves the detection of “specific nucleic acid motifs or global statistical patterns” (Frishman et al. 1998), and extrinsic involves the detection of homologies with known genes.

Intrinsic methods include GeneMark (Borodovsky and McIninch 1993), GRAIL (Uberbacher et al. 1996) and TigrScan and GlimmerHMM (Majoros et al. 2004), which use training sets of sequences in order to enable the algorithms to ‘recognise’ potential gene features in unknown query sequences. For example, if the training set reveals a certain pattern for most or all of the known genes in that training set, the algorithms will be able to seek out the same or similar pattern in a query sequence. Alternatively the likelihood of certain features can be calculated based on known data and used to predict the occurrences of potential gene features as in GeneFinder (Sanger Institute 2003).

Extrinsic methods include an EST-driven gene annotation (Bailey et al. 1998b) and Gene2EST (Gemund et al. 2001), both of which search for homologies between a genomic sequence and an expressed sequence tag (EST) database. ESTs are a product of sequencing cloned mRNA, which contains the introns from DNA during
the translation process, Figure 1.1. Consequently, they relate to coding regions only and a "similarity to an EST is a highly reliable indicator that a sequence is associated with a gene" (Bailey et al. 1998b). Alignments can also be made with mRNA sequences, and translated protein sequences, as in the EAnnot program (Ding et al. 2004).

Some algorithms incorporate both intrinsic and extrinsic approaches to gene prediction though the emphasis between intrinsic and extrinsic may vary. For example, GenScan (Burge and Karlin 1997) uses a 'trained' algorithm to identify gene features and a homology search as the next step. On the other hand, Orpheus (Frishman et al. 1998) uses the results of a homology search to generate statistics based on the frequency of three nucleic acid units, or codons, which it can then use to search novel areas of DNA.

2.2 Similarity Searches

Similarity searches produce hits to sequences in the databases that have regions of homology to the query sequence and if the database hit is of a known function, it can be inferred that the query sequence may also have the same function (Shoop et al. 1995). Previously, algorithms, such as the Needleman and Wunsch algorithm, were used, which aligned sequences to represent the maximum similarity between the two (Needleman and Wunsch 1970). However, this process is computationally expensive and, within a large database, is only feasible on specialist hardware, such as a supercomputer (Altschul et al. 1990). BLAST, however, uses a heuristic algorithm, which vastly reduces the time and effort involved in a search but heuristics may also reduce the algorithm's sensitivity leading to a trade-off between speed and effectiveness.

It has also been noted (Altschul and Gish 1996) that related proteins or DNA often only share isolated regions of similarity. Therefore, if only the overall, or global, alignment was considered, as in the dynamic algorithms, these isolated region relationships would not be detected. Since BLAST seeks local alignments it does not suffer from this problem.
Similarity search algorithms such as BLAST (Altschul et al. 1990) or FASTA (Lipman and Pearson 1985) score alignments between a query sequence and those found in online sequence databases using amino acid substitution matrices such as BLOSUM (Henikoff and Henikoff 1992) or PAM (Dayhoff et al. 1978) and a number of different search algorithms have been devised dependent on the search sequence and output required (NCBI 2005a). Functional annotation of a gene or protein can then be performed by transfer from the best ‘hit’ (Yuan et al. 1998). However, there are certain limitations due to a) the degree of contamination in the query sequence and database entries, b) the shear volume of data that can be returned by BLAST and c) the ability to differentiate between orthologues and paralogues.

2.2.1 Similarity Search Contamination

Contamination can originate from either external sources or within the sequence itself. External sources include vector sequences, vectors being used during sequence cloning and can result in very good hits to completely irrelevant sequences, since the hit is to the vector and not the sequence. Within the sequence there can be regions of low complexity, or bias regions, and sequence repeats, which can cause difficulties in sequence searches as they can produce meaningless alignments (Altschul et al. 1994). Low complexity regions are regions of a sequence that are rich in a particular nucleic acid or amino acid (Koonin and Galperin 2003) and repeats are duplicated sections of DNA (Smit et al. 2003). Low complexity regions in DNA can be used to identify genes (Bork et al. 1998) and in protein sequences these areas relate to non-functional regions of the protein, which may have another, equally important role in the protein function (Koonin and Galperin 2003). However, these low complexity regions can affect the calculation of the Karlin-Altschul statistics (Altschul no date), and produce good hits to biologically irrelevant database sequences. It would, therefore, be advantageous to filter out both repeats and low complexity regions, such as in SEG (Wootton and Federhen 1996) and Reputer (Kurtz et al. 2001), which use the composition of the amino acids to determine regions of low complexity and repeats, respectively. In BLAST the score and e-value can be recalculated using “composition-based statistics” (Schäffer et al. 2001) and RepeatMasker (Smit et al. 2003) and CENSOR (Jurka et al. 1996) locate repeats and hits to known sequences, such as vector sequence.
2.2.2 Volume of Similarity Search Results

BLAST similarity searches can produce hundreds of pages of text (Chi et al. 1995), which must be sifted in order to identify the relevant data. Displaying the results graphically can highlight significant sections in a much more uncluttered and easier to interpret manner (Carter and Bellgard 2003). The BLAST program itself (NCBI 2000) has a small display above the results, Figure 2.1, showing the locations of the hits in relation to the query sequence. Visual BLAST (Durand et al. 1997), MASV (Carter and Bellgard 2003), BEAUTY (Worley et al. 1998, Worley et al. 1995) and PowerBLAST (Zhang and Madden 1997) also use a display of alignment against the query sequence. Other programs, such as BLASTQuest (Farmerie et al. 2003) and MuSeqBox (Xing and Brendel 2001) generate tabulated summaries that are hyperlinked to the original data. Alternatively, the BLAST hits may be clustered as in CBLAST (Miller and Fuchs 1997) or filtered to remove redundant or over-represented results as in the program by Berman et al. (2000b).

![Color Key for Alignment Scores](image)

Figure 2.1. Display above results list in BLAST file

2.2.3 Orthologue/Paralogue Distinction

Orthologues are genes of differing species that have evolved from common ancestors maintaining the same or similar function whereas paralogues are genes with common ancestry that have undergone duplication resulting in a different, though related function (Nagl 2003, pp. 12-14). Although both orthologues and paralogues can
produce homologies to a query sequence, it is only hits to orthologues that may result in correctly assigning a function from a hit. Therefore, it would be advantageous to filter out these paralogues leaving just functionally relevant hits, such as by an orthologue detection program (Yuan et al. 1998). This algorithm compares the relationships between the genes in the BLAST search with the relationships between the species present in the BLAST search to identify duplication sites and, therefore, paralogues.

2.3 Protein Domain and Family Prediction

In a protein, certain residues of the sequence are vital for the stability of its structure or function (Orengo 2003). These conserved residues form modules or domains within a protein. When multiple sequences are aligned together these domains, or conserved regions, can become evident (Do et al. 2005).

Therefore, numerous algorithms that calculate the multiple alignment of a group of sequences have been developed, the majority of which generate the multiple sequence alignment via progressively aligning sequences. Algorithms, such as CLUSTAL (Chenna et al. 2003, Higgins et al. 1996, Thompson et al. 1994), ProbCons (Do et al. 2005), MAFFT (Katoh et al. 2005) and MuSiC (Tsai et al. 2004), generate a guide tree for the multiple alignment to follow. A guide tree is an approximation of the phylogenetic relationship between the sequences involved in the multiple alignment. Phylogenetic relationships can, themselves, be used for function determination as in LumberJack (Lawrence et al. 2004) or Galaxie (Nilsson et al. 2004) where related proteins are thought to have related functions. Other multiple sequence algorithms, such as MultAlign (Corpet 1988), MAP2 (Ye and Huang 2005) and ProtoMap (Yona et al. 2000) conduct all-by-all pairwise similarities and cluster the sequences progressively, aligning the most similar first, progressing down to the least similar. Two other multiple alignment programs are PSI-BLAST (Altschul et al. 1997, Schäffer et al. 2001), which aligns BLAST results to the query sequence and MaxHom (Sander and Schneider 1991), which generates a profile as each sequence is added; with an additional check of the overall multiple alignment.
These multiple alignments can show a pattern of conserved residues in the sequence, which is also known as a motif (Koonin and Galperin 2003), domain or profile. These motifs can then be searched against the databases (Staden 1990), such as in PHI-BLAST (Zhang et al. 1998), which uses motifs to restrict the sequence database for BLAST searching to only those sequences with a certain pattern. Alternatively, a sequence can be searched for a motif of known function in motif databases, such as InterPro (Apweiler et al. 2001), PRINTS (Attwood et al. 2003), which uses multiple motifs called fingerprints, PROSITE (Falquet et al. 2002), BLOCKS (Henikoff et al. 2000, Henikoff et al. 1999), and SMART (Letunic et al. 2002, Schultz et al. 2000). Two such programs are RNAMotif (Macke et al. 2001), using formal descriptions of RNA structural elements to locate similar regions in a DNA sequence, or FingerPRINTScan (Scordis et al. 1999), which finds all hits between a query sequence and the PRINTS database.

The motif databases mentioned above seek to produce patterns to uniquely identify related proteins, or protein families. This can be advantageous from the perspective of function determination especially as it can reveal distantly related proteins that are not detected by similarity searches (Falquet et al. 2002). The Motif databases use in-house or external motif generating programs. These motif generating programs can use information from sets of functionally related sequences to determine a pattern, which can then be used to predict these motifs in other sequences, such as in eMotif (Nevill-Manning et al. 1998), MOTIF (Smith et al. 1990) or PRATT (Jonassen et al. 1995). Alternatively, position specific scoring or weight matrices for the motif are generated, as in Bipad (Bi and Rogan 2004) or Stadens program (Staden 1990), which can then be used as the substitution matrix in a similarity search. One other method of motif generation uses the physiochemical properties of amino acids, such as hydrophobicity. This is achieved either directly in motif prediction, as in the Nagarajan and Yona domain prediction server (Nagarajan and Yona 2004) or in a scoring system based on the different properties, as in the Smith and Smith algorithm (Smith and Smith 1990). The use of physiochemical properties is advantageous as different amino acids with similar physiochemical properties can be interchanged in a protein without affecting the protein function (Jonassen et al. 1995).
2.4 Structural Predictions

Another aspect of finding conserved regions in a sequence is the actual three-dimensional structure of the protein. It is the three-dimensional structure that is critical to the function of a protein (Lyon 2004), if a protein has the incorrect three-dimensional shape it will not be able to interact with, for example, the cell membrane. Structural prediction can also give functional clues in the absence of homologies (Koonin and Galperin 2003). Protein structure can be predicted by fold recognition using multiple alignments and neural networks trained on available databases of protein structure, such as PDB (Berman et al. 2000a), for example, PHD (Rost 1996) and GenTHREADER (Jones 1999a, McGuffin and Jones 2003). Alternatively structure-structure alignments may be used for structural similarity searches, such as in PROFESY (Lee et al. 2004), though this also uses fold recognition if no homologies can be found, or by predicting the three-dimensional structure from the secondary structure, as in the program by Russell, Copley and Barton (1996).

Secondary structures, Figure 2.2 a and b, are the base units, so to speak, that fold together to produce the tertiary, or three-dimensional structure of the protein, Figure 2.2.c. They are made up of α-helices, β-sheets, loops and tight-turns, of which the first two are deemed regular and the latter two irregular structures (Kaur and Raghava 2003b). Programs, such as PSIPRED (Jones 1999b) and HYPROSP (Wu et al. 2004) use multiple sequence alignments and machine learning to predict the regular secondary structure. On the other hand, GammaPred (Kaur and Raghava 2003a) and BetaTPred2 (Kaur and Raghava 2003b) use position specific matrices with the machine learning for the prediction of irregular secondary structure.
Another structural element that must be considered is coiled coils, when two or more \( \alpha \)-helices with hydrophobic regions twist together such that these regions are buried inside the structure, Figure 2.3. These coiled coils can be detected using profile searching, as in the program also called Coils (Lupas 1996, Lupas 1997). There are two reasons for finding coiled coils. Like low complexity regions, Section 2.2.1, they can have an adverse effect on similarity searches but they can also give an indication of functionality.
2.5 Integrated Systems

Although the systems already mentioned give considerable insight into discovering functional details, an integrated approach would provide a much greater evaluation capability (Medigue et al. 1999) and there has been considerable research done in this area. Here, these strategies are categorised into two groups, that of graphical display of results and automatic annotation.

2.5.1 Graphical Display

As mentioned in Section 2.2.2 the amount of data that can be returned by BLAST or similar programs may be immense, which can impede direct analysis (Farmerie et al. 2003). According to Helt et al. (1998) biology is a visual discipline with information stored in diagrams, graphs or three-dimensional models. Therefore, many systems have been created to resolve the overload problem utilising graphical displays to summarise the data, as described above. Others also incorporate further analysis, increasing the information on which decisions may be based.
Depending on the input format and output required, there are various systems that deal with whole genome analysis or basic sequence analysis. In the former case, all systems studied to date incorporate similarity searches and some form of gene prediction, such as BioViews (Helt et al. 1998) and Imagene (Medigue et al. 1999). Other systems also filter the sequences before using similarity searches, including Alfresco (Jareborg and Durbin 2000) and GAIA (Bailey et al. 1998a). Genotator (Harris 1997) incorporates all the above features plus the ability to use user-defined pattern searches in the results. All of the programs mentioned here display the query as a line with the features aligned to it, Figure 2.4, though both Imagene and Genotator also incorporate multiple sequences aligned to the query sequence.

**Figure 2.4.** Example of graphical display showing predicted features, orange and blue bars, along a query sequence, black line, adapted from Gilbert (2002)

The analysis of basic sequences is more varied. Beauty (Worley et al. 1998, Worley et al. 1995) and Wilma (Prlic et al. 2004) use both homology searches and searches to domain and family databases, such as Prints and Prosite. GeneMachine (Makalowska et al. 2001), ESTannotator (Hotz-Wagenblatt et al. 2003), ViewGene (Kashuk et al. 2002) and PowerBLAST (Zhang and Madden 1997) filter their sequences but ESTannotator uses clustering and GeneMachine uses gene prediction for their further evidence. SeqHelp (Lee et al. 1998) is another system that both filters its sequences and uses gene prediction, and also makes use of the extensive Discovery package of programs (Accelrys 2005), which includes homology searches,
multiple alignments, gene prediction and structure prediction. As with the whole genome analysis, the results are displayed aligned against the query sequence.

Other visualisation packages are used for the graphical display of structural elements, from the domain up to the three-dimensional structure. In the first instance, programs, such as CINEMA (Parry-Smith et al. 1998) and ProteinAnalyst (Saqi et al. 1999), provide a colour-coded representation of a multiple sequence alignment, though ProteinAnalyst also reduces the redundancy in the results by only accepting lower similarity scores and rejecting short hits. Secondary and tertiary structure can be visualised by programs such as POLYVIEW (Porollo et al. 2004) and SRS 3D (O'Donoghue et al. 2004) and ViTO (Catherinot and Labesse 2004) respectively.

The advantage of graphical displays is that all the information can be condensed into one image. Although this may render fine detail invisible, many of the programs incorporate the ability to zoom, such as in Alfresco and Genotator, allowing the sequence to be viewed at different levels of detail. This also allows the quick and efficient discovery of interesting features, with many programs, such as AV (Chi et al. 1995) and GeneMachine, which use interactive images to allow further textual details to be retrieved.

2.5.2 Automatic Annotation

The main goal of sequencing is to identify the function of proteins. However, as outlined in Frishman et al. (2001), it is not always possible to give a definite prediction of what the function of a protein is. However, an initial indication can be proposed as to what the function is most likely to be, based on the current evidence. To this end, several systems have been created that, as well as collecting and displaying relevant information from multiple sources, use this evidence to give the scientist an idea as to the function.

For example, GeneQuiz (Andrade et al. 1999) parses the evidence gleaned from similarity and motif searches, multiple alignments and structural predictions to create a feature table sorted by reliability of source. This is used for rules based reasoning to determine the most likely functional descriptor, the highest scoring descriptor after the application of the rules. MAGPIE (Gaasterland and Sensen 1996) and Gain
(Karaoz et al. 2004) also use a deductive rules based assessment of parsed feature details to assign functionality. These rules are of the form, if A is true then B is true, or if A and B are true then C is true. For example, in simplified terms, there is a coding region if there is both a global similarity and a local similarity (Gaasterland and Sensen 1996).

Another method, used in Pedant (Frishman et al. 2001, Frishman and Mewes 1997, Frishman et al. 2003), in addition to feature characterisation, uses a stringent comparison with the MIPS protein classification database (Mewes et al. 1997) to assign functionality based on comparing a given sequence with all sequences in "curated master gene sets" (Frishman and Mewes 1997). Other systems like CAAT-box (Frangeul et al. 2004), FOUNTAIN (Buerstedde and Prill 2001) and EASY (Selley et al. 2001) base their function prediction on the best hit or the most common descriptor respectively. However, this is could lead to misleading function assignment as the best result does not always equate with the function transferability (Doerks et al. 1998). An alternative would be to retrieve keywords associated with results, such as Anagram (Pérez et al. 2004) assigning the most frequent to the query. Whilst UniBLAST (Zhou et al. 2002) clusters the results by degree of similarity and uses the highest scoring hit in the highest scoring cluster as an indication of function, UPF (Doerks et al. 1998) also clusters the results but by family with the premise that a functional assignment for one member of the family should equate to all.

Other functional prediction programs combine multiple sources of information either weighted as to their relevance, as in MAGIC (Troyanskaya et al. 2003) or GeneMerge (Castillo-Davis and Hartl 2003), or combined with logical operators, such as ProteoMix (Chikayama et al. 2004). The program by Chen and Xu (2004), NetMark (Letovsky and Kasif 2003) and Gotcha (Martin et al. 2004) use the Gene Ontology, GO, terms as a standardised "classification system" (Martin et al. 2004) for each sequence in a form of 'guilt by association'. Finally, some programs are used for the whole process from raw sequence data to functional prediction, such as SABIA (Almeida et al. 2004) and PipeOnline (Ayoubi et al. 2002) or are used to link proteins that already have functional assignment in genomic sequences, such as the program by Pellegrini et al (1999).
2.6 Summary

In the process of function determination of both DNA and protein sequences, there has been considerable research, of which only a small number of the programs produced have been discussed here. However, all the programs mentioned here, provide a guide to functional annotation and the proposed function needs to be confirmed with biochemical experimentation in the laboratory (Bailey et al. 1998a, Bork et al. 1998). That said, in the literature the similarity search is frequently referred to as an important bioinformatics tool or the first step in function determination (Attwood 2000). Therefore, this research centres on the analysis and interpretation of the results returned by a similarity search program, specifically that of the BLAST algorithm. Both BLAST and FASTA, Section 2.2 balance speed with sensitivity, however, BLAST appears the more commonly used (Pearson 1998) and, therefore, it is this program that will be used in this research.
3 SIMILARITY SEARCHING WITH BLAST

As was discussed in Section 2.2, similarity searches are an effective way of discovering functional connections between an unknown DNA or protein sequence and DNA or protein databases. This chapter gives an in depth description of one of the similarity searching methods, namely BLAST, and the layout of the results returned from the DNA and protein databases maintained at NCBI.

3.1 Investigation of the BLAST Algorithm

BLAST (Basic Local Alignment Search Tool) uses heuristic methods in order to find local alignments between a query sequence of DNA or protein and a database of other DNA or protein sequences. These databases have been created and maintained by various corporations and can be found at their sites on the Internet, such as Genbank at NCBI (National Centre for Biotechnology Information) at www.ncbi.nlm.nih.gov/ or EMBL at EBI (European Bioinformatics Institute) at www.ebi.ac.uk/embl/. The local alignments are graded by a scoring system as to the degree of similarity, the similarity score, with a calculated e-value indicating the degree to which a particular hit may have occurred by chance. Since the calculation of the similarity score is virtually the same for DNA or protein searches, except for the specific scoring values used, only the process for protein sequences is described here.

3.1.1 The Similarity Score

Local alignments consist of two aligned sub-sections, one from the query sequence and the other from the hit found in the database, which are of arbitrary but equal lengths. The alignments similarity score is the cumulative score of each of the amino acid pairs, or alignment pairs, within the local alignment. Each individual alignment pair is scored in accordance with a substitution matrix such as Blosum (Henikoff and Henikoff 1992) or PAM (Dayhoff et al. 1978), Figure 3.1. As can be seen from the figure a substitution matrix consists of a 20x20 grid in which each of the amino acids is compared against each amino acid in turn, including itself, (Risler et al. 1988) and is scored as to its suitability for replacement within a protein sequence without affecting the function (Koonin and Galperin 2003, Section 4.2.1). Identical amino acids and conservative amino acids, where a different amino acid is substituted with no effect to the proteins function, have positive scores (bold in Figure 3.1), whereas
those that are incompatible and do affect the function if substituted, have negative scores (italic in Figure 3.1). Dependent on the stringency required for the amino acid substitutions, there are a number of different substitution matrices, where a scientist may opt for more severe penalties for an incompatible amino acid substitution with greater negative values or visa versa.

In this way, when two local alignments are compared each alignment pair (one in the query and one in the database sequence) is assigned a particular score as defined by the matrix and the overall score is the summation of these individual scores. The more identical or conservative amino acids there are in the alignment the greater the score is and, therefore, the more similar the two sequences are. The actual length of the alignment is determined by the similarity score itself, in that extending or trimming the alignment will not improve the score, (Altschul et al. 1997); this is known as a maximal-scoring segment pair, MSP.

**Figure 3.1.** The Blosum62 substitution matrix with an example of its use in calculating a similarity score

|   | C | S | T | P | A | G | N | D | E | Q | H | R | K | M | L | V | F | W |
| C | 9 |
| S | 1 | 4 |
| T | 1 | 5 |
| P | 3 | 1 | 7 |
| A | 0 | 0 | 4 |
| G | 3 | 2 | 0 |
| N | 3 | 1 | 6 |
| D | 3 | 1 | 6 |
| E | 4 | 0 | 2 |
| Q | 3 | 0 | 5 |
| H | 3 | 1 | 8 |
| R | 3 | 1 | 5 |
| K | 3 | 0 | 5 |
| M | 1 | 1 | 5 |
| I | 1 | 4 |
| L | 1 | 2 |
| V | 1 | 2 |
| F | 2 | 2 |
| Y | 2 | 2 |
| W | 2 | 2 |

**Total = 45**
The BLAST algorithm calculates these similarity scores in three steps (Altschul et al. 1990, Altschul et al. 1997). Firstly it breaks the query sequence up into segments, or words, for searching, hits to these words are then found in the database and lastly the hits are extended to produce the completed MSPs.

As local alignments are segments of the overall sequence, so too, the words are very small segments of the alignment, only a few residues in length, and are known as w-mers (Altschul et al. 1990). These w-mers have a fixed length of w (usually 3 for proteins) and the BLAST algorithm finds all w-mers that score a minimum of a predefined value $T$ (default of $11^2$) when compared against w-mers from the query sequence and scored using one of the substitution matrices. For example, in Figure 3.2 a list of all possible words scoring over 13 when compared against the w-mer PQG are listed, termed neighbourhood words.

The BLAST algorithm takes this adjusted list of w-mers and searches the database for exact hits, as in the lower half of Figure 3.2. If two non-overlapping hits are found within a distance of $A$ residues of each other (default 40), then the segment is extended in both directions to form the MSP, Figure 3.3. As it is extended the

---

2 All default values quoted here are taken from the summary statistics at the bottom of an actual BLAST search conducted by the author.
substitution matrix score for each alignment pair, positive or negative is added to the cumulative score, $S$. Once the score has fallen $X$ (default 16 or 38) below the best score obtained for that alignment so far, the extension is discontinued. However, the alignment will only be reported if the overall score is higher than another predefined threshold (default 42), termed Tot here.

![Diagram](image)

**Figure 3.3.** Extension of database hits within $A$ residues

Lowering the $T$ value increases the chances that a database sequence will contain a $w$-mer of score $Tot$ or more but it also means that there will be more hits, taking more time and memory to execute the search. Altering the values of $w$ and $Tot$ also affects the execution time. Therefore, a balance must be maintained between these three parameters to give a high sensitivity but also a rapid execution time (Altschul et al. 1990).

The cumulative score, $S$, indicates the degree of similarity within a local alignment between a query sequence and a sequence from a database. However, this score is dependent upon which of the various versions of scoring matrix is used and it is not possible to compare results without knowing the details of the matrix (NCBI no date-a). It is, therefore, necessary to normalise the score using

$$S' = \frac{\lambda S - \ln K}{\ln 2},$$

where $\lambda$ and $K$ are calculable parameters (Karlin and Altschul 1990, Karlin et al. 1990). This gives a 'bit score' $S'$ with standard units enabling different scores to be directly compared, (Altschul et al. 1997).
3.1.2  The E-Value

The similarities between a query and a database sequence may give indications of the biological function of the query sequence. However, there is always the possibility that any of the hits might have occurred by chance and it may only be a coincidence that the two sequences under comparison are similar. In this case there is no biological relevance and that particular hit could be deemed irrelevant. Therefore, it is useful to know how many hits are expected to occur by chance. In BLAST this is done using the E-value, with the higher the E-value the more likely that the sequence could occur by chance and the less likely it is an actual homology. The E-value can be calculated using

\[ E = K \frac{m n e^{-S}}{2^S}, \]  

where \( m \times n \), \( n \) being all the sequences in the database, is the search space size (Altschul et al. 1997) and is the product of the lengths of the two sequences under comparison, \( S \) is the score and \( K \) and \( \lambda \) are Karlin-Altschul statistical parameters related to the scoring matrix used. The two constants, \( K \) and \( \lambda \), can be estimated, though in typical matrices \( K \) is 0.1 and \( \lambda \) is 1, so that the magnitude of any score is determined by \( m \times n \) (Altschul 1991). However, once the score has been normalised using equation 3.1 the bit score takes the statistics of the scoring matrix into account and, therefore, only the search space size and normalised score are needed to calculate the E-value

\[ E = mn 2^{-S}. \]  

The probability, termed P-value in the literature, of obtaining a hit by chance can also be calculated using the equation:

\[ P = 1 - e^{-E}, \]  

where \( E \) is the E-value calculated in equation 3.2. However, the BLAST program reports the E-values because “it is easier to understand the difference between, for example, E-value of 5 and 10 than P-values of 0.993 and 0.99995” (Altschul no date).

3.1.3  Gapped Sequence Searches

A gap occurs in an alignment when either of the contributing sequences is completely lacking an amino acid at an alignment position, Figure 3.4. The calculation of the similarity score and the E-value described above (Sections 3.1.1
and 3.1.2) were all developed using ungapped alignments. However, ungapped BLAST searches can often overlook biologically relevant hits, due to mutational difference or evolutionary changes generating gaps of several residues that do not match within alignments. Also, several ungapped alignments in a sequence may be biologically relevant when considered together and if even one were missed the whole hit may erroneously be discarded (Altschul et al. 1997). With a gapped alignment, only one of the ungapped alignments needs to be found and extended through the gaps. Experimental data (Altschul no date, Altschul et al. 1997) suggest that the methods and equations for ungapped alignments will work for gapped alignments as well, as long as there is a penalty for the inclusion of a gap.

![Figure 3.4. Example of a gapped alignment for a DNA sequence (Madden 2003)](image)

The scoring of gapped alignments follows the same method as for ungapped alignments but with an additional step. Once the alignment has been extended, if its score exceeds a threshold $S_g$ (default 68), then a gapped extension will also be triggered. There is a score penalty for starting a gap and an additional lower penalty for continuing a gap (default 11 and 1 respectively), known as affine gap costs (Altschul et al. 1997). However the extension will proceed in the same manner as for ungapped extensions, extending the alignment until the score drops $X_g$ (default 64) below the best score. Therefore the addition of gaps may give lower scores but may produce a more biological relevant result.

### 3.2 The BLAST Programs and Layout of the BLAST Results

There are various different BLAST programs available, the choice of which is dependent on three factors, “1) the nature of the query, 2) the purpose of the search, and 3) the database intended as the target of the search” (NCBI 2005a). The five main programs described in the BLAST Help Manual (NCBI no date-a) are:
• BLASTp – comparing a protein query against a protein database
• BLASTn – comparing nucleotide query against a nucleotide database
• BLASTx – comparing a six-frame translation of a nucleotide query against a protein database
• tBLASTn – comparing a six-frame dynamic translation of a protein query (both strands) against a nucleotide database
• tBLASTx – comparing a six-frame translation of a nucleotide query against a six-frame translation of a nucleotide database

In addition, there is also PSI-BLAST for more sensitive BLAST searches and PHI-BLAST for searching for specified patterns or motifs within a protein. This research centres on the results obtained from BLASTp comparisons.

Once the BLAST algorithm has calculated the similarity scores and e-values for any hits between the query sequence and the database, this information is returned to the scientist. This file, which can either be viewed on a web page or received via e-mail has five main sections (NCBI no date-a). The first of these gives details of which of the BLAST programs were used, its version number and release date. This is followed by a reference to literature on the BLAST program, a query ID, the length of the query sequence and details on the database, Figure 3.5.

A graphical display visualising where the regions of similarity are, follows this, Figure 3.6. The key at the top shows differing colours for varying levels of similarity, black being least similar and red being most similar. It also shows where on the test sequence the hit actually is, which can be useful if the sequence contains a conserved region.
BLASTP 2.2.12 [Aug-07-2005]

Reference:

RID: 1126104931-27495-174712434326.BLASTQ3

Database: All non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF excluding environmental samples 2,822,727 sequences; 968,621,020 total letters

If you have any problems or questions with the results of this search please refer to the BLAST FAQs

Taxonomy reports

Query=  
(408 letters)

Figure 3.5. Introductory section of the BLAST result file

**Distribution of 32 Blast Hits on the Query Sequence**

![Color Key for Alignment Scores]

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;40</td>
<td>Dark</td>
</tr>
<tr>
<td>40-50</td>
<td>Medium</td>
</tr>
<tr>
<td>50-80</td>
<td>Light</td>
</tr>
<tr>
<td>80-200</td>
<td>Strong</td>
</tr>
<tr>
<td>&gt;=200</td>
<td>Bold</td>
</tr>
</tbody>
</table>

Figure 3.6. BLAST hits distribution graph

The next section contains a list of the hits to the database, where each hit consists of which database the hit is from, two identifying accession numbers, the hit’s description, score and E-value, Figure 3.7.
Sequences producing significant alignments:

<table>
<thead>
<tr>
<th>gi</th>
<th>Score</th>
<th>E Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>gi</td>
<td>21593241</td>
<td>gb</td>
</tr>
<tr>
<td>gi</td>
<td>9955531</td>
<td>emb</td>
</tr>
<tr>
<td>gi</td>
<td>31074969</td>
<td>gb</td>
</tr>
<tr>
<td>gi</td>
<td>3759184</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>21593043</td>
<td>gb</td>
</tr>
<tr>
<td>gi</td>
<td>7267537</td>
<td>emb</td>
</tr>
<tr>
<td>gi</td>
<td>21537055</td>
<td>gb</td>
</tr>
<tr>
<td>gi</td>
<td>50912931</td>
<td>ref</td>
</tr>
<tr>
<td>gi</td>
<td>50912931</td>
<td>ref</td>
</tr>
<tr>
<td>gi</td>
<td>51535371</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>51535371</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>50912937</td>
<td>ref</td>
</tr>
<tr>
<td>gi</td>
<td>51964936</td>
<td>ref</td>
</tr>
<tr>
<td>gi</td>
<td>50912931</td>
<td>ref</td>
</tr>
<tr>
<td>gi</td>
<td>13194680</td>
<td>gb</td>
</tr>
<tr>
<td>gi</td>
<td>51535376</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>5196361</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>51535378</td>
<td>dbj</td>
</tr>
<tr>
<td>gi</td>
<td>177214</td>
<td>gb</td>
</tr>
</tbody>
</table>

**Figure 3.7. Summaries of BLAST hits**

Below this are the actual alignments for each hit, showing the areas of similarity between the query sequence and its database match, Figure 3.8. Each one also includes the full description, the length of the matching sequence, the score and the e-value. It also includes the number and percentage of identically and positively matched residues and mismatched residues, or gaps, if any.

**Figure 3.8. Alignments between query sequence and matching database sequence**

<table>
<thead>
<tr>
<th>Query</th>
<th>Score</th>
<th>E Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;gi</td>
<td>31074969</td>
<td>gb</td>
</tr>
<tr>
<td>Length</td>
<td>301</td>
<td></td>
</tr>
<tr>
<td>Score</td>
<td>263 bits (673), Expect = 1e-69</td>
<td></td>
</tr>
<tr>
<td>Identities</td>
<td>139/200 (69%), Positives = 162/200 (81%), Gaps = 6/200 (3%)</td>
<td></td>
</tr>
<tr>
<td>Query: TTEKYKGGSSTLVUGKQLLENYPLGKSLNPYLRALSTKLNGGLRSITTVLTA KDVTE 60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sbqt: TTEKYKGGSSTLVUGKQLLENYPLGKSLNPYLRALSTKLNGGLRSITTVLTA KDVTE 60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query: RFCMRSGTGFGSGSPNPRAANAAYVVVGNSETQCPCPGYCAWPFHQ1YGPFTPLVPAN 120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sbqt: RFCMRSGTGFGSGSPNPRAANAAYVVVGNSETQCPCPGYCAWPFHQ1YGPFTPLVPAN 120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query: GDVGVDGMIINLALTALLANTVTFNFFNYQGPTAPELASVACPGIFGS GSYGGYAGRWL 180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sbqt: GDVGVDGMIINLALTALLANTVTFNFFNYQGPTAPELASVACPGIFGS GSYGGYAGRWL 180</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query: 181 VDKTTGGSYNARGLAGKRYL 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sbqt: VDKTTGGSYNARGLAGKRYL 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query: 206 VDKTGASYNAR NVRFL 285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sbqt: VDKTGASYNAR NVRFL 285</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Finally, there is a summary of the statistics generated during the calculation of the similarity score and e-value.

This list in itself does not give sufficient detail to determine whether any one particular hit is of particular relevance and often the description is truncated, as can be seen in Figure 3.7. However, further details (the annotations) stored in the database can be accessed using the accession numbers referred to in each hit. Even so, there are certain issues to take into consideration when analysing the BLAST results and these will be discussed in the next Chapter.
4 BLAST ANALYSIS

The previous Chapter described how BLAST finds similar sequences to a test sequence giving these alignments a similarity score and e-value. However, since the purpose of BLAST is to assist in biological function determination of a query sequence it is only the initial filter enhanced by other tools, such as was described in Chapter 2. As a filter BLAST is a powerful tool for function determination, nevertheless, there are a number of issues that need to be taken into consideration with respect of the analysis of the results returned.

4.1 The BLAST Analysis Issue

Here, these have been divided into three broad categories of data overload, lack of data details and data relevance. As a result of these it is not always obvious what the possible function of the query sequence is based on the BLAST results returned (Worley et al. 1998). Each of these issues will be discussed below, after which, the methodology for finding a solution to these problems will be examined.

4.1.1 Data Overload

The first of these issues is that of data overload. Considerable improvements in technology, such as automated sequencers and mass spectrometry (Attwood and Parry-Smith 1991) have meant that sequencing proteins has become faster and less expensive. As a result of this, the number of sequences being searched against the databases and those entered into the databases is increasing exponentially. For example, the number of sequences in the Genbank database (NCBI 2005e) has risen from 606 in 1982 to 40,604,319 in 2004, Figure 4.1 and it is estimated that the sizes of the databases are doubling every nine months (Miller and Attwood 2003).

With this information explosion the number of search hits that are returned is also rapidly increasing in most cases, though the current maximum the NCBI BLAST site can be set to is 20,000 descriptions and/or alignments (NCBI no date-b). For that reason, to manually perform detailed analysis of this data is becoming increasingly impractical (Berman et al. 2000b, Shoop et al. 1995). Also, there is the possibility that more relevant hits may become buried amongst a large number of less relevant hits (Altschul et al. 1994). For example, modern sequencing machines, such as the
ABI PRISM® 377 DNA Sequencer (AME Bioscience 2004) can produce up to 96 sequences in a single run, each of which having a possible 20,000 results.

Figure 4.1. Graph illustrating the exponential growth of sequence entries in the Genbank database from 1982 to 2004, (NCBI 2005e)

4.1.2 Lack of Data Details
Secondly, is the issue of the details, or lack of them, that are present in the BLAST results. When viewing the hits list alone, the matching protein may be uncharacterised or its definition may be uninformative or ambiguous (Worley et al. 1995) or even contradictory (Attwood 2000). Also the definition may be truncated (Zhang and Madden 1997) and followed by ellipsis, as in the results in Figure 3.7 (page 38). Although the complete definition can be found with the corresponding alignment further down the results file this would not alleviate the problem of the definition content. Further details are stored in the source databases in an annotation file that is accessed using the accession numbers in each hit. However, reading
through each annotation file may be time-consuming to perform manually, especially if there are many hits to analyse, each with their own annotation file.

There is also no guarantee that there will be any meaningful details in the annotation file. For example, for a completely unique sequence there may only be that the source of the sequence has been entered into the annotation file. Even when details are known and have been added it may be time-consuming to find them due to the format of the annotation file. The annotation file is split up into fields including its identification number, organism source, any references using this sequence and the sequence itself. There is also a feature table specifying where and for how many residues known protein features, such as active sites, are situated along the sequence. Although many important details can be found in their own fields, additional comments may be added to a comments field, which has no set format but may contain relevant information. Also, all the fields are written in natural language making them difficult to parse (Ruepp et al. 2004) with the risk that the same function may be defined in different ways in different annotation files (Martin et al. 2004). Programs such as Easy and AnaGram, described in Section 2.5.2 go some way to solve this issue and this will not be dealt with in this project.

4.1.3 Data Relevance

As well as a lack of details, there is also the matter of the relevance of the details that are there. Even if a particular result has been annotated it may not actually be relevant to the scientist's research.

The amino acid sequence of a protein determines the distinct three-dimensional shape of that protein (Rost 1999). This, in turn, directs the protein function since “proteins function as three-dimensional objects” (Ginalska et al. 2005). Provided functionally, and structurally, important amino acids are conserved the overall composition of the sequence can vary with more distantly related proteins being more divergent than closer relatives (Letunic et al. 2002). However, the amino acids can diverge beyond the ability to identify proteins of similar function by sequence similarity alone (Falquet et al. 2002). For these, programs such as those described in Section 2.3 are valuable in identifying the conserved domains in these more distantly related proteins.
Then again, even with the identification of domains, due to the multi-domain nature of proteins it may not be clear to which of the motifs the assigned function relates (Attwood 2000). For example, in Figure 4.2, if the functional annotation related to domain 2 it would not be possible to transfer this annotation to the query sequence, which aligned to domain 1. In these cases it is necessary to confirm the hit location by studying the hit alignments. Also, even if the correct domain has been aligned, dependent on the biological context the actual function may differ (Attwood 2000).

![Figure 4.2. Schematic representation of domain alignments between two sequences A and B](image)

There is also the consideration that the functional annotation of a protein may, itself, have been inferred from a similarity search. Due to the reasons highlighted above, this inference may be incorrect and if used as the “basis for subsequent predictions” (Koonin and Galperin 2003), this error could be propagated through a database. These inferences are also not labelled as such (Bork et al. 1998); consequently there is no way of distinguishing them from experimentally proven functional annotations.

4.1.4 Automation

To alleviate the issues described above (Sections 4.1.1 to 4.1.3) it would, therefore, be desirable to automate the analysis of the BLAST output. In the first instance, computers could quickly run through the large number of hits collecting the full definition lines from the alignments and any further details from the annotation files. For the issues with quantity and quality of the data present further searches could easily be conducted in other bioinformatics resources, such as in the integrated programs described in Section 2.5. Additionally, crucial information can be gleaned from the literature, either cited in the annotation file or by a general search in the literature, such as in the MedBLAST program (Tu et al. 2004). However, it is preferable for the final decision making to remain in the hands of the scientist to
ensure that an incorrect inference of function is not propagated (Attwood 2000). To that end, research was conducted through knowledge elicitation to ascertain how the scientists currently conduct their manual BLAST.

4.2 The Current Manual System
In order to convert a manual operation into an automatic system, it is first necessary to understand exactly how the process is currently accomplished.

4.2.1 Knowledge Elicitation
Knowledge elicitation is the “process of extracting, structuring, and organising knowledge from one or more sources” (Turban and Aronson 1998). These sources could be written, as in documentation and literature, visual, as in diagrams and graphs, or verbal, as in described by a person or people. However, in those situations where the process to automate is the decisions of an ‘expert’, the best source is the experts themselves whose experience gives them a unique perception of the situation. Here the problem is to find the most effective means to get the expert to divulge the relevant information. There are many methods by which knowledge can be gathered; a list of many may be found in Cordingley (1989), with each method having their own merits and flaws.

However, Lafrance (1987) says knowledge elicitation is time-consuming, painstaking, complicated and a bottleneck to the automation process. One reason for this is that knowledge is buried deep in the expert’s memory (LaFrance 1987) and accessed unconsciously (Diaper 1989). Therefore, experts often have difficulty verbalising this knowledge (Diaper 1989, Turban and Aronson 1998) and even what they can say may be corrupted due to memory lapses (Diaper 1989). Lafrance (1987) also mentions that the more ‘expert’ the expert is, the less likely they are to be able to articulate their knowledge.

Background reading in the area of interest, on the part of the person eliciting the knowledge, can give an overall view of the domain and highlight particular issues of relevance to the problem in hand (McGraw and Harbison-Briggs 1989). However, care should be taken to ensure that this reading is only for an overview and does not
lead to the mistake of talking like an expert but thinking like a novice (Greenwell 1988), which can confuse matters.

4.2.2 Methods Used

To evaluate the BLAST analysis, a series of informal and semi-structured interviews were arranged, which were coupled with background reading to acquire an understanding of the manual process currently utilised. The basic idea was to allow each interviewee to discuss, from their own perspective, the steps they would take in analysing the data in BLAST results and the annotation files.

To put the interviewees at ease, the interviews took place in either their own office or a meeting room they were familiar with. An additional advantage to the former location was that they had all their material at hand and could easily find examples to illustrate relevant points. To keep details of each interview three recording methods were used:

- The interviewer took brief notes on the salient points in the discussion
- Audio recordings were made using firstly an analogue Dictaphone then a digital Dictaphone, as it was found that the sound quality was insufficient with the analogue Dictaphone
- A secondary interviewer was present to listen and take extensive notes on the discussion

The use of the Dictaphone and secondary interviewer enabled the primary interviewer to give the interviewee their full attention only noting down points of particular relevance. A digital Dictaphone has a far superior sound quality to an analogue version; therefore to improve the efficiency and speed of transcribing the tape and analysis of the data collected all but the first two interviews were recorded using a digital Dictaphone.

The first interview conducted gave a general overview of the process of BLAST analysis and subsequent interviews delved in more detail in certain areas. This also gave a broader view on the different biological areas in which BLAST is used, specifically in the fields of Bioinformatics, Embryo genetics, Plant pathology, Plant virology and Soil micro-ecology.
Secondary interviews were conducted to enable clarification of the BLAST analysis process. These took the form of a talkback with the interviewer reiterating the salient points of the previous interviews. These ensured that the interviewer had understood the interviewee and that no details had been missed. By this time a proposed automation had been postulated, on which feedback as to usability and ideas of functionality could be gained.

4.2.3 Outcome

The analysis of the interviews revealed that there is a number of key features in the BLAST result that the scientists would use in determining whether a particular result was of relevance to their work or not. The key areas of the BLAST results used in these determinations were the graphical display of the distribution of the results along the query sequence, Figure 3.6 (pp. 37); the description and score and e-value, Figure 3.7 (pp. 38); and the alignments between the test sequence and particular result found, Figure 3.8 (pp. 38). A summary can be found in Figure 4.3. As can be seen from the diagram, further details are extracted from the annotation files for the purposes of the decision process. However, this can be time consuming, especially on large result files, Section 4.1.2, and in this research, only the analysis of the actual BLAST result will be considered.
4.2.3.1 Graphical Display

The graphical display, Figure 3.6 (pp. 37), displays each of the hits of the BLAST result lined up to the query sequence as to their region of similarity. This is advantageous as it displays, at a glance, the areas of the query that have equivalent hits (Lyon 2004) and how much of the query is covered by hits in the database, the query’s coverage. It is preferable that any hit returned by BLAST is as long as possible, since the longer the hit, the less likely it is to have occurred by chance (McMurran 2004). Therefore, looking at the graphical display allows a rapid check of the coverage of the test sequence by the BLAST hits. However, if the hits do not cover the entire test sequence they may be localised in one area, which would indicate a conserved region, possibly an active site (Marshall 2001), which would also be evident in the graphical display much like in the multiple alignments described in Section 2.3.
Another advantage of the graphical display is that it may reveal possible sequence contamination (Lyon 2001). Sequence contamination occurs when elements used in the cloning of DNA remain in the sequence during searching, such as vectors or adaptors. It is advisable to remove these as they can produce irrelevant hits to the section of contamination sequence rather than to the desired query sequence. If there are very short hits at the start or end of the query sequence, as circled in Figure 4.4, it could mean that not all of the vector or adaptor sequence had been removed prior to the search. This could be further verified by checking the alignments of these hits since the matching sequences would all be virtually identical.

Figure 4.4. Example of sequence contamination showing short matches to vector sequence, circled. Generated by adding a short piece of vector sequence to the end of a sequence used in a BLAST search

The display is also ‘clickable’, each line in the display being hyperlinked to the alignments they represent allowing easy access to interesting looking alignments to check details, such as, if it is vector sequence. However, since the display only shows the hits as a line against the query sequence, other than the benefits outlined here, there is very little detail in the display itself as to any indications of possible function.
4.2.3.2 Description, Score and E-value

In the actual hits list there are a number of features that help in relevance determination, though there is no set priority to the order in which they are considered.

The actual value of the scores and e-values is very important (Lyon 2001, Marshall 2001). The score quantifies the similarity between two sequences. The higher the score, the more closely related the sequences may be and the more likely any functional annotation will be transferable. A scientist may have a score threshold value below which they consider a particular hit useless, though this may be flexible if the best there is “grey” (Marshall 2001). The e-values quantify the chance of obtaining a hit by chance, the lower the e-value, the better. Therefore, once the e-value becomes too high the hit is less likely to be a true similarity and increasingly a chance association. However in both cases it is the choice of threshold that is the unknown, “at what point of this scale is the sequence different” (Lyon 2001).

In the descriptions the scientists are looking for consistency in keywords (Daniells 2001, Lyon 2001, Marshall 2001). These keywords include any protein, gene and enzyme names. If these keywords appear in many of the hits down the list, this may give increased confidence that this is what the query sequence may be. For example, if the majority of the hits in the list state that the protein function is a protein phosphatase (Lyon 2001), it gives greater confidence that the query sequence will also be a protein phosphatase.

With the species name, certain species are more closely related, such as potato and tomato (Marshall 2001), therefore, as well as a hit to the same species, a hit to a closely related species may also give increased confidence in the relatedness of the matching sequence. A hit to the same, or a similar species, or to the same region gives increased confidence that the same functional region has been identified. Also, if the hit is to a similar species that has been completely sequenced, then it is possible to anchor the query sequence in the genome. In doing so it is possible to make inferences regarding the function of not only the current query sequence but to neighbouring regions of the genome from which the query came, Figure 4.5.
Different proteins are conserved to varying degrees, for example, the histone proteins are very highly conserved across taxonomic boundaries (Marshall 2001). Therefore, a good hit to a highly conserved protein gives a high confidence for functional assignment. Less conserved proteins may have conserved domains or active sites (Daniells 2001, Lyon 2001, Marshall 2001). The identification of domains may provide useful functional insights into the query protein. For example, in viruses, Leucines are known to be required for protein-protein interaction, so a hit to a Leucine rich region may provide clues as to how the virus functions (Taliansky 2001). It would, therefore, be of benefit to have some pre-knowledge of the degree of conservation of particular proteins (Marshall 2001).

Other keywords of interest are the appearance of “mRNA” or “EST”, which would indicate that the hit was within a coding region (Lyon 2004, Marshall 2001). Since it is only the coding region that translates into the protein, references to the coding region increases the confidence in that particular hit. Also, the amino acid code defines the three-dimensional structure of the protein, which is how it accomplishes its function (Lyon 2001). Therefore, any structural keywords would be highly relevant.

4.2.3.3 Alignments

The alignments show where in the sequences the actual regions of similarity are. In highly similar hits there will be very few gaps and these would naturally be considered relevant. However, in hits with lower scores and/or more instances of gaps within the alignments the distribution of the matches within the alignment could be of significance. For example, if there is a hit to a weakly conserved protein then a clustering, or clumping, of residue matches in the alignment could indicate where an
active site may be, which would also be of interest (Lyon 2001, Taliansky 2001). To be relevant these clumps need not be very large, and can be as little as five amino acids long (Lyon 2001). Also, if this same ‘clump’ was found in other hits in the list that would give increased confidence in it being an active site (Marshall 2004).

4.2.3.4 Other Issues

There are several other issues of importance in the determination of relevance of any particular hit in the BLAST results. With the different BLAST programs available, if searches were conducted on each of them and similar results were returned this would, again, give increased confidence that the any functional elements within that hit could be transferred to the query sequence (Lyon 2001, Marshall 2001). Finally, there is the consideration of whether any functional annotation within a particular hit has been biochemically tested (Daniells 2001, Lyon 2001, Marshall 2001). The use of the BLAST similarity search is as a first step to give an indication of function, which must then be confirmed in the laboratory (Daniells 2001). If this is not the case and any functional assignment is based on inference from a previous BLAST search then a chain of inference situation could arise as described in Section 4.1.3.

4.2.4 Annotation Files

Once a result is deemed relevant, further details can be obtained from that particular hit’s annotation file linked to from the BLAST results via the associated accession number. Again there is a list of features that the scientist will look for in determining if the extra details increase or decrease the confidence in the chosen hit. As in the BLAST results file it is necessary to check for keywords, relatedness of species and contamination. Additionally, the presence of publications (Taliansky 2001), cross-referencing and any translations can give further information, though care is needed if the translation is putative. Details of the sequence’s orientation, from which end the DNA was sequenced, may also be found here with poly T heads and poly A tails, giving increased confidence if the sequence is in the same direction as the query sequence. This is important for the functional transferability since, if the matching sequence is in the opposite orientation to the query sequence it is likely that only a domain homology has been found rather than a complete gene (Marshall 2001).
4.3 Existing Approaches to Feature Automation

As described in Chapter 2, there are many programs available to automate the features described above:

- Contamination – the masking programs described in Section 2.2.1 should remove any remaining vector sequence or other forms of contamination, thereby ensuring that any hits found are hits to the desired query sequence rather than a common element. Additionally, since these programs are primarily concerned with DNA sequences and this research with protein sequences, contamination is not of as great a concern in this research.

- Score/e-value – the use of thresholds in the score and e-value may only be a time-reducing step, or they may be used to remove sequences that are less likely to be true homologies, such as in MAGPIE (Section 2.5.2), which uses cut-off values below which the results are rejects.

- Keywords in the descriptions – programs such as those described in Section 2.2.2 parse and summarise the different keywords in the descriptions allowing for a rapid inspection of what has been returned and how often it appears in the results.

- Species – the BLAST report itself contains a link to taxonomy reports, Figure 3.5.

- Anchoring – given a list of the completed genomes (NCBI 2005c) any species names in the keywords could also be checked against this list allowing for the determination as to whether a query sequence can be anchored in a genome and which genome.

- Conserved proteins – the use of the multiple sequence alignments and motif prediction programs, Section 2.3, identifies conserved proteins and active sites within a protein sequence.

- Coding region – the use of gene prediction, Section 2.1, may lead to the identification of coding regions and start/stop codons in a DNA sequence.

- Structure – there have been many programs designed to predict structural elements from a protein sequence.

- Domains in alignments – any of the multiple sequence and motif prediction software, mentioned in Section 2.3, would be able to identify possible motifs. However, a good selection of high quality sequences may not be available for
the multiple alignment or there is the risk that the quality of the multiple 
alignment may be less than desired for accurate analysis. Additionally, motif 
searches may miss highly relevant but distantly related proteins

- Biochemistry – frequently in the literature it is cited that these methods are 
only guides for the biochemistry, for example (Andrade et al. 1999, Bailey et 
al. 1998b, Bork et al. 1998), and there is currently no tag in the annotations to 
indicate that a functional assignment has been backed up in the laboratory 
(Bork et al. 1998). However, the idea of this research is to give the scientist 
an additional clue as to the function and, therefore, this issue will not be dealt 
within this research

- From annotations (as well as keywords, contamination, species)
  
  o Orientation – again, gene prediction programs, Section 2.1, may be 
  used to determine a sequences orientation. Alternatively searching for 
a poly A tail or poly T'head would indicate the orientation
  
  o Literature – programs such as MedBLAST, Section 4.1.4, may be 
  used to extract beneficial articles related to the matching sequence
  
  o Cross-referencing - the use of the different BLAST programs may be 
  used to ascertain if the same or similar hits have been found, such as 
  the programs described in Section 2.2.2
Table 4.1. Comparison of BLAST analysis features identified in manual work against possible solutions proposed by other research

<table>
<thead>
<tr>
<th>BLAST feature</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reducing the affect of contamination</td>
<td>Masking programs, Section 2.2.1</td>
</tr>
<tr>
<td>Score/E-value</td>
<td>Some programs use cut-off values to reduce the number of hits, Section 2.5.2</td>
</tr>
<tr>
<td>Multiple instances of keywords</td>
<td>Keyword extraction programs, Section 2.2.2</td>
</tr>
<tr>
<td>Similarity of species</td>
<td>BLAST taxonomy reports, Figure 3.5</td>
</tr>
<tr>
<td>Anchoring (genome hit)</td>
<td>Using species keyword with reference to a completed genome list</td>
</tr>
<tr>
<td>Conserved protein domains/ active sites</td>
<td>Domain and family prediction programs, Section 2.3</td>
</tr>
<tr>
<td>Hit to coding region</td>
<td>Gene prediction programs, Section 2.1</td>
</tr>
<tr>
<td>Protein structure references</td>
<td>Structural prediction programs, Section 2.4</td>
</tr>
<tr>
<td>Distribution of matches in alignments</td>
<td>Multiple alignment sequence programs and motif search/prediction programs, Section 2.3</td>
</tr>
<tr>
<td>Grounded in biochemistry</td>
<td></td>
</tr>
<tr>
<td>Sequence orientation</td>
<td>Gene prediction programs, Section 2.1, or searching whole sequence from annotations for poly A/T</td>
</tr>
<tr>
<td>Literature</td>
<td>Literature extraction programs, such as Med BLAST, Section 4.1.4</td>
</tr>
<tr>
<td>Cross-referencing</td>
<td>Use of multiple BLAST programs, Section 2.2.2</td>
</tr>
</tbody>
</table>

The summary indicates that the majority of the features used by the scientists in their decision-making have already been incorporated into some form of automation. There has been considerable work in the field of analysing the alignments of protein sequences in the form of multiple alignment or motif searches. However, it has been postulated that a motif search may overlook highly relevant but distantly related sequences (Attwood 2000) and the construction of a multiple alignment may have inaccuracies that result in invalid conclusions (Do et al. 2005), though research continues in these fields ever improving these methods.

Whereas multiple alignments would consider all available alignments and a motif search would consider the query sequence in isolation, this research will centre on the analysis of individual alignments in isolation. Specifically looking at the distribution of the matches in the alignment and to what degree these are clustered together. The idea being that those sequences with a higher degree of clustering are
more likely to contain domains or motifs than those with a lower degree of clustering. This degree of clustering would be quantified as a clumpiness value with the higher the value, the higher the degree of clustering. Clearly, highly similar hits in the BLAST result list would not require further analysis of this kind, especially those of close to 100% identity, as they would already be of interest to the scientist. Plus, of course, highly similar hits would mask any regions of clustered matches in their alignments simply because there would be few or no gaps to delineate the clusters. This further analysis would be of more relevance to the lower hits where there is less similarity and a greater likelihood of biologically relevant sequences being lost amongst similar but biologically irrelevant hits, as mentioned in Section 4.1.1. The scientist would be presented with this clumpiness information, in addition to the similarity information already provided by BLAST, allowing for a more informed decision regarding individual BLAST hits.
5 THE NOTION OF CLUMPINESS

As mentioned in the summary of desired features against current work (Section 4.3), most of the features have already been automated in various ways. For example, the Anagram program mentioned in Section 2.5.2 retrieves keywords associated with any hits and displays them in a tabulated form, with a histogram of frequencies. In this way, the scientist may quickly perceive what and how many functional keywords have been returned. Though in the Anagram program this has been applied to motif searches, it is feasible to apply this keyword retrieval to a similarity search as well. The evaluation of the distribution of matches within the alignment is used for the discovery of conserved protein domains or active sites. One way to automatically find these is by a motif search (Section 2.3), either by searching against a motif database or using known motif patterns to predict the presence of similar patterns in other sequences. However, in both cases, these require a motif to be known. If there is currently no instance of a particular motif it will not be possible to apply either of these methods to a sequence to find possible motifs. This could result in biologically relevant sequences being overlooked, (Attwood 2000). Additionally, most motif programs use a multiple sequence alignment to predict the presence of possible motifs; however, as mentioned in the previous Chapter, there may be inaccuracies in the multiple sequence alignment leading to flawed conclusions being drawn about the motif content of a sequence.

That said, these motif-searching methods would be an excellent additional step to combine with the similarity search, such as in EASY, mentioned in Section 2.5.2, which, similarly to Anagram, extracts descriptor lines from searches but for both sequence and pattern databases. However, this research focuses on the alleviation of the workload associated with the analysis of the BLAST results in isolation, specifically, the identification of alignments that possibly contain motifs. In this way, those results that may contain motifs would be highlighted and, if a motif search/prediction program highlighted the same results, this would give greater confidence in the assessment of that particular result. Alternatively, the highlighting of possible motif containing alignments may discover a relevant result that the motif search/prediction programs may have overlooked. Therefore, an additional function that would highlight those hits whose alignments are more likely to contain motifs would be advantageous.
An alignment more likely to contain motifs is more likely to contain matches in clusters or clumps. Therefore, this project focuses on the discovery of a measure of the degree of clumpiness within an alignment. This measure could then be used to reorder the BLAST results so as to bring those hits that have greater clustering and, therefore, possible motifs, to the attention of the scientist sooner and assist with their decision making.

As well as the defined purposes described in this project it is also envisioned that this quantification could be generalised into other fields. The presence of a match/mismatch in a similarity alignment could be generalised to the occurrence of discrete event/non-event in a one-dimensional field, therefore, allowing the use of this method in other fields of research, as will be discussed later, Chapter 10.

5.1 Definition
Within the focus of this project, clumpiness refers to the distribution of matches or positives in an alignment produced by BLAST between a query sequence and a database hit. The quantification of clumpiness measures this distribution such that alignments that have a higher degree of matches clustered together in groups would produce a higher clumpiness value as opposed to those where the matches were dispersed evenly along the length of the alignment.

It must be noted that for the direct comparison of the clumpiness from two alignments both must have the same length and the same number of matches, i.e. the same intensity. If this were not the case, then the comparison would be by the degree of similarity as is the case in the BLAST program. For example, in Figure 5.1.a where both alignments have the same length, more of the matches in the second, blue, alignment are clumped together, though this is only because there are many more matches, therefore, making the second alignment more similar rather than clumpier. In Figure 5.1.b, although the number of matches is the same, the lengths of the two alignments are clearly different. Thus, though the matches in the second, blue, alignment are more clustered this is merely because there is less space in which they can be spread resulting in a higher similarity value. Therefore, in any quantification of clumpiness it will be necessary to 'normalise' the value to take into
consideration the differences in the alignment intensity and alignment length that is inherent in the BLAST result, being dictated by degree of similarity rather than any outside factor. This will be considered in more detail in Appendix E.

\[
\begin{align*}
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV \\
TVRLYRGNSNTVLVSLGSEFHDSTSHSGSSVQSVIRALTSLKPNAVNGLYLV \\
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV \\
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV \\
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV \\
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV \\
TTEKYKGGSSTLVVGKQLLENYPLGSKLNPLYRALSTKLNGGLRSITVV
\end{align*}
\]

Figure 5.1. Variation in match intensity or alignment length will have affects on the distribution of matches and, therefore, clumpiness.

5.2 Differentiation between Different Levels of Clumpiness and Reordering

Given that the intensities and lengths are equal, it is necessary to define the differences between the differing levels of clumpiness. The consideration is then what, exactly, would be considered a clumpy alignment and an unclumpy alignment? Consider Figure 5.2, which shows alignments with virtually the same intensity of matches along the same length of alignment but with varying distributions and, therefore, degrees of clumpiness.

In the very regular alignment, A, all the matches are dispersed singularly along the length of the alignment apart from two groupings of two, this would be considered an unclumpy alignment and would receive a low clumpiness value. On the other hand, Alignment B has many of its matches clustered together in groups, circled, with few singular matches making this a clumpier alignment, receiving a higher clumpiness value. In between these two would be the alignments C and D where the matches are more clustered than A but less then B, giving both C and D and a medium clumpiness value. However, the ordering becomes more difficult when trying to discern which one of these two would have a greater value, the closer the degree of
clumpiness between two alignments, the more difficult it is to determine which has the greater degree of clumping by eye.

```
A
TTEKYKGGSTLTVGKQLLENYPLGKLKNPYLRALSTKLNGGLRSITV
T + Y G + T V ++ L E Y P G + K P L S L G L S T
TMQPYFGIACTFVMEREPLMEWYPDGADKLPLWEMSVLGGILFSPTN
B
TTEKYKGGSTLTVGKQLLENYPLGKLKNPYLRALSTKLNGGLRSITV
  TEKY+ + L+ GK LL+NYP S P +LSTK R ITV
LTEKLRRYAFLLKTYGLQQNYTPSYREPMCKSLSTKFWNDVRPITV
C
TTEKYKGGSTLTVGKQLLENYPLGKLKNPYLRALSTKLNGGLRSITV
  T+K + G +TL V L Y (K+LK) P L S L G R I
LTQKGREGPATLMVCLDFLAWPYQRLAKKPPALMGDSPMLFGDDRLIMN
D
TTEKYKGGSTLTVGKQLLENYPLGKLKNPYLRALSTKLNGGLRSITV
  T + K G K L L YP+ KN (LR+L) KL L+S TV
LTPRVGYFVWMDKYLFLNPYIMNDAKNRCRLSRLKAKLPMWLKSPTV
```

Figure 5.2. Examples of varying degrees of clumpiness

Nevertheless, these visible differences can then be used to reorder the BLAST results, thereby bringing less similar but possibly more biological relevant hits to the notice of the scientist. In the example in Figure 5.2 the alignments could be reordered such that B, being the clumpiest, would be at the top and A, being the least clumpy, would be at the bottom. Again, the placement of the intermediate alignments would be difficult, with different people placing them in different orders dependent on their personal perspective. For example, one such approach could be that since both have one group of four and two groups of three but that only D has groups of two, then D would be placed higher than C, as in Figure 5.3.

There are two aspects to the quantification, that of what to measure and how to measure it, which can either be easy or complex to define and implement, respectively. For example the measurement of a person’s height may be easily defined as the distance, when standing upright, that the top of the head is from the ground. From that, the measurement could be the use of a tape measure, held straight between the ground and the top of a person’s head. On the other hand, counting the number of dolphins in a body of water is not as straight forward. The definition of the measurement is simply how many dolphins are there but the logistics are
complicated by the fact that not all the dolphins will be visible, and they would not be cooperative by staying still. In this situation, the number of matches is known but it is the actual measurement of the distribution of these matches that is complicated

The use of a computational quantification of clumpiness for the reordering of the alignments will eliminate the uncertainties inherent in a manual approach biased by personal perspective. Just as a tape allows for the precise measurement of a person’s height, rather than an indication of that person being so tall when judged by eye.

![Figure 5.3. Reordered alignments in accordance with clumpiness, the blocks have been used for clarity](image)

### 5.3 Circular Problem

In order to quantify the notion of clumpiness it is necessary to have a model on which to base this quantification. That is, to have a model that explicitly portrays the progression from clumpy to unclumpy, containing alignments with varying degrees of clumpiness from most too least clumpy. This model can then be used to evaluate the effectiveness of various measure methods in their ability to quantify the clumpiness of the alignments in the model. For example, if a model of clumpiness
places five alignments, A, B, C, D and E, in the order B, D, A, E, C, a valid measure would also need to place the five alignments in the same order.

Again, it is fundamental that the issue under investigation can be clearly defined and measurable. From the examples outlined earlier both were easily defined, that of height and the counting of dolphins. However, with the issue of clumpiness, the differentiation between varying levels of clumpiness is dependent on the personal perspective of the person ordering the alignments. Due to this ambiguity, there is currently no known model for the discrimination between different levels of clumpiness on which to evaluate a clumpiness measure. Many models of clustering of objects in two-dimensional fields discriminate between regular, random and clumpy when compared to a calculated value (Upton and Fingleton 1985). However, these do not put a value to the level of regularity or clumpiness, which is required here for the reordering of the alignments. Therefore, it would be necessary to create this model of clumpiness before it could be used to create a measure of clumpiness. Then again, to create the model it is necessary to have the measure by which each of the alignments to be placed in the model can be correctly positioned. In the example of the five alignments it is necessary to know that the alignment B would have the greatest clumpiness value in order to position that in the first place of the clumpiness model. This results in a circular problem in defining the clumpiness measure in that a model is needed to produce the measure and a measure is needed to produce the model, left side of Figure 5.4, requiring the definition of both the model and the measure in order to define either.

One solution to this problem is to generate an approximate model, right side of Figure 5.4, on which a measure can be based and then test that measure's ability on a test set. This test set is simply another approximate model, which would allow for the ambiguity in the different personal perspectives by incorporating those different
perspectives in the approximate model. This is the method that will be used in this research during the creation and verification of a clumpiness measure, as detailed in Chapters 6, 7 and 8. There, a manually ordered simulated dataset is used as the approximate model to detect the most appropriate measure out of a number of possible, described in Chapter 6, which may then be verified against further manually ordered sets and in situ, in real BLAST situations using a prototype, Appendix E.

Once the approximate model has been created manually, the proposed measure must be evaluated in order to validate the ability of that measure to order a set of alignments into the same order as that of the approximate model. To do this, it is first necessary to establish the goals, or criteria, of the measure by which each measure candidate can be evaluated.

5.4 Clumpiness Criteria
The first, and by far the most important of these criteria is the measure’s capability to evaluate the degree of clumpiness in a particular alignment and assign a value relating to this degree of clumpiness. Directly as a consequence of this quantification is the measure’s ability to reorder the alignments as to their degree of clumpiness in relation to the approximate model defined in the previous Section. This should then allow the straightforward visual assessment of the alignments by the scientist in order to determine which are of relevance to their research.

As stated, the purpose of quantifying clumpiness in the BLAST alignments is to highlight those alignments that possibly contain conserved regions. However, the degree to which functional elements of a protein are conserved can vary widely dependent on the protein in question. This can range from large numbers of conserved amino acids, as in the bacterial and bacteriophage DNA binding proteins, which have 22 conserved proteins (Jonassen et al. 1995), to very few amino acids, such as in the globins, which only have a few positions conserved (Falquet et al. 2002). The measure produced by this research is unlikely to be of any help in the detection of very weakly conserved regions, due to too many mismatches within a clump rendering it indistinguishable from surrounding unclumpy areas. On the other hand, it should be capable of assisting in instances where there are few or no
mismatches in the conserved region of a protein. To this end, the second criteria is
the ability of the measure to allow for single mismatches in a conserved region or
clump, for example, Figure 5.5 shows a possible conserved region, boxed, containing
one mismatch. An incorporation of a mismatch in a clump would, in theory, reduce
the clumpiness measure by its nature; the introduction of a ‘gap’ would split a clump
into two. However, the presence of a single mismatch, or gap, should have as little
affect on a clumpiness measure as possible, the appearance of another member of the
clump immediately after the gap preventing the clumpiness score from decreasing
excessively as in gapped similarity searches.

TTEKYKGGSTLVGKQLLENYPCLKSLLNPLALSTKLNGGLSITV
T+K + G + TL V L Y K+LK P L S L G R I
LTQKGREGPATLMLVCLDFLAWYPQRLKALKPPALMGDSFMLFDRLIMN

Figure 5.5. Incorporation of mismatches into a conserved region

Other criteria for the clumpiness measure are of a more technical nature. Any
additional computational processes should not require much in the way of computer
resources, either time-wise or memory-wise. If this were not the case, the
incorporation of the measure would be considered more troublesome and, therefore,
pointless to install. Therefore the other two criteria under consideration are that the
measure is fast and that it does not require all a computer’s resources.

In summary, an effective clumpiness measure would:

• Have the ability to quantify the degree of clumpiness in a BLAST similarity
  alignment
• Be able to use these quantifications to correctly order a set of alignments as
  compared to an approximated clumpiness model
• Allow the incorporation of single mismatches into a conserved region, or
  clump, with little or no affect on the quantification

In addition a program created to perform the measure calculations should

• Be able to be installed and ran on a stand alone personal computer and
  o Run the calculations quickly
  o Use as little computer resources as possible
These criteria in mind, the next Chapter describes various possible clumpiness measure ideas with discussions as their expected performance on the simulated dataset described in Chapter 7.
6 CLUMPINESS MEASURES

This Chapter describes a number of potential measures for the quantification of clumpiness within BLAST result alignments. As described in Section 5.1, the aspect under investigation is the line showing the matches between a query sequence and the sequence found in the database by BLAST, referred to here as the similarity line, Figure 6.1. This similarity line may consist of 22 different characters, the 20 amino acids, a '+' and a space. For the purpose of clumpiness quantification this has been simplified to two characters by converting the similarity line to a binary line of ones and zeros, a one indicating an identical or positive match, and a zero indicating a mismatch, as may be seen in the binary line in Figure 6.1. This binary line may then be analysed for its degree of clumpiness, which is quantified for the purpose of reordering the BLAST results, as described in Section 5.2.

![Match found by BLAST](image)

<table>
<thead>
<tr>
<th>TTEKYGGSSTLGVGKQLLENYPSCPNSLNSLRLSTKLNGGLRSITVV</th>
<th>T+ + Y+ GS+TLV+ + Y+ G+ S+++ AL++KL + + +V</th>
</tr>
</thead>
<tbody>
<tr>
<td>TVRLXRDGSNTLVSQGFHDSTYSHGSSVQSVIRALTSLPKHPAVNGLYLV</td>
<td>Binary line</td>
</tr>
</tbody>
</table>

**Figure 6.1.** Conversion of the similarity line to binary

For the rest of this Chapter the term 'line' refers to the generated binary line and the ones and zeros represent the matches/positives and mismatches respectively. The candidate measures for this quantification that this work concerns itself with are described below with examples for illustrative purposes. They have been categorised into elementary statistical, sliding window and miscellaneous measures.

Many of the measure ideas described below, use, as part of their calculations, a comparison to a randomly distributed line. Within the scope of this research a randomly distributed line represents a line that contains an equal number of ones as the query line, distributed randomly along the whole length of the 'random' line.
6.1 Elementary Statistical Measures

The ideas behind the first class of measures are based on elementary statistical methods and being such, are easy to implement, using little computational resources.

6.1.1 Average Lengths of the Groups

This measure is the determination of the average group size of the ones in the line by dividing the number of ones by the number of groups of ones, Figure 6.2. A group can be of any length from one to the length of the line, if it were a perfect match, with the idea being that the clumpier a line is, the higher its average would be. However, when using averages misleading results can arise through data skewing. A line that contains some large groups but many smaller groups will have a reduced average, which could, in turn, reduce that line’s clumpiness. On the other hand, the more smaller groups that are present in the line, the less clumpy it really is and so a lower score would be expected.

![Figure 6.2. Example of the process of calculating the average group size](image)

6.1.2 Largest Group Length and Proportion

Another measure is that of the largest group length, Figure 6.3.a, with the supposition that the larger the largest group the clumpier the line. However, the presence of the largest group does not necessarily lead to that line being the clumpiest, it may contain the largest group but the rest of the matches may be dispersed along the rest of the line, thereby reducing the overall clumpiness. The relative size of the mean to the largest group, the proportion, may correct this since that line would have a lower average, Figure 6.3.b, however, this may incorporate the issues from both measures.
Figure 6.3. Example of the process of calculating a. the largest group size and b. proportion the average is of this largest group

By its very nature, the largest group measure would not identify a clump containing a ‘lonely zero’, since it looks for the largest run of matches. In essence, it identifies ungapped clumps just as the original BLAST located ungapped regions of similarity. Therefore, in order to incorporate ‘lonely zeros’ into the measure, what constitutes a largest group requires redefinition. A number of approaches to this redefinition are illustrated in Figure 6.4 where on each of the example sequences, the larger groups have been underlined, with brackets differentiating between overlapping groups. The first four methods can produce two measure results, one with just the summation of the number of ones in the group and the other the summation of the number of both the ones and zeros present in the group. In each of the redefinitions the incorporation of zeros is not compulsory, i.e. if there is a group with no zeros larger than one with, the one without would be selected as the measure result. Also, any additional zeros may not be in the first or last position in a particular group. These redefinitions, as they appear in the figure are:

a. Allowing a single additional zero in a group. In the figure the largest group has two ones and seven ones separated by a zero, resulting in the largest group being either nine or ten.

b. Allowing two additional zeros, either singularly or together, in a group. The example in the figure shows two overlapping groups, bracketed, the first being the same as for a., the second with the seven ones and a single one separated by two zeros. Summed without the zeros gives nine and eight respectively but the summation including the zeros gives both groups the value of ten.

c. Allowing three additional zeros, singularly or together, in a group. In this case the two overlapping groups of b. would be combined into one larger group of ten, thirteen if the zeros were included. There are also two
overlapping groups at the end of the line each containing three additional zeros.

d. Allowing any number of zeros, up to two zeros at a time, provided a group of two ones follows the zeros. So ‘10011’ would be allowed to continue, whereas, ‘10010’ or ‘10001’ would not. So, again, the largest group in the example line has two ones, a zero, seven ones, two zeros and a single one. However, the larger groups later in the line found by method c. are no longer present.

e. Since BLAST itself allows for gaps using affine gap costs, Section 3.1.3, it is reasonable to assume that a similar technique could be used here. Therefore, the final method assigns a value to each of the positions in the line giving a positive value if they are a one, a negative value if they are the first zero in a gap or a greater negative value for subsequent zeros in a gap, the red, negative value, and green, positive value, line in the figure. Starting from each one in the line, these values can be summed to find the highest value reached giving the measure value, the bottom two black lines, here seven.

<table>
<thead>
<tr>
<th>Overlapping Groups</th>
<th>Measure Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. extra single zero</td>
<td>10001101111111001000001001011110000111110001001011</td>
</tr>
<tr>
<td>b. extra zeros up to 2 zeros</td>
<td>1000110111111100100001001011110000111110001001011</td>
</tr>
<tr>
<td>c. extra zeros up to 3 zeros</td>
<td>1000110111111100100001001011110000111110001001011</td>
</tr>
<tr>
<td>d. any number of single zeros provided each is followed by two ones</td>
<td>1000110111111100100001001011110000111110001001011</td>
</tr>
<tr>
<td>e. gap and continuation costs</td>
<td>1000110111111100100001001011110000111110001001011</td>
</tr>
<tr>
<td></td>
<td>12345675231... (value becomes increasingly negative)</td>
</tr>
<tr>
<td></td>
<td>highest value = 7</td>
</tr>
</tbody>
</table>

Figure 6.4. Examples of various approaches to the redefinition of the largest group measure
6.1.3 Frequencies of the Different Sizes of the Groups

Since this project focuses on the distribution of ones in a line in order to measure the distribution of clumps in a line, it is reasonable to investigate the distribution of these clumps by measuring the number of times each group size appears in the line. It would be expected that clumpier lines would have greater frequencies at the larger group sizes and lower frequencies at the lower group sizes. This may be seen in Figure 6.5 where, although both lines have their highest frequencies in the group size of one, the clumpier line, a, has lower frequencies at the lower group sizes and higher frequencies at the higher group sizes than line b.

This produces the same number of values as the maximum group size, e.g. a maximum group size of 7 gives 7 values. However, the purpose of this quantification is to reorder the BLAST results, which is not feasible with multiple values and, therefore, these need to be combined into a single measure value. To combine all the frequencies into one measure, it is not advisable to take the average as this would produce the same answer as the averages measure (Section 6.1.1). Therefore the number of times the frequency rises above what would be expected from a dataset of randomly distributed lines is calculated instead, Figure 6.6. The
frequencies of the different group sizes for a hundred randomly distributed lines are calculated and the average for each frequency over the hundred lines is used to compare against the frequencies of the query line. The measure value is then the number of times that the frequencies in the query line are greater than the frequency averages from the randomly distributed lines, circled in Figure 6.6. Those lines that are 'clumpier' should have a greater value than those that are not.

No. times test frequency > random frequency = 3 (circled)

Figure 6.6. Example of the process for calculating the frequency of groups compared to random

However, if the distribution of ones is regularly dispersed along the length of the line, this could lead to the number of lower group sizes being greater than expected. This would be an issue since it is not the lower frequencies that are of interest in this research, as they would not be considered clumps and, therefore, give a false positive. A solution to this is to only use the frequencies of the larger group sizes of four and upward as defined in the interview, Section 4.2.3.3, as a minimum size for possible motifs in the alignment and, thereby, avoids this issue. For the example in Figure 6.6 the resultant measure value would still be 3 but there is now no risk that the lower group sizes would be able to have any effect.
6.2 Sliding Window Measures

All the measures discussed in this Section use a sliding window. A window frames a number of characters in the line and is slid along the line to make a calculation at each position in the line for this number of characters. For example, in Figure 6.7, the number of characters, window size, is five and as the window slides along the line, a calculation is made from every position for these subsections of five characters. This would create the same number of values as there are positions in the line, so all the separate measure values are then combined in some manner to give the final overall single measure value. The choice of window size is important. On the one hand, if the window is too large and the measure will be overly specific and incorrectly downgrade lines of medium clumpiness. On the other hand, if the window is too small, larger clumps will be lost, although they would increment the measure value multiple times as the window slides across them. However, this sliding window method should alleviate the issue of the previous measures in allowing the inclusion of 'lonely zeros', a single extra zero should not alter the value for a particular a window too greatly. These measures are categorised by intensity, Markov chain theory and spatial statistics.

\[
\begin{align*}
&10001101111110010000100101110001111110010010111 \\
&\vdots
\end{align*}
\]

Gives the windows:
\[
\begin{align*}
&10001 \\
&00011 \\
&00110 \\
&01101 \\
&11011 \\
&\ldots
\end{align*}
\]

number of characters of line framed by window, on which measure calculations are done

Figure 6.7. Sliding window along line framing consecutive subsections of the line from each position in the line

Another commonality between these methods is their use of the simulated dataset of randomly distributed ones to combine the subsection values to create a measure value.

6.2.1 Level of Match Intensity along the Length of the Line

The intensity is the number of ones in a line divided by the total number of characters in, or length of that line. For any line the exact number of ones and the
The total length of the line is known and, therefore, the intensity. However, clumpiness is a description of the distribution of the matches, therefore, intensity alone would not differentiate between two sequences that contain exactly the same number of ones, but where the ones are in different positions. Figure 6.8 shows two sequences with exactly the same length, the same number of ones and, therefore, the same intensity but with different distributions, sequence a. has a more clumpy distribution, with the larger clumps circled, whereas in sequence b. the ones are distributed roughly evenly along the line.

<table>
<thead>
<tr>
<th>a.</th>
<th>1000110111111001000001001011100001111110001001011</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. ones</td>
<td>27</td>
</tr>
<tr>
<td>Intensity</td>
<td>0.529</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>b.</th>
<th>101011010101001010101010101010101010101010101010110</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. ones</td>
<td>27</td>
</tr>
<tr>
<td>Intensity</td>
<td>0.529</td>
</tr>
</tbody>
</table>

**Figure 6.8.** Comparison of two sequences with the same number of ones but different distributions.

Therefore, instead of purely using the intensity as a measure, the intensity is calculated using a sliding window of various sizes, Figure 6.9.a with each of the position values being combined by counting the number of times the window intensities are greater than the overall intensity, Figure 6.9.b, and the number of times the window intensities are greater than expected, Figure 6.9.c.
a. Shifting window (size 5)
   Intensity in window α = 1 ones / window size 5 = 0.2
   Intensity in window β = 5 ones / window size 5 = 1

b. Greater than overall intensity
   Overall intensity = 27 ones / line length 51 = 0.529
   ∴ Window α intensity of 0.2 would NOT increment the measure
   Window β intensity of 1 would increment the measure

c. Greater than expected
   Random window α has an average intensity of 0.524
   ∴ Window α intensity of 0.2 would NOT increment the measure
   Random window β has an average intensity of 0.492
   Window β intensity of 1 would increment the measure

**Figure 6.9.** Example of the intensities measure calculating a. the number of windows with a greater intensity than the overall and b. the number of windows with a greater intensity than expected

6.2.2 Markov Chains

Discrete Markov chains consist of a set of states \( q_i \in \{S_1, S_2, \ldots, S_N\} \), each of which being an observable event, and at any time interval, \( t \), the process will occupy one of these states (Ewens and Grant 2001). At each time interval the process will either stay in the same state or move to another state, Figure 6.10.a, though not necessarily using all of the possible state movements, Figure 6.10.b. Markov processes are not restricted to temporal situations using the term ‘space’ in place of time interval to indicate movement between discrete physical locations, such as in this project moving from one character in the line to another.
Figure 6.10. Movements through a Markov chain between states $a$, $b$, $c$ and $d$ with a. all possible movements, b. restricted movement between certain states

These movements, or transitions, are dictated by probabilities based on the current state

$$ a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N, $$

where the transition probability $a_{ij}$ is the probability that the event $q$ at $t+1$ is state $j$ given that at $t$ it was state $i$. For example, based on the four states shown in Figure 6.10.b, if the first event is state $a$ there is a set of probabilities for determining a transition to itself or the other three states, Table 6.1 row $S_1 - a$, $a$ to $d$ being 0 since that transition is not permitted in this example. From state $d$ there is only one possible transition, that of $d$ to $b$, this would therefore have a probability of 1 since movement of some sort must be made. Also, all the probabilities for moving from a particular state (the rows in Table 6.1.) always sum up to one.

Table 6.1. Example of probabilities for movement between states in a Markov chain with the probabilities shown as $a_{xy}$ where $x$ is the initial state and $y$ is the destination state

<table>
<thead>
<tr>
<th>Time $t + 1$</th>
<th>Total probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>$S_1 (\alpha)$</td>
</tr>
<tr>
<td>$S_1 (\alpha)$</td>
<td>$a_{11}$</td>
</tr>
<tr>
<td>$S_2 (\beta)$</td>
<td>$a_{21}$</td>
</tr>
<tr>
<td>$S_3 (\gamma)$</td>
<td>$a_{31}$</td>
</tr>
<tr>
<td>$S_4 (\delta)$</td>
<td>0</td>
</tr>
</tbody>
</table>

There are two assumptions associated with the Markov model:

- Each subsequent state is dependent only on the current state
• The transition probabilities are independent of the point in time the transition occurs

The first of these has already been alluded to in the description of the transition probabilities, where the probability is described as movement from one state to another, not one state following a sequence of states. The latter refers to the transition probability’s independence from when or where it occurs, irrespective of when a particular transition occurs; the probability will remain the same. For example, even if the transition from \( \gamma \) to \( \beta \) had already occurred five times the probability of moving from \( \gamma \) to \( \beta \) would still be the same as it was the first time.

6.2.2.1 Kth Order Markov Chains

In this research a two state Markov model is used, Figure 6.11, and the transition probability of obtaining a one following a one may be estimated. This is done by dividing the number of instances of a one following a one by the total number of instances of a one, Figure 6.12. If these probabilities are higher than expected, it could indicate that the ones tend to form clumps and, therefore, indicating this is a ‘clumpy’ line.

![Systematic diagram for the 1st order Markov chain with states 1 or 0](image)

**Figure 6.11.** Systematic diagram for the 1st order Markov chain with states 1 or 0

<table>
<thead>
<tr>
<th>10011101111100100000100101110000111110001001011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instances of one followed by one = 16 (tick marks)</td>
</tr>
<tr>
<td>Total number of ones = 27 (ones followed by one or zero)</td>
</tr>
<tr>
<td>Probability of one followed by one = 16/27</td>
</tr>
<tr>
<td>= 0.62 (2 d.p.)</td>
</tr>
</tbody>
</table>

**Figure 6.12.** Estimation of transition probabilities

For the purposes of this research, the Markov assumption that only the previous state is used in determining the movement probabilities will be relaxed. Instead a \( k \)th order Markov chain is used, ‘looking back’ multiple steps to see if the previous \( k \) characters, or words of length \( k \), affect the probability of obtaining a one next. In the
example of Figure 6.13.a the word, in red, is ‘1100’ and the desired probability here, as in the measure, will be of getting a one next, as in Figure 6.13.b with the probability of obtaining a one given the presence of the letters of the defined word. The idea is that a clumpy line will have a higher probability of staying, or returning to a one than would be expected in a sequence where the ones were randomly distributed along the entire length.

\[
P(q_{i+1} = 1|q_i = i_1, q_{i-1} = i_2, ..., q_{i-k} = i_k)
\]

\[
\begin{align*}
100011011111110000000100101111000111111001001011
\end{align*}
\]

\[
q_{t-k} (k = 3) \quad q_{t+1}
\]

**Figure 6.13.** Example of a \(k^{th}\) order Markov chain. a. \(k^{th}\) order Markov chain probability formula. b. Example of the probability formula using \(k = 3\)

As described above, the estimation of the probability for a particular word is done by dividing the number of instances of that word followed by a one by the total number of instances of that particular word in the line. For example, in Figure 6.13 there are three instances of the word ‘1100’ in the line, underlined, but a one follows only one of these, highlighted in red, therefore, the probability of the transition from ‘1100’ to ‘1’ is one divided by three, 1/3, or 0.3333.

Since the object is to measure the degree of clumpiness, the selection of words used in the calculations is restricted to the more ‘interesting’ cases, those of few or no zeros, as in a clump. In the smaller orders this is restricted to no zeros, whilst in the larger orders an increasing number of zeros may be tolerated, Table 6.2. Also for the larger orders an additional zero may be tolerated if it is in the first position to capture those instances when the word is at the start of a clump. For example, in the 5th order Markov chain only those probabilities calculated from the words that contain one zero or if the last four only have one zero are used:
One zero in window - 01111  One zero in last four - 01011
10111 01101
11011 01101
11101 01110
11110
11111

Table 6.2. Zero restrictions in words used in Markov chain calculations

<table>
<thead>
<tr>
<th>Word size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zeros allowed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Zero in 1st place</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Max zeros allowed</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

The word lengths chosen were from one to eight, eight being double the minimum clump size as defined in Section 4.2.3.3. These probabilities can then be used to generate two measures of clumpiness. The first calculates the average relative frequency for the selected words in each of the \( k \)th orders, Figure 6.14.a, though in the case of the first three, there is only one selected word so the measure is just this probability. The second is the number of times the probability each of these words in the eight Markov chain orders is greater than expected, Figure 6.14.b, though, the first three orders, again having only one word, is the difference from expected.

| 10001101111111001000001001011110001111110001001011 |
|----------------|----------------|----------------|----------------|----------------|
| 6th 5th 7th 8th 4th |

**a.** average probability for each word size

<table>
<thead>
<tr>
<th>Word size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. words</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>12</td>
<td>44</td>
<td>58</td>
</tr>
<tr>
<td>Average</td>
<td>0.593</td>
<td>0.688</td>
<td>0.727</td>
<td>0.658</td>
<td>0.577</td>
<td>0.486</td>
<td>0.182</td>
<td>0.095</td>
</tr>
</tbody>
</table>

**b.** the number of probabilities greater than expected with examples of calculated and expected probabilities taken from word examples above

<table>
<thead>
<tr>
<th>Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. words</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>32</td>
<td>44</td>
<td>58</td>
</tr>
<tr>
<td>Probability</td>
<td>0.593</td>
<td>0.688</td>
<td>0.727</td>
<td>0.667</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Expected probability</td>
<td>0.524</td>
<td>0.688</td>
<td>0.518</td>
<td>0.413</td>
<td>0.327</td>
<td>0.335</td>
<td>0.2</td>
<td>0.07</td>
</tr>
<tr>
<td>No. probabilities &gt; expected</td>
<td>0.069</td>
<td>0</td>
<td>0.254</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

**Figure 6.14.** Examples for calculating **a.** the average probability for each of the 8 order Markov chains and **b.** the number of probabilities greater than or different from expected

77
Since biology is not a simple system the expansion of the Markov chains, Hidden Markov Models, a more complex modelling system was also investigated.

6.2.3 Hidden Markov Model

The Hidden Markov model is an extension of the Markov chain, in which the states cannot be seen, i.e. are hidden, and may only be derived from observable events. To illustrate this, Rabiner (1989) describes the *Urn and Ball* model where there are a set number of urns each containing the same number of coloured balls, Figure 6.15. The urns represent the hidden states and the balls the observable events. An initial urn is chosen at random, from which a ball is chosen based on the probabilities of selecting balls from that particular urn. The colour is recorded as the observation and the ball returned to the urn. A new urn is then selected based on the probabilistic rules governing the movement between the urns and again a ball is selected and the process is repeated generating an observation sequence. As in the Markov model (Section 6.2.2.1) the state transition probabilities are defined as in equation 6.1. In addition, the observation probabilities also need to be considered and are defined by the probability of obtaining the observation $V_k$ at time $t$ given that the state is $S_j$:

$$b_j(k) = P(V_k \text{ at } t \mid q_t = S_j), \quad 1 \leq j \leq N$$

$$1 \leq k \leq M$$

6.2

There is also the determination of which will be the initial state and this is governed by the probability of the first state event being $S_i$:

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N$$

6.3

Figure 6.15. Urn and ball model showing four urns each with seven coloured balls.

Underneath are a sample of the associated probabilities of obtaining each of the colours from each of the urns (adapted from Rabiner (1989))
In this research the observable events are the ones and zeros in the line, each one or zero representing an individual observation, whereas the states are defined by being in a clumpy region or an unclumpy region, section, or being in a clumpy region, and unclumpy region or a 'lonely zero' region, section. As with the Markov chain measure (Section 6.2.2.1) clumpier lines would have a greater tendency to stay in the clumpy state and produce mostly ones, whereas unclumpy lines would remain in the unclumpy state and the bulk of these observations would be zeros. The lonely zero state was included in the second model to take into consideration that the biological system being modelled is complex and, therefore, a more complex model would be more likely to accurately define it.

As in the Markov chain, Section 6.2.2.1, the probabilities for the model can be estimated and used as an indication of the clumpiness of a line. However, unlike in the Markov chain, the estimation of the probabilities is a more complicated process for the Hidden Markov model. This is not only because of the estimation of probabilities of unknown entities, namely the states, but also because the probabilities of the observable elements may vary dependent on which state they are from. Rabiner (1989) describes a method using the Baum-Welch method for parameter estimation. This method has five parts, the Forward algorithm, the Backward algorithm, the Xi algorithm, the Gamma algorithm and, finally, the parameter estimation itself. Each calculation looks at each time instance, or position in the line, in turn either from the beginning or end and performs a calculation making use of the initial, transition and observation probabilities of equations 6.1 – 6.3. Each of these calculations needs to be saved for the use by some of the other calculations. Full details of these algorithms can be found in Rabiner (1989), therefore, the following is a summary for reference.

As the name suggested the forward algorithm 'steps' through the line from the beginning and involves three steps:

1. Initialisation

\[ \alpha_t(i) = \pi_i b_i(O_t), \quad 1 \leq i \leq N \]
2. Induction

\[ \alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \]  

3. Termination

\[ P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \]  

where \( N \) is the number of states and \( t \) is the position in the line. Therefore, in the induction phase each calculation uses the result of the previous calculation and determines all possible routes to the state \( S_j \) at time \( t+1 \) from the \( N \) states \( S_i \) at time \( t \).

Again, as the name suggests, the backward algorithm 'steps' through the line from the end and involves two steps:

1. Initialisation

\[ \beta_T(i) = 1, \quad 1 \leq i \leq N \]  

2. Induction

\[ \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N, \quad t = T-1, T-2, ..., 1 \]  

where, again, \( N \) is the number of states and \( t \) is the position in the line. As in the forward calculation, in the induction phase each calculation uses the result of the previous calculation but, this time, determines all possible routes from the state \( S_i \) at time \( t \) to the \( N \) states \( S_j \) at time \( t+1 \).

The \( \xi \) algorithm uses the values from both the Forward and Backward

\[ \xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \]  

And the Gamma algorithm uses \( \xi \)

\[ \gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \]  

All these calculations can then be used to estimate the parameters of \( \pi, a \) and \( b \):

\[ \bar{\pi}_i = \gamma_1(i) \]
\[
\begin{align*}
\alpha_y &= \frac{\sum_{i=1}^{T-1} \xi_i(i, j)}{\sum_{i=1}^{T-1} \gamma_i(i)} \\
\bar{b}_j(k) &= \frac{\sum_{i=1}^{T} \gamma_i(j)}{\sum_{j=1}^{T} \gamma_i(j)}
\end{align*}
\]

6.12 6.13

where \( k \) is a specified observation, in this research either 0 or 1. Once these estimated parameters have been calculated the probability of the observation sequence occurring given the parameters can be determined. This probability is found using the Forward algorithm and is, in fact, the result of termination phase, equation 6.6. It can, therefore, be determine which of the two parameter sets is the more likely, the original or the estimated. If the estimated set is more likely it can replace the original and the estimation process reiterated until the new parameters are no longer more probable than the old. This leads to the maximum likelihood estimate of the HMM (Rabiner 1989).

Rabiner (1989), does mention that this method “leads to local maxima only” and that the search space may have many local maxima. However, an analysis of the search spaces under investigation has shown that in these instances there are few or only one maxima, Figure 6.16. This analysis took the form of conducting the complete estimation process starting with various parameter probabilities, i.e. altering one probability by degrees of 0.01 whilst maintaining all other transition and observation probabilities and repeating this for each of the probabilities. Figure 6.16 shows the results of altering the observation probability of obtaining a 1 in the clumpy state whilst all other probabilities were constant.
6.2.3.1 Two State HMM

In the two state Hidden Markov Model used here, the observations are the ones and zeros in the line and the hidden states are defined as being in a region of clumpiness or a region of unclumpiness, Figure 6.17.

**Figure 6.16.** Three dimensional representation of the search space for the HMM parameter estimation showing two local maxima, where \( c_1 \) is an observation probability and \( ccuuu1 \) represents all the other probabilities.

**Figure 6.17.** Representation of the two state Hidden Markov Model for producing lines of ones and zeros
The actual clumpiness value was generated by determining the optimum combination of the probabilities when summed together and it was found by the process of elimination that this was a summation of four of the probabilities, from Figure 6.17

\[ 2HMMC = a_{cc} + a_{uu} + b_{c}(1) + b_{u}(0) \]

That is the transition probabilities clumpy to clumpy and unclumpy to unclumpy states and the observation probabilities of a $1$ from a clumpy state and $0$ from an unclumpy state.

6.2.3.2 Three State HMM

To introduce an additional degree of complexity to the model, a third state was introduced, that of the ‘lonely zero’ state, Figure 6.18, which would allow for the incorporation of single gaps, or ‘lonely zeros’ in the line.

![Figure 6.18. Representation of the three state Hidden Markov Model for producing lines of ones and zeros](image)

As with the two state Hidden Markov Model the transition and observation probabilities were estimated using the Baum-Welch method and the optimum combination of these probabilities was determined. In this case a summation of six of the probabilities:

\[ 3HMMC = a_{cc} + a_{cu} + a_{uu} + a_{u1} + a_{c1} + a_{u1} \]

being the transition probabilities from clumpy to clumpy, clumpy to unclumpy, unclumpy to unclumpy and lonely zero to lonely zero and the observation probabilities of a $1$ in the clumpy state and a $0$ in the lonely zero state.
6.2.4 Spatial Data Analysis

This project is focused on the analysis of the distribution of ones and zeros in a line, though it could also be considered as the analysis of the distribution of points in a defined spatial area, also known as the analysis of spatial data. In this case it is the analysis of ‘mapped’ spatial data since the precise location of each of the points, or ones, is known and this field of statistics inspires the next three methods. All three use counts at or within a specified distance of each point on the line for varying distances, the sliding window.

6.2.4.1 Combined Count and Distance Analysis

In clumpy lines there would be, by definition, a greater number of ones next to each other and, so, counting the number of ones within a certain distance of each position containing a one, will have a greater value in the subsections containing clusters, though reduced in those without. This is analogous to considering sheep in a field, if all the sheep are clustered near the shepherd the counts within a couple of meters of him will be very high but if the sheep are randomly distributed throughout the field the counts within a couple of meters of the shepherd will be much lower. This forms the basis of the combined count and distance measure, which counts the number of ones within a distance \( r \) of each position in the line. For example, Figure 6.19 shows the process of counting ones from the first position, circled, where \( r \) is three, five and seven. For clarity, the three distances have been reproduced below the line with the chosen position, first, in red and the ones in the subsequent \( r \) positions underlined. These counts may then be used to determine if there is deviation from random to clumpy. This idea is motivated by L- and K-functions (Upton and Fingleton 1985, pp. 87-90) for 2-dimensional sampling techniques that may be adapted to measure the 1-dimensional situation here.

\[
\begin{align*}
\text{\texttt{0001101111110010000010010111110001111110001001011}} \\
\text{1000 count = 0 matches within a distance of three} \\
\text{100011 count = 2 matches within a distance of five} \\
\text{10001101 count = 3 matches within a distance of seven}
\end{align*}
\]

**Figure 6.19.** From the first position, counting the number of ones a. 3, b. 5 and c. 7 characters along
However, as the end of the line is reached, there are no longer any subsequent positions in which to make the count. This is referred to as an edge effect (Upton and Fingleton 1985) and a couple of solutions to this problem include adjusting the calculation to include distances to the edge or wrapping the area on itself, effectively making the line loop. It is this latter solution that will be used in this research so that once the end of the line is reached counting continues from the beginning again. The actual BLAST alignments end due to the similarity falling below a cut-off value, Section 3.1.1, however, here it is necessary to quantify the clumpiness at all points in the line. To do this, the line is ‘continued’ so as to give an estimation of the clumpiness at the end of the line based on the distribution of the ones in the rest of the line.

Once all the counts for all positions and windows sizes were obtained, the data may be visually analysed in graphs, Figure 6.20. This graph is an image graph drawn using the statistical data analysis package, R (Center for Computational Intelligence 2004). In the graph, the lighter the colour the higher the count and the black lines, at the top and bottom of the graph, show the actual composition of the line, raised sections are ones, lower sections are zeros. As can be seen from the graph, the large area of ones towards the end of the line (circled) produces a wedge of lighter colours progressing from the larger to smaller distances.

**Figure 6.20.** Graph showing the counts of ones at the various positions in a line against the different distances, created using the R package
It is necessary to convert these multiple counts into a single value to enable the reordering of the BLAST results. This combination is done by the number of counts greater than expected, Figure 6.21.a, and by the number of counts where the value of the count is greater than half that particular distance, window size, plus one, Figure 6.21.b both for distances 3 to 15. The plus one has been added as the use of the half window size alone was found not to be sensitive enough for the reordering of lines.

**Figure 6.21.** Example of the process for the combined distance and count, showing the calculation of a. the number greater than expected and b. the number greater than half the distance plus one

6.2.4.2 Nearest Neighbours

A simplified form of the combined distance and count method merely ‘looks’ at either side of positions containing a one and increments the measure in the event that both neighbours are also ones. This measure more closely resembles the combined count and distance analysis described in Upton and Fingleton (1985) by searching the total area with a distance $r$ of the chosen point, i.e. to either side of the selected position. However, in this instance the distance $r$ is restricted to one, counting only those positions that are themselves a one with a one on either side. In essence, this measure counts the number of triplets of ones there are, thereby, counting any group size of three and up with the larger clumps having multiple instances of triplets in them, Figure 6.22.
In Figure 6.22 all the positions that would contribute to the measure have been coloured red showing that only those ones that are actually in a clump would contribute to the measure. This is different from the combined distance and count measure of Section 6.2.4.1, which would include all the odd ones as well.

### 6.2.4.3 Transect L-Function

Spatial point processes are often used to model the distribution of individuals in biological populations in space (Aldrin et al. 2001). Second order summary statistics are used to assess the regularity, clumpiness or randomness of a population. One such summary statistic is the K-function of a spatial point process (Ripley 1977) and with intensity \( \tau \), is defined as

\[
K(h) = \tau^{-1} \text{Expected} \left( \frac{\text{No. points within distance } h \text{ of a randomly chosen point}}{} \right)
\]

If the process under investigation has no spatial dependences \( K(h) = \pi h^2 \), whereas in regular and clustered processes \( K(h) < \pi h^2 \) and \( K(h) > \pi h^2 \) respectively (Cressie 1993). Here a one-dimensional version of the K-function (Aldrin et al. 2001) will be used to describe the clumpiness of a sequence. The K-function has been used to characterise spatial dependency in two or more dimensions and has been estimated for characterising spatial dependency in one-dimension (Aldrin et al. 2001) using an estimate of \( K(h) \) on a transect line:

\[
\hat{K}_1(h) = 2^{n-2} L \sum_{i<j} I(|y_i - y_j| < h),
\]

where \( n \) is the number of matches, \( L \) is the length of the line and \( y_i \) and \( y_j \) are particular event positions along the line. I designates a function where, if the formula inside the brackets is true I is 1 otherwise it is 0:

---

**Figure 6.22.** Example of the process of counting neighbours with a selected position underlined and its two neighbours circled
\[ I() = \begin{cases} 1 & \text{if } |y_i - y_j| < h \\ 0 & \text{otherwise} \end{cases} \quad 6.16 \]

So for two positions in a line, if they are both ones and the absolute value of the subtraction of the position numbers is greater than \( h \), then I would equal one for that position. These values of I are summed for all the positions in the line, values of \( i \) and \( j \), then this summation is multiplied by the length of the line and \( 2^{n-2} \), Figure 6.23.

\[
\begin{array}{cccccccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1
\end{array}
\]

a = two ones at distance 5 in positions 1 and 6
b = two ones at distance 7 in positions 5 and 12

If \( h = 6 \)
Then for a \(|y_i - y_j| = |1 - 6| = 5\), which is less than 6 and so increments \( L(h) \)
And for b \(|y_i - y_j| = |5 - 12| = 7\), which is greater than 6 and so does not increment \( L(h) \)

**Figure 6.23.** Example of how counts at distances can affect \( L(h) \) if \( h = 6 \)

As mentioned above, in the random case \( K(h) = \pi h^2 \) for spatial data. Thus, in order to identify non-random data, the estimated K-function \( \hat{K} \) is typically compared to \( \pi h^2 \).

It has been suggested to use the L-function

\[ L(h) = h - \sqrt{(K - \pi)}, \quad 6.17 \]

which involves comparing

\[ \hat{L}(h) = h - \sqrt{\hat{K}/\pi} \quad 6.18 \]

to \( h \) (Besag 1977). This facilitates the comparisons and also reduces the variance of the estimator. Figure 6.24 shows an example of the K function, Figure 6.24a, as compared to the L function, Figure 6.24b.
Figure 6.24. (a) Estimated K-function for a clustered spatial pattern (full line) and theoretical K-function (dashed line), (b) Estimated L-function for the same clustered spatial pattern (full line) and theoretical L-function (dashed line); data courtesy to Paul Armstrong.

This gives the same number of values as there are values of $h$, in this example the number of values is the same as the number of positions on the line. In order to combine these into a single clumpiness value each of the different $L(h)$ values are compared against expected. These expected values are calculated using the average value for each $h$ from a set of randomly distributed lines. Three clumpiness values may then be produced, specifically over all values of $h$:

- The number of times the query line $L(h)$ value is greater than expected $L(h)$ value
- The sum of the differences between the query $L(h)$ values and the expected $L(h)$ values
• The largest difference between the query $L(h)$ value and the expected $L(h)$ value

In this way a line with a greater degree of clumpiness would have a greater number of $L(h)$ values greater than expected. Similarly, the difference between the $L(h)$ value from the query line and the expected value should vary more in clumpier lines than less clumpy lines producing both a greater sum of these differences and have the greater largest difference.

6.3 Miscellaneous Measures

The next two methods do not fall under either the intuitive statistical or sliding window categories, Sections 6.1 and 6.2 respectively. Although the first of these methods does divide the line into subsections, these are discrete and not overlapping, whereas the second method takes a reading at every one of the positions in the line.

6.3.1 Adaptation of Chi-Squared Goodness-of-Fit Test

The Chi-squared test "is used to test for a significant difference between the distribution suggested by a data sample and a selected probability distribution" (McCuan 2003). In this research it is the difference between the distribution of a specific line and a randomly distributed line. If the selected line had few or no clumps there would be no significant difference whereas a clumpy line would have a significant difference. The Chi-squared test uses the following test statistic:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$  \hspace{1cm} (6.19)

where $O$ is the observed values for the data and $E$ is the expected values if the data came from a randomly distributed line. However, the value of the observed and expected would be counts of ones so the total number of ones in the entire line could not be used for the same reason that the entire line could not be used in the match intensity measure described in Section 6.2.1, if the number of ones in the test line and random line were the same $\chi^2$ would equal 0.

The line is split up into smaller sections and the counts of ones per subsection used as the observed elements, from which the Chi-square test-statistic could be calculated. However, the location of these ‘cuts’ can cause an issue in that if a cut is placed in the middle of a clump that clump is effectively lost, Figure 6.25.a,
separated into two clumps so reducing its impact on a measure. To remedy this, multiple runs are taken of the line with each pass starting at the next position, Figure 6.25.b, and the average over all of them constitutes the measure. There is the same number of runs as the window size since the next position would simply be the first position of the second window in the first run as underlined in the first and last runs in Figure 6.25.b. The fact that the last window will ‘drop off’ the end of the line can be dealt with by looping to the front of the line again as in Section 6.2.4. Also, as with the sliding window measures, Section 6.2, this should alleviate the issue of the lonely zero, a subsection with a single zero in it will still have a high value, though not quite as high as one with none.

Therefore, the observed counts are the counts of ones in each window and the expected counts are the expected number of ones for a specified window size for a line of a particular intensity. That is, given the intensity the expected value may be calculated by multiplication by the window size, for example, the sequence in Figure 6.25 has an intensity of 0.53, with a window size of 5 it would be expected that there were 2.647 ones in each of the windows. Given that 10 windows of size 5 can be placed end to end along this example line the total expected number of ones in the entire line is 26.47, very close to the actual 27 ones in this line. As mentioned, there

---

3 As in the Intensity measure, section 6.2.1, intensity is the number of matches in a line divided by the length of the line
is the same number of runs as the size of the window and taking the average Chi-squared value combines these to give the overall measure value, examples for window sizes 1 to 8 can be seen in Figure 6.25, where the clumpiness value has been calculated for a clumpier line, line a, and a less clumpy line, line b. As can be seen, on averaging the Chi-squared values for the five runs for both lines, the clumpier line produces a greater clumpiness value than the less clumpy line.

Line a
\[1000110111110010000100101110000111110001001011\]
Line b
\[100111010011001010111001011010100110010110010110\]

Window size = 5
Intensity = No. ones / length of line
= 27 / 51 = 0.53
Expected No. ones = intensity \times window size
= 0.53 \times 5 = 2.647

<table>
<thead>
<tr>
<th>run</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Clumpiness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi^2 values</td>
<td>line a</td>
<td>7.715</td>
<td>6.959</td>
<td>8.359</td>
<td>3.826</td>
<td>6.093</td>
</tr>
<tr>
<td></td>
<td>line b</td>
<td>0.804</td>
<td>1.559</td>
<td>0.804</td>
<td>1.559</td>
<td>1.671</td>
</tr>
</tbody>
</table>

Figure 6.26. Example of the calculation of the clumpiness value using an adaptation of the Chi-squared test on a clumpy and non-clumpy line

6.3.2 CUSUM
CUSUM, or cumulative sum, charts are generally used for quality control, that is, ensuring that a process is functioning within set parameters. For example, in a factory maintaining a mean weight of 10kg for bags of sugar, quality control is used to ensure that the weight does not vary considerably from that mean. Shewhart control charts, which use the average and range of a set of samples to produce the X-bar and R charts respectively, show that, if the observed values lie outside a upper or lower limits it would indicate that the system has deviated significantly from normal, it is out of control (Hawkins and Olwell 1998). However, these charts have no memory for previous observations and, therefore, may not detect persistent but small deviations from the mean (Hawkins and Olwell 1998). The CUSUM formula:

\[ C_i = \sum_{j=1}^{i} (x_j - \mu_0) \] 6.20
does have a memory with the addition of each successive value so that any deviations either above and/or below the mean will accumulate (Montgomery 2001). The use of a V-mask chart would visually display any deviations but here the “decision interval” formulae (Hawkins and Olwell 1998) will be used:

\[
\begin{align*}
C_0^+ &= 0 \\
C_i^+ &= \max(0, y_i - k + C_{i-1}^+),
\end{align*}
\]

where \( k \) is a set reference value, or allowance (see below) and \( y_i \) is used to standardise the CUSUM for direct comparison and is defined as

\[
y_i = \frac{x_i - \mu_0}{\sigma},
\]

where \( x_i \) is a particular value, \( \mu_0 \) is the specified mean and \( \sigma \) the standard deviation.

The system would signal that it is out of control, outside specifications, if the value of \( C_i^+ \) were to rise above the upper limit of \( h \). In quality control charts the choice of \( h \) and \( k \) are done so as to minimise the instance of false signals but maximise the possibility of catching a true out-of-control signal. Montgomery (2001) suggests that a \( k \) of half the standard deviation and an \( h \) of four or five times the standard deviation are “reasonable values”.

Only the equations for deviating above the mean have been shown since it is only those lines that are clumpier than expected that are of interest in this project. Also, since it is possible that the lines could start with a higher mean than expected the use of the Fast Initial Response (Lucas and Crosier 2000) head start enables faster signalling

\[
C_0^+ = \frac{h}{2}
\]

In Figure 6.27, forty values were randomly generated from a normal distribution, the first twenty with a mean of 6 and a standard deviation of 0.5 and the latter twenty with a mean of 7 and standard deviation of 0.5. Using the CUSUM formula to detect any deviations from the mean of 6 it can be seen that while the system is within specifications, a mean of 6, the CUSUM value remains close to zero and under the cut-off value of \( h \), indicated by the horizontal line just above the x-axis. However, after the mean change from the 21\(^{st}\) reading, there is a steady increase in the CUSUM value, with all CUSUM values from the 23\(^{rd}\) reading onwards being greater than the
value of $h$. Thereby indicating that this process is out-of-control, having deviated from its specifications.

![CUSUM Detection of a Mean Shift](image)

**Figure 6.27.** Example of CUSUM detection of a mean shift

In this research, an adaptation of the CUSUM formula was used, where the mean, $\mu_0$, was the overall intensity of that particular line and the standard deviation, $\sigma$, was calculated using:

$$\text{Standard deviation} = \sqrt{\frac{1}{L-1} \left( \frac{n - n^2}{L} \right)},$$

where $L$ is the length of the line and $n$ is the number of ones in the line. This is an abbreviation of the standard deviation for frequency tables:

$$\text{Standard deviation of a frequency table} = \sqrt{\frac{1}{\sum f - 1} \left( \sum x^2 f - \frac{\left( \sum xf \right)^2}{\sum f} \right)}$$

However, in this research, the values of $x$ are either zero or one. Multiplication by zero always equals zero and multiplication by one equals the value the one was multiplied by, therefore, a frequency table produced by any particular line would resemble:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f$</th>
<th>$xf$</th>
<th>$x^2f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\alpha$</td>
<td>$0 \cdot \alpha = 0$</td>
<td>$0^2 \cdot \alpha = 0$</td>
</tr>
<tr>
<td>1</td>
<td>$\beta$</td>
<td>$1 \cdot \beta = \beta$</td>
<td>$1^2 \cdot \beta = \beta$</td>
</tr>
<tr>
<td>$\sum$</td>
<td>$\alpha + \beta$</td>
<td>$\beta$</td>
<td>$\beta$</td>
</tr>
</tbody>
</table>
Since there are only ones or zeros in the lines $\Sigma f$ is the length of the line, hence $L$, and both $\Sigma xf$ and $\Sigma x^2f$ are $\beta$, which is the number of ones in the line, hence $n$, which gives the equation 6.24.

The value of $x_i$ is either the one or zero found in the corresponding position in the line and it was found that a $k$ of 0.6 times $\sigma$ and $h$ of four times $\sigma$ gave the best during the evaluation of the measure, as described in Chapter 7. For example with two sequences with the same length and number of ones, Figure 6.28.a, will, therefore, have the same means, standard deviations, $ks$ and $hs$, as calculated in Figure 6.28.b. These values can then be used to calculate the CUSUM value of each position in the line, as seen in Figure 6.28.c for the first 15 positions in the first sequence of Figure 6.28.a. The label of OOC indicates where the CUSUM value has exceeded the threshold of $h$ and is, therefore, considered out-of-control and Figure 6.28.d, which shows all the instances (marked O) when this occurs in both sequences. As can be seen, the clumpier sequence has 11 out-of-control instances compared with none in the unclumpy sequence and it is this count that constitutes the value used for the BLAST alignment rearrangement. As this measure uses a cumulative affect, the event of a lonely zero should not bring the CUSUM value back below the threshold if it appears in the middle of a clump. Indeed in Figure 6.28.d, it can be seen that in the upper line the large clump at the start of the line takes the CUSUM value above the threshold but in the event of a zero at the end it does not reduce back below the threshold and the event of a single one two positions further on brings the value back above the threshold, highlighted red.
CUSUM calculations are very similar to the L-function in that the value is incremented if it meets a certain criterion. However, in the case of the L-function the incrementation occurs right up to the end of the line only ever increasing the measure value. Conversely, in the CUSUM measure subsequent position values can also reduce the overall value again, taking it right back to zero if the summation step results in a negative value. If there has been a permanent shift in mean there is an upward (or downward) swing in the line on a graph, otherwise, the line will fluctuate around zero, as was seen in Figure 6.27.

6.4 Summary
This Chapter has given a description of the various potential measures proposed for the quantification of clumpiness within the BLAST alignments. For simplification purposes, each alignment is first converted into a binary line of ones representing
matches and positives and zeros representing mismatches. During the description of these measures various definitions have been used, specifically:

- A group refers to a clump or grouping of ones for which a clumpiness measure would need to be sensitive
- A randomly distributed line refers to a line that contains an equal number of ones as the query line, distributed randomly along the whole length of the ‘random’ line
- A window refers to a small subsection of the line, which is moved along the line making calculations at each position for the length of the window
- The intensity refers to the number of ones in a line divided by the length of that line

A summary of these potential measures can be found in Table 6.3.

During the creation of these measures, any testing undertaken was used to determine the accuracy of the measures in their calculations. The next phase of the research, and the next Chapter in this thesis, was the evaluation of the validity of each of the measures to quantify clumpiness. The former being the technical details ensuring that the coding of the calculation formulas is correct, the latter being the goal of the research. However, in order to evaluate the measures, it is necessary to have a model against which the measure may be compared. Although this model does not currently exist, Section 5.3, an estimation may be generated and this is also described in the next Chapter.
<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Averages the number of ones over the number of groups of ones in the line</td>
</tr>
<tr>
<td>Largest group</td>
<td>Finds the value of the largest group</td>
</tr>
<tr>
<td>Redefined largest group</td>
<td>Finds the value of the largest group but defined so can include the ‘lonely zero’</td>
</tr>
<tr>
<td>Proportion</td>
<td>Calculates the relative size of the mean to the largest group</td>
</tr>
<tr>
<td>Frequency</td>
<td>All groups: Obtains the frequencies of the different group sizes for either all group sizes or greater than 4 only and counts the number of times these frequencies are greater than expected compared to randomly distributed lines. Larger only: Takes the number of ones over the length of small windows of the line and counts the number of times this is greater than the overall intensity or number greater than expected compared to randomly distributed lines.</td>
</tr>
<tr>
<td>Intensity</td>
<td>Overall: Estimates the probability of obtaining a one following specified 'words' of length 1-8 and, for each word size, either calculates the average probability or the number greater than expected compared to randomly distributed lines. Expected: Takes the number of ones over the length of small windows of the line and either counts how often this is greater than the window size plus 1 or the number greater than expected compared to randomly distributed lines.</td>
</tr>
<tr>
<td>Markov</td>
<td>Mean: Obtains the frequencies of the different group sizes for either all group sizes or greater than 4 only and counts the number of times these frequencies are greater than expected compared to randomly distributed lines. Expected: Takes the number of ones over the length of small windows of the line and either counts how often this is greater than the window size plus 1 or the number greater than expected compared to randomly distributed lines.</td>
</tr>
<tr>
<td>HMM</td>
<td>Two State: Estimated the transition and observation probabilities for either a two or three state HMM and uses a combination of these for the measure.</td>
</tr>
<tr>
<td>Distance and count</td>
<td>½ window + 1: Obtains the frequency of ones within a specified window of the line and either counts how often this is greater than the window size plus 1 or the number greater than expected compared to randomly distributed lines. Expected: Counts the number of times a one in the line is adjacent to two other ones and either takes this value or the difference from expected.</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum: Obtains the summation of the number of ones at specified distances and returns the total summation over all distances, the largest distance or the number greater than expected compared to randomly distributed lines. Expected: Splits the line up into specified window sizes and calculates the chi-squared value for each window, conducting multiple runs to cover all possible clumps and taking the average of these runs.</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum: Splits the line up into specified window sizes and calculates the chi-squared value for each window, conducting multiple runs to cover all possible clumps and taking the average of these runs. Largest: Uses the intensity of the line as the mean uses the CUSUM quality control calculation to count the number of times areas in the line deviates above this mean. Expected: Uses the intensity of the line as the mean uses the CUSUM quality control calculation to count the number of times areas in the line deviates above this mean.</td>
</tr>
<tr>
<td>Chi-squared</td>
<td>Uses the intensity of the line as the mean uses the CUSUM quality control calculation to count the number of times areas in the line deviates above this mean.</td>
</tr>
<tr>
<td>CUSUM</td>
<td>Uses the intensity of the line as the mean uses the CUSUM quality control calculation to count the number of times areas in the line deviates above this mean.</td>
</tr>
</tbody>
</table>
7 MEASURE EVALUATION

In the previous Chapter detailed descriptions were given of the potential methods for the quantification of clumpiness. However, as was described in Section 5.3, there is a circular problem to this and the solution proposed in Section 5.3 and used here is to create an estimation of this model on which to validate the clumpiness measures. As in Chapter 6, a binary line will be used in the measure evaluation and the same terminology applies, with line referring to the binary line and one and zero referring to the occurrence of a match/positive and mismatch respectively.

7.1 Methodology

The motivation for evaluating each of the measures is to ascertain its ability to fulfil the performance criteria of a clumpiness measure. These criteria were defined in Section 5.4, but in summary a suitable clumpiness measure should have these properties:

a. The assignment of a valid clumpiness value to the alignment in question, thereby mimicking a realistic reordering of the alignments from most to least clumpy
b. The incorporation of single mismatches, ‘lonely zeros’, without significant degradation of the clumpiness value
c. The reasonable running time of the measure’s calculations in terms of time and memory consumption

In the case of the first criteria the fitness of purpose of the measure may be evaluated by assessing its ability to order a set of lines from clumpy to unclumpy as compared to a test set. This test set represents the approximation of the progression model mentioned above and was created by manually ordering the set of lines, Section 7.3. It is important to ensure that it is only the clumpiness of the alignment that is either being measured, or used in the manual ordering process. Therefore, all other aspects of each line are maintained, that of the length of the line and the intensity of the ones with the only variation being the distribution of these ones. To accomplish this, a simulated dataset is created, see Section 7.2 below, using lines of the same length in which the same number of ones are randomly cited along the line.
Given the order by the measure and the manual order the two may be compared using Spearman’s rank correlation coefficient

\[
rs = \frac{n \sum_{i=1}^{n} x_i y_i - \left( \sum_{i=1}^{n} x_i \right) \left( \sum_{i=1}^{n} y_i \right)}{\sqrt{n \sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - \left( \sum_{i=1}^{n} y_i \right)^2}},
\]

where \(x_i\) and \(y_i\) are the rank values for observation \(i\), and \(n\) is the number of observations. The simplified form

\[
rs = 1 - \left( \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \right),
\]

where \(d_i\) is the difference between the two ranks for observation \(i\), is equivalent, for non-tied measurements. These occur when two or more measurements in a rank have the same value. Though this is not an issue with the manually ordered lines, each one being placed in a distinct rank position, a clumpiness measure may give the same clumpiness value to more than one line. In the case of tied measurements, the simplified Spearman’s correlation gives a good approximation provided the number of ties is small in comparison to \(n\), where it is necessary to use the full formula, Section 7.4.1 (Zwillinger and Kokoska 2000). This gives a value between \(-1\) and \(1\), where a 1 would indicate that the orders are exactly the same, a \(-1\) would indicate that one order is the reverse of the other and a 0 indicates a complete lack of correlation. Therefore, in this research a value as close to 1 as possible is desirable.

The second criterion requires that the measure not be substantially affected by the introduction of single zeros, the ‘lonely zero’. The assessment of this criterion is more difficult as the introduction of zeros will, by its nature, render the line less clumpy and, therefore, reduce its clumpiness value. That said, here the testing for the tolerance of this lonely zero is accomplished by generating various sets of data that have differing numbers of lonely zeros, Section 7.4.2.

The final criterion is of a technical nature, focusing on the measure’s consumption of computer resources. This can simply be the time it takes for the program to run given a specific dataset and can be tested via running the simulated dataset and timing the
performance on commonly found non-specialist computers, Section 7.4.4. However, a program that performs well on a small sample dataset may not perform proportionally as well when that dataset is ‘scaled up’. For example, when the dataset is doubled in size, the program could take twice as long or four times as long to run. This is the efficiency or complexity of the algorithm, (Hasti 2001), and is expressed in what is popularly called the Big-O notation, where O stands for order, Section 7.4.3.

7.2 Simulated dataset

As mentioned above, in order that only the measure’s ability to quantify the clumpiness of a line is assessed, the length and intensity of ones must be maintained. To this end a simulated dataset is used, however, the actual creation of the set is not straightforward. The question is how to randomly create lines of the same intensity but differing degrees of clumpiness. That is randomly siting the ones in a line so as the line is a set length with a set total number of ones but the ones are not randomly distributed.

Since a measure has been based on Markov chains, it is also feasible that Markov chains could be used to generate the simulated dataset. Markov chains are described in detail in Section 6.2.2, however, to briefly review, Markov chains consist of a set of states \( q_t \in \{S_1, S_2, \ldots, S_N\} \) with associated transition probabilities \( a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N \) dictating the movement between these states. At each time interval, or spatial point, a state is chosen based on the transition probabilities of the previous state, creating an observation sequence. The two main assumptions of the Markov models are:

a. Each state choice is dependent only on the previous state
b. The transition probabilities are independent of the time the transition occurs

As in the Markov chain measure described in Section 6.2.2.1, the data simulation Markov chain uses two states, that of the event of a one or the event of a zero.

7.2.1 Markov

In the generation of a simulated dataset, a Markov chain involving just two states is used, that of the event of a one or the event of a zero, Figure 7.1.a, which uses a the
transition matrix of four probabilities to determine which state the model will move to at each successive step, Figure 7.1.b. For each line in the dataset, the probabilities of moving between the states are varied to create the progression from unclumpy to clumpy alignments, Figure 7.2. A clumpy alignment would have higher probability of remaining or returning to a one and a non-clumpy alignment, which would tend to

\[
\begin{array}{c|cc|c}
\text{State} & S_1 (1) & S_2 (0) & \text{Total} \\
\hline
S_1 (1) & a_{11} & a_{12} & a_{11} + a_{12} = 1 \\
S_2 (0) & a_{21} & a_{22} & a_{21} + a_{22} = 1 \\
\end{array}
\]

remain or return to a zero.

**Figure 7.1.** Markov chain used to generate a simulated dataset showing a. schematic diagram of the Markov model and b. the probability matrix for state transitions

<table>
<thead>
<tr>
<th>State 0 to state 1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated sequences</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.2.** Progression through probabilities to generate sequences. As the two probabilities for each state total 1, the probability of the transition to state 0 in each case is the 1- probability to state 1

However, it was found that this did not mimic the progression from clumpy to unclumpy as expected. Rather than altering the distribution of ones, it altered the number of ones and therefore mimicked the similarity progression seen in the BLAST results, Figure 7.3.
As mentioned above, this is the expected difficulty of generating a simulated dataset with no obvious way to control the intensity of ones within the simulated data whilst altering the distribution of these ones. However, being a more complex descriptive model, it was thought that the more in-depth description of a process by the Hidden Markov model could describe the more complex concept of clumpiness.

7.2.2 HMM

Hidden Markov models are described in detail in Section 6.2.3; however, in the generation of a simulated dataset it is not necessary to estimate the probabilities as was done there since in this case the probabilities are set and used to create the observation sequence. Figure 7.4 shows the hidden Markov model used to create the simulated dataset, which uses the two states of clumpy and unclumpy with the transition probabilities

\[ a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \leq i, j \leq N \]  

as well as the two observations of a one or a zero with their associated observation probabilities

\[ b_j(k) = P(V_k \text{ at } t \mid q_t = S_j), \quad 1 \leq j \leq N \]

\[ 1 \leq k \leq M \]
Figure 7.4. Hidden Markov model used to generate a simulated dataset

As in the Markov chain model, Section 7.2.1, for each line in the dataset, the transition probabilities and now the observation probabilities were varied to create the progression from clumpy to unclumpy. The generation of the simulated dataset uses varying values for the state transition probabilities for clumpy to clumpy from 0.1 to 0.9 in steps of 0.2 and varying values for the observation probabilities used are for producing a one from each state, again from 0.1 to 0.9 in steps of 0.2.

Although, like the Markov simulated dataset (Section 7.2.1), the sequences produced do not have a consistent intensity throughout, the volume of sequences produced allows for the selection of a smaller subset of similar intensities to form the test dataset. The model described above gives five probabilities for the four transitions and observations giving a total of $5^4$ or 625 sequences with intensities ranging from around 0.4 to 0.96. The intensities do not need to be precisely the same, therefore, a selection of those sequences within a small range, around an intensity of 0.5, are sufficient to give the required sequences of the simulated dataset.

7.3 Subjects

Since it was not possible to automatically produce the progressive model from clumpy to unclumpy, once the lines for the simulated dataset had been created, it was necessary to manually order them, for the use of comparing with each of the proposed clumpiness measures. Also, since the actual data to be reordered are genetic sequences, the desired order needs to reflect the variations in clumpiness from the viewpoint of the scientists, i.e. the order into which geneticists would place them. To this end, a group of eight 'experts', from various biological fields that use BLAST ordered a small dataset made up of ten simulated lines. These lines were generated using the hidden Markov model described above and ten lines being
selected so as to minimise variance in the intensity. However, care was taken to include lines that appeared very clumpy, very dispersed and having various degrees of clumpiness in between to give a better estimated model than if all the lines contained the same level of clumpiness. The chosen lines were then converted into strips of white and black blocks, Figure 7.5, where a black block is a one and a white is a zero.

![Figure 7.5. Strips used in ordered dataset creation](image)

This gives eight ordered datasets from which a consensus order may also be produced, Table 7.1, with the consensus taking the most common strip letter for each position across the eight orders.

**Table 7.1. Summary of ordering by subjects and generated consensus order**

<table>
<thead>
<tr>
<th>Consensus</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>G</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>G</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>
During this process it turned out that the field of research the experts came from and how they used BLAST had an effect on the order into which they placed the strips. For example, subject four is a molecular ecologist, using BLAST as an aid to designing primers rather than for function determination. In this case they would prefer to see the matches/ones spread evenly across the whole sequence, in essence the opposite of the clumpiness measure, therefore, the actual order was reversed. However, on the whole, the experts were looking for longer groups of ones, black blocks, though allowing for the odd gap, white blocks, which reiterates the criteria where single gaps should be permitted.

7.4 Results and Discussion

Given this manually ordered dataset, comparisons with the ordering by the clumpiness measures may then be conducted using the Spearman’s rank correlation coefficient, as detailed in Section 7.1. Also, since the experts included ‘lonely zeros’ in their interpretation of the clumps in each line, this should be reflected in the correlation with the clumpiness measures, however, additional testing should confirm a measure’s ability to incorporate the lonely zero.

7.4.1 Correlation

Using Spearman’s rank correlation coefficient the ordering of each proposed measure is compared against each of the eight subjects ordering and the consensus order. For those measures based on window size, only the data using the optimal window size is shown, based on the window size giving the best correlation. Table 7.2 summarises these results, with the first column showing each of the measures, subdivided into the various combining methods used, where necessary. The other columns show the correlation of each of the measures compared firstly to the consensus order and then to the order each of the eight subjects defined. For clarity the correlation values have been colour coded as to their degree of correlation:

- Green – a correlation greater than 0.9 defines an extremely good correlation
- Yellow – a correlation between 0.7 and 0.9 defines a good correlation
- White – a correlation between 0.5 and 0.7 defines a fair correlation
- Orange – a correlation less than 0.5 defines a poor to very poor correlation
Initially, the simplified form of the Spearman’s correlation was used, equation 7.2, and the tables for these results can be found in Appendix C. However, it was felt that several of the ordered lists had too many tied measurements compared to the number of observations, therefore, as stated above, equation, 7.1, has been used in the construction of Table 7.2. Though it should be noted that the difference between the two results is minimal, nevertheless, accuracy is important for the determination of the most feasible clumpiness measure.

Overall it would be expected that a good to extremely good correlation would be achieved though falling short of a perfect correlation as the comparison is against manually ordered lines where individual perspectives can vary widely. Nevertheless, a measure ‘in the green’, as it were, would be a very good prospect for the chosen clumpiness measure.

Table 7.2. Spearman’s rank coefficient for all subjects and consensus orders over all possible measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>C</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 4</td>
</tr>
<tr>
<td>Average</td>
<td>0.85</td>
<td>0.90 0.94 0.82</td>
</tr>
<tr>
<td>Largest group</td>
<td>0.96</td>
<td>0.96 0.94 0.99</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>0.84</td>
<td>0.84 0.80 0.80</td>
</tr>
<tr>
<td>Gapped largest group (1.5x)</td>
<td>0.99</td>
<td>0.99 0.97 0.99</td>
</tr>
<tr>
<td>Proportion</td>
<td>0.76</td>
<td>0.73 0.67 0.83</td>
</tr>
<tr>
<td>Frequency</td>
<td>All groups</td>
<td>0.77 0.77 0.77</td>
</tr>
<tr>
<td></td>
<td>Larger only</td>
<td>0.87 0.89 0.89</td>
</tr>
<tr>
<td>Intensity</td>
<td>Overall (16)</td>
<td>0.87 0.85 0.87</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>0</td>
</tr>
<tr>
<td>Markov</td>
<td>Average (2)</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Average (5)</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Expected (8)</td>
<td>0.87</td>
</tr>
<tr>
<td>2 HMM</td>
<td>Sum (4p)</td>
<td>0.93</td>
</tr>
<tr>
<td>3 HMM</td>
<td>Sum (6p)</td>
<td>1.00</td>
</tr>
<tr>
<td>Distance and count</td>
<td>½ win + 1 (4)</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Expected (4)</td>
<td>0.73</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>0.30</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>Largest</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>0.92</td>
</tr>
<tr>
<td>Chi-square (6)</td>
<td>0.87</td>
<td>0.83 0.78 0.85</td>
</tr>
<tr>
<td>CUSUM</td>
<td>0.97</td>
<td>0.97 0.98 0.96</td>
</tr>
</tbody>
</table>
As can be seen, the correlation values for subject four differ considerably from the other seven, though this is most likely due to the fact that, as mentioned before, subject four uses BLAST in a different way and is more interested in evenness than clumpiness. Another notable detail is that there are only eight measures where the majority, 5 or more, of their correlation values are green, greater than 0.9. These are the Largest group, the Largest group with gap costs, the Markov average (both window sizes), both HMM, the Distance and count greater then ½ window size plus one and the CUSUM measures. Also of interest is the proportions measure, which gave the clumpiest line the lowest value and least clumpiest the highest. That taken into consideration, ranking by least rather than most, the correlation between the orders is still fairly good.

Of the other results, most faired well, obtaining correlations greater than 0.7. Exceptions to this are measure methods Intensities, Distance and counts and Nearest neighbours, all counting the number of times greater than expected, in fact the Intensities greater than expected produced no values at all. However, the three measures with the highest overall correlations and, therefore, the most suitable for the clumpiness measure are the Largest group with gap costs, the three state HMM and CUSUM measures. With the first of the criteria satisfied, the next is the tolerance of the 'lonely zero'.

7.4.2 The 'Lonely Zero'

This is a difficult concept to evaluate, as the addition of zeros, by its nature, lowers the clumpiness of a line. However, it is hoped that the change would not alter the clumpiness value by any substantial degree, with the need then to define what is meant by 'substantial degree'. Another issue to be taken into consideration is the variation with clumpiness, where the effect of a lonely zero could be lost simply because overall a particular line is more clumpy than another. However, in an attempt to test each measure's tolerance to 'lonely zeros' extra zeros are inserted into two test sets.
7.4.2.1 Additional Zeros in Line J

The first test set takes line J from the simulated dataset, Figure 7.5, since this was initially chosen as a very clumpy line, and replaces none, one, or two ones with a zero, Figure 7.6.

<table>
<thead>
<tr>
<th>0000000010111111_111_01111 0</th>
<th>100000001111111111110 ~ 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000010111111111011110 ~ 01</td>
<td></td>
</tr>
<tr>
<td>00000001011111111111111_110 0</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 7.6. Adding extra zero into two locations in line. Lines are truncated with tildes for illustrative purposes*

The premise is that there will be virtually no difference between the four values, though, obviously, there may be some due to the introduction of the additional zeros.

It should be noted that for the purposes of these tests some of the original measures are not included. This is because they either faired exceptionally poorly, such as the intensity greater than expected measure, or they produced multiple clumpiness values, such as in those measures that used different window sizes. In this latter case only the optimal clumpiness value, or window size, was used.

As with the correlation results, Table 7.2, the results of the lonely zero test have been colour coded. Green indicates virtually no change (coded 0), yellow a steady increase or decrease in value across the four lines (coded 1) and white an erratic variation across the four lines (coded 2), though that may only be due to the order of the two lines with a single extra zero. Integer values are deemed unchanged if they are within 1 of each other and decimal values are unchanged if they are within 0.01 of each other.
Table 7.3. Measure values with the incorporation of none, one or two extra zeros

<table>
<thead>
<tr>
<th>Measure</th>
<th>Extra zeros</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Average</td>
<td>3.562</td>
<td>3.294</td>
</tr>
<tr>
<td>Largest group</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td>Gapped largest group (1.5x)</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>Proportion</td>
<td>0.297</td>
<td>0.366</td>
</tr>
<tr>
<td>Frequency</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Intensity (16)</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>Markov</td>
<td>Average (2)</td>
<td>0.780</td>
</tr>
<tr>
<td></td>
<td>Expected (8)</td>
<td>23</td>
</tr>
<tr>
<td>2 HMM</td>
<td>Sum (4p)</td>
<td>3.578</td>
</tr>
<tr>
<td>3 HMM</td>
<td>Sum (6p)</td>
<td>2.224</td>
</tr>
<tr>
<td>Distance and count</td>
<td>½ win + 1 (4)</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Expected (4)</td>
<td>51</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>5</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum</td>
<td>14.848</td>
</tr>
<tr>
<td></td>
<td>Largest</td>
<td>0.437</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>65</td>
</tr>
<tr>
<td>Chi-square</td>
<td>17.158</td>
<td>16.201</td>
</tr>
<tr>
<td>CUSUM</td>
<td>61</td>
<td>62</td>
</tr>
</tbody>
</table>

As mentioned, the Largest group, Largest group with extra zeros and Proportion measures only seem to be erratic due to the order the data has been entered into the table, swapping the two lines with one extra zero would alter both the Largest group measures to a decrease and proportion to an increase. That said, the first three, do not appear to allow for lonely zeros, however, neither do five of the other methods that were expected to fair better.

From Table 7.3 it would appear that the CUSUM measure cannot tolerate the lonely zero, however, Figure 7.7 shows the actual calculation for ten positions of line D of the ordered dataset, Figure 7.5. As can be seen, the presence of a single zero in the clump does not prevent the calculation being considered out-of-control\(^4\) and, therefore, still increments the measure. This is because of the nature of the measure; the accumulation of the values above the specified value \(h\) may not be brought low enough to be below \(h\) when a single zero is introduced, as in Figure 7.8, which shows the CUSUM value at each position in the line. As can be seen at position eight,

\(^4\) Out-of-control is the phrase used in quality control to indicate that the system is not operating as designed. Here, if refers to a line that would be clumpier than expected.
where the single zero is the CUSUM value dips back down but does not cross the threshold value of $h$, horizontal line.

**Line D**

<table>
<thead>
<tr>
<th>Xi</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0.560</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>std</td>
<td>0.499</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td>-1.422</td>
<td>0.583</td>
<td>1.165</td>
<td>1.748</td>
<td>2.331</td>
<td>2.913</td>
<td>3.496</td>
<td>2.074</td>
<td>2.657</td>
<td>3.239</td>
</tr>
<tr>
<td>max</td>
<td>0.299</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**(C+)** 0.000 0.583 1.165 1.748 2.331 2.913 3.496 2.074 2.657 3.239

Table 7.7: CUSUM catching lonely zero

**Figure 7.8.** Graph showing the variation in the CUSUM value including a single zero

Although this seemed to deal with the testing of the lonely zero, the introduction of an extra zero altered the intensity of the lines within the dataset, which would introduce an additional influence on the testing procedure. To ensure that it is the lonely zero that is being tested and not the change in intensity, an additional test set was created.

### 7.4.2.2 Additional Test Set

An additional test set was generated to provide a further test of the incorporation of the lonely zero and to assess whether the change in intensity affects the results. This was done using the Hidden Markov model described in Section 7.2.2 and as well as maintaining a minimal variation in intensity, the lines were also filtered by their clumpiness as defined using the CUSUM measure. The CUSUM measure was
chosen as it already shows signs of allowing for the lonely zero, as described in the previous Section. Once the lines had been created, the line, number six in Figure 7.9, with the largest clump had an additional zero replacement in that clump creating one test set and another set was created using the same additional zero but with an extra 1 replacing a zero at the start of that clump, i.e. by moving a zero, Figure 7.9. This was done in order to maintain the intensity of that line compared to the unaltered line.

The measures were then run twice, once with the unaltered test set and the test set with just the additional zero, and once with the unaltered test set and the test set with the relocated zero. Also, to remove any additional influences on the results, the lines of randomly distributed ones used in some of the measures were only created once in each run. Each line of the test set had its own set of randomly distributed lines, however, except for the altered line, it was the same set of randomly distributed lines that was used in both the unaltered test set and the altered test set. For the altered line a set randomly distributed lines needed to be generated for both the unaltered test set and the altered test set since the intensity could change and the randomly distributed lines were generated based on the current query lines intensity. This ensured that the values on all the unaltered lines was the same between the two sets and the only differing values would appear between the unaltered and altered sixth line.
Figure 7.9. Hidden Markov generated set, a., with a replaced zero to create one test set, b., and a relocated zero to create another set, c., circled. For clarity, only the line with the replacements is shown and white blocks denote a zero and green a one.

For comparison purposes the ordering of the two lines in each run were compared with the correlation as well as a visual assessment of the difference in the value and order position of the altered line, Table 7.4 and Table 7.5. In both tables, Rs denotes the correlation value, Value1 and Value2 denote the actual clumpiness values generated for the unaltered line and the altered line respectively and Pos1 and Pos2 denote the rank positions of the unaltered line and altered line respectively. If there is no change in the values and positions and, therefore, the Spearman’s coefficient, it is assumed that that particular measure is able to tolerate the ‘lonely zero’. If this is the case, the measure is assigned the code ‘0’, however, if the value changes but the order position does not the measure is assigned the code ‘1’ and if both change the measure is assigned the code ‘2’. However, it should be noted that, although, a position may not change, this may only be because its nearest positional neighbour is
too far away in value terms for a change to occur, conversely, there may only be a position change because the nearest positional neighbour is exceptionally close value-wise. Also, as mentioned above, the addition of an extra zero will alter the clumpiness value of a line when calculating a direct measurement and the number greater than expected will be dependent on the lines of random distribution used. Therefore, although a code ‘0’ would be preferable, a code of ‘1’ is also acceptable.

Table 7.4. First run using line 6 from Figure 7.9.b. RS denotes Spearman’s coefficient and Pos is the order position for the unaltered and altered line 6

<table>
<thead>
<tr>
<th>Measure</th>
<th>Rs</th>
<th>Value1</th>
<th>Value2</th>
<th>Pos1</th>
<th>Pos2</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1</td>
<td>3.176</td>
<td>2.944</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Largest group</td>
<td>0.99</td>
<td>14</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Gapped largest group (1.5x)</td>
<td>0.99</td>
<td>14</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Proportion</td>
<td>0.96</td>
<td>0.218</td>
<td>0.284</td>
<td>11</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Frequency</td>
<td>0.99</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Intensity (16)</td>
<td>1</td>
<td>56</td>
<td>55</td>
<td>9</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Markov</td>
<td>Average (2)</td>
<td>0.69</td>
<td>0.264</td>
<td>0.7</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Markov</td>
<td>Expected (8)</td>
<td>0.88</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2 HMM</td>
<td>Sum (4p)</td>
<td>0.99</td>
<td>3.411</td>
<td>3.269</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3 HMM</td>
<td>Sum (6p)</td>
<td>1</td>
<td>2.046</td>
<td>2.046</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Distance and count</td>
<td>½ win + 1 (4)</td>
<td>1</td>
<td>21</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Distance and count</td>
<td>Expected (4)</td>
<td>0.99</td>
<td>43</td>
<td>45</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum</td>
<td>0.93</td>
<td>17</td>
<td>14</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Expected</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum</td>
<td>0.98</td>
<td>1.53E+10</td>
<td>1.05E+10</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>L-function</td>
<td>Largest</td>
<td>0.99</td>
<td>2.72E+8</td>
<td>1.98E+8</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>L-function</td>
<td>Expected</td>
<td>0.99</td>
<td>87</td>
<td>90</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Chi-square</td>
<td>1</td>
<td>10.529</td>
<td>10.530</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CUSUM</td>
<td>1</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 7.5. Second run using line 6 from Figure 7.9.c. RS denotes Spearman’s coefficient and Pos is the order position for the unaltered and altered line 6

<table>
<thead>
<tr>
<th>Measure</th>
<th>Rs</th>
<th>Value1</th>
<th>Value2</th>
<th>Pos1</th>
<th>Pos2</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>1</td>
<td>3.176</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Largest group</td>
<td>0.99</td>
<td>14</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>1</td>
<td>14</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Gapped largest group (1.5x)</td>
<td>1.00</td>
<td>14</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Proportion</td>
<td>0.96</td>
<td>0.218</td>
<td>0.289</td>
<td>11</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Frequency</td>
<td>1.00</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Intensity (16)</td>
<td>1</td>
<td>56</td>
<td>56</td>
<td>9</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Markov</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average (2)</td>
<td>0.69</td>
<td>0.264</td>
<td>0.703</td>
<td>11</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Expected (8)</td>
<td>0.88</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2 HMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum (2p)</td>
<td>0.99</td>
<td>3.411</td>
<td>3.266</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3 HMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum (6p)</td>
<td>1</td>
<td>2.046</td>
<td>2.035</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Distance and count</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>½ win + 1 (4)</td>
<td>1</td>
<td>21</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Expected (4)</td>
<td>0.99</td>
<td>40</td>
<td>42</td>
<td>12</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Neighbours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td>0.98</td>
<td>17</td>
<td>15</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Expected</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>L-function</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sum</td>
<td>1</td>
<td>1.55E+10</td>
<td>1.55E+10</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Largest</td>
<td>2.90E+8</td>
<td>2.86E+8</td>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Expected</td>
<td>1</td>
<td>88</td>
<td>87</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Chi-square</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUSUM</td>
<td>0.99</td>
<td>10.529</td>
<td>110.24</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CUSUM</td>
<td>0.97</td>
<td>25</td>
<td>26</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

From the rank order correlation in Section 7.4.1 and Table 7.2, only four measures had the majority of their correlations against the ordering by the subjects above 0.9. These were the largest group, Markov Average, Distance and count with half the window size plus one (DCHWPO) and the CUSUM measure. In both the test sets described here, the addition or the relocation of an additional zero in the largest group of line six causes a considerable difference to the Largest group and Markov average values. For the DCHWPO, although there is no change in the rank position, the actual values of the measure varied in both cases. This means that if there were any other lines with similar values a rank order change would have occurred. For the CUSUM measure although the insertion of an additional zero caused no change in either value or position, the relocation of the zero caused a considerable change in rank position for a very small change in measure value. However, as mentioned in the previous Section (7.4.2.1), contrary to the evidence in the second test, it would appear that the CUSUM measure may tolerate the occasional ‘lonely zero’.
Of the other measures, the good result in the Neighbours expected measure for both test sets might only be due to the small values in all the lines for this particular measure. Another point to note, is that the first eleven measures, except the Intensity measure, fair the same regardless of maintaining the intensity or not. The variation in intensity may affect the Intensity measure since this is central to that measure’s approach.

7.4.3 ‘Big-Oh’ Notation

The ‘Big-Oh’ notation, or O-notation, is used to specify “relative, rather than absolute, performance”, (Solter 2005), of computer programs. Rather than specifying the actual time in seconds a program requires to perform its operations, which varies dependent on a computer’s specifications, the O-notation denotes how the program will scale as the problem, or input size increases, (Hasti 2001). This is also known as the complexity of a program. Solter (2005) describes some common complexity categories as:

- Constant complexity – there is no change to the number of operations performed, it is independent of the input size, written $O(1)$
- Linear complexity – the number of operations is directly proportional to the input size, such that doubling the input size, doubles the number of operations, written $O(n)$
- Quadratic complexity – the number of operations is a function of the square of the input size, such that doubling the input size would quadruple the number of operations, written $O(n^2)$
- Logarithmic complexity – the number of operations is a function of the logarithm base 2 of the input size, written $O(\log n)$

When determining the complexity of a program, it is the worst-case-scenario that is desired, what is the maximum possible number of operations that a program requires to run. For example, when searching an unordered list, a program would at most need to do a ‘check list element’ operation for every element in the list if the item searched for was in the last position or not present, resulting in a linear complexity. Additionally if a program contains multiple methods, each with its own complexity, it is the worst complexity that is reported as the program’s O-notation. For example,
if a program has a constant complexity method and a linear complexity method, the overall complexity would be linear.

There are no constants in O-notation, such as $O(2^n)$, since as $n$ becomes increasingly large, constants are of no consequence since more complex programs will still be slower. Constant complexity is faster than linear complexity, which is faster than quadratic complexity (Hasti 2001). However, in comparison of different programs, where both have the same O-notation, the constants may be included to distinguish which is the faster program, $O(2n)$ would be slower than $O(n)$.

7.4.3.1 Determining Complexity

The determination of complexity of a program is dependent on the type of operations used in the program:

1. A basic operation, such arithmetic operations, tests or read/writes – the complexity will be constant, $O(1)$
2. A sequence of operations – the worst case will be the most complex, or ‘maximum’ of the sequence of operations. If there are four operations, three constant and one linear the maximum would be the linear operation giving $O(n)$. If multiple operations have the same complexity, such as three at $O(n)$, this would be $O(3n)$ but as constants are dropped this reduces to $O(n)$
3. If-then-else operation – the rate-limiting-step is the most complex of each part of the if-then-else operation. If the ‘if’ part is linear and the ‘else’ part constant, the overall complexity will be linear. If both have the same complexity the overall complexity will simply be that complexity
4. For loop operation – if the upper bound of the loop is a specified value, the loop would loop the same number of times regardless of the input and would be constant. However, if the upper bound of the loop is $n$, then it would loop $n$ times giving a linear complexity, $O(n)$
5. Nested If-then-else operation – the most complex part of the inner if-then-else operation would become the complexity of the part of the outer if-then-else operation where it appeared. The outer if-then-else operation would then be treated the same as single if-then-else operation.
6. Nested for loop operation – provided the upper bound on either loop is not specified, for each iteration of the outer loop, the inner loop will iterate $m$
times. Therefore, if the outer loop iterates \( n \) times, this would give a complexity of \( n^m \), which would be \( n^2 \) if \( m = n \).

7. Programs with methods – a method can be viewed as a separate ‘mini-program’ and as such will have its own complexity. Within the main program these method calls would then be viewed as a sequence of operations as described in point 2 above. For example, if a program calls a method with linear complexity then two further methods each with constant complexity, the overall complexity of the program would be linear.

As an example of how the \( O \)-notation can be applied to a program is described in the next Section, which details how the complexity of the CUSUM clumpiness measure was determined. The CUSUM measure was chosen since its code is not too long but illustrates most of the operations types given above, specifically, instead of a nested ‘if-then-else’ or for loop there are ‘if-then-else’ operations within for loops.

7.4.3.2 Complexities in programs – an example

This Section uses the CUSUM clumpiness program as an example as to using the \( O \)-notation to determine the complexity of a program. Figure 7.10.a shows an abbreviated version of the code for the main CUSUM measure program. Initialisations, assignments and increments are all basic operations with a complexity of \( O(1) \), and if an ‘if-else’ operation does not have an else component the complexity will be that of the if component since the else will, effectively, have a complexity of zero, \( O(0) \). Figure 7.10 also shows how the various complexities in the program are combined using right-braces with either a multiplication sign for operations within for loops or ‘max’ for a sequence operations described in point 2 of Section 7.4.3.1. The input for the program is a binary sequence containing \( n \) characters.

The method, Figure 7.10.b, contains a for loop, that loops through all \( n \) positions in the sequence, \( O(n) \), which itself contains an ‘if’ operation with complexity \( O(1) \), giving the for loop a complexity of \( O(n) \times O(1) \), which is \( O(n) \), shown by the multiplication right-brace. Before and after this for loop are two instances of basic operations, each with \( O(1) \) complexity, and these are combined with the for loop complexity as to which is maximal, i.e. \( \max (O(1), O(n), O(1)) \), as shown by the
‘max’ right-brace. This gives the methods overall complexity as $O(n)$, which can then be taken back into the main block of the program.

In the main block, Figure 7.10.a, the inner most, left most, right-brace relates to the complete if-else operation and since both parts have a complexity of $O(1)$, its complexity is $O(1)$. The second right-brace combines all the operation complexities inside a for loop, which loops $n$ times, $O(n)$. The operations inside the for loop are a sequence of operations and as such the complexity is the maximum of their complexities, i.e. $O(1)$ and this is combined with the for loop by a multiplication, giving $O(n)$, the third right-brace. The final, right most right-brace combines, by maximisation, the for loop with four basic operations and the complexity of the method determined earlier resulting in an overall complexity of the CUSUM clumpiness measure program of $O(n)$, or $O(2n)$ if it is necessary to compare this program to another program with $O(n)$ complexity.

| Initialise 2 variables | \[ O(1) \] |
| Call method Stats | \[ O(N) \] |
| Initialise 2 variables | \[ O(1) \] |
| For $i = 1$ To $N$ | \[ O(N) \] |
| Assign 2 variable values | \[ O(1) \] |
| Initialise variable $C(i)$ | \[ O(1) \] |
| If $i = 1$ Then | \[ O(N) \] |
| Assign value to $C(i)$ | \[ O(1) \] |
| Else | \[ O(1) \] |
| Assign value to $C(i)$ | \[ O(1) \] |
| If $C(i) < 0$ Then | \[ O(1) \] |
| Re-initialise $C(i)$ | \[ O(N) \] |
| If $C(i) > \text{variable}$ Then | \[ O(1) \] |
| Increment $C(i)$ | \[ O(1) \] |
| Next $i$ | |
| a. | |

| Initialise variable | \[ O(1) \] |
| For $s = 1$ To $N$ | \[ O(N) \] |
| If $\text{N value} = "1"$ Then | \[ O(N) \] |
| Increment variable | \[ O(1) \] |
| Next $s$ | |
| Assign 2 variable values | \[ O(1) \] |
| b. | |

Figure 7.10. Pseudo-code for the CUSUM measure program showing a. the main body of the program and b. the method ‘Stats’ called from the main body.

In the same way the complexity of the other clumpiness measure programs was determined and these are shown in
Table 7.6. The second column shows the O-notations as they would normally be reported and third shows the O-notation with their constants and lower order terms included for comparison between those measures with the same O-notation. The rows have been coloured green if both the reduced and expanded O-notations are \( O(n) \) and yellow if the reduced is \( O(n) \) but the expanded includes a constant, all other greater magnitudes are left white.

### Table 7.6. O-Notation for each of the clumpiness measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>O-Notation</th>
<th>Expanded O-Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Largest group</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gapped largest group (1.5)</td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td>Proportion</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Frequency</td>
<td>All groups</td>
<td>( O(n) )</td>
</tr>
<tr>
<td></td>
<td>Larger only</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Intensity</td>
<td>Overall (16)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Markov</td>
<td>Average (2)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td></td>
<td>Expected (8)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>2 HMM</td>
<td>Sum (2p)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>3 HMM</td>
<td>Sum (6p)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Distance and count</td>
<td>( \frac{1}{2} ) win +1 (4)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td></td>
<td>Expected (4)</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum</td>
<td>( O(n) )</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td></td>
<td>Largest</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td></td>
<td>Expected</td>
<td>( O(n^3) )</td>
</tr>
<tr>
<td>Chi-squared</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>CUSUM</td>
<td>( O(n) )</td>
<td>( O(2n) )</td>
</tr>
</tbody>
</table>

As can be seen from the table, the majority of the measures have a complexity of \( O(n) \). The notable exceptions are the measures based on the L-function, which each have cubic complexity and the Gapped largest group measure, which has quadratic complexity. It should be noted that the extra zeros largest group measure does not have a O-Notation because this particular measure was only done manually and therefore, having no program to assess.
7.4.4 Summary

Table 7.7 summarises the results of both the Spearman’s correlation, $R_s$, to a manually ordered dataset, though only the consensus order correlation is shown, the introduction of a ‘lonely zero’ into the three test sets and O-notation data. The last two columns in the table give the time in seconds for the measure to run the initial dataset (the Spearman’s correlation) on a Pentium II processor with 130Mb of RAM and the decision as to whether that particular measure is a suitable candidate for the clumpiness measure. This decision is based on the three tests, correlation, lonely zero tolerance and time taken. Following the colour coding defined for the Spearman’s correlation, Section 7.4.1, and lonely zeros, Section 7.4.2, a measure is discarded if

1. The correlation is less than 0.9 (green colour coded in the table)
2. Two or more of the lonely zero test sets resulted in a white code ‘2’

An exception to this second rule is the CUSUM measure for the reasons outlined in Section 7.4.2.1.

The colour coding of the timing is as follows:

- Green represents faster than 0.05 seconds
- White represents between 0.05 and 0.1 seconds
- Yellow represents between 0.1 and 1 seconds
- Orange represents slower than 1 second

with a measure being discarded if it takes longer than 0.05 seconds to complete, any non-green. In the last column is written which of the three tests a particular measure has failed, for example, the average measure is very fast and fairs well with the lonely zero tolerance but its accuracy, correlation, is only 0.85, therefore it fails on the correlation test and “$R_s$” is written in the discard column. Similarly “LZ” and “Time” denotes discarding because the measure failed the lonely zero and time tests respectively.
<table>
<thead>
<tr>
<th>Measure</th>
<th>Rs</th>
<th>Lonely Zero</th>
<th>0.02119</th>
<th>0.01213</th>
<th>0.01213</th>
<th>R_s</th>
<th>LZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.85</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.02119</td>
<td>O(n)</td>
<td></td>
</tr>
<tr>
<td>Largest group</td>
<td>0.96</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.01213</td>
<td>O(n)</td>
<td>LZ</td>
</tr>
<tr>
<td>Xtra 0s largest group (2)</td>
<td>0.82</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Gapped largest group (1.5)</td>
<td>0.99</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0305</td>
<td>O(n^2)</td>
<td>R_s</td>
</tr>
<tr>
<td>Proportion</td>
<td>0.76</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.02213</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>Frequency</td>
<td>All groups</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.64388</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>Larger only</td>
<td>0.77</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.38916</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>Intensity</td>
<td>Overall (16)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>138.8896</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>Expected</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>138.9096</td>
<td>O(n)</td>
<td></td>
</tr>
<tr>
<td>Markov</td>
<td>Average (2)</td>
<td>0.95</td>
<td>1</td>
<td>2</td>
<td>11.25000</td>
<td>O(n)</td>
<td>LZ</td>
</tr>
<tr>
<td>Markov</td>
<td>Average (5)</td>
<td>0.95</td>
<td>-</td>
<td>-</td>
<td>11.31300</td>
<td>O(n)</td>
<td>Time</td>
</tr>
<tr>
<td>Markov</td>
<td>Expected (8)</td>
<td>0.87</td>
<td>0</td>
<td>2</td>
<td>0.17078</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>2 HMM</td>
<td>Sum (2p)</td>
<td>0.93</td>
<td>1</td>
<td>2</td>
<td>28.14944</td>
<td>O(n)</td>
<td>LZ</td>
</tr>
<tr>
<td>3 HMM</td>
<td>Sum (6p)</td>
<td>1.00</td>
<td>1</td>
<td>0</td>
<td>55.90572</td>
<td>O(n)</td>
<td>Time</td>
</tr>
<tr>
<td>Distance</td>
<td>½ win +1 (4)</td>
<td>0.95</td>
<td>1</td>
<td>1</td>
<td>0.01600</td>
<td>O(n)</td>
<td></td>
</tr>
<tr>
<td>and count</td>
<td>Expected (4)</td>
<td>0.73</td>
<td>0</td>
<td>2</td>
<td>12.14100</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Sum</td>
<td>0.91</td>
<td>1</td>
<td>2</td>
<td>0.06300</td>
<td>O(n)</td>
<td>LZ</td>
</tr>
<tr>
<td>Neighbours</td>
<td>Expected</td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>1.23500</td>
<td>O(n)</td>
<td>R_s</td>
</tr>
<tr>
<td>L-function</td>
<td>Sum</td>
<td>0.87</td>
<td>1</td>
<td>2</td>
<td>700.3889</td>
<td>O(n^2)</td>
<td>R_s</td>
</tr>
<tr>
<td>L-function</td>
<td>Largest</td>
<td>0.77</td>
<td>0</td>
<td>2</td>
<td>700.3989</td>
<td>O(n^2)</td>
<td>R_s</td>
</tr>
<tr>
<td>L-function</td>
<td>Expected</td>
<td>0.92</td>
<td>0</td>
<td>2</td>
<td>689.7970</td>
<td>O(n^2)</td>
<td>Time</td>
</tr>
<tr>
<td>Chi-squared</td>
<td>0.91</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.23888</td>
<td>O(n)</td>
<td></td>
</tr>
<tr>
<td>CUSUM</td>
<td>0.97</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0.02009</td>
<td>O(n)</td>
<td></td>
</tr>
</tbody>
</table>

After the decision process, excluding the CUSUM measure for the lonely zero tolerance rule, only three measures are left, that of the gapped largest group, distance and count greater than $\frac{1}{2}$ the window size + 1 and the CUSUM measure itself. However, the gapped largest group measure as a O-notation of $O(n^2)$, whereas the other two are only $O(n)$ and would, therefore, scale better. Between the Count and CUSUM measures, the Count is slightly faster, but the CUSUM has a better correlation. Although the difference in correlation is small in the test dataset so is the difference in speed and it was felt that a greater accuracy would be preferable over the slight increase in speed.

The CUSUM measure was chosen as the most suitable measure for the purposes of clumpiness quantification since it had the best correlation of all the candidates, was reasonably efficient computationally and should scale well with an O-notation of $O(n)$. It was felt that the issue of the lonely zero was adequately covered, as
described in section 7.4.2.1, despite the appearance that the introduction of an additional zero to a larger clump would render this measure unsuitable. The next Chapter describes the evaluation of a prototype incorporating this chosen measure for the purposes of in-situ evaluation of the CUSUM measure to quantify clumpiness. The design of this prototype can be found in Appendix E and its code in Appendix F.
Chapter 7 detailed the evaluation of the proposed clumpiness measures resulting in the selection of the measure that appeared most suitable in laboratory conditions, specifically, that performed best on synthetic data as defined by specified criteria. However, it is also necessary to test this method in a more realistic situation, i.e. in BLAST results retrieved by scientists in the course of their research. To this end, a prototype was created incorporating this ‘best’ method, specifically the CUSUM measure. In this case a prototype being an automation, or semi-functional program that would conduct BLAST searches and limited subsequent analysis, returning results to the scientist on which they could draw conclusions. The design and implementation of this prototype can be found in Appendix E.

With the prototype created it is necessary to evaluate its effectiveness as a system by which the users may obtain relevant information for protein function determination. To this end, the first half of this Chapter is concerned with the evaluation and feedback obtained from the testing of the prototype by BLAST users, ‘experts’.

However, there is also the issue of ensuring that the clumpiness measure has properly captured the concept of clumpiness within the distribution of events in a one-dimensional field. That in mind, the second half of this Chapter focuses on the ability of the CUSUM measure to quantify clumpiness, in itself, which needs further testing by subjects from varying fields, ‘non-experts’, with the aim to eventually enable the concept to be transferable to other fields of research.

8.1 Prototype Usage
The purpose of the prototype is to assess the ability of the measure to assist the scientist, ‘experts’, in their research. Therefore, the desired outcome of the prototype usage would be that the extra analysis would bring ‘twilight’ BLAST result hits to the notice of the scientist by bringing them further up the list. Although these results would not be as similar to the query sequence, they may be functionally relevant and otherwise missed within the bulk of less similar and irrelevant hits. This increase in biologically relevant results should assist in the decision-making processes and accelerate research progress or indicate new avenues of investigation.
8.1.1 Methodology

The evaluation of the prototype takes the form of feedback from ‘experts’ using the system within the scope of their research and has three main objectives. The first of these is the minor issue of the ease of use of the program itself. If the program is difficult to use or requires a steep learning curve to operate it will hinder rather than help the user. Secondly, and much more importantly, is the measure’s ability to capture the notion of clumpiness. Is the assignment of a clumpiness value, and thence the alignment order a valid portrayal of the differing levels of clumpiness within the BLAST result alignments. This is the most important issue, since the basis of this research project is the quantification of the notion of clumpiness. The third objective is, again, of great importance to the overall evaluation of the clumpiness measure and is explored in more detail in Chapter 9. This is the ability of the measure to assist in the function determination of a protein sequence; does it actually bring less similar but still relevant hits to the users notice or does it just confuse the matter.

The feedback will take the form of a semi-structured interview with the interviewee discussing a set of results produced by the prototype, with reference to questions posed by the interviewer. In order that the interviewee has ample time in which to study the prototype and the results are familiar to them, the interviewee is provided with a copy of the prototype at least two weeks prior to the interview. In this way, the interviewee can use the prototype on their own data, with which they are most familiar, and the added bonus of possibly highlighting a result that they would otherwise have missed, thereby, assisting in their research.

The questions posed form a guideline only, allowing the interviewee the freedom to express their views on the prototype and to illustrate their points with aspects of the reports, but still ensuring that the focus remains on the usability of the prototype and the validity of the measure. In terms of the ease of use of the prototype the questions are straightforward:

1. Did the prototype install?
2. Was the manual easy to follow, explaining all relevant procedures from the prototype in a clear manner?
3. Was the prototype easy to use with no difficulty entering and retrieving data?
Questions for the more important aspects of the validity and effectiveness of the measure are broken down into rearrangement, relevance and clumpiness. The question of arrangement is concerned with the reordering of the BLAST hits and whether any hits from the lower regions have been brought into the higher regions, with the view of emphasising a result the interviewee may not have previously considered based on the BLAST measure alone. For example, in Figure 8.1, one particular hit, circled, has been moved from the seventeenth position in the BLAST result to the eleventh in the Clumpiness report and, though this is not an extreme example, it illustrates the potential of the clumpiness measure to move possibly relevant hits higher up. Related to this, the question of relevance focuses on these migrated BLAST hits and whether they were relevant to the interviewee’s work, confirming ideas they may already have or introducing new possibilities. Alternatively, did shifting these BLAST hits only confuse the data by introducing irrelevant hits? The third element of this part of the feedback relates to the most important aspect of the capturing of the notion of clumpiness and concentrates on the actual alignments. How well the interviewee felt that the clumpiness measure distinguished between clumpy and non-clumpy alignments and would they, therefore, agree with the order into which the prototype had rearranged the alignments. Would they agree with the order into which the clumpiness measure placed each hit or would they make any alterations?
8.1.2 Results and Discussion

In terms of the technical feedback all the subjects found the prototype easy to use, though there were a few ‘teething problems’ with the installation of the program. The inclusion of the associated Visual Basic runtime files with the installation, alleviated this. However, though the prototype ran fast on a standard desktop computer, older machines, with lower specifications, took longer. For example, a sequence of length 172 producing 509 results took two and a half minutes on a desktop machine with a 2.66 Ghz Intel Celeron processor with 1Gb of RAM and on an older machine, using a Pentium II processor with 130Mb of RAM, this took six minutes. As there is no guarantee that the users will have access to up-to-date machines, one solution suggested by one of the subjects was to restrict the number of results returned, however, this opposed the idea of this analysis, where as many results as possible are wanted in order to ensure that all potentially relevant results are found. An alternative solution would be to place the program and database onto a web server and allow access through the Internet as is currently done with the BLAST program suite. This will be discussed further in Section 10.1.1.
Another issue arose with the layout of the clumpiness report. Initially, any BLAST hit with the number of matches in its alignment greater than a set percentage was not included in the reordering of the hits, they were assigned the value ‘N/A’, left in the original order but placed above those hits with an actual clumpiness value, Figure 8.2. This was done, since a hit with many matches would be of interest to the user and viewed anyway, however, since this cut-off was based on percentage the actual number of matches in a short alignment will be a lot less than that in a longer alignment. Therefore, short but highly similar hits were also included in this ‘N/A’ group leading to false indications, especially if the short hit was to vector sequence or other contamination. As a result, subsequent runs of the prototype assigned a clumpiness value to all the hits; those that were highly similar should also receive a higher clumpiness value, though the standardisation should counteract the risk of moving up any of the short hits, which are more likely to be irrelevant.

**Figure 8.2.** Original clumpiness report with higher clumpiness values, circled, not used in the reordering process

Another example of how the reordering of the BLAST results may be useful to the scientist may be seen in Figure 8.3. Here, the subject already knows the function of the query sequence, an Ubiquitin conjugating enzyme, which catalyzes the attachment of the Ubiquitin protein to its target (Jentsch 2001) and is looking for confirmatory information and other family members. Figure 8.3 shows a hit with a much lower score (score circled) being placed in a high position amongst high scoring hits due to its higher clumpiness value. Figure 8.4 shows the actual alignment for this hit (first alignment) and the area of the PROSITE signature (Jentsch 2001) for the Ubiquitin conjugating enzyme highlighted in the box showing this to be a good candidate for either another example of a known member of this protein family or a new protein family member. This is useful when there is a huge family tree with the more distant relative being lost in the mass of closer and,
therefore, more similar relatives. Also, though a search with the PROSITE signature may accumulate the majority of members of a particular family, it is not guaranteed to find them all. Indeed in the PROSITE signature for the Ubiquitin conjugating enzyme details an exception to the pattern in yeast UBC6, i.e. the pattern in PROSITE will not find this yeast member of the Ubiquitin conjugating enzyme family member. Therefore, purely searching the sequence databases with the PROSITE signature may, in fact, miss some of the more distant relatives. The subject felt that this method offered a different perspective that may catch these distant, yet important family members.

<table>
<thead>
<tr>
<th>Sequences producing significant alignments:</th>
<th>Score</th>
<th>E</th>
<th>Clumpiness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothetical LOC496585 [Xenopus tropicalis]</td>
<td>318</td>
<td>2.00E-85</td>
<td>216.05</td>
</tr>
<tr>
<td>novel protein [sgc:63554] [Danio rerio]</td>
<td>317</td>
<td>3.00E-85</td>
<td>93.409</td>
</tr>
<tr>
<td>PREDICTED: similar to ubiquitin-conjugating enzyme E2,...</td>
<td>354</td>
<td>2.00E-95</td>
<td>91.953</td>
</tr>
<tr>
<td>Hypothetical protein CB513052 [Caenorhabditis briggsae]</td>
<td>251</td>
<td>2.00E-65</td>
<td>66.723</td>
</tr>
<tr>
<td>hypothetical protein [Debaryomyces hansenii CBS767]</td>
<td>41</td>
<td>3.00E-02</td>
<td>63.283</td>
</tr>
<tr>
<td>Ubiquitin conjugating enzyme protein 6 [Caenorhabditis...</td>
<td>247</td>
<td>3.00E-64</td>
<td>55.572</td>
</tr>
<tr>
<td>hypothetical protein CNBE3270 [Cryptococcus neoformans...</td>
<td>174</td>
<td>3.00E-42</td>
<td>52.358</td>
</tr>
<tr>
<td>ubiquitin conjugating enzyme, putative [Cryptococcus n...</td>
<td>174</td>
<td>3.00E-42</td>
<td>52.358</td>
</tr>
</tbody>
</table>

Figure 8.3. Section of the report page for the search using ube2ji showing sequence raised in order (circled)

Another use of sequence signatures, or motifs, is to highlight the conserved sequence of possible structural elements. Again, this method gives the structural prediction a new perspective that may detect otherwise missed relevant information. However, care should be taken, as it is not always possible to predict the three-dimensional structure from the Amino acid sequence alone (Russell et al. 1996). That said, it was opined that the conversion of the positive matches, ‘+’ in the alignments, into ones when creating the binary line on which the clumpiness calculations are conducted, would not pose an issue from the perspective of structural analysis. The replacement of amino acids with similar physiochemical properties, as in the case of the positive matches, should not affect the three-dimensional shape of the protein provided that all internal bonds, as described in Section 4.1.3, are maintained.
49654761:CAG87262.1: unnamed protein product [Debaryomyces hansenii CBS767]
Length = 568

Score = 41 (bits), E-Value = 0.037, Clumpiness = 63.283
Identities = 64/258 (24%), Positives = 99/258 (38%), Gaps = 52/258 (20%)

Query: 11
  AVKRLMKEAAELKDPDTHYHAQP----------EDNLFEWHTVRGPSSD-----FDGGVHY
  A K R E   +L K    H A P   N+ E W    V   P S + G +
Sbjct: 2
  AEKRLFEYQNKLKTPAH-ENFIQVSSLPPDSNILEEAVSVSPSKSGSRYYNGKRW

Score: 60 (bits), E-Value = 3E-64, Clumpiness = 55.572
Identities = 116/188 (61%), Positives = 146/188 (77%), Gaps = 1/188 (0%)

Query: 1
  RIVLPPEYPMKPPS--------ILLTANGRFVYGGKICLBI5GHHPETWQPSWSIATTALL
  I   P  YP  PP  I  ++ N + G  ICL  I E W  P+W+++ ++
Sbjct: 2
  LTVHPPTYPMKPPNIDLPDLNLFLQDKTFESVQYMYKDTCTSDKGD----DKITGVK

Figure 8.4. Alignments for the moved sequence and the one immediately below it

However, there was an issue in the assessment of the reordering of the alignments by the clumpiness values of each of the hits. When considering the actual alignments it appeared that certain alignments were placed higher than others even though, on manual inspection, the clumpiness seems lower. For example, Figure 8.4 shows the repositioned unknown protein matching to the Ubiquitin conjugation enzyme, introduced above, and the subsequent alignment of the next hit. As can clearly be seen, the clustering of the matches in the repositioned alignment are much more dispersed than in the subsequent alignment. Even though the length of the first alignment is longer than the second, the difference is not so great as to position these...
alignments as they are if done manually. Also, the intensity is much lower, as defined by the much lower similarity score. This possibly incorrect clumpiness valuation may be a result of the standardisation method used, as defined in Appendix Section E.3.5. In that case the standardisation takes the form of multiplying the clumpiness value only by the coverage of query sequence by the hit. The addition of multiplying by the intensity reduces the overall clumpiness value of each of the hits and lowers the repositioned hit highlighted in Figure 8.3 from 15th to 31st, which is still much higher than its original 806th position but the new subsequent alignment seems more suitably less clumpy, though its intensity is higher, its length is much shorter. In the previous order the subject disliked the migrated hit being immediately after a hit that was virtually exact. He also felt that the migrated hit may have picked up on more distant motifs missed by the subsequent hit in the new order since the migrated hit spanned protein positions 11 to 224, whereas the subsequent hit only spanned protein positions 12 to 176.

Further testing is required to determine if further alteration of the standardisation method improves the ordering or, indeed, if anymore is necessary. It may be that it is in the standardisation's original or current form that the most benefit will be gained.

Another evaluation of the prototype to ensure that the clumpiness measure is reordering the data and not simply giving the same information as the E-value is to perform a comparison of the E-values and clumpiness values. If graphically viewed any large variations in the clumpiness will be immediately visible, provided the data are ordered by the E-value. For example, Figure 8.5 shows the comparison of the Ubiquitin conjugating enzyme BLAST E-values and clumpiness measure values. As this data is ordered by E-value, this has a fairly smooth curve, the blue line, and is displayed on a logarithmic scale in order to be visible when compared to the score. However, the clumpiness values do not follow a smooth curve with variation across all the sequences, the red line, though with a general reduction towards the poorer E-values, as shown by the logarithmic trend line, in black. The reverse would be true if the data were ordered by clumpiness, with a smooth clumpiness curve and an irregular line for the E-value, therefore, showing that there is a clear difference in the preferred order for each of the measurements. This relates to the differences between the clumpiness and E-values with each of the 'blips' indicating where the clumpiness
value would have rated that hit a higher position and it is this that the proposed purpose of the clumpiness measure, raising to notice those possibly relevant hits that may be missed when viewed by E-value alone. For example, the clear spike towards the right of the graph, labelled X, is the interesting result highlighted above where the clumpiness value repositions this particular hit from 806th to 31st, indicating a possible distant relative.

![Graph of Clumpiness vs. E-Value](image-url)

**Figure 8.5.** Comparison of BLAST e-values to Clumpiness value ordered by e-value

Unfortunately, the subject set on which the prototype has thus far been tested is woefully small, and consist of only three subjects. Therefore, though the preliminary results are good, more study is required in order to verify these results. Also, it must again be stressed that any information highlighted by BLAST and the clumpiness measure are purely indicators and any firm decisions must be verified by biochemical testing in the laboratory.

### 8.2 ‘Non-expert’ Testing for Possible Measure Transferability

As this measure quantifies the degree of clustering of events within a one-dimensional domain, it need not be confined to the evaluation of BLAST results. However, it is first necessary to conduct further testing on the validity of the ranking by the clumpiness measure. That is, given a set of one-dimensional domains containing distributed discrete events, does the measure place these domains in an
order such that their distributed events are graded from most to least clustered. To this end, as in the initial measure evaluation, Chapter 7, subjects were asked to order a number of strips depicting variations in the distribution of events, as to how they would grade them from clumpy to unclumpy. These orderings could then be used to assess the order into which the clumpiness measure places these same strips via the Spearman’s rank correlation coefficient.

8.2.1 Methodology

The hidden Markov model described in Section 7.2.2 was used to generate a set of lines with distributed events as ones and non-events as zeros and these were converted to strips with two coloured blocks designating an event or a non-event. During the creation of the approximate model, Section 7.3, black and white strips were used, Figure 7.5. However, from feedback from the subjects involved in the model creation, alterations to these strips were deemed necessary. Firstly, there was some confusion when scanning the original strips as to the separation of the individual events or blocks. Since black and white were used to colour code the blocks, it was necessary to convert the lines between consecutive black blocks from black to white in order to visually separate them, thereby giving strips with black lines on white blocks and white lines on black blocks, Figure 8.6. This gave rise to some subjects being confused as to whether the white lines were separate white blocks or merely dividers between the black blocks. This may stem from the determination of figure, or object, from background in perception where comparatively smaller areas will be viewed as the object in front of the larger area of background (Schiffman 2000).

![Figure 8.6. Section of original strips illustrating confusion of white lines](image)

To alleviate this problem the lines needed to be maintained as a constant colour and, therefore, either all the lines needed to be changed to a third colour using the block colours of white and black, or all the lines kept black with different colours used for the different blocks. The latter solution is adopted here, maintaining the black lines around the blocks. Non-events also remained white since white is a stereotypical
background colour such as the white of pages in a book. For the event blocks the colour green was chosen for three reasons:

1. Green stimulates all three colour sensors, cones, in the eye, which is advantageous if any subjects are ‘colour deficient’, lacking input from one or more cones (Goldstein 2002). There is still input from those that remain to differentiate from the white blocks.

2. The contrast between the green events and the black lines is much lower than the contrast between the white non-events and the black lines. This emphasises the clumps since there is less distinction between neighbouring event blocks so they appear more related. In the original strips the contrast between the blocks and dividing lines was the same for both events and non-events.

3. The choice of white as the background stems from the white frequently being used as the background, as in the pages of a book.

Another factor is the size of the individual blocks within the strip. If the blocks were too small then it will be a strain to view them, conversely, if they were too big then the strip would have to be so large, and still have a meaningful number of blocks, that it would not be easy to visualise the entire strip within the field of view. This would result in difficulties in analysis of each of the strips, as the subject would have to scan along the strip rather than examining it as a whole and may result in them considering the strip in subsections. An example may be seen in the original black and white strips where the confusion arose as to whether the white lines between the black blocks were merely lines or an event. Related to this is the number of blocks used in the strip, too few blocks and there will not be enough to form any meaningful distributions and too many blocks and, again, the strip will become too long to view as a whole. Therefore, fifty blocks was used each being 5mm wide.

The final consideration is the number of strips, even with more aesthetic colours, larger blocks and a reduced number of blocks, viewing too many of these strips will eventually cause considerable strain on the eye, with rather unpleasant distorted

Figure 8.7. Example strip used in effectiveness of clumpiness measure
after-images. Nevertheless, it would be more helpful to be able to order as many as possible in order to adequately test the clumpiness measure. Therefore, 25 strips was deemed sufficient to challenge the clumpiness measure without causing too much strain on the subjects, though in creating the binary lines used to make the strips only 24 had the necessarily similar intensities but it was felt that this was close enough.

The actual form of the testing was two-pronged and the full instructions may be found in Appendix H. Firstly the subjects were asked to order the 24 strips described above and secondly a second, much larger set of strips is created and ordered by the clumpiness measure. As mentioned above, the larger the test dataset, the better the evaluation, therefore, much as the ‘expert’ subjects were asked to assess the order of the alignments in the prototype report and describe any alterations, Section 8.1.1, the same is done here with a dataset of 120 sequences. That is, given the ordered dataset, the subjects scanned down the list, describing possible alterations, if any, to the order and describe why.

In addition to this question, the subjects are asked to describe, where possible, the method or methods used in ordering the strips and if their scanning technique differed in any way. This parallels the ‘expert’ ordering of strips with the emphasis on clarifying the intuitive thought processes behind the decisions.

8.2.2 Results and Discussion

For the first part of this evaluation, the ordering of the 24 strips by the clumpiness measure can then be assessed compared to each of the subjects order by Spearman’s rank correlation coefficient (Section 7.1). The results of this may be seen in Table 8.1, where the letters used in column one are the designation of each of the strips, C, in the second column, is the clumpiness measure ranking for each strip and Subject 1 – 12, in the remaining columns are the ranking for each strip by each of the subjects.

As can be seen from the table there is a fair to good correlation between the clumpiness measure order and that of all but two of the subjects. As already mentioned, these subject orders are based on the personal perspective of the subject, which can vary dramatically, however, subject two may not have fully understood the purpose of the experiment and ordered the strips in a different manner.
Table 8.1. Comparison of the clumpiness measure to each subject

<table>
<thead>
<tr>
<th>Strip</th>
<th>C</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>22</td>
<td>23</td>
<td>4</td>
<td>24</td>
<td>22</td>
<td>22</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>23</td>
<td>21</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>16</td>
<td>12</td>
<td>14</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td>16</td>
<td>7</td>
<td>16</td>
<td>16</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>C</td>
<td>16</td>
<td>20</td>
<td>17</td>
<td>22</td>
<td>12</td>
<td>18</td>
<td>21</td>
<td>22</td>
<td>22</td>
<td>24</td>
<td>16</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>16</td>
<td>14</td>
<td>5</td>
<td>15</td>
<td>17</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>17</td>
<td>12</td>
<td>11</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>E</td>
<td>21</td>
<td>22</td>
<td>9</td>
<td>21</td>
<td>23</td>
<td>23</td>
<td>17</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>F</td>
<td>22</td>
<td>24</td>
<td>18</td>
<td>23</td>
<td>24</td>
<td>24</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>10</td>
<td>19</td>
<td>24</td>
<td>20</td>
<td>21</td>
<td>20</td>
<td>22</td>
<td>19</td>
<td>22</td>
<td>21</td>
<td>23</td>
<td>22</td>
<td>18</td>
</tr>
<tr>
<td>H</td>
<td>5</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>15</td>
<td>14</td>
<td>12</td>
<td>14</td>
<td>6</td>
<td>11</td>
<td>8</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>I</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>13</td>
<td>12</td>
<td>15</td>
<td>11</td>
<td>9</td>
<td>13</td>
<td>10</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>J</td>
<td>12</td>
<td>11</td>
<td>11</td>
<td>17</td>
<td>20</td>
<td>13</td>
<td>10</td>
<td>13</td>
<td>19</td>
<td>18</td>
<td>12</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>K</td>
<td>7</td>
<td>7</td>
<td>15</td>
<td>10</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>12</td>
<td>8</td>
<td>5</td>
<td>13</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>L</td>
<td>19</td>
<td>15</td>
<td>2</td>
<td>18</td>
<td>14</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>13</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td>M</td>
<td>5</td>
<td>3</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>N</td>
<td>9</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>12</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>O</td>
<td>10</td>
<td>17</td>
<td>16</td>
<td>16</td>
<td>18</td>
<td>16</td>
<td>11</td>
<td>10</td>
<td>18</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>P</td>
<td>4</td>
<td>4</td>
<td>19</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Q</td>
<td>12</td>
<td>8</td>
<td>23</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>14</td>
<td>6</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>R</td>
<td>3</td>
<td>12</td>
<td>21</td>
<td>4</td>
<td>8</td>
<td>11</td>
<td>13</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>S</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>T</td>
<td>14</td>
<td>13</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>14</td>
<td>15</td>
<td>10</td>
<td>8</td>
<td>17</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>U</td>
<td>7</td>
<td>6</td>
<td>22</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>15</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>V</td>
<td>16</td>
<td>18</td>
<td>8</td>
<td>13</td>
<td>16</td>
<td>19</td>
<td>19</td>
<td>17</td>
<td>11</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>W</td>
<td>22</td>
<td>21</td>
<td>14</td>
<td>19</td>
<td>19</td>
<td>21</td>
<td>20</td>
<td>20</td>
<td>14</td>
<td>20</td>
<td>19</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Spearman's ρ: 0.85 -0.29 0.85 0.71 0.78 0.79 0.81 0.68 0.83 0.75 0.85 0.85

As with the original strips of Chapter 7, a consensus order was generated from the 10 subject orders. However, in this case, it was not possible to assign a single rank value to each of the strip letters, their positions in each of the subject orders varying too greatly. Instead, a consensus range was assigned to each strip letter within which 60%5 or more of the subjects placed that particular strip with a desired range of four or less positions. For example, strip A has a range of 22 to 24 inclusive as eight of the ten subjects, 80%, placed that strip in one of those three order positions. However, in some cases the range of positions needed to be extended since four positions was insufficient to cover the required 60%, specifically strips D, H, J, O, R, R.

---

5 A value less than 60% prevented a consensus range being created for the majority of the strips.
T and V, in which case the smallest possible range was used to cover at least 70%\(^6\) of the subject positioning. The first three columns of Table 8.2 show the strip letter plus the required range, the fourth column is the order as defined by the clumpiness measure and the fifth whether the clumpiness measure order position is within the consensus range, true if it is, false if not. In Spearman’s rank correlation coefficient, each value is subtracted from a mean, however, here there is no single value from which to subtract since the consensus is formed in a range. Instead, using \(d\) as the distance from the range, each clumpiness measure value will either have a \(d\) of 0 if it is within the range or a value representing its distance from the range. For example, for strip B the clumpiness measure ranking is outwith the range of 14 to 16 by four positions, therefore its \(d\) is four. Each of these \(d_s\) can then be used in equation 7.2 (Section 7.1, page 100).

In Table 8.2, more of the clumpiness measure rankings are outwith the consensus range than in them, 54\% as opposed to 46\%. However, when examining the distance from the range it can be seen that more than half of those outwith ranges are only just outside with a distance of one, therefore, resulting in the extremely good correlation of 0.94. With more subjects it should be possible to streamline the consensus ranges, reducing them far enough for a single value per strip rather than a range.

---

\(^6\) A more stringent percentage was used since the number of positions in the range was more flexible.
Table 8.2. Comparison of consensus ranges to the clumpiness measure order

<table>
<thead>
<tr>
<th>Strip</th>
<th>Lower</th>
<th>Upper</th>
<th>C</th>
<th>In range</th>
<th>d</th>
<th>d^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>22</td>
<td>24</td>
<td>22</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>14</td>
<td>16</td>
<td>20</td>
<td>FALSE</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>20</td>
<td>22</td>
<td>16</td>
<td>FALSE</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>17</td>
<td>16</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>20</td>
<td>23</td>
<td>21</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>22</td>
<td>24</td>
<td>22</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>19</td>
<td>22</td>
<td>10</td>
<td>FALSE</td>
<td>9</td>
<td>81</td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td>14</td>
<td>5</td>
<td>FALSE</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>I</td>
<td>10</td>
<td>13</td>
<td>14</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>11</td>
<td>19</td>
<td>12</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>M</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>FALSE</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>N</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>FALSE</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>O</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q</td>
<td>8</td>
<td>11</td>
<td>12</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>7</td>
<td>15</td>
<td>14</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>U</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V</td>
<td>15</td>
<td>19</td>
<td>16</td>
<td>TRUE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>19</td>
<td>21</td>
<td>22</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>FALSE</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

| TRUE | 11  |
| FALSE| 13  |

sum 137 spearman 0.94

In addition to ordering the strips, the subjects were asked to describe their methodology for placing the strips in that order and also if any strips were difficult to place. In the former case the most common methods involved counting the number of green blocks in a cluster or white blocks in a gap or counting the frequencies of the different clump sizes. The larger the counts of blocks or larger clump sizes would indicate a greater degree of clumpiness. Other methods included the number of groups of green blocks, the largest and smallest clump size or the number of single instances of green blocks.

For the question of the most difficult to place strips, the most often cited were strips O and A. Also, it may be interesting to note that all but one of the strips with a wide range for the consensus were also difficult to place. The most common reason for
the difficulty was that the particular strip in question seemed very similar to neighbouring strips in the order.

In the second part of this evaluation, subjects were asked to assess the order into which the clumpiness measure had placed 120 strips. On the whole, the subjects felt that the order given by the clumpiness measure was a fair representation of a progression from clumpy to unclumpy. However, there were a number of individual strips that the subjects felt should have been placed at a different order position. Again, most commonly cited was

- the reversing of the first and second position strips
- a variation to the order of the sixth, seventh and eighth strips
- placing strips 13 and 25 higher up in the order

Therefore, from the studies to date, it would appear that the CUSUM measure captures the notion of clumpiness within the distribution of events and non-events in a one-dimensional field. Further work in this area would be advantageous and will be discussed in the next chapter.
9 BENCHMARK TESTING
The previous Chapter discussed the utility of a clumpiness measure when used in actual research situations with the view of how well the measure
a. Captured the notion of clumpiness, reordering the BLAST result file so as those alignments with higher clustering in the alignments were placed higher up than those without
b. Assisted in the function determination of a novel protein sequence, in that the sequences raised up the list are of actual biological relevance.

The first of these was explored through the feedback interviews with the experts, leading to the alteration of the standardisation of the clumpiness measure described in Section 8.1.2, page 130.

During the course of the feedback, one of the subjects noted that one of the migrated alignments appeared to be a distant relative of the query sequence, Section 8.1.2, page 128. In this particular instance the subject knew what they were looking for and, therefore, recognised the relevance of that particular alignment, showing the potential of the clumpiness measure to highlight distant yet biologically relevant sequences. However, the clumpiness measure’s ability to identify the distant relatives of a protein family requires more stringent investigation and this is the subject of this chapter.

9.1 Methodology
This investigation will involve conducting BLAST searches with a protein family with distant relatives and assessing the movement of these distant relatives within the list of BLAST results after the clumpiness measure has been applied. However, in order for an effective evaluation of the measure the dataset, containing the chosen family and database sequences, used will have to have a certain composition.

9.1.1 Composition of dataset
If either the chosen protein family, or subset, is too closely related or the other sequences in the database, or otherset, are not similar enough to the subset then there would be no mixture of subset and otherset sequences in the BLAST result and, therefore, nothing for the distant relatives to be raised over. In this situation, a scientist could be highly confident in the function assignment of the top hits and
would be unlikely to require any additional assessment. However, since this is not the desired situation for the usage of the clumpiness measure the structure of the subset and the BLAST results are particularly important and will be discussed in sections 9.1.1.1 and 9.1.1.2 respectively.

9.1.1.1 Structure of Subset

As mentioned the subset would consist of a family of proteins that includes distant relatives, specifically of a 'comet' like form. That is, a 'head' of closely related sequences with a 'tail' of more and more distantly related relatives, Figure 9.1. The query sequence used in the BLAST search would be taken from the head and the positions of the distant relatives could then be compared before and after the clumpiness measure was applied. The distant relative, as well as some closer relatives, that are of most interest are those that are 'twilight' zone BLAST hits, referred to in the literature as sequences of 25-30% identity, (Fetrow and Skolnick 1998, Pible et al. 2005, Rost 1999). Therefore, as well as inspecting the distant relatives as defined by phylogeny, those hits that have twilight similarity will also be assessed.

![Figure 9.1. Schematic representation of a comet to illustrate the form the dataset needs to take](image)

Since the hits to 'head' sequences are not as important as hits to the distant relatives, the head of the subset need not be very large. However, consideration needs to be taken in the overall size of the subset, too small and there will be insufficient data on which to draw meaningful conclusions; too large and there is the risk of the subset swamping the BLAST results list so no otherset sequences are visible. With this in mind a subset size being between 10 and 25% of the otherset was deemed suitable.
9.1.1.2 Structure of BLAST results

As well as the subset being structured in a certain way, it is also necessary for the BLAST result list to have a certain appearance. As has already been mentioned if all the subset sequences appear above the otherset sequences there will be no non-related sequences over which to raise distant relatives and the clumpiness measure would be of no use. Therefore, in order to effectively assess the clumpiness measure's capabilities any BLAST result file used must adhere to a particular form.

Of vital importance in this form is that there is a 'smudge' zone where subset and otherset sequences are intermingled, Figure 9.2. Since one of the subset sequences will be used as the query sequence in the BLAST search there may be a subset only zone above the smudge zone. Equally, as similarity decreases, an otherset zone of otherset only sequences may be seen below the smudge zone. However, the most important factor is that the 'smudge' zone shows a good mix with sub and otherset sequences.

**Figure 9.2.** Schematic representation for the desired appearance of the BLAST result hit list

This smudge zone is dependent on there being sufficient similarity between the otherset sequences and the subset sequences so as the BLAST program will view
some otherset sequences are more similar than the more distantly related members of the subset. This would mimic the situations in true BLAST searches where there can be ambiguity in the hits returned where there are several possibilities as to the function or family and it is in these situations that the clumpiness measure is more likely to be of use. However, the clumpiness measure is unlikely to have any significant effect on the hits in the higher regions of the BLAST result file simply because the higher levels of similarity would mask any clustering that may be present in the alignment.

Therefore, in summary, the subset requires a protein family with a core of closely related sequences with additional more distantly related sequences and the otherset requires similarity overlap with the subset family to produce a 'smudge' zone in a BLAST result list. Candidates for this dataset will be explored in Section 9.2.

![Schematic representation of the desired dataset](image)

**Figure 9.3.** Schematic representation of the desired dataset

9.1.2 **Testing methodology**

As mentioned, the purpose here is to test the clumpiness measure's ability to identify distant relatives of a protein family and promote them up a BLAST results list in order to come to the attention of scientists sooner. The basic approach is in three steps:

1. Take the top sequence from the phylogenetic tree created from one of the datasets described in Section 9.2, as the query sequence in an in-house BLAST similarity search
2. Apply the clumpiness measure to the BLAST result list
3. Compare the sequence positions of the other family members when the results list is ordered by:
   a. The BLAST score
   b. The CUSUM measure

The first step conducted an in-house BLAST search using the stand-alone suite of BLAST programs downloaded from the NCBI website (ftp://ftp.ncbi.nih.gov/blast/executables, lastest folder). Given a text file of sequences in FASTA format, a database could be generated using the formatdb program (NCBI 2005d) and then, using a query sequence in another text file, a BLAST search was conducted using the blastall program (NCBI 2005b).

The second step took each of the BLAST result alignments in turn and assigned the two clumpiness values as calculated using the CUSUM and HMM measures described in Sections 6.3.2 and 6.2.3.2 respectively. For the final step a table was drawn up, as in Figure 9.4 so that the positions of the testing dataset sequences could be compared. The list of BLAST results was reordered as per each of the table headers and the position number of each of the subset sequences was noted.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>BLAST position</th>
<th>Score position</th>
<th>EValue position</th>
<th>CUSUM position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>56</td>
<td>56</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>369</td>
<td>369</td>
<td>369</td>
<td>175</td>
</tr>
</tbody>
</table>

**Figure 9.4.** Example table for the comparison of order positions

Given the subset structure, Section 9.1.1.1, the BLAST result list structure, Section 9.1.1.2, and the testing methodology to follow, Section 9.1.2, the next step is to identify a candidate dataset and conduct the clumpiness measure testing.

### 9.2 Candidates for Dataset

With the desired structure for subset and BLAST result list candidate, complete datasets can be assessed for suitability. In each case a particular protein family was selected for the subset and phylogenetic analysis was used to assess the relatedness of the family members. This allowed for the selection of one of the closely related sequences as the query sequence for the BLAST search. The suite of programs called
Phylip (Felsenstein 2007) was used to generate a phylogenetic tree from a Multiple Sequence Alignment, MSA, of the family sequences in the following manner:

1. The MSA was given to the ProtDist program to generate a matrix of all-against-all protein sequence distances and saved to file
2. This file was taken to the Neighbor program to define the branching within the tree
3. The program Drawgram was used to generate the graphical diagram of the tree from the data provided by the Neighbor program
4. Finally the program Retree was used to adjust the tree so that the densest branchings and, therefore, most closely related sequences are shown at the top of the diagram

It should be noted that branches within a phylogenetic tree can freely pivot about their node with either representation being just as valid. The use of the Retree program was merely for the purposes of clarity.

In a phylogenetic tree, the flatter the branches are the more closely related sequences, or groups of sequence, are, Figure 9.5.a. Conversely, the more distant sequences or groups of sequences are the longer the branches are, Figure 9.5.b.

Figure 9.5. Subsections of a phylogenetic tree to illustrate the difference in degrees of distance with a. closely related sequences and b. distantly related sequences
With the tree created a query sequence could be selected. This was simply the top most sequence in the tree, for example hsUBE2D2 in Figure 9.5.a, though hsUBE2D3 would have been equally as valid.

9.2.1 Ubiquitin Conjugating Enzyme
The first of the prospective datasets was the ubiquitin conjugating enzyme, UBE2, family mentioned in Section 8.1.2, page 128, which gave the initial good indication of the utility of the clumpiness measure. As mentioned before, the UBE2 family catalyse the attachment of the Ubiquitin enzyme to its target. The Ubiquitin enzymes are involved in protein degradation. An Ubiquitin activating enzyme, E1, is activated by an energy dependent reaction and this in turn activates a complex of Ubiquitin conjugating enzyme, E2 and another enzyme, E3, to form Ubiquitin ligase, Figure 9.6.

Figure 9.6. Process of Ubiquitin ligase complex activation, (Alberts et al. 2002)

Phylogenetic trees were generated from a MSA donated by the subject in Section 8.1.2 (Mukherjee 2006) for the Homo sapiens and Arabidopsis thaliana sequences in the MSA. These two species were chosen as they gave the desired minimum of 30 sequences and their phylogenetic trees revealed that as well as close relatives there were also a good selection of distant relatives giving the desired comet like structure, Figure 9.7.
For the otherset the E3 part of the Ubiquitin ligase complex, Figure 9.6, was suggested as a possible close enough relative to form the desired smudge zone in the BLAST result. However, it was discovered that the desired BLAST result structure could not be obtained. Several BLAST searches were conducted using the protein sequences at the top of the phylogenetic tree, the most closely related, labelled '1' in Figure 9.7. The resulting BLAST results list had no overlap between the subset and otherset sequences and, therefore, was of no use for the clumpiness measure assessment. In addition a BLAST search was conducted using a more distant relative, labelled '2' in Figure 9.7, to see if any overlap could be obtained but again the two sets were distinct.
9.2.2 Transforming Growth Factor - Beta

The Transforming Growth Factor β, TGF-β, are cell signalling polypeptides involved in a wide range of functions that “regulate embryo development and tissue homeostasis”, (Herpin et al. 2004). A TGF-β ligand binds to a phosphorylated type II receptor, which then forms a heterotetramer with a type I receptor. The type I receptor is then transphosphorylated by the type II receptor and this complex, in turn, phosphorylates a gene regulatory protein of the Smad family, which produces the cellular response, Figure 9.8.

Figure 9.8. Process of cell signalling by the TGF-β, adapted from Alberts et al (2002)

The subset was generated using the 29 sequences labelled ‘TGFB’ from the Prosite pattern PS00250 and the initial otherset contained all the other sequences from the Prosite entry. Unfortunately, this first otherset was unsuitable for the creation of the desired BLAST result list structure and no alternative otherset could be identified for use in the in-house BLAST program. Equally, running an online BLAST against the Swissprot database was unsuccessful. However, against NCBI’s Genbank non-redundant database a fair representation of the desired BLAST result list structure was obtained. Although the majority of the BLAST hits to TGF-β were to be found at the top of the BLAST result list, there were also a number scattered through the rest of the BLAST hits, the majority of these being other members of the TGF-β superfamily as were found in Prosite entry PS00250. The sequences in the BLAST result list were classified as subset or otherset based on their description line. If the description identified the sequences as TGF-β then it was classified as subset, otherwise it was classified as otherset. Unfortunately, because of the nature of the non-redundant database there were also unclassified sequences in the BLAST result
list, which could have been TGF-β sequences or any other similar protein. In order that these sequences did not affect the analysis of the known sequences, a third grouping was created, that of unclassified sequences.

9.2.3 Fibronectin

Fibronectin is an “extracellular matrix protein that is involved in adhesion of cells to the matrix”, (Alberts et al. 2002), via interactions with the integrin receptor. In addition a soluble form circulates in the blood and is involved in wound healing. The interesting aspect, from the perspective of this research, is its multiple domain structure. The fibronectin molecule consists of two subunits, linked by a disulfide bridge, each containing approximately thirty domain modules, or motifs, one subunit is shown at the top of Figure 9.9. Through exon duplication, motifs can appear in other protein families as illustrated in Figure 9.9. As can be seen, fibronectin contains twelve finger modules, or type I domains, which also appear in the tissue plasminogen activator, TPA, and prourokinase, PK. These latter two proteins also contain kringle modules, which are absent from fibronectin.

Figure 9.9. Modular characteristics of protein domains, (Strachen and Read 1999)

The purpose of the dumpiness measure is to identify those sequences that are more likely to contain similar protein domains or motifs and, thereby, identify distant relatives of a particular protein family. It was felt that using a protein family that contained a motif that appeared in other protein families may result in the desired BLAST result list. To this end the TPA and PK families were combined to form the subset with fibronectin sequences forming the otherset. To create the subset the NCBI protein sequence database was searched for TPA and PK, resulting in a list of 282 sequences. This list was then assessed for suitability, only keeping those sequences that:
- Contained the type I domain
- Were actually a TPA or PK sequence and not an inhibitor, just referenced TPA or PK or a precursor
- Were not too short, to allow for similarity to areas other than just the type I domain

This gave a subset of 26 sequences, 24 TPAs and two PKs, with the phylogenetic tree showing, as expected, the two PKs being the most distant sequences. The otherset was generated in a similar manner, searching the NCBI protein database for fibronectin and ensuring that all the sequences selected contained the type I domain and were not TPA or PK. This gave 182 sequences for the otherset, unfortunately on conducting a BLAST search it was discovered that there was insufficient overlap to produce the desired smudge zone in the BLAST result.

An alternative otherset was generated using sequences containing the kringle motif taken from Prosite entry PS50070, again, ensuring that it did not contain any TPA or PK sequences. However, there was still insufficient overlap to create the desired BLAST results list.

Attention was then directed at the type III domain, labelled ‘Module shared with cell surface receptors and other extracellular matrix proteins’ in Figure 9.9. The interleukin sequences from Prosite entry PS50853 were used as the subset and the rest of the entry’s sequences were used as the otherset. Interleukins are a class of cytokines involved in various aspects of the immune response, though the exact action is dependent on the specific interleukin. The interleukin sequences were used in preference to any other member of the type III domain entry as they had sequences in both the true positive list and the false negative list, indicating that there were some sequences not found by the Prosite pattern. This was of interest to see how the clumpiness measure would assess these sequences. The resulting BLAST result list had a good spread of interleukin sequences through the otherset sequences.
9.2.4 SNF2

The SNF2 family of proteins was used as an illustrative family in the Blocks+ paper, (Henikoff et al. 1999), where it is defined as a “family of mostly DNA-stimulated ATPases” performing the “central catalytic role in eukaryotic chromatin remodelling complexes” and in the basal transcription regulation. Henikoff, Henikoff and Pietrokovski also mention that this family shares similarity with several other families making it difficult to distinguish by similarity alone. The commonality between these families is a series of helicase motifs, which, in some families, unwinds nucleic acid duplexes, though has not been found to be active in the SNF2 family. This would appear to be exactly the situation where the dumpiness measure could highlight the SNF2 sequences above the other helicase containing families.

Therefore, a number of SNF2 sequences were used as the subset and members of other helicase containing families were used as the otherset. The specific SNF2 sequences were taken from Jonathan Eisen’s website (http://www.tigr.org/~jeisen/SNF2/snf2.html) using 22 of the 28 sequences from the detailed family sequences, (Eisen et al. 1995), six of the sequence’s no longer being maintained. From the phylogenetic tree sequence SNF2L, Swissprot accession P28370, was taken as the query sequence. For the otherset Swissprot was searched for other helicase containing families, which did not contain any of the SNF2 sequences. Of the eighteen families found, Prosite entry PDOC51293 gave the desired BLAST results list structure, however, it should be noted that although the otherset contained 113 sequences only 21 were identified by BLAST.

This gave three useful datasets the TGF-β against the non-redundant database, interleukin against fibronectin, and SNF2 against another helicase family. Of these, both the TGF-β and interleukin othersets contain hits to members of the same superfam­ily. Therefore, despite the desired BLAST result list structure, there is a chance that any movements seen in the list orders may be due to rearrangements of the family rather than promotion of specific members of the subfamily.

---

7 An ATPase is an enzyme that catalyzes a process involving the hydrolysis of ATP
9.3 Results and Discussion

Given the three suitable datasets the clumpiness analysis was conducted as described in Section 9.1.2.

9.3.1 Positional Movement Analysis

There is currently no ‘quick-fix’ method for the analysis of movements within rank orders, (Staines 2007). However, some form of significance testing is desired in order to assess the possibility that the movement induced by the clumpiness measure occurred by chance. In a similar manner the E-value assesses the similarity score in terms of the possibility of obtaining that score by chance. Often when comparing two rank orders a rank correlation is used. However, this method assesses the similarity of the two rank orders are, with no assessment as to the movement of individual elements in the rank. Here, it is the degree of movement of specific elements that is of interest, therefore, the movements of these specific elements will be assessed against non-specific elements. That is, the movements of the subset sequences will be compared against the movements of the otherset sequences.

In order to do this, the position table of Figure 9.4 was separated into subset sequences and otherset sequences. An Anderson-Darling normality test was then conducted and it was found that all the datasets were not normally distributed. This being the case, a Mann-Whitney test was conducted to test for differences between the two groups. The Mann-Whitney test assesses whether two independent groups of observations are significantly different, though in this situation there is no guarantee that the subset and otherset will be completely independent, especially in those datasets where the otherset is taken from the same super-family. Nevertheless, the data was transferred into the Minitab statistical package, (Minitab Inc. 2007), and a one-sided Mann-Whitney test conducted. For this, the null hypothesis was that there would be no significant difference between the two sets and the alternative that the subset differences are greater than the otherset differences.

However, it was found that there was no significant difference with any of the datasets. When elements are moved around a list, in order for one element to be raised it is necessary for another to be lowered. As a result, the lack of significance
in the Mann-Whitney tests may have been due to the movements down the list 
masking any positive movements. Therefore, an alternative assessment was adopted.

In this alternative assessment, only the positive movements were used, recording the 
following for both sub and otherset sequences:

- Maximum positive movement
- Number of positive movements
- Percentage of positive movements of sequences within set
- Percentage of positive movements of sequences in complete dataset
- Number of large positive movement, greater than one hundred, fifty or ten 
dependent on the size of the list
- Percentage of large positive movements of the positive movements within set
- Percentage of large positive movements of sequences within set

The results for these values can be seen in Table 9.1, where s is the subset values, o 
is the otherset values and, in the case of TGF-β, u is the unclassified sequence 
values. The bracketed values in the TGF-β data correspond to the percentages 
calculated using the number of subset and otherset only.

<table>
<thead>
<tr>
<th></th>
<th>TGF-β</th>
<th>Fibronectin</th>
<th>SNF2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s</td>
<td>o</td>
<td>u</td>
</tr>
<tr>
<td>Max +ve</td>
<td>526</td>
<td>978</td>
<td>607</td>
</tr>
<tr>
<td>No. +ve</td>
<td>73</td>
<td>482</td>
<td>55</td>
</tr>
<tr>
<td>No. in set</td>
<td>119</td>
<td>961</td>
<td>108</td>
</tr>
<tr>
<td>% +ve of set</td>
<td>36.7</td>
<td>50.2</td>
<td>50.9</td>
</tr>
<tr>
<td>% +ve of dataset</td>
<td>5.8</td>
<td>38.0</td>
<td>4.3</td>
</tr>
<tr>
<td>(6.3)</td>
<td>(41.6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. large +ve</td>
<td>13</td>
<td>359</td>
<td>35</td>
</tr>
<tr>
<td>% large of +ve</td>
<td>17.8</td>
<td>74.5</td>
<td>63.6</td>
</tr>
<tr>
<td>% large of set</td>
<td>6.5</td>
<td>37.4</td>
<td>32.4</td>
</tr>
<tr>
<td>%large of dataset</td>
<td>1.0</td>
<td>28.3</td>
<td>2.8</td>
</tr>
<tr>
<td>(1.1)</td>
<td>(30.9)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Both the TGF-β and Fibronectin datasets showed a greater degree of positive 
movement in the otherset as opposed to the subset. This may be due to the fact that 
both of these datasets used sequences from the same Prosite entry to create both the
subset and otherset. Consequentially, the two sets may have been too relationally close, resulting in the clumpiness measure performing interfamily shuffling rather than distant relative identification. In contrast, the SNF2 dataset showed the greater degree of positive movement in the subset as would be desired if the clumpiness measures were promoting members of the same family. Furthermore, the majority of the positive movements were of a greater value than those of negative movements, the exceptions being to secondary hits to a couple of the sequences. These secondary hits occur when BLAST identifies more than one distinct similar regions between the query and database sequences.

In addition, a visual assessment of the data was undertaken, looking at where subset sequences were moving to and if any of the distant relatives or sequences with twilight similarity were being moved up the results list.

9.3.2 Visual Assessment

The visual assessment was a more abstract analysis of the positional movements within the BLAST result list before and after the clumpiness measure had been applied. Focusing on the:

- overall distribution of the subset sequences
- distant relative, as defined by the phylogenetic trees
- twilight zone sequences based on their percent identity, below 30% or below 25% as extreme

As with the positional movement analysis in the previous Section, the TGF-β and fibronectin datasets did not give very good results. In both cases the distribution of the subset sequences through the BLAST result list became more scattered and lower down after the clumpiness assessment. Again, this may have been due to the use of members of the same super-family for both subset and otherset sequences. For the SNF2 dataset there was virtually no change at the top of the results list and a small amount of shuffling in the mid-section. However, the more interesting aspect is in the lower section of the list where the lowest subset sequences have been promoted, Figure 9.10, the bottom most have the greatest positive movement in this particular dataset.
Figure 9.10. Representation of the change in the distribution subset sequences, grey, between a. the BLAST score order and b. the Clumpiness score order

With the distant relatives, for both the fibronectin and SNF2 datasets there was an equal occurrence of positive and negative movements. For the TGF-β dataset, with a phylogenetic tree rooted to the query sequences, the majority of the small positive movements occurred in the more distant relatives and the negative movements were also small in comparison to the negative movements seen elsewhere in the dataset. It should be noted that the larger negative movements in both the TGF-β and fibronectin datasets all appeared to correspond to short alignments with high degrees of identity, indicating there may be an issue with the standardisation technique used on the clumpiness score.

Finally, the investigation of those sequences that had only twilight similarity gave more interesting results. Although the fibronectin dataset had equally positive and negative movements in these sequences, for both the TGF-β and SNF2 datasets, the majority of the larger positive movements fell within the twilight zone sequences. Since the main purpose of the clumpiness measure is to identify twilight zone sequences that may be of biological relevance to the query sequence this result is a good indication of the usefulness of the measure. However, as will be discussed in Section 10.1, additional analysis will be necessary to further validate the method.
10 CONCLUSION

Considerable research has been conducted into computer-assisted identification of protein sequence function. These include gene prediction, domain and motif searches, structural prediction and searches with novel protein sequences against online sequence databases, such as in the BLAST programs. The purpose of this research was to develop an algorithm for the assistance of the analysis of the results returned by the BLAST program. During the course of this development, the criteria used for selecting relevant BLAST hits (relevant to scientist's needs) were investigated. This highlighted various aspects of the BLAST results, such as the score, the e-value, the presence of keywords and the analysis of the alignments, which would be utilised by the scientists in their decision-making process. Most of these aspects have already been researched resulting in many useful approaches; however, in the analysis of the alignments, the solutions, such as motif searches or multiple alignments, proposed may miss relevant but distantly related 'twilight zone' sequences or produce invalid conclusions resulting from inaccurately constructed multiple alignments, Section 4.3 (page 52).

Therefore, this research centred on the development of an algorithm that considers each of the alignments in isolation, returning what has been termed a clumpiness value. This could indicate the possibility of the presence of functionally important motifs or active sites within an alignment without the need for constructing a multiple alignment or an additional search to the motif database, which might not even find all possible motifs. However, there is a circular problem (Section 5.3, page 60) in that a model differentiating levels of clumpiness is needed to generate the measure but the measure is needed to accurately create the model. In an attempt to break this circle, an estimation of the model was created based on the differentiation of clumpiness levels by subjects who ordered a simulated dataset from most to least clumpy. This simulated dataset took the form of a synthetically created dataset that maintained all influences, such as length and number of matches, except the distribution of those matches along the length of each of the alignments, or lines.

A number of potential methods for the quantification of clumpiness were proposed and these were then tested with regard to criteria defined during the initial investigation period. The first of these is their ability to validly order an artificially
generated dataset by their degree of clumpiness as compared to the same set ordered manually by a number of subjects, who themselves use BLAST. Secondly is the ability of the measure to tolerate the possibility of point mutations resulting in single gaps in the alignments. The third criteria is concerned with the technical details of the time taken for the measure to run. This resulted in the selection of the cumulative sum, or CUSUM measure, a quality control method that detects whether a process has deviated from specified criteria. This was then incorporated into a limited prototype to further test the measures abilities within actual genetic research involving the use of BLAST searches.

Although the overall results from the prototype testing were good with the rearrangement of the BLAST hits, in one case, possibly highlighting a distant relative of the protein under study, it was felt that it was lacking in two main areas. Firstly, the program became exceptionally slow when larger sequences or numbers of results required processing or on older computers. A possible solution would be to store the program on a fast server and providing a web service via the Internet using client-server technologies, as is already done with many bioinformatics programs, including the BLAST algorithm itself. The other issue was the severe lack of subjects willing to test the prototype, therefore, of highest priority of any further work would be the accumulation of more agreeable test subjects and from there the further evaluation of the clumpiness measure to assist in the determination of protein function.

In conjunction with the prototype testing, the CUSUM clumpiness measure was further tested with regards to its capturing of the concept of the distribution of events and non-events within a one-dimensional field. In this case the selection of subjects was more flexible allowing for the possibility of a larger number of subjects, however, at this time there were still only 12 volunteers. However, the results are very good, showing that the CUSUM clumpiness measure gives a good approximation of the degree of clustering of events within a one-dimensional field.

The prototype and abstract testing gave a indication of the clumpiness measure's ability to capture the notion of clumpiness, further testing as to its ability to identify the distantly related or twilight zone sequences there were of biological relevance to
the query sequence. To this end an attempt was made to identify a suitable dataset that would create a BLAST result list that contained a protein family or subfamily with distant relatives intermixed with other sequences similar enough to be identified by BLAST but, hopefully, distant enough not to be considered of the same family. Though the identification of a protein family was a fairly simple task, identifying additional sequences with enough similarity to appear in the BLAST result list was a non-trivial undertaking. However, three suitable BLAST results lists were created and reassessed using the clumpiness measure. There not being any current method by which to analyse the changes in a rank order, the assessment of the clumpiness results was mainly visual. That said, there was some indication that the clumpiness measure had promoted twilight zone sequences of biological relevance up the BLAST result list. On the other hand, in two of the datasets there was also considerable promotion of sequences occurring in the non-family sequences, though this may have been due to the nature of these sequences and being relationally too close to the protein family resulting in movements within the super-family rather than the desired analysis of the subfamily. In addition, the examination of the distant relatives, as defined by phylogenetic analysis, proved inconclusive and on examination of those family members that were demoted down the list a possible issue with the standardisation of the clumpiness score arose.

10.1 Further work

As mentioned above, one of the first priorities for further research in this area is the addition of further test subjects for both the evaluation of the prototype and verification that the concept of the quantification of clumpiness has been adequately captured. This should give confirmation that the clumpiness measure is accurately capturing the notion of clumpiness in the one-dimensional field. In addition further assessment of the clumpiness measures's ability to assist in identifying distant but still relevant hits is required. Firstly, the standardisation technique used to render the clumpiness scores comparable requires re-evaluation. This will ensure that hits are promoted or demoted solely on the basis of its clumpiness and not for its length, degree of identity or the coverage of the matching hit on the query sequence, though these factors may form part of the standardisation process. Secondly, additional distant relative datasets giving the desired BLAST result list, as outlined in Section 9.1.1.2, need to be formulated, which could then be used for the further testing of the
clumpiness measure. One possible solution would be to use randomised protein sequences in the otherset rather than actual sequences from other protein families, either generated completely at random or randomising the amino acids present in the subset sequences. Third and finally, research is required into possible methods for statistically analysing movements within a rank order. This could then be used to statistically analyse the results from applying the clumpiness measure to the distant relative datasets.

In addition, it has been noted briefly in Section 3.2 on page 35, that the PSI-BLAST program is more sensitive than BLAST. However, since the result file produced by PSI-BLAST is identical to the BLAST result file it would be possible to conduct the clumpiness assessment on the PSI-BLAST results with no alteration to the current prototype program. Therefore, after the further testing of the clumpiness measure, it would be interesting to assess the clumpiness measures abilities when applied to the PSI-BLAST results instead.

Further work may also be conducted into extending the assistance in the BLAST analysis or in the statistical methodology used.

10.1.1 Assistance of BLAST Analysis

Within the field of BLAST analysis assistance, it would also be advantageous to incorporate the additional features as outlined in Figure 4.3 (page 47). This includes adapting the clumpiness measure for nucleotide sequences as well as the protein sequences, the latter being used in this research since the larger alphabet allowed clustering within the alignments to be more prominent. It may be that no adaptation is required with the CUSUM method working for both protein and nucleotide sequences in its current form. However, it may be that, since clustering of matches within a nucleotide sequence is more likely due to the smaller alphabet, a more sensitive measuring method may be required. This may simply entail the altering of the specifications of the CUSUM measure, namely the $h$ and $k$ values used in the calculations, Section 6.3.2. Alternatively, it may be necessary to use one of the other proposed measures described in Chapter 6 that may better capture the notion of clumpiness portrayed in the nucleotide sequence.
An attempt was made to include a keyword search in the prototype, which was only partially successful. Ideally all keywords associated with hits would be extracted from their source database and pubmed and clustered as to the different keywords found, such as in the Anagram program (Section 2.5.2, page 27). Other features outlined in Figure 4.3, may also be incorporated by using some of the many algorithms created in other research as outlined in Table 4.1 on page 54. The addition of the extra analysis should increase the knowledge base with which the scientist may extrapolate their decision as to the possible function of their sequence, as in the integrated systems described in Section 2.5. However, certain data sources may have greater or lesser influence on the decision-making process, therefore, weighting of the information returned, as in the MAGIC system (Section 2.5.2, page 27) may be required. This weighting could be rigidly assigned or, as different scientists will have different requirements, the assignment could have a default setting that may be changed as per a particular scientists needs.

However, with each additional data source, the amount of data to analyse will also increase and there is the risk that there is too much information to assimilate. Since the purpose of this research was to alleviate this issue within the BLAST results, care would be needed to display the results in an easily readable view. Specifically the disinterring of relevant results from within less relevant ones, bringing only those hits that are of interest to the notice of the scientist. The use of visual aids, as in the algorithms described in Section 2.5.1, can alleviate this by presenting graphical overviews of the data where clicking on specific features brings up further details. The idea is the generation of an integrated source that supplies the scientist with as full a detail as possible displayed in a user-friendly manner and without bombarding them with irrelevance.

10.1.2 Statistical Development of the Clumpiness Measure
As was mentioned in Section 5.3, there is currently no known model for the distribution of events and non-events within a one-dimensional field. In this research the creation of the clumpiness measure was based on an estimated model for grading the distribution of events along a one-dimensional line. Given that any additional validation of the clumpiness measure proves beneficial, it would be logical to complete the circle and formally describe this model of clumpiness.
The data used in this research is discrete, being finite and countable, however, other data samples may contain continuous data involving an infinite number of real values. For example a person's height, could be 1.6m, 1.7m or any possible value in between, with the number of decimal places dependent on how precisely that person's height is measured. The clumpiness measure described in this research may be used in instances where objects are either in specific locations, such as, in a line of people, the distribution of those who wear spectacles, or where values are classified into ranges, as in the distribution of people's heights in ranges of ten centimetres.

However, it would also be useful if the measure were applicable to continuous data as well by adapting the clumpiness measure described in this research. The clumpiness measure was adapted from the CUSUM quality control method (Section 6.3.2), which is generally used in time series sampling to ensure quality control in the specified system. With time also being a continuous data-type, it should be feasible to relate the discrete clumpiness measure to a continuous data sample. In the CUSUM measure samples are taken from the system at specified time intervals, thereby reducing the continuity of time into discrete snapshots of the system under investigation. Similarly, spatial sampling is used as a model of the overall population, however, is it possible to account for possible overlap of sample subjects, such as in dolphins in the sea, in the quantification of clumpiness using the measure described in this research?
BIBLIOGRAPHY


Lyon, G. 2004. Interview by the author. Scottish Crop Research Institute, Invergowrie. 10th April.


APPENDIX A - INTERVIEW TRANSCRIPTS


001.0 J – OK, so… I have a couple of BLAST results, two from BLASTn and two from BLASTx, from those huge files you sent us. If we start with that one, which is quite a large one.

004.5 D – OK these are, if my memory serves me correctly and it was some time ago, these are potato sequences.

006.0 J – I have no idea what it is that you actually sent us.

006.5 D – OK, I’m fairly sure they’re potato sequences… And I presume that this is also BLAST… you know, so this is … BLASTn

008.5 J – That is BLASTn

009.0 D – And it’s BLASTn against the non-redundant database.

009.5 J – There’s differences between the databases?

010.0 D – Oh yeah, OK. Um… the first thing is that there are a whole series of different databases. What we do typically is three different BLASTS by default. We will BLASTn against the non-redundant nucleotide database. We will BLASTx against the non-redundant protein database. Non-redundant databases are databases where they try to get a set of a much as possible that’s in there but they try to avoid duplication.

016.5 J – Why’s that?

017.0 D – Because there are some cases, for example, where an individual gene has been sequenced and then a larger fragment that contains that gene has been sequenced. So what the idea is to try and produce a more efficient database for searching. They will have played around with this so it isn’t possible that the same thing is in there twice.

019.5 J – Right

020.0 D – And we also do a BLASTn against DBEST, which is an EST database. That’s a database of cDNA sequences or (*) high throughput to the (*) cDNA sequences.

023.0 J – Right

023.5 D – OK. Um, it’s quite important that, because to some extent how we interpret these things will depend on not just any one of them in isolation but to some extent we will look at all the results together.

026.5 Right. So they… If we’re going to look at the BLAST results, the first thing we’ll obviously look for is to see whether there are hits that are in some part of the arbitrary view as significant. So if I look down this what I’m interested in is the identifiers. I’m also interested in the BLAST score and the E-value.

033.0 Now, what tends to happen is that we tend have a sort of view as to what is a significant either score or E-value. For the same database these two will be comparable but if you use different databases, these things will actually vary depending on what database you’re using but within the one database they’re relatively straightforward. The score gives you no idea really of the probability of… sorry … the E-value gives an idea of the probability of finding a hit that’s at least that good by chance in a database of that particular size. So what we tend to be looking for is things that have really small values, but it’s quite difficult to come up with a cut off because it will depend an awful lot on whether there are some really very very high values or whether the best you’ve got is something that is in a bit of a grey area, OK.

047.0 So, if I look at this then there are clearly a … there’s a cluster of things that are relatively high BLAST scores and low E-values. And there’s a whole series of things that come up that look like they may be very similar products here. So the first three of these and then you skip another one and there’s another one, which look like they’ve got a very similar descriptive… OK.

055.0 Now the first thing, how I interpret or how anyone interprets this will depend on what sort of approach you’re taking and you’ll find this if you talk to some of my colleagues. To some extent I’m looking at it from a Bioinformatics point of view, so to some extent I’m looking at it to try and get some form of identifier for a given sequence. Whereas a number of my colleagues will be looking very definitely for a certain subgroup of things. I mean, they have a wish list in terms of things the want to find. Whereas my wish list is essentially as broad…

062.0 So, if I look at that, if I look at the descriptors, there’s a whole cluster of them that are… that look like they might be related. But from the actual descriptor itself, it gives me nothing in the way of detail information as to what that actually is. In essence I might be able to make some guesses from reading this. It might be a translation initiation factor because there are a few hits that say that, but the best BLAST hits give you just a designation for the gene.
So we’re in a situation where I think these BLAST hits are sufficiently good that whatever this is is going to be quite homologous to it. So it’s probably quite a good hit. So what we would want to then do is to try and find out what this is. And the ways we would try and find out what this is is to actually follow the chain of inference. So we’d go to the database because what you’ve got to remember when you do a BLAST search against the database you’re actually not searching against all sorry the BLAST database that’s being searched doesn’t have the full database information. It just has a limited amount, which is basically the identifiers and the description line.

So what we’ve got here is the little bit from the description line (*** to the database identifier. So the next thing you would do you’d go try and go to the full database entry for these to see what these are. And it may well be the full database entry tells you what this is or you may have to look for a Medline entry or something like that, which you’d go along to give you more information in terms of what it is.

Now, if I look at that as well, the other factors of interest to us is there are also there’s a very good hit to Arabidopsis genomic DNA. And that’s also of interest to us for different reasons. It’s of interest to us because that may enable us to locate that on the Arabidopsis genomic sequence. Now that’s useful in a number of ways because with most of the projects that we have we’re interested in not only finding identifiers for the gene we’re also interested in trying where possible to anchor the organisms we’re interested in onto genomes that are likely to be fully sequenced. So Arabidopsis is already fully sequenced. So if we have a potato gene or a whole series of potato genes that we can identify the equivalent genes in Arabidopsis we may be able to as it were link up bits of the potato genome and bits of the Arabidopsis genome. So that may actually help even for maybe able to predict what’s in between the bits that we can anchor. So we can develop a comparative framework.

We have the same interest going in particular from Barley to other cereals, but in particular from Barley to Rice because it’s likely that the whole Rice genome will be sequenced. And so then again if we can anchor the Barley genome that we’re interested in onto the Rice genome then we may be able to make inferences about the bits of the Barley genome we don’t have any information about being able to identify what’s in that chromid position in Rice.

So from the BLAST hits we’re interested in try to find an identifier and also in terms of the individual gene but in particular hits into Arabidopsis in the case of dicotyledons plants such as potato or hits into Rice are of particular interest to us.

The other thing which is really important is when I was talking about this difference in terms of the what is the you know the threshold we take as significant. If you actually scan down this I mean the BLAST scores range from one hundred and sixty five at the top here down to a BLAST score of sort of fifty-two which is still giving you a relatively low probability of finding this result by chance. If you actually scan down in the lower reaches here we have an entity there which I suspect which says C. elegans mRNA for eIF-4A homologue. My impression of that and also there’s another Nacoshi tabacum one up here which is has a BLAST score of sixty-six. My impression of both of these again that gives you information that’s consistent with what that’s really likely to be.

Now if we have some very high significance BLAST hits like these ones at the top we can to some extent we can forget the lower ones down and we probably put a in this case we’d think our threshold of acceptance is really quite high. But there a many cases when we almost have a the best we can do is going to be something that is actually down here. If there’s not something that’s obviously been cloned and sequenced from something that’s closely related to potato So we may have to have to interpret the BLAST scores in the light of just what significant hits are. So, I mean we may say well this not a very strong hit but it’s the best information we have currently. So there’s a difference between something that’s obviously a dead ringer and something like this gives us an indication a suspicion in term of what this actually is.

The other thing that we would do and the other thing which is very relevant is that is to look at the alignments. Because BLAST will also give you an alignment of the sequence and the alignments can be very useful both in terms of nucleotide alignments but also in terms of protein alignments. Because they can for example if it’s if we’re doing a query with cDNA which we know that we’re doing a query with the coding region so we know that if we’re looking for homology we’re looking for homology to coding region. We also may know something about that particular cDNA sequence it may have been sequenced from the five prime end or three prime end. So we also may know where we are as it were or potentially we have some context of where we are in the gene.
And, if we then look at the alignment, looking for homology, then we’ve got to be careful that it’s consistent with what we know about what we’re doing (*). So, for example, if we do … if we have a sequence, in the case of our Barley sequences, they’re all five prime sequences, so we’re sequencing in from the amino end of the protein. If we then get a hit that’s to the carboxy terminus, the other end of the gene, that gives us some suspicion that OK, we may have a domain homology, but we don’t actually have a gene homology.

And also, if we look at the alignments, it may be useful to know whether the homology is spread all across the sequence or whether it’s very strongly localised.

Do you have the BLASTx for the same sequence?

I don’t think I’ve got the exact same sequence, I’ve just basically trundled down the list and pulled out…

The reason … again…

No that says one hundred and four

Let’s go through BLASTx

If we look at the BLASTx hits, then a lot of the … the basis of what we look at are very similar to what we look at with the BLASTn hit. The one big difference is that because of the difference in size of the alphabet, as it were, for protein sequence as opposed to nucleotide sequence, then to some extent the BLASTx hit gives us stronger information.

And also, if we’re doing a BLASTx we’re searching against the protein database so, with cDNAs, we expect it to be a protein. So to some extent that part of the logic’s already sorted out when we look at that.

If you look at the alignment that’s here for the first hit, the first hit is against a ribosomal protein and it’s a ribosomal protein from lycopersicon exculentum, which is tomato. And if we’re doing a hit with a potato sequence the first thing that’s very important to realise is that the potatoes and tomatoes are very closely related. So, again, a very strong hit with tomato is something we’d expect, we’d have lots of confidence in it.

If we look at the alignment, w see … just scanning it by eye, in the sense, for a reasonably long stretch of protein sequence and this is … for one hundred and twelve out of one hundred and twenty four amino acids have got a dead h it... a dead ringer. And even for things which … the remainder of the amino acids, a significant proportion of them are positives. And what it means by positives here, it means that they’re from a related group of amino acids. So they’re not an identical amino acid but they’re an amino acid that is similar in properties and almost certainly similar in terms of the coding goes with it as well.

So what we’ve got here is a chunk of protein that’s really very, very strongly homologous. And one thing that’s also rue is that the … when you look at these results, the diversity in sequence varies enormously between different genes. So some genes are very highly conserved and other genes are very weakly conserved. So that, to some extent, some sort of pre-knowledge of certain proteins helps tremendously. So, for example, one classic case would be histone proteins, which are very, very highly conserved. So a very strong hit to histone proteins is what we’d expect (*) across enormous taxonomic boundaries you’d probably find quite strong hits. If you’ve got a histone you’ll see it very probably.

Similar thing is also true about ribosomal protein, which is almost certainly what this is because there are lots of hits very strong hits to ribosomal proteins across wide range of different organisms. Well, there’re really quite strongly conserved.

But there are some proteins where the conservation is very, very much weaker, where perhaps a particular domain or series of domains or if it’s an enzyme reactive site is really quite strongly conserved. And lots of other parts of the protein may under very loose evolutionary constrain.

Now in that case, again, knowing something about the protein can help tremendously. So if you look at … again if you look at … in that sort of case, a relatively weak alignment where there’s a BLAST score and an e-value, which is comparatively (*) on a level with what you think is significant. If it’s spread very evenly across the protein the it’s difficult to have a lot of confidence in it but if there are one or two regions, which are very highly conserved you may have either just a conserved domain or you may have a conserved, for example active site in a protein that’s relatively conserved.

So what you’d want to know is any feature information that you can obtain from the database about that particular protein. Now, again, the feature information that’s available ranges from incredibly complete to non-existent. So, again, if it’s somebody who has a very strong interest in a particular group of proteins and knows a lot about them or basically wants to find out something about it you’d probably put a lot of effort into following up through the full database entry, through Medline, through searching the literature for anything to do with it, to find all
the structural features of the particular protein. Use that to say, look I’ve got very strong hits here so that means that’s really quite likely to be a good hit.

241.0 But it’s really quite difficult to do it purely from a BLAST output. The BLAST output would really be the first stage in terms of (*).

244.5 The other thing is that with the different BLAST outputs almost certainly what you’d want to do is to put them together. So if you have a BLASTx output and a BLASTn output both against the equivalent non-redundant databases, it’s particularly useful to actually put them together and you’d be able to see what’s there. So, clearly if both flag up exactly the same protein then you’ve got increased confidence in it.

252.5 Again, what you tend to find is that, if you get a hit, a BLASTx hit, this often gives you stronger because of the difference in the size of the vocabulary, give you more confidence.

255.5 One particular problem that you have is that when you’re doing this, in particular plant sequences, that it’s not uncommon to find items where the BLASTn hit ... a number of the strongest hits are against some anonymous piece of genomic sequence. So it might be against a Bac clone, for example, which is a chunk of Arribadopsis sequence of maybe a few K. So the problem you have then is how do you interpret that?

264.5 And, again, there are a number of ways you can interpret that, is that if you go to the full annotation for that particular Bac you may find that in the region where you’ve got the hit there is a known ... either a known or a predicted gene in that hit. But there are some of the BLAST ... sorry ... there are some of the chunks of genomic sequence are not fully annotated. So you may find that the best you get from that is a very strong hit in this hundred kb of genomic sequence that they haven’t got a clue what it is.

273.0 Now there are a number of things that you could do on that basis, it just depends on how much time you have available to you. If it was something you were potentially very interested in, for some reason the cDNA you were doing the search with, you wanted to find out exactly what it was.

277.0 One thing you could do is you could actually take a genomic sequence and run some of the gene prediction software on it yourself. So you could actually do some of the predictions that some of the larger groups are doing on the genome (*).

283.0 The other thing that you could do is that even if these are the best hits, to some extent, you could ignore them and look a little bit further down the BLAST output to see if there are still strong, significant hits available, which you could then use in interpretation. They may not be the best hits but they’re the best-annotated hits.

290.0 The other factor, in particular for these ESTs and cDNA ESTs, is that, to some extent, the first pass BLAST that we’re doing, we’re using them in isolation, we’re just taking each individual sequence and running BLAST against it. But the other thing that we do is, we take all of our ESTs and we compare them against one another and, if we do that, we find that some of them group together, some of them are essentially the same sequence. And what we can do is we can do is what’s called assemble them. So we can take all the ones, which we think might be the same sequence and we can run them through a software tool that will assemble the together using alignment and work out where the overlaps are. It will produce a consensus sequence that’s based on all the different bits, different ESTs.

303.5 And we can then look at the individual BLAST outputs from when we did the individual ones and see if they’re all telling the same story or if perhaps only one has a really good hit, some of the others are not, together may help. But also we can use the consensus sequence that we get, which essentially, it’s usually a longer sequence, but it’s also a longer sequence that’s been additionally verified by having multiple goes at doing it. So it’s essentially some parts of it will be overlap, so we’ll have several different sequence runs. So if there’s an anomaly in one or other of the sequences and that maybe sorted out by comparing them. So we can redo the BLAST hits with, as it were, the consensus sequence based on assembling our own sequences.

315.5 But the other thing that we can do is we can do not only assemble our own sequences but we can go to DBEST, which is the EST database, and if we do hits ... if we get hits against that we can pull out very strong hits and see if we can make some judgement if they’re identical things to what we’ve sequenced. And in some cases it’s not uncommon to find things in there which are ninety-eight/ninety percent, ninety-nine percent homology.

323.0 We can cross assemble those with our sequences as well. And then we can repeat the process again. Of BLASTing the consensus sequence derived from assembling our sequences with other people’s sequences. So essentially we get several goes at doing that.

330.0 In the case ... and the important thing about that, is there’s a large number of sequences out there that other people have. Currently we have a target within Barley to generate forty
thousand EST sequences. There is public funding under way, which will generate, I don’t
know, maybe, two hundred and fifty, three hundred thousand of these sorts of sequences. So, I
mean, we’re producing a fairly major amount but there will also be a lot more out there that we
can actually make use of and compare.

340.0 I’m drying up. Is there anything you want to direct me to in terms of questions?

341.5 J – It’s basically with the results its just look down and see if there’s any ... if there’s multiple
ones you go further into the annotations

344.5 D – Yes.

344.5 J – Or if there’s significant ones depending on the ... what particularly you’re looking at with
your test sequence and if it’s got any matching or similar organisms like between Barley and
Rice. So if your test sequence was Barley and Rice came up, that would also be of interest and
you might follow up that one.

351.0 D – So, basically the taxonomic closeness of what we hit is a relevant factor. We’d expect to
find the strongest hits to organisms, which are taxonomically the closest to what we’re actually
doing the search with. Now, the reality is that if nobody sequenced that equivalent of thing in
anything close to us the best hits maybe something relatively exotic. But, you know, we’d
expect to find something close to it. So, for example... but also I think the other factor that’s
important is that the extent of the population of the database is also reflects in terms of what
you get.

363.5 So, for example, there are certain things, which are very highly represented in the databases
simply because there’s lots of sequencing effort gone into it. So Aribadopsis, where the whole
genome has been sequenced, there’s lots of EST sequences in there and there’s an enormous
amount of molecular biology been done over the world. There’s lots and lots aof Aribadopsis
sequences.

369.0 Barley, until comparatively recently, and Potatoes, until comparatively recently, wasn’t a
vast amount of sequences, there were only a few thousand sequences. Now, I think, for Barley,
certainly in terms of ESTs, there’s seventy thousand ESTs of Barley now in the database.
Although there’s a smaller number of defined sequences in these random ESTs. But there’s
also some genomic sequences of Barley now in the database. So we can see a few pieces of the
Barley genome that’s been sequenced, not just the genes but everything in between them that
have been sequenced.

379.5 Actually, one other thing I should say is that when we’re carrying out these searches, one of the
things we have to be very careful about is that there is ... there are lots of issues about
contamination of the databases with things, which shouldn’t really be there. Now, there are
two issues about that if we’re doing searches. One is that we attempt to pre-screen our material
to make sure that what we’re doing the query with is as clean as possible and doesn’t have
anything that shouldn’t be in it.

388.0 So the sorts of things I’m talking about is, for example, contamination from the sequencing
vector or from the adapters. So things that are not really plant sequence but are part of the
sequencing process but not everybody is as careful as we are and there’s lots of contamination
in the sequence databases. So, for example, a very high proportion of sequencing vectors
contain the lac Z gene from E. coli and if you take the lac Z gene from E. coli and search
against the sequence database you’ll find string hits all over the place.

400.5 There are other things that are present, which, to some extent, are noise but although they’re
there for better reasons, they’re not just a, as it were, not just an error, not just cleaning up your
task, the bioinformatics properly. So, for example, there’s a lot of retrotransponson sequence
and that comes from, you know, most higher organisms have a large amount of retrotransposon
sequence ... humans do but certainly many plants do

410.0 In particular plants with large genomes such as Barley and most of the cereals, really a quite
significant proportion of the genome is retrotransponson. So, for example, Barley, it’s estimated
that something like ten percent of the Barley genome is one particular retrotransponson family.

415.0 So you one all over the place at the genomic level. But these retrotransponson also, you know,
contain genes, which under certain conditions are expressed and so if you look at expressed
genes then you’ll find, to some extent, retrotransponson genes primarily present. And it
depends, a little bit, under what conditions the libraries are produced and what the plants were
... sorry ... what conditions the plants were when the libraries were produced. There’s
certainly some indication that if you generate libraries from plants that are under high stress
that you get more retrotransponson activity in that sort of condition and you get more
retrotransponson sequences in the library.
So a few percent of the libraries are likely to contain retrotransposon sequences. So certainly some of what we are doing the searching with will contain retrotransposon sequence in. And certainly some of the libraries out there will contain retrotransposon sequences even when they don't known they're retrotransposon sequences.

So, for example, the DBEST ... there are ... there's quite a lot of the EST database is retrotransposon sequences. It's plant sequence but it's a particular subset that you're not really interested in. You can almost think of it as almost a bit like noise. And genomic library ... the genomic fragments that are around ... are out there, quite a lot of genomic sequence contains retrotransposons. So what you sometimes find is that you may have a piece of genomic sequence, which contains two or three ... one or more genes of interest and the sequence in between...

And that fragment, that clone will be described as being, you know, this particular gene or these particular genes. But if you do a query with something that contains retrotransposon in it, you may get very strong hit but it's not the actual genes, it's the retrotransposons that are in the environment, not the genes. So you have to watch out for things like that. But this is something that we're starting to accumulate, as it were, a checklist of items, which we want to look out for.

Right, anything else

J - There are other things you have to look ... on this list you have to look out for. You mentioned before, Chinese whispers, to do with proteins.

D - right, one of the problems is that many of these hits, even if you get a very strong hit, and to something that's very well described, the bottom line is you want to be able to anchor this to something, which has strong confidence. It's definitely this particular gene with this particular function. Now, the difficulty is that there's actually becoming a smaller and smaller subset of the sequences out there, which have been properly anchored to some sort of biochemistry or functional test.

So frequently you'll find a very strong hit to something that's said it's such and such a gene but then if you go and follow up that particular gene you'll find that you only know that it's such and such a gene because they've done the same process six months before and found it was a hit to such a and such a gene and so on... so there is a great danger that ... that first of all that your confidence is not tied down directly to something that's of major interest.

But the other problem is that if you have a chain of inference, so if you start of with one thin where the biology is known, for example, if it's an enzyme where somebody's done the enzymology for this particular protein identifier. You then ... somebody then comes out with another sequence and identifies that sequence based on a hit to that particular thing. They may give ... if it's a very, very strong hit then you've got some confidence that it's definitely that but if it's a hit that's sort of OK then it may be a hit that's only to that particular class of proteins, not to that particular protein.

And that can generate a sort of chain of inference that gets further and further and further and further away from being really sure that you have a strong confidence that it's something that's been actually tested out by biologically or by biochemically. So what you'd like to be able to do is be able to validate these things against some biological test or biochemical test or physiological test of function. The difficulty is that's, by far, the rate limiting step.

And also the way the information is accessible means it's actually quite hard to follow that chain back. So, ideally what you'd like to be able to do ... if your inference's by a strong hit against something in the first instance the you ... and that is ... that's function has been identified by a hit with something else by a hit with something else by hit with something else etcetera.... ideally what you'd want to do is get back to something, which is ... has a validation. But the difficulty is that may be a very long chain to get there.

And doing it with one sequence is not a problem. So if there's something I'm really interested in and I want to find about what it its I can put a lot of effort into following it through, reading the papers, developing confidence and going all the way back and perhaps taking the thing that has been biologically verified and just checking the alignment with my sequence with it. The difficulty we have is that's OK with one or ten but trying to do that on a hundred or a thousand sequences is just mind-blowing, there's just no way you could actually do it.

And it's difficult to see and automatic way of doing it at present because the way the information is structured doesn't always allow for that sort of chain of inference. They're trying to move towards having a tag in the database that says this has been validated. And if that's the case then one can imagine automatic ways of going through till you hit that. For example, we could actually look against all of those ... all of the BLAST outputs and then to
try and find which of them had been properly validated. And, again, that would be an additional way in terms of how we viewed and interpreted any given sequence.

539.5 J – How do you know at the minute?
540.0 D – Sorry?
540.0 J – How do you know at the minute? Just (*) not validated
541.5 D – the only way you know is by following the BLAST output to a particular database entry and following any information that’s on … that’s associated with the database entry whether it’s back to Medline, linked to a particular paper. And, again, what you find is sometimes when you get to the Medline entry or even … that doesn’t always tell you whether or not this has been properly validated. What it may say, for example, some of these ones that I follow down, you get to that point and it says “this we think is such and such but because of homology to such and such”. At that point you know it’s not been validated and you’d have to go beyond that

557.5 And, I mean, one of the difficulties is that some of these decisions as to how well they’ve been validated depend on somebody’s interpretation of it and there’s not always a scoring associated with that. So, there’s not a quantitative score associated with it. It’s a statement by somebody, that says I think it’s … it’s almost a text based thing, which is very difficult to try and quantify.

570.5 J – And putative proteins?
571.0 <Phone rings, David answers it, tape turned off>
571.5 <Error in restarting, rest of side blank>

David said that if result based on other information expect there to be a gene, intron/exon sites, etc. If so the gene would have this combination of Amino acids. However, if it has not been shown to be expressed, the only confidence that this is a real protein is there’s something that makes it look like protein-coding region. There are two options, mark up for future reference or look further down the BLAST table to find something that is lower but real.

Jacky thought David had answered all her questions

David tried to think if he had forgotten anything. Remembered that in the descriptor line there is a range of different ways of describing proteins; genome code, genome name linked to function, systematic nomenclature, EC number. And there is an EC number website where lots can be found out about the protein. Trying to put into place a more limited set of names for genes with a limited vocabulary, which is not in place at the moment

704.0 D – generating a mutant population in Barley by using a range of chemical and radiation mutations. Now, what we will try and do with that is to be able to come up with a limited set of words, which describe these mutations. A standard approach, not just a well it looks vaguely shorter than it should be normally. We need to try to be a bit more systematic. And then if we have genes that are associated with these particular mutant phenotypes then having a more systematic naming system will mean that it’s a lot easier to then to do this sort of tracking.

710.5 But that’s not in place yet, but it’s a very active movement and it’s in one or two organisms it’s really quite advanced, particularly in Drosophila.

713.0 <Tape turned off>

001.0 D – In some cases you may find the exactly the same sorts of hits with BLASTx and BLASTn, but in some cases one or other may be more diagnostic. It just depends on what actually in the database.

003.5 J – So it’s OK that I’ve kinda, not got those … just the … I do still have the other two BLAST results that I brought last time but we only looked at one of them for each of them. So, I mean, if you want to … Though I’m not actually going to do much talking today, though, because my voice is vanishing.

009.0 So, from what I understand of what you said last time, when you’re looking at them, you’ll look down and look for similar … clumps of identifiers. It’s got Tritisome, whatever that is…

013.0 D – Triticum, Triticum aestivum, that’s a species name.

013.5 J – Yeah.

D – So that’s wheat
J – That’s Wheat.
D – Yeah.

014.0 J – And with two tomato. It doesn’t seem to … like on yours you had … the one we looked at last time had a lot of tobuti…tobicum… or whatever it…
Its name wasn’t Tabacum was it? Tobacco?

OK, the hits that you’ll get depend on the sequence that you’re searching in terms of the taxonomy of the hits you get and it will depend on the nature of the sequence and it will also depend on just the history of work on that particular sequence. So in some cases particular sequences have been worked on a tremendous amount in a particular organism or group of organisms.

So you find that the database is incredibly over represented for a particular group of sequences. In other cases that it’s very spread wide taxonomically. In something like five S ribosomal RNA is used taxonomically so you tend to find there’s an enormous collection of five S across a very wide taxonomic range.

Can I just test this has worked?

I was going to say...

<Tape stopped to test for noise>

OK where were we? Oh yeah, looking at this you look for ... from what I understood from what you said, you look for any clumps either in species or genetic identifiers.

It’s probably worth rolling back a bit. The first thing that’s really important is the score you’re going to get.

Either the score or the E-value. So, I mean, if you look at it, there’s nothing that’s got a significant score or E-value, you might as well not bother.

For this one it’s got ... although the scores are a bit low, the higher E-values, well, the lower E-values were a wee bit down. Though only the first nine have scores over a hundred.

Yeah but that’s...

You were saying that it’s hard to decide...

It’s hard to decide a threshold but I mean there are some searches you do, where you’ll get ... the E-values will be close to one and with very low scores. And in these cases you might as well not bother because there’s just really nothing of any significance there.

Right.

You’re going to get them from BLAST... The other thing is that ... So that’s really obviously the first thing you would do and then, once you’ve done that, you then have a look and see across the ... and scan across just to see what’s there in terms of any identifiers.

One of the difficulties is that the BLAST database is based on just the descriptor element from the original EMBL or whatever database. So ... and if you just ... for example, if you just look at this summary table, the descriptor line is truncated. You get it fully along with each alignment but it’s truncated here. So sometimes you have to look not only at this table but you have to look further down to get an idea in terms of what’s actually going on.

If you look at this one in particular, virtually all the hits are to the same entity. They’re all to superoxide dismutase and one that’s Copper/Zinc dependent. So that’s relatively ... that looks a relatively clean thing.

What you can also see from that is there’s a very strong hit at the top. I’m not sure what the original queries were, I think this may be a something that came from our Spruce ... Spruce library, I’m not sure.

Last time you said it was Potato

Well, I’ve given you past both Spruce and Potato stuff and it could be one or the other, I’m not sure in this case. I’d have to check on that count.

That’s a ... that’s actually P. sylvestris is actually Scots Pine. And there’s a strong hit to ... a very, very strong hit to Scots Pine and then there’s a bit of a gap to other things.

So if you get this sort of large gap between the top one and the next one you would ... would you definitely follow up the top one?

Yeah, you’d definitely follow ... you’d definitely follow up the top one anyway unless it’s something very vague like an anonymous hit to a big chunk of genomic sequence, it’s not particularly useful.

OK, of the others. How ... You’ve got clumps really saying the same thing. How would you decide, which ones to follow up?

If they’re all saying the same thing then it doesn’t really matter ... if they’re very similar sequences...

It’s just pick one and go well I’ll have a look at that one. Just a random choice.
Yeah. What you can do, and this is something we’ve only just discovered, is it depends on which version of BLAST... which BLAST... which flavour of BLAST you’re doing. This is a BLAST...

J – A BLASTn.

D – Yeah. If you’re doing a BLASTn, apart from the default output, which will just give you the table and will give you, for a certain number of the best hits, it will give you the full description and an alignment. You can actually, as an alternative way to display it; you can get it to display it so that it gives you all of these aligned together but ordered by the quality of the BLAST hit. And that can be a very good way of looking for similarities.

J – That’s one thing that me and Louis noticed... We actually went onto EMBL and looked at the BLAST input area and they have... you can choose the different BLAST matrices to use. We were wondering... One thing we were wondering is, is that going to affect what we’re doing?

D – The different ma... You can choose different matrices but the default is perfectly robust and you... we use it. I wouldn’t worry too much about that, the defaults are actually really quite robust. You’ll get differences if you’re using different matrices but not really greatly different and the defaults are pretty sensible.

J – So, but if a user does want a different one we would need to keep the parameters the same if they do a reBLAST

D – Yes you’d have to keep them the same. But people will almost never really change the lookup tables, I think you’ll find they very rarely do.

J – A couple of other questions we were a bit... cropped up while me and Louis were looking through the last interview.

You were mentioning that being able to have another species... that the species that was fully sequenced was an anchor your sequence into that sequence. We were a little confused of the significance of that. Is that just like... mapping your sequence or does it actually help in identifying the function?

D – It may well help tremendously to predict... to find the function. The reason being is that if you can locate something to a mapped entity then it’s not just the sequence homology, it’s also a mapped position homology. And that gives you increasing confidence in the identity because... so, for example, if you have a gene that could be a member of a gene family, if you have a whole series of potential points around a genome where it might be located.

If you have... if you have the most... the strongest hit it’s this one that’s unmistakable half way along chromosome one then you know that... which member of the gene family it’s likely to be, it’s most likely to be. It gives you an increased confidence in that. So you’re actually getting two pieces of information. You’re using the sequence homology but you’re also using a map location as well.

J – Is there not a risk that you got... I remember you saying before that you got domain homology and actual gene homology. Is there a risk that it might look... might have a homology to this area but that particular part appears in several different locations.

D – There is very much a risk of that. It will depend, again, tremendously on what sort of, type of sequence you are actually dealing with. For example, a group of sequences, such as protein kinases, the kinase domain will show comparative homology. So one thing that you might want to do if you are trying to follow it up properly is that you would actually not only look at the total homologies present but you’d look to see where homology is.

If its only in the kinase domain then what you can say is you’ve got something that’s almost certainly kinase. But if it’s in one of the other domains then... sorry... if there’s no homology in one of the other domains then you say well look it’s Ok, these two... the query sequence and the test sequence are both kinases but you can’t say they’re the same kinase. So you can get it down to a sort of family level, as it were, but you can’t get further than that.

But the interesting fact, the guy that just came in the door... when you arrived, Roger. He’s got an Arabidopsis sequence that he’s interested in and he’s interested in finding a Barley equivalent of that sequence. And he’d ask if you could... he hadn’t been able find anything homologous so I showed how easy it was to find things that are homologous but what I was actually warning him as you came in was fact that he hadn’t looked carefully to see where the homology is. To see whether this is really just a gene family homology or whether you can say with any confidence that it’s a homology linked to the specific thing that he’s interested in.

J – The other question that cropped up. I know you mentioned... when you’re looking for things... contamination, looking out for that. I know you said you had to look out for vector DNA, and I know what that is, but you also mention adapter...
That’s part of ... it depends on the cloning process that’s used. Some cloning processes that ... the vector is basically the plasmid or the lambda vector that the thing is finally cloned in. But there are some cloning processes that use short adapter or link homologies on the end of the piece of DNA you want to clone. There’s a number of reasons why it’s there. So, for example, in the EST work that we do, we directionally clone. And we use adapter strategy to directionally clone.

It means that we know which end is five prime and which end is three prime because otherwise it would go a random insert and we wouldn’t know which end of the clone we were sequencing from. So we clone with ... using an adapter so ... or whatever you want to call them. So that we end up with a situation where we know which is the three prime end and which is the five prime end and most of our sequencing is done from five prime end. That’s not ... Some of the sequencing is done with ESTs... don’t have a clue which end it is done from.

It should do but, again, cloning is a funny thing so you can get wobbles on the cloning. And it also depends a bit on the quality of the sequence around the vector and adapter sequence. So, in particular, if you look at sequence quality in a particular sequence run, the probability of error is U-shaped.

So right at the start of the sequence you’ve got high probability of error and you go through ... into the middle where you’ve got good quality sequence and then as you get towards the end of the sequence the probability of error rises again. Now, it’s likely that the vector adapter intermittent clone bit can often fall in an area of relatively poor sequence. So sometimes being able to identify that, as being contaminating sequence can be a problem. So sometimes the trimming that you do is confounded by that affect.

The other difficulty is that you can get bizarre cloning artefacts where you may get multiple inserts, for example, in the same clone. So that ... and we’re not completely sure yet how our trimming strategy works if you get bizarre things happening. But we’re just trying to identify any other sort of pre-screens that you could do that would actually sort out bizarre cloning artefacts. So what we might do is we might have a whole series of pearl scripts that we could run to look for certain things, certain seq ... things to look for in the sequences after we’ve trimmed them.

That’s ... it’s true for our sequences but we pre-process them if we’re going to do similar things with the sequences that are already in the database. The problem is that you’ve got to rely on other people’s ability to trim and prepare they’re sequences and we know by experience in some cases that it’s diabolical or completely missing. Some of the quality of the sequences is very poor and there lots of contamination in the databases, either with sequencing vector, with adapters or with all sorts of other bizarre things.

So, again, what we’re trying to do is accumulate, as it were, a checklist of things we need to look out for so we can actually pre ... look more closing and do data hygiene checks on the sequences.

I remember last time you were saying that to give increased confidence you’ll look at the other two BLAST type searches, BLASTx and DBEST, and also to make ... assembling ESTs to make a longer sequence. So ... can you also...

The pertinent information you can almost think of from a particular EST would consist of the BLASTn against the non-redundant database, BLASTx against the non-redundant database and BLASTn against DBEST. And then assembling that with any other thing that we can find that’s essentially very, very close to being the same sequence and repeating that process with the assembled one, but also potentially the same process with all the other sequences that contributed to that assembly.

The difficulty is that once you actually start to use all the sequence information that’s potentially out there the shear scale of the task of BLASTing becomes enormous. We did some calculations yesterday; we worked out the sequence program that we have, i.e. to generate forty thousand Barley ESTs. For us to BLAST it against the current EMBL database, in house, it would take us fifty-five days of CPU time. And we worked out to do all the ESTs of Wheat and
Barley that we expect to be in the database within the next two or three years would take us up
three years of CPU time BLASTing

227.0 J – So wait until you get better computers then?

228.0 D – What you can do ... you can then think of strategy you can assemble ... perhaps if you
don’t want to do it to that sort of scale you can BLAST the consensus sequence not all the
individual ones.

231.0 J – Right. You also mentioned last time looking at the alignments. Is that to see how many
gaps were in there or...?

234.5 D – There’re various reasons to looking at the alignments. One of them is to see ... yes ... the
number of gaps there are, to see the location of gaps. The BLAST score is dependent,
obviously, on the number of similarities and the length. But you can get similar scores based on
a reasonable hit but not fantastically all across the sequence or you can get very strong clumps
of homology and that can be quite important as well because you might want to follow that up.

242.0 So, for example, the active site of an enzyme might be very highly conserved so may be very
strong homology around that whereas the other parts of the enzyme structure maybe have less
selective constrains. So, for example, if you find there’s a very, very strong homology to a
particular area and if you have the time to follow up what that area of the ... represents. If it
has a very strong homology to the active site of an enzyme that, again, increases the probability
that you have a good hit.

251.0 J – That should be all.

251.5 A couple of the annotations, I thought we could go through the annotations now. I’ve got the
... This is just the annotation for the first one of that one, which is your Scots Pine. So if you
could just run through what you look at when you’re going through this. I couldn’t really find
ones with large comments areas.

257.5 D – It depends an awful lot on whether it genomic s ... a large piece of genomic sequence.
You can get an enormous amount of annotation if that’s the case.

259.5 So what is this. Obviously the important things ... It’s sometimes useful to be able to look at
the date, in particular the date and time it was created and certainly when it was last updated.
Because the knowledge of sequence information has increased dramatically and certainly some
of the very old entries in the database are potentially time-expired. I mean, they were OK at the
time but there may be a problem, in particular it’s not uncommon to find things which are
actually wrong.

269.0 Even if people know, now, that they’re wrong that will not necessarily get changed in the
database because the only person that has got a right to change an entry is the person who
submitted it in the first place. So the person who submitted a particular entry may have gone
on to do something completely different and is not interested any more in this sequence. So
there’s no incentive to change it ... the sequence ... to update it.

274.0 Right. So date and time. The next thing is the general description. The general description in
some cases it is very, very comprehensive, in other cases it’s very, very lightweight. So you
cannot ... you don’t have any expectations of standard of description. So there’s not a sort of
standard description that’s got to go in for everything. So the first thing, this says it’s a
messenger RNA, so it’s DNA sequence but it’s only from the coding region and that can be
quite important to know, whether it’s messenger RNA sequence or whether it’s genomic
sequence.

284.0 J – Is it the same if it said EST there?

284.5 D – Yes. That’s partially equivalent. The reason why it’s not ... I suppose you could think of
an EST as a subset of mRNA because ESTs are the general quick and dirty mRNA sequencing.
So, within a messenger RNA sequence, in some cases anyway, you might expect that
somebody had taken a cDNA clone, and ideally a full-length cDNA clone, and would have
sequenced it carefully.

292.0 Sequencing angle, orientation, made sure it ... the sequencing was all okay and then assembled
it into a finished ... a robust, clean, tidy sequence. ESTs are all about doing lots of sequence,
very high through put, quick and dirty sequences, short sequences. So there’s ... so although
ESTs are a special sort of mRNA sequence.

298.5 Right. So a descriptive line is a general description, al ... it usually tells you something about
species but it is often in an abbreviated or truncated format. So, unless you had an expectation
as to what P. sylvestris ... it could be a large number of different P. sylvestris covering, you
know, plants or nematodes or all sorts of different things. So that doesn’t tell you a great deal
about it.
You might really want to then just double check on the taxonomic descriptor lines, which tell you really what the species actually is. One of the things that you're looking at doing is automating retrieving this information because although we've got ... tell from this, in many cases, what the species is, this is the definitive description of the species.

We also ... this tells us Superoxide dismutase ... it tells you that this is a particular enzyme. Now, again, there is ... there are systematic ways of naming enzymes in particular. And this is not doing that so you can't ... you can't always expect that. But if there is a ... if the enzyme has been systematically named that's actually really quite useful because it gives you a much more robust nomenclature for the enzymes.

So it will often give you a formal name and it will give you one of these EC numbers, which is a sort of ... is a formal tag, which you can then go and follow up. And the nice thing about the EC numbers is there's a number of routes that you can actually go and get a lot more descriptions about a particular enzyme. You can go to either the Expercee website, or to the K metabolic pathways, which are in Japan. And you can find a lot more about the enzyme and find out its occurrence in all sorts of different species. You can often find out what pathways you'd expect it to occur on.

So that's a ... is a way ... If you have an enzyme that's not systematically ... it can be problematic in terms of finding a way into the sort of ... the reservoir of information. You might be able to jump to information quite quickly but there may be problems. And, in fact, in some cases it may very much be a dead end or it may be erroneous as well.

So it's a pain in the ass to have formal nomenclature ... it's terrible to actually almost do it but there's actually good reasons for doing it, as you want to make connections between things is becoming more and more important. There's a lot of interest, and I said this to you before, in setting up formal ontologies and limited vocabularies.

So where you have a standard set of descriptors that you can use for things, in the case of enzymes, but in more general terms looking up for genes and for proteins. There are active campaigns to try and do this particularly around the ... the group that's working on Drosophila are very pro-active in this area.

The only difficulty with that is that it's difficult to make it retrospective ... to take standard names and nomenclatures and work ... and tie them into everything that's in the literature. You do it by having synonyms for thing but you can still be a problem.

Right. So that's the name, you got the family. What's also useful is who's actually submitted the sequence and where its come from. In some cases that ... by experience you can identify people who's sequence you trust. You can identify institutions who's sequence you trust. Either that or ... I suspect it's more likely to be the other way round. Is that, with experience you can find if there's a lot of hits to ... maybe bad hits to a particular set of data. Particularly if it's something sequenced from a library with a lot of contamination in. You identify that as being a problem or a difficulty.

Then the next thing that's clearly of importance is that there is some reference to a publication that refers to it. Now that can be very good and very definitive or it can be relatively useless. In this case, for example, it's a reference to a publication, which is about that sequence and about this particular cDNA so it's suggests that there's actually a whole journal article, which is relevant to following that up.

What you sometimes find, particularly with a large-scale sequencing project, you may that the reference that's here is not one that's specific to that particular sequence but it may cover twenty thousand sequences so it's actually not particularly ... it's not particularly useful. But this one ... in this case, looks like it's going to be really quite useful.

The next thing that's down is cross-references to other databases. These ... there's a whole range of things where cross-references are potentially useful. These, in some cases, are to ... if it's a DNA sequence, it's to the equivalent sequence in protein database. So SWISS-PROT there refers to the equivalent sequence in the protein database.

In some cases the reference is to nomenclature databases and they are potentially useful to you. What you'll get there depends an awful lot on the organism you're working with because, in some cases, cross-links are very wide spread and in other cases they're not. Sod's law, from our point of view, is that if you're working entirely with human or certainly mammalian sequences or microbial sequences you're far more likely to find lots of cross-references but not so much in plant sequences.

OK. Then finally you've got information, which says something about the origin of the material and other information about the sequences. So the source information basically is
potentially quite useful. It is a whole series of mark ups that are really quite relevant. In this case it’s come from Scot’s Pine, it’s a confirmation of the organism.

407.0 It also tells you something about the development stage it’s come from. Included here, it tells you the cloning vector that’s been used originally. So it’s been cloned in lambda vector g t eleven. The final thing is actually a reference to a clone name, which, again, can potentially be of importance. Just in terms of distinguishing it from other versions of the same sequence. It’s not uncommon to find several versions of the same sequence in a database, they’re not non-redundant.

420.0 And there’s the cross-reference to the SWISS protein, the database equivalent. So that’s essentially a repetition of what’s there. And, in this case, it also gives ... because it’s a cDNA sequence, it also gives you protein translation from that.

426.0 And finally there’s the actual ... sorry, hang on ... Then it tells you something about the structure of it. The total length of it is six hundred and sixty-seven base pairs long. Of which the messenger RNA occupies a subset of that. So it tells you where the messenger RNA begins and ends. It also tells you there’s a poly-Adenolation site on it as well. You can also ... could be useful.

435.5 It tells you the base composition, which may be of interest but it’s not terribly useful and then it goes on to give you the actual sequence. There’s a poly-tail on the end of it. But what you find ... so, for example, one thing that the poly-tail tells you is that it confirms the orientation of it. So that’s the three prime end of gene.

444.5 So, for example, if you got something, which you thought was a significant homology but was in opposite orientation from what you were comparing it with, you might say well I thought it was a homology but it’s not ... the context is a bit strange.

449.0 J – So this is the same way you would do it? You would have poly-A at the end of your sequences?

450.5 D – We would not ... if were to sequence all the way through the clone we’d have poly-A at the end.

452.0 J – Right.

D – But you’d be quite right because we’re actually sequencing a cDNA clone, that’s been cloned in a particular orientation. So you know which is the five prime and which is the three prime end of the sequence. But the thing that ... not all cDNA clones are full-length. So that means that the cDNA cloning process may be truncated at the five prime end. So we have a five ... we know its five prime direction but we don’t know it’s definitely at the very start of the five prime end of the gene.

461.0 And similarly, if we’re sequencing you may only get four to five hundred base pairs but not a great deal longer than that. So you’ve got a gene that’s going to be one to one and a half kilobases in length, you won’t get right to the poly-A tails on the end.

465.5 J – But if it does, if you know it’s in the same direction as yours?

467.0 D – Well...

467.5 J – If it is there.

468.0 D – Yes. It tells you something about orientation but you can also find ... what you sometimes find as well is you find at the start of the sequence a poly-T. And that’s a poly-A at the end of the gene but it’s been swapped round.

475.0 J – So you look down and look for ... look at the name, look for things in the descriptor, and you look at the date to make sure it’s not too old. There’s who done it, where it came from, any papers or Medline links that might be useful, cross-references... Do you follow these up all the time or do...

483.5 D – No you don’t follow them up all the time. Some of them are actually repetitive and not particularly useful. This sort of features table, is what this region is, in some cases is non-existent. In some cases is much more heavily populated. It’s quite ... a lot of the information is ... potentially in the feature table depends on the size and also the nature of the sequence and how much time somebody’s spent annotating it.

493.0 J – Right.

D – So, for example, it may tell you about functional domains or it may tell you about whether there’s micro-satellites. There are all sorts of bits and pieces that can be there but there’s no expectation having to be there and just because a feature is not there it doesn’t mean that sequence doesn’t have it ... that particular feature. It just means that person didn’t get round to marking it up.

502.0 J – But looking at the features table is of important ... if there’s a features table there. Right, is it the same kind of process for the BLASTx. I’ve got a rather short example there. There
should be another one at the back. Yeah, they’re both quite short. I couldn’t really find any
longer ones.

D - Most of the protein information tends to be much shorter in terms of what’s ... it’s not
usually as well populated.

J – But you’d look for similar sorts of things in there.

D – yes

J - Do you sometimes get a feature table ... can you get features table in there?

D – Yes, you can get a feature table. Certainly what you might find a lot more about function
domains within the protein. So, for example, you might find a potential phosphorylation site in
a sort of... Some of these functional domains that you get marked up are very descriptive, in
the sense that they’re ... you know, very diagnostic. So they’re very diagnostic in that you can
say which particular domain really has this banged to rights. There are other things, which are
relatively weak and couldn’t really tell you much.

D - OK then, that ... is that about it then, you just...

D - yes

J - OK I’ll take that away ... write up this transcription and I’ll see if we get any more
questions (*) as we’re going through. And arrange...

D – Next week I’m out...

J – It’ll take me that long at least...

<Tape stopped. End of Interview>


00.45 J – I have a couple of questions that arose from David’s but I’ll leave that just now. Unless
you what to ...
00.51 G – No it’s OK. Do it any way you want.
00.55 J – If you go through...
00.57 G – Um... OK, what we’ve got normally at the institute is a number of DNA sequences. We’re
pulling them out as ESTs.

01.09 J – Yeah.

G – Expressed Sequence Tags. And this would be an example of the sort of sequence
information we get. OK. So this would be an unknown sequence pulled out of potatoes, in this
case infected with soft rot bacteria.

01.30 J – So this is all just the test sequence details?

01.32 G – This is me putting in formation into a HTML based database

01.38 J – Right.

G – the first thing that we tend to do with all our sequences is store them in some sort of format
and if we use HTML based pages we can all view them on the intranet at the institute.

01.50 J – Right.

01.52 G – Then we do BLAST searches. Usually BLASTn for amino ... for DNA sequence
similarity, BLASTx so we can look for amino acid similarity. And in fact it’s the BLASTx,
which we tend to regard as far more useful in many cases.

02.14 <Microphone moved closer to interviewee>

02.15 G – Good idea.

02.20 The reason is that with protein sequences you can get a much stronger conservation of
sequence information than DNA sequences. We use searches over the Internet. Two of the
places that I’ve... I use is NCBI in the states and I also use a Japanese web site as well, which
is slightly better but it’s different, it’s also a bit slow at the moment.

02.51 The advantage of NCBI is that when you put your sequence in you can actually see a
visualisation of what the matches are and they come up in colour on the screen, they’re not in
colour here. And you get a score, so if it’s very similar you get a score of over two hundred,
usually indicates it’s identical, in fact, and as you come down the range it indicates the match is
not quite as good.

03.20 J – This just a colour key then at the top here

03.23 G – Yes. It’s coloured different colours and these will also be in different colours down here as
to percentage similarity.

03.34 So we start of initially ... yes with amino acids so it’s a BLAST ... sorry ... it’s a BLASTn
search. And this is one that we’ve called ... it’s known in the literature as a Phi like, so
phosphate induced. Several years ago a Japanese scientist was looking at tobacco cells, which
were starved of phosphate and when he added phosphate there was a gene, which was up regulated.

04.02 He sequenced this. It has no function. So there's no enzyme activity associated with it. It's an unknown protein, which he has given this name to, it's not like a phosphatase or kinase or something like this. And, whenever people find similar sequences they call them Phi-like. So, they are like, Phosphate induced like and it was Phi-one, which was the first one, and that's as far as anyone's got.

04.31 So in this particular case we put the DNA sequence in and you can see where it's aligning as well. This is also quite important. If you have a sequence it may be that only part of that sequence is aligning with information in the database and with a graphical representation like this you can see if it's the whole sequence, which is matching.

04.56 This is quite important as well for picking out vector sequences. If there's a little bit of vector still in your sequence it tends to show up as short matches, usually at one end and it tends to be with matches of genes from organisms, which are totally unrelated. So it's not unusual to find matches with human genes. And in general, when we're looking at plants, the matches we're looking for are other plants.

05.29 Potatoes are what we are looking at but in fact there may not be many sequences in the international databases from potatoes so we do expect them from tomato, which is a similar ... a taxonomically related plant to potato, tobacco, and then Arabidopsis. If it's Human, human, rat, mouse, something is usually wrong. So what we're looking for is plants, plants, plants and some consistency with what the naming of it is as well.

06.08 Also we've got these score values. Now there is no ... with the E-values ... there is no precise number above which the match is real and below which it is not real. Absolutely it doesn't exist. If you have a long DNA sequence then you expect a good match over a long distance and this should give you a high E-value. If you have a short DNA sequence the E-values will be quite low by the very nature.

06.41 But you can look at it and critically say this is a very good match even if the E-number's low and you can look at other factors and say this is still real. So here we've got a list with e-numbers in this case the e-value of nought, which is actually the same sequence, which it's picked up in the database or close enough, down to minus fifty, thirty-seven, minus thirty-five, thirty-two, twenty-four, twenty-three, minus sixteen, fourteen, fourteen, ten and then you go to minus four.

07.12 It's obviously that you start get very suspicious when you get E-values of minus four but at what point of this scale is the sequence different. It's not always possible to tell from DNA sequences. But here, we look at this particular page, this is showing the alignment with the best match. You are looking for long sequence similarity, huge numbers, it doesn't matter if there is the odd base, which is different and you are looking for alignment over the whole length.

07.47 So we are looking here, we start with the first base of our sequence that we submitted, that's looking good, right the way through to the last one. If it stops half way through you think why is it not aligning in that particular case. If we move down the page we can see that we have an attempt by the BLAST search engine to put the alignment in different parts of the sequence as well. So here we're looking at our sequence numbers four hundred and three to four hundred and seventy-four it's trying to match with another part of the gene.

08.25 This is also not unusual, when you get repeating motifs within a protein, that the BLAST engine will try and match the correct match but also try and do an alignment further down the gene as well. So this isn't a problem. As we go further down this list of possible matches, again we're seeing, still quite long lengths of alignment.

09.03 – 09.24 <mutterings> Right. I've forgotten what I was going to do. Yes, I've shown you the wrong thing. Right. This is the perfect thing. This alignment was something I was going to come to after.

09.24 This is the alignment for Phi. And here we can see some sequences, again coming in at this end, exactly what I indicated before, this does suggest that there's some vector sequence in there, even though we think we've trimmed it out this is probably vector. And, again, we've got some huge long lists of numbers.

09.56 J – There's human in there

09.57 G – There's human in there. This human is probably vector alignment. So we've got E-values of ten to minus ten or nine, it's probably vector and the way that we find this is if we turn to these, you've got a short sequence, which is a perfect alignment, this is over forty bases long. It is perfect. And it is a perfect match to a number of different genes, totally unrelated. And you see the same numbers cropping up all the time and this is obviously vector.
10.40 When you find matches to the real gene again you’ve got a long distance, some variation in base pairs, this is because you’re looking at a different organism, maybe a different member of the gene family as well. What we are looking at is genes, which are up regulated in response to infection. It may be that with much of the sequencing that’s in the databases it may be genes from healthy cells and ours maybe slightly different. So it can be a different family member.

11.12 And many of the genes, which we’re looking at, do in fact belong to families of related genes. There’s never an absolute number of how many genes are in the same family. It may be two or three, it could be sixty or seventy. Any number between those two and we have no indication of what it is.

11.32 If we do a BLASTx then this is where we put a lot more emphasis on really accepting that this is a real match or not. Amino acids tend to be much more highly conserved than DNA sequences. And again, we’re looking for an alignment, which is a long match not just a few amino acids but in this case hundreds of amino acids long.

12.06 You can see that there are some areas where there is no alignment, it doesn’t have to be perfect all the way through. You can see long areas (one, two, three, four, five, six, seven, eight) maybe about ten or eleven amino acids, which are perfectly matched up. We are going to get conserved areas so that within an enzyme you would have an active site, for instance, and this active site would be conserved within every family member.

12.37 As you move to the five prime end, which is the beginning, you tend to lose sequence similarity and when you go to the three prime end, as well, you get lots of variation. So we don’t always expect three prime ends of genes to be perfectly well matched but we do expect the central region, which is usually where the active site is or binding site … We expect these to be highly conserved.

13.03 If we look down this list, we have here the best match is a Phi phosphate induced one, the second one is phosphate induced. The third one actually is not named, it’s from Arabidopsis but then they’ve put in as a possible Phi one-like. The fourth one, phosphate induced. The fifth one, it’s not named by the people who’ve put it in, but you can see here, we’ve got, again, an long alignment, strong conservation, it is going to be a phosphate induced.

13.37 It’s just the people who put this information in the database didn’t do their bioinformatics properly and there’s a lot of this going on. So they’ve not named it. We then carry on moving down the list of matches and two or three later we get someone else who’s said this is phosphate like, phosphate induced.

13.54 And you can see from here we’ve still got this long sequence similarity but as we move down the list the absolute alignment is breaking down but we do have areas, which are still highly conserved. And, again, this would indicate maybe some phosphorylation sites, active sites. As we move down the page still we’ve got the software trying to align over a long sequence.

14.23 This is, again, critical and if we look at what is conserved we would expect now to see … these amino acids, which are conserved, to be conserved in every single case, all the way up. Again this is useful. If we don’t see this conservation we might say well this isn’t right but in fact, that doesn’t happen. Once you start getting these family members something’s conserved all the way up.

14.46 Even though there is no information in the literature about what the active site is, you can see for yourself that something is getting important here. But you’re only talking a small number here, so for here you’ve gone for five amino acids and then there’s a gap of two and another three. It’s not the case that we would have to have twenty amino acids in succession, which are perfectly aligned.

15.11 We can have a small number, a gap, another couple, a gap, another couple and it depends upon the three dimensional structure of the protein as to how this would fit in. Because a protein turns on itself you can expect parts of a protein, at a distance to each other, to be physically quite close because the protein’s turning over. And interacting with a substrate.

15.40 So this is a … one of the sequences, which we think is very interesting, this phosphate induced one simply because we were the first people to show it induced by a pathogen. And therefore that makes it much more interesting with us. We’ve tried to look for active sites (I’ll come back to that one later) but finding the function of an unknown gene is very difficult … or an unknown protein.

16.12 I have pulled out another sequence to look at. This is again from potato induced by erwinias and here we do a BLASTn. It’s not … the matches are quite good but it’s not over the whole of the sequence, it’s probably a three prime end, which is not matching perfectly. But we have got some consistency with a name and this is a protein phosphatase, P P two A regulatory sub-unit. Phosphatase’s exist as three proteins binding together and this is one of the three proteins.
The matches are good enough that it actually matches with something with a known function. So a phosphatase is involved in dephosphorylating, or taking away phosphate groups from proteins and is a well-known regulatory enzyme. We can have … again you look at the best alignment, it’s seriously long, this is with a tobacco one. We’re talking about potato protein. There’s only a few bases, which are varying all over the sequence. You’ve got it right from the very beginning.

It’s attempting to make matches in other parts of the gene again. If we keep going down we’ve gone for Cicer, it’s a Spruce I think, protein phosphatase. So a different plant. Medicago, again, protein phosphatase two A regulatory sub-unit. We’re fortunate here that quite a number of genes have been discovered in the databases. They’ve all been accurately identified as a phosphatase two A and, again, we get total consistency all the way down.

As you move further down, this two A would probably exist as a number of different genes and we expect the sequence similarity to be not quite as good. We’ve actually come here … we’ve got one in mouse and, again, it’s matching with a two A regulatory sub unit but you can see the sequence similarity is very short. It’s so short we would never ever have accepted this as accurate but because it’s at the end of a lot of good matches, it’s giving us confidence in our alignment.

The best matches, though, again, are back to BLASTx, which allows us to compare protein sequences. And, again, this one is quite stunning because it’s protein phosphatase, protein phosphatase, all the way down the list. And I think we’re looking at thirty different alignments and it’s still protein phosphatase two A, two A, two A, all the way down with nothing else in the list.

And you can see here, strangely enough, you’re still getting matches with protein phosphatases and a kinase, which is a some sort of enzyme, at what is a very bad E-number, these are really appalling E-numbers and on their own you never would have accepted them. But because it’s coming at the tail end of a list where the E-numbers are good.

And the E-numbers here, to minus ninety-eight, minus ninety-six and if you come part way down the list it’s minus forty eight. These are very good matches and if we look at the alignment, it’s a good match. What we are never quite sure is which … which frame shift is … which frame … translation frame is the accurate one for making amino acids. You know that three bases make each of the amino acids?

J – Yeah
G – Well the problem with this is where do you start reading? If it’s in the first, second or third you can get three possible amino acid sequences and if you read it in the opposite direction you get another three. So there’s always a possible six translations, or six different frames. And here it always comes frame minus one, minus one, minus one. If you’ve got a frame shift it will suddenly say maybe a minus two as we do here on the second alignment down.

Here’s a minus one (²) here, we obviously have a fr … a misread in our sequence and it says at a minus two (²) frame shift, here we get a match. And this indicates that we’ve sequenced slightly wrong, our DNA sequence, we’ve got one base out and it’s suddenly shifted the reading frame but because it’s perfect you think that at this particular junction here (³) … it’s here, in fact (³), you’ve got one base out.

And we might, in fact, if you were publishing this information you’d probably go back and look very closely at that particular point and see if the DNA sequence is out by one base. And, again, we are looking very closely at the numbers (pity we can’t show you on the microphone) but if you look here, this is the protein that we’re trying to match to one hundred and thirty-nine all the way up to two hundred and eighty-nine.

So we’re looking here at a slightly early region ninety-five to one hundred and fourteen, so it’s going that way. It’s also attempting to align to a different part of the sequence three hundred and ninety, which is further along that way, to five hundred and twenty-three. So it’s taken the sequence that we’ve given it and tried to align it with two different regions, the second match is very bad, because it’s not the accurate one. But, again, we’re talking … it’s not unusual to get similar domains within a protein.

So it’s a poor match later on. But it just adds to the overall feeling that we are talking something accurate here. And as we go down the list of possible alignments, again, you see this consistency of lots of attempts here …. OK, here’s one from rice, PP2A from rice, it’s tried to align it with different frame shifts, minus one, minus two, somewhere else at minus one, somewhere else at minus one. And you can see this long length, if you like, of possible alignments. The length is always important.

< Taped stopped briefly >
23.26 So that's the main route by which we try and find matches for ESTs. Occasionally we have sequences, which don't match with anything, even a phosphate induced, it's not a function but a least we can put a name to it. So we use pieces of software over the Internet, which attempt to give us further information. This is a printout of one, nice piece of software, which is available in Edinburgh, it's looking at three-dimensional structure of proteins.

23.59 The problem with this is that you've got to be sure that you know how to read the protein sequence. As I've said, there's six possible frame shifts, if you've got any mismatches in the DNA sequence you can't produce one complete frame shift or one amino acid sequence. It would be in several little bits. But if you have some good DNA sequences you might be able to tell which frame is correct because it's ... there's no stops in it, if you like, it just looks accurate.

24.28 If you can find one of these then you can put into this data ... this piece of software, which looks at three-dimensional structure. Obviously to get an accurate three-dimensional structure of proteins, you need to make crystals and it's time consuming. There is an alternative way, looking at some general characteristics of various amino acids and this piece of software is quite good at that.

24.57 It looks at beta-sheets and coils and it looks at the families of amino acids. For instance, amino acids belong to families, which are either acidic, hydrophobic, hydrophilic or neutral. So if you have a leucine, for instance, you may be looking at amino acids, which are like thalines or isoleucines, which have similar physical characteristics hence physical chemical characteristics, which are not quite the same.

25.38 So you can group amino acids in terms of families and it can be that you can have proteins, which are structurally related, functionally related, which have slight differences in amino acid sequence because the amino acids are belonging to other members of that family. So we can put our sequence in and this software will find matches according to the structure of the protein and it will give an indication of, if you like, what else is in there.

26.12 It gives you, in the printout, code numbers for sequences, if you go back into the code numbers, you might find if one of these has a function. By in large, when we've done this with sequences like the phosphate induced one, all we pull out is all the other phosphate-induced sequences. We don't pull out enzymes with a name, which is really quite frustrating. I had hoped that we'd pull out something with a proper enzyme name.

26.42 If we put the phosphatase two A in we pull out all the other phosphatase two As and a few more, which haven't yet been assigned a name. But it is possible that maybe one in twenty or one in thirty unknown proteins that you'll put in might then match up with something with a known function. It also can be useful in that it shows proteins with different five prime ends or three prime ends, i.e. the different lengths so if you have ... as I've said before proteins exist in certain domains, if you like, and functions.

27.25 You may have a binding region for a protease and then you have a protease core, which has protease activity and this binding domain would show up very quickly and very nicely, with this piece of software. It is totally dependant, though, upon being able to produce an amino acid sequence and that, often, is very difficult.

27.49 The first DNA sequences we get out are not perfect, they are close but it can be very difficult, therefore, sometimes to interpret what the amino acid sequence is. So we can be stuck. An alternative way of looking at information is to use some software, which is available at the Stanford University. This is called matrix searches, again, it can be a little bit useful but not always successful.

28.32 Basically what you are doing is putting in a protein sequence, amino acid sequence, and it will find regions, which are associated with certain types of proteins of known function and, again, it gives scores with e to the minus seven, or what ever, in this case. We put our sequence in here and looking at a printout. What is obvious here is that it's looking at different sequences of amino acids within the protein.

29.08 The first one would be amino acid ten to amino acid twenty-three, the second one down the list is amino acid ninety-seven to amino acid one hundred and eighteen. The first one it says, this is typical of seed storage proteins, the second one says, it's typical of inhibiting beta/alpha chain signatures. These are quite different proteins and therefore it instantly makes you think that this isn't going to be real. If you had same, same, same all the way down the list, as we talked about before, then you can start getting excited.

29.39 Here the E-numbers are not that high but it's this lack of consistency down the list, which is beginning to tell. If we were to put the protein phosphatase regulatory sub unit into this piece of software it should find sequences, which are very typical of protein phosphatases and the
consistency down the list would be there, instantly. If we were to put in the phosphate-induced one, which we have done, it tends to come up with nothing of serious significance.

30.11 And, again, we’re fighting an uphill battle trying to put a name to something with no function. Obviously there are sequences in there, which are highly conserved and are going to be good characteristics and good signatures, if you like, but until someone can come up with some wet biochemistry and has found a function for this protein, that information won’t go into the databases.

30.36 And, again, it’s only maybe one in twenty or one in thirty proteins of unknown function that we can pull out something with this software, which tells us, it may be this or it may not be. That is not proof but at the moment we are looking at large numbers of ESTs and we are trying to find out which ones are the most important. So if we can put a name to them and it says this is an enzyme involved in signal transduction like the phosphatase, all phosphatases are going to be important.

31.13 We’ll flag this one up as important. The phosphate-induced one, it’s nice be able to put some sort of name to it but we don’t know how important it is and then there are other sequences, which show no alignment. Now these could be the most exciting of all because no one has yet worked on these or it could be the least exciting.

31.35 And the problem we have is that if we talk of unknowns we’re then consistently talking of unknown, unknown, unknown. And we don’t have a clue as to what the possible function is. So we have that group as well.

31.55 < Taped stopped briefly >

32.02 So, on screen we’re looking at the amino acid translation of the Phi. Proteins start with methionine traditionally, we’ve gone a little bit before methionine, and they end up with a stop region, here, and, again, our DNA sequence goes beyond the stop. So if you want to look at the protein you can cut out, starting with methionine and go to the stop. So we copy what the amino acid sequence is of the accurate protein.

32.37 Go to Japan, if they’re working. So we’ll do a BLASTp in this case. So it’s comparing protein sequences with protein databases. Do a BLASTp, put the sequence in. It says maximum number of sequences to be reported, five hundred, this is a bit high, so we’ll knock it down a bit, fifty, two hundred and fifty. There are fewer protein sequences in the databases than DNA sequences so it’s usually a lot quicker to do these alignments. And we tend to use the default settings for the software.

33.21 So we now have, on screen, the alignments looking at protein versus protein, that’s a BLASTp, and, again, we can see, Phi-one, Phi-one, Phi-one, probable Phi-one. And then these are all good matches because it’s proteins. So the worst of these good ones is to the minus thirty-seven and suddenly you’ve switched to something of four and this is against mouse. So you can see that we’ve got this nice little grouping here of all phosphate-induced ones.

33.56 So let’s go and see who published on the best of these. So we’re looking at the best match.

34.19 Right, the first match it’s come up with is someone who’s been sequencing part of the Arabidopsis sequence so we’ve got huge amounts of sequence information on screen. They’ve gone for an open reading frame, they’ve said this is a likely gene and it’s a phosphate one-like, surrounded by lots of other genes. Let’s see ... it’s Arabidopsis, part of a Arabidopsis chromosome.

34.58 So sequencing information where people have been simply sequencing the Arabidopsis genome, they put up ... put into the databases huge amounts of DNA sequence. They guess at what the open reading frames are and they try and do matches and we pick these out. Let’s find the tobacco one. This one should be ... yes it is. Someone’s just put the sequence information in for this one gene.

35.33 Yes, this is the Japanese group who originally published the Phi-one in June ninety-nine. So there’s their sequence, DNA sequence. There’s their translation of it. From tobacco and they don’t tell you much more about it and that because they don’t know much about it. If we look at the Medline reference it’ll tell you just how they isolated it from some tobacco cell cultures.

36.20 The difficulty of mining information like this is there is sometimes information in abstracts and text, it’s difficult to mine it by computers. So we tend to look at it manually, if you like. And if we look at ... let’s see, what else can we look at. There’s one from rice, a wee bit further down. I would guess this is not going to be annotated properly. Rice is probably part of the rice genomics ... sequencing long lengths of the rice genome.

37.04 And the bioinformatics side of it might be done well or it might not. That’s right, it’ll be this one probably. So it doesn’t in fact give a name, it is part of a rice sequence. The annotation’s
not been done. So they don’t even name it as a phosphate-induced-like sequence. It’s just
unknown, hypothetical protein.

37.40 J – So when you’re going through these annotations do you jus ... do you start at the top? Or
do you...

37.45 G – Always start at the top, looking at the best match. Here the best match is E to the minus
one hundred and fourteen, that’s seriously good.

37.52 J – Not in the BLAST results but in the actual annotation file.

37.58 G – Oh, sorry. What this one?

38.01 J – So you start with the first hit.

38.03 G – Always.

38.04 J – Then what would you look at first in the annotation’s file?

38.12 G – See if there’s any information about how they isolated it, whether or not it was from
diseased plants. We are plant pathologists preoccupied with responses to a pathogen. So we
are looking to see if anyone has information about stress responsive gene. As I mentioned here,
there’s a couple of these sequences, at least, are simply sequencing parts of the genome. So
there is no information about how these genes might be up regulated.

38.44 Any sort of stress, if it’s drought, pathogen, (*), no information what so ever. If we can find
sequences, which are stressed, maybe a drought, we can get excited. And, in fact, some of our
pathogen-induced genes can also be induced by drought, salt stress, heat stress. And we then
say, yes they are stress responsive, we have this general phrase of a stress responsive concept.
And we say, that’s good, someone else also have information.

39.18 If you like, it’s very difficult in life to know that you’ve got confidence to be first and the best.
If you find a gene that no one else have ever found before, it’s to be induced by a pathogen, you
can be very excited and say we are first. But equally, it’s a little bit more reassuring if someone
else has found something that’s stress responsive. You say, right we have confidence in our
data now. It’s a new angle to it that someone has got something similar. You’re on the right
lines here.

39.50 J – One thing that David mentioned was that, the date in the annotations something ...
sometimes the work can be time-expired. How important is that in your ... from your point of
view.

40.05 G – We often look at the date also, the first thing at the top is the date at which the sequence
was put into the database. The amount of sequence information is getting more and more and
more. With a phosphate-induced this sequence information was only put in the two years and
we have to resubmit our sequences on a regular bases. And we can see that suddenly one week
a new gene has been put in, if it’s stress responsive we know, for instance, that we are into the
latest area.

40.42 Ours is bang up to date. And that here’s someone else, in another part of the world, also
potentially competing with us. They’ve put a sequence in that we’re working on. They’re
clearly working on it. What are they up to? It might have a publication linked to it with
Medline, it might not. If it is, then, OK, they’ve published. If it’s not, it suggests that they’ve
submitted a paper and it will turn out in press in a few months time. People tend not to put
sequences in unless it is part of a paper.

41.15 Although that’s not true for these big genome-sequencing programs, they just put sequences in
anyway. But if it’s a single gene, there’s a reason why it’s in there. So, yes, date is important.
We look also critically at the authors and which part of the world is doing it. There are a
number of scientists around the world involved in plant pathogen interactions, who are very hot
scientists, if you like. So the last ... the boss of the lab, the head of the lab, his name is always
last.

41.52 And if it’s a lab that we know of we will give this more value, if you like. Oh, he’s working on
that. Perhaps we know about it, perhaps we don’t. If we have never heard of this gene before
we’ll have a serious look at what the implications are for why he’s working on it, then go and
look up any papers associated with it. If he’s interested, it must be interesting.

42.21 J – Do you also have, like, ones you’d be a bit suspect about, authors or places, because you’ve
had experience that their sequences they’ve submitted have been ... very bad contamination in
there.

42.35 G – We’ve tended not to remember that. We are conscious that there are huge numbers of
sequences in the databases, which have not been trimmed of vector. It is a disturbingly
common event. I would almost say the medical sciences are worse than the plant sciences but I
just don’t know, we’ve not done it in any quantitative way. If it’s a single sequence ... no. I
was going to say, if it’s a single seq … single gene, it might be trimmed better but I don’t know if that’s the case or not.

43.21 But there is no doubt about it, there’s vast quantities of vectors, it’s quite disturbing. One of the things about these databases is that even if I was to find an error we can’t modify the database, the people who look after it will only modify the sequence if it’s the authors themselves who ask for the sequence changes to be made. And quite often some labs have moved on. They don’t tend to go back and check their own sequences. You can tell that from … sometimes from the annotations.

43.54 The sequence information may be two or three years old and if was to do a BLASTx on their sequence I can come up with a match and it’s not yet annotated. So it’s quite clear that the aut…

44.05 <End of tape>

Gary only the author can change a sequence in the database

Jacky asked about protein conservation

Gary They were ninety eight percent conserved within a species
If not important for function, even within the same species can see variation
Expect variation in three prime ends because it has no consequence

Jacky Explained that David looks at E numbers and the value of the first hit and asked if this was a time reducing step

Gary Agreed that it could be time constraints
In a case where very bad score but could think it was a real match because there were ten perfect amino acids at the three prime end
The location with in the gene is an important factor
May not always know, but sometimes know the position from, for example, stop codons

Jacky Asked how close do the species need to be for Gary to consider it
Gary If it goes outside the family, expect poor DNA but still reasonable amino acid match

Jacky Asked if it is possible if it is a highly conserved protein to have this in animals as well as plants
Gary Yes, if believe that the evolucion of plants and animals have a common ancestor
If there is similar machinery, expect sequence similarity at the protein level

<End of interview>
T – Only if you think in terms of plants whereas I work with soil micro-organisms. So, you have some soil and you extract total DNA from the soil. So the smart thing, the way that we’re studying this is we did a lot of cloning in order to exhaust the types that we have. Well, the first step, actually, is to pull all the Nir K types that you can out of the database. With this particular project we brought all the Nir K types out that we could, both by looking for Nir K as a word search in Entrez or something.

And then BLAST it as well in case there is any examples of the gene, which don’t happen to have the same name or the name doesn’t appear in the title or somewhere else where the Entrez program can’t find it. We then align all those together and design pacific primers that will amplify all of the Nir K types that you have in the database. And then you go out into the field, extract total DNA and amplify it using those primers and you have to assume that you’re amplifying all of the Nir K types that are in that system.

J – So, basically, you’re given a clump of soil and you have an experiment to just pull out anything that is just DNA?

T – Essentially.

J – And it will ignore all the dirt, all the cell membranes.

T – Yeah, we extract total DNA from soil, and you can extract total RNA as well.

J – So you would … That way you wouldn’t know what it’s come from, it could be any…

T – Because the diversity of soil is so high you could get tens of thousands of organisms in a gram of soil.

J – I didn’t think it was that much

T – It’s frighteningly high.

J – Right. OK.

T – So the problem that you have is in separating out all the types that you have within that. So, obviously, you do a lot of cloning and sequencing in order to work out which types you have and this is an example of that that we’ve used. I can just print that off for you, we can collect it later.

<Checked progress of BLAST similarity search... still long time to wait>

T – You’ll probably find that this is a rubbish piece of sequence now. It’s not Nir K; it’s just a piece of rubbish. So then you identify all the different types and then we’ve been designing a high through put approach to measure the proportion of those types that are present in an experimental system. So, different plant types or different treatments or whatever but you need to know what’s in there first. If that makes sense.

J – Kinda, I’ll go over it again.

T – The actual Bioinformatics side of it, I suppose, is you get your sequence back from upstairs, you need to make sure it’s a Nir K for which you could use BLAST, for example. And then you produce an alignment and you look at phylogenetic trees to separate them into clusters of types and anything else and that’s where you start selecting the groups, that gives you the information.

J – So what’s … after you’ve done the BLAST, then you would work out what species you’ve just got that from.

T – Yeah, sometimes if I’m very … when you’ve been doing the same sequence for a long time, you can sometimes recognise that it’s the right thing and you’d put it straight into your tree or something anyway. But essentially the BLAST is a quick screen. You can put it in and, if you do it in the morning, in twenty seconds you discover that yes it is or no it isn’t and you can know which direction it is, which you would know from the primer.

J – Right.

T – It’s just a confirmation tool really, which is slightly different to randomly shotgun cloning cDNAs or whatever else you’re doing and saying, Oh, I have no idea what this sequence is, let’s try and find out.

J – Right.

T – Which is something that we do as well but it’s probably not most common thing. So I would expect when this comes up...

J – Eventually.

T – Eventually that the ... there would be about ten or fifteen Nir K things at the top of the list and then it’ll start going away from that.

J – So, for you it would be basically, when you’re doing the BLAST, you’re just looking for the same function as what you’ve got.

T – Just looking … to confirm what you think you’ve amplified is what you think you’ve got, in this instance.
When you do have an unknown function when you’re going through it you’re looking, I mean, talking to David and Gary, they say you’re looking for clusters of the same type of thing of quite high...

With a high support in the...

<Checked progress of BLAST similarity search... still long time to wait>

It keeps going up.

Yeah, sadly.

Gary was also saying that you look for alignments and looking for long alignments.

But that’s ... Yeah.

Would you look for that in you’re confirmations as well?

Yes, absolutely. I would expect ... I would expect that anything that is Nir K, it would find similarity along the length of the whole product.

Right. If it was just for a small section would you be a bit ...

I’d be a bit concerned about that.

... suspect.

If I’d only found about ten base pairs or something then it may be some sort of small motif that’s shared between the Nir K, in that area of the Nir K gene and in other.... Yeah, for sure and also if the (trying to think of the word) confidence limits are very low as well that would obviously affect your choice.

Gary mentioned that it depends on the length of the sequence, depending on ... that varies.

Well I would assume although I don’t really know that if you only had a small section of your sequence, which is recognising something in the database then that would have a knock on effect for the confidence limit. I’d be amazed if it didn’t but maybe I’m set to be amazed.

Because, obviously, if you put in five hundred base pairs and only fifty base pairs of it is being recognised as similar, that’s a far less rigorous match than if it’s across five hundred even if across the five hundred the actual level of similarity if lower. If that makes sense. So if you have seventy percent over five hundred base pairs that is, presumably a more rigorous match than hundred percent over fifteen base pairs and the rest of it is completely dissimilar.

I think it’s more dependent on the actual length or your sequence.

No, of course, it has to be set. You would get absolute perfect ... if you had a hundred percent match then you would expect that that would be the best possible significance. But as I mean as a ratio coming down from that. I mean, there is a limit below, which it just won’t really work. But if you have five hundred base pairs or if you have fifty base pairs of your five hundred base pairs then you’d expect then you would expect a far lower confidence from your fifty base pairs than.... That’s what I meant, if that has clarified it at all.

Gary told me that it depends on the length of the sequence, depending on ... that varies.

Well I would assume although I don’t really know that if you only had a small section of your sequence, which is recognising something in the database then that would have a knock on effect for the confidence limit. I’d be amazed if it didn’t but maybe I’m set to be amazed.

Because, obviously, if you put in five hundred base pairs and only fifty base pairs of it is being recognised as similar, that’s a far less rigorous match than if it’s across five hundred even if across the five hundred the actual level of similarity if lower. If that makes sense. So if you have seventy percent over five hundred base pairs that is, presumably a more rigorous match than hundred percent over fifteen base pairs and the rest of it is completely dissimilar.

Still waiting. I tried to print some up this morning, myself. My machine decided it wasn’t playing, it just kept running out of memory and crashing.

Really? What, just printing them off?

No, just trying to put them into word.

Oh, right. So you copied the page and tried to paste into word.

Yeah, it normally doesn’t do that it just wouldn’t play cause I can’t ... I can’t actually print from my computer. It’s not linked up to a printer yet.

Ah, so you have to save it and go to someone else’s machine.

I have to save it and go somewhere else and print up, which is just a drag.

Sorry, I should have done this, this morning but things just got a bit out of hand. I sort of thought that it’d take about two minutes or something. What I could do is ... let’s see now. Open another window and slow it down more by setting it of again just in case that one’s not working very well for any particular reason. Just to make us feel better about having a lower number that I’m doing this, it’s not going to make any difference, I’m sure.

See, it’s starting to go up already.

It’s going up slower.

Do you use the annotation files in the databases at all?

When you asked me that over the phone. What I tend to do, particularly in the initial stages of designing a primer, then what I would do is go from the list from your BLAST to the actual file, in Genbank. Take the sequence out of that file, check it, maybe check that the reference is one that is relevant.
10.32 Make sure that the gene is really what I want and then take that DNA sequence out and then align that to my own. Either as a standard in terms of to compare our experimental sequences from or as a sequence in order to either design or approve primer sequences. Does that ... is that what you mean? It'd be easier if it actually came up.

11.06 J - So you take the sequen... You get the BLAST result, you'd follow the...

11.14 T - So what I would do if I was designing a primer is ... I would open up another one, slowing it down even more. I'd start of with say Entrez, OK, which it would search Genbank for and I'm going to put in Nitrite Reductase, for arguments sake, and press go. And then that would give you a range of different types, for argument's sake.

11.43 And you would scan through this, you could probably find it ... this Aquifex presumably is fairly new I'd guess, I can't remember seeing that before. But for arguments sake, here's a Paracoccus Nir S, which is an iron containing Nitrite Reductase. So you could open that up and this gives you ... is this what you mean, that you can go to this from your list. So what this tells ... you can check that it is exactly what you think it is.

12.12 J - So do you look at these before you did the BLAST?

12.15 T - No...

12.18 J - Or do you follow these from...

12.18 T - It depends on what I'm doing. If I'm checking one of our sequences obviously you can't look at this before hand because you haven't got that information relating to the sequence you've put in. But when I'm designing a primer I would obviously want to check all of this before I actually did anything else with it because you want to validate what you are using before you actually start doing that.

12.36 J - Yeah.

12.37 T - So you can check out exactly where it is, it's accession number and everything else. All of which is obviously important if you want to go back and then I would remove this section, here. Put that into word, remove all the formatting, put it into a system, which the alignment program ... a format which the alignment program will accept and then just align it with everything else that I had before, if that makes sense.

13.06 Or, if you go for the Aquifex, if you got a BLAST search that pulled up Aquifex complete genome, obviously that's not really helping you very much. But if you look at... I'm sure I can find another ... it's interesting this ... right, OK. This is a mycobacterium, complete genome. In this you've ... if you pulled this out as a BLAST it would actually be quite a small section of this but presumable it would recognise the whole of your section.

13.55 But you can ... this contains any number of genes so you would check through here to check which section of this is actually the bit you want and you only remove the section that relates to the gene that you're interested in, which we should be able to find. I haven't really done this in too rigorous of a way because ... so, say that said Nir S or Nir K or whatever protein you were looking for, you know where it is in that because it's between fifteen thousand, eight hundred and six and four and eighteen thousand, five hundred and twenty five.

14.42 So then you would locate that in here and just cut out that section, so you only actually use the section, which is relevant because the rest of it is other information. Because things get put into the database either as sequence relating to a particular gene because someone has cloned a gene by function or whatever else or, increasingly, chunks of complete genome sequence. So you're, obviously, you're not interested in twenty thousand base pairs of that, you're only interested in one thousand that relates to your gene of interest.

15.14 J - What do you use the aligning for, again?

15.17 T - Right, OK. This is still going. You can use it for two things. What you're doing is aligning sequences by similarity. If we look at this one. You can use ... if you look at the named types, these are ones that we've pulled back of the database, the Rhonabactus pseudomonous for example, these are all experimental sequences and these are also database types.

16.15 So where it's named it's come from the database as such or you can generate them from type strain if you so desire using set primers. So what you can do with this is you can look for areas where there is high degrees of homology to design the PCR primers that you're going to use on your environmental samples to try and ensure that you will amplify all the types that are present within them.

16.33 J - So it will be, you'd look for something that's going same on all of them and cut that particular part

16.40 T - So if you had a primer, I mean, a primer's only about 20 base pairs long. So if, for example, it finished here and you put that 'C' in then you wouldn't amplify Rhisobium because
it doesn’t share that base position but if you ended it on double ‘G’ you would amplify all of those, for arguments sake. I mean, that’s just pulling out … but obviously you’d do this over quite a long section.

17.00 J – Right, so you’d be looking for a sequence before the gene you wanted.
17.04 T – No, well it depends. Obviously the longer the sequence you can get back the more accurate the information, the better the support you can get for everything you’re doing. But gene order can vary between organisms anyway and things can be inserted/moved around so you generally look within the coding region, which is likely to be more highly conserved than (*).
17.27 So you don’t … because all we’re trying to do is … we’re not looking at the actual … we’re not looking at studying the function of the gene or anything. What we’re looking at is the diversity of the organisms that are capable of performing that function and the tag that you’re using is the gene that does the function, in this instance at least.
17.44 J – OK.
17.45 T – OK, so you’re not interested in having every … all the information that relates to that gene. All you’re interested in is being able to assess the proportions of the types that have that. So you don’t need the full gene then.
17.57 J – Yeah.
17.58 T – So you only need a few hundred base pairs, four/five hundred base pairs, however much you can manage to get. That gives you enough information to produce a tree and then be able to separate the types out in order to assess their relative proportion within your sample.
18.13 J – That is what you’re aiming at.
18.15 T – Well, that’s 16S but that’s just a 16S tree that’s just a … that’s just playing. That’s just a random selection of sixteen S sequences from each of the major clades of the proteobacteria.
18.26 J – That’s a small tree would you do a…
18.28 T – That’s a Eubacterial tree. No, with NirK there’s far less information than for sixteen S.
18.33 J – Ah, right. Something like that is what you’re looking at, is a tree that spreads out like that.
18.40 T – Yeah, I can show you…
18.42 <Mutterings about BLAST searches… Still not finished>
19.16 T – You can produce a tree in this. <Mutterings> It might not be the same example but it will be a… Right, this is a radial tree so these are all the types, which happen to contain Nir K that we know of and all of these ones up here and here and here and here are experimental types. And the first thing that you can notice, I can notice from this, because I’ve seen it before, is that none of our types coincide with known ones. So most of the types that we’re using are producing … nobodies discovered them.
19.53 J – So this tree is generated by the differences in the alignments?
19.56 T – Yeah, well and the similarities.
19.58 J – And the similarities, right.
19.59 T – And the program that you use to generate it takes account of things like saturation (*). So if you have only ‘A’s or ‘T’s or ‘A’s or ‘C’s or whatever at a particular position then that’s … then you can measure that in some degree as a difference. But if you have ‘A’s, ‘T’s, ‘C’s and ‘G’s, then you’ve got a saturated position.
20.20 So you can never tell if you’ve got an ‘A’ or a ‘T’ whether you’ve gone from ‘A’ to ‘T’ or from ‘A’ to ‘C’ to ‘T’, which is obviously a greater distance. So most programs tend to either ignore or down … down weight … the weighting isn’t the same for something that’s saturated than it is for something, which is not saturated, in most models.
20.42 J – Right.
20.44 T – And the reason that this is a radial tree is because we can’t root the tree. Do you understand what that means?
20.52 J – No.
T – So when you look at a lot of the trees … can you pass me that thing…
20.56 J – This?
20.57 T – Yes. So this tree here, this is a far easier way to really see clustering and types and particularly to assess bootstrap values, which is a degree of support for a cluster but it’s rooted. And the reason that we can root this is this is an eighteen S fungal tree but you can root it against these because these are near neighbours to the group you’re studying but they’re not the same thing. Does that make sense?
21.30 So if you were looking at human variability you might root against a chimpanzee or gorilla or a chimpanzee and a gorilla, for arguments sake, as an analogy. So you can do that with a ribosomal gene because everything’s got them, there’s an awful lot of information. But given
that we’re looking at all Nir K things, I don’t know what the next nearest neighbour to that would be. So it’s impossible to root the tree accurately.

21.56 J – Right
T – Yeah? Because ... so you end up with something like ... different. So when you look at, I don’t know if you’ve seen Carl Wolseley papers, for example, where he’s got the diversity of everything. He’s got like plants, animals, fungi, bacteria, archea, they’re all the tree, that’s a radial tree because what do you root a tree of everything against?

22.18 Because there’s nothing that you can possibly use. So if we knew what the ... you know, what the Nir K gene had evolved from then you could use the nearest thing to that as a root and produce a tree that is shaped like that rather that. Does that help?

22.36 J – mmhm. So basically for you BLAST is just simply there as a ... checking that you’re ... got it ... got what you wanted?

22.53 T – In this instance, yeah, I mean ... I would say the most day-to-day use of it is to check sequences that you’ve generated using primers that have been designed against particular genes. But you also use it when you’re trying to design those primers and there will become a time, fairly soon, where we will have to look at unknown function. So we will be pulling out genes by function not in relation to the sequence that you previously had.

23.22 And in that case it will be far more like what Gary and David Marshall use it for, in terms of, Oh, I’ve got this sequence, what is it? And then the ways that we will use that, in that case, will be far more similar to the ways that they use it rather than quick look see, which is essentially what we’re doing.

23.42 J – Right. OK, without an example of BLAST
23.45 T – Pardon?

23.46 J – Without the example, can you ... will you be able to like, off the top of your head, think what you would go through if it was an unknown function search?

23.56 T – Oh, if it was an unknown function search, well I would be looking for clusters of homology to known genes over a significant proportion of the length with good support. If that makes sense?

24.17 J – What kind of support?

24.19 T – I don’t know, I think that’s probably something that you would ... you would look at it and see what you’ve got. Obviously you’re going to be happier if it’s a very high support in the way that what we would get at the top of this would be because I know that this is a Nir K sequence so you would get very, very high support for that. And if you get that with, you know, a range of things that are all the same coming out for them that’d be good.

24.44 J – Do you use more than the one BLAST or do you do it like ... Gary was saying that you look at BLASTn and BLASTx?

24.53 T – Yes. If I was doing what they’re doing then I would do more than one thing and an example of that is that one of the Abertay students that’s working with me was doing ... was doing something ... and some of them looked really strange. So I got him to essentially compare the DNA sequence by translating it and putting it through a different BLAST program, essentially to check it against protein databases to see whether it was still coming up with the right thing.

25.21 So yeah, I mean, I think you would use every tool that you had at your disposal, the whole range of different BLASTs. But also other database searching programs, which look for motifs or profiles and things like that as well. And the first thing that you’d do, again, presumably be a quick BLAST search to see if it flashes up something very rapidly. You have got a malate dehydrogenases or something.

25.48 J – Right.

25.50 T – Something that’s really, really common, there’s a lot of information for it.

25.54 J – Like with Bruce and Gary it would be like clusters, good scores, good e-values, good length in the alignments. Would the date be....

26.06 <Checked progress of BLAST similarity search ... finally done>

26.11 T – Finally, so this is the sort of thing we get because there aren’t very many Nir K sequences on the databases, OK. So obviously the colour ... well, you know all about the colour coding and everything. But that gives you a quick look over but some of these e-values are really quite good and when you start getting down to the bottom they’re very, very poor.

26.32 Which is not surprising when you consider that what it’s comparing it against is something from a human (*) in that instance. But this will be Nir K, that will be Nir K, that will be Nir K and that will be Nir K, at least, very similar sequences I think it’s identical. A lot of things in
Now I would expect that when you get down to this sort of level, what does that actually mean when it's just a tiny little region of homology, like this? Which is Arabinos transport. What does that... I mean, that doesn't really mean anything, it's over such a short region of homology. Presumably that has affected its confidence in itself really. But that presumably may reflect a pocket of... some binding pockets, so maybe in a protein profile or motif search you might discover why it's pulled that out.

I've not really used them to any great length at all. But if, for example, this came up here, it still doesn't really tell you what it is, does it? It just tells you that it's blastobacter denitrotificans, I mean it's just because I know what it is. So, yeah, I would click on the link and, then when the machine got round to it, it would tell you that it's a Nir K gene partial sequence. So that would make me quite happy and I would include in the database, in our own database.

In an unknown function search, when you've looked at the BLAST and you've seen Oh that one looks... that's a nice cluster of those. Would you then follow up into these?

I think that if I put in an unknown sequence and got this out, I don't think I'd be writing a paper that said we have discovered a Nir K gene from the organism or from the community. You would need to, obviously, check the efficiency of this and preferably you would need to over-express your DNA and at least show that it at least has the activity that you're claiming for it before you were certain that it did that or compliment a mutant with it.

This would give you a better idea of maybe what experiment would assist you

Exactly. I mean because you know that you would put it into a Nir K mutant, does it compliment that mutant. You would over express it and see whether you got Nitrite Reductase in the presence of everything else that you would need for arguments sake. Probably not a good example but...

But in this, what would make you say this... these people have worked this out properly, they've not just gone, Oh well, this is similar to something else we've found. What would you look for in the annotations to say well this is good.

Yeah, I see what you mean. Well, the problem with this, I know the paper that this came from. The paper is obviously something you would go and get if you were wanting to check out in that way. All that you know from this, essentially, I would say, is that your sequence has got a high homology to something, which someone else has labelled as Nir K.

So, but with out actually going into the information that would be held within that paper you've got no idea whether they... how confident they are that it's a Nir K. All you know is that they've labelled it as a Nir K.

So the references would be very important?

I think so, yeah, because you need to check exactly what they've done and, in this case, they've done virtually nothing because I know the paper. It's a Molecular Ecology paper so they've been looking at the diversity of a particular sequence type.

If you checked every sequence type that you were using to study the diversity of something then you would never publish anything because you're looking at the diversity of the sequence group. We have, like, twenty-three different types, each of which has a diversity within it, so, I mean, where do you draw the line? I think if I was attempting to seriously assign a function to an unknown sequence I would be far more rigorous in there.

One thing David mentioned was the date on it. Do you have to look out for them being too old...

Yeah, you do. It is something, which needs bearing in mind and probably more so for our sort of... for this sort of work in some ways than a lot of other stuff because what we're is looking at variability within a type. So old sequences tend to be riddled with errors. Just
because of the technology that was used to generate those sequences. So, yes, the older it is the less likely it is to be an error free sequence, for example.

32.03 J - Would you have a cut off point for that one as well?
32.06 T - I think it's just something that I would bear in mind.
32.08 J - Right, so it would just be a sort of flag up, Oh, that's a bit old. It's similar but....
32.13 T - Well, if I had something that was really close to something then it maybe something I would use but I think that you have to accept the limitations to the sequences you have coming out and it depends on how that then affects what your ... what you do afterwards. I mean, in this particular situation, nothing we've got is anything remotely like it so it would be way beyond the area that would likely to be found in our sequence.
32.36 J - So, currently for you, it's just keep looking.
32.38 T - Well, it depends, I mean, if you look at, going back to this <rooted tree> there are relatively small distances between these sequence, which we have and Glomus mossy, for arguments sake. And, in fact, it's sequence identical in a lot of cases but the variation's actually within what we found. But I would be looking at the confidence limit on the clustering on the tree rather than the date on there because that comes into all these things. But yes, I agree with Dave, the date of an entry into a database does affect what you would decide.
33.14 J - OK.
33.15 T - But I think probably more so, probably what would be a more rigorous check, would be, in an unknown gene type search, would be how rigorously have people checked that what they have got is in fact 'A' rather than by sequence homology.
33.31 J - For your, like, checking searches would you not need to be sure that they're correct as well.
33.38 T - Yeah, but I think that you have to allow ... if you've got a whole cluster of everything that you know to be present on the database coming out in the top ten, or whatever. Then I think you would have to assume that that's fairly much been taken care of. But, yeah, you are right within a point. If they happen to label something that isn't a Nir K, Nir K then it's going throw everything out, it does happen.
34.06 J - OK. Um, think of another question ... because you do this, like, not the same, very differently.
34.13 T - Differently. Yeah, it's just a different format ... I think that, you know, if I was doing the same things I would probably give you very similar answers to the ones they have.
34.24 J - The few questions ... the few things you have said about unknown are pretty similar as well.
34.31 T - Yeah, you have to be very careful. It's the same as anything else, you know, I mean, you can't, you know, you're not casting dispersions as to what the persons abilities or anything but you wouldn't necessarily believe someone when they told you that a bus left at ten past four.
34.45 You would go and check the timetable for yourself as to what the persons abilities or anything but the older the bus timetable was, the less likely you'd be to absolutely believe that the bus really was going to come at ten past four.
35.01 J - Also their ability to read bus timetables.
35.03 T - And the ability of the bus company to follow them.
35.06 J - I got the bus time wrong because I thought the bus would come up here at half past and no it was quarter past. So it was a good job we got there early.
35.14 T - Well, there you go. I'll remember not to ask you what time the bus goes.
35.24 J - Yes, so ... what was the other ... oh yeah, contamination was the other thing that cropped up in the other interviews I've done. Is like, you have to be careful of that and Gary pointed out that you could pick it out, like, if some ... if a bit of vector has managed ... you use vector?
35.43 T - Yeah.
35.44 J - If a bit of vector....
35.46 J - Managed to creep into your sequence that you hadn't managed to trim it out properly, you'd get a lot of hits would just be to that. But you should see it in that....
36.00 T - But you should pick that out because it should tell you what that is.
36.04 J - Yeah, in the NCBI...
36.10 T - Trying to remember where they are now.
36.13 J - Yeah, in that diagram at the top. You could ... he showed it appear at one end.
36.17 T - You would expect ... in fact this, I guess if this was the entire length of your sequence here and these two ends, you know, if it finished here and that was red and that was red. Then you would start wondering whether you had vector in.
36.27 J - Yeah.
T - But I wonder as well, one of the differences with what we do is that we’re far more intimate with our sequences. So, because we’re dealing with ... correct me if I’m wrong, but certainly with a lot of the software that Dave is actually producing, which uses BLAST.

So the sort of PHRED and things, they take sequence information directly from the sequencer. The machine screens it for vector and everything else and removes it and for low quality sequence and removes it. And the machine puts that through BLAST. So what you’re left with is this sort of readout for every one of your sequences that’s just come out the other end.

T - Well, we actually do the sequencing slightly different because what we’re interested in is getting error free or relatively error free sequence from defined PCR products that we can use to produce a tree or design a high through-put approach to studying the diversity of natural populations. So we tend to sequence in two strands, we use different sequencing and things, we actually align the two strands together so we do the sequencing in both directions from both ends. And we go along the sequence checking that the machine has correctly called each base pair. Yeah?

J - Yeah.

T - And then we take the sequence from that into a text file and then put it through this and into the alignment through BLAST, sorry and into the alignment program. So we spend much more time looking at it, so it’s far more likely that we would have vector left on that. Yes?

J - Yeah.

T - And when I am BLASTing a sequence or producing an alignment form a sequence, I will remove the primer that we use as well because you fix the sequence with the primer. So you are skewing your data if you don’t remove that.

J - Right.

T - Does that answer that question?

J - Yeah, kinda, I think. I’ll go over that again and think about it.

T - Yeah, sure, that’s fine. Because essentially we spend more time sitting down looking at the sequences on the screen as electropherograms from upstairs. We actually remove all of that so I’m far more confident that we don’t have vector sequences as part of it. PCR chymers are more of a problem. Do you know what they are?

J - Nope. Not heard of them.

T - Sometimes you get artifacts where if you get an incomplete PCR cycle you can ... particularly if you using TACK, which we tend not to use. But you can get an incomplete elongation, OK? So say you get, for arguments sake, half this much of your sequence and then, for some reason, it stops.

And then that will be denatured and will aneal to another sequence, which may not necessarily be of the same sequence type and then be extended. So what you can end up with is a proportion of your sequence from one sequence type and a proportion of your sequence from another. So it’s chymeric. And that can cause problems.

J - OK.

T - But there are ways of checking that out. But again, it rather depends ... it will be very rare. So you tend to lose them anyway. . . .

J - So you don’t tend to get that much contamination.

T - With vector.

J - With vector.

T - Quite often you get things that have absolutely no relation to what you were expecting and you just throw them out at the B BLAST stage.

J - Right. So if you do your BLAST of a particular sequence and you’re looking for.

T - And it comes up with something that is completely unconnected to Nir K, yeah.

J - Then you go, That’s not what I’m looking for, goodbye.

T - Yes. And there’s a whole range of steps, I mean, because you’ll look at your sequence far more, we do things like make sure you’ve got the right primers at each ends and things like that, which is, obviously, a way of screening out stuff. But, even then, you will get some contaminate sequences, which are either the wrong length and all the rest of it. But when you BLAST them, it will not come up Nir K, it will come up something completely bizarre like ... not that because that’s not Nir K ... like that. That’s not a good example either but ... see we’re just too good <Couldn’t find a decent example>. This complete genome again, and lets get down to something ... yeah, like this. Hydrogenase.

J - Yeah.
40.39 T – And you would just assume that that’s.... But, again, that’s a rare thing that will come up. Unless your PCR is rubbish. If you’re getting more of these than Nir K then you haven’t designed your primers very well and you need to go back and start again. Because obviously your primers aren’t very specific.

40.54 J – Right, OK. I think, because you do things so differently.

41.00 T – Well I did tell you that. But I hope that I’ve validly....

41.06 J – Yeah. I think ... that’s all I can think to go over just now.

41.12 T – Yeah, feel free to get back in touch.

41.14 J – I will. If I go over that data, make sure I understand it and then if I think of anything else.

41.21 T – Yeah, just get back in touch.

41.23 J – E-mail you the questions or ask you if we can arrange another meeting.

41.26 T – Yeah, OK.

41.29 J – That was quite good.

41.30 T – Hope it helps.

41.33 <Tape stopped. End of interview>

Interview starts at 41:33 on tape>

0:41.33 J – There we go <Chairs movement covers speech>. So if you just want to go through what you look at in the BLAST.

0:41.40 M – Right, OK. I’d like to tell you the story about Umbra viruses. And with Umbra viruses there is a region in virus genome, which codes for two proteins, which are not in frame. You know, overlapping amino acid sequences. But they are encoded by the same nucleotide sequence. And one of these proteins is a cell to cell movement protein, which has similar proteins ... functionally similar proteins in other virus groups.

0:42.23 Another protein does not have similar proteins in other virus groups and also when we searched databases we cannot find very similar proteins in all viral or non-viral databases. So, I am going to show you two BLAST searches for this protein, which is cell to cell movement protein, which has similar proteins and with the Orf three protein, which does not have obvious similar proteins. So this is the BLAST with the cell to cell movement protein.

0:43.12 This is the sequence, and we have a lot if different proteins and I can understand from this list that many of these proteins have similar functions. So if proteins have similar functions then I can predict that they may have similar sequences and when we looked at the comparison, for example, here I can see long stretches of similar amino acid sequences. And I can think about functions, which can be coded by these sequences.

0:44.13 And then if we go from here to here we see that similarity goes down, however, functions are still the same. So we can see some very important amino acids, probably, which are responsible for these functions. And then if we do, you know, the alignment with different proteins we can find what residues, which are absolutely required for this function. Well with this particular protein it’s not a big problem because the function is known and when we (*) proteins with the same function then it’s clear, OK.

0:45.13 J – So you would use it just for confirmation?

0:45.17 M – Well, for confirmation and for relations, you know taxonomic relations between different virus groups.

0:45.26 J – Right.

0:45.27 M – And another thing ... well the general function is the same, is a cell-to-cell movement. However, different viruses use different strategies for ... to express the same function. All of them must move from cell-to-cell but they do that in different ways. So, well, there may be a big difference ... a comparison between different amino acid sequences will give us a key to understand what’s the general thing is common to all viruses and what’s different, OK?

0:46.11 J – Yeah.

0:46.12 M – So, it’s more or less clear with proteins with known functions and the problem is with proteins with unknown function and you can see here an example. So, we inserted a sequence of the Orf3 protein and in parallel we looked at functions of this protein and functions are not obvious because this an unusual virus group and the ... because this is the only virus group which does not form virus particles.

0:47.03 So, this is a unique virus group and if we don’t have virus particles we can predict that some properties of this virus should be different from all others because virus particles are required
for movement of virus, protection of viral RNA and so on. So these viruses must compensate for lack of the capsule protein, for the lack of virus particles. And the only unknown protein with unknown function is the Orf three protein.

0:47.41 So, from one hand we looked at properties of this protein and on the other hand we were trying to find some proteins, which can give us a key, you know, for functions of this protein. And when we looked at the BLAST, what we found, we used SWISSPROT sequences, and we got here these sequences, Spectral Alpha chain, breakpoint cluster region, and cyclic AMF dependent transcription factor, OK. If you look at the alignments we ... well, the scores are very low so there are no obvious similar regions, OK?

0:48.52 However, there might be that these similarities are really important. But how we can differentiate between important similarities and non-important, well, I don’t (*) ... well I happy because ... well this is a protein, which is encoded by viruses belonging to the same group. So I have sequences of this protein for relatives, you know, for groundnut rosette virus, pea evation mosaic virus, tobacco mottle virus and carrot mottle mimic virus.

0:49.32 So, I have this alignment for all this proteins, so I don’t know which residues are conservatives and, for example, in this particular case, I can find ... well, even with the, you know, with relative, the similarity is not very high. However, I can select two areas, which contain similar conservative residues. The first one is Arginine rich domain, another one Leucine rich domain and when we looked at properties of this protein; we found that this protein can bind RNA.

0:50.30 So, prediction should be that this protein should contain basic charged domain, so Arginine rich domain may serve as an RNA binding domain. So, I think it may be important. Another domain is Leucine rich domain. It’s known that Leucines are often required for protein-protein interaction and we do know that this protein can form polygermers, can interact with ... well, can self-interact, you know, form dimers, trimers and so on.

0:51.18 So, I think that this domain should be also important. Some other properties, nuclear localisation also ... well, in comparison with other proteins with similar properties predict that these two domains are important. Well if I look at the ... this search I can not find any similarities with these proteins in this particular areas which should be important.

0:51.54 So from this search I can not say anything because ... well if I see the similarity here and then I look at the particular area in this protein. I cannot find similar sequence in the protein, which has the same functions from another member of the same virus group. So, I think is just, you know, happened by occasion, or what ever. So it’s ... I cannot see any sense in this area. So, in this particular case I cannot get any useful information from BLAST.

0:52.54 J - Right. So you ... on this particular case, no one else has really done...

0:53.03 M - No.

0:53.04 J - Any other work. So you would just look at that and go. Well there’s no help here and you have to do the work yourself.

0:53.09 M - Right, yeah.

0:53.10 J - But on other cases could you ... unlike when you’ve done ones where you already know the function you’re finding the similar, so if you hadn’t known the function you would say Oh, well this is a useful search because it’s coming up with several the same.

0:53.26 M - Yep. This is at this stage of our research but when we started, I started with the new virus, I knew just the name of this virus from biological studies, groundnut rosette virus. My task was to determine nucleotide sequence of it’s genome. And I did that, I got some sequences and then it was necessary to find any similar sequences and I compared nucleotide sequences first and then ... well, translated areas, of course translated areas from nucleotide sequences without open reading frames.

0:54.12 And I also used BLAST and then I found similar proteins with very high score and it was very useful because I found similar members of similar virus groups but it was a case only for proteins, which have, you know, high similarity, identity. This is movement protein or this RNA polymerase so functions are known, many viruses code similar, functionally similar proteins.

0:54.59 So it works but when you are working with a protein, which does not have, you know the ... if the protein has a small area, small region which is functionally important then it is very difficult to find ... to identify this function by comparison with other proteins.

0:55.35 J - So if only the functional area is conserved, it’s harder to find. Whereas, if a lot of the protein is conserved it’s a lot easier to find. Right, that makes sense.

0:55.50 M - Yeah, especially when with the (*) nucleotide sequence when we compared with a lot of viruses we found very similar viruses, which are members of same virus groups. Then we can
start to work with sequences, which are not very similar, OK, and one of these here, this is a ... well if we look at other regions of the genome, the similarity’s much, much higher. So we found similarity in these areas and then we are working with these areas.

0:56.34 So we can have this alignment and the … if I do the BLAST, we need to do BLAST search for all this numbers and look at all similarities, which correspond to all members of the same group but not to one of them because function is the same. And in this case what I would prefer is to find, you know, conservative domain and then just to understand. OK, Arginine region, what it may be.

0:57.14 I know from the literature that many Arginines may be responsible for RNA binding so I’m looking at known RNA binding proteins. And even here when I see these three Arginines … somewhere here … all right, here … OK this is near this area but ... well, not other residues are similar. So I think that this area may be just responsible for, you know, interaction with either nucleic acid or with AMP, alright, this is just nucleotide and the … it might be that this is just because of basic charge, can react with this area.

0:58.16 And this is very common thing but maybe does not relate to function of this protein. Well, so I’m looking at all proteins, which have similar functions and then I’m trying … maybe I’m not right, you know, but this just my word of actions. So I’m going from known function to sequence.

0:58.50 J  -  Right, so you’re using a known function from another part of the genome to find other family members.

0:58.58 M  -  Exactly.

0:58.59 J  -  And then you’re using that to try and work out the function of an unknown sequence.

0:59.05 M  -  Exactly.

0:59.07 J  -  Right, and when … and you’re finding some similarities between them, in that region, but when you’re doing the search you’re not finding anything really helpful at all because….

0:59.22 M  -  It happens with this particular sequence.

0:59.24 J  -  That particular one because you’re probably the first to look at it.

0:59.30 M  -  Yeah, but … well, when we compare sequences of related proteins, of proteins from related members of the group then I can see this conservative domain and I can predict this may play a roll in RNA binding or nuclear localisation. So what I should do, I should make mutagenesis in this area and to look at properties of new proteins, of modified proteins.

1:00.12 And also I’ll look at similar proteins with similar function RNA binding protein and common thing is many Arginine or Lysine residues and that’s it and well … but using BLAST it’s very difficult to say that. So we need to play with two parameters, with function, with known function, and sequence, in those cases where we don’t have apparent sequence similarity.

1:00.57 J  -  OK, so … well, when you go through a BLAST, what … well, obviously you’re looking for similar regions in the sequence. So, is it the alignments that are the most important part or, I mean, the score would be important as well?

1:01.17 M  -  For me alignment is most important. Well, sometimes … well, score as well but I like to have a look at alignment because when … it’s more obvious. OK, if you see something like this then … well it should be important and then if you see the same … OK the same, right, or something similar then you see, yeah, this is important.

1:02.02 J  -  Do you ever look at the nucleotide similarities or do you always go … (*)

1:02.07 M  -  No, I look, of course, at nucleotide similarities as well but with nucleotide similarities, well, it’s … when you make your sequence first probably, well you can use both comparisons between nucleotide sequences and amino acid sequences. Why, because if you have a sequence you don’t know if this sequence is translatable or non-translatable.

1:02.42 It might be because, for example, in this particular virus group genome contains very long sequences, which are not translatable but might be important for some functions, for protein RNA recognition, for example, or for RNA host component recognition. So more over, you know, there are some sub viral RNAs, for example, satellite RNAs, which do not code any proteins.

1:03.24 And in this particular case I would not … well, of course I would try to compare amino acids translatable sequences but, you know, it maybe not important at all. So, in some cases, it is very good to compare both because if you don’t know that this is translatable region, you need to do both searches.

1:03.55 J  -  But you’d go through the same process with the nucleotide, you’d look at the alignments and the scores. And if in the descriptions you’ve got same function that would … would that be significant, if it was an unknown … was unknown and you had the same, the same, the same, that would be good?
M - Yes. At least, for example, if we're talking about satellite RNAs and if I have some more or less similar sequences then I read that this is sequence from satellite RNA. OK, I know that this is, yeah, a very important sequence.

J - Do you look at the annotation files in the databases at all?

M - Of course.

J - What would you do when you're looking through that, I mean, would you just read down and go Oh that's a good bit and that's a good bit. What would you decide would be an important part of that.

M - OK, well, of course I'll look through the list of the names of the proteins which are in the list and then ... well, it's just a, you know, feeling that this protein, my protein, is important for long distance movement and it localises to nuclide, for example.

And I just ... I don't know literature for all these proteins which are in the list but I can just, you know, feel that Oh, it might be important, then I'm going to description of these protein. And then I'm trying to understand more and more and maybe I need some more literature to look at functional similarities, what happens in....

J - So, if you're going into the database you look at literature references that they might have or Medline references there might be.

M - Yeah, sometimes it's not necessary, sometimes when I can find the similar function in the list because, well, for example, if I'm doing this movement protein I can ... how I understood that this is movement protein, I found that many movement proteins are similar. So I know that, but if I have, for example, this is... then I know this is movement protein, OK, and then I see some protein ... well, I don't know, RNA-dependant RNA polymerase and I have more or less long sequence, which is identical.

Well, it might be interesting, so I need to understand how this RNA-dependant RNA polymerase does work. So I need to go to the literature, to references, to look at possible mechanisms of action of this particular protein. And if I see some stages, I can predict that these stages may be common for both proteins then I can come closer to the mechanism of action of my protein.

J - So if you've just got, maybe, a very good match to a single function, rather than several in your list. As well as ... would you ... so that. You would look at general literature and look for that function, rather than just look for a reference in the annotations, you would look at all literature for that particular function? And see if you can find similar situations to what you've got your virus in and that.

M - I would do both.

J - Right.

M - You know, first I would look at general literature and then, sometimes, you know; well, even sequences which in the end of the list give some ... not long, you know, sequences may give, may not but may give some interesting features. So, if I am working with this protein then I'm trying to get as much information as possible. And if I am very pessimistic with this protein now it's not because of my first search. But this is because my long ... well, I did it many times and not only me but all my colleagues here.

And we tried to fish any similarities, we were unable, that's why I'm pessimistic. But when I started I tried to look at all proteins to get as much information as possible. This is just a specific ... very specific case but usually with other proteins, for example, with proteins that are common for different viruses, like polymerases or just normal cell-to-cell movement proteins, situation is much easier. Then you can get much information.

And we tried to fish any similarities, we were unable, that's why I'm pessimistic. But when I started I tried to look at all proteins to get as much information as possible. This is just a specific ... very specific case but usually with other proteins, for example, with proteins that are common for different viruses, like polymerases or just normal cell-to-cell movement proteins, situation is much easier. Then you can get much information.

J - OK, so you just go through, you look at the alignments, you look for references, you look in the literature and.... I've told before date was important. Do you look at ... ever look at the date on the annotations to see how old that particular submission is?

M - No, to be honest, no. if something would be interesting, yeah, but not really not.

J - Yeah, any information would be interesting for you.

M - Yeah. Sometimes you can get some, even some mistakes, you know, because sometimes people put into the database some data which are wrong and then when you do BLAST search you can get some similarities. Oh, fantastic, it's very interesting, for example, I worked with one sequence, with plant virus. And I searched for database and I found ... I tried to compare nucleotide sequences and I found long region, about seventy nucleotides, which are completely ... which are identical to my sequence.

And then I looked at annotation, I found that this sequence is originated from HIV, the Human Immunodeficiency Virus. Oh, it's fantastic, I have sequence in plant virus, which is
completely the same. It was fantastic result and then, of course, I took this particular sequence and I looked through all databases and I found exactly the same sequence in bacterial plasmids.

1:11.58 And, you know, when you’re doing sequence, you insert you RNAs in bacterial plasmids just to make sequence. And somebody recognised this sequence in bacterial plasmid as a sequence of HIV RNA and inserted in database, OK. And I found that I have the same sequence and it’s a ... so what I would ... but this is question to databases because, you know, it’s necessary ... how they do that, they just accept sequences you send with the annotation.

1:12.42 Nobody checked your sequences properly and if you insert mistake then ... for example, I found that, so how to (*) because other people can also have the similar problem. What to do, well, OK I check everything, another excited person can, you know, just publish this as quickly as possible.

1:13.11 J - So even if you do find something you have to be careful.
1:13.15 M - Very careful, yes.
1:13.17 J - You need to check to make sure it’s OK.
1:13.19 M - Yes, because, well, the annotation from database may be wrong.
1:13.30 J - Do you usually use the same plasmids, I mean, do most labs use the same plasmids for cloning?
1:13.37 M - Yes, similar ... well, all these plasmids belong to the same type, to several types, and most ... the most of them have similar ends close to inserted sequence.

1:13.55 J - So you would be able to ... if you had the plasmid sequence with ... as a, sort of ... you could check yourself?
1:14.03 M - Exactly.
1:14.04 J - Is that of that or....
1:14.06 M - Yes, so when you ... well, the rule should be, you have your sequence and first thing you need to run this through vector sequences ... plasmid sequences to remove all non-viral sequence. But not always people do that because, you know, what may happen during cloning some plasmid sequences may be inserted into ... well there might, you know, just be mistakes of cloning. Well but it’s necessary to, you know....

1:14.47 J - OK then. I mean, you’ve pretty much said quite similar to what the other people have said as well, how you use that, so that’s quite good. And you’ve also ... could I get you to write down the names of those viruses so I can get the correct spellings.
1:15.02 M - Yeah, sure.
1:15.05 J - I have trouble with that sometimes.
1:15.08 M - Yep, not only you.
1:15.17 <Michael writes down virus names, mutters some comments about them (already mentioned)>

1:17.42 J - OK, that’s great. So, if I ... I’ll go over this data and then if I have any other questions I’ll be able to e-mail you.
1:17.51 M - Sure, but not during the next two weeks because I’m away on holiday.
1:17.58 J - That’s fair enough, I mean, it will take me a while to transcribe this and analyse it anyway, so that’ll be fine.
1:18.08 M - OK.
1:18.10 <Tape turned off. End of interview>


00.03 J - Right well. First I wanted to just ... I’ve got my little structure to make sure I don’t forget anything. Um. Basically just to recap so you remem ... if you remember what we were doing, it was a ...

00.17 G - Vaguely, yes.
00.19 J - trying to find a way to automate the analysis of the BLAST results and put them into order of relevance rather then just order of similarity, though, of course similarity is important and from that ... I’ve got this wonderful little tree
<Shows BLASTtree.ppt>

Basically it was this ... this ... I did this from a sort of amalagamation of all of them ... Basically ... looking at ... what was looking at ... it was ... well you look at more than just one BLAST ... And then there was looking at the scores and although there’s arrows on here that’s not actually the order they were done in, that was just, well there’s these things as well. So there was clumps ... there’s a written list as well here. Sort of ... do you have ... these other clumps is ... do you have multiple instances of species or enzyme names and things like that.
01.24 G - OK
01.25 J - Ah, how close is the species
01.27 G - yep
01.29 J - And ah ... is there ... sort of looking for conserved regions and the ones "is there mRNA/ EST" is it sort of from a coding region is what I get told there ... Then there's clumps in the alignments and similarities between these BLASTs and then from that you sort of go onto the annotations to see if there's any further information. I assume with your nodding I'm not saying anything that's comp
01.59 G - no, that's making sense, yes
02.02 J - completely out of the ... From that what we then went on to do was, well, we said, well we said "well" automating all of that is, well it's too big a project for starters for just a PhD and then there's other things like, Is there a particular name there is just checking against a glossary so that really isn't as interesting. But we looked at this one <points at 'is there clumping in the alignments'> and ... decided well that ... that one's most interesting because to reorder this data in relevance we need to put a value to that so that it can change the order so we ... we were trying to measure the clumpiness
02.44 G - OK
02.45 J - For want of a better word, which is where my project has not centred for the last two years and I've been doing various ... learning how to do stats again ... Over the last two years and doing various measures. So I mean that's ... that's a nice little diagram of what we foresaw the program would do.
03.05 G - <sneezes>
03.06 J - Bless you. So you'd get you're ... you're various questions and then you'd get the ... a ***** score and then that would rearrange the list based on similarity ... and all these other various things
03.24 G - OK
03.25 J - and get a new list that would be in a different order that would hopefully bring more relevant hits higher up the list. So that's basically what ... what I've been doing and I've got ... got another ... number of possible measures which I now need to test ... to make sure they're actually working ... And for that I generated a synthetic dataset, which basically just consists of ones and zeros sort of matches and mismatches. I have these wonderful little cards here
04.00 G - nice colour
04.01 J - yes so like a white would be a mismatch and a black
04.05 G - is a match
04.06 J - is a ... is a match and what I would ... if you're willing ... Is if I could get you to order ... I've got ten of these things ... if I could get you to order them as the ones you would find ... feel well as the most clumpiest and least clumpiest.
04.22 G - Oh, OK
04.23 J - as ... and ... which would be the most interesting to least interesting and then also ... if whilst you're doing that ... describe why
04.35 G - Explain why
04.36 J - you're putting a particular one above another. Get a pen ... I'm kinda spreading out across your office here, sorry about that.
04.48 G - That's all right. I'll start them out in alphabetical order to start with then I can see what they all giving me. A, B, C, D, E, F, G, H, I, J. I'm guessing that there are ... approximately the same number of blacks on each one.
05.23 J - yes, it ... the inten ... we figured keep everything the same bar one thing so the lengths are all the same and the intensity of matches is possibly not exactly the same but virtually so there's not very ... there's not a lot of difference. So that we can actually then see if its looking at ... the clumps rather than sort of distance that the a ... just the actual number.
05.48 G - OK
05.48 J - and just these little white lines are actually the lines between the blacks so you can see where they ...
05.53 G - yep, of course. Well, the first one that strikes me as different is this one black, white, black, white, black, white all the way along ... I'm trying to think if I see that in reality and I'm not sure that I do. ••
06.21 J - well, the synthetic data does ... doesn't mimic what you get from BLAST ...
06.30 G - bottom of this, J, ... is looking much more typical of a good match and there's some difficult ones in the middle. Well you have clumps of blacks and you have within a clump a lot
of blacks next to each other. These are more typical of what we get, the fact you’ve got a
bunch of whites together is of little significance. Usually ... there’s a qualifier ...
07.22 Of course what this one may indicate, I’ll use this one as an example, is that there may be
active domains within the sequence ... and that’s why you’ve got one, two or three ... would
almost imply two or three clumps here ... so that could be a candidate for a very good match or
it could be a candidate for a family member but you’ve not quite got it right.
08.11 We often get poor matches at three prime ends of sequences ... Into three prime ends? Those
are variable regions and therefore we expect them to have lots of not matches and not so
important. Usually you might expect a be ... a better at the five prime end that one ...
08.30 J – Yeah, those sequences were actually generated using Hidden Markov model so they’ve got
... they’re nothing to do ... they were not generated at all using BLAST. It was just ...
08.42 G – That’s fine
08.43 J – can we create a very clumpy and can we create a not very clumpy ... so ... I mean, I do
want to go on and generate a more realistic dataset using BLAST but this is ... this was my first
... this is my first one
09.02 G – That’s fine
09.03 J – So to speak
09.09 G – Not a lot to choose between some of the others ... that’s got ... so for ... I’ve picked out
three here, G, H and I. I think there’s little to choose between those three.
09.44 They seem to have similar bunches of blacks together; there are regions of white though I
notice in G there’s a bigger region of white that there is in I so I suppose ... I might come out
as a slightly better match because we don’t have such a big a region of white. There’s not
much in it really ... there is the other feature of matches ... are these BLASTns or BLASTxs?
10.29 J – These are actually proteins ... these are actually proteins ... but ... I know we’ve been very
bad, we have just changed the positives into matches for simplis ... sim ...
10.40 G – simplistic ... right, then if it’s proteins you should be considering families
10.44 J – Yeah
10.47 G – and I have a list up on the wall ... round the corner, if you want to look at it in a minute.
So you’ve got ... out of my list, you’ve got three hydrophobic amino acids; two, three, four,
five, six, seven, eight hydrophobics. So it’s possible that even if it’s not the right amino acid, if
it’s another member of the same family ... That is also good quality information that it’s ... it’s
closely related
11.21 J – Yeah
11.22 G – So, if in some of these whites it wasn’t the right amino acid but it was ... a member of the
same family this is additional information that ... you’ve a good match. Some like ... isoleucine and valine are very interchangeable ... and it doesn’t make any difference.
11.42 J – Would they ... would they not be put in as positives on ... on the BLAST?
11.52 G – Not necessarily
11.54 J – Right
11.55 G – Depends on the software that you’re using. Sometimes they just say yes/no and it’s got to
be the right one.
11.59 J – Right
12.00 G – And others they have this coloured information which is available to suggest other family
members. If you can show the match according to ... the classification, hydrophobic,
hydrophilic, basic, acidic ... then if it’s the right family member that is also good quality
information that it is something
12.18 J – Right
12.20 G – Because these ... all family members give the structure ... have a similar structure so the
protein has the same shape at the end of the day
12.27 J – Right
12.28 G – It’s the shape of the protein, which is critical. So if you put something on the side arm and
it suddenly goes down like that, it’s very different from when it’s like that.
12.36 J – Yeah
12.36 G – So you can change that amino acid and if it still gives you that you’re OK. Do you really
insist that I put these in order?
12.51 J – Well I as close ...
12.51 G – because it isn’t easy
12.54 J – as close as ... as ... as possible and ...
13.30 G – Oh God. I’m going to start counting these in a minute. Am I allowed to do that? One, two,
three, four, five, six, one, two, three, four, five, six...
13.40 J – I mean this ... this is what we’re trying to ... mimic, is how ... how would you rate one above the other.

13.51 G – A longer sequence of communality is one feature. So if we can get ... here we have some strong runs under J ... that’s looking good, but the next group down, one, two, three, four, five, six, seven, eight, one, two, three, four, five, six, seven, eight ... we’ve got bunches G and I have got eights in ... is looking good, especially if you had then separated by one that’s a mismatch but would be the same family member.

14.25 J – Yeah

14.26 G – It, basically, would give you confidence that the run is much longer than that. Um ... if it’s a totally contrasting member then maybe, but you’ll have the odd mismatch. Um ... up here I’m looking for little indication of long runs. One, two, three, four, five, six. One, two, three, four, five. One, two, three, four, five. Five OK, I’m happy with those. Right, I’m getting there.

16.33 I think ... J is looking quite good for a long series in the middle but I’m a little bit bothered by the fact that ... the left hand end isn’t matching anything. I’ll stick that down as the five prime end ...

16.55 G – Well, like I said, this didn’t actually come from BLAST, this was made up.

17.00 J – I know, I know. But I’m trying to imagine it is a BLAST, so there you go.

17.02 G – Yeah

17.05 G – So I might be worried that, oh, that’s actually ... a related gene that’s n ... that’s looking very good, but I would want to start checking that one out for ... let me see, an enzyme ... an enzymic reaction but there’s something else in there that’s not matching or in there that’s not matching but it’s looking pretty good. I like this group because they’ve got matches that ...

17.32 J – F, H, I and G?

17.34 G – F, H, I, G. Um ... so I and G have got, I think it’s groups of seven, they’ve the longest groups of black, I like that ...

17.45 J – Why would you put I above G or I worse than ... ?

17.50 G – Um ... well again, I’m just looking at it thinking if that was a five prime end I’ve got a match right at the five prime end, I like that ... it’s not critical. Sometimes at the five prime end we have sequences, which ... direct the protein to certain parts of the cell. It’s not critical but it’s marginal ... there’s not much in it between those two, but that’s the reason I’ve gone for that.

18.19 J – Right

18.20 G – Over the ... the significance of the whites ... it’s more whites there, that’s not the problem really, the blacks I regard as more important.

18.30 J – Right

18.31 G – And maybe the position of the blacks a little bit. And then, F and H I think we have one, two, three, four ... oh ... we’ve got sevens there. One, two, seven. But A and D we’ve got sixes I think, fives, one, two, three, four, five, six. I don’t think there’s much in it there either. I think A and D there’s not much in it. Just trying to fit it. one, two, three, four, five, six, one, two, three, four. Oh, it’s a dirty cheat this lot, isn’t it. You’ve got a couple of runs as well in A, one, two, three, four, five, separated by only two whites. So I could almost mentally add those together and say there’s a potential for a much longer run there.

19.35 So it’s the relative position of the blocks of blacks to each other, which is playing some role. I’m adding those up and saying there’s a big island of similarity ... If it was at opposite ends then it’s just a block on it’s own. Yeah, ... in C we have one, two, three ... well a block of five but it’s at the end but it’s next to lots of whites so you can’t really add it other blacks, it doesn’t look right. E is not giving as long runs of blacks and B definitely isn’t ... Now B in fact ... it’s not quite typical of BLAST but what we have seen occasionally is more of a random matching, which is poor throughout the whole sequence.

20.36 Maybe every five or six would match and that I would consider to be a poor match. And it’s almost it’s matching by random ... and I think once or twice, I suspect, people have made the mistake of just because you have a long sequence, which reacts ... which matches slightly with a long sequence from something else, that this is in fact a good match but there are too many whites

21.09 G – Yeah

21.10 J – And, therefore, it just isn’t right.

21.12 G – Yeah

21.14 G – I have seen things like that. So that is moving towards that e ... that aspect. So ... I don’t like spaced matches like that, it’s not right, something’s wrong. You need clumps. I suppose
mem ... image wise, I'm looking at ... I look at J ... I can add up all the blacks as one ... well, half of them, are one sequence, just ignore a few of the whites and say I have a very long match here don't I?

21.49 I'm allowed one or two mismatches, not an issue; we have a very long match in this sequence. That's nice. So it's the length of the match, which is also quite important ... but then you would expect that you are not getting another long length of whites because that would imply a total mismatch and you'd expect a lot of blacks to match where we have whites then the occasional match just to confirm that we're hanging in there.

22.21 J - Yeah
22.24 G - If it's all mismatch then it's wrong
22.26 J - Yeah
22.27 G - Is that OK?
22.28 J - Yeah, I think ... obviously, as well, in BLAST you'd probably hope to have a longer ... because these are only fifty ... they're only fifty long coz I didn't want to really ...

22.38 G - No, it gets difficult
22.40 J - over over over over flow
22.41 G - Yeah, I mean er ...
22.42 J - with data.

22.44 G - In reality we get EST, short sequences initially and say that's exciting, that looks possible now lets get the rest of the gene out just to confirm it. You are right if we have a short sequence, then we can't guarantee that we've got the identification right. What we used to do from a short sequence ... it gives us great confidence to go chasing it and then you'd expect the sequence to keep going.

23.11 J - Yeah
23.16 G - We've had more successes than failures that way, pretty much, so once you get a match with a short distance it tends to be good.

23.24 J - Yeah, I mean at ... at ... at the moment, unfortunately, the measures aren't complex enough to deal with the single gaps. I mean, I have thought of a couple of ways of dealing with that by basically pretending ... getting the program to look for just the single match and then pretending it's not there ... I'm going to be writing prototypes as well, it will hopefully include that but might not ... I might not get it that far ... cause I've a ... PhDs coming to an end now. If I get a prototype done up, that would just find the large ... you know ... measure this by just the clumps it's got in it. Would you be willing to see if it's any use?

24.12 G - I could try it out
24.13 J - Yeah
24.14 G - and see if we
24.16 J - I mean, obviously, from what you're saying the being ... ability to ignore these single mismatches in these groups would be a good idea

24.24 G - Yes
24.26 J - Whether I can actually get the program to do that in the time allowed
24.31 G - Is a problem. That's right. The other thing of course if is ... what you've not ... what we've not yet discussed is that occasionally ... we get genes with characteristic motifs

24.46 J - Yeah, that ...
24.46 G - in them and I it's got the motif, the match is correct; end of story and it can be a short motif. Now you get these with ... oh ... hang on ... I can't remember at the moment ... I've forgotten the name of it, it'll come back in a minute ... but certainly we get these ... a different group, which are transcription factors, which are wrkya, W-R-K-Y-A in them ... and if got these amino acids and some of the others present, you know you've got a wrkya ...

25.24 J - Yeah
25.36 G - And in that case you can have quite separate ... occasionally you can have blacks which are quite a long way away and as long as they are the right distance, the right motif and I think some of the search engines on the internet do these sort of searches and that can also be quite good.

25.53 J - I ha ... I have looked into the motifs but I figure in that little tree that would be another thing to look for

26.01 G - Yeah
26.02 J - So ... so like looking for how close is the species, how clo ... you know, does it have any of these motifs, does ... you get ... the other thing that we were thinking would go in the further work is OK we've found this clump, does it appear in the other sequences as well? Sort
of like do the multiple alignments it be ... it reappears, which would give ... if I remember that gives greater confidence that ...

26.29 G – Yes, it’s true ... we do sometimes see that of ... we’ve got a match and it just ... is the match down the whole way, it’s the same amino acids it’s not

26.38 J – Yeah

26.39 G – Yeah, and dat ... those are intriguing as to the significance of it is ...

26.48 J – So we ... we figure tho ... those are the other features and this is ... but we’re just concentrating on the one ... just now, which is the level of ... I mean, obviously on the really good matches where you just have a whole string of matches with only a couple of mismatches, this wouldn’t really work on because that would just be considered extremely clumpy. So that was the ... another thing is where is that threshold under which you go well we actually need to start measuring that but ...

27.21 G – Yeah I mean, final confirmation can only come from actual experimentation

27.26 J – Yeah

27.29 G – It’s been pointed out with a few things like protein kinases that ... at the end of the day you have to do the wet science to confirm the identification ... and the best we can do is provide pointers

27.46 J – It’s hoped this would give us a slightly better pointer ... Though not this ... this is the thing I worry about, is like this on it’s own it might be jus ... you know, it might bring something further up that might have been missed further down.

28.01 G – It might. I mean, one of the problems we have is there’s lots of genes out there and we can only focus on a small number, which ones look the most exciting and if we can put a tentative identification of some things it might suddenly leap out as this is potentially a signalling gene and that’s exciting, let’s put some more effort into that one

28.21 J – Yeah

28.24 G – Yeah, we’re faced with dozens and dozens and dozens of unknowns. Find, potentially the most exciting genes in amongst those unknowns and occasionally if you just flag up one or two that’s really hits, so it is worth doing.

28.46 J – So is there anything you want to add to how you got that or ...

28.49 G – No, that’s fine. You OK with that?

28.55 J – Yes ... that ... that’s fine ... I’m sure there was something else I wanted to ask you while you were going through that but I didn’t want to interrupt and I’ve forgotten now, which worries me. I did ... I did actually mean to bring ... because like I say I’m ... I’m testing some of measures at the moment and I’d actually got them to order this and I actually meant to bring it with me but the ... I’ve got it on a bit of paper here but, unfortunately, it’s got the wrong codes on it

29.26 G – OK

29.27 J – it’s got the codes used ... I used for the ... that made these cause these sort of define the ... the probabilities I used to make these and it’s the wrong codes. So I apologise for that, I’ve not been very organised today and I don’t know. Not the best of ... not the best of days. But ... I mean that ... that gives me some other ideas on how my measure ... my measure things might need to be tweaked, especially at trying to bring in those single ... I mean, how ... how big a gap would you consider ... would not effect whether you’d have a clump? So you’ve got this ... this one here and it’s got a gap of two but you’re still quite happy with that ... how big a gap would need to be in there before you’d go oh well that’s two separate? Or would that depend?

29.31 G – OK with four/five, maybe

29.35 J – Four/five

30.40 G – but I’m not really ... well, I’d prefer less. Four

30.54 J – but you’d need to have good clumps on either side

31.01 G – Then I’d be looking quite closely, I’d expect these whites to be similar family members, I would look at that aspect.

31.09 J – Right

31.10 G – and if they’re totally different I’d say hmmm, don’t know why, OK, maybe, maybe not. But don’t ... what we are capable doing is saying ... using the word ‘like’ for identification purposes, it’s this ‘like’... and we do that in vast numbers of cases

31.29 J – Right

31.32 G – So you need a very, very high match to say ‘is’, this gene ‘is’ that and once you start getting mismatches say ‘like’ or family member ... and I’ve seen the identification of these change quite dramatically ... sometimes as well. Some authors have put some sort of family and you just look it up and think it’s not ... not at all ... I don’t know why. I think,
occasionally, some people ... you do a BLAST search ... and the name that they’re allocating to it is whatever is at the of that list
32.07 J – So they don’t do any follow up at all
32.10 G – No, they’re using it non-critically and the match is quite poor
32.13 J – Because that… that was one thing mentioned in the previous interviews. I don’t know if you mentioned it as well, is the … the worry of Chinese whispers
32.22 G – Totally, yes. Chinese whispers is a big problem
32.27 J – Well, I think that’s … that’s scribbled down on my sheet somewhere. Yeah, is there … you know … care of Chinese whispers.
32.43 G – Yeah ... I think … in some ways round that is the wet science
32.53 J – Yeah
32.55 G – But we have … we’ve got a big database of genes and … I take names from the literature and … occasionally you see a name given to a gene and it’s almost got two ends to it, it’s like a protein kinase or a lectine. And you think oh, OK these are two different families. Clearly it’s got similarities to a protein kinase, it’s got similarities to a lectine, which is, it’s either that end or that end of the matching sequence. It can’t be both … I’m thinking actually of there’s things like kinases and polygartuases as well … and clearly the Chinese whispers have worked towards the middle and, sort of, fudged the issue and something’s not right.
33.39 J – Yeah
33.40 G – You don’t know which way it’s going to be and gene name’s exciting and one gene name is not exciting … you get that.
33.48 J – Wondering which ones which
33.50 G – Yeah … yeah identification’s a problem. You’ve still got about twenty percent of genes with no … no name, no function
34.04 J – Yeah
34.05 G – They just don’t match anything and … its just whites all the way along and you think
34.12 J – I mean, would you think it would be useful to have the program flag up hey this has got clumps
34.17 G – Yes
34.18 J – it’s got them clumps, it’s got them clumped here and push that up the list to bring it to your attention sooner
34.27 G – Yes, I mean, a clump is much more important than randomness
34.34 J – Yep, justification for my project, this is good. Err, so I can go over this, if I … if I remember that question because there was something I wanted to ask half way through but I didn’t want to stop ya. I’ve completely forgotten what it was, as always happens. So what I can do is, I can have the fun of listening to my hideous voice and then when you get that point again I’ll probably remember it again, oh yeah I really must remember
35.06 G – Send us an e-mail
35.07 J – Yeah, and … if you like I can actually send you that list in the alphabetical one if …
35.16 G – I do go into Abertay on a regular basis on a Wednesday mornings, half past nine.
35.21 J – Wednesday mornings, right
35.23 G – We usually have our meetings half nine to half ten but it’s drifted onto about eleven.
35.27 J – Yes, well, I mean
35.28 G – So there’s some flexibility of when I can meet after then
35.29 J – If you want to meet up on Wednesday after your meeting then … I can what … what I’ve come up with so far. But like I say tha … not … I haven’t really done a lot of testing, I’ve just put a couple through, I actually just put them through for … I think it was actually just for my transfer report to show hey here’s an idea of what we’re doing for the … the testing. So I just kinda shoved stuff through, but I do have … I then had this thing going … this is how they’ve ordered it.
36.04 G – OK
36.05 J – Like is say, at the moment, it doesn’t take into consideration the fact that the clump can go over a couple of wee gaps. It just kinda goes this is … this is … it has clumps like this, sort of. Almost like ungapped BLAST … But … I think it is … it is possible that I could then go back over it and any clumps that are … what would you say, five or six or more …
36.38 G – Well, what you can do is … I’ve seen software for alignment purposes on the internet and you have almost like a default value and then tick boxes and just rerun it with … try that option, try that, try that
36.55 J – Yeah
36.56 G – Those are always useful ... they have routines in them with a name ... BLAST against this one, BLAST using this formula, BLAST using this formula and see what you get.

37.08 J – Yeah
37.11 G – I think if it was the right match you’d hope to see some consistency coming out ... as to what you like and maybe, I don’t know, if something throws it up unusually you might just take a very close look at it and say, right, why is it, what’s special about that one, yes that’s worth a look at maybe ... I like easy options that’s under my control

37.38 J – Yeah ... So like, sort of have a allow gaps, allow gaps of x length
37.49 G – Yes, you can play about with them sometimes. Yes, I’ve seen software I really liked on the Internet for finding proteins of similar function; it seems to have disappeared off the Internet now. And you three proteins in then four and then four and a mismatch and something like this and put these in. So you actually play about with it ... and ... look for similar sequences. I’ve done that and it’s disappeared now, it was on the Internet but someone’s pulled the plug on it

38.18 J – Can you remember what it was called
39.07 G – It might be that one. Hmm, <searching bookmarks.... Mutterings>
39.42 J – Is that Motif...
39.44 G – Dot genome dot A D dot J P ...
39.52 J – It might have just moved servers
39.53 G – It might well have done
39.55 J – Cause I can have a wee hunt for that, I might even be able to find the ... journal to go with it. If I find it again I’ll let you know

40.10 G – You see we moved from netscape to explorer some time ago but I’ve still got all my bookmarks from netscape ... so all the old ones are still here ... There was a really good one, I used to play about with it loads it and it just ... Another one was ... That one, that was great, that’s disappeared ... no it’s not. Yeah, that was an interesting one if I remember ... if that’s the right one. Well, that says motif, I wonder if that would be something similar. Damn. Ah, right so we can do ... D G I dot, dot P and I’ll have no mismatches ... retrieve sequences ...

42.35 J – I thought that said Fatal brain proteins
42.36 G – Yeah ... so hopefully what you’d end up with is something ... you can make these longer and longer and you get a match and have no ... and one mismatch but obviously this is not going to work, this needs more sequence, dot, dot, dot, dot, dot, dot G, I think we’ll allow mismatches and we’ll see what we get. Yeah, so, almost any combination.

43.03 If you’ve got your sequence ... which is showing matches, it’s possible to put them into something like this and say, OK, we’ve got a match four/five space, space, space, space, space then something is this actually coming up with a family and ... if these are all sort of random names then it’s, yes ... or you get a large number there to suggest it’s by random but once you’re down to half a dozen all the same name you think ah, now I’ve got a family

43.35 J – So, for example, for the one with J, that large clump you’ve got on the left hand side, you could stick in the letters with the dot for the gaps and see what it comes up with

43.45 G – Yeah
43.50 J – Actually that doesn’t look too hard to ... because ... I’ve actually been programming visual basic, which is not the greatest programming language but it works. You can access the Internet through the program so there might be a way of actually doing that. Something to think about anyway. So I’ll take ... take that away

44.15 G – And that one, if I remember right allows you to put in ... an S or T. Right, serine and threonine are interchangeable because they’re both hydrophilic. So in that particular point you can either ... you can either have S or T and it’ll pull up both.

44.45 J – If you ... could you just click retrieve and see what it comes up with for us, just so I can see it ... one

44.56 G – One, well, if we go back you can see that ... You’ve got Ds and Vs. so I were to change that to ... Ds and Vs, that should pick up both of those. D and V because it’ll pick up that one and that one

45.14 J – Yeah
45.18 G – God knows where they are now because ... it didn’t pick up the V ...
45.23 J – Yes it did
44.25 G – Yes it did, last one
44.30 J – Cause it’s now in the bold.
44.35 G – Yeah
44.41 J – That’s quite good
And D is a hydrophilic acidic and V is a hydrophobic, so nothing like one another. So it’s not good ones to try.

But it does show that it can do it. But that’s a … yeah, that could be a … an interesting little add on.

Certainly same families

You could just … even if you just had a wee button on … on the … the GUI

But it does show that it can do it. But that’s … yeah, that could be a … an interesting little add on.

Certainly same families

You could just … even if you just had a wee button on … on the … the GUI

Sort of, like, click here to see if it’s got any other … and press it and I could send you off to there and … Yes … yes, that could be interesting … I could do that … it’s not too difficult. Well I think that’s it … So, I’ll go over this. I’ll see if I can remember that question is, it’s probably completely irrelevant if … from what you’ve said afterwards. But … if I could at least just remember it … I could know. And we could meet up on … on a Wednesday morning, I mean, I don’t really do anything else, sort of, set on Wednesdays and … I can show you that actual list of … well, one, how I’d ordered them because I sat there I did them myself. I did a hundred of them and it was completely boring. You only had ten; I had a hundred of these things. But, obviously, I wasn’t looking at it in the same way you would look at it with oh, well, there’s just one gap there that’s still pretty good. I was just going well what are the clumps, well that one looks more clumpy, well I’ll stick that one above that one … and then I can also show you what the programs have put out. We can see … maybe see what you think of that, that would be really good.

OK, thank you

Recapped how she saw Tim used BLAST for the purposes of primer creation and verification

• Looking for (in Entrez) sequence of known function,
• Use BLAST for anything missed, if anything uses different names.
• Throw in a sequence and decide which of the results are related
• Problem with the way BLAST works, say BLAST gene X, first few are X next aren’t
• Used to confirm sequence generated in survey are what they are looking for, are correct
• In alignment looking for ones and twos together if following a codon
• The evolutionary rate of the three base pairs in a codon differs
• Different issues
  o DNA – variations on codon
  o Protein – variations on positives, looking at amino acid families, what amino acids can replace other without a huge impact on structure

Agreed that there were differences and that project may not be as helpful than to others

Pointed out that future projects may use BLAST in a more function determining way and would help project in any way he could

Asked Tim if he would be willing to do the card sort

• B very even, J very clumpy … very different from each other
• First thing could do is separate into number of blacks and number of whites

Explained this was intensity and that it was virtually constant on all the sequences so as to see changes in distribution rather than similarity

• Sorting distribution will be difficult
• Looks for a more even match to show it’s the same protein
• J could show, for example, an ATP binding site but the overall sequence has a different function
• Looks for even-ness across whole sequence to show match to whole functional protein
• This leads to putting B and C over J
• We’ve done experiments of high throughput sequencing of genes
• Want to put the resultant sequences into ‘bins’

- Have trees of sequences, evolutionary distances
- Tried to use BLAST but it was useless
- Used phylogenetics program instead
- Therefore likely to order by even-ness because affected by the experience of what happened in experiment
- Quick reorder:
  - B, C, A, D, I, E, H, G, F, J

Jacky
Mentioned that looking at clumpy may indicate active sites

Tim
• Example, if have enzyme that converts one to two binding sites for one and two conserved
• But, enzyme could also have ATP or ADP etc… binding sites
• Which leads to BLAST finding the common binding site
• At the moment the scores get will be higher because throwing known function at BLAST
• Future maybe starting projects that are looking for genes expressed in certain situations and this may find more use out of a clumpiness metric
• One the pictures in BLAST get red to white sequences
• Ours are all red

Jacky
Was confused

Tim
Opened up EMBL to illustrate, putting a sequence in and BLASTing it. Showed the graphical representation of the matches of the hits to the query sequence
• Know we use primers for things we’re interested in
• In nucleotide to nucleotide BLAST
• What we find is the top rank you get full matches
• Cluster very tightly
• All the same thing because molecular ecology is the distribution of functional groups or organisms related by phylogeny or function
• Environmental samples
• When you publish a paper, any sequences in a tree in that paper have to be put in the database even if sequence is identical
• This saturates the database
• All the hits discovered by BLAST are the same gene, in this example, Glo three
• But would prefer samples of each
• Let’s try a different sequence (one of Susan’s sequences)

Puts Susan’s sequence into BLAST
• In know there are ways to send many sequences (batch file) but we just don’t
• See here most of this is uncultured, unknown sequence
• Because of the nature of molecular ecology you get a lot of uncultured
• But you can get a hit to the occasional specified gene, which could be someone who’s done the wet science
• Look at the alignment for these ones you see patchiness
• For us there are tens of projects doing the same thing in different environments/situations, using the same primer
• In Susan’s case there are only four/five projects on the go
• Still, everything in the results would be in the tree somewhere but probably in a different order
• Molecular ecologist is more likely to pick B as best; geneticist is more likely to pick J
• Using BLAST, when see what it pulls out, would be more impressed with B

Asks Maarja as a person who searches for function which she would be more impressed with B or J

Maarja
Depends on the sequence

Tim
Depends on the database

Maarja
Don’t know

Tim
• Molecular ecologist would like B but preferable with more black
• But with a gene of unknown function J could indicate active sites

Maarja
B is fifty/fifty spread so wouldn’t think that was related, so would go for J

Tim
Depends on the gene, if looking at ITS, looking at J

Maarja
Yes, if got a hit to B, wouldn’t be impressed

Tim
• Lengths of the lines related to the primers used
• Very few are full length
• Full length could be a full strain or same primer
• Shorter ones are just different primers
• Or possible a gene with a patch of similarity
• Would like to group the results to get an idea of all the different types of hits possible
• For example, in the results the top fifty-five could all be identical

Maarja
If you get types with many entries that are very similar so you would look further down for the next similar

Tim
• With these sorts of sequences with the top fifty all the same
• Want a working link that puts them all in a bin then looks for the next
• If it’s for an unknown sequence
• If looking at fifty different strains of mouse with alcohol dehydrogenase ...
• Suppose if you get so many you would know what the function is and be delighted
• Geneticists have their organism and pull a sequence out of the EST database and send it to BLAST saying that matches to alcohol hydrogenase
• Many hits the same
• New UAD project that takes the output and compacts into bins

Jacky
Compresses the results

Tim
Yes that would be very nice

Jacky
Getting back to the sequence match distribution asked if the molecular ecologist would look for an even spread

Tim
• Depends what you’re looking for
• Molecular ecology looks at the diversity of a single gene
• They build up a mental image of that the gene looks like because they BLAST so much
• Some sequences stick out because they look different
• Molecular ecologist looks at lots of copies of one gene/DNA sequence form lots of organisms
• Rather than lots of genes from one organism
• For each gene you get an intimate knowledge of what it looks like
• It’s really sad when you can tell what the next 10 characters will be before they are sequenced

Maarja
Doesn’t know all of her sequence but does know the first bit so can see it coming

Tim
• Molecular ecologists never really get to the point of seeing clumpiness
• Would prefer to see more black in B
• If transposed J onto B so had all the black from J and B would be very happy
• We believe that species come from the same ancestor
• They evolve away from each other
• B has evolved more evenly than J
• You have constraints on what can evolve in the gene/sequence
• So it's still very dependent on what gene/sequence you are working on

Jacky
Tried to get back to Tim explaining why he put one particular sequence above another

Tim
• B and J are extremes
• When looking at two hundred sequences from the same gene you can create a multiple alignment
• You spend a lot of time looking at this multiple alignment
• And learn where there are conserved regions and where there are variable regions
• Depending on the gene depends on the pattern you are looking for
• Distant neighbours are more likely to look like J
• Which means it's very difficult to rank the list
• We look at DNA sequences
• Look at the codon positions
• 3rd codon position varies more than the other two (second also?)
• Which makes B even more attractive because could see where a white could just be where there is a change in the third codon position
• Does the third vary more than the second? ... sure it's the third
• You can look at a codon table and see the differences
• For example, praline is CCn where n can be A, T, G or C
• The first two positions would be conserved it was necessary to have praline but the third wouldn't need to be
• Other examples are arganine or glycine
• Therefore the evolutionary rate across the three codon positions can be very different
• So we tend to look for pairs together
• But, again, it depends on the gene/alignment you are working with
• Ordering is very difficult
• We look more at a sort of global similarity
• BLAST does a local similarity that extends a short word
• We are looking at phylogeny and substitutions/base pair which is the threshold limit
• BLAST doesn't look at similarities within a gene
• Unless they were very distant
• BLAST wouldn't produce the correct order of the phylogenetic tree
• It would only shuffle them up a bit

A.8. Michael Taliansky – 26th April 2004

00.11 J – So we spoke a couple of years ago about how you use BLAST and what you did when you were looking through it to decide whether they were partic ... of particular use for you and as I remember, you deal with viruses.

00.31 M – Yes ... mainly

00.34 J – Mainly. I did this was based not just on the interview I did with you but also on the interviews I did with Gary Lyon, Tim Daniells and David Marshall. So I was ... I'd just like to show you this

0.52 M – OK

0.53 J – See if you disagree drastically with any of it. So for ... when you're looking at

0.59 M – As I wrote to you, I'm not a great expert in this area. I'm using BLAST just occasionally and I'm maybe not very, you know good, with different types of this analysis

01.14 J – No, I understand that it's just ... but to get your perspective it w ... is also important. I am trying to get in touch David, he hasn't replied yet though so I'm still waiting for that. So, I mean, looking at BLAST, you've got the different sorts of BLAST, obviously, then you've got
your scores, how good are the scores; do you see multiple instances of a function or a species or what-not; sort of, how close are the species, I suppose that would be the virus for you

01.49 M - Yep

01.50 J - How ... you know ... the amount of conservation and I think there was this ... the mRNA/EST was mentioned because that gave you and indication whether you had just a coding region, which is more important.

02.05 M - Yes

02.07 J - Where ... where I have actually focused my project is on the actual alignments showing where matches are and looking for whether there’s clumps in them. And the reason ... if you remember we were talking about automating BLAST to rearrange the results into a more relevant list. So not just based on how similar but also are there other important things so that maybe a hit that would be low down in the similarity but would still be quite important can brought be further up.

02.43 So that’s what we’ve been trying to do and to do that we’ve been trying to measure the degree of clumpiness ... nice word there, dictionary hates it. So, I mean, I’ve got a couple of examples here, I was hoping you’d maybe sort these for me as well. So like if you’re looking ... look at these, you’ve got the black are the matches and the whites are the wee gaps in between. So you’d say well that’s got a clump there so maybe that’s an active site or something that might be interesting whereas this one, it’s very sparse

03.31 M - Yes

03.33 J - So, I mean, are you quite ... you agree with what I’ve ... my ... obviously, you might use BLAST slightly differently but ...

03.43 M - Of course, this depends on your task and I have very specific tasks and in general I think it’s right. If we look at the distribution of similarities and differences, then if you have a region which is more similar ... much more similar than others and this is very specific segment then you can suggest that this is conserved site.

04.15 And we use such an approach, of course, but there are some other tasks, you know, and I need to use BLAST just for, you know, for preliminary research, which I will use for further predictions, which can be tested experimentally

04.40 J - yes, this does ... we’re not saying this would replace the wet science cause nothing could replace the wet science but we’re just saying that maybe this bit might give you a better idea and may give you a better hint of what wet science to do.

04.57 M - It’s about that

05.01 J - So, I mean, that’s ... this is ... so this is what I’m doing for my project and I was ... as I said in the e-mail, I was wondering if you would be willing ... cause I’ve got a number of measures to measure this clumpiness and I need to test it, obviously, to make sure it’s working correctly.

05.20 So that’s what these are for if for my ... I’m creating this dataset to test it on and, obviously, I want it to go from clumpy to unclumpy so I can check it’s putting it in the correct order. So if I could get you ... there’s only ten, it’s not too bad, I had a hundred to do, these are only ten. If I could get you, as best you can order from clumpiness ... clumpy to unclumpy or visa versa and while you’re doing that, tell me why you’re putting one above the other.

05.59 M - Well, first of all, what I’d like to see, not just distribution of these black and white boxes but also description of what is that because to me it is important. When I have this BLAST analysis then I’m starting, not with this distribution but what is what

06.24 J - Well, I mean, that’s that one there and that’s another one and, of course, that would also be important but I can only really focus on one particular aspect so for that we’ve chosen this clumpiness

06.41 M - OK, I’m not sure I’m very good with this but ... well, again, you know if you do BLAST with a sequence then you can see direct homologues of this sequence. So then you will have distribution along the whole sequence which is very similar to the ... your sequence and then there will be some more sequences which will have similarity only in active sites. So, first thing you can do is see at very high level of similarity along the whole seq. Then you can see protein structural homology this protein. OK? So you’d like me to try to find some similar structures here or what

07.56 J - It’s more, sort of, if you looked at these which one would you put above the other as more important to yourself. If you came ... I mean, obviously if you have ... if there’s matches all the way along, that’s an obvious ... yes good hit but we’re trying to find in that more sort of bla ... grey area
08.19 M – OK. If I’m looking for more conserved domains I would prefer this one <J> because this one consists of some very conserved motifs, which maybe something like functional domains, OK, and if you look at different sequences there might be that … well, actually different … in viral proteins some of them are very conserved, some are not conserved at all but only functional domains are conserved. And when we did something like this then we found in the whole protein just two areas with high similarity.

09.12 So its looks like this, I would say here you have this zone of high similarity, this zone, this zone, this zone and this. So what I’d like to know, more than that, if this is a protein sequence, then I’d like to see what sort of domains are here, Arginine rich domain, Leucine rich domain or whatever or 1-Lysine rich domain. So then you can see that … if you have here many negatively charged amino acids then you can suggest some properties of these domains, it maybe RNA binding area, for example, or nuclear localisation signal, something like that. So in terms of search for functional domains I would be very much interested in these areas.

10.13 If you look just at evolution, or some very close relatives and you know that this similar proteins from different viruses, for example, they are arranged in a similar way in different viral genomes then, of course, you may be interested in something like this <B> where you have very, well, similar level of similarity but this random distribution. Here <J>, distribution is not random this is some protein with high similarity and here <B> you have just random levels of similarity. So it might be interesting for other type of the work. This depends on your tasks.

11.22 J – Yeah

11.27 M – well, ok, and also with such a distribution you have you can have something like this <D>, for example, and in this case you see both more or less random distribution of similarity but in addition you can have some motifs of higher similarity and again look at the sequence of this, if this is the protein sequence, then you can have some ideas about it. Then you can look at, not only BLAST for comparing sequences, but also for searching some functional domains, it will give you more ideas. Ok, it’s what you’d like me to talk about?

12.23 J – Yes, what would you look for in them. It’s just when you saying you look for both that one <J> and that one <B> it could be a bit tricky for you to put these in an order because it does depend on …

12.36 M – Depending on my task. If I am looking for functions I would prefer this one <J> then I would start with this one <J>, then I will go to this one <G> to … then I will go to maybe this one <D>, just to look at these conserved domains.

13.03 J – Can I just the letters of the side.

13.11 M – Well maybe … well not this one … this one … this one … but if we are talking about … we are talking about relations then I would be interested in the pictures like … I’ll put it here as well because there are some clumps … like this. If I’m looking for just relations between different proteins I … because it’s not occasional figure, I would say, it’s very interesting.

14.09 If I don’t care about functions, because some works are dedicated not to functions but to their relations and always when you sequence anything, new virus for example, or new viral strain then somebody can ask you about similarity, total similarity. You don’t know about some functions of this sequence but just … then … it’s very interesting and it might be that the level of similarity here is higher than here. So for this work or for evolution work or phylogenetic relations between different organisms this one <B> may be very interesting.

14.59 J – I got a similar response from Tim Daniells who does also do phylogenetic work as well. So he kinda put it into reverse order, rather than clumpy to unclumpy he was saying he was looking for evenness and put it into reverse order so …

15.15 M – OK, so if I’m looking at functions, I would go for this <*> <J>, if I’m looking at relations then I will go to … I will put this as the first one. But, actually, sometimes if you’d like to have both types of <*> then you cannot put it first and then this one. It depends on your current task. So in my practice there are some tasks when I looked at this … much more interested in this type sequence similarity distribution and I had tasks when I was interested more in … just in the percentage of similarity and equal distribution along the whole sequence.

16.12 J – Right, if I could … if you were looking for function. Let’s just stick with function just now, could I just get you to order the … just order all them … so you would have I first.

16.28 M – I maybe mistaken but … again, I need to look at the sequence also, because, for example … well, not this one … this one, the first one I would say, then maybe this one … like to see something like … this one … and then all this because it’s difficult … maybe this one is not so bad because there are some <*> … and the same <*>.
17.40 Another thing is, what I’m doing is ... I’m doing these from my biological background because for me it’s very important to look ... what’s within the black box. If this is Arginine it’s interesting, if this is just ... just Alanine maybe it’s not so interesting. If this is a reactive amino acid which can be used for certain type of function, it’s interesting, not just figures

18.18 J – That can also be brought into it but this is, sort of, the first step

18.24 M – I’d like to find something like this clusters <J>, not like this because with such a distribution it’s also interesting but, again, you need to look at what’s that, what’s within these black boxes. But for work like relations, relationships between different amino acids, of course, I would start with this one <B>

18.57 J – Right, so it would be, sort of, the reverse order if you were just looking for relationships. That’s good, can you tell me why you would have put this one <H> below this one <G>? For example just ...

19.17 M – It’s just ... I can do it this way ... it doesn’t matter for me ... I cannot see big difference between this one and this one. Maybe this is better, the first order, because I would say that these two domains <G> look more suitable or as a candidate for some function domain, they are longer than ... in here <H> I have just one domain which is longer and if I have some short domains it might be that they are too short to fulfil any function.

20.04 In this case there’s two domains and also I don’t like to see the protein many such domains because ... well we can, again, this depends on the protein but usually protein has two free conservative domains which are recognisable and even multifunctional protein, they have just some specific <*> consisting of very few motifs so I would prefer to look at this particular domain of higher similarity.

20.53 That’s why I put it here ... well just I would see here two domains with very high speci ... similarity so I’d like to know something about that and here I would say just one domain here. Here ... well I would put this one <I> here because one, two domains. One ... I would put all of them in on group <H, I, F, A> and here just one domain maybe with these. That make sense?

21.53 J – Yes that makes perfect sense, I mean, it’s kinda tying in with what I heard from both Tim and Gary because Gary ... Gary Lyon was looking for more the function so he was looking more for the clumps whereas Tim was looking for the phylogeny so you’ve got the eveness. So I’ve got ... I’m getting both from you so this is great.

22.14 It’s nice because it’s sort of confirmation for me whereas you use BLAST for confirmation for your function this is confirmation for me for the thing, I mean, obviously, it’s something else that both Gary has ... and Tim have also said, knowing what the actual amino acids are is also important. But, again, we’re kinda just focusing on the one thing, getting that ... getting that correct and then we can add in, right, OK, so we know what this clump is, we know where it is, so what’s it made of, I mean, we can also tell you that and see if that helps as well.

22.50 I mean, I am going to be doing a prototype; it might just only do looking for ... just looking for the clumps and that or I could put a little switch on it so you could reverse it and look for eveness. Would you be willing to test that for me, just to see if it’s useful, any help ... whether it helps at all

23.14 M – How could I do that?

23.16 J – Well if you ... when you actually do a BLAST search, I know you don’t do them often, but when you do if you just run it through the program and then see what it comes up with and see if that’s any help for you. But you don’t have to tell me what the project you’re working on is or anything to do with that if you don’t want to but just to say, well this was helpful cause it told me some information I didn’t find in this way. That would be really helpful for me as well. It will probably be a couple of months time, it’ll probably be in the summer that I’ll doing that but if I e-mail you again closer to the time, that be OK?

23.50 M – Sure

23.52 J – I think that’s ... yeah, I think that’s everything I wanted to discuss with you and, like I say, it’s very helpful because it does tell me right OK I’ve got roughly the right idea but I need to add in little bits and pieces, which should be feasible

24.09 M – So you will e-mail me and I will send you my accession number or whatever and you will do BLAST there and then you will send results to me or would you like I will do BLAST here

24.23 J – What I can do I can give you a copy of the program and you just ... if you give the program a sequence it should send it to BLAST for you and then come back with results and I can give ... if you want two copies of the results, one what BLAST would just give you and what my program would give you and you can, sort of, compare them and see ...

24.48 M – Very interesting. Certainly I’d like to do that
24.53 J – OK, thank you very much. Just gather these back up again. Yeah, I think that’s everything I really wanted to speak with you about but thank you very much that was really interesting and helpful for me.

25.10 M – OK

25.12 <tape stopped>


<Tape started at 25.13>

25.19 J – What I wanted to do, to speak to you about was basically what we talked about before just to make sure that I have everything straight in my head and then sort of briefly discuss what I’m doing now and ask if you’d be willing to help me with a little test.

25.33 D – OK

25.35 J – So, basically, there … from the previous interviews I did I came up with this little diagram of the things that you look for in a BLAST result to decide whether it’s a good hit or not. So, obviously, you’ve got the … all the different types feed in, you look at what the score is, whether you’ve got several of the same descriptions and matches and that and if it’s got coding regions, if it’s a similar species.

26.12 What we’ve concentrated on is this one here was in the alignments you see clumps in the actual matches and what I’m trying to do now is … I went to a conference so I’ve got a baby poster here … what we’re doing is we’re kinda looking at that and saying well, we want to reorder the BLAST results by, as well as the similarity, but also by how clumpy are the alignments, if you’ve got clumps. So we’re trying to put a measure to that to help order it. So that’s what we’ve got there, I’ve changed it to a binary just to make life easier and then you reorder it.

26.54 And these are just a couple of the ideas; I still need to test them. And that’s the next step is getting the data, getting a sort of proper dataset that actually has clumpy alignments to unclumpy alignments and if I can get my measures to put it in as similar order as possible, then I know that it’s working.

27.21 D – Yeah, I suppose that one thing that would relate to that, I suppose, more and more <*> a lot of this has happened since I last talked to you in terms of how we do things and work with things. We’ve done quite a lot of work looking at individual sequences or groups of sequences and what we’ve done a lot of work on is that rather than concentrating on a single sequence we quite often will try and work on a, essentially, what is a small gene family.

27.55 <Tannoy announcement>

28.07 And so, quite often, what we do is we do a, essentially, a combination of … we would do a BLAST search, we would actually then pull back the sequences which are within some definition of being related to the sequence we are interested in. We would then, in protein coding sequence in particular, we would then do a multiple alignment using all of these sequences and we would then, based on that alignment, do a phylogenetic analysis.

28.41 So that would enable us to actually structure the relationship between the sequences. We’ve done this quite a lot now with a range of different sort of proteins. So, typically what would happen is someone would come along to me, say I’m interested in this particular gene or this particular group of genes and it’s been cloned or something or other and we’re interested in some of the stuff that more locally relevant and some of the species, now what can you actually tells us about it.

29.10 And we will mine the sort of EST databases and we will then find all the related sequences, we’ll then do protein translations of those, then we do alignments and then from there phylogenetic analysis. That’s a really quite a powerful way of being able to sort out just what’s what because again one of the things that we’ve found out increasingly, I think, over the last couple years is the … one of the big issues about defining relationships between sequences is the amount of errors that are present in the sequence databases, which is a big, big issue.

19.52 J – Yeah, multiple alignments is another thing we would have like to have done but there is already a lot a work already done on it, which is why we concentrated on … it’s the same as all the other sort of bits and pieces, I’ve been doing a lot of reading and a lot of them, like hunting just doing word … just looking for words in the descriptions, that’s already been done. So we’ve chosen the one that hasn’t had a lot of work done on it, in fact I haven’t found anyone that’s done much work on the alignments, which is kinda of reassuring.

30.29 What was I going to say, oh yeah. At the moment, like I say, I’m trying to get a test dataset done up and I was wondering if you would be able to help me with that. It’s basically; I’ve got these ten little cards and on that a white block is a match and a black block is a mismatch, obviously we’ve kinda ignored the positives, which I know we shouldn’t do but for simple …
to keep things simple we have and if you would be willing to kinda put them in order and say well this one looks most interesting most clumpy because and it would be higher than such and such

31.09 D – OK
31.10 J – That would be really, really helpful
31.11 D – So this is essentially a set of alignments
31.15 J – Yeah, they’re made up
31.16 D – Yeah, as a notion of that
31.19 J – They’re completely made up, they’ve got absolutely no relation to actual alignments but it’s just a … to give an idea because I’ve already … I’ve ordered some myself, I’ve actually ordered a whole set of a hundred of them, which was very, very boring. But it’s, obviously, I don’t have the same idea of what you’re kinda looking at, looking for in the alignments.

31.44 D – So what you’re really just talking about is just from the pattern of matching nothing to do with the actual nature of the matching because, obviously, one of the things that you can do is that if there’s clumps, one of the issues is just the fact that they are clumps, is actually what
32.08 J – What the actual amino acids are
32.10 D – Yeah, what is the information content. So, for example, quite often if you’re looking at an alignment of say, involving enzymes if you find clumps, even if there’s a comparatively low overall degree of homology but there are blocks of high homology or over a large number of sequences if there is a … if there always tends to be conservation in a particular region that can be interesting and sometimes that may reflect an active site in an enzyme or something like that. Though, active sites in enzymes aren’t always just here, there’s maybe something, say a histidine conserved here, another amino acid conserved here but …

32.50 J – Yeah, that is one of the issues that we might not be able to deal with in the scope of my project. Another thing that … I’ve already spoken to Gary Lyon and Tim Daniel … another thing they’ve said is that sometimes you might get an active site but it’s to something that’s so common, it doesn’t really give you such a big clue as to what you’ve got.

33.14 It’ll just tell you that oh well it’s also got that in it, which, obviously, can be useful but if you’re looking for something more specific and I think Michael also mentioned that sometimes even if it’s a mismatch if you look at the amino acid, if it’s close enough then you could say oh well that’s … I could really think of that as a match anyway.

33.40 D – That’s where you use an alignment matrix on that sort of basis. So that’s one of these cases where we’ve actually … we’ve been working on a particular set of sequences and this is … what we’ve done is we’ve taken a related set of sequences and done alignments with them and they’re obviously nicely conserved because they have regions that fall out quite nicely.
34.26 J – Yeah, well, that would be a sort of, the next step, if you find these clumps, do you find the same clump in the other matches in your BLAST result would be the sort of … which is kinda the multiple alignment there, which is what we were also talking about.

34.44 D – So that’s a really a powerful way of looking at these things. This is a much less refined one but, again, there are similar … sort of nicely homologous clumps in there. The thing is if you find stuff that is conserved all the way through, there are also some amino acids that are more important than others. So, for example, conserved histidines and one or two amino acids are often associated with functional … they’re also … you can do motif searches…

35.35 <Discussion about background noise…. Move to another room>
35.59 <A lot of static… and silence>
38.21 J – Yeah I’m trying create an algorithm that would quantify the degree of clumpiness in each one an then I can use that to reorder the list, the BLAST list, so obviously …
38.30 – 51.59 <More static>
APPENDIX B – DATA FLOW DIAGRAM FOR CURRENT SYSTEM

B.1. Context level

BLAST Program

Test Sequence

BLAST results

0 Context

Genetic Function Determination System (Scientist)

Accession Number

Details from Annotation files

Internet Databases
B.3. Level 2 – Submit Sequence

1. Submit Sequence
   1.1 Check the test sequence for contamination
   1.2 Translate into ORF protein sequence for BLASTp searches
   1.3 Paste into BLASTn, BLASTp and BLASTx

2. Determine which results are good enough to follow up

Test sequence

BLAST Programs

BLAST E-mail/HTML file

Nucleotide

Nucleotide and amino acid sequences

Retrieve results from internet or e-mail account
B.4. Level 2 – Determine which results are good enough to follow up

1
Submit Sequence

BLAST results

2
Determine which results are good enough to follow up

2.1 Are there clumps in the results?

2.2 Is there a similar/same species?

2.3 Are there any conserved proteins or active sites?

2.4 Is the result from a coding region (mRNA, EST)?

2.5 Are there clumps in the alignments?

Selected results from individual BLASTs

2.6 Are there similarities between the three BLASTs

3
Obtain additional information

Selected results

Selected results from individual BLASTs
B.5. Level 2 – Determine which results + extra data are relevant

<table>
<thead>
<tr>
<th>4</th>
<th>Determine which results + extra data are relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Check for contamination in database sequence</td>
</tr>
<tr>
<td>4.2</td>
<td>Check if contamination is in similarity region</td>
</tr>
</tbody>
</table>

Results + Annotation data

| 3 | Obtain additional information |

| 4.3 | Check date for time expiry |
| 4.4 | Look for keywords and species in descriptor |
| 4.5 | Obtain further information |

External Databases

| 4.7 | Look at publications & cross references |
| 4.6 | Verify submitting person/place |
| 4.8 | Check sequence orientation and translation |

Non-contaminated sequences

Inconsequential contamination

Standard keywords

Relevant results
APPENDIX C – SPEARMAN’S RANK CORRELATION COEFFICIENT TABLES FOR MEASURE EVALUATION

Lay out of the tables:
- Measure R is the rank order into which the particular measure puts each line
- C and subject R is the rank order into which consensus or subject puts each line
- C and subject d² is the difference between the consensus or subject order and the measure order squared – used for spearman’s rank coefficient calculation
- Bottom of each d² column is the sum of all these differences
- Finally there is the spearman’s rank coefficient value colour coded as to how well correlated the measure and consensus or subject orders are, the closer to one the better the value:
  - Green = greater than 0.9 = correlates very well
  - Yellow = between 0.7 and 0.9 = fair correlation
  - White = between 0.5 and 0.7 = poor correlation
  - Orange = less than 0.5 = very poor correlation

Table C.1. Average

<table>
<thead>
<tr>
<th>measure</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2</td>
<td>16</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>24</td>
<td>16</td>
<td>10</td>
<td>30</td>
<td>88</td>
<td>30</td>
<td>42</td>
<td>18</td>
</tr>
<tr>
<td>Spearman</td>
<td>0.85</td>
<td>0.90</td>
<td>0.94</td>
<td>0.82</td>
<td>0.47</td>
<td>0.82</td>
<td>0.75</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Table C.2. Largest group size

<table>
<thead>
<tr>
<th>measure</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>3</td>
<td>43</td>
<td>3</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Spearman</td>
<td>0.96</td>
<td>0.96</td>
<td>0.93</td>
<td>0.98</td>
<td>0.74</td>
<td>0.98</td>
<td>0.95</td>
<td>0.93</td>
</tr>
</tbody>
</table>
### Table C.3. Redefined Largest Group

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5 0</td>
<td>5 0</td>
<td>6 1</td>
<td>8 9</td>
<td>6 1</td>
<td>5 0</td>
<td>4 1</td>
<td>5 0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8 0</td>
<td>8 0</td>
<td>8 0</td>
<td>9 1</td>
<td>8 0</td>
<td>9 1</td>
<td>8 0</td>
<td>8 0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>4 9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9 0</td>
<td>9 0</td>
<td>9 0</td>
<td>5 16</td>
<td>9 0</td>
<td>8 1</td>
<td>9 0</td>
<td>9 0</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6 1</td>
<td>6 1</td>
<td>6 1</td>
<td>5 0</td>
<td>2 9</td>
<td>5 0</td>
<td>4 1</td>
<td>6 1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2 0</td>
<td>3 1</td>
<td>4 4</td>
<td>2 0</td>
<td>3 1</td>
<td>2 0</td>
<td>3 1</td>
<td>2 0</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4 0</td>
<td>4 0</td>
<td>3 1</td>
<td>4 0</td>
<td>4 0</td>
<td>4 0</td>
<td>6 4</td>
<td>5 1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3 1</td>
<td>2 0</td>
<td>2 0</td>
<td>3 1</td>
<td>6 16</td>
<td>3 1</td>
<td>3 1</td>
<td>2 0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

Sum 2 6 52 8 20

Spearman 0.99 0.99 0.96 0.99 0.68 0.99 0.95 0.98 0.88

### Table C.4. Proportion

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5 9</td>
<td>5 9</td>
<td>6 4</td>
<td>8 0</td>
<td>6 4</td>
<td>5 9</td>
<td>4 16</td>
<td>5 9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>8 9</td>
<td>8 9</td>
<td>8 9</td>
<td>9 16</td>
<td>8 9</td>
<td>9 16</td>
<td>8 9</td>
<td>8 9</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>4 9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9 0</td>
<td>9 0</td>
<td>9 0</td>
<td>5 16</td>
<td>9 0</td>
<td>8 1</td>
<td>9 0</td>
<td>9 0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>6 16</td>
<td>6 16</td>
<td>6 16</td>
<td>5 9</td>
<td>2 0</td>
<td>5 9</td>
<td>4 4</td>
<td>6 16</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2 1</td>
<td>3 4</td>
<td>4 9</td>
<td>2 1</td>
<td>3 4</td>
<td>2 1</td>
<td>3 4</td>
<td>2 1</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>4 4</td>
<td>4 4</td>
<td>3 9</td>
<td>4 4</td>
<td>4 4</td>
<td>4 4</td>
<td>6 0</td>
<td>5 1</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>3 1</td>
<td>2 4</td>
<td>2 4</td>
<td>3 1</td>
<td>6 4</td>
<td>3 1</td>
<td>3 1</td>
<td>2 4</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
<td>1 1</td>
</tr>
</tbody>
</table>

Sum 41 57 29 47

Spearman 0.75 0.72 0.65 0.72

### Table C.5. Frequency

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>5 4</td>
<td>5 4</td>
<td>5 4</td>
<td>6 1</td>
<td>8 1</td>
<td>6 1</td>
<td>5 4</td>
<td>4 9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
<td>10 0</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>8 1</td>
<td>8 1</td>
<td>8 1</td>
<td>8 1</td>
<td>9 4</td>
<td>8 1</td>
<td>9 4</td>
<td>8 1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>7 25</td>
<td>7 25</td>
<td>7 25</td>
<td>7 25</td>
<td>7 25</td>
<td>7 25</td>
<td>7 25</td>
<td>4 4</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9 4</td>
<td>9 4</td>
<td>9 4</td>
<td>9 4</td>
<td>4 5</td>
<td>9 4</td>
<td>8 1</td>
<td>9 4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6 9</td>
<td>6 9</td>
<td>6 9</td>
<td>5 4</td>
<td>2 1</td>
<td>5 4</td>
<td>4 1</td>
<td>6 9</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2 1</td>
<td>3 1</td>
<td>3 1</td>
<td>4 1</td>
<td>2 1</td>
<td>3 1</td>
<td>3 0</td>
<td>2 1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4 1</td>
<td>4 1</td>
<td>4 1</td>
<td>4 1</td>
<td>1 4</td>
<td>4 1</td>
<td>6 9</td>
<td>5 4</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3 0</td>
<td>2 1</td>
<td>2 1</td>
<td>3 0</td>
<td>6 9</td>
<td>3 0</td>
<td>3 0</td>
<td>2 1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

Sum 45 45 45 37 45 45 45 53 39

Spearman 0.73 0.73 0.73 0.78 0.73 0.78 0.73 0.68 0.76

231
### Table C.6. Frequency of upper group sizes only

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5 4</td>
<td>5 4</td>
<td>5 4</td>
<td>6 9</td>
<td>8 25</td>
<td>6 9</td>
<td>5 4</td>
<td>4 1</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
<td>10 1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8 0</td>
<td>8 0</td>
<td>8 0</td>
<td>9 1</td>
<td>9 0</td>
<td>8 1</td>
<td>9 1</td>
<td>8 0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
<td>7 16</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9 0</td>
<td>9 0</td>
<td>9 0</td>
<td>5 16</td>
<td>9 0</td>
<td>9 0</td>
<td>8 1</td>
<td>9 0</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6 9</td>
<td>6 9</td>
<td>6 9</td>
<td>5 4</td>
<td>2 1</td>
<td>5 4</td>
<td>4 1</td>
<td>6 9</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2 1</td>
<td>3 0</td>
<td>4 1</td>
<td>2 1</td>
<td>3 0</td>
<td>2 1</td>
<td>2 1</td>
<td>3 0</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4 1</td>
<td>4 1</td>
<td>3 0</td>
<td>4 1</td>
<td>4 1</td>
<td>4 1</td>
<td>6 9</td>
<td>5 4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3 1</td>
<td>2 0</td>
<td>2 0</td>
<td>3 1</td>
<td>6 16</td>
<td>3 1</td>
<td>3 1</td>
<td>2 0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

| Sum     | 33| 31| 31| 33| 77| 33| 35| 31| 33 |
| Spearman| 0.80| 0.81| 0.81| 0.80| 0.53| 0.80| 0.79| 0.81| 0.80 |

### Table C.7. Intensity overall (window size 16)

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2 5</td>
<td>9 5</td>
<td>9 5</td>
<td>9 6</td>
<td>16 8</td>
<td>36 6</td>
<td>16 5</td>
<td>9 4</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10 10</td>
<td>0 10</td>
<td>0 10</td>
<td>0 10</td>
<td>0 10</td>
<td>0 10</td>
<td>0 10</td>
<td>0 10</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9 8</td>
<td>1 8 1</td>
<td>8 1 9</td>
<td>1 8 0</td>
<td>8 9 1</td>
<td>8 1 0</td>
<td>9 8 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6 7</td>
<td>1 7 7 1</td>
<td>7 1 7 1</td>
<td>7 1 7 1</td>
<td>7 1 7 1</td>
<td>7 1 7 1</td>
<td>7 1 7 1</td>
<td>7 1 7 1</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>8 9</td>
<td>1 9 1 9</td>
<td>1 9 1 5</td>
<td>9 9 1</td>
<td>8 9 1</td>
<td>9 9 1</td>
<td>8 9 1</td>
<td>9 9 1</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7 6</td>
<td>1 6 1 6</td>
<td>1 5 4 2</td>
<td>25 5 4</td>
<td>4 3 4 9</td>
<td>6 1 6 1</td>
<td>4 3 4 9</td>
<td>6 1 6 1</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4 2</td>
<td>4 3 1 4</td>
<td>0 2 4 3</td>
<td>1 2 4 2</td>
<td>4 2 4 3</td>
<td>1 2 4 2</td>
<td>4 2 4 3</td>
<td>1 2 4 2</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3 4</td>
<td>1 4 1 3</td>
<td>0 4 1 4</td>
<td>1 4 1 4</td>
<td>1 6 9 5</td>
<td>4 7 4 7</td>
<td>1 6 9 5</td>
<td>4 7 4 7</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>5 3</td>
<td>4 2 9 2</td>
<td>3 9 4 6</td>
<td>1 3 4 3</td>
<td>4 3 4 9</td>
<td>2 9 3 2</td>
<td>4 3 4 9</td>
<td>2 9 3 2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

| Sum     | 22| 24| 22| 74| 32| 74| 32| 36| 22 |
| Spearman| 0.87| 0.85| 0.87| 0.81| 0.55| 0.81| 0.78| 0.87| 0.76 |

### Table C.8. Intensity expected

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
<td>R d²</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>5 4</td>
<td>5 4</td>
<td>5 4</td>
<td>6 1</td>
<td>8 1</td>
<td>6 1</td>
<td>5 4</td>
<td>4 9</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>8 4</td>
<td>8 4</td>
<td>8 4</td>
<td>8 4</td>
<td>9 9</td>
<td>8 4</td>
<td>9 9</td>
<td>8 4</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>7 0</td>
<td>4 9</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>9 64</td>
<td>9 64 9 64</td>
<td>9 64 9 64</td>
<td>5 16</td>
<td>9 64 8 49</td>
<td>9 64 9 64</td>
<td>9 64 9 64</td>
<td>9 64 9 64</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6 9</td>
<td>6 9</td>
<td>6 9</td>
<td>6 9</td>
<td>5 4</td>
<td>2 1 5 4</td>
<td>4 1 6 9</td>
<td>6 9</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2 4</td>
<td>3 1 4 0</td>
<td>2 4 3 1</td>
<td>2 4 3 1</td>
<td>2 4 3 1</td>
<td>2 4 3 1</td>
<td>2 4 3 1</td>
<td>2 4 3 1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4 4</td>
<td>4 4 3 1</td>
<td>4 4 3 1</td>
<td>6 4 4 4</td>
<td>6 4 4 4</td>
<td>6 4 4 4</td>
<td>6 4 4 4</td>
<td>6 4 4 4</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>3 16</td>
<td>2 25 2 25</td>
<td>3 16</td>
<td>6 1 3 16</td>
<td>3 16</td>
<td>2 25 2 25</td>
<td>3 16</td>
<td>2 25 2 25</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>1 36</td>
<td>1 36 1 36</td>
<td>1 36</td>
<td>1 36</td>
<td>1 36</td>
<td>1 36</td>
<td>1 36</td>
<td>1 36</td>
</tr>
</tbody>
</table>

| Sum     | 166| 172| 168| 158| 94| 158| 160| 182| 196 |
| Spearman| 0.01| 0.04| 0.02| 0.04| 0.43| 0.04| 0.03| 0.10| 0.19 |
### Table C.9. Smaller Markov windows (1-3, best 2)

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>10</td>
<td>64</td>
<td>10</td>
<td>24</td>
<td>8</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td></td>
<td></td>
<td>0.95</td>
<td>0.98</td>
<td>0.99</td>
<td>0.94</td>
<td>0.94</td>
<td>0.85</td>
<td>0.95</td>
<td>0.77</td>
</tr>
</tbody>
</table>

### Table C.10. Markov mean (averaged over all words for each window size – best window size 5)

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>8</td>
<td>12</td>
<td>14</td>
<td>6</td>
<td>50</td>
<td>6</td>
<td>22</td>
<td>18</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td></td>
<td></td>
<td>0.95</td>
<td>0.93</td>
<td>0.92</td>
<td>0.96</td>
<td>0.96</td>
<td>0.87</td>
<td>0.89</td>
<td>0.77</td>
</tr>
</tbody>
</table>

### Table C.11. Markov expected (window size 8)

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
<td>d²</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>21</td>
<td>15</td>
<td>13</td>
<td>15</td>
<td>43</td>
<td>15</td>
<td>19</td>
<td>19</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td></td>
<td></td>
<td>0.87</td>
<td>0.91</td>
<td>0.92</td>
<td>0.91</td>
<td>0.74</td>
<td>0.91</td>
<td>0.88</td>
<td>0.88</td>
</tr>
</tbody>
</table>
### Table C.12. Two State Hidden Markov Model

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>#</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td>16</td>
<td>76</td>
<td>16</td>
<td>36</td>
<td>16</td>
<td>36</td>
</tr>
</tbody>
</table>

| Spearman | 0.93 | 0.94 | 0.96 | 0.90 | 0.54 | 0.90 | 0.78 | 0.90 | 0.78 |

### Table C.13. Three State Hidden Markov Model

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>52</td>
<td>2</td>
<td>10</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

| Spearman | 1.00 | 0.99 | 0.96 | 0.99 | 0.68 | 0.99 | 0.94 | 0.98 | 0.89 |

### Table C.14. Distance and count half window plus one (window size 4)

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>8</td>
<td>6</td>
<td>10</td>
<td>14</td>
<td>78</td>
<td>14</td>
<td>16</td>
<td>2</td>
<td>20</td>
</tr>
</tbody>
</table>

| Spearman | 0.95 | 0.96 | 0.94 | 0.92 | 0.53 | 0.92 | 0.90 | 0.99 | 0.88 |
Table C.15. Distance and count expected (window size 4)

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7</td>
<td>16</td>
<td>7</td>
<td>16</td>
<td>7</td>
<td>16</td>
<td>7</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>2</td>
<td>16</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>16</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>45</td>
<td>43</td>
<td>35</td>
<td>53</td>
<td>97</td>
<td>53</td>
<td>67</td>
<td>45</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td>0.73</td>
<td>0.74</td>
<td>0.79</td>
<td>0.68</td>
<td>0.41</td>
<td>0.68</td>
<td>0.59</td>
<td>0.73</td>
<td>0.69</td>
<td></td>
</tr>
</tbody>
</table>

Table C.16. Neighbours

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>15</td>
<td>9</td>
<td>5</td>
<td>21</td>
<td>79</td>
<td>21</td>
<td>29</td>
<td>9</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td>0.91</td>
<td>0.95</td>
<td>0.97</td>
<td>0.87</td>
<td>0.52</td>
<td>0.87</td>
<td>0.82</td>
<td>0.95</td>
<td>0.76</td>
<td></td>
</tr>
</tbody>
</table>

Table C.17. Neighbours expected

<table>
<thead>
<tr>
<th>measure</th>
<th>R</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>5</td>
<td>16</td>
<td>5</td>
<td>16</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>10</td>
<td>49</td>
<td>10</td>
<td>49</td>
<td>10</td>
<td>49</td>
<td>10</td>
<td>49</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>25</td>
<td>8</td>
<td>25</td>
<td>8</td>
<td>25</td>
<td>9</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>1</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>16</td>
<td>2</td>
<td>16</td>
<td>3</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sum</td>
<td>113</td>
<td>121</td>
<td>123</td>
<td>107</td>
<td>109</td>
<td>107</td>
<td>131</td>
<td>133</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>Spearman</td>
<td>0.32</td>
<td>0.27</td>
<td>0.25</td>
<td>0.35</td>
<td>0.34</td>
<td>0.35</td>
<td>0.21</td>
<td>0.19</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>
**Table C.18. L-function sum**

<table>
<thead>
<tr>
<th>Measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td>22</td>
<td>26</td>
<td>28</td>
<td>24</td>
<td>86</td>
<td>24</td>
<td>52</td>
<td>34</td>
</tr>
<tr>
<td>spearman</td>
<td></td>
<td>0.87</td>
<td>0.84</td>
<td>0.83</td>
<td>0.85</td>
<td>0.48</td>
<td>0.85</td>
<td>0.68</td>
<td>0.79</td>
</tr>
</tbody>
</table>

**Table C.19. L-function largest**

<table>
<thead>
<tr>
<th>Measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>8</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td>25</td>
<td>6</td>
<td>25</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td>62</td>
<td>62</td>
<td>62</td>
<td>48</td>
<td>46</td>
<td>48</td>
<td>58</td>
<td>76</td>
</tr>
<tr>
<td>spearman</td>
<td></td>
<td>0.62</td>
<td>0.62</td>
<td>0.62</td>
<td>0.71</td>
<td>0.72</td>
<td>0.71</td>
<td>0.65</td>
<td>0.54</td>
</tr>
</tbody>
</table>

**Table C.20. L-function expected**

<table>
<thead>
<tr>
<th>Measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td>32</td>
<td>36</td>
<td>38</td>
<td>42</td>
<td>114</td>
<td>42</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>spearman</td>
<td></td>
<td>0.81</td>
<td>0.78</td>
<td>0.77</td>
<td>0.75</td>
<td>0.31</td>
<td>0.75</td>
<td>0.64</td>
<td>0.76</td>
</tr>
</tbody>
</table>

236
### Table C.21. Chi-squared

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>22</td>
<td>28</td>
<td>36</td>
<td>24</td>
<td>52</td>
<td>24</td>
<td>16</td>
<td>22</td>
<td>40</td>
</tr>
<tr>
<td>Spearman</td>
<td>0.87</td>
<td>0.83</td>
<td>0.78</td>
<td>0.85</td>
<td>0.68</td>
<td>0.85</td>
<td>0.90</td>
<td>0.87</td>
<td>0.76</td>
</tr>
</tbody>
</table>

### Table C.22. CUSUM

<table>
<thead>
<tr>
<th>measure</th>
<th>C</th>
<th>subject 1</th>
<th>subject 2</th>
<th>subject 3</th>
<th>subject 4</th>
<th>subject 5</th>
<th>subject 6</th>
<th>subject 7</th>
<th>subject 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
<td>d²</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sum</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>55</td>
<td>7</td>
<td>23</td>
<td>11</td>
<td>35</td>
</tr>
<tr>
<td>Spearman</td>
<td>0.97</td>
<td>0.97</td>
<td>0.98</td>
<td>0.96</td>
<td>0.67</td>
<td>0.96</td>
<td>0.86</td>
<td>0.93</td>
<td>0.79</td>
</tr>
</tbody>
</table>
APPENDIX D – PROTOTYPE

Chapter 7 detailed the evaluation of the proposed clumpiness measures resulting in the selection of the measure that appeared most suitable in laboratory conditions, specifically, that performed best on the synthetic data as defined by specified criteria. However, it is also necessary to test this method in a more realistic situation, i.e. in BLAST results retrieved by scientists in the course of their research. To this end, a prototype was created incorporating this ‘best’ method, specifically the CUSUM measure. In this case a prototype being an automation, or semi-functional program that would conduct BLAST searches and limited subsequent analysis, returning results to the scientist on which they could draw conclusions.

D.1 Purpose of prototype

The purpose of the prototype is two-fold:

1. To demonstrate the validity of the chosen measure to quantify the degree of clumpiness within a protein hit alignment
2. To demonstrate the ability of a system containing a clumpiness measure to assist in the function determination of protein sequences

In the first instance, the measure has been evaluated as to its ability to validly quantify clumpiness within a simulated dataset, Section 7. By incorporating the measure into a prototype, the measure may also be evaluated as to its ability to validly quantify the clumpiness within ‘real life’ protein sequences with all the variations in length and intensity that will be included in these sequences. In essence, the previous validation was based on lines with only one influence, that of clumpiness, all others being kept equal. Here, these additional influences have been re-introduced by conducting actual BLAST searches and analysing the results.

The second purpose is to evaluate whether the analysis of the clumpiness within an alignment is, in fact, helpful in the determination of protein function. It is whether this additional piece of analysis of the BLAST results would assist the scientists in their research that is under question.

D.2 Prototype design

Section 4.2.3 described the characteristics of BLAST used by scientists to determine whether a particular BLAST result was of relevance or not, summarised in Figure 4.3. Here Figure D.1 illustrates the basic steps behind this decision-making process, that of submission, analysis, information retrieval and final relevance determination, the outcomes of which being used to make a functional decision about a particular query sequence. The idea behind any analysis automation is not to replace the underlying decision-making but to augment it by giving as much information displayed in an easily interpreted manner. This may be seen in Figure D.2 where the submission and subsequent decision-making are the same as in Figure D.1 but the initial results to be analysed are altered with the intention that the scientist be able to conduct more informed reasoning.
Figure D.1. Representation of the basic steps in the decision-making process

Figure D.2. Alteration of the decision-making steps for the addition of a prototype
D.2.1 System design

A prototype, by definition, may not have the complete functionality that is envisaged of the finished program but will exhibit the "key functionality of the target product" (Schach 1993). Since the purpose of the prototype is to evaluate the clumpiness measure the key features of this prototype are:

1. To obtain the results generated by the BLAST program for a particular test sequence
2. To analyse the data and return a clumpiness score for each individual hit within these results
3. To report the findings back to the user

This has been summarised in Figure D.3.

![Diagram](image)

**Figure D.3.** Diagrammatic representation of prototype functions

In addition, it would be desirable if the prototype also:

a. Acquired information of the user for the purposes of data retrieval and protection
b. Acquired data concerning the query sequence's species for taxonomic purposes
c. Identified keywords, such as proteins, genes or species within the descriptor line of the BLAST results so as to highlight these for the users benefit
d. Evaluated any user-defined score or e-value thresholds and warn the user if any BLAST results are outwith these thresholds

The first two of these, that of acquiring information on the user and query sequence species, may be easily accomplished during the data entry process. Fields could be provided to enable the user to enter details for identification purposes and the species of the query sequence, if known. The identification information could then be used to enable a user to access previous results and restrict access to results a user deems sensitive with regards to intellectual property.

---

In the descriptors, checking for a species’s name is simply a case of checking each word in the description against the taxonomic database at NCBI, which is made easier by NCBI’s taxonomy database link (Bischoff et al. no date)\(^2\). The descriptions could also be checked for protein and gene names using NCBI’s protein and gene search engines respectively. For these checks small words like ‘a’, ‘it’, ‘as’ would first need to be removed so as to reduce the time spent checking non-relevant words. Also, words that indicate that the function definition, if any, has not been proven by biochemical testing, such as ‘putative’ or ‘hypothetical’, could also be removed and a flag noted to the user to warn them.

The final additional feature is the checking of score and e-value against user-defined thresholds, which could also be entered during the data entry process. The scores and e-values of each of the result hits would then be compared to these thresholds and those results below the specified score or above the specified e-value would be flagged to warn the user.

In summary, the prototype would

1. Identify the user
2. Send a BLAST query
3. Retrieve the BLAST results
4. Perform the additional analysis elements described above
5. Conduct the clumpiness assessment on the alignments
6. Generate a report for the user’s perusal.

as can be seen in the data flow diagram in Figure D.4.

D.2.2 User interface design
The primary purpose of the prototype is to test the clumpiness measure with ‘real’ data, however, it is also necessary that the program be user friendly. Schach (1993)\(^3\) defines user friendliness as “the ease with which human beings can communicate with the software product”; meaning that the functionality should be easily accessible and usable and any visual elements should not by confusing or irritating to the eye.

In the prototype, the colour aesthetics were not deemed a priority provided the basic appearance and layout were easy to use. Figure D.5 shows the initial input screen, which uses the colour schemes carried through from the windows options and all menus and buttons are clearly labelled. Also, since the prototype was designed for BLAST experts to use, no help files were created to explain any aspect of the BLAST process. However, a manual (Appendix F) was written to assist the users in installing the program, the steps for inputting data, submitting to BLAST and a description of the layout of the report generated.

D.3 Prototype implementation
The prototype has been implemented in Visual Basic 6 and, as specified in Figure D.4, has six main processes:

1. Check if new user
2. Prepare for BLAST
3. Process results
4. Process score and description
5. Process alignments
6. Generate report

All data is stored in an Access 2000 database with five tables, Figure D.6. One table holds all the data on each of the users; another the specifics for each test sequence; the third contains the specifics for instances of each BLAST search and the fourth holding the data concerning each individual result. These first four tables are simply stores for data produced in the current, manual system; however, the fifth table contains data generated by the prototype. Although these fields from the calculation table could feasibly be incorporated into the results table with these two tables having a one-to-one relationship, it was felt that keeping them separate would allow for less confusion during implementation and for future development.
D.3.1 Check if new user
To check whether a user is already entered in the database, ‘surname’ and then ‘forename’ fields in the User table are scanned against those entered by the user. If there is already that surname and forename in the database the user is informed that that name is already present and asks if they are an existing user. If not, they are prompted to re-enter their name, however, it would have been better to alter the user ID to differentiate between users with the same name. The user ID is used for the purposes of the primary key in the user table of the database, highlighted purple in Figure D.6. The user ID is made up of the user’s initials plus the day of the year and the last two digits of the year, therefore John Smith first using the program on the 5th January 2004, would have a user ID of JS504. Therefore, if two John Smiths first used the program on different days there would be no problem, but if it were the same day an additional digit could be added to the end so there was a JS504 and a JS5041. This method could also be used to ensure that no user ID were the same, for example, if Jane Saunders also used the program on the 5th January 2004 her ID would need to be JS5042. This could be done by looking first if there were any other JS504s then searching for a JS5041, JS5042 and so on until the search returned negative, at which point that would be the user ID.

If the user were asked every time they used the program if they were a new user, this could very quickly become irritating, therefore, existing users can select their names from the dropdown list, which by-passes the name check procedure, Figure D.7.

D.3.2 Prepare for BLAST
Due to implementation issues, to be discussed below, the submission to BLAST is not fully automated leading to a slightly tedious process, which still requires some
user interaction. The query sequence is copied from the input screen but the user must still copy it into the BLAST search box, labelled ‘2’ in Figure D.8. They would then click the BLAST button, labelled ‘3’, which is the same process as if the user was conducting a normal BLAST search. Once the BLAST results have appeared in the main window, the user would click the capture button, label ‘4’ and the program would continue automatically with no further need for user interaction until the report page is produced.

D.3.3 Process results
This process parses all the details from the BLAST results to the databases tables ‘BLAST’ and ‘Results’ using SQL code. Textual cues are used to identify the relevant data in the BLAST results, which are then parsed to the relevant database table. With reference to Figure D.9, these are:

1. The BLAST type and version, found on the first non-blank line and parsed until the square bracket, ‘[‘, labelled ‘1’ in the figure, into the BLAST table under the same names
2. The BLAST reference number, cued by ‘RID:’, labelled ‘2’ in the figure, and parsed from the rest of that line into the BLAST table under ‘RID’
3. The database used, cued by ‘database:’, labelled ‘3’ in the figure, and parsed from the rest of that line into the BLAST table under ‘Database’
4. The number of results, cued by ‘Distribution of’, labelled ‘4’ in the figure, and parsed until the next space into the BLAST table under ‘NoResults’

---

Figure D.8. Screen shot for BLAST submission, (NCBI no date)⁴

---

Once the total number of results is known, a user-defined array is created to hold all the details prior to depositing them in the database tables. Each instance of the array must contain all relevant fields of the tables including the query sequence ID, BLAST date and result number, which forms the primary key for both tables. Since the description in the hits list is often truncated and all relevant information is also detailed in the alignments section of the BLAST result, the hits list is by-passed and parsing continues at the start of the alignments.

Firstly the accession numbers are parsed using the upright, ‘|’, as guides, though these have been converted to ‘:’ since SQL does not tolerate uprights. Then the description is parsed, checking that it has not been wrapped onto the next line by checking for other uprights, which would indicate another accession number, or the phrase ‘Length =’, which is the next line from which to parse the matching sequence’s length. The next two lines to parse contain the score and e-value and the number of identical matches, positive matches and gaps, which are located using the phrases ‘Score =’ and ‘Identities =’ respectively.

Due to, in the most part, the length of the alignments, they are generally found on multiple lines. Therefore, to enable reassembly of the alignment into the report file a colon separates each line, ‘:’. So an alignment would appear in the database as:

```
Query section : similarity line : matching sequence section : Query section :
... etc.
```
The colons are then used to split the alignment back to:

Query section
similarity line
matching sequence section
Query section
... etc.

During this process the similarity line between the query and matching sequences is copied to another variable and converted to binary, where a identity and positive are a 1 and a gap is 0. This is necessary as the clumpiness measure works only on a binary line.

The end of the alignment is indicated by the presence of either ‘>’, which indicated the next hit, ‘Score =’, which indicates a second alignment for the same hit, or ‘Database:’, which indicates the end of the results file. If there is more than one matching section between the query and matching sequences the BLAST program will place these sequentially in the alignments section even if they would appear distant on the hits list. Therefore, the prototype does not store the results in hit list order but alignment order, however, since the hit list order can be altered by the clumpiness measure, this was deemed to be unimportant. If there are multiple alignments for one matching sequence they do not repeat the accession number and description information, therefore, these need to be copied from the previous entry.

Additionally, once the total number of results has been parsed, the score bar, labelled ‘5’ in Figure D.8 can be set so as to keep track of the progress of the calculations. This is achieved by setting the scrollbars maximum value to the total number of results and shifting it along by one once each result has been processed.

D.3.4 Process score and description
This process checks the score and e-value against any thresholds the user may have set on the initial input page. For each hit the process enters the word ‘use’ in the 'ScoreEvalue' field of the calculations table if they are inside the set threshold and ‘ignore’ if they are outwith. Note that any words could be used as these are only cues used to determine the colour of the results in the report, black for OK results and red for those outwith the thresholds.

This process also deals with the descriptor lines in the BLAST results, searching for keywords to flag for the notice of the user. Firstly, all small words, column one of Table D.1, are removed from the description since these will not give any relevant information if looked for in the online databases. Next words that would indicate that that particular result is not founded in any experimental data, column two of Table D.1, are also removed, though, ‘hypothetical’, or ‘unknown’ for the last two, is added to the keywords field of the calculations table. Also removed are the words ‘family’ and ‘domain’, which, though important, would produce unnecessary hits if searched with in the NCBI database, but they are added to the keywords field in the calculations table.
Table D.1. Words to be removed from description

<table>
<thead>
<tr>
<th>Small words</th>
<th>Hypothetical indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>hypothetical</td>
</tr>
<tr>
<td>i</td>
<td>like</td>
</tr>
<tr>
<td>am</td>
<td>-like</td>
</tr>
<tr>
<td>an</td>
<td>possible</td>
</tr>
<tr>
<td>as</td>
<td>potential</td>
</tr>
<tr>
<td>at</td>
<td>predicted</td>
</tr>
<tr>
<td>by</td>
<td>putative</td>
</tr>
<tr>
<td>do</td>
<td>related</td>
</tr>
<tr>
<td>go</td>
<td>-related</td>
</tr>
<tr>
<td>if</td>
<td>similar</td>
</tr>
<tr>
<td>in</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td>unknown</td>
</tr>
<tr>
<td>it</td>
<td>un-named</td>
</tr>
<tr>
<td>no</td>
<td></td>
</tr>
<tr>
<td>of</td>
<td></td>
</tr>
<tr>
<td>on</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>so</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td></td>
</tr>
<tr>
<td>up</td>
<td></td>
</tr>
<tr>
<td>us</td>
<td></td>
</tr>
<tr>
<td>and</td>
<td></td>
</tr>
<tr>
<td>the</td>
<td></td>
</tr>
</tbody>
</table>

Once these words have been removed each word in the description may be used as a search element on the species, gene and protein databases on NCBI. As was mentioned previously, the use of the taxonomic link (Bischoff et al. no date)\(^5\) provides a simple method to determine whether any of the words are that of a species. If it is not a species name the term ‘No results found’ will appear on the web page and it is a simple matter to check for this. If the word is present then an additional search may be conducted to check if there is a full species name in the description by sending the word plus the next word along. For example if the description contains ‘Drosophila melanogaster’ the two searches would be for drosophila and then drosophila+melanogaster. If both are found, both are entered into the species field of the calculation table, but if only one is found, only one is entered.

It was hoped that a similar method could be used for searching for gene and protein names. However, the gene and protein search methods will find any word that is present in the database, such as ‘protein’, ‘regulatory’ or ‘transcription’. Although these may be of some interest to the user, they are not the specific protein and gene names that it was hoped would be located. The EASY application (Selley et al.\(^5\)Linking to the NCBI Taxonomy Database. [online]. NCBI. Available from: http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=howlink [Accessed 25/07/07].

2001)\(^6\) seems to have resolved this in a more efficient manner. However, for the purposes of the prototype this issue was not deemed a priority and not resolved at this time and not included in the prototype.

**D.3.5 Process alignments**

As mentioned in Section 0 above and described in Section 5, this process uses a binary line created from the similarity line between the query and matching sequences, Figure D.10.

![Match found by BLAST](image)

**Figure D.10. Conversion of a similarity line to binary**

The measure used, CUSUM, was described in detail in Section 6.3.2, here we describe, in pseudo-code, the implementation of that measure:

For reference:

\[
C_i = \max \{0, y_i - k + C_{i-1}\},
\]

where:

\[
y = \frac{\text{value at position - mean}}{\text{standard deviation}}
\]

\[
\text{mean} = \frac{\text{No. ones}}{\text{binary line length (BLL)}}
\]

\[
\text{Standard deviation} = \sqrt{\frac{1}{(\text{BLL} - 1)} \times \left( \frac{\text{No. ones} - (\text{No. ones})^2}{\text{BLL}} \right)}
\]

\[
H = h \times \text{standard deviation}
\]

\[
K = k \times \text{standard deviation}
\]

And:

\[
h = 4
\]

\[
k = 0.6
\]

---

Therefore:
For each position in the binary line
  If standard deviation is 0 then
    \[ y = \text{value at position} \ (0 \text{ or } 1) \]
  Else
    \[ y = (\text{value at position} - \text{mean}) / \text{standard deviation} \]

If it is first position then
  \[ \text{CUSUM value at position} \ 1 = y - K + (H / 2) \]
Else
  \[ \text{CUSUM value at position} = y - K + \text{CUSUM value at previous position} \]

Since the maximum is needed of the CUSUM value or 0:
  If CUSUM value at position is less than 0 then
    \[ \text{CUSUM value at position} = 0 \]
  Else is the CUSUM value

The actual measure is the number of times the CUSUM indicates that the system is clumpier than a randomly distributed line, or out-of-control, therefore:
  If CUSUM value at position is greater than H then
    Increment No. out-of-control by one

This gives the clumpiness value for that particular query and matching sequence. However, not all the hits have the same length, the clumpiness value must be standardised:
  \[ \text{Coverage} = \frac{\text{length of binary line}}{\text{length of query sequence}} \]
  \[ \text{Normalised clumpiness value} = \text{clumpiness value} \times \text{coverage} \]

And this has been rounded to three decimal places.

D.3.6 Generate report
Here, the report returned to the user is generated. The layout is has been kept very close to the layout of an ordinary BLAST result such that users familiar with the BLAST output will find this report easy to read as well. The differences between this report and the original BLAST report, as described in Section 3.2, are:
  • That if a hypothetical type keyword has been found in the description, ‘hypothetical’ has been added in front of the description, Figure D.11.
  • That as well as the score and e-value, there is also the clumpiness value, Figure D.11.
  • That if thresholds were set, any results with scores and/or e-values below them are coloured red, Figure D.11.
  • That the clumpiness value is also added after the score and e-value in the alignments section, Figure D.12.
  • That a keywords section has been added to just above the actual alignment with a species name, if found, and any other keywords thrown up by the protein search engine on the NCBI site, Figure D.12.
**Figure D.11.** Altered hits list in report

**Figure D.12.** Altered alignments section in report

The actual generation of the report consists of retrieving the relevant information from the correct database tables and fields using SQL searches. Firstly, the general BLAST information is retrieved from the BLAST file and inserted into a text file, separated by blank lines. Then inserting the phrase ‘hit list’ followed by the contents of Results table with the addition of the Clumpiness value from the Calculations table generates the hit list. Finally, the alignments are reassembled as shown in Figure D.12 using both the Results and Calculations tables. The actual alignment assembly involves retrieving the data from the Alignment field in the Results table and converting the single line of text to the multiple lines of the alignment using the colons inserted into the lines for this purpose, Section D.3.3.

This report is displayed on the screen along with the original BLAST report for the perusal of the user. The retention of the original BLAST report is primarily for the purposes of the prototype evaluation, which is the topic of Chapter 8.
APPENDIX E – PROTOTYPE CODE

E.1. Forms

E.1.1. Input form - frmProtInput

Option Explicit

'fill comboboxes
Private Sub Form_Load()
Dim FS As String
'open Database
DBFileName = App.Path & "\prototype.mdb"
Set DBs = OpenDatabase(DBFileName)
'fill combo boxes
Set RS = DBs.OpenRecordset("User", dbOpenSnapshot)
Do Until RS.EOF
    cboFore.AddItem RS.Fields("Forename").Value
    cboSur.AddItem RS.Fields("Surname").Value
    RS.MoveNext
Loop
RS.Close
Set RS = Nothing
Set RS = DBs.OpenRecordset("Sequences", dbOpenSnapshot)
Do Until RS.EOF
    cboTestID.AddItem RS.Fields("TestID").Value
    RS.MoveNext
Loop
RS.Close
Set RS = Nothing
FS = App.Path & "\Results"
If Not FileObject.FolderExists(FS) Then
    FileObject.CreateFolder (FS)
End If
FS = App.Path & "\Reports"
If Not FileObject.FolderExists(FS) Then
    FileObject.CreateFolder (FS)
End If
End Sub

'fill text boxes with seq details if picked from combobox
Private Sub cboTestID_Click()
Dim i As Integer
i = cboTestID.ListIndex
Set RS = DBs.OpenRecordset("Sequences", dbOpenSnapshot)
RS.Move (i)
txtSpecies.Text = RS.Fields("Species").Value
txtSeq.Text = RS.Fields("Sequence").Value
RS.Close
Set RS = Nothing
End Sub

Private Sub cmdRun_Click()
Dim i As Integer
Dim j As Integer
'get values off form
Fname = cboFore.Text
Sname = cboSur.Text
userID = Left(Fname, 1) & Left(Sname, 1) & DatePart("y", Now)
& Right(DatePart("yyyy", Now), 2)

Email = txtEmail.Text
SeqID = cboTestID.Text
Species = txtSpecies.Text
seq = txtSeq.Text
letters = Len(seq)
txtLetters.Text = letters
SThresh = txtSThresh.Text
EThresh = txtEThresh.Text
If optSThresh.Value = True Then
    Pref = "Score"
Else
    Pref = "EValue"
End If
'Enter input data into DB
i = cboFore.ListIndex
j = cboTestID.ListIndex
Call EnterInputData(i, j)
frmProtInput.Hide
frmProtBLAST.Show
End Sub

'clear User information
Private Sub cmdClearUser_Click()
    cboFore.Text = ""
    cboSur.Text = ""
    txtEmail.Text = ""
    cboFore.SetFocus
End Sub

'clear sequence information
Private Sub cmdClearSeq_Click()
    cboTestID.Text = ""
    txtSpecies.Text = ""
    optSThresh.Value = True
    txtSThresh.Text = ""
    txtEThresh.Text = ""
    txtSeq.Text = ""
    txtLetters.Text = ""
    cboTestID.SetFocus
End Sub

'clear whole form
Private Sub cmdReset_Click()
    cboFore.Text = ""
    cboSur.Text = ""
    txtEmail.Text = ""
    cboTestID.Text = ""
    txtSpecies.Text = ""
    optSThresh.Value = True
    txtSThresh.Text = ""
    txtEThresh.Text = ""
    txtSeq.Text = ""
    txtLetters.Text = ""
    cboFore.SetFocus
End Sub

'exit program
Private Sub cmdExit_Click()
    End
End
End Sub

E.1.2. Conduct BLAST search – frmProtBLAST

Option Explicit

'open webpage and put sequence in text box ready for copy and paste
Private Sub Form_Load()
    Dim url As String
    wbrBLAST.Navigate (url)
    txtSeqCopy.Text = seq
End Sub

'once in BLAST results webpage
'select all, copy and write to a text file
Private Sub cmdCapture_Click()
    Dim txt As String
    Dim i As Integer
    Dim maps As Integer
    Dim mape As Integer
    Call wbrBLAST.ExecWB(OLECMDID_SELECTALL, OLECMDEXECOPT_DONTPROMPTUSER)
    Call wbrBLAST.ExecWB(OLECMDID_COPY, OLECMDEXECOPT_DONTPROMPTUSER)
    txt = Clipboard.GetText(vbCFText)
    BDate = Now
    SaveName = App.Path & "\Results\" & SeqID & "Res-" & Format(BDate, "Medium Date") & ".txt"
    SaveObject.CreateTextFile (SaveName) 'for storage until get date
    Set SaveHandle = SaveObject.GetFileName(SaveName)
    Set outstream = SaveHandle.OpenAsTextStream(ForWriting)
    outstream.WriteLine (txt)
    outstream.Close
    Set outstream = Nothing
    'start parsing files and evaluating data
    'open data file
    ResFileName = App.Path & "\Results\" & SeqID & "Res-" & Format(BDate, "Medium Date") & ".txt"
    Set FileHandle = FileObject.GetFileName(ResFileName)
    Set instream = FileHandle.OpenAsTextStream(ForReading)
    'processfile
    i = GetBLAST()
    hscProgress.Max = i + 4
    hscProgress.Value = 1
    ReDim BLSTRes(l To i)
    ReDim Calcs(1 To i)
    Call GetRes
    instream.Close
    Set instream = Nothing
    Call wbrBLAST.ExecWB(OLECMDID_CLEARSELECTION, OLECMDEXECOPT_DONTPROMPTUSER)
    DBs.Close
Set DBs = Nothing
frmProtReport.Show
End Sub

'close window once done
Private Sub cmdClose_Click()
    frmProtInput.Show
    frmProtInput.cmdExit.SetFocus
    Unload Me
End Sub

E.1.3. Keyword Searching – frmProtWords

'locate protein, gene and species words in description
Option Explicit
Dim words() As String

Private Sub Command1_Click()
    ReDim Calcs(1)
    Call FindWords(1)
End Sub

Public Sub FindWords(i As Integer)
    Dim descrip As String
    Dim senten As String
    Dim taxon As Boolean
    Dim space1 As Integer
    Dim space2 As Integer
    Dim j As Integer
    descrip = " " & BLSTRes(i).Desc & " "
    senten = Prepline(descrip, i) 'find 1st space (at start of line)
    space1 = InStr(1, senten, " ")
    space2 = InStr(space1 + 1, senten, " ") 'find next space (till last char)
    ReDim words(0)
    j = 0
    Do While space2 <> 0 'until no more spaces
        If j > UBound(words) Then
            ReDim Preserve words(j)
        End If
        words(j) = Mid(senten, space1 + 1, space2 - (space1 + 1))
        space1 = space2
        space2 = InStr(space1 + 1, senten, " ")
        j = j + 1
    Loop
    For j = 0 To UBound(words)
        If FindSpecies(i, j) = False Then
            If FindGene(i, j) = False Then
                Call FindProtein(i, j)
            End If
        End If
    Next j
End Sub

'remove small words from description
Private Function Prepline(line As String, i As Integer) As String
    Dim RegExObj As New VBScript_RegExp_55.RegExp
    Dim Small() As String
    Dim check() As String
    ...
Dim j As Integer
Call FillSmall(Small())
Call FillCheck(check())
' remove small words
For j = 0 To UBound(Small) ' remove small words
    With RegExObj
        .Pattern = Small(j)
        .IgnoreCase = True
        .Global = True
    End With
    line = RegExObj.Replace(line, " ")
Next j
' check if hypothetical or putative
For j = 0 To UBound(check)
    With RegExObj
        .Pattern = check(j)
    End With
    If RegExObj.Test(line) Then
        Calcs(i).key = "Hypothetical, "
    End If
    line = RegExObj.Replace(line, " ")
Next j
' check if unknown or unnamed
With RegExObj
    .Pattern = " unknown 
End With
If RegExObj.Test(line) Then
    Calcs(i).key = Calcs(i).key & "Unknown, "
End If
line = RegExObj.Replace(line, " ")
With RegExObj
    .Pattern = " unnamed 
End With
If RegExObj.Test(line) Then
    Calcs(i).key = Calcs(i).key & "Unknown, "
End If
line = RegExObj.Replace(line, " ")
' check for family or domain
With RegExObj
    .Pattern = " family 
End With
If RegExObj.Test(line) Then
    Calcs(i).key = Calcs(i).key & "Family, "
End If
line = RegExObj.Replace(line, " ")
With RegExObj
    .Pattern = " domain 
End With
If RegExObj.Test(line) Then
    Calcs(i).key = Calcs(i).key & "Domain, "
End If
line = RegExObj.Replace(line, " ")
Set RegExObj = Nothing
line = Replace(line, "[", ""
line = Replace(line, "]", ""
line = Replace(line, ":", "")
Prepline = line
End Function

' look for species words
Private Function FindSpecies(i As Integer, j As Integer) As Boolean
Dim urls As String
Dim urlf As String
Dim w As Integer
Dim fullname As String
Dim txt As String
urls = "http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?name=
urlf = urls & words(j)
wbrWords.Navigate (urlf)
txt = GetURL(urlf)
If InStr(l, txt, "No result found", vbTextCompare) = 0 Then
    If j <> UBound(words) Then
        fullname = words(j) & "+" & words(j + 1)
        urlf = urls & fullname
        wbrWords.Navigate (urlf)
        txt = GetURL(urlf)
        If InStr(l, txt, "No result found", vbTextCompare) = 0 Then
            Calcs(i).Sps = fullname
            FindSpecies = True
            j = j + 1
        End If
    Else
        Calcs(i).Sps = words(j)
        FindSpecies = True
    End If
Else
    FindSpecies = False
End If
End Function

'look for gene words
Private Function FindGene(i As Integer, j As Integer) As Boolean
Dim urls As String
Dim urlf As String
Dim w As Integer
Dim txt As String
urlf = urls & words(w)
wbrWords.Navigate (urlf)
txt = GetURL(urlf)
If InStr(l, txt, "No items found", vbTextCompare) = 0 Then
    If InStr(l, txt, "1: " & words(j), vbTextCompare) <> 0 Then
        Calcs(i).Gen = Calcs(i).Gen & words(j) & ";"
        FindGene = True
    Else
        FindGene = False
    End If
Else
    FindGene = False
End If
End Function

'look for protein words
Private Sub FindProtein(i As Integer, j As Integer)
Dim urls As String
Dim urlf As String
Dim w As Integer
Dim txt As String
urlf = urls & words(j)
wbrWords.Navigate (urlf)
txt = GetURL(urlf)
If InStr(l, txt, "No items found", vbTextCompare) = 0 Then
    Calcs(i).Prt = Calcs(i).Prt & words(j) & "; "
End If
End Sub

Public Function GetURL(url As String) As String
Dim vtdata As String
Dim txt As String
Dim bdone As Boolean
txt = ""
With InetProt
    .Execute (url)
    Do While .StillExecuting
        DoEvents
    Loop
On Error GoTo BadConnection
    vtdata = .GetChunk(1024, icString)
    DoEvents
    Do While Not bdone
        txt = txt & vtdata
        DoEvents
        ' Get next chunk.
        vtdata = .GetChunk(1024, icString)
        If Len(vtdata) = 0 Then
            bdone = True
        End If
    Loop
End With
GetURL = txt
Exit Function
BadConnection:
    GetURL = "No Result Found"
Exit Function
End Function

Private Sub FillSmall(S() As String)
ReDim S(22)
S(0) = " a "
S(1) = " i "
S(2) = " am "
S(3) = " an "
S(4) = " as "
S(5) = " at "
S(6) = " by "
S(7) = " do "
S(8) = " go "
S(9) = " if "
S(10) = " in "
S(11) = " is "
S(12) = " it "
S(13) = " no "
S(14) = " of "
S(15) = " on "
S(16) = " or "
S(17) = " so "

S(18) = " to "
S(19) = " up "
S(20) = " us "
S(21) = " and "
S(22) = " the "

End Sub

Private Sub FillCheck(C() As String)
    ReDim C(9)
    C(0) = " hypothetical "
    C(1) = " like "
    C(2) = "-like "
    C(3) = " possible "
    C(4) = " potential "
    C(5) = " predicted "
    C(6) = " putative "
    C(7) = " related "
    C(8) = "-related "
    C(9) = " similar "
End Sub

E.1.4. Report creation – frmProtReport

'Generate Report for user
Option Explicit
Dim sqlstring As String
Dim RRS As Recordset 'report recordset
Dim ProtAccess As Access.Application

Private Sub Form_Load()
    'open Database
    DBFileName = App.Path & \"\prototype.mdb"
    Set DBs = OpenDatabase(DBFileName)
    Call CreateReport
End Sub

Private Sub CreateReport()
    SaveName = App.Path & \"\Reports\Temp.html"
    SaveObject.CreateTextFile (SaveName) 'for storage until get date
    Set SaveHandle = SaveObject.GetFileName(SaveName)
    Set outstream = SaveHandle.OpenAsTextStream(ForWriting)
    Call FrontPart
    Call ListPart
    Call AlignPart
    wbrReport.Navigate (SaveName)
    outstream.Close
    Set outstream = Nothing
End Sub

Private Sub FrontPart()
    Dim txt As String
    Dim taxurl As String
txt = "<HTML>" & vbCrLf & "<TITLE>Report Page</TITLE>" & vbCrLf & "<BODY>" & vbCrLf & "<CENTER><H3><U>Report Page for Query Sequence " & SeqID & ", BLASTed on " & BDate & "</H3></U></CENTER>"


BLAST Version : " & BVersion & " & <P>
RID : " & RID & " & Query : " & <B>Query</B> & " & Database : " & BDB & " & <P>
Number of BLAST Hits to the Query Sequence : " & NoRes & " & Score E Clumpiness:

<table>
<thead>
<tr>
<th>Score</th>
<th>Clumpiness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

outstream.WriteLine (txt)
frmProtBLAST.hscProgress.Value = frmProtBLAST.hscProgress.Value + 1

End Sub
If RRS.Fields("ScoreEvalue") = "Ignore" Then
    txt = "<FONT COLOR=""RED"">"
Else
    txt = ""
End If

'link to alignments below
    & RRS.Fields("Accession1") & ":</A> " & RRS.Fields("Keywords")

'want score starting at pos 72
where = Len(RRS.Fields("Accession1")) + Len(RRS.Fields("Keywords")) + 7
DHold = RRS.Fields("Description")
DLen = Len(DHold)
If DLen > (72 - where) Then
    DHold = Left(DHold, (72 - where) - 3) & "...
ElseIf DLen < (72 - where) Then
    For i = where + DLen To 71
        DHold = DHold + " 
    Next i
End If

txt = txt & DHold & " <A HREF="# & RRS.Fields("Accession1") & ">" & RRS.Fields("Score") & ">" & RRS.Fields("Score") & ":</A> " & Format(RRS.Fields("E-value"), "Scientific") & " " & CValue

'end red font after each (may not be in order at end)
If RRS.Fields("ScoreEvalue").Value = "Ignore" Then
    txt = txt & "</FONT>"
End If
outstream.WriteLine (txt)
RRS.MoveNext
Loop
    txt = "</PRE>" & vbCrLf & "<CENTER><B><FONT color=""green"">Alignments</FONT></B></CENTER>" & vbCrLf & "<P>" & 
vbCrLf & "" & outstream.WriteLine (txt)
    RRS.Close
Set RRS = Nothing
frmProtBLAST.hscProgress.Value = frmProtBLAST.hscProgress.Value + 1
End Sub

Private Sub AlignPart()
Dim txt As String
Dim line As String
Dim dummy As Boolean
Dim spacer1 As Integer
Dim spacer2 As Integer
Dim CValue As String
Dim where As Integer
Dim DLen As Integer
Dim DHold As String
Dim DRest As String
frmProtBLAST.hscProgress.Value = frmProtBLAST.hscProgress.Value + 1
End Sub

Private Sub cmdSave_Click()
Dim txt As String
Call wbrReport.ExecWB(OLECMDID_SELECTALL,
OLECMDEXECOPT_DONTPROMPTUSER)
Call wbrReport.ExecWB(OLECMDID_COPY,
OLECMDEXECOPT_DONTPROMPTUSER)
txt = Clipboard.GetText(vbCFText)
SaveName = App.Path & "\Reports\" & SeqID & "Res-" & Format(BDate, "Medium Date") & ".txt"
SaveObject.CreateTextFile(SaveName), 'for storage until get date
Set SaveHandle = SaveObject.GetFile(SaveName)
Set outstream = SaveHandle.OpenAsTextStream(ForWriting)
outstream.WriteLine(txt)
outstream.Close
Set outstream = Nothing
End Sub

Private Sub cmdExit_Click()
Dim FN As String
Dim sure As String
FN = App.Path & "\Reports\Temp.html"
sure = MsgBox("Would you like to save a copy first?", vbYesNoCancel + vbExclamation, "Are you sure?"")
If sure = vbYes Then
Call cmdSave_Click
SaveObject.DeleteFile(FN)
DBs.Close
Set DBs = Nothing
frmProtBLAST.cmdClose.SetFocus
Unload Me
ElseIf sure = vbNo Then
SaveObject.DeleteFile(FN)
DBs.Close
Set DBs = Nothing
frmProtBLAST.cmdClose.SetFocus
Unload Me
ElseIf sure = vbCancel Then
Exit Sub
End If
End If
End Sub

E.2. Modules

E.2.1. Global variables – modProtGlobal

Option Explicit
' user-defined types for Results info
Type ResultsInfo
' fields of results table
    TestID As String
    BDate As Date
    ResNo As Integer
    Acc1 As String
    Acc2 As String
    Desc As String
    Length As Integer
    Score As Integer
End Type
Public DBFileName As String
Public DBs As Database
Public RS As Recordset

'vars for file access
Public ResFileName As String
Public FileObject As New FileSystemObject
Public FileHandle As File
Public instream As TextStream

'vars for file saving
Public SaveName As String
Public SaveObject As New FileSystemObject
Public SaveHandle As File
Public outstream As TextStream

'vars holding inputted data
Public userID As String
Public FName As String
Public SName As String
Public Email As String
Public SeqID As String
Public species As String
Public seq As String
Public letters As Integer
Public SThresh As String
Public ETthresh As String
Public Pref As String
Public BDate As Date
Public RID As String
Public BDB As String
Public BType As String
Public BVersion As String
Public NoRes As Integer

'vars for processing data
Public lastline As String
Public BLSTResO As Resultslnfo
Public Calcs() As Calclnfo

E.2.2. Initial data entry – modProtEntryInput
'put user and sequence details in DB
Public Sub EnterlnputData(Ni As Integer, IDi As Integer)
Dim sqlstring As String 'to add to DB
Dim r As Integer
Dim temp As String
Dim i As Integer
'store user info
Set RS = DBs.OpenRecordset("User", dbOpenDynaset)
If Ni = -1 Then
    Do While CheckNames(Fname, Sname) = True
        r = MsgBox("Name already present" & vbCrLf & "Existing User?", vbYesNo + vbQuestion, "Query")
        If r = vbNo Then
            temp = InputBox("Enter new name ", "User name")
            i = InStr(1, temp)
            Fname = Left(temp, i - 1)
            Sname = Mid(temp, i + 1)
        Else
            Exit Do
        End If.
    Loop
End If
'INSERT INTO User VALUES ('userlD', 'Fname', 'Sname', 'Email')
sqlstring = "INSERT INTO User VALUES (" & userlD & ", & Fname & ", '  " & Sname & ", " & Email & ")"
DBs.Execute(sqlstring)
RS.Close
Set RS = Nothing
'store sequence info
Set RS = DBs.OpenRecordset("Sequences", dbOpenDynaset)
If IDi = -1 Then
    Do While CheckID() = True
        r = MsgBox("ID already in use" & vbCrLf & "New BLAST for existing sequence?", vbYesNo + vbQuestion, "Query")
        If r = vbNo Then
            SeqID = InputBox("Enter New ID ", "TestID")
        End If.
    Loop
End If
'INSERT INTO Sequences VALUES ('SeqID', 'userlD', letters, 'seq', 'species')
sqlstring = "INSERT INTO Sequences VALUES ('" & SeqID & ", & userlD & ", " & letters & ", & seq & ", " & Species & ")"
DBs.Execute(sqlstring)
RS.Close
Set RS = Nothing
End Sub

'check if names are in DB already
Private Function CheckNames(FN As String, SN As String) As Boolean
Dim temp As String
Dim temp1 As String
Dim sqlstring As String
On Error GoTo NoRecords
RS.MoveFirst
Do Until RS.EOF = True
    temp = RS.Fields("Surname")
    If temp = SN Then 'first record 'end of file 'get record from field 'check if same
NoRecords: Exit Function
End If
End Sub
tempi = RS.Fields("Forename")
If tempi = FN Then
    CheckNames = True 'both same
    Exit Function
Else
    CheckNames = False 'one same
End If
Else
    CheckNames = False 'neither same
End If
RS.MoveNext
'next record
Loop
NoRecords:
    CheckNames = False
Exit Function
End Function

'make sure unique testid
Private Function CheckID() As Boolean
Dim temp As String
On Error GoTo NoRecords
RS.MoveFirst
Do Until RS.EOF
    temp = RS.Fields("TestID")
    If temp = SeqlD Then
        CheckID = True
    Else
        CheckID = False
    End If
    RS.MoveNext
Loop
NoRecords:
    CheckID = False
Exit Function
End Function

E.2.3. Store data in database – modProtF2DB

'Store data in database
Option Explicit

'retrieve BLAST details and put into database
Public Function GetBLAST() As Integer
Dim dummy As Boolean
Dim i As Integer
Dim sqlstring As String
    'find BLAST type and version (1st line with text on)
Do While lastline = " " Or lastline = ""
    dummy = GetALine
Loop
i = InStr(lastline, " ")
BType = Left(lastline, i - 1)
BVersion = Mid(lastline, i + 1)
    'get RID
Do While InStr(1, lastline, "RID", vbTextCompare) = 0
    dummy = GetALine
Loop
RID = Mid(lastline, 6)
    'get database used
Do While InStr(1, lastline, "Database:", vbTextCompare) = 0
    dummy = GetALine

Loop
BDB = Mid(lastline, 11)
'No of results
Do While InStr(1, lastline, "Distribution of", vbTextCompare) = 0
  dummy = GetALine
Loop
i = InStr(17, lastline, " ")
NoRes = Val(Mid(lastline, 17, i - 1))
'send to database
Set RS = DBs.OpenRecordset("BLAST", dbOpenDynaset)
'INSERT INTO BLAST
'VALUES('SeqID', #BDate#, 'RID', 'BDB', 'BType', 'BVersion',
Nores)
sqlstring = "INSERT INTO BLAST VALUES('"
DBs.Execute (sqlstring)
GetBLAST = NoRes
RS.Close
Set RS = Nothing
End Function

'retrieve Results from BLAST results file
Public Sub GetRes()
Dim dummy As Boolean
Dim i As Integer
Dim hold1 As Integer
Dim hold2 As Integer
Dim hold3 As Integer
Dim hold4 As Integer
Dim hold5 As Integer
Dim c1 As Integer
Dim c2 As Integer
Dim ctot As Integer
'start of each match
Do While InStr(1, lastline, ">") = 0
  dummy = GetALine
Loop
'for each result
For i = 1 To UBound(BLSTRes)
  'store testID, BLASTDate and ResultNo
  BLSTRes(i).TestID = SeqlD
  Calcs(i).TestID = SeqlD
  BLSTRes(i).BDate = BDate
  Calcs(i).BDate = BDate
  BLSTRes(i).ResNo = i
  Calcs(i).ResNo = i
  'find Accession line with description and length of matching seq
  hold1 = InStr(1, lastline, ":")
  hold2 = InStr(hold1 + 1, lastline, ":")
  BLSTRes(i).Accl = Mid(lastline, 2, hold2 - 2)
  hold3 = InStr(hold2 + 1, lastline, ":")
  hold4 = InStr(hold3 + 1, lastline, ":")
  BLSTRes(i).Acc2 = Mid(lastline, hold2 + 1, (hold4 - 1) -
hold2)
  hold5 = InStr(hold4 + 1, lastline, ":")
  BLSTRes(i).Desc = LTrim(Mid(lastline, hold5 + 1))
dummy = GetALine
Do While InStr(1, lastline, ":", vbTextCompare) = 0 And _
InStr(l, lastline, "Length = ", vbTextCompare) = 0
BLSTRes(i).Desc = BLSTRes(i).Desc & LTrim(lastline)
dummy = GetALine
Loop
Do While InStr(l, lastline, "Length = ", vbTextCompare) = 0
dummy = GetALine
Loop
holdl = InStr(l, lastline, ")")
BLSTRes(i).Length = LTrim(Mid(lastline, hold1 + 1))
'Score, E-value, identities, positives and gaps
Call SEPIG(i)
'put alignments into file
Call Aligns(i)
Call EvalScore(i)
Call frmProtWords.FindWords(i)
Call Clumps(i)
Call DBEntry(i)
If InStr(l, lastline, "Database:") <> 0 Then
Exit Sub
End If
frmProtBLAST.hscProgress.Value =
frmProtBLAST.hscProgress.Value + 1
Do While InStr(l, lastline, ")") = 0
'more than one alignment for an accession No.
If InStr(l, lastline, "Score = ") <> 0 Then
i = i + 1
BLSTRes(i).TestID = SeqID
Calcs(i).TestID = SeqID
BLSTRes(i).BDate = BDate
Calcs(i).BDate = BDate
BLSTRes(i).ResNo = i
Calcs(i).ResNo = i
BLSTRes(i).Accl = BLSTRes(i - 1).Accl
BLSTRes(i).Acc2 = BLSTRes(i - 1).Acc2
BLSTRes(i).Desc = BLSTRes(i - 1).Desc
BLSTRes(i).Length = BLSTRes(i - 1).Length
Call SEPIG(i)
Call Aligns(i)
End If
'put in database
Call EvalScore(i)
Call frmProtWords.FindWords(i)
Call Clumps(i)
'put in database
Call DBEntry(i)
frmProtBLAST.hscProgress.Value =
frmProtBLAST.hscProgress.Value + 1
Loop
Next i
outstream.Close
Set outstream = Nothing
End Sub

'score, e-value, identities, positives and gaps
Private Sub SEPIG(i As Integer)
Dim dummy As Boolean
Dim hold1 As Integer
Dim hold2 As Integer
Dim hold3 As Integer
Dim hold4 As Integer
Dim hold5 As Integer
'find score and e-value
Do While InStr(1, lastline, "Score = ") = 0
dummy = GetALine
Loop
hold1 = InStr(1, lastline, "=")
hold2 = InStr(hold1, lastline, "b")
BLSTRes(i).Score = LTrim(Mid(lastline, hold1 + 1, (hold2 - 1) - (hold1 + 1)))
hold3 = InStr(hold2, lastline, ")
BLSTRes(i).Evalue = Val(LTrim(Mid(lastline, hold3 + 1)))
'identities, positives and gaps
dummy = GetALine
hold1 = InStr(1, lastline, ")
hold2 = InStr(hold1, lastline, ",")
BLSTRes(i).Idents = LTrim(Mid(lastline, hold1 + 1, hold2 - (hold1 + 1)))
hold3 = InStr(hold2, lastline, ")
If InStr(hold3, lastline, ",", vbTextCompare) = 0 Then
    BLSTRes(i).Posit = LTrim(Mid(lastline, hold3 + 1))
    BLSTRes(i).Gaps = 0
Else
    hold4 = InStr(hold3, lastline, ",")
    BLSTRes(i).Posit = LTrim(Mid(lastline, hold3 + 1, hold4 - (hold3 + 1)))
    hold5 = InStr(hold4, lastline, ")
    BLSTRes(i).Gaps = LTrim(Mid(lastline, hold5 + 1))
End If
End Sub

' storing alignment
Public Sub Aligns(i As Integer)
Dim dummy As Boolean
Dim cl As Integer
Dim c2 As Integer
Dim c3 As Integer
Dim ctot As Integer
Dim x As Integer
Dim y As Integer
Dim similine As String
Dim txt As String
Do While InStr(1, lastline, "Query:") = 0
dummy = GetALine
Loop
c1 = InStr(1, lastline, ">")
c2 = InStr(1, lastline, "Score = ")
c3 = InStr(1, lastline, "Database:")
ctot = c1 + c2 + c3
txt = ""
Do While ctot = 0
    If InStr(1, lastline, "Query:") <> 0 Then
        'store query line
        txt = txt & lastline & " |"
        'find the first space after the sequence counter
        x = InStr(8, lastline, " ", vbTextCompare)
        'find next non-space
        While (Mid$(lastline, x, 1) = " ")
            x = x + 1
        Wend
        'remove following No., find space at end
        y = InStr(x, lastline, " ", vbTextCompare)
        'parse similarity line (on next line)
dummy = GetALine
txt = txt & lastline & "|
similine = Mid$(lastline, x, y - x)
Call Convert(similine)
' store similine
Calcs(i).Binary = Calcs(i).Binary & similine
Else
  txt = txt & lastline & "|
End If
dummy = GetALine
c1 = InStr(1, lastline, ">")
c2 = InStr(1, lastline, "Score = ")
c3 = InStr(1, lastline, "Database:")
ctot = c1 + c2 + c3
Loop
BLSTRes(i).Align = txt
End Sub-

' changes chars to 1, 0 or +
Private Sub Convert(line As String)
Dim i As Integer
Dim LenLine As Integer
' get the length of the sequence
LenLine = Len(line)
For i = 1 To LenLine
  If (Mid$(line, i, 1) = " ") Then
    Mid(line, i, 1) = "0"
  Else
    Mid(line, i, 1) = "1"
  End If
Next i
End Sub

' returns false at end of file else puts line in current string var
Private Function GetALine() As Boolean
Dim complace As Integer
If instream.AtEndOfStream Then ' instream remembers where it has got to
  lastline = ""
  GetALine = False
Else ' replace "'", and "|", pos IDed by complace
  SQL doesnt like apostrophes
  lastline = instream.ReadLine
  complace = InStr(1, lastline, "'")
  Do While complace <> 0
    Mid(lastline, complace, 1) = "\n"
    complace = InStr(1, lastline, "'")
  Loop
  ' SQL doesn't like uprights
  complace = InStr(1, lastline, "|")
  Do While complace <> 0
    Mid(lastline, complace, 1) = ":"
    complace = InStr(1, lastline, "|")
  Loop
  GetALine = True
End If
End Function

Private Sub DBEntry(i As Integer)
Dim sqlstring As String
Set RS = DBs.OpenRecordset("Results", dbOpenDynaset)
'check for blank entries
If BLSTRes(i).Accl = "" Then
    BLSTRes(i).Accl = "not present"
End If
If BLSTRes(i).Acc2 = "" Then
    BLSTRes(i).Acc2 = "not present"
End If
If BLSTRes(i).Desc = "" Then
    BLSTRes(i).Desc = "not present"
End If
'INSERT INTO Results
VALUES ('TestlD', #BDate#, ResNo, 'Accl', 'Acc2', 'Desc',
Length,
'Score, Evaluate, 'Idents', 'Posit', 'Gaps', 'Align')
sqlstring = "INSERT INTO Results VALUES ('" & BLSTRes(i).TestlD & 
' ', '#', " & BLSTRes(i).BDate & ", " & BLSTRes(i).ResNo & ", " & 
BLSTRes(i).Length & ", " & BLSTRes(i).Score & ", " & 
DBs.Execute(sqlstring)
RS.Close
Set RS = Nothing
Set RS = DBs.OpenRecordset("Calculations", dbOpenDynaset)
'INSERT INTO Calculations
VALUES ('TestlD', #BDate#, ResNo, 'Scores', 'Sps', 'gen',
'prt', 'Binary', Clump)
sqlstring = "INSERT INTO Calculations VALUES ('" & 
Calcs(i).TestlD & ", " & 
Calcs(i).BDate & ", " & Calcs(i).ResNo & ", " & 
Calcs(i).Prt & ", " & Calcs(i).key & ", " & Calcs(i).Binary & ")"
DBs.Execute(sqlstring)
RS.Close
Set RS = Nothing
End Sub

E.2.4. Evaluate scores - modProtScores

'Evaluate score against thresholds
'If Ok no change to rank position
'Else may move dependent on other factors
Option Explicit

Public Sub EvalScore(i As Integer)
Dim S As Double
Dim ST As String
Dim E As Double
Dim ET As String
Dim A As String
S = BLSTRes(i).Score
ST = STThresh
E = BLSTRes(i).Evaluate

272
ET = EThresh
'using Resultsinfo.score and .evalue and threshold from form
If ST = "" And ET = "" Then 'both blank
A = "Use"
ElseIf ST <> "" And ET = "" Then 'only score
    If S >= Val(ST) Then
        A = "Use"
    Else
        A = "Ignore"
    End If
ElseIf ST = "" And ET <> "" Then 'only evalue
    If E <= Val(ET) Then
        A = "Use"
    Else
        A = "Ignore"
    End If
Else 'both
    If S >= Val(ST) And E <= Val(ET) Then
        A = "Use"
    ElseIf S < Val(ST) And E <= Val(ET) Then
        If Pref = "Score" Then
            A = "E"
        Else
            A = "Use"
        End If
    ElseIf S >= Val(ST) And E > Val(ET) Then
        If Pref = "Score" Then
            A = "Use"
        Else
            A = "S"
        End If
    Else
        A = "ignore"
    End If
End If
End Sub

E.2.5. Calculate clumpiness value – modProtClump

'clumpiness measure
Option Explicit

Public Sub Clumps(i As Integer)
Dim bline As String
Dim percent As Integer
Dim coverage As Single
Dim clumpvalue As Single
Dim stndrdclump As Single
Dim spc1 As Integer
Dim spc2 As Integer
bline = Calcs(i).Binary
'check if borderline enough to do clumpiness measure
spc1 = InStr(1, BLSTRes(i).Posit, "(")
spc2 = InStr(spc1, BLSTRes(i).Posit, ")")
percent = Val(Mid(BLSTRes(i).Posit, spc1 + 1, spc2 - 2))
If percent >= 60 Then
    stndrdclump = 10000
Else
    clumpvalue = GetClumpValue(bline)
    'get coverage of query by hit for normalising
coverage = Len(bline) / letters
'normalise by coverage (length of match)

stndrdclump = clumpvalue * coverage

End If
Calcs(i).Clump = Round(stndrdclump, 3)

End Sub

'Calculate the clumpiness value
Private Function GetClumpValue(line As String) As Single
'using CUSUM with h = 4, k = 0.6 for a single sequence
Dim h As Single
h = 4
Dim k As Single
k = 0.6
Dim l As Integer
l = Len(line)
Dim m As Single
m = Mean(line)
Dim d As Single
d = Stdev(line)
Dim Hv As Single
Hv = h * d
Dim Kv As Single
Kv = k * d
Dim i As Integer
For i = 1 To l
    x = Val(Mid(line, i, 1))
    If d = 0 Then
        y = x
    Else
        y = (x - m) / d
    End If
    C(i) = 0
    If i = 1 Then
        C(i) = y - Kv + (Hv / 2)
    Else
        C(i) = y - Kv + C(i - 1)
    End If
    If C(i) < 0 Then
        C(i) = 0
    End If
    If C(i) > Hv Then
        NumOOC = NumOOC + 1
    End If
Next i
GetClumpValue = NumOOC
End Function

'calc mean and stdev
Private Sub Stats(l As String, m As Single, d As Single)
Dim C As Integer
C = 0
For i = 1 To Len(l)
    If Mid(l, i, 1) = "1" Then
        C = C + 1
    End If
Next i
End If
Next i
m = C / Len(l) 'num hits/len line
d = Sqr((1 / (Len(l) - 1)) * (C - (C ^ 2 / Len(l))))
End Sub
APPENDIX F – USER MANUAL FOR PROTOTYPE

The JB-BLAST-Analysis Prototype

CONTENTS

Contents 276

1. Installation 276
   To install the program:

2. Getting Started 277

3. Data Input 277
   User Information 278
   Test Sequence 278

4. Submission 279

5. The Clumpiness Report 280

1. INSTALLATION

The Program is made up of an exe file and an associated database. It also creates files
during runtime, which are stored in two folders, also created in runtime if not already
present. The files in the folders ‘Results’ and ‘Reports’ are BLAST results files and
reports generated by the program and are accessible outside the program provided
they are not moved while the program is still running.

WARNING! Do not separate the program file from the database; they must remain
in the same folder otherwise the program will not run.

If the program is separated from it’s associated folders ‘Reports’ and ‘Results’
it will create new ones so it is advisable to leave everything in the initial folder to
avoid creating multiple instances of folders throughout the hard drive.

To install the program:

1. Insert disc into floppy drive
2. Double click on InstallJB file
3. Choose where to install program, default is on the C:\ drive in a folder named
   ‘JBClumpAnalysis’
4. As this is a Visual Basic application it may also require the VB runtime files.
   The program ‘vbrun60sp5.exe’ will automatically install these files for you
   into the system32 folder when it is double clicked.
5. However, three other runtime files are not included in the ‘vbrun60sp5.exe’
   package. These can be found in ‘MSINET.exe’ and will install the files into
   the system32 folder, provided that is located on the C:\ drive.

NOTE: Please ensure that these files are installed in the correct location otherwise the
program will not run.
6. Installation is now complete. Open the ‘JBClumpAnalysis’ folder and double
   click the ‘JBClumpAnalysis.exe’ file to run the program.
2. GETTING STARTED

Once the program is successfully installed, it can then be used. In the following sections the various screens presented to the user will be described. To summarise:

- The initial screen asks for user and query sequence details
- The second screen is the BLAST submission form and its usage is very similar to manual submissions
- The third and final screen shows a report on the analysis performed on the BLAST results. The layout of this Report is very similar to that of the BLAST report and, therefore, hopefully makes it easier to use

Being a prototype this program has very limited functionality. The program will currently only process amino acid sequences using BLASTp on the NCBI web site. The initial submission of the sequence to the BLAST program is not fully automated and requires some user intervention. Also it is not possible, at this time, for a copy to be made of the hits distribution graphic at the top of the BLAST results page, therefore, the results page will remain open for any other perusal the user may wish to conduct.

WARNING! Dependent on the PC specifications and size query sequence and of the BLAST result file to analyse, the program may take several minutes to run once the analysis has been started. It is recommended to use sequences of with lengths of 500 or less where possible.

3. DATA INPUT

When the program is started the first screen that is presented is the Data Input screen. On this screen there are two areas, labelled ‘User Information’ and ‘Test Sequence’, figure 3.1. Both areas have their own clear button to empty all the fields in that particular area and there is also another reset button at the bottom of the page that will clear all the fields on the screen. Finally there is an Exit button to close the program and a Run button to start the analysis

![Figure 3.1. Screen shot of initial Input page](image)
**User Information**

The user information would provide access to a particular user's sequence submissions in the full program. These could also have options to be password protected so as to maintain each user's scientific property. The e-mail address would be used as an alternative means of returning the results to the user, other than through the forms, though this is not currently implemented.

**NOTE:** In the prototype the user information is only used as a form of identification for each submission, with the e-mail address not being used at all and, therefore, may be left blank.

In the first instance of a name being entered a unique ID is generated using the initials and the current date, which is stored on the database. These can then be accessed using the drop-down menus for both Forename and Surname, figure 3.2a. If a name is used that is already present in the database then a query message box is displayed on running the analysis, figure 3.2b. If this is the first instance for a particular user they would need to click ‘No’ and enter a different fore or surname, otherwise just click ‘Yes’ to continue.

**NOTE:** The program is not currently configured for multiple users with the same name.

**Test Sequence**

Here an identifying name for the query sequence, the query sequence itself and its species, if known, are entered, figure 3.3. There is also the opportunity to specify whether certain scores and/or e-values are unacceptable with preference to which is least acceptable, though this is not necessary. Again, there is a drop-down list to enable access to previously submitted sequences to conduct reBLASTs for any new data. At the bottom there is a field for the number of letters in the query sequence but this need not be filled in, as the program will do that.
Once all the necessary areas have been filled pressing the ‘Run’ button will start the analysis

4. SUBMISSION

As mentioned before, the submission of a query sequence to the BLAST system on the NCBI site is not fully automated. The next screen to be presented is the BLAST submission page from the NCBI site and there are 5 steps required to produce the BLAST results page:

1. Select the query sequence that has been copied into the text box at the top of the screen, label 1 in figure 4.1. This can also be done by double clicking the text box.
2. Copy and paste the query sequence into the textarea, label 2 in figure 4.1, in the BLAST submission form as per a normal manual sequence submission
3. Click the ‘BLAST!’ button, label 3 in figure 4.1, and when then next page loads, the ‘Format’ button, again as per a normal manual sequence submission. To refresh the page either right click in the window and select refresh or press F5 then Enter
4. As mentioned earlier the hit distribution graphic is only accessible in the BLAST results, figure 4.2, so the user may wish to take a moment to look at this. The BLAST results window is not closed, except by the user, so it is possible to return to it later
5. Once the BLAST results have loaded the capture button, label 4 in figure 4.1, must be pressed to enable the program to conduct it analysis
6. During the analysis a scroll bar displays the progress, label 5 in figure 4.1. The value, or size of bar, varies depending on the number of hits to process, so a file with few hits will have a long bar and one with many hits will have a short bar.

At this point the automatic analysis starts and a second window will open after a short period showing a report on the analysis done.

WARNING! On large sequences or large BLAST results files they can take several minutes, for example, a sequence of 2378 letters giving 162 results took up to 6 minutes to run whereas a sequence of 162 letters giving 6 results took about a minute.
With a faster processor these times will be reduced.

**Figure 4.1.** Screen shot of the BLAST submission screen.

**Distribution of 6 Blast Hits on the Query Sequence**

*Mouse-over to show defline and scores. Click to show alignments*

**Figure 4.2.** Hit distribution graphic from a BLAST results page

**5. THE CLUMPINESS REPORT**

The layout of this report is very similar to that of the BLAST results page, therefore knowing where something is on one enable the user to find it on the other. There are, however, certain differences. At the very top of the page the query sequence ID and date and time the BLAST search was submitted are incorporated into the title. Below this is a link to the NCBI site followed by details of the BLAST program and version, the BLAST submission ID, the database searched and the Query sequence length. Finally there is a link to the Taxonomy report produced by the BLAST program and the number of hits found figure 5.1.
Figure 5.1. Top of Analysis report

After this comes the list of hits set out very similarly as the BLAST results file but with two extra pieces of information, figure 5.2. The first is between the accession number and description to warn if the hit is hypothetical or putative, or if the words ‘family’ or ‘domain’ are in the description. The other can be found at the end and shows the clumpiness value for each of the hits. Those sequences with a positives percentage of 60% or higher are regarded as over clumpy and do not require a clumpiness measurement and are automatically given a maximal value.

NOTE: This percentage value is arbitrary in this prototype but could very easily be user specified. Also, if score and e-value thresholds were entered on the initial input page and hits with scores or e-values below those thresholds are highlighted in red.

Additional hits are generated where there are multiple alignments from the same hit sequence to the query sequence. In the original BLAST results file only the most similar of these would appear in the hit list with the additional alignments appearing under this ‘best’ match in descending order of similarity.

Figure 5.2. Hit list

Finally at the bottom are the alignments and as in the BLAST results clicking on a particular hits score can link to these. In addition to the BLAST details of Accession numbers, description, score, e-value, identities, positives and gaps percentages, there is also the clumpiness value and any keywords found in the description. Clicking on a
keyword will link to the NCBI taxonomy site for species keywords or the gene or protein databases in the entrez search engine.

**NOTE:** entrez will return results on most words including ‘component’ or small words such as ‘of’, ‘in’ or ‘the’, therefore these small words are removed before any searches are done. However, only certain larger words are removed, specifically those associate with hypothetical or unknown proteins and the words ‘Domain’ and ‘Family’.

**WARNING!** due to difficulties with the NCBI web interface, keywords will not always be found as when an error occurs the program will skip the search producing a blank or partial keyword result.

**Figure 5.3. Alignments**

**NOTE:** There is currently no evidence to suggest that clumpiness values produced by one search are comparable with those in another. Also, there is no indication of a good clumpiness value as there is in the score, i.e. at present it is not possible to define a threshold value.
APPENDIX G – INSTRUCTIONS FOR ‘NON-EXPERT’ CLUMPINESS MEASURE ASSESSMENT

Instructions – Part I

Thanks for participating in this experiment!

Your task is to order the given strips so that the most clustered is at the top and the least clustered at the bottom.

• Each strip consists of a number of green and white blocks, where a green block represents an event and a white block the absence of an event. For example, in an avenue the area in front of a house can either contain a tree or not contain a tree:

In the strip below the avenue, each block represents the area in front of one house, coloured green if there is a tree in front of the house, white if there is no tree.

• On each strip, or avenue, the distribution of green blocks (trees) varies from being mostly clustered together to being regularly spread along the whole strip. In the avenue, that would be most of the trees being in front of neighbouring houses to spread along the whole avenue.

• Procedure:
  a. Choose one strip, place it on the table
  b. Select a second strip, compare this to the first strip
     i. Place it above the first strip if you consider it to be more clustered
     ii. Place it below the first strip if you consider it to be less clustered
  c. Select a third strip, compare to each of the previous two
     i. Place it above both strips if you consider it to be more clustered
     ii. Place it below both strips if you consider it to be less clustered
     iii. Place it between the two strips if you consider it to be more clustered than the bottom one but less clustered than the top one
  d. Repeat until all strips have been placed
  e. Recheck that the strips have been placed as you think best shows the progression from most to least clustered

• Once satisfied with the order, turn to the second sheet and answer the questions there
Questions – Part I

1. Were there any strips that you found difficult to place? Write the number(s) below (found of the back/top right corner of the strip) and explain why:

<table>
<thead>
<tr>
<th>Number of strip</th>
<th>Difficult because</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Use the space below to describe the method or strategy you used, for deciding if a particular strip was more clustered or less clustered than another.
3. Did you use the same method for all strips?
   Yes ☐  No ☐

4. If no please describe what you did below:
Instructions – part II

Task: to review and critically appraise the order generated automatically

- Here you are given 50 strips automatically ordered by the computer

- Look down the list and decide whether you agree with the order or whether you would move some/all of the strips to different positions

- Once satisfied, turn to the next sheet and answer the questions there
Questions – Part II

1. Were there any strips that you would consider significantly out of place? Enter the number(s) below and explain why you think so:

<table>
<thead>
<tr>
<th>Number of strip</th>
<th>Why consider out of place</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Whilst reviewing the list did you use the same or similar strategy to that used in part I
   Yes ☐          No ☐

3. If no, what strategy did you use this time?