Analysis and Comparison of Sequence Analysis Tools

A thesis
Submitted in partial fulfillment of the requirements for
the award of degree
of

Master of Engineering
In
Software Engineering

Under the Supervision of
Mrs. Rinkle Aggarwal
Lecturer,
Computer Science & Engineering Department
Thapar Institute of Engineering & Technology, Patiala.

Submitted By
Navjot Kaur
(8033111)

Computer Science & Engineering Department
Thapar Institute of Engineering & Technology
(Deemed University), Patiala-147004 (India)

May 2005

DECLARATION

I hereby certify that the work which is being presented in the thesis entitled, "***Analysis and Comparison of Sequence Analysis Tools***", in partial fulfilment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted at Computer Science and Engineering Department of Thapar Institute of Engineering and Technology (Deemed University), Patiala, is an authentic record of my own work carried out under the supervision of Mrs. Rinkle Aggarwal. I have not submitted the matter presented in this thesis for the award of any other degree or to any other University.

Navjot Kaur

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mrs. Rinkle Aggarwal)

Lecturer
Computer Science. & Engg. Department,
Thapar Institute of Engg. & Technology,
Patiala.

Countersigned by

(Mr. R. S. Salaria)

Head,
Computer Sc. & Engg. Department,
Thapar Institute of Engg. & Technology,
Patiala.

(Dr. D. S. Bawa)

Dean (Academic Affairs),
Thapar Institute of Engg. & Technology,
Patiala.

ACKNOWLEDGEMENT

A journey is easier when you travel together. Interdependence is certainly more valuable than independence. This thesis is the result of work carried out during the final year of my course whereby I have been accompanied and supported by many people. It is a pleasant aspect that I have now the opportunity to express my gratitude to all of them.

No amount of words can adequately express the debt, I owe to Mrs. Rinkle Aggarwal, Lecturer, Computer Science & Engineering Department, for her uncanny guidance, support and motivation that triggered me for the thesis work. I owe her lots of gratitude for having me shown this way of research. I learnt a lot from her.

I am highly indebted to Mr. R. S. Salaria, Head, Computer Science & Engineering Department, for providing me the requisite environment and for being the constant source of Inspiration.

I am also thankful to the entire faculty, staff members and my colleagues who were always there at the hour of the need and provided with all the help and support, which I needed, for the completion of my thesis.

At last but not the least I would like to thank God for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Navjot Kaur
Roll no. 8033111

ABSTRACT

Necessity is the Mother of Invention. Every day the world creates large amount of data of data but only part of this whole percentage is used for any purpose. Data in biology are very diverse and abundant. With the significant growth of the amount of biomolecular data, it becomes increasingly important to develop new techniques for extracting knowledge from the data. Data mining is a fundamental operation in such a domain. As Coal-mines are mined to find the diamond in the similar way data mining is applied to raw data to get the valuable information.

The gene sequences of related species of plants, animals and microorganisms show complex patterns of similarity to one another and many molecular biologists are convinced that an understanding of sequence evolution is the first step towards understanding the evolution of life itself. There is variety of different tools available to perform sequence analysis. Various software packages of automated tools have been developed that had improved the efficiency of much biological research.

I studied a few tools of sequence analysis and in the end selected the BLAST, FASTA, BLAT and CLUSTERW for comparison; All known tools have some advantages over the other. All have different functions, different areas of applications. I am analyzing these tools based on two underlined criteria. One is based on algorithm and the second one is parameter based. All four tools have some underlined algorithm. So the algorithms are compared to know that which algorithm is more efficient than the other. What steps of the particular algorithm are different and what are same. Different tools work according to the different parameters. These parameters add to the performance of the algorithm. Various parameters are compared to know that which parameter is used in which tool and what is its function. What same parameters are used in the different algorithms and what is the particular use of that parameter in that particular tool.

TABLE OF CONTENTS

LIST OF FIGURES

Number ***Page***

LIST OF TABLES

Number ***Page***

ORGANIZATION OF THESIS

The Thesis entitled ***"Analysis and Comparison of Sequence Analysis Tools"*** is mainly concerned with comparison of various available tools. All tools are compared according to some defined criteria

The first chapter briefly introduces introduction to the data mining. All necessary concepts of data mining that are used in bioinformatics are discussed.

In the second chapter basic introduction to field of bioinformatics is given. Combining the best features of both data mining and biology are illustrated using applications and issues of bioinformatics.

Third chapter is mainly devoted to biomolecules sequences. Chapter starts with the basic introduction to the biomolecules and extended with the description why sequences differ. Underlying biology concepts are discussed. Chapter ends with the types of sequence alignment algorithms.

Fourth chapter is about various sequence analysis tool. Steps for running the various programs are discussed. Then the results are interpreted using examples, features of all tools are explained along with the advantages and limitations of all tools.

Fifth chapter is devoted to research analysis of various tools. I have selected four tools for comparison and they are compared to each other according to two criteria: algorithm based and parameter based. Results are represented in this chapter.

Finally the thesis is concluded is sixth chapter describing results. Results are concluded in the form of tables one is based on algorithm and the other is on the basis of parameters.

CHAPTER 1

INTRODUCTION TO DATA MINING

1.1 OVERVIEW

We are in an age often referred to as the information age. In this information age, because we believe that information leads to power and success, and thanks to sophisticated technologies such as computers, we have been collecting tremendous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, counting on the power of computers to help sort through this amalgam of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming. So

“We are drowning in data, but starving for knowledge!”

This initial chaos has led to the creation of structured databases and database management systems (DBMS). The efficient database management systems have been very important assets for management of a large corpus of data and especially for effective and efficient retrieval of particular information from a large collection whenever needed. With time many new advances have been done in the field of databases. New efficient techniques have been developed to solve the problems of existing systems(1). Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the “essence” of information stored, and the discovery of patterns in raw data. As

“**Necessity is the Mother of Invention.”**

So field of data mining is born. With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making (1).

1.2 WHAT IS DATA MINING

Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. Data Mining can be viewed as an analytical process designed to explore data in search for consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data (3). There are many terms carrying a similar or slightly different meaning to data mining, such as knowledge mining from databases, knowledge extraction, data/ pattern analysis, data archaeology, and data dredging. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. The following figure (Figure 1.1) shows data mining as a step in an iterative knowledge discovery process.

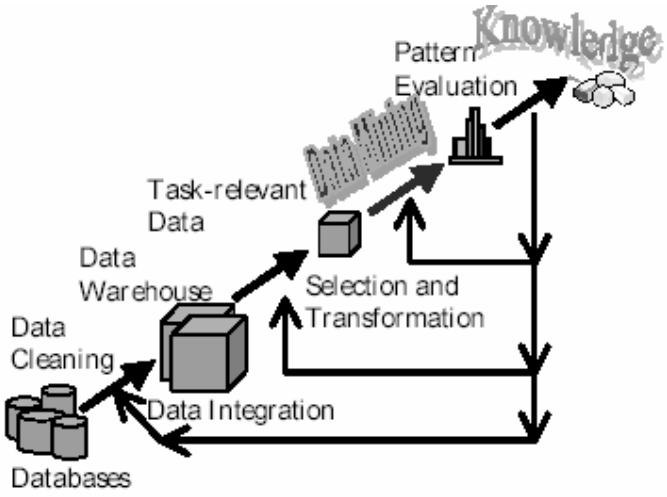


Figure1.1 Data Mining As a Step In KDD

The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

- **Data cleaning:** Also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration:** At this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- **Data selection:** At this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation:** Also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining:** It is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- **Pattern evaluation:** In this step, strictly interesting patterns representing knowledge are identified based on given measures.
- **Knowledge representation:** Is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining rocks for a vein of valuable ore. Both imply either sifting through a large amount of material or ingeniously probing the material to exactly pinpoint where the values reside. It is, however, a misnomer, since mining for gold in rocks is usually called “gold mining” and not “rock mining”, thus by analogy, data mining should have been called “knowledge mining” instead(2).

1.3 THE FOUNDATIONS OF DATA MINING

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time(4). Data mining is ready for application in the

business community because it is supported by three technologies that are now sufficiently mature:

- Massive data collection
- Powerful multiprocessor computers
- Data mining algorithms

In the evolution from business data to business information, each new step has built upon the previous one. This has been explained with the help of example of a business unit. The steps are:

Evolutionary Step	**Business Question**	**Enabling Technologies**	**Characteristics**
Data Collection (1960s)	"What was my total revenue in the last five years?"	Computers, tapes, disks	Retrospective, static data delivery
Evolutionary Step Data Access (1980s)	**Business Question** What were unit sales in New England last March?"	**Enabling Technologies** Relational databases (RDBMS), Structured Query Language (SQL), ODBC	**Characteristics** Retrospective, dynamic data delivery at record level
Data Warehousing & Decision Support (1990s)	"What were unit sales in New England last March? Drill down to Boston."	On-line analytic processing (OLAP), multidimensional databases, data warehouses	Retrospective, dynamic data delivery at multiple levels
Data Mining (Emerging Today)	"What's likely to happen to Boston unit sales next month? Why?"	Advanced algorithms, multiprocessor computers, massive databases	Prospective, proactive information delivery

Table1.1 Foundation of Data Mining Using Layers

1.4 SCOPE OF DATA MINING

Data mining is a tool, not a magic wand. It won't sit in your database watching what happens and send you e-mail to get your attention when it sees an interesting pattern. It doesn't eliminate the need to know your business, to understand your data, or to understand analytical methods. Data mining assists business analysts with finding patterns and relationships in the data — it does not tell you the value of the patterns to the organization(4). Furthermore, the patterns uncovered by data mining must be verified in the real world. Data mining does not replace skilled business analysts or managers, but rather gives them a powerful new tool to improve the job they are doing.

1.4.1 Data Mining Finds Patterns And Relationships In Data

By using sophisticated techniques association, sequence or path analysis, classification, clustering and forecasting to build models — abstract representations of reality. A good model is a useful guide to understanding your business and making decisions. There are two main kinds of models in data mining: ***predictive*** and ***descriptive*.** Predictive models can be used to forecast explicit values, based on patterns determined from known results. Descriptive models describe patterns in existing data, and are generally used to create meaningful subgroups such as demographic clusters.

1.4.2 Automated Prediction Of Trends And Behaviors

Data mining automates the process of finding predictive information in large databases. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings.

1.4.3 Automated Discovery Of Previously Unknown Patterns

Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of

retail sales data to identify seemingly unrelated products that are often purchased together.

1.4.4 Faster Processing In Large Databases

Data mining techniques can yield the benefits of automation on existing software and hardware platforms, and can be implemented on new systems, as existing platforms are upgraded and new products developed. When data mining tools are implemented on high performance parallel processing systems, they can analyze massive databases in minutes. Faster processing means that users can automatically experiment with more models to understand complex data. High speed makes it practical for users to analyze huge quantities of data. Larger databases, in turn, yield improved predictions.

1.4.5 Simplify The Graphic Representation Of The Data

Addition to algorithms, data mining software usually has features to simplify the graphic representation of the data (visualization tools) plus interfaces to common database formats. Data mining can be performed on data represented in quantitative, textual, or multimedia forms.

1.5 ARCHITECTURE FOR DATA MINING

Many data mining tools currently operate outside of the warehouse, requiring extra steps for extracting, importing, and analyzing the data. Frequently, the data to be mined is first extracted from an enterprise data warehouse into a data mining database or data mart. There is some real benefit if your data is already part of a data warehouse. As, the problems of cleansing data for a data warehouse and for data mining are very similar. If the data has already been cleansed for a data warehouse, then it most likely will not need further cleaning in order to be mined(2).

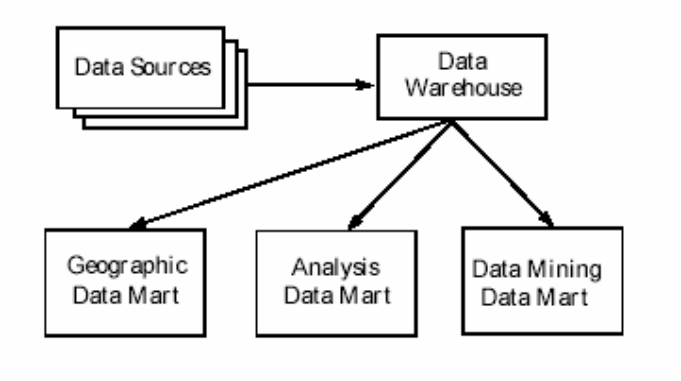


Figure 1.2 Data Mart Extracted From Data Warehouse

Furthermore, you will have already addressed many of the problems of data consolidation and put in place maintenance procedures. The data-mining database may be a logical rather than a physical subset of your data warehouse, provided that the data warehouse DBMS can support the additional resource demands of data mining. If it cannot, then you will be better off with a separate data-mining database. A data warehouse is not a requirement for data mining. Setting up a large data warehouse that consolidates data from multiple sources, resolves data integrity problems, and loads the data into a query database can be an enormous task, sometimes taking years and costing millions of dollars. You could, however, mine data from one or more operational or transactional databases by simply extracting it into a read-only database. This new database functions as a type of data mart.

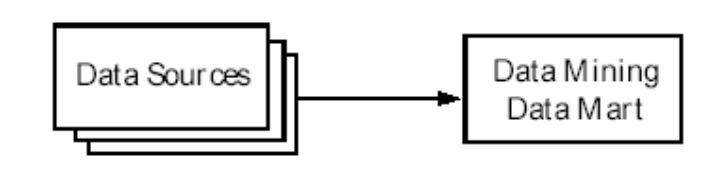


Figure1.3 Data Mart Extracted From Operational Databases

So components of data mining architecture are illustrated using (Figure1.4)

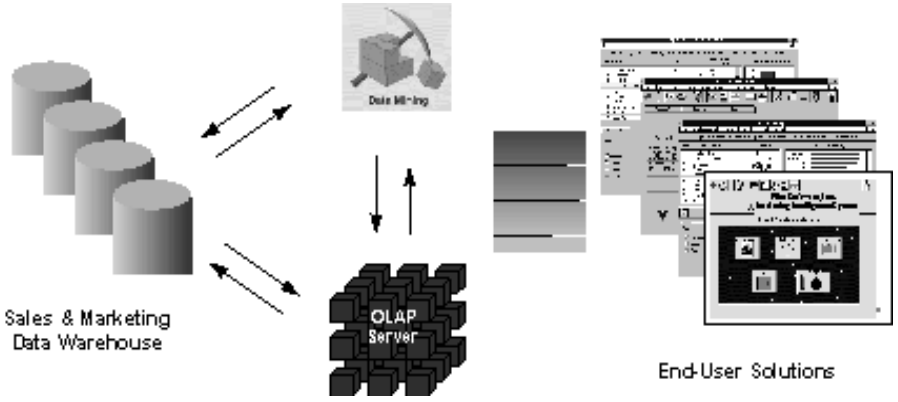


Figure 1.4 Integrated Data Mining Architecture

1.5.1 Data Warehouse: The ideal starting point is a data warehouse containing a combination of internal data tracking all customer contact coupled with external market data about competitor activity. Background information on potential customers also provides an excellent basis for prospecting.

1.5.2 An OLAP (On-Line Analytical Processing) Server: Enables a more sophisticated end-user business model to be applied when navigating the data warehouse. The multidimensional structures allow the user to analyze the data, as they want to view their business – summarizing by product line, region, and other key perspectives of their business.

1.5.3 The Data Mining Server: Must be integrated with the data warehouse and the OLAP server to embed ROI-focused business analysis directly into this infrastructure. An advanced, process-centric metadata template defines the data mining objectives for specific business issues like campaign management, prospecting, and promotion optimization. Integration with the data warehouse enables operational decisions to be directly implemented and tracked.

1.5.4 End User Solutions Representation: This design represents a fundamental shift from conventional decision support systems. Rather than simply delivering data to the end user through query and reporting software, the Advanced Analysis Server applies users' business models

directly to the warehouse and returns a proactive analysis of the most relevant information..

Broadly data mining architecture can be divided in three main categories

- **One-Tier:** In architecture there are mainly three parts. Graphical user interface, Database and data access part. If all these three parts are at client side then it is one tier architecture.

- **Two-Tier:** In this architecture graphical user interface and some part of database is at client side. Whereas at server side there is remaining part of database and data access.
- **Three-Tier:** in this architecture at client side there is graphical user interface and part of database whereas at server side lies the remaining part of database. in between the client and server there is one layer and on this data access services are situated.

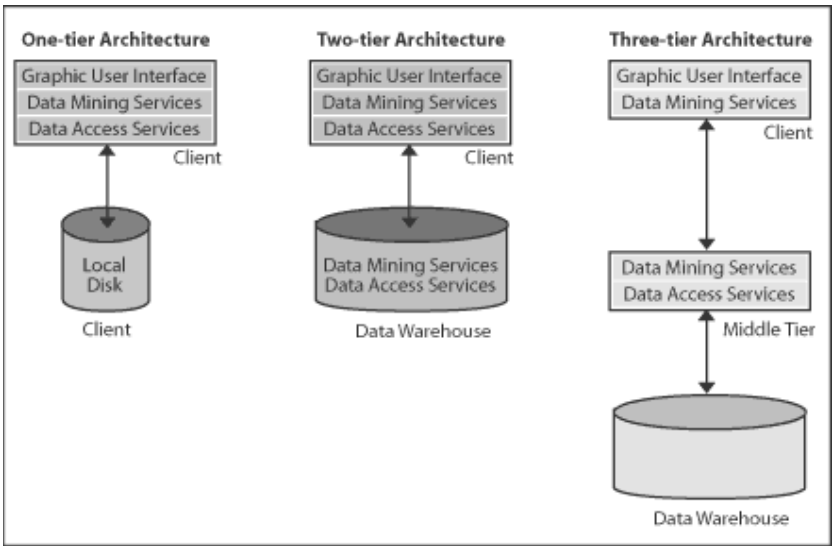


Figure 1.5 Architecture of Data Mining

1.6 DATA MINING FUNCTIONALITIES

The kinds of patterns that can be discovered depend upon the data mining tasks employed. By and large, there are two types of data mining tasks: descriptive data mining tasks that describe the general properties of the

existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data(1). The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list:

1.6.1Characterization: Data characterization is a summarization of general features of

Object in a target class, and produces what is called characteristic rules. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions.

1.6.2 Discrimination: Data discrimination produces what are called discriminate rules and is basically the comparison of the general features of objects between two classes referred to as the target class and the contrasting class. The techniques used for data discrimination are very similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.

1.6.3 Association Analysis: Association analysis is the discovery of what are commonly called association rules. It studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets. Another threshold, confidence, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules. Association analysis is commonly used for market basket analysis. The discovered association rules are of the form: P®Q [s,c], where P and Q are conjunctions of attribute value-pairs, and s (for support) is the probability that P and Q appear together in a transaction and c (for confidence) is the conditional probability that Q appears in a transaction when P is present. For example, the hypothetic association rules:

RentType (X, “game”) Ù Age (X, “13-19”) ® Buys (X, “pop”) [s=2%, c=55%]

Would indicate that 2% of the transactions considered are of customers aged between 13 and 19 who are renting a game and buying a pop, and that there is a certainty of 55% that teenage customers who rent a game also buy pop.

1.6.4 Classification: Classification analysis is the organization of data in given classes.

Also known as supervised classification, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. The classification analysis would generate a model that could be used to either accept or reject credit requests in the future.

1.6.5 Prediction: Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values.

1.7 DATA MINING TECHNIQUES

The most commonly used techniques in data mining are:

1.7.1 Artificial Neural Networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure. Neural networks are of particular interest because they offer a means of efficiently modeling large and complex problems in which there may be hundreds of predictor variables that have many interactions. A neural network starts with an input layer, where each node corresponds to a predictor variable. These input nodes are connected to a number of nodes in a hidden layer. Each input node is connected to every node in the hidden layer. The nodes in the hidden

layer may be connected to nodes in another hidden layer, or to an output layer. Advantage of neural network models is that they can easily be implemented to run on massively parallel computers with each node simultaneously doing its own calculations.

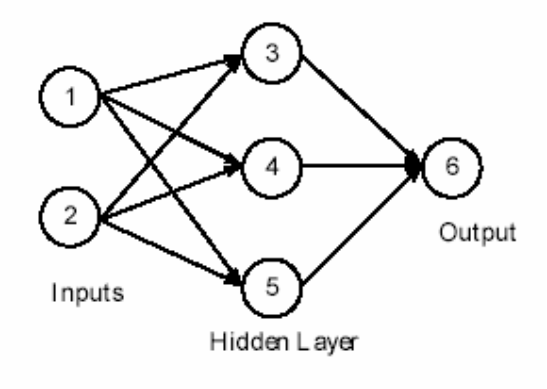


Figure 1.6 Neural Networks with One Hidden Layer

1.7.2 Decision Trees: Tree-shaped structures that represent sets of decisions. Decision trees are a way of representing a series of rules that lead to a class or value. For example, you may wish to classify loan applicants as good or bad credit risks.

Figure 1.7 A Simple Classification Tree

These decisions generate rules for the classification of a dataset. Decision trees make few passes through the data (no more than one pass for each level of the tree) and they work well with many predictor variables. Decision trees handle non-numeric data very well.

1.7.3 Genetic Algorithms: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design

based on the concepts of evolution. Genetic algorithms are not used to find patterns. Essentially, genetic algorithms act as a method for performing a guided search for good models in the solution space. They are called genetic algorithms because they loosely follow the pattern of biological evolution in which the members of one generation (of models) compete to pass on their characteristics to the next generation (of models), until the best (model) is found.

1.7.4 Linear Regression: A statistical technique used to find the best-fitting liner relationship between a numeric target variable and its set of predictor variables. Linear regression can be used to predict the amount of overdraft protection to offer a customer based on their account balances, years of service and other characteristics.

1.7.5 Nearest Neighbor Method: When trying to solve new problems, people often look at solutions to similar problems that they have previously solved. K-nearest neighbor (k-NN) is a classification technique that uses a version of this same method. It decides in which class to place a new case by examining some number — the "k" in k-nearest neighbor — of the most similar cases or neighbors. It counts the number of cases for each class, and assigns the new case to the same class to which most of its neighbors belong. For example N is new case. It would be assigned to class X because the seven X's within the ellipse outnumber the two Y's.

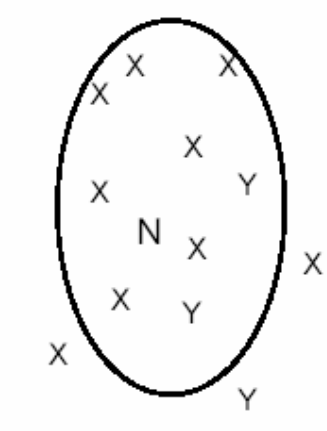


Figure 1.8 Nearest Neighbour Method

1.8 INTERESTINGNESS AND USEFULLNESS OF DATA

Data mining allows the discovery of knowledge potentially useful and unknown. Whether the knowledge discovered is new, useful or interesting, is very subjective and depends upon the application and the user. It is certain that data mining can generate, or discover, a very large number of patterns or rules. To reduce the number of patterns or rules discovered that have a high probability to be non-interesting, one has to put a measurement on the patterns. However, this raises the problem of completeness. The user would want to discover all rules or patterns, but only those that are interesting. The measurement of how interesting a discovery is, often called interestingness, can be based on quantifiable objective elements such as validity of the patterns when tested on new data with some degree of certainty, or on some subjective depictions such as understandability of the patterns, novelty of the patterns or usefulness.

Discovered patterns can also be found interesting if they confirm or validate a hypothesis sought to be confirmed or unexpectedly contradict a common belief. Typically, measurements for interestingness are based on thresholds set by the user. While some concrete measurements exist, assessing the interestingness of discovered knowledge is still an important research issue.

1.9 ISSUES IN DATA MINING

Data mining algorithms embody techniques that have sometimes existed for many years, but have only lately been applied as reliable and scalable tools that time and again outperform older classical statistical methods. While data mining is still in its infancy, it is becoming a trend and ubiquitous. Before data mining develops into a conventional, mature and trusted discipline, many still pending issues have to be addressed (6). Some of these issues are addressed below.

1.9.1 Security And Social Issues: Security is an important issue with any data collection that is shared and is intended to be used for strategic decision-making. In addition, when data is collected for customer

profiling, user behavior understanding, correlating personal data with other information, etc., large amounts of sensitive and private information about individuals or companies is gathered and stored. Moreover, data mining could disclose new implicit knowledge about individuals or groups that could be against privacy policies, especially if there is potential dissemination of discovered information.

1.9.2 User Interface Issues: The knowledge discovered by data mining tools is useful as long as it is interesting, and above all understandable by the user. Good data visualization eases the interpretation of data mining results, as well as helps users better understand their needs. There are many visualization ideas and proposals for effective data graphical presentation. However, there is still much research to accomplish in order to obtain good visualization tools(6).

1.9.3 Mining Methodology Issues: These issues pertain to the data mining approaches applied and their limitations. Topics such as versatility of the mining approaches, the diversity of data available, the dimensionality of the domain, the broad analysis needs (when known), the assessment of the knowledge discovered, the exploitation of background knowledge and metadata, the control and handling of noise in data, etc. are all examples that can dictate mining methodology choices.

1.9.4 Performance Issues: Many artificial intelligence and statistical methods exist for data analysis and interpretation. However, these methods were often not designed for the very large data sets data mining is dealing with today. Terabyte sizes are common. This raises the issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. Other topics in the issue of performance are incremental updating, and parallel programming

CHAPTER 2

INTRODUCTION TO BIOINFORMATICS

2.1 OVERVIEW

What is now called "Bioinformatics" began in the late 1960s and early 1970s, pretty much as the hobby of a small number of researchers scattered primarily through biology, mathematics and computer science departments at various universities and research centers. Now Bioinformatics Computing has become the buzzword throughout the world. The people from diverse backgrounds such as mathematics, computer science, biology, information technology, chemistry, medical science, agriculture engineering and life sciences are jumping on the bioinformatics track to get some pie of the benefits that may come. The subject of biology is not new. It is as old as the birth of any living species on earth. We have tons of information available on various biological entities and processes that has been written in different languages in different parts of the world. Apart from this the information is scattered so much that there is no standardization, classification, reusability and applicability of the data. So the biology field is combined with informatics so as to make effective use of computers for the better management of biological information. And new field known as Bioinformatics is developed. Bioinformatics computing is the major thrust area all over the world. It's a field of science in which various fields have merged into single discipline. Bioinformatics is playing a vital role in fulfilling the future expectations of the society(7).

2.2 WHAT IS BIOINFORMATICS

Bioinformatics is the science of storing, extracting, organizing, analyzing, interpreting, and utilizing information from biological sequences including nucleotide and amino acid sequences, protein domains, and protein structures, using networks of computers and databases. In simple words, bioinformatics is application of information technology to the storage, management and analysis of biological information, which is facilitated by

the use of computers. It also includes the development of new algorithms and statistics with which to assess relationships among members of large data sets. The use of computers to characterize the molecular components of living thing is bioinformatics(7).

2.3 FOUNDATIONS OF BIOINFORMATICS

Bioinformatics is a management for molecular biology. It is a study of how information is represented and transmitted in biological systems for various practical applications. Information retrieval has been the focus of widespread attention for last few decades. Bioinformatics is a quickly growing field. It began out of necessity in the late 1960s and 1970s when scientists began sequencing genes and proteins. They soon realized that the amount of data would be too large for humans to interpret without the aid of computers. Databases were created to store the data and tools had to be developed to search them. Algorithms that could search this type of data were developed and implemented into search tools like the Basic Local Alignment Search Tool (BLAST) and FASTA. The recent focus on accurate and fast access to biological information was triggered by the availability of a large volume of unstructured biological data. The purpose of information retrieval techniques is to retrieve all the relevant information. In order to efficiently retrieve relevant information and improve precision, modern information retrieval systems were developed(7).

2.4 THE SCOPE OF BIOINFORMATICS

The Bioinformatics computing is the area that involves applying computational powers of the computing tools & machines on the biological data so as to help the biologists and the life scientists in their work. The evolution of life is still a debatable topic. Everything not understood is attributed to some supreme force called GOD. We are not able to find out why the life is limited to 100 years or so? Why the people get old? Why the people cannot live a disease free life? There are so many

questions like this that still look for an answer(8). Bioinformatics is the area that is capable of solving all these queries and that is why people are having a lot of expectations from the people involved in this field. Billions of dollars are being invested to get something out of the Bioinformatics research. Bioinformatics include both biology and information technology.

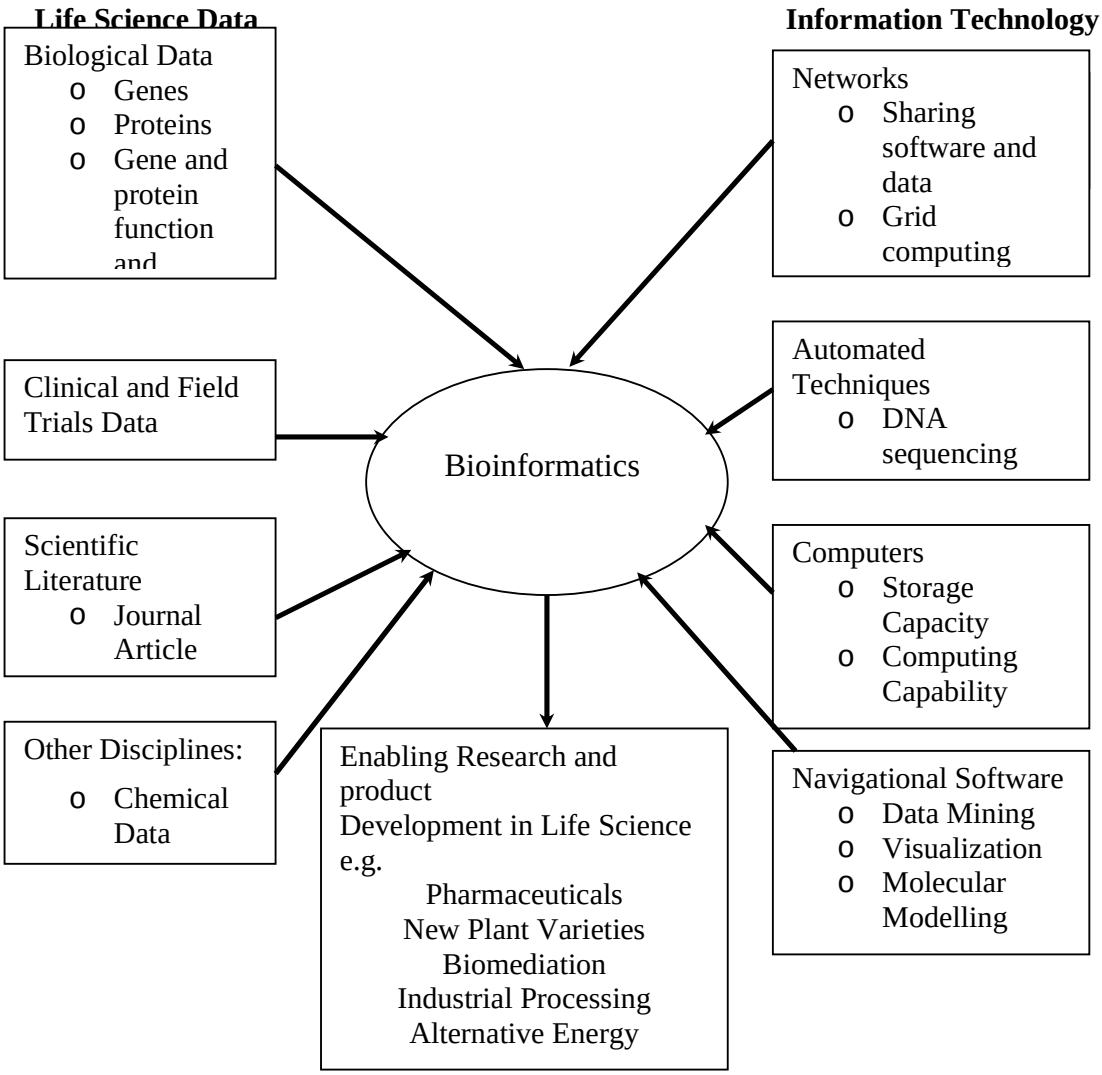


Figure 2.1 Components of The Bioinformatics Field

Both fields are combined in order to solve the biological problems in more efficient way using information technology.

2.5 NEED OF DATA MINING IN BIOINFORMATICS

Data in biology are very diverse and abundant. They can be catalogued and classified, but often cannot be easily summarized or abstracted using a formula. With the increase in biological knowledge, computer-based databases have become essential for this task.

Bioinformatics databases includes following types of databases:

- Sequence databases
- Structural databases
- Motif databases
- Genome databases

Moreover the data of even a single microorganism is very large. *Rickettsia conorii* is the smallest bacteria whose complete gene sequence is known. This bacterium is 1.3 million bp long and this size is still on the small side of bacteria. So with the significant growth of the amount of biomolecular data, it becomes increasingly important to develop new techniques for extracting knowledge from the data. Data mining is a fundamental operation in such a domain.

2.6 KDD FOR BIOINFORMATICS:ON WHAT KIND OF DATA

KDD for Bioinformatics can be applied on bimolecular data. Bimolecular Data consists of the following types:

- DNA (deoxyribonucleic acid)
- RNA (ribonucleic acid)
- Protein sequences (2D & 3D structures)

2.7 APPLICATIONS OF BIOINFORMATICS

Bioinformatics originated mainly as a set of computational techniques applied to the study of the large data sets being generated by the human

and other genome sequencing programmes. As it has the combination of various fields it is used in vast variety of applications:

2.7.1 Modeling And Prediction Of Enzyme Kinetics: The genome sequence and proteome provide only a template for the construction of components, and the real key to how living systems work requires an understanding of the many metabolic pathways and kinetic processes in the cell.

2.7.2 Gene Expression And Protein Arrays: The study of parallel gene expression and the protein products allows the effect of different conditions on thousands of genes to be measured simultaneously. The resulting profiles can then be clustered together and related to phenotype and physical expression.

2.7.3 Sequence Assembly: Whole genome sequencing methods require genomic DNA to be fragmented and the individual fragments sequenced separately. The genome sequence is then reconstructed using the information contained in overlapping regions.

2.7.4 Prediction Of Protein Function From Structure: As the number of known protein structures and sequences held in databases continues to grow, homology Modelling is expected to be the most reliable method of assigning likely functions and folds to many of the newly identified proteins.

2.8 BIOINFORMATICS TECHNIQUES

All areas of Bioinformatics are growing in scope and in complexity with the amount of stored information available, and the maturation of the associated information technology and analysis techniques has had to keep pace with the expansion of the science. So many of the data mining techniques are used in field of Bioinformatics like clustering, neural networks, nearest neighbour.

The main techniques used in Bioinformatics and their applications are shown in Table 2.1

Technique	Purpose
Database searching: Sequence alignment Gene ontology	 Identification of homologues Related documents, distant homologies
Statistical methods: Analysis of variance Significance testing Bayesian statistics Hidden Markov models	 Separation of error sources Sequence analysis
Principal component analysis of multivariate data	Determination of most significant variables in a data set
Clustering: Nearest-neighbour Agglomerative	 Grouping genes with similar behaviour
Dynamic programming	Solution of large-scale numerical problems (e.g. alignment of sequences, used in database searches)
Neural networks	Determination of networks of interactions
Decision trees	

Table 2.1 Techniques Used In Bioinformatics

2.9 CHALLENGES IN BIOINFORMATICS

The field of Bioinformatics is full of challenges. However a few of them are listed below.

2.9.1 Explosion Of Information

Need for faster, automated analysis to process large amounts of data.

Need for integration between different types of information (sequences, literature, annotations, protein levels, RNA levels etc…).

Need for "smarter" software to identify interesting relationships in very large data sets.

2.9.2 Lack Of "Bioinformatician"

Software needs to be easier to access, use and understand

Biologists need to learn about the software, its limitations, and how to interpret its results.

2.9.3 Lack Of Automated Tools

The automatic tools for analyzing the DNA sequences are very less. The results of existing tools are not very accurate.

2.9.4 Inapplicability Of Existing Algorithms

Though a lot of algorithms exist for data mining in biomolecular data but most of them does not give a correct result and generally they fail when applied to the real world data of Bioinformatics.

2.10 ISSUES IN BIOINFORMATICS

As a field, Bioinformatics is highly diverse, gaining greater importance in biology, and developing with great rapidity. However, the largest sources of uncertainty in the data do not relate directly to measurements, but to such influences as the inherent variability of biological test materials, their provenance, and the complexity of the biological systems from which responses arise. The issue will also become increasingly important where Bioinformatics techniques are applied in a regulatory context. Thus, uncertainty issues are likely to increase in importance(7).

In considering software issues, four different classes of software were identified:

- Database technology
- Database growth issues

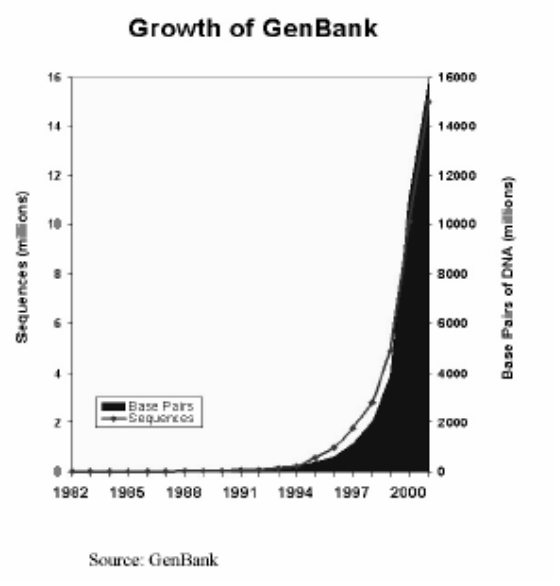
These are recognized issues, and many activities are under way in the Bioinformatics and instrument development community.

2.10.1 Database Technology Issues

Currently, a lot of Bioinformatics work is concerned with the technology of databases These databases include both "public" repositories of gene data like GenBank or the Protein Databank (the PDB), and private databases, like those used by research groups involved in gene mapping projects or those held by biotech companies.

2.10.2 Database Growth Issues

The quantity of biological data stored in central databases has increased enormously over the last five years (Figure 2.3). This in itself has brought new challenges for storage and interpretation, but the fact that the nature of biological data presents specific issues is also a factor to take into account. The second relevant issue is that of the raw data itself and how it is processed. Modern biological measurement techniques frequently involve the acquisition of large numbers of individual units of information and drawing useful conclusions is a matter of robust statistical analysis and replication where appropriate. This, combined with the treatment of raw data, presents a significant validation problem, which, in some emerging new fields, will need to be urgently addressed in the near future.



Figue2.2 Growths of Biological Databases

CHAPTER 3

BIOMOLECULAR SEQUENCE ALIGNMENT

3.1 OVERVIEW

In recent years, innovations where IT and the life sciences converge have created vast quantities of data. The development of automated DNA sequencing and other innovative methods have reduced the costs and time needed to discover the genetic makeup of various organisms. Before people can appreciate biological advances they need to have an understanding and familiarity with living organisms. Biology is at heart of information science, thus biology depends on informatics and understanding informatics furthers the advancement of biology. For understanding the living organisms we have to start from the most basic unit of life.

3.2 ORIGINS OF LIFE ON EARTH

Although Earth was created around 4.5 billion years ago, life began to exist not long after. The unique circumstances of our Solar System and our planet gave rise to life. However, Earth, for billions of years, has possessed all the materials and suitable conditions for supporting life. All living things possess the element carbon within them. While other elements were present, various chemical reactions began to take place, which would result in the creation of new compounds and elements. One of the family of compounds created over time were the amino acids, the building blocks of protein.

3.3 BIOMOLECULES

Bimolecules are the organic compounds, which form the basis of life that is they build up living system and are responsible for their growth and maintenance. These include carbohydrates, proteins, enzymes, nucleic acids, lipids, vitamins, hormones and compounds for storage and exchange of energy such as adenosine triphosphate (ATP)(8)

3.4 TYPES OF BIOMOLECULES

A diverse range of biomolecules exist, including:

- Small molecules
  - Lipids
  - Vitamin
  - Hormone
  - Carbohydrate, Sugar
  - Disaccharide
- Monomers
  - Amino acid
  - Nucleotide
  - Phosphate
- Polymers
  - Peptide, Protein
  - Nucleic acid, i.e. DNA, RNA

3.5 THE CELL - MOST BASIC UNIT OF LIFE

The cell is the smallest unit of life. All life is comprised of cells. Some life forms have only one cell, but many have millions of cells. The cell is made up of molecules, atoms and ions, just as are non-living things, but the cell is capable of carrying on metabolic processes and is capable of self-replication.

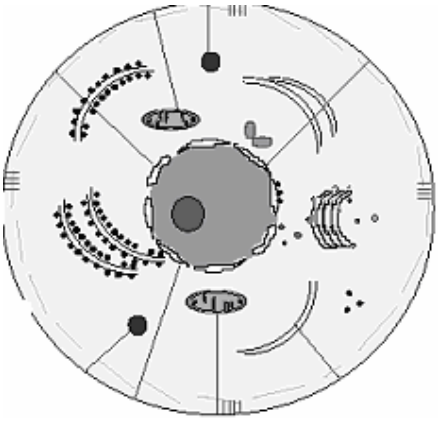


Figure 3.1 The Cell – Most Basic Unit Of Life

3.6 NUCLEIC ACIDS

Nucleic acid constitutes an important class of biomolecules, which are found in nuclei of all living cells in the form of nucleoproteins. Nucleic acids are genetic materials of cell and are responsible for transmission of hereditary effects from one generation to the other and also carry out the biosynthesis of protein(7). Nucleic acids are long, linear biomolecules that can have molecular weights of several millions

There are two types of nucleic acids:

- **Deoxyribonucleic Acid Better Known As DNA.**
- **Ribonucleic Acid Better Known As RNA**

DNA contains the “code of life.” During cell division, exact copies of DNA are made. RNA is essential for the synthesis of proteins in the cells. Messenger RNA (mRNA) is synthesized in the cell nucleus as a transcript of a specific part of DNA. DNA contains the "programmatic instructions" for cellular activities. The mRNA leaves the nucleus and enters the cell cytoplasm where it dictates the synthesis of proteins from amino acids. Transfer *RNA* (tRNA) delivers amino acids to the exact place in the cytoplasm where the proteins are synthesized.

3.6.1 Composition Of Nucleic Acid

Nucleic acids are composed of nucleotide monomers. Nucleotides have three parts:

- A Nitrogenous Base (purine or pyrimidine)
- A Five-Carbon Sugar
- A Phosphate Group

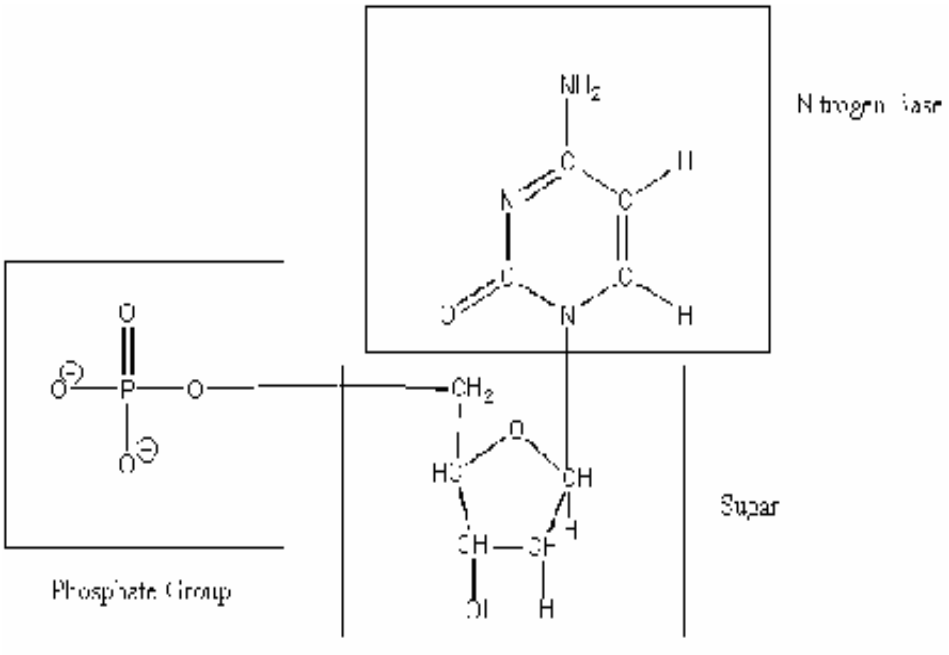


Figure 3.2 A Nucleotide

A Nitrogenous Base (Purine Or Pyrimidine): The nitrogenous bases are derivatives of purine and pyrimidine. Both DNA and RNA contain the purines Adenine (A) and Guanine (G). Of the pyrimidines, Thymine (T) and Cytosine (C) are components of DNA whereas Uracil (U) and Cytosine (C) are components of RNA.

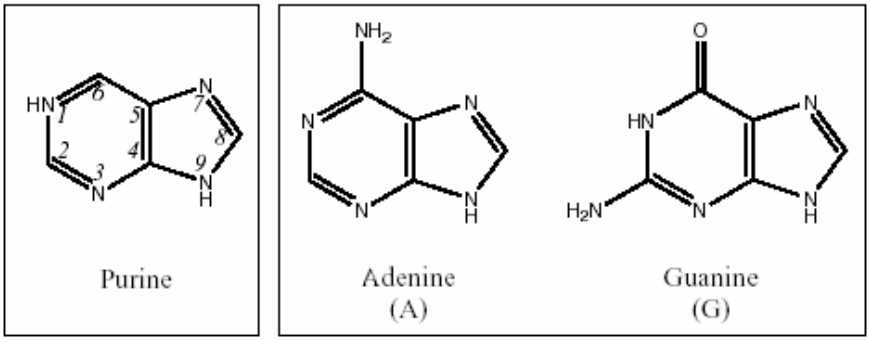


Figure 3.3 Left: The Structure Of Purine, Right: The Purine Derivatives Adenine And Guanine Are Found As Bases In Both DNA And RNA.

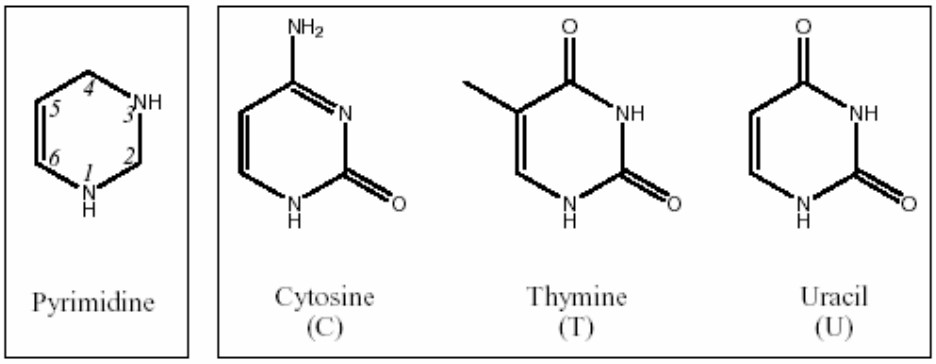


Figure 3.4 Left: The Structure Of Pyrimidine. Right: Cytosine And Thymine And Uracil

A Five-Carbon Sugar:

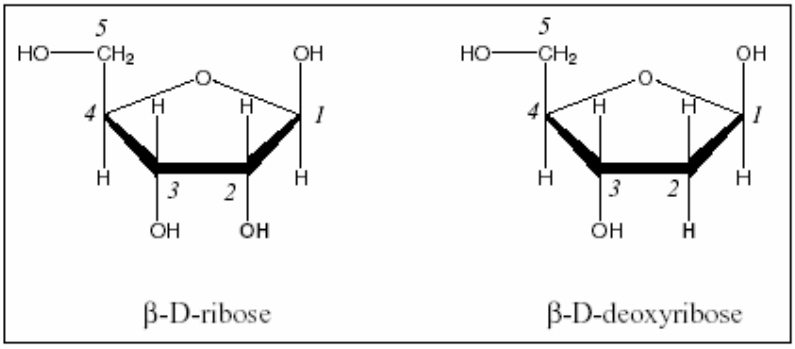


Figure 3.5 The Pentose *B*-D-Ribose Occurs In RNA. *B*-D-Deoxyribose is The Sugar Component In DNA.

3.6.2 Types Of Nucleic Acids

There are two types of nucleic acids:

- Deoxyribonucleic acid better known as DNA.
- Ribonucleic acid better known as RNA

3.6.2.1 DNA (Deoxyribonucleic Acid):

DNA determines all the characteristics of an organism, and contains all the genetic material that makes us who we are. This information is passed on from generation to generation in a species. DNA is arranged into a double helix structure where spirals of DNA are intertwined with one another continuously bending in on it but never getting closer or further away. One nucleotide, which is basic unit of DNA, is explained in following diagram

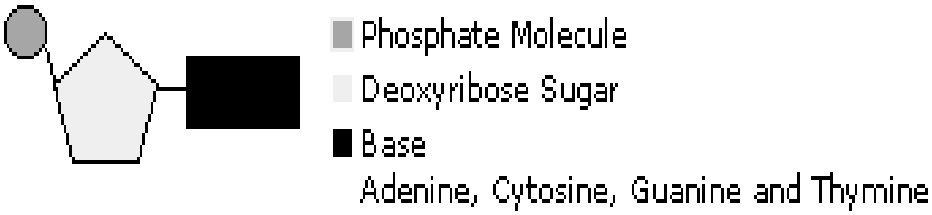


Figure 3.6 Basic Unit of DNA

There are four different types of nucleotide possible in a DNA sequence, adenine, cytosine, guanine and Thymine (can be replaced with A, C, G and T). There are billions of these nucleotides in our genome, and with all the possible permutations; this is what makes us unique. The following rules apply in regards to what nucleotides pair with one another.

- There are four possible types of nucleotide, adenine, cytosine, guanine and thymine.
- Thymine and adenine can only make up a base pair
- Guanine and cytosine can only make up a base pair

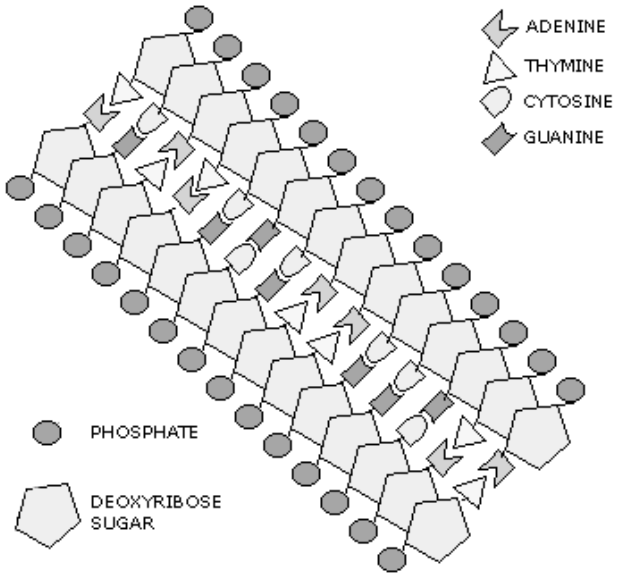


Figure 3.7 Pairing Of Nucleotides Forming DNA

3.6.2.2 RNA (Ribonucleic Acid): It is a long molecule but usually Single stranded, except when it folds back on itself. They differs chemically from DNA by containing ribose instead of deoxyribose & containing Uracil (U) instead of Thymine (T). So the only important differences between RNA and DNA are that

- ❖ RNA differs from DNA by one nucleotide.
- ❖ RNA comes as a single stranded.

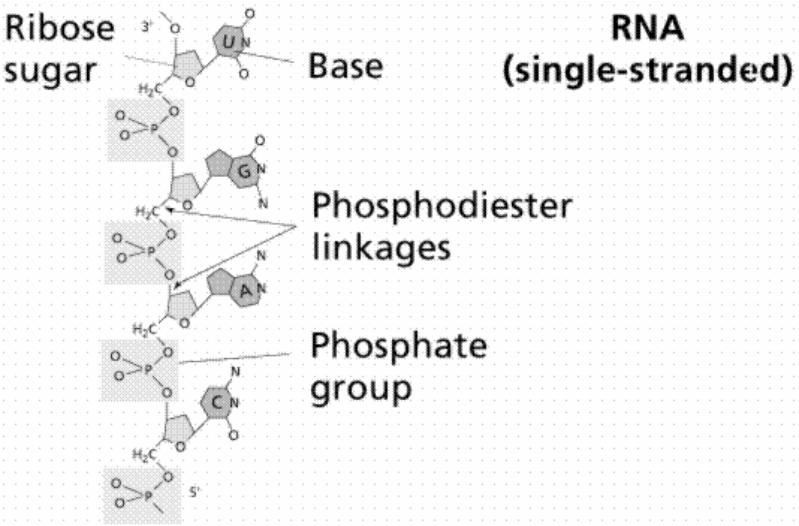


Figure 3.8 RNA Molecule

3.7 AMINO ACIDS, PEPTIDES AND PROTEINS

Amino acids are the building blocks for peptides and proteins and play an important part in metabolism. 20 different amino acids are found in living organisms. Proteins may consist of thousands of amino acids and can have molecular weights of up to several million Dalton (Da).

3.7.1 Amino Acids

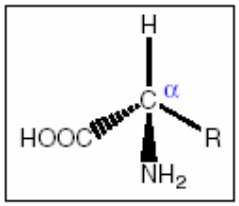


Figure 3.9 General Structure Of An *A*-L-Amino Acid.

Total number of amino acids known is 20, which are, describes in Table 3.1

S.No	1-Letter Code	3-Letter Code	Name
1	A	Ala	Alanine
2	R	Arg	Arginine
3	N	Asn	Asparagine
4	D	Asp	Aspartic acid
5	C	Cys	Cysteine
6	Q	Gln	Glutamine
7	E	Glu	Glutamic acid
8	G	Gly	Glycine
9	H	His	Histidine
10	I	Ile	Isoleucine
11	L	Leu	Leucine
12	K	Lys	Lysine
13	M	Met	Methionine
14	F	Phe	Phenylalanine
15	P	Pro	Proline
16	S	Ser	Serine
17	T	Thr	Threonine
18	W	Trp	Tryptophan
19	Y	Tyr	Tyrosine
20	V	Val	Valine

Table 3.1 20 Amino Acids With Their Official Codes

3.7.2 Peptides & Proteins

Peptides and proteins are macromolecules made up from long chains of amino acids joined head-to-tail via peptide bonds. The three-dimensional structure of a protein is very well defined and is essential for it to function.

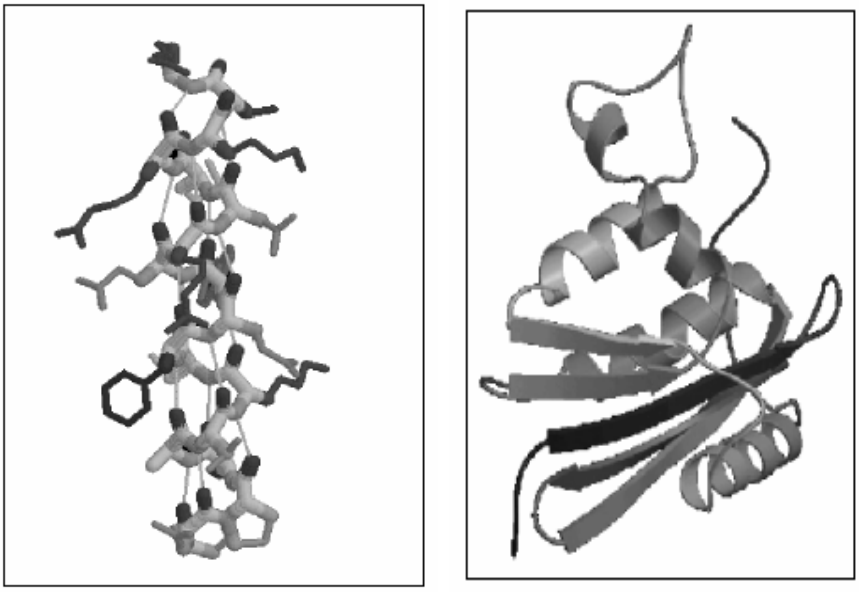


Figure 3.10 A Protein Molecule (Structure)

3.8 WHY SEQUENCES DIFFER

Biological sequences show complex patterns of similarity to one another. In this regard, they mirror the external morphologies of the organisms in which they reside. Sequences change over time due to four forces:

- Mutation
- Natural selection
- Genetic drift
- Neutral theory of Evolution.

3.8.1 Mutation

A mutation is simply a change in a DNA sequence. The reason for mutation is that many chemicals and conditions damage DNA, so its sequence either changes or ceases to be recognizable. Other reason for it is that the process of DNA replication is not perfect. The human genome is about three billion letters long, and the error rate of DNA replication is about on error in every 300 million letters, so you can expect about 10 mutations per genome duplication.

3.8.2 Natural Selection

Charles Darwin, a biologist from England, developed the theory of Natural Selection. Natural selection is considered to be the biggest factor resulting in the diversity of species and their genomes. It explains why organisms look the way they do and why they seem to fit their environments so well. Principals of the Natural Selection Theory are explained below:

- One of the prime motives for all species is to reproduce and survive, passing on the genetic information of the species from generation to generation.
- The organisms that die as a consequence of this competition were not totally random; Darwin found that those organisms more suited to their environment were more likely to survive.

Darwin's finches are an excellent example of the way in which species gene pools have adapted in order for long-term survival via their offspring. The Darwin's Finches diagram below illustrates the way the finch has adapted to take advantage of feeding in different ecological niche's.

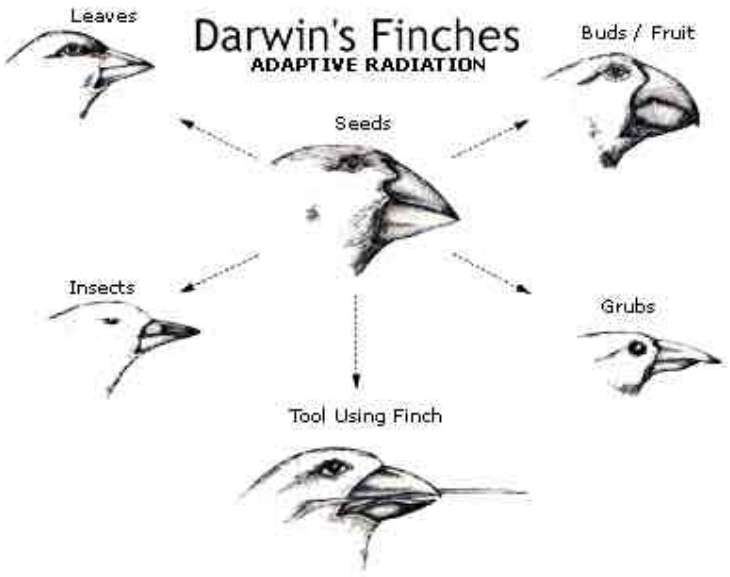


Figure 3.11 Natural Selection

3.8.3 Genetic Drift

Darwin was not aware of how variation is transmitted from generation to generation; he didn't have the concept of genes. Genes were introduced by Gregor Mendel to explain how hereditary information is transmitted from one generation to the next. Mathematical simulations show quite clearly that allele frequencies can change by purely random processes. This behavior is called genetic drift, and it's based on the fact that population aren't infinitely large.

3.8.4 Neutral Theory of Evolution

Motoo Kimura proposed the neutral theory of evolution in the late 1960s and early 1970s. The theory states that the majority of sequence evolution is purely random. According to Kimura, when one compares the genomes of existing species, or looks between a species and its forebears, the vast majority of single-nucleotide differences are selectively "neutral." That is, these differences do not influence the fitness of either the species or the individuals who make up the species. Such changes are presumed to have little or no biological effect. A second assertion or hypothesis of the neutral theory is that most evolutionary change is the result of genetic drift acting on neutral alleles.

3.9 SEQUENCE ALIGNMENT ALGORITHMS

Many different algorithms can be used as for sequence alignment. Different types of algorithms have different strengths, and there are specific types that are more applicable to Bioinformatics.

3.9.1 Exact Matching

Exact matching algorithms are used to find small identical segments within sequences. For example, if you want to find the number of times the word 'dad' appears in the sequence: daddaadadadadadda, an exact matching algorithm would tell you that it appears 5 times and where each appearance started.

3.9.2 Pairwise Sequence Alignment

Pairwise sequence alignment is based on the similarities between two sequences. Two fast search tools use Pairwise sequence alignment algorithms, the basic local alignment search tool (BLAST) and FASTA.

3.9.3 Multiple Sequence Alignment

Multiple string comparison is a very important problem in computational biology. Instead of comparing a test sequence to a longer pattern, multiple string comparison would allow that test sequence to be compared to many different patterns at the same time. Multiple sequence alignment is now used extensively in molecular biology, and has functions which include:

- To find diagnostic patterns
- To characterize protein families
- To detect or demonstrate homology between new sequences and existing families of sequences
- To help predict the secondary and tertiary structures of new sequences.
- To suggest oligonucleotide primers for PCR
- As an essential prelude to molecular evolutionary analysis.

This method has two major issues, however. First, a mistake in one of the early Pairwise alignments cannot be corrected. This affects the accuracy of the final multiple alignment. The second major issue has to do with how to allow gaps in sequences and which substitution matrix is used.
The Multiple Alignment Algorithm has three main steps:

1) All pairs of sequences are aligned separately in order to calculate a distance matrix giving the divergence of each pair of sequences;
2) A guide tree is calculated from the distance matrix;
3) The sequences are progressively aligned according to the branching order in the guide tree.

CHAPTER 4

SEQUENCE ANALYSIS TOOLS

4.1 OVERVIEW

All species in the world are unique, yet there is some degree of relatedness in all species. In order to fine that similarity we have to analyze the underlined sequences, which make the species. These sequences may be the most basic one like DNA sequences or the gene sequences. Determining and using specific DNA sequences is the backbone of molecular biology, and sequencing is the gold standard in DNA identification. Thus the comparison of gene sequences or biological sequence analysis is one of the processes used to understand sequence evolution. Just as the ancient Greeks used comparative anatomy to understand the human body, today we can use comparative sequence analysis to understand genomes. There is variety of different tools available to perform sequence analysis. Sequences can be aligned across their entire length (global alignment) or only in certain regions (local alignment). This is true for pairwise and multiple alignments.

4.2 BLAST

Basic Local Alignment Search Tool Is used to compare a query sequence with those contained in databases by aligning the query sequence with previously characterised genes, therefore helping in identifying genes. The emphasis of this tool is to find regions of sequence similarity. The fundamental unit of BLAST algorithm output is the High-scoring Segment Pair (HSP). An HSP consists of two sequence fragments of arbitrary but equal length whose alignment is locally maximal and for which the alignment score meets or exceeds a threshold or cutoff score. A set of HSPs is thus defined by two sequences, a scoring system, and a cutoff score; this set may be empty if the cutoff score is sufficiently high. The sensitivity and speed of the programs can be adjusted via the standard BLAST algorithm parameters W, T, and X. The approach to similarity

searching taken by the BLAST programs is first to look for similar segments (HSPs) between the query sequence and a database sequence, then to evaluate the statistical significance of any matches that were found, and finally to report only those matches that satisfy a user-selectable threshold of significance. There are many variants of BLAST available, which can be used at different situations.

4.2.1 Steps for Running BLAST(12)

Consider a sequence, and look for sequences that are similar in the EMBL Nucleotide Sequence Database. This sequence is a real entry in this database, so we will expect to find a sequence that is a perfect match to our test sequence.

- The sequence, is entered into the textbox in fasta format, which consists of a one-line header starting with a ">" symbol, followed by the sequence name. The sequence is then entered on new line(s).

- The BLAST program is chosen, which is designed to search a nucleotide query sequence against a databank

- The number of scores (hits to the database) is limited to 10 and the number alignment of these against the query sequence is limited to 5, this is done to limit the size of the output results.

- Other options have been left on "default"

4.2.2 Result of BLAST Alignment

The one-line sequence descriptions and summaries of results are useful for identifying biologically interesting database matches and correlating this interest with the statistical significance estimates. We will first consider the score list, the search was limited to 10 scores in this search, and one result is taken and is explained as:

EM_MUS: MMDYSA M68859 Mouse dystrophin mRNA, complete cds. 69075

- **EM_MUS** is the EMBL database division that the entry is in, in this case EMBL mouse (**EM**BL **MUS** musculus).
- **MMDYSA** is the entry name in the database and the link to the entry.
- **M68859** is the accession number associated with this entry.
- **Mouse dystrophin mRNA,**
- **Complete cds** is the description of the database entry.
- **69075** is the high score of the entry, the greater this value, the better the match between the query sequence and the database entry.

4.2.3 Features Of BLAST

- **It uses Heuristic approach:** BLAST is not guaranteed to find the best alignment between your query and the database; it may miss matches. This is because it uses a strategy, which is expected to find most matches, but sacrifices complete sensitivity in order to gain speed.
- **Local alignments:** BLAST uses local alignments for matching sequences rather than global alignments. BLAST tries to find patches of regional similarity, rather than trying to find the best alignment between your entire query and an entire database sequence.
- **Ungapped alignments:** Alignments generated with BLAST do not contain gaps. BLAST's speed and statistical model depend on this, but in theory it reduces sensitivity. However, BLAST will report multiple local alignments between your query and a database sequence.
- **Rapid:** BLAST is extremely fast. It does not explore the entire search space between two sequences as it uses the three layers of

rules to sequentially refine potential HSPs. This minimization of search space is the key to its speed but at the cost of a loss in sensitivity.

4.2.4 Limitations of BLAST

- Needs islands of strong biology and homology, which is similarity in DNA or protein sequences between individual of the same species or different species.
- Limits on the combination of scoring and penalty values
- The variants (blastx, tblastn, tblastx) use 6-frame translation-miss sequences with frameshifts)
- Finds and reports ONLY local alignments

4.2.5 Improving BLAST Sensitivity

BLAST is widely used for searching protein, nucleotide databases for sequence similarities, especially distant homologies. Position specific score matrices are constructed during its running. These give the program the power to capture remote relations of a query sequence. BLAST needs multiple iterations in most circumstances and is time-consuming. We modified the vector seed-optimizing algorithm and used dynamic programming to apply to position dependent scoring systems. The annotation aids by computational methods aims to execute repetitive and time-consuming tasks, speeding up the analysis of biological data. The use of an automatic re-annotation module can make the work easier and much faster. We also modified substitution matrixes.

4.3 CLUSTERW

ClusterW is a general purpose multiple sequence alignment program for DNA or proteins. It produces biologically meaningful multiple sequence alignments of divergent sequences. It calculates the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen. Multiple alignments of protein sequences are important tools in studying sequences. The basic information they provide

is identification of conserved sequence regions. This is very useful in designing experiments to test and modify the function of specific proteins, in predicting the function and structure of proteins, and in identifying new members of protein. ClustarW is a fully automatic program for global multiple alignment of DNA and protein sequences. The alignment is progressive and considers the sequence redundancy. The program has some adjustable parameters with reasonable defaults. ClustarW currently supports multiple sequence formats(15). These are:

NBRF/PIR

EMBL

UniProt/SwissProt

GDE GCG/MSF

4.3.1 Steps for Running Clusterw

We will consider aligning the several sequences, represented by the accession numbers. These can be added using accession number or in fasta format.

- The multiple sequences were uploaded in fasta format, which consists of a one-line header starting with a ">" symbol, followed by the sequence name/description. The sequence is then entered on new line(s).

- The title of the alignment was changed to MHC align. Because it is only used for multiple alignment. The sequence which we have entered is compared with different sequences in database.

- The comparison is done in progressive manner.
- Other options were left on "default".

4.3.2 Result of Clusterw Alignment

The plain text version of the alignment result will be temporarilly stored in an .aln file. An alignment with display by default the following symbols denoting the degree of conservation observed in each column:

- "*" Means that the residues or nucleotides in that column are identical in all sequences in the alignment
- ":" Means that conserved substitutions have been observed, according to the colour table.
- "." Means that semi-conserved substitutions are observed.

4.3.3 Features of Clusterw

ClusterW is a multiple alignment program for DNA and proteins. The program is now more compatible with GCG and has new features for more powerful aligning of diverged sequences. Some of the features of ClustalW include:

- It can be used via an X windows based user interface, ClustalX
- No longer has a limitation to the number or length of sequences in the input file.
- Output file default order of the sequences has changed. Instead of maintaining the input file order the sequences are now in the alignment order. Thus, closely related sequences are grouped.

4.3.4 Limitations of Clusterw

- The input order in analyzing the bootstrapped samples is not randomized; therefore, we have no phylogenetic information at all and thus we get 100% bootstrap values.
- If we have very different branch lengths, even if we have a "molecular clock" running, long branches have the tendency to attract each other.

4.3.5 Improving ClusterW Sensitivity

The sensitivity of the commonly used progressive multiple sequence alignment method has been greatly improved for the alignment of divergent protein sequences. Firstly, individual weights are assigned to each sequence in a partial alignment in order to downweight near-duplicate sequences and upweight the most divergent ones. Secondly, amino acid substitution matrices are varied at different alignment stages according to

the divergence of the sequences to be aligned. Thirdly, residue specific gap penalties and locally reduced gap penalties in hydrophilic regions encourage new gaps in potential loop regions rather than regular secondary structure. Fourthly, positions in early alignments where gaps have been opened receive locally reduced gap penalties to encourage the opening up of new gaps at these positions. These modifications are incorporated into a new program, CLUSTAL W that is freely available.

4.4 FASTA

FASTA (pronounced FAST-Aye) stands for **FAST All,** reflecting the fact that it can be used for a fast protein comparison or a fast nucleotide comparison. This program achieves a high level of sensitivity for similarity searching at high speed. This is achieved by performing optimised searches for local alignments using a substitution matrix, in this case a DNA identity matrix(16). The high speed of this program is achieved by using the observed pattern of word hits to identify potential matches before attempting the more time consuming optimised search. The trade-off between speed and sensitivity is controlled by the ktup parameter, which specifies the size of the word. Increasing the ktup decreases the number of background hits. Practically, FastA is a family of programs, allowing also cross queries of DNA versus protein.

4.4.1 Steps for Running FASTA

We will consider a sequence, and look for sequences that are similar in the EMBL Nucleotide Sequence Database. This sequence is a real entry in this database, so we will expect to find a sequence that is a perfect match to our test sequence. Also we expect to find similar sequences, perhaps from closely related animals, or from nucleotide coding sequences for closely related proteins.

- The sequence is entered into the textbox in fasta format, which consists of a one-line header starting with a ">" symbol, followed by the sequence name. The sequence is then entered on new line(s).

- "Email" is chosen so that I will have the results delivered to the email address as soon as they are available. As a fasta search is very resource-intensive, it is not usually possible to search the whole of the EMBL database interactively.

- The title of the search is left as "_Sequence" although you can give your search title any name you wish to help you identify the results.

- The fasta program is used, which is designed to search a nucleotide query sequence against a DNA databank

- The number of scores (hits to the database) to is limited 10 and the number of alignments of these against the query sequence to is also limited 10, this is in order to limit the size of the output results.

- Other options have been left on "default".

4.4.2 Result of FASTA Alignment

The one-line sequence descriptions and summaries of results are useful for identifying biologically interesting database matches and correlating this interest with the statistical significance estimates. Considering the top (best) match with our query sequence:

EM_HUM:HS405721 U40572.1 Human beta2-syntrophin (1700) [f] 8491 1435 0

- First entry is the EMBL database division that the entry is in, in this case EMBL human (**EM**BL **HUM**AN).

- **HS405721** is the entry name in the database.

- **U40572.1** is the accession number associated with this entry.

- **Human beta2-syntrophin** is the start of the description of the database entry.

- **(1700)** Is the number of nucleotides present in the entry.

- **[f]** This stands for forward or reverse, and represents which strand from from which the alignment was based.
- **8491** is the optimised score (joins of gapped fragments).
- **1435** is the bits score, which is a normalized score calculated from the raw score.

4.4.3 Features of FASTA

Some of the features of FASTA are:

- **Local alignments:** FASTA tries to find patches of regional similarity, rather than trying to find the best alignment between your entire query and an entire database sequence.
- **Gapped alignments:** Alignments generated with FASTA can contain gaps.
- **Rapid:** FASTA is quite fast. You can either run the program locally or send queries to an E-mail server
- **Heuristic:** FASTA is not guaranteed to find the best alignment between your query and the database; it may miss matches

4.4.4 Limitations of FASTA

- Larger ktuple increases speed since fewer “hits” are found but it also decreases sensitivity for finding similar but not identical sequences since exact matches of this length are required
- FASTA can miss significant similarity since, for proteins, similar sequences do not have to share identical residues and for nucleic acids, due to codon “wobble”, DNA sequences may look like XXyXXyXXy where X’s are conserved and y’s are not.
- FASTA-DBS and FASTA-FILES have no locking mechanism. If the data files were to change while someone had a FASTA-DB or FASTA file open, that person would be hosed.

4.4.5 Improving Sensitivity of FASTA

It is almost always advantageous to perform similarity searches with protein rather than DNA sequences. This is true for three reasons: (1) The information content per residue of protein sequence is greater than DNA because the amino acid alphabet is much larger than the DNA alphabet (ii) No identical amino acids can be scored for similarity using a mutation frequency matrix such as PAM or BLOSUM; and (iii) The protein databanks are much smaller than the DNA databanks, so the likelihood of chance matches is reduced. So FASTA has better sensitivity than BLAST. The problem of similarity searching with DNA sequences that contain frameshifts has been addressed. FASTA 3.0 now contains the program TFASTX, which compares a DNA sequence to a protein database, translating in three frames and allowing for frameshifts. FASTA 3.0 uses the same methods and statistics as FASTA 2.0, but it is designed in a modular fashion and runs not only on conventional UNIX workstations but also on multiprocessors in parallel(16).

4.5 READSEQ

This program reads and writes nucleotide and protein sequences in various useful formats. Its main contribution to bioinformatics is it takes on the job of guessing what your input biosequence data format is, and converting it to what your software knows how to handle. Don Gilbert developed it. Readseq was written originally around 1989 a component of a sequence analysis program, in Pascal. But now it is available in two versions(20).

i) Classic - the 1993 release, in C code

ii) Java - the 1999 release, in Java code

Readseq was converted to a C program in early 1990's. In the classic version a small, simple command-line interface was added. It is particularly useful as it automatically detects many sequence formats, and interconvert among them.

4.5.1 Classic Version

- Fixed Olsen format input to handle files w/ more sequences, not to mess up when more than one seq has same identifier, and to convert number masks to symbols.
- Added a few new formats like GCG MSF multi sequence file format, PIR/CODATA format, NCBI ASN.1 sequence file format, PAUP multi seq format, Phylip formats (interleave & sequential)
- Phylip format can now be used as input. Options to reverse-compliment and to degap sequences have been added. A menu addition for users of the GDE sequence editor is included.
- Reverted Genbank output format to fixed left margin (change in 30 Dec release), so GDE and others relying on fixed margin can read this.

4.5.2 Java Version

The java version 2, first available in 1999, continues support for the "classic" C version, in that it includes the same command-line options. The main addition in the Java version can be listed as following:

- This version handle sequence documentation; the original ignored all but a few fields of information other than sequence data.
- Version 2 added document and feature table parsing, which has become an essential need in sequence manipulation. One can currently extract sequence of a given set of features from a bio sequence with feature tables Release 2.1 further enhances and along with many other additions.
- It has pretty print menu whose options set parameters for the Pretty-print format, such as numbering and labels
- It has new sequence format conversions, and a lot of bug fixing.
- Readseq handles sub ranges as the intersection with a given feature location. Sub range math respects complement orientation, as well as the end value to specify end of a given location.

- The java version includes a classic Command line interface. Besides command line interface it includes Graphic User Interface for those who prefer not to learn the many command line options.
- This version also includes a Common gateway interface (CGI) for use in web server. The options in this 'cgi' interface include all of the command line options. With a few additions.
- A Perl script to convert readseq source to javac compatible form is included.
- Various bug fixes; Java 1.2/3 compatibility
- This java version is also more efficient, working faster than the compiled C classic version.

4. 5.3 Features Of Readseq

ReadSeq has lot of features. A few of them are listed blow:

- Release 2 in Java of readseq has been completely revised to object-oriented and efficient data handling. It is faster than release 1, which was compiled C code, by a factor of 2 to 4 times.
- It has user interface for those who prefer not to learn the many command line options
- Software is freely available to the public for use.
- It automatically detects many sequence formats, and interconvert among them.

4.5.4 Limitations Of Readseq

- The main limitation is memory usage - it is not optimized for large data sets, but reads all of a sequence record in memory, generating numerous objects for documentation and a byte array for sequence.
- It’s not efficient enough to handle large sequences (genome sized or full GenBank/EMBL data release files).
- In its current Java incarnation, interfacing Readseq with other languages is done mainly through command-line calls to the main program. If the programs are in Perl, we may want to use the collection with its SeqIO package.

- Readseq is currently not recommended for very large (100+MB) sequence files, whether as a single record or multiple records.

4.6 BLAT

BLAT (BLAST-Like Alignment Tool) is a very fast sequence alignment tool similar to BLAST, but it is structured differently. BLAT was written by Jim Kent. Blat is commonly used to look up the location of a sequence in the genome or determine the exon structure of an mRNA. Blat produces two major classes of alignments(11):

- At the DNA level between two sequences that are of 95% or greater identity, but which may include large inserts, and
- At the protein or translated DNA level between sequences that are of 80% or greater identity and may also include large inserts.

The output of BLAT is flexible. By default it is a simple tab-delimited file which describes the alignment, but which does not include the sequence of the alignment itself. On DNA, Blat works by keeping an index of an entire genome in memory. Thus, the target database of BLAT is not a set of GenBank sequences, but instead an index derived from the assembly of the entire genome. The index -- which uses less than a gigabyte of RAM -- consists of all non-overlapping 11-mers except for those heavily involved in repeats. This smaller size means that Blat is far more easily mirrored. Blat of DNA is designed to quickly find sequences of 95% and greater similarity of length 40 bases or more. It may miss more divergent or short sequence alignments.

On proteins, BLAT uses 4-mers rather than 11-mers, finding protein sequences of 80% and greater similarity to the query of length 20+ amino acids. The protein index requires slightly more than 2 gigabytes of RAM. BLAT has several major stages. It uses the index to find regions in the genome likely to be homologous to the query sequence. It performs an alignment between homologous regions. It stitches together these aligned regions (often exons) into larger alignments (typically genes). Finally, BLAT revisits small internal exons possibly missed at the first stage and

adjusts large gap boundaries that have canonical splice sites where feasible.

4.6.1 BLAT Is Different From BLAST

- Speed (no queues, response in seconds) at the price of lesser homology depth.
- The ability to submit a long list of simultaneous queries in fasta format.
- Five convenient output sort options.
- A direct link into the UCSC browser.
- Alignment block details in natural genomic order.
- An option to launch the alignment later as part of a custom track.
- BLAT will do in a day what BLAST will do in a month, and in many cases a single CPU will handle the load. The output is small and easy to parse. Splice sites are found without exon bleed-over. Give your staff and your server a rest and switch to BLAT.

There are three main programs in the BLAT suite: a stand-alone program called ‘blat’. A server, which maintains an index of a genome in memory called ‘gfServer’, and a client that, can query the index over the network called ‘gfClient’. Since it takes some time (10 to 25 minutes) to index an entire genome, the gfServer/gfClient model is best suited for situations where interactive users wish to quickly locate a few sequences in the genome. Blat source and executables are freely available for academic, nonprofit and personal use.

4.6.2 Steps For Running BLAT

BLAT is bioinformatics software a tool that performs rapid mRNA/DNA and cross-species protein alignments. BLAT is more accurate and 500 times faster than popular existing tools for mRNA/DNA alignments and 50 times faster for protein alignments at sensitivity settings typically used when comparing vertebrate sequences. BLAT is not BLAST. All alignment programs break the alignment problem into two parts:

- Initially in a “search stage,” the program detects regions of the two sequences, which are likely to be homologous.
- The program then in an “alignment stage” examines these regions in more detail and produces alignments for the regions, which are indeed homologous according to some criteria. The alignment stage performs a detailed alignment between the query sequence and the homologous regions. For historical reasons, the alignment stage for nucleotide and protein alignments is quite different. Both have limitations, and are good candidates for future BLAT upgrades. On the other hand, both are quite useful in their present form for sequences, which are not too divergent.

The goal of the search stage is to detect the vast majority of homologous regions while reducing the amount of sequence that is passed to the alignment stage.

4.6.3 Results of BLAT Alignment

BLAT implements a very quick algorithm for finding multiple nearby perfect matches, which allows the search stage to be specific enough that the genome itself can be kept on disk and only the index kept in RAM in memory in the client/server mode. BLAT is able to unsplice all the human mRNA in GenBank, including the ESTs, in less than a day on a 100-CPU computer cluster. BLAT working in translated mode is capable of rapidly aligning data across vertebrate species without significant compromise.

4.6.4 Features Of BLAT

Some of the features of BLAT are:

- BLAT is a very effective tool for doing nucleotide alignments between mRNA and genomic DNA taken from the same species.
- Genomic coordinates of mRNA or protein within a given assembly can be found with the use of Blat.
- The exon structure of a gene can be determined.
- A coding region within a full-length gene is displayed.
- It isolates an EST of special interest as its own track.
- It is used for searching of gene family members.

- Human homologs of a query from another species are easily found with Blat.

4.6.5 Limitations Of BLAT

- Program-driven use of BLAT is limited to a maximum of one hit every 15 seconds and no more than 5,000 hits per day.
- For users with high-volume Blat demands, we recommend downloading Blat for local use.
- Up to 25 sequences can be submitted at the same time.
- The BLAT program requires approximately two bytes for each base in the genome in DNA mode, and three bytes for each base in translated mode. The other programs use relatively little memory.

4.6.6 Improving BLAT Sensitivity

Analyzing vertebrate genomes requires rapid mRNA/DNA and cross-species protein alignments. A new tool, BLAT, is more accurate and 500 times faster than popular existing tools for mRNA/DNA alignments and 50 times faster for protein alignments at sensitivity settings typically used when comparing vertebrate sequences. BLAT's speed stems from an index of all non-overlapping K-mers in the genome. This index fits inside the RAM of inexpensive computers, and need only be computed once for each genome assembly. BLAT has several major stages. It uses the index to find regions in the genome likely to be homologous to the query sequence. It performs an alignment between homologous regions. It stitches together these aligned regions (often exons) into larger alignments (typically genes). Finally, BLAT revisits small internal exons possibly missed at the first stage and adjusts large gap boundaries that have canonical splice sites where feasible.

CHAPTER 5

COMPARISON OF TOOLS AND RESULTS

5.1 OVERVIEW

Bioinformatics is a buzzword that is becoming increasingly audible in the world. Bringing the life sciences and information technology together to develop powerful, user-friendly software packages that enable scientists to efficiently examine, interpret and store data to speed up discovery and advance scientific knowledge. Various software packages of automated tools have been developed that had improved the efficiency of much biological research. One most important task in bioinformatics is DNA sequence analysis. In this chapter DNA sequence analysis tools are discussed.

5.2 NEED FOR SEQUENCE ANALYSIS TOOLS

All species in the world are unique, yet there is some degree of relatedness in all species. In order to fine that similarity we have to analyze the underlined sequences, which make the species. These sequences may be the most basic one like DNA sequences or the gene sequences. In recent years, DNA sequencing has become a familiar term to everyone. Thus the analysis comparison of gene sequences or biological sequence analysis is one of the processes used to understand sequence evolution. There is variety of different tools available to perform sequence analysis. Sequences can be aligned across their entire length (global alignment) or only in certain regions (local alignment).

5.3 COMPARISON CRITERIA

All known tools have some advantages over the other. All have different functions, different areas of applications. So according to the different situations different tools are used. I am analyzing four tools BLAST,

CLUSTERW, FATSA, and BLAT. I am analyzing these tools and the criteria for comparison are one from the following.

- **Algorithmic Based**
- **Parameter Based**

5.4 ALGORITHMIC BASED

All four tools have some underlined algorithm. So the algorithms are compared to know that which algorithm is more efficient than the other. What steps of the particular algorithm are different and what are same.

5.4.1 BLAST Algorithm

BLAST (Basic Local Alignment Search Tool), is a sophisticated software package for rapid searching of nucleotide and protein databases developed by Altschul. BLAST uses a heuristic algorithm, which seeks local as opposed to global alignments and is therefore able to detect relationships among sequences, which share only isolated regions of similarity. BLAST searches the database in two phases. First it looks for short subsequences, which are likely to produce significant matches, and then it tries to extend these subsequences. In brief BLAST algorithm is described by three steps, which are explained as follows(12):

(1) In step 1, BLAST filters low complexity regions removes them from the query sequence. Next, BLAST generates a list of all of short sequences, or words, that make up the query. Then, BLAST uses a scoring matrix to determine all high-scoring matching words for each word in the query sequence. There is a trade-off at this stage between speed and sensitivity: a higher threshold gives greater speed but increases the chance of missing relevant pairs. The steps are:

- For the query find the list of high scoring words of length w.
- For a given word length *w* (usually 3 for proteins) and a given score matrix create a list of all words (*w*-mers) that can can score $>T$ when compared to *w*-mers from the query.

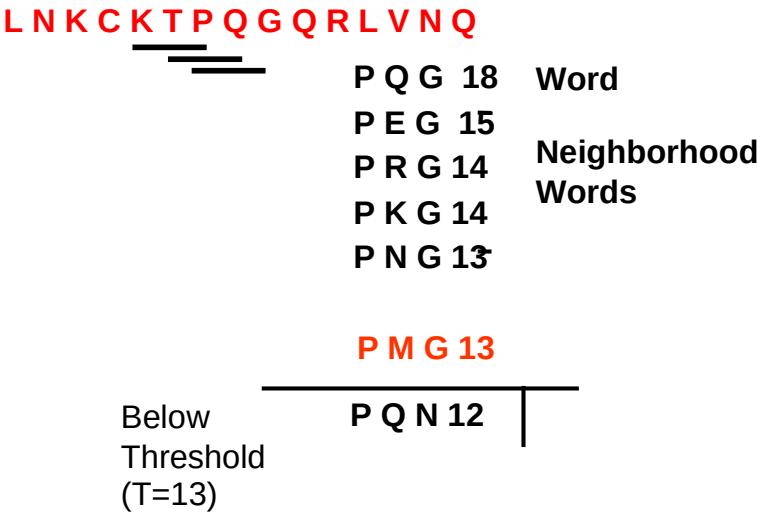


Figure 5.1 Word Length for BLAST

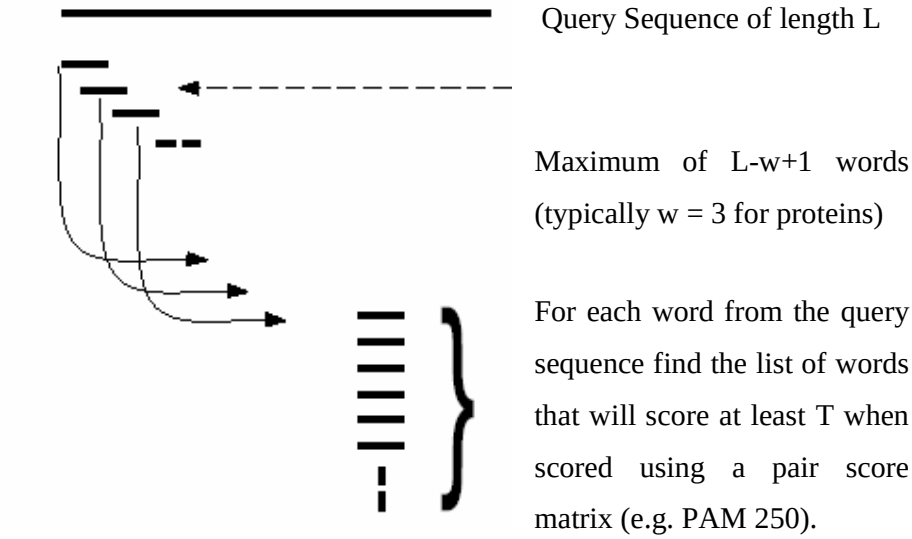


Figure 5.2 List of HSP for BLAST

(2) In the second step, BLAST searches for exact matches for the word list. Because BLAST has already pre-processed and indexed the databases for the occurrence of all words in each sequence in the database, this search is extremely fast.

❖ Compare the word list to the database and identify exact matches.

❖ Each neighbourhood word gives all positions in the database where it is found (hit list).

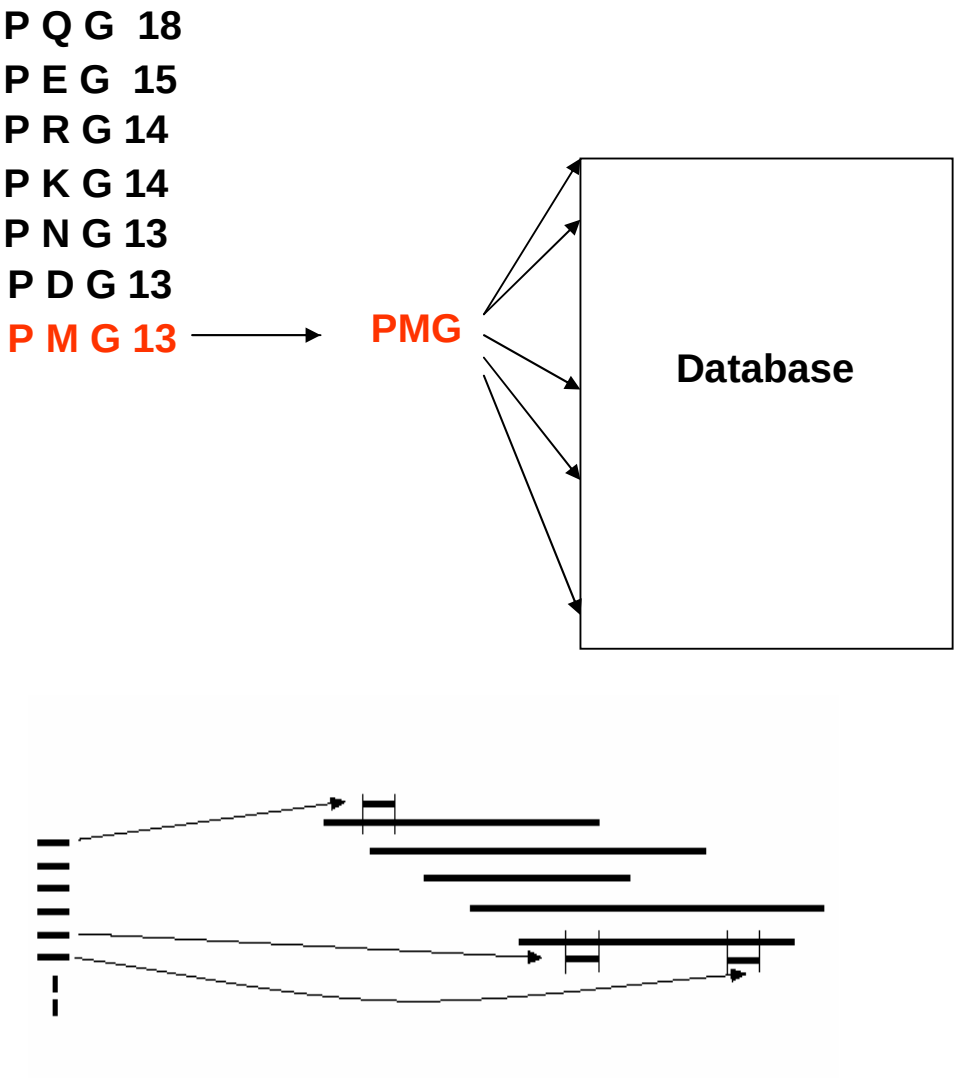


Figure 5.3 Exact Matches of Words Form Word List

(3) In the third step, the original BLAST method tried to extend the alignment from the matching words in both directions as long as the score continued to increase.

For each word match, extend alignment in both directions to find alignments tat score greater than score threshold S. The program tries to extend matching segments (seeds) out in both directions by adding pairs of

residues. Residues will be added until the incremental score drops below a threshold

Query: **325**

SLAALNKT**PQG**QRLVNQWIKWLPDNRPDLEHNDENEA 365

+LA+ L+ TP G R++ +W+ P+ D +

A

Sbjct: **290** TLASLKWT**PMG**SRMLKRWH K

KPLEAPDTQAGDAWA 330

High-Scoring Segment Pair(HSP)

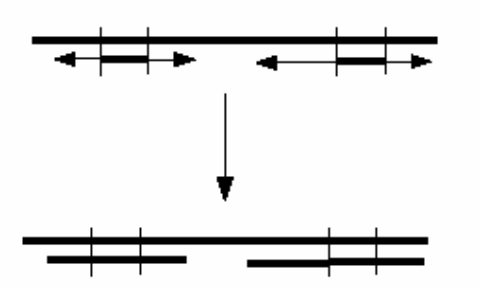


Figure 5.4 Maximal Segment Pairs (MSPs)

The resulting alignment was called a high-scoring pair, or HSP. These joined regions are then extended using the same method as in the original BLAST. Next, BLAST determines whether each score found by one of the above methods is greater in value than a given cutoff score *S*. Finally, BLAST determines the statistical significance of each score, initially by calculating the probability that two random sequences, one the length of the query sequence and the other the length of the database with the same composition could produce the calculated score.

5.4.2 CLUSTERW Algorithm

Multiple alignments of protein sequences are important tools in studying sequences. The basic information they provide is identification of conserved sequence regions. This is very useful in designing experiments to test and modify the function of specific proteins, in predicting the function and structure of

proteins. ClusterW is a fully automatic program for global multiple alignment of DNA and protein sequences. The alignment is progressive and considers the sequence redundancy. ClusterW is a general purpose multiple sequence alignment program for DNA or proteins. It calculates the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen. In brief clusterW algorithm is described by three steps, which are explained as follows(15):

The basic multiple alignment algorithm consists of three main stages:

1) The distance matrix/pairwise alignments

In the original clusterW programs, the pairwise distances were calculated using a fast approximate method. This allows very large numbers of sequences to be aligned, even on a microcomputer. The scores are calculated as the number of k-tuple matches in the best alignment between two sequences minus a fixed penalty for every gap. We now offer a choice between this method and the slower but more accurate scores from full dynamic programming alignments using two gap penalties and a full amino acid weight matrix. These scores are calculated as the number of identities in the best alignment divided by the number of residues compared. In figure 5.5 we give the 7x7 distance matrix between the 7-globin sequences calculated using the full dynamic programming method.

2) The guide tree

The trees used to guide the final multiple alignment process is calculated from the distance matrix of step 1 using the Neighbour-Joining method. This produces unrooted trees with branch lengths proportional to estimated divergence along each branch. The root is placed by a "mid-point" method at a position where the means of the branch lengths on either side of the root are equal. These trees are also used to derive a weight for each sequence. The weights are dependent upon the distance from the root of the tree but sequences, which have a common branch with other sequences, share the weight derived from the shared branch. In the example in figure 5.5, the leghaemoglobin (Lgb2_Luplu) gets a weight of

0.442, which is equal to the length of the branch from the root to it. The Human beta globin (Hbb_Human) gets a weight consisting of the length of the branch leading to it that is not shared with any other sequences (0.081) plus half the length of the branch shared with the horse beta globin (0.226/2) plus one quarter the length of the branch shared by all four haemoglobins (0.061/4) plus one fifth the branch shared between the haemoglobins and the myoglobin (0.015/5) plus one sixth the branch leading to all the vertebrate globins (0.062). This sums to a total of 0.221. The rooted tree with branch lengths and sequence weights for the 7 globins is given in figure 5.5.

3) Progressive alignment

The basic procedure at this stage is to use a series of pairwise alignments to align larger and larger groups of sequences, following the branching order in the guide tree. You proceed from the tips of the rooted tree towards the root. In order to calculate the score between a position from one sequence or alignment and one from another, the average of all the pairwise weight matrix scores from the amino acids in the two sets of sequences is used. If either set of sequences contains one or more gaps in one of the positions being considered, each gap versus a residue is scored as zero.

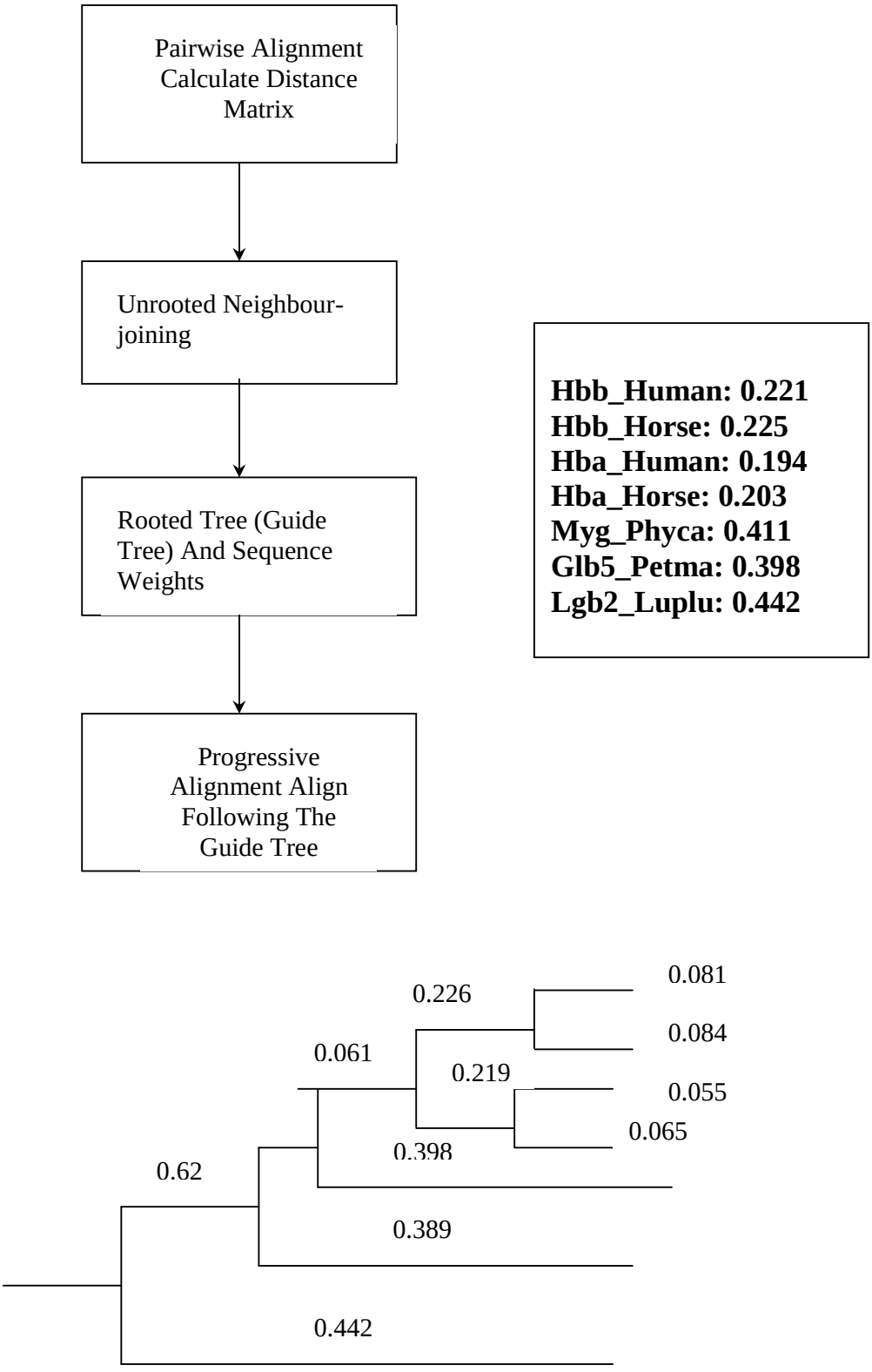


Figure 5.5 The Basic Alignment Procedure

5.4.3 FASTA Algorithm

FASTA compares a query string against a single text string. When searching the whole database for matches to a given query, we compare the query using the FASTA algorithm to every string in the database. The algorithm uses a fast search to initially identify sequences from the database with a high degree of similarity to the query sequence. Then it conducts a second comparison on the selected sequences. While FastA is actually just a fast approximation to the Smith-Waterman algorithm, it is slower and more sensitive than the BLAST algorithm because FastA tolerates gaps in the aligned sequences(16).

The stages in the FASTA algorithm are as follows:

In step 1) We specify an integer parameter called ktup (short for k respective tuples), and we look for ktup-length matching sub strings of the two strings. The matching ktup-length sub strings are referred to as *hot spots*.

In step 2) In this stage we wish to find the 10 best diagonal runs of hot spots in the matrix. A diagonal run is a sequence of nearby hot spots on the same diagonal. A run need not contain all the hot spots on its diagonal, and a diagonal may contain more than one of the 10 best runs we find.

In step 3) A diagonal run specifies a pair of aligned substrings. The alignment is composed of matches and mismatches. We next evaluate the runs using an amino acid (or nucleotide) substitution matrix, and pick the best scoring run. The single best sub alignment found in this stage is called $init_1$.

In last step) In the last stage, the database sequences are ranked according to $init_n$ scores or *opt* scores, and the full dynamic programming algorithm is used to align the query sequence against each of the highest-ranking result sequences.

These steps for the algorithm are described using the following figure.

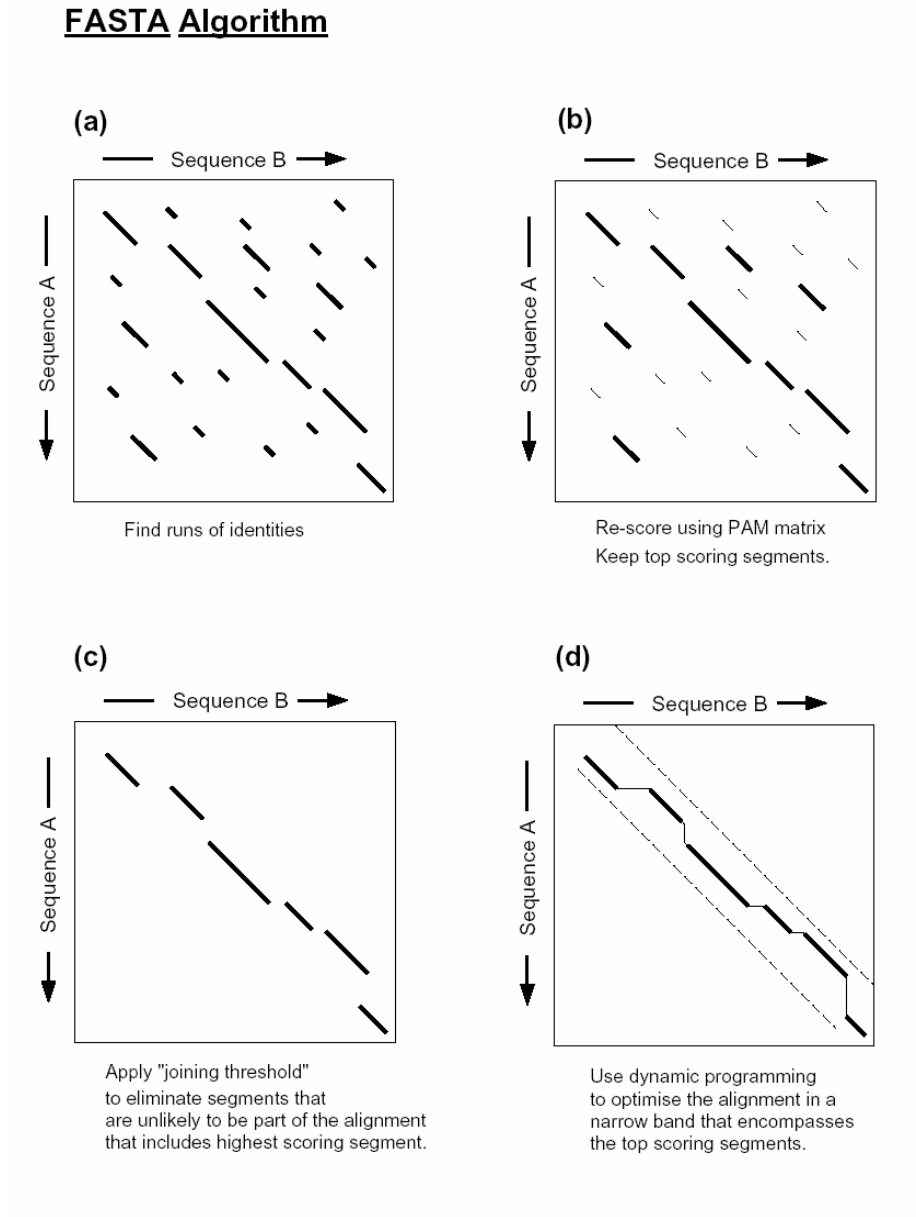


Figure 5.6 FASTA Algorithm

5.4.4 BLAT Algorithm

BLAT is bioinformatics software a tool and is BLAST LIKE ALIGNMENT TOOL, which performs rapid mRNA/DNA, and cross-species protein alignments. BLAT is more accurate and 500 times faster than popular existing tools for mRNA/DNA alignments and 50 times faster for protein alignments at sensitivity settings. BLAT is not BLAST. BLAT

will do in a day what BLAST will do in a month, and in many cases a single CPU will handle the load. The output is small and easy to parse. BLAT's speed stems from an index of all no overlapping K-mers in the genome.

The program rapidly scans for relatively short matches (hits), and extends these into high-scoring pairs (HSPs). However, BLAT differs from BLAST in some significant ways. Where BLAST builds an index of the query sequence and then scans linearly through the database, BLAT builds an index of the database and then scans linearly through the query sequence. Where BLAST triggers an extension when one or two hits occur in proximity to each other, Where BLAST returns each area of homology between two sequences as separate alignments, BLAT stitches them together into a larger alignment. BLAT has special code to handle introns in RNA/DNA alignments. Therefore, whereas BLAST delivers a list of exons sorted by exon size, with alignments extending slightly beyond the edge of each exon, BLAT effectively "unsplices" mRNA onto the genome—giving a single alignment that uses each base of the mRNA only once, and which correctly positions splice sites(11).

BLAT Algorithm has following stages:

All fast alignment programs break the alignment problem into two parts. Initially in a "search stage," the program detects regions of the two sequences, which are likely to be homologous. The program then in an "alignment stage" examines these regions in more detail and produces alignments for the regions

1) Search Stage

1.1 Searching With Single Perfect Matches

A simple and reasonably effective search stage is to look for subsequences of a certain size, k, which are shared by the query sequence and the database. In many practical implementations of this search, every K-mer in the query is compared against all no overlapping K-mers in the database.

1.2 Searching With Single Almost Perfect Matches

What if instead of requiring perfect matches with a K-mer to trigger an alignment, we allow almost perfect matches, that is, hits where one letter may mismatch? The probability that a no overlapping K-mer in a homologous region of the database matches almost perfectly the corresponding K-mer in the query.

1.3 Clumping Hits and Identifying Homologous Regions

To implement the match criteria, BLAT builds up an index of non-overlapping K-mers and their positions in the database. BLAT then looks up each overlapping K-mer of the query sequence in the index. In this way, BLAT builds a list of "hits" where the query and the target match. Each hit contains a database position and a query position. Hits that are within the gap limit are bundled together into proto-clumps. Hits within proto-clumps are then sorted along the database coordinate and put into real clumps if they are within the window limit on the database coordinate. Clumps with less than the minimum number of hits are discarded, and the rest are used to define regions of the database, which are homologous to the query sequence.

2) Alignment Stage

The alignment stage performs a detailed alignment between the query sequence and the homologous regions. For historical reasons, the alignment stage for nucleotide and protein alignments is quite different. Both have limitations, and are good candidates for future BLAT upgrades.

2.1 Nucleotide Alignments

The nucleotide alignment stage is based on a cDNA alignment program. The algorithm starts by generating a hit list between the query and the homologous region of the database. Because the homologous region is much smaller than the database as a whole, the algorithm looks for relatively small, perfect hits. If a K-mer in the query matches multiple K-mers in the region of homology, the K-mer is extended by one repeatedly until the match is unique or the K-mer exceeds a certain size. The hits are then extended as far as possible allowing no mismatches, and overlapping

hits are merged. If there are gaps in the alignment on both the query and database side, the algorithm recurses to fill in these gaps. This continues until either the recursion finds no additional hits, or the gap is five bases or less.

2.2 Protein Alignments

The protein alignment strategy is simpler. The hits from the search stage are kept and extended into maximally scoring ungapped alignments (HSPs) using a score function. A graph is built with HSPs as nodes. If HSP A starts before HSP B in both query and database coordinates, an edge is placed from A to B. The edge is weighted by the score of B minus a gap penalty based on the distance between A and B. In the case where A and B overlap, a "crossover" point is selected which maximizes the sum of the scores of A up to the crossover and B starting at the crossover, and the difference between the full scores and the scores just up to the crossover is subtracted from the edge score.

In step 3) stitches together these aligned regions (often exons) into larger alignments (typically genes). These alignments are stitched together using a minor variation of the algorithm used to stitch together protein HSPs.

In step 4) Finally, BLAT revisits small internal exons possibly missed at the first stage and adjusts large gap boundaries that have canonical splice sites where feasible.

5.5 PARAMETER BASED

Different tools work according to the different parameters. These parameters add to the performance of the algorithm. Parameters are the options that are selected for the more sensitive results. These are set according to the query and according to the requirements for the result. Various parameters are compared to know that which parameter is used in which tool and what is its function. What same parameters are used in the

different algorithms and what is the particular use of that parameter in that particular tool.

5.5.1 BLAST Parameters

BLAST uses various parameters and the importance of each one is disscussed below. The parameters are disscussed in the same order as they are used in the algorithm. There are different types of parameters which are used for example some parameters are used for the filtering and masking. Some parameters are for selectivity, translation. Some are used for the final score of the output. The results after the final alignment is complete are also formatted using various options for the report format(12).

1MAIN PARAMETERS

1.1 Blast program

The five BLAST programs described here perform the following tasks:

- Blastp compares an amino acid query sequence against a protein sequence database
- Blastn compares a nucleotide query sequence against a nucleotide sequence database
- Blastx compares the six-frame conceptual translation products of a nucleotide query sequence against a protein sequence database
- Tblastn compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames
- Tblastx compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.
- Psi-blastn compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands) using a position specific matrix created by PSI-BLAST.

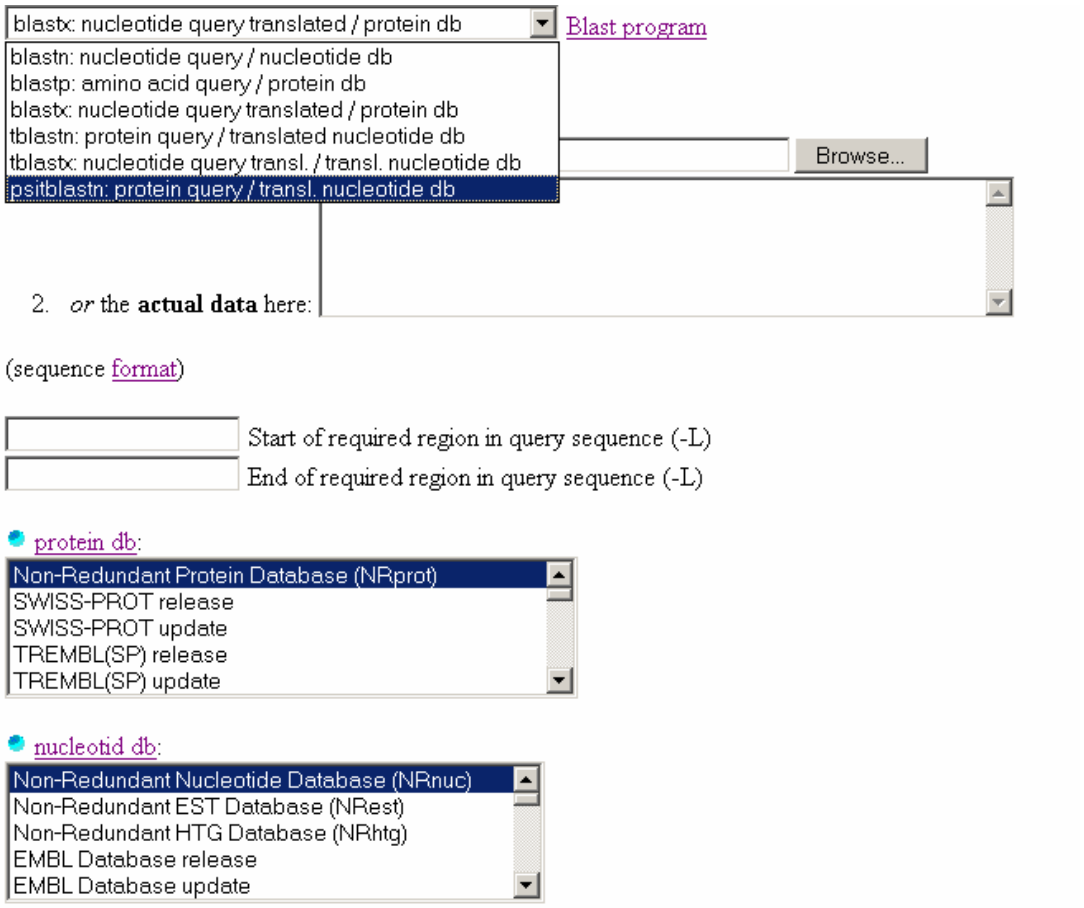


Figure 5.7 Main Parameters Of BLAST

1.2 Enter Either the Name Of A File Or The Actual Data

You can select a file by typing its name, or better, by selecting it with the Netscape file browser (**Browse** button) OR you can type your data in the next area, or cut and paste it from another application.

1.3 Protein Db

Choose a protein db for blastp or blastx.

Please note that SwissProt usage by and for commercial entities requires a license agreement.

1.4 Nucleotide Db

Choose a nucleotide db for blastn, tblastn or tblastx

2. FILTERING PARAMETERS

- The -F argument can take a string as input specifying that seg should be run with certain values or those other non-standard filters should be used.
- It is possible to specify that the masking should only be done during the process of building the initial words by starting the filtering command with 'm', e.g.: -F 'm S' which specifies that seg (with default arguments) should be used for masking, but that the masking should only be done when the words are being built.

2.1 Sequence format

The sequence will be automatically converted in the format needed for the program providing you enter a sequence either in plain (raw) sequence format or in one of the following: GenBank, NBRF, EMBL, GCG, DNA Strider.

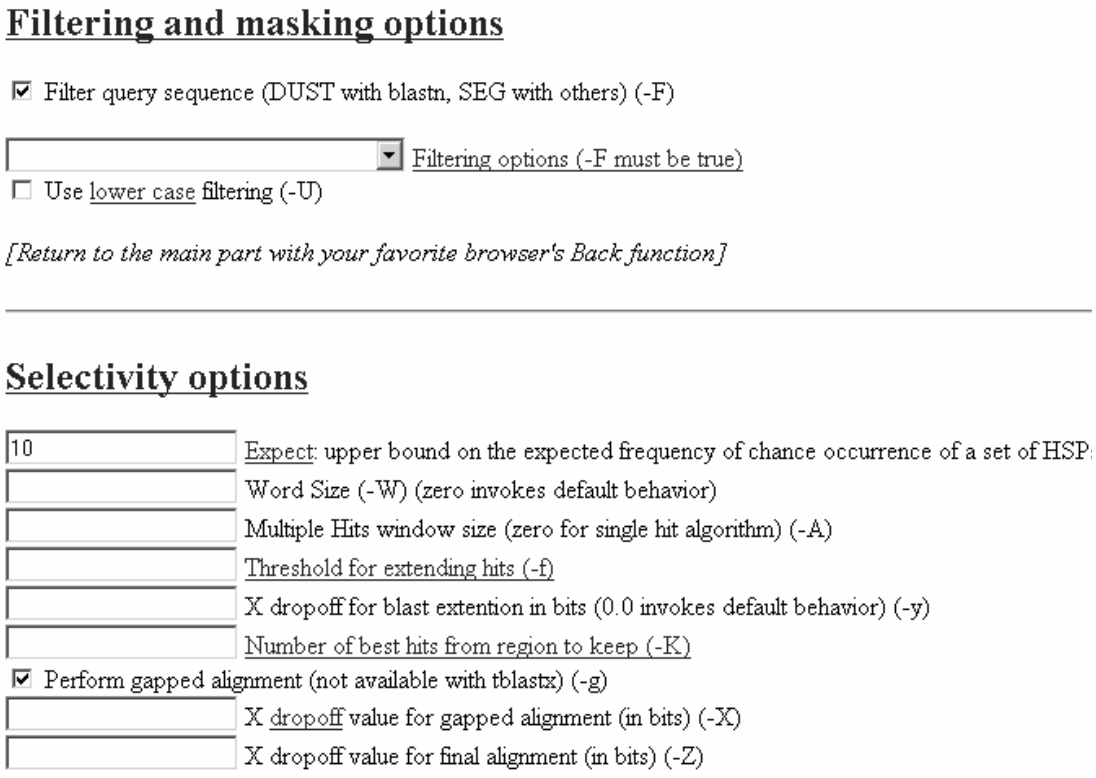


Figure 5.8 Filtering, Masking And Selectivity Parameters of BLAST

3. SELECTIVITY PARAMETERS

3.1 Expect Upper bound on the expected frequency of chance occurrence of a set of HSPs (-e)

The statistical significance threshold for reporting matches against database sequences; the default value is 10, such that 10 matches are expected to be found merely by chance, according to the stochastic model of Karlin and Altschul (1990).

3.2 Threshold for extending hits (-f)

Blast seeks first short word pairs whose aligned score reaches at least this value (default for blastp is 11)

3.3 Number of best hits from region to keep (-K)

If this option is used a value of 100 is recommended.

3.4 X drop-off value for gapped alignment (in bits) (-X)

This is the value that controls the path graph region explored by Blast during a gapped extension (default for blastp is 15).

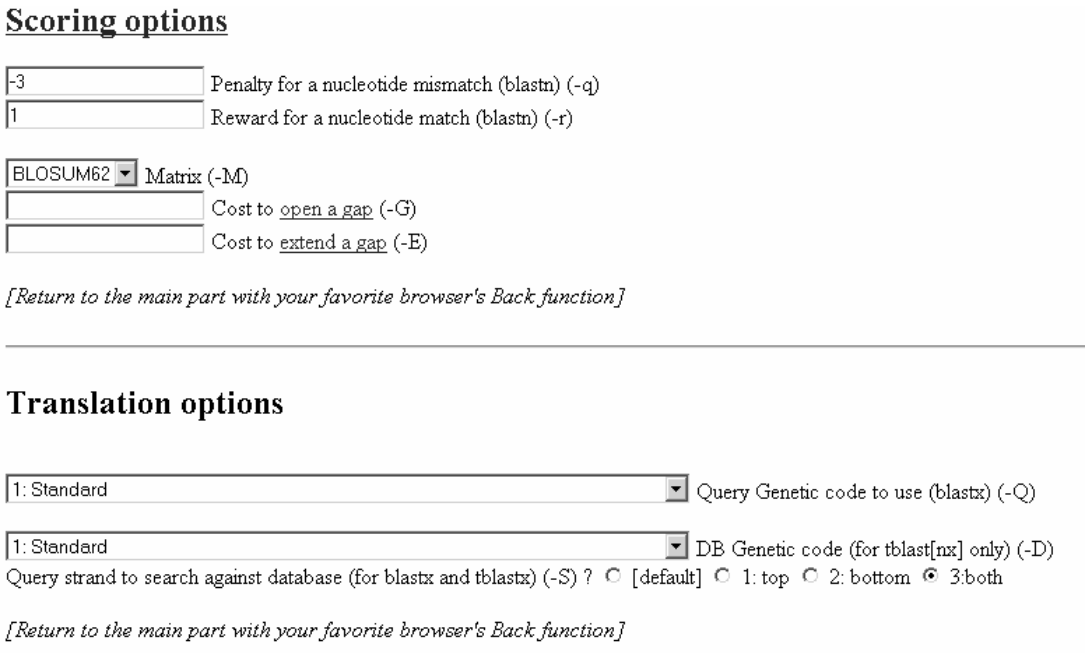


Figure 5.9 Scoring and Translation Parameters of BLAST

4. SCORING PARAMETERS

4.1 Cost to open a gap (-G)

Default is 5 for blastn, 10 for blastp, blastx and tblastn

4.2 Cost to extend a gap (-E)

Default is 2 for blastn, 1 for blastp, blastx and tblastn

Limited values for gap existence and extension are supported for these three programs

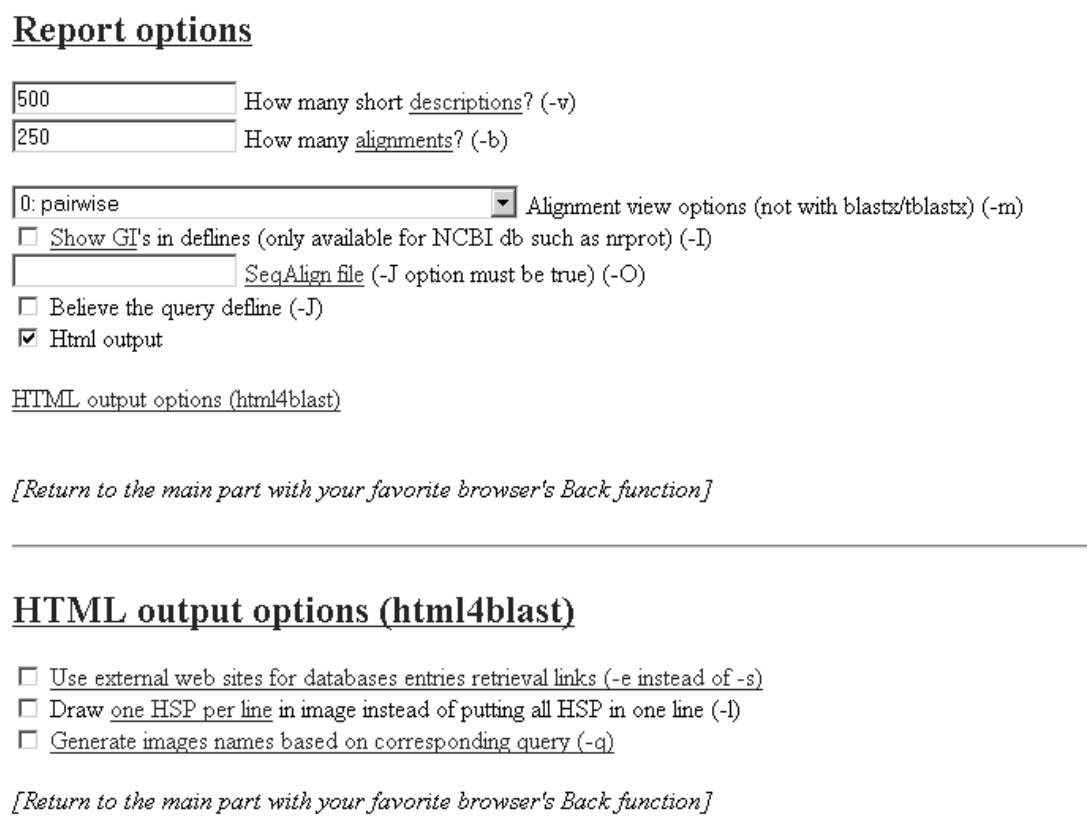


Figure 5.10 Report and Output Parameters Of BLAST

5. REPORT PARAMETERS

5.1 How many short descriptions? (-v)

Maximum number of database sequences for which one-line descriptions will be reported (-v).

5.2 Show GI's in defines (only available for NCBI db such as nrprot) (-I)

Causes NCBI gi identifiers to be shown in the output, in addition to the accession and/or locus name.

5.3 SeqAlign file (-J option must be true) (-O)

SeqAlign is in ASN.1 format, so that it can be read with NCBI tools (such as sequin). This allows one to view the results in different formats.

6. HTML OUTPUT OPTIONS (HTML4BLAST)

6.1 Use external web sites for databases entries retrieval links (-e instead of -s)

-s option will use SRS for databases entries retrieval links, whereas -e will use the original database site links.

6.2 Draw one HSP per line in image instead of putting all HSP in one line (-l)

Useful for genomes searching, where there is only one sequence in the database.

5.5.2 CLUSTERW Parameters

Parameters are the options that are selected for the more sensitive results. These are set according to the query and according to the requirements for the result. ClusterW uses various parameters and the importance of each one is disscussed below. The parameters are disscussed in the same order as they are used in the algorithm(15).

YOUR EMAIL

You must type your email address in this text box if you are running a job via email. It is not necessary to fill in the box if you are running your search interactively.

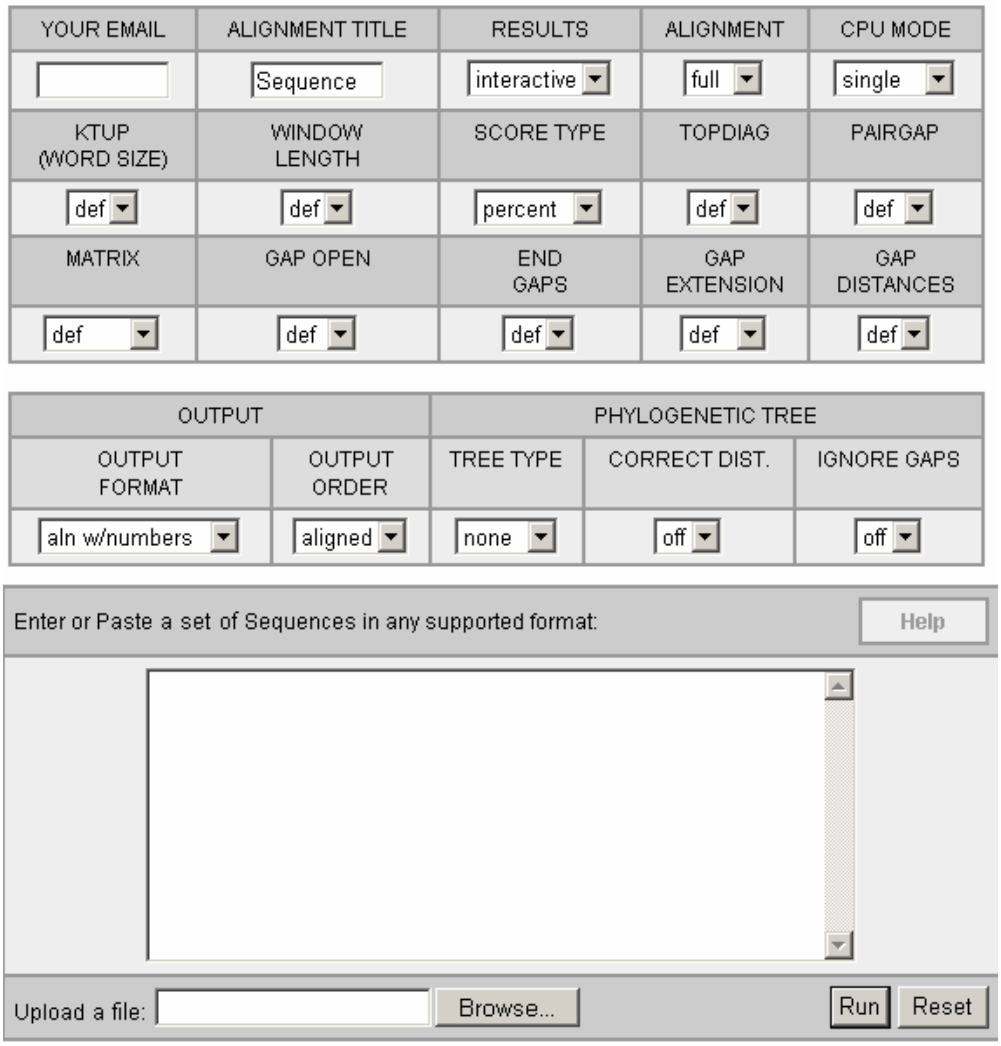


Figure 5.11 Parameters of ClusterW

ALIGNMENT TITLE

You may type any text you want to help you identify your search results.

RESULTS (EMAIL OR INTERACTIVE)

This option lets you choose between email and interactive runs. The email run requires you to type an email address in the email text box, and your results will be delivered when they are ready to your email address, thus avoiding waiting for your results as with an interactive run.The default is interactive.

ALIGNMENT (FULL OR FAST)

You may choose to run a full alignment or using a stringent algorithm for generating the tree guide or a fast algorithm.

CPU MODE (SINGAL OR MULTIPLE)

The multiple CPU option run a special version of ClustalW using several Linux pc nodes in a parallel fashion to increase the speed of the job without compromising the quality of the results. This option is to be chosen when the user has a large number of sequences (50+ but less than 500) to align. However, care should be taken not to overestimate the quantification of the results. A very large alignment is difficult to read and handle by other software.

KTUP (Def, 1, 2, 3, 4, 5)

This option allows you to choose which 'word-length' to use when calculating fast pairwise alignments.

WINDOW (Def, 0-10)

Use this option to set the window length when calculating fast pairwise alignments.

SCORE (PERCENTAGE OR ABSOLUTE)

This option allows you to decide which score to take into account when calculating a fast pairwise alignment.

PAIRGAP (Def, 1,2,3,4,5,10,25,50,100,250,500)

Select here to set the gap penalty when generating fast pairwise alignments.

MATRIX (Def, BLOSUM, PAM, GONNET, ID)

This option allows you to choose which matrix series to use when generating the multiple sequence alignment. The program goes through the chosen matrix series, spanning the full range of amino acid distances.

- **BLOSUM.** These matrices appear to be the best available for carrying out data base similarity searches. The matrix used is Blosum30.
- **PAM.** These have been extremely widely used since the late '70s. We use the PAM 350 matrix.
- **GONNET.** These matrices were derived using almost the same procedure as the BLOSUM but are much more up to date and are based on a far larger data set. They appear to be more sensitive than the Dayhoff series. We use the GONNET 250 matrix.
- **IDENTITY MATRIX.** We also supply an identity matrix which gives a score of 10 to two identical amino acids and a score of zero otherwise. Default values are:

  DNA: DNA Identity matrix and for Protein it is Gonnet 250.

GAPOPEN (Def, 1, 2, 5, 10, 25, 50,100)

You can set here the penalty for opening a gap. The default value is 10.

ENDGAP (Def, 10, 20)

You can set here the penalty for closing a gap.

PHYLOGENETIC TREE

Phylogram is a branching diagram (tree) assumed to be an estimate of a phylogeny; branch lengths are proportional to the amount of inferred evolutionary change.

UPLOAD A FILE

You may upload a file from your computer which containing a valid **set** of sequences in any format (GCG, FASTA, EMBL, GenBank, PIR, NBRF, Phylip or UniProt/Swiss-Prot) using this option.

5.5.3 FASTA Parameters

FASTA (pronounced FAST-Aye) stands for **FAST-A**ll, reflecting the fact that it can be used for a fast protein comparison or a fast nucleotide comparison. This program achieves a high level of sensitivity for similarity searching at high speed. The high speed of this program is achieved by using the observed pattern of word hits to identify potential matches before attempting the more time consuming optimised search. Parameters not only make the search fats but also make it more efficient(16).

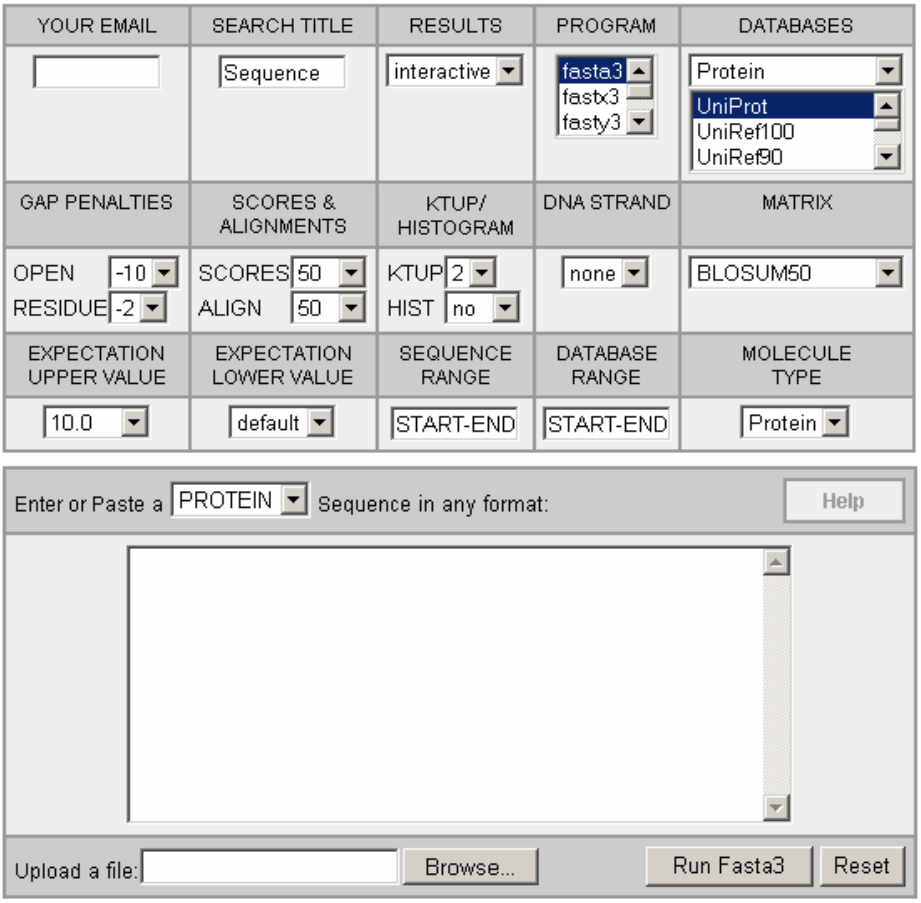


Figure 5.12 Parameters of FASTA

YOUR EMAIL

You must type your email address in this text box, it must be a valid internet email address. It is not necessary to fill in the box if you are

running your search interactively, where your results will be delivered to the browser window when they are ready.

SEARCH TITLE (INTERACTIVE OR EMAIL)

You may type any text you want to help you identify your search results.

RESULTS

This option lets you choose between email and interactive runs. The email run requires you to type an email address in the email text box, The default value is email. You will be delivered your results to your browser, when they become available with an interactive job.

PROGRAM

The programs available and their uses:

Program	Function	Submission Type
fasta3	scan a protein or DNA sequence library for similar sequences	interactive/email
fastx/y3	compare a DNA sequence to a protein sequence database, comparing the translated DNA sequence in forward and reverse frames.	interactive/email
tfastx/y3	compares a protein to a translated DNA data bank	interactive/email
fasts3	compares linked peptides to a protein databank	interactive/email
fastf3	compares mixed peptides to a protein databank	interactive/email

Table 5.1 Programs of FASTA

DATABASES

Choose here the databases you which to run your protein sequence against. You can choose multiple databases by clicking on them. The choices will

appear highlighted. To deselect a database simply click on it again. There is large number of databases available but some of them are listed in the table.

Abbreviation	Database Name
UniProt	UniProt is the central access point for extensive curated protein information, including function, classification, and cross-references. Search UniProt to retrieve "everything that is known" about a particular sequence.
UniRef	The UniRef databases combine closely related sequences into a single record to speed searches. There are three different non-redundant databases with different sequence identity cut-offs. In UniRef100, UniRef90 and UniRef50 databases no pair of sequences in the representative set has >100%, >90% or >50% mutual sequence identity. The three UniRef databases allow the user to choose between a fast search and a truly comprehensive one.
UniParc	UniParc contains available protein sequences collected from many different sources. The sequence data are archived to facilitate examination of changes to sequence data. Search UniParc if you want to examine the "history" of a particular sequence.
UniProt/Swiss-Prot	UniProt/Swiss-Prot Protein Database
prints	FingerPrints
IPI	International Protein Index
EURO patents	European patents database.
JAP patents	Japanese patents database.
USPTO patents	American patents database.
SGT	Structural Genomic Targets Database
PDB	Protein Database of Brookhaven

Table 5.2 Databases Available With FASTA

ALIGNMENTS (10, 20, 30, 40, 50, 60, 70, 80, 90,100)

Setting these options to any number available in the menu allows you to set the maximum number of reported alignments in the output file. Note that matching sequences are connected with a " |" symbol. Mismatches

would be connected with a space. A gap would be represented with a " -" symbol.

KTUP (1, 2)

Change this value to limit the word-length the search should use. A word-length of 2 is sensitive enough for most protein database searches. The thumb rule is that the larger the word-length the less sensitive, but faster the search will be.

STRAND

This option lets you choose which DNA strand to search with when you are using a DNA sequence to compare against the DNA databanks. 'Top' means the sequence will be searched, as it is input into the form. 'Bottom' means: reverse and complement your input sequence.

SEQUENCE RANGE

This option allows the user to denote which region within the query sequence should be searched. The default is to search using the whole query sequence.

DATABASE SEQUENCE RANGE TO SEARCH

This option is similar to the above except that it sets the sequence range to search within the database. The default is to search against the whole database entry

UPLOAD A FILE

You may upload a file from your computer which containing a valid sequence in any format (GCG, FASTA, EMBL, GenBank, PIR, NBRF, Phylip or UniProt/Swiss-Prot) using this option.

5.5.4 BLAT Parameters

BLAT on DNA is designed to quickly find sequences of 95% and greater similarity of length 40 bases or more. It may miss more divergent or

shorter sequence alignments. In practice DNA BLAT works well on primates, and protein blat on land vertebrates(11).

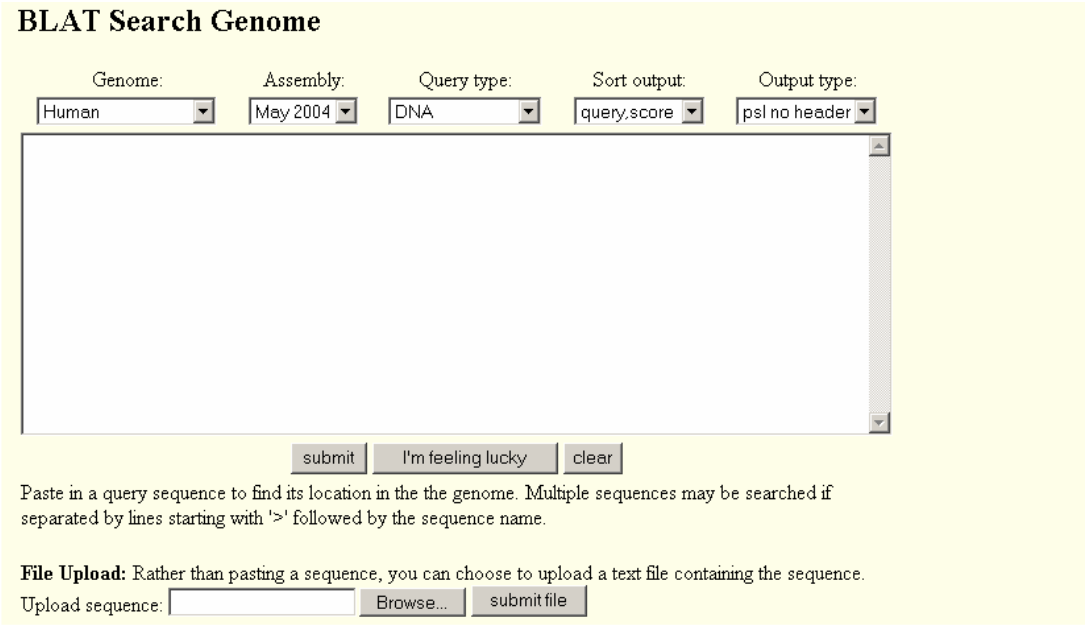


Figure 5.13 Parameters Of BLAT

GENERAL OPTIONS

t=type Database type.

Type is one of:

- dna - DNA sequence
- prot - protein sequence
- dnax - DNA sequence translated in six frames to protein

The default is dna

The following files are available:

- Human Genome, May 2004 build in /fdb/genome/human-aug2003
- Mouse Genome, June 2004 build in /fdb/genome/mouse-oct2003
- Other nucleotide databases in /fdb/fastadb, updated weekly:

-tileSize=N

Sets the size of match that triggers an alignment.

Usually between 8 and 12

Default is 11 for DNA and 5 for protein.

-minMatch=N

Sets the number of tile matches. Usually set from 2 to 4
Default is 2 for nucleotide, 1 for protein.

-minScore=N

sets minimum score. This is the matches minus the
mismatches minus some sort of gap penalty. Default is 30

5.6 SUMMARY OF COMPARISION

Each tool has some best features in it, which can be used in prior to the other. Brief summary of which tool is used under which condition is given here. Further advantages of one tool over the other are also discussed.

5.6.1 BLAT Is Preferred For

From a practical standpoint, Blat has several advantages:

- Speed (no queues, response in seconds) at the price of lesser homology depth
- The ability to submit a long list of simultaneous queries in fasta format
- Five convenient output sort options
- A direct link into the UCSC browser
- Alignment block details in natural genomic order

5.6.2 BLAST Is Preferred For

- The BLAST program is preferred it executes much faster than FASTA. A typical BLAST search done locally will execute in less than a minute, whereas a local FASTA search will take about 30 to 60 minutes or more.
- The BLAST program is usually more sensitive than the FASTA program for detecting protein sequence similarity when both programs are used with their default parameters because it does not require a perfect match in the first stage of the search.

- The BLAST program can directly translate a nucleotide sequence into six frames and search a protein database. This would require six separate searches with FASTA.

5.6.3 FASTA Is Preferred For

- The long word size in a BLAST DNA sequence similarity search allows the program to execute extremely fast, but the price of speed is a loss in sensitivity. The FASTA program will show some weak DNA hits that will not be found in your BLAST report.

- It is preferred for availability of various variants of FASTA under different situations.

- An advantage of FASTA is that it provides relatively timely results while missing less homolog than BLAST. Due to its smaller word size when searching DNA, it is also better than BLAST at detecting longer regions of lesser homology. FASTA is a good choice for running DNA queries, and overall it performs well for stringent comparisons where accuracy concerns are moderately important.

5.6.4 CLUSTERW Is Preferred For

- It shows good performance on protein and it has high quality in stringent comparison where accuracy is essential.
- It is dynamic algorithm used for multiple sequence alignment.
- It is computationally intensive and rigorous and takes longer time to compute than the heuristic, but the sensitivity of results is more.
- It is preferred when we have to identify short identical sequences.
- It compresses information even further while giving the final results by not only removing comments, but also transforming the plain text of sequences into a compressed, more efficient bit code that can be read faster.

CHAPTER 6
CONCLUSION

Biosequences searching poses a particular challenge to the bioinformatics. The basis for measuring sequence similarity is rooted in evolutionary dogma. DNA holds the information of a protein's sequence, structure, folding, and ultimately, its function. Changes to a unique nucleotide or amino acid sequence, however, do not necessarily render the protein inactive. In fact, several conservative substitutions may occur in a given sequence without disrupting protein function. In light of this, a method is needed to determine just how similar two sequences really are when such variations exist. Generally, sequence searchers assume the tool provided by a database is sufficient for all searching purposes. This tool, known as an algorithm, determines sequence similarity. A problem encountered by searchers is that databases often contain limitations as to which algorithms may be used to conduct the analysis.

Algorithms are the programs that run the sequence analysis; they perform the crunching that yields the best alignments and, thus, the final results. Most biosequences algorithms create local alignments (analyzing part of a sequence) or global alignments (analyzing over the whole sequence) and are typically characterized as heuristic or dynamic. There is vast variety of tools available that can be used and In this regard, there are often user-defined criteria to be set before a sequence is queried: picking the scoring matrix, setting the gap penalties, adjusting e-scores, filtering low-complexity regions, and determining word size, to name a few. Though algorithms are indispensable tools for determining homologies, they are not without their flaws. Heuristic programs, for example, can produce variable results depending on how the search parameters are set. Further, inconclusive results may be obtained if the default parameters are not adjusted prior to a search.

I had compared four sequence analysis tools. The summary of the results obtained is below. I have used two criteria for comparison one is algorithm

based and the other one is parameter based. Here I am concluding the comparison of both criteria

- **Conclusion of Comparison of Tools Based On Algorithm**
- **Conclusion of Comparison of Tools Based On Parameters**

A COMPARISON OF FOUR MAJOR BIOSEQUENCE ALGORITHMS				
	CLUSTERW	FASTA	BLAST	BLAT
Introduced	1990s	1980s	1990s	2000
Type	Dynamic, Local	Heuristic, Local	Heuristic, Local	Dynamic, Best-fit
Works best for	Protein	Protein	Protein, DNA	DNA and Protein
Ideal sequence length	Very short to long	Average length	Average length	Very short to long
Speed	Slower than BLAST	medium	very fast	Generally slower than CLUSTERW
Reliability for homology opinions	Highly reliable	Moderately reliable	Least to moderately reliable	Highly reliable
Default word size for DNA, Protein	1,1	6,2	11,3	11,3
Comment	USPTO “Gold standard” for alignments	Good for general searches or finding sequences of similar homology	Speedy; good for general searches or finding sequences of high homology	Most promising new algorithm and arguably most powerful algorithm

Table 6.1 Comparison Based On Algorithm

Summary of Comparison Based on Parameters

BLAST	CLUSTERW	FASTA	BLAT
Variety of programs is available: BlastN, BlastP, BlastX, TblastN, TblastX.	No variants are present. Only different versions are there.	Variety of programs available: FastA, FASTX, FASTY, TFASTA, Fasta3, Fastx/y3, Tfastx/y3, Fasts3.	Mainly three programs are available: A stand-alone program called 'blat', GfServer, GfClient.
Filtering and masking options are present in order to have more efficient results.	No filtering and masking options are present.	No filtering and masking options are present	No filtering and masking options are present.
Database selection type: Protein and nucleotide databases are selected	No database selection option is present.	Database selection options allows to select from protein databases	Protein and Nucleotide databases are selected specified by: DNA, Prot, DNAX (DNA sequences translated to six frames to protein).
Input sequences can be nucleotide or proteins.	Only Protein and DNA sequences are entered.	Input sequences can be DNA, Protein, DNAX	Input query can be DNA, RNA, dnax, RNAX, Protein
Sequence format can be one from the following: Plain text, GenBank, EMBL, NBRF, GCG, DNA strinder, Fitch, Fasta.	Sequence format can be one from the following: ALN, GCG, Phylip, PIR, GDE.	Sequence format can be one from the following: GCG, FASTA, EMBL, PIR, NBRF, Phylip, UniProt, Swiss-Prot.	Sequence format can be one from the following: Psl, Pslx, Axt, Maf, Sim4, Wublast, Blast.

BLAST	CLUSTERW	FASTA	BLAT
Selectivity options are: Expect (e), Word size (-w), Threshold for Extending hits (-f), X drop-off value for gapped alignment (x), X drop-off value for final alignment (in bits)(-t).	Selectivity options are: Alignment (full or fast), CPU Mode (Single or Multiple), Window size, TOPDIAG, PAIRGAP, GAPOPEN, ENDGAP, GAPNEXT, GAPDIST.	Selectivity options are: Gap Penalties (open or residue), Expectation upper value, Expectation lower value Seque0nce range, Database range to be searched.	Selectivity options are: Word size, X Drop-off value for gapped alignment (X), X Drop-off value for final alignment (in bits), -Title size=N Sets the size of match that trigger an alignment.
Results are obtained directly from NCBI site and kept using specific ID For a particular sequence for 24 hours.	Results can be either obtained using email or can be taken directly using interactive mode.	Results can be either obtained using email or can be taken directly	Results can be taken from UCBC website that hosts BLAT tool.
Scoring options are: Penalty for a nucleotide mismatch, Reward for a nucleotide match, Cost to open a gap, Cost to extend the gap.	Scoring options are: Score (Percentage or Absolute), Alignment score (full or fast), Sequence No, Sequence Name, Sequence length.	Scoring options are: Scores(set to max no. Of reported scores), Alignment (set to max no. Of reported alignment).	Scoring options are: Cost to extend a gap, Cost to open a gap, -minscore=N sets the min score [(Matches-mismatches)-some sort of gap penalty].
No parallel CPU is used.	CPU Mode= single or multiple. Allows increasing speed without compromising the quality.	No parallel CPU are used	No parallel CPU are used
No phylogenetic tree is generated	Phylogenetic tree is generated.	Phylogenetic tree is generated.	No phylogenetic tree is generated.

BLAST	CLUSTERW	FASTA	BLAT
Direction of search is described for the DNA strand. Whether to search from top or in the reverse using DNA strand option.	No direction specification option is available. No DNA strand option.	Direction of search is described for the DNA strand. Whether to search from top or in the reverse using DNA strand option.	No direction specification option is available. No DNA strand option.
There is no limit on the range of databases to be searched.	There is no range specification for DNA query range and database range.	There is parameter specifying the range to be searched for databases and for the input query.	There is no range specification for DNA query range and database range.
Word size is changed to make search fast, but search can be made less sensitive in this manner.	KTUP is used to choose word length that is used for fast pairwise alignments.	KTUP is used to limit the word length the search should use word length of 2 for protein databases for DNA databases it is 6.	Word size parameter is used instead of KTUP.
MATRIX parameter is used to define which matrix series to use. Options available are: BLOSUM 80 BLOSUM 62 BLOSUM 45 PAM 30 PAM 70	MATRIX parameter is used to define which matrix series to use when generating multiple alignments. BLOSUM PAM GONNET Identity Matrix is used which gives score of 10 to two identical amino acids and of 0 otherwise.	MATRIX parameter is used to define which matrix comparison matrix should be used when searching the database. The default matrix is BLOSUM 62. Theses matrices cover various evolutionary constraints.	MATRIX parameter is used to define which matrix series to use. Options available are: BLOSUM 80 BLOSUM 62 BLOSUM 45 PAM 30 PAM 70

BLAST	CLUSTERW	FASTA	BLAT
There is no molecule type specified in BLAST. But DNA and Proteins are used for molecule type.	There is no molecule type specified in CLUSTERW. But only Proteins are used for molecule type.	Molecule specifier is here and available options for this is Prot DNA DNAX MRNAX MRNA TRNA But mainly proteins and nucleotide molecules are used.	There is no molecule type specified in BLAT. But DNA, Proteins and dnax are used for molecule type
Report and format options are available for the final formatting of the result report.	No such options are used.	No such options are used.	Report and format options are available for the final formatting of the result report.

Table 6.2 Comparison Based On Parameter

It is evident that searches conducted here often vary from those conducted by investigators. The goal of the search should indicate the choice of the algorithm used, and careful measure should be used in determining the optimal algorithm. The aim of the algorithm used should be not to let sequences of significant homology fall through the cracks. Since different results are obtained with different algorithms, it is important to have a healthy distrust of what the results indicate about the query sequence homology. If the initial results do not meet expectations, particularly with one algorithm, it is advisable to repeat the search with a different algorithm or tune the user-defined parameters. Finally, it is important to realize a choice algorithm is only part of the requirements in providing a definitive answer as to the true novelty of a sequence. Without comprehensive and consistent updating of the database being searched, valuable art will be missed no matter which algorithm is employed.

REFERENCES

1) Jiawei Han, Micheline Kamber and Simon Fraser University "***Data Mining Concepts and Techniques***" Morgan Kaufmann Publishers, USA 2001.

2) U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy "***Advances in Knowledge Discovery and Data Mining***" AAAI/MIT Press, 1996.

3) J. Han and M. Kamber "***Data Mining: Concepts and Techniques***". Morgan Kaufmann, 2000.

4) G. Piatetsky-Shapiro, U. Fayyad, and P. Smith "***Data mining to knowledge discovery: An overview***". In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining, 1-35. AAAI/MIT Press, 1996.

5) G. Piatetsky-Shapiro and W. J. "***Frawley Knowledge Discovery in Databases."*** AAAI/MIT Press, 1991.

6) Jagadish et al., ***"Special Issue on Data Reduction Techniques".*** Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997

7) Jean Michel Claverie and Cedric Notredame "***Bioinformatics A beginner's Guide"***. Wiley Publishing, Inc. 2003.

8) Jason, Bruce, Dennis, ***"Pattern Discovery in Bimolecular Data",*** Oxford University Press, New York 1999.

9) Dan E. Krane, Michael L. Raymer ***" Fundamental concepts of Bioinformatics"*** Pearson Education, 2003.

10) Scott Markel, Darryl Leon ***" Sequence Analysis In a Nutshell"*** O'reilly 2003.

11) Kent, W. James ***"BLAT- The BLAST- like Alignment Tool"*** Genome Research 12 (4) 656-664" 2002.

12) Stephen F. Altschul, warner Gish, Webb Miller ***" Basic Local Alignment Tool"*** Mol Biol 215 , 403-410, 1990.

13) Stephen F. Altschul, Thomas L. Madden, Alejandro A. Shäffer 1, and David

14) Lipman ***"Gapped BLAST and PSI-BLAST: a new generation of protein database search programs"*** 3389–3402 Nucleic Acids Research, Vol. 25, No. 17, 1997.

15) Julie D.Thompson,Desmond G.Higgins+ and Toby J.Gibson* ***"CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice"*** European Molecular Biology Laboratory, Postfach 102209, Meyerhofstrasse 1, D-69012 Heidelberg, Germany

16) William R. Pearson ***"FASTA3 program package"*** Department of Biochemistry, University of Virginia, Charlottesville, VA 22908, August 28, 1998

17) David R. Powell, David L. Dowe, Lloyd Allison and Trevor I. Dix ***"Discovering simple DNA sequences by compression"*** Department of Computer Science, Monash University, Clayton, Vic. 3168, Australia

18) William R. Pearson* And David J. Lipman ***"Improved tools for biological sequence comparison"*** Department of Biochemistry, University of Virginia, Charlottesville, VA 22908; and tMathematical Research Branch, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, MD 20892

19) Stephen F. Altschul*, Thomas L. Madden, Alejandro A. Schäffer1, Jinghui Zhang, Zheng Zhang2, Webb Miller2 and David J. Lipman ***"Gapped BLAST and PSI-BLAST: a new generation of protein database search programs"*** National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA

20) Gilbert, D. G. ***"ReadSeq version 2, an improved biosequence conversion tool "*** Bionet.Software(Aug), 1999.

21) By blast-help group, NCBI User Service, ***"BLAST Program Selection Guide",*** NCBI, NLM, NIH, 8600 Rockville Pike, Bethesda, MD 20894

22) Dan E. Krane, Michel L. Raymer ***"Fundamentals concepts of Bioinformatics",*** Pearson Education, 2003.

23) Dr. Joanne Fox, ***"Sequence Similarity Searching: Understanding and Using Web Based BLAST"***, Wednesday January 26th, 2005 Rm 220 FNS Building, UBC

LIST OF PUBLICATIONS

1. Navjot Kaur, Harpreet Kaur, Ms Rinkle Aggarwal **"Algorithmic And Non Algorithmic Issues In Database Search Of Sequence databases"**, National Conference On Bioinformatics Computing, NCBC'05. Thapar Institute of Engineering and Technology, Patiala.
2. Navjot Kaur, Harpreet Kaur, Ms Inderveer chana **"Issues Of Software Engineering And Knowledge Engineering In Bioinformatics",** National Conference On Bioinformatics Computing, NCBC'05. Thapar Institute of Engineering and Technology, Patiala.