Lisa Mullan
lisa@ebi.ac.uk

David Judge
dpj10@mole.bio.cam.ac.uk

Bioinformatics – A User's Approach

Table of Contents

Course Objective

Our aim is to present a hands-on approach to bioinformatics utility software for computing novices. We will provide an overview of available software, discuss some of the ideas behind the approaches and tackle some common tasks. The course has many practical examples and these try to follow typical mini projects so that the relevance is apparent.

WHY SHOULD I CARE ABOUT BIOINFORMATICS?

These days computers are part of our every day lives. Everywhere you go they are used to keep track of data, process transactions and support communication. The Human Genome Project (and other genome sequencing projects) is producing data at an astounding rate; currently sequence entries are being added to the EMBL sequence database faster than you can read their one-line descriptions - on average, one new sequence per second.

The task of processing all this data and converting sequences into gene and protein predictions would be impossible without the development of computer based analysis tools. The last few years have seen a rapid acceleration in the development of new bioinformatics approaches, and a dramatic increase in the number of researchers involved in this field.

So why do you need to learn bioinformatics? At the very least, you will need to identify whether your gene of interest has already been sequenced, and whether there are related sequences. If you aren't working on humans, or one of the other organisms that is being sequenced, there will be fewer pre-prepared resources available to you, although this discrepancy is narrowing, as more and more Genome projects are completed.

There will be a mixture of talks and frequent practical sessions, during which we will move amongst you helping you when you get stuck. Please ask questions. There will inevitably be a mixture of abilities in this course. If you find that we are going too fast or not making ourselves clear, feel free to interrupt us! If you don't understand, then it is probable that half the other people here don't understand either and will be grateful to you. The only way we can improve this course is for you to tell us that we are not being clear. Shout out if you need help during the practicals. There will be a questionnaire to fill out at the end of the course to give us feedback on what you thought of it.

Practical exercises are presented using a set of three symbols.  indicates a practical involving resources on the World Wide Web.  indicates analysis, and indicates where you need to write something down for future reference.

Further Reading

Bioinformatics is a rapidly expanding area. Much of the literature and methods are in the form of papers or just simply on the WWW. The following are good starting points:

Introduction to Bioinformatics (Addison Wesley Longman 1999); T.K. Attwood and D.J. Parry-Smith

Introduction to Bioinformatics (OUP 2002); Arthur M. Lesk

Bioinformatics (Cold Spring Harbor Laboratory Press 2004) David W. Mount

Bioinformatics for Dummies (Wiley 2003) Jean-Michel Claverie, Cedric Notredame

BLAST (O'Reilly, 2003) Ian Korf, Mark Yandell, Joseph Bedell

Internet Data Resources

In order to simulate a situation that is hopefully not too different from the ones you will encounter, the entire course is based on a case study devoted to one gene. It is involved in the eye disease **aniridia**.

Non-sequence Databases

For a subject (such as aniridia) you know little or nothing about, the best plan is to find some information on it! **GeneCards** is an integrated database, which assimilates biochemical information from a wide range of resources and is a good place to start. It is automatically curated at the Weizmann Institute in Israel and offers concise information on human genes and their functions. The data it provides is free to academics, although use by industrial employees requires a licence.

Go to http://www.genecards.org/index.shtml [1] and search the site using the keyword[2] **aniridia**. You are rewarded with a screen offering you the summary of the contents of several GeneCards followed by mini-cards underneath. These mini-cards are sorted by relevance, so select the one that is most relevant to your search (and the only one with "aniridia" in the summary). To display the complete Card, follow the link in the blue banner above this summary.

There is a wealth of information available from this GeneCard. All names are used according to the HUGO Gene Nomenclature Committee[3] and the correct name of the gene involved in aniridia is given as the name of the GeneCard, with synonyms used by various databases also listed. The Card is divided up into two vertical columns. The left hand column displays a definition of the information held in the right hand column. The links in this left hand column will take you to the homepage of any of the databases listed. In contrast, the links in the right hand column take you to the database entry for the specified gene.

Words that are identical to those specified in your original search are marked in red.

It is worth noting down certain information from this and subsequent databases as it will help to constrain other searches you may do.

[1] Although this site is held in Israel, there are several mirror sites around the world. Because internet traffic can be slow, it is generally a good idea to use sites as close to home as possible. Unfortunately, some mirror sites are not always up-to-date.

[2] Ensure that the "Search/Display GeneCards™ by" is set to keyword.

[3] URL: http://www.gene.ucl.ac.uk/nomenclature/

Gene responsible for aniridia ……*

Cytogenetic location ……

Number of UniProt isoforms ……

Each sequence is given a unique accession number when it is deposited into a database. This number should remain with it permanently and enable it to be found rapidly as the result of a keyword search. These **accession numbers** are unique to the entry and the database. It is unlikely that the same entry will have the same accession number in a difference database.

Scroll down until you reach the sequences section. Check out the sequences listed under "Additional Gene/cDNA sequence". Why are there several listed?

GeneCards is not the only integrated database. There are many others with links to entries relating to your keyword search. For example, GDB, Ensembl and OMIM. You can also access both these databases from the GeneCard.

Ensembl Accession Number ……

The **Genome Database (GDB)** started life as the central repository for human mapping and genomic data and is currently hosted at RTI International (www.rti.org) who will run it as a free, public resource. The term "Genomic Segment" is used to define a region of the genome, which may be anything from a cytogenetic marker, to a complete clone. The entries are owned by the individuals who deposited them, and these owners maintain sole editing rights. The GDB does not store sequence or raw mapping data.

Link to the Pax6 entry in GDB from your Genecard.

GDB also offers plenty of information of the names and cytogenic localisation of the Pax6 gene. It also displays information of the experimental determination of genome localisation. Further down there are links to mutations, phenotypes and homology.

Follow the link "PAX" under the section header "Families" and note down the consensus pattern of the paired box in the "definition" section.

*You should have found out that the gene implicated in aniridia is pax6 - a paired box homeotic gene.

Paired box pattern .....

Pattern span (residue to residue) .....

The information to be found includes the identity of other members of the family and what their function is.

Number of PAX genes .....

Possible gene function .....

OMIM is the **Online** version of Victor McKusick's **Mendelian Inheritance in Man**. It is a database of phenotypes of human diseases - with a substantial genetic component.

Follow the link to this database from the Pax6 GeneCard. Read some of the OMIM entry. In particular have a look at some of the database links. Are there any that might be useful to you in the future?

If you were really going to enter this field, you may wish to subscribe to one or more "alerting services". These will automatically send you an email when, for example, relevant papers are published or nucleotide or peptide sequences are released. **PubCrawler** is an alerting service that scans daily updates to the PubMed and Genbank[4] databases and can be found at: http://www.pubcrawler.ie/. You need to register to use this service, but registration and the subsequent service is free. You might want to check out their sample results page.

Clinical Databases

If you are studying a disease, the chances are that the cause of this is an underlying genetic mutation. The **Human Gene Mutation Database (HGMD)** is held at the Institute of Medical Genetics in Cardiff, Wales and contains sequences and phenotypes of human disease-causing mutations. The database can be searched in a variety of ways including disease name, gene name or symbol and also by OMIM or GDB accession number.

[4] Genbank is the American repository for nucleotide sequences and contains the same data as EMBL – the European databank.

Go to http://www.hgmd.org and choose “HGMD Search”. Type **pax6** into the keyword field and submit the search. The results of your search are at the bottom of the page where you will see a link to “PAX6”. Follow this link to the index of mutations where the entries are grouped in tables according to mutation type or phenotype.

The data below the links from the phenotype groupings link directly to several databases with entries for PAX6.

The HGMD only records the first literature report of a mutation, so it is not impossible that reported mutations may also cause other diseases.

Follow the “Get mutations” link for the **Splicing** mutations. Check there are the same number of mutations as specified in the original table. Do all mutations cause aniridia? The reference attributed to that mutation is on the right side of the table. Follow one of these reference links – they display the abstract of that particular reference from the PubMed entry.

Each splice mutation is given a unique accession number; IVS[5] and whether it is a donor or an acceptor splice site[6]. The location of the substitution is then relative to the relevant splice site of the defined intron. The substitution is then displayed together with the phenotype and the number of references.

Substitution (accession number CS982309)……..

Location ……..

One of the features to be observed during the sequencing of the Human Genome, was the presence of individual base-pair mutations, or Single Nucleotide Polymorphisms (SNP). There are several SNP databases held around the world, possibly the most comprehensive of these is **dbSNP** held at the NCBI in the USA. Searching these databases, however, is not the easiest of tasks if you do not have the SNP identification number. There are several ways of obtaining this, and we will look at a couple in the next section using sequence database searches.

Sequence Databases

There is one main nucleic acid sequence database and one main protein sequence

[5] IVS – Intervening sequence – i.e. introns.

[6] Donor splice sites are located on the boundary between the right end of an exon and the left end of an intron. Acceptor splice sites are located on the opposite boundary. The mutation is given relative to the ds or as of a specified intervening sequence.

database in widespread general use among the European biological community. The nucleotide database is EMBL[7] and the protein database is UniProt[8].

As these databases contain hundreds of thousands of sequences, searching through them requires the processing power of a computer search engine. The **Sequence Retrieval System (SRS)** has been designed to do just that. SRS is available at many sites over the world. However, every site allows access to a different set of databases and, sometimes, search and analysis tools.

Go to SRS at the EBI at http://srs.ebi.ac.uk. Click on the "Library Page" link and you should be faced with a page offering you several databases[9] under a "Nucleotide Sequence Database[10]" heading. Select the EMBL[11] database by clicking in the little box to the left of it and then select the **standard** query form on the left hand side of the page.

In the first editor field, enter **pax6** and leave the query as "All Text". In the second field, enter **human** and alter the query to "Organism Name" using the scroll menu. At the bottom of your page, under the "Create your own view" section, highlight "AccNumber". Now press the Control key on your keyboard and also highlight "Description".

Ensure that the "Use Wildcards" option on the left hand side of the page is turned off (there should be no tick in the box) and click on the yellow "Search" button.

Scroll down the list of hits looking for a genomic sequence. Why would the rest of these be unsuitable? (Hint the list goes over two pages and you are looking for "clone" in the description line)

You should have received a list of approximately 37[12] sequences. The left hand column contains the sequence identifiers. An identifier consists of the name of the databases housing the sequence, followed by a colon and some alphanumeric characters defining the sequence. These identifiers are unique to the sequence and database. The subsequent columns are the results of your choice on the standard query form – in this case accession number and description. Accession numbers are unique to the sequence, and should not change between database releases. The final column contains the requested description of the sequence and is perhaps a better initial indication of what you are looking at.

You should also have noticed that one clone mentioned in GDB, with accession number

[7] Nucleic acid sequences may be submitted to any of the three databases, depending on whether you are resident in Europe (EMBL), America (Genbank) or Asia (DDBJ – DNA DataBase of Japan). Newly submission sequences are transferred between the three databases on a daily basis.

[8] This database was released in December 2003 and is the coming together of Swissprot, TrEMBL and PIR to form a non-redundant set of all known protein sequences.

[9] If you do not know what these databases are, click on the name and it will link to a description of the data held.

[10] EMBL contains millions of sequences, and new sequences are added every day. There is a new release of these databases approximately every three months, and any new submissions before that date are released as updates. The separate libraries can be seen by looking under the Nucleotide Sequence Databases – subsections" heading.

[11] EMBL is composed of the main release, and the "updates" which include any new entries between releases.

[12] As the database gets updated, this may change.

M77844, was not pulled out from this SRS search. If you go back to SRS and search for M77844 in the field AccNumber, you will find that the word “pax6” is not mentioned. This gene is described as oculorhombin, which you may remember has been mentioned before as an as alternative name for PAX6. SRS does text-based searches; you need to think hard about the best terms to use in your search.

From the description you should have identified **EMBL:HSA1280 (Acc: Z83307)** as one of the genomic sequences. There is another one, EMBL:HSCFATS, which also shows a complete clone sequence. This is the clone upstream of our A1280 clone and actually contains several exons of the PAX6 gene in the un-translated region. The other sequences are either incomplete or contain an mRNA sequence.

Select the genomic sequence link and go directly to the EMBL entry. Examine the EMBL entry for the full length PAX6 clone, as an illustration of the format used in EMBL data files.

The view you will see initially has been created specifically for the user. To see the “format” of the entry and understand what the computer is seeing, click the “Text Entry” link at the top of the page which will display the same information in textual EMBL format.

The EMBL Data Format

ID	Sequence identity, type, organism and length in short form
AC	Accession number
DT	Dates of entry creation and modification
DE	Sequence description
KW	Keywords
OS, OC	Organism and full taxonomy
RN	References linked to Medline (PubMed entry numbers)
DR	Link to protein sequence in SwissProt
CC	Comments
FT	Feature Table: table of features within the sequence; each can be accessed as a separate “pseudo-entry” including: source coding sequence (CDS) “miscellaneous features”, here a CpG island repeat regions
SQ	Sequence (beginning with sequence composition)

The feature table is one of the most important aspects of the EMBL data format. Any

sequence feature can be expressed in this format. It is easy to express complex ideas, as features can be combined and arranged hierarchically. As one example, a number of exons can be arranged into a single coding sequence.

The sequence format used in GenBank, which is similar to EMBL format, also includes feature tables.

UniProtKB (UniProt Knowledge Base) is the protein sequence repository and was created in 2003 with grants from both the European Union and the National Institute of Health in the USA. It amalgamates three protein databases. SwissProt has its entries curated by experts meaning that the literature and other sources of information surrounding a particular protein have been searched by biologists and data entered into the database is as accurate as it can be. TrEMBL is the automated annotation of proteins based on open reading frames available in the EMBL database. This was set up as the curation process is much slower and now SwissProt curators take their sequences from the TrEMBL entries. PIR was the Protein Information Resource hosted in the USA. When the three databases were amalgamated, the redundancy was removed between the three databases, and any additional protein entries from PIR were put into the TrEMBL section of the database. Both SwissProt, TrEMBL and PIR are accessible as divisions in the new UniProt database.

It would be possible to select the UniProt (protein) entry for the PAX6 protein sequence just by clicking on the single link in this entry. This may not be the case in a more complex query. We will therefore go back to SRS to retrieve the protein sequence by linking to the UniProt database.

Use the “Back” button on your browser’s tool bar to go back to the query page of SRS defining your hits. Tick the box to the left of the correct entry and ensure that the “selected results only” box in the left hand column is ticked. Then select the yellow “Link” button. Select the “UniprotKB/SwissProt” database and click on “Search”. Follow this link to the entry of the Pax6 protein. Verify the name of the genomic clone in the EMBL cross-references. Why are there several references?

UniProt Accession Number……

UniProt Identifier (or entry name) ……

The entry held in the database at the actual UniProt site will offer more information on this entry.

Launch another browser window and go to the EBI homepage at http://www.ebi.ac.uk. Type the accession number into the top box and alter to pull down menu to read “Protein Sequences”. Examine the UniProt entry for the PAX6 protein. Search for the table which describes the isoforms (Alternative Products).

This should correspond to the information you obtained through GeneCards. You will see that there are three isoforms. The first one is 422 amino acids in length and the second has a length of 436 amino acids (click the sequence links to see this information). The third isoform has a “sequence not described” notice.

"Sequence not described" indicates that Western blot results using a high specificity monoclonal antibody suggesting 2 or more distinct forms of the protein have been discussed in the literature.

"No experimental evidence..." in this section means that the splice variant is only identified as a result of a cDNA which is generally regarded as less reliable than sequenced DNA. Thus, flagging up the possibility that whilst it has been annotated as a splice variant, it may turn out to be a sequencing error.

All splice variants in UniProt/SwissProt are however checked out first to ensure they are not actually just errors due to frameshifts or intron retention etc.

Alter the pull down menu at the top of the page to read "protein Sequences" and type in the UniProt accession number into the "Search" field. Hit the "Go" button. Scroll down the entry to the "Comments section". Follow the "Display all Isoform sequences in Fasta format" link. Check the boxes to the left of the sequences and hit the "Submit query" button to perform a global alignment. Leave all settings as default and hit "Run Query". Follow the "Clustal(aln)" link.

ClustalW is a very common method for aligning two or more sequences. In this case, the two sequences are more or less the same length and extremely similar and thus it provides a fast way of searching for the differences. The extra amino acids in the second sequence show the additional peptide chain in the second isoform.

Extra sequence in **sp_vs|P26367-2** _____

Return to your first protein entry (the one you got to through SRS) and scroll down to the **feature table** and the "varsplic" feature. Where does this occur? Follow the link to the actual feature sequence by clicking on the residue numbers. Compare this with the extra sequence from the global alignment above. Look at the information on this isoform further down in the feature table.

If you are searching for anything else related to the protein, the relevant database can also be queried further using the "Link" facility[13] once more.

Return to the protein entry page from your SRS query and click on the "Link" button on the left hand side towards the top of the page. Select the "Mutation and SNP databases" header in black towards the bottom of the page and explode it by clicking the plus sign in the box. Click the box beside "HGVBASE[14]" so that it is ticked, and "Search".

HGVBASE identifier (first entry) .....

[13] This approach is only possible where the SRS search engine is connected to the required databases.

[14] The Human Genome Variation Database attempts to summarise all sequence variation data in the Human Genome. This also includes repeats and indels. It is currently being re-developed into a phenotype/genotype database.

HGVBASE (Human Genome Variation Database) is highly curated, and currently holds over 2.8 million entries. These include all sequence variations in Humans, such as Indels and repeats, as well as SNPs.

Unfortunately it is currently undergoing a funding crisis and currently cannot accept new submissions. This means that the database slowly becomes out of date and links to and from it may no longer be maintained. The curators are currently working on a solution.

Follow the link for the first HGVBASE entry.

Allele change .....

DNA Ref Molecule (Ensembl) .....

DNA Ref Molecule (EMBL) .....

This locus pinpoints the SNP in both the EMBL database entry M93650 (the cDNA sequence for the human pax6 gene) at bp position **837** and the Ensembl Genome Browser. The Ensembl link is shown as `Ensembl` (database)`::Chr11` (chromosome)`.14.31` (sequence and build version)`:31786651` (base pair position).

Is this correct – if not, why do you think this is?

Are there any references for dbSNP? What about any other databases?

dbSNP ref .....

There have always been various sites that combine the information gleaned from the sequencing of the Human Genome and if you wish to use the NCBI in America, you would find the information at http://www.ncbi.nlm.nih.gov/genome/guide/human/ very useful. This site incorporates cytogenetic data with sequence information, to offer you a detailed insight into the Human Genome.

This information can also be found at the European site http://www.ensembl.org. **Ensembl** is a collaborative project between the Sanger[15] Institute and the EBI[16] and is designed to allow you free access to all the genetic information currently known about the Human Genome. It automatically annotates current genomic data, and as this

[15] Centre at Hinxton, Cambridgeshire, responsible for sequencing one third of the Human Genome aswell as other genomes. Visit the centre at http://www.sanger.ac.uk

[16] European Bioinformatics Institute –responsible for the EMBL database and specialised bioinformatics tools. Located at Hinxton, visit the EBI at http://www.ebi.ac.uk

project advances, the site will change, and information will be added or removed as more evidence for or against a particular genetic feature becomes available. Genes are predicted using the Genscan[17], FGENESH and GeneWise[18] programs, and corroborated with supporting evidence found in the protein databases.

Information on each gene is presented in a variety of interlinked webpages known as "View"s. Thus GeneView presents information about the gene, ProtView about the protein, ContigView on the contig[19] and so on.

Go to http://www.ensembl.org and select the species "*Homo sapiens*" (by clicking on either the picture or the "browse" link) and then browse the chromosome you know the Pax6 gene to be located on. On the ideogram presented to you, find the location where you know the gene to be. Is it a gene dense region? What about SNP density? (***Hint: Do not click on anything here, just look at the various densities***)

Under the "Jump to ContigView" banner on the right hand side of the page select "Gene" from the pull down menu selections in the Region "From" and "To" and type the name of the gene causing aniridia into each text field. Hit the red **GO** button.

You are now in the Ensembl ContigView. The red marker defines which region of the chromosome is being presented and the overview indicates any neighbouring genes. The detailed view shows the gene in question in detail – displaying transcripts and supporting evidence.

The detailed view box displays one megabase of sequence and the contig region, which contains it. The long arrow at the top of the detail view screen represents the length of sequence you are viewing. Below are the various features found on the other strand of that particular section of the genome. This strand looks much more promising. Just below the blue contig line, there are transcripts of the gene and the information from the Genscan program. You will see homologies found in the Uniprot and EST databases, any markers known in that area, and any CpG islands that have been found. The bottom line displays the megabase scale of the sequence in that region of the chromosome. You should see that four exons belonging to the pax6 gene are located on a different contig.

Contig numbers (containing pax6) ……

The PAX6 transcripts are marked. The thick vertical blocks represent the exons, connected by the thinner horizontal lines (intronic sequence). You can see the discrepancy in the number of exons in each transcript.

[17] Burge, C. B. (1998) Modeling dependencies in pre-mRNA splicing signals. In Salzberg, S., Searls, D. and Kasif, S., eds. Computational Methods in Molecular Biology, Elsevier Science, Amsterdam, pp. 127-163

[18] Ewan Birney, Michele Clamp and Richard Durbin (2004) GeneWise and Genomewise *Genome Research 14: 988-995*

[19] Contig is short for contiguous sequence and defines the region of the original clones used to sequence the genome. They are separated in the contig view by an alternating light/dark blue display.

There are ten possible transcripts – the blue ones indicate that the transcript has been found in the literature and been curated as part of the VEGA project (http://vega.sanger.ac.uk/index.html) by the HAVANA group at the Sanger Institute. The red transcripts are Ensembl genes which have been predicted in the Ensembl pipeline and corroborated using protein evidence in a variety of databases.

Look back at the Overview display to see a gold transcript.

This signifies an exact overlap between an Ensembl transcript and a VEGA gene.

So, with all these transcripts, how do you know which one is the correct one? Indeed, they may all be correct in a biological sense. However, in this case, there is evidence for only one protein in UniProt/SwissProt and the relevant transcripts are highlighted in green.

Are all transcripts contained in one contig?

As part of the transcript neighbourhood view, exons (the vertical boxes) may be filled in, indicating true exons, or unfilled, indicating that this is part of the un-translated region (UTR).

Are there differences between the numbers of exons in each of the transcripts? Where are those differences most likely to occur?

Mouse over one of the red transcripts and click to retrieve the menu. Select the Gene option (the gene accession number here should be the same one you wrote down earlier).

You are now in the Ensembl GeneView, which contains a summary of all the information on the gene, plus links to that gene entry in many other databases. Do

you recognise any of them?

Gene Location .....

The ten transcripts are also listed in this view. Unfortunately those listed in the “Transcript Structure” are not necessarily in the same order as those depicted in the exon diagram. Click on “exon information” for each of the transcripts and scroll to the bottom to see the “supporting evidence”. The coloured boxes indicate evidence for a particular exon from each reference sequence listed on the left hand side. Are there any exons you are less sure about after looking at the evidence?

There is an additional exon within the contig Z83307 on several transcripts (as evidenced in proteins NP_001595 and UniProt/TrEMBL Q6UMP0). This results in an

alternatively spliced protein – isoform 5a.

Follow the link to the UniProt/SwissProt protein entry in the "description" section of the GeneView.

You should now be looking at the protein entry in UniProt for the PAX6 protein. Scroll down the page to the "database cross-references" and you will see the EMBL entry M93650. This is the entry giving the true cDNA for the protein.

Follow the EMBL link to the M93650 entry

Scroll down to where the feature table begins (identified with "Features" on the left hand side of the page). At some point, the exons transcribed to create the protein are described. Count them. How many are there?

There should be 13 exons, but note the start and end of the CDS (coding sequence). This represents the bases which are translated into the protein. Some of the exons do not fall into this region. Why?

Return to the GeneView in Ensembl and follow the "Peptide Info" link to "ProtView" for the Pax6 transcript. Note down the first two Interpro IDs that appear in the relevant section of the cross references.

Interpro IDs ……

In addition to the information on the gene overview, you may view SNPs and other information, by selecting further sources of data:

Return to the "GeneView" page of Ensembl and follow the top "Genomic Location" link to the ContigView once again. Select the yellow "Features" menu in the "Detailed View" and activate the SNPs data trace by clicking in the box next to this feature. Close the menu and the ContigView will be redrawn with this information. Look at the information on the "SNPs" row towards the bottom of the display.

The SNPs are colour coded according to what type they are. There are many more intronic SNPs than exonic ones. Can you find the same SNP you discovered in HGVBase?

Ensembl has a built in search system to allow you to query the entire database which is called BioMart. We can use it to see if there are any other genes that produce proteins with a similar function domain topology to PAX6 that cause disease.

Return to the Ensembl homepage and follow the link on the left hand side of the page to the data mining tool Biomart.

The initial page will also default to searching the Homo Sapiens genome, so if you wish to work on another organism, then this must be altered.

Select Ensembl38 from the pull down menu if it is not shown be default. The second menu should record the gene set chosen as that of *Homo sapiens* genes NCBI build 36.

Proceed to the **next** page to specify your query further.

The summary on the right hand side of the page will have duly noted that you have chosen to focus the search on "Ensembl Genes" within the species "*Homo sapiens*" and that there are 34,583 entries within that geneset (these contain non coding genes such as pseudogenes and ncRNA).

On this, the "Filter" page, enter the following data:

REGION: Ensure this box is unchecked to search the entire genome

GENE: Check and alter the first menu to "with Disease Association"

Check "Transcript count" and make it equal or greater than 2 to limit to genes with multiple transcripts only

Check "Gene Type" and leave the menu as "protein coding"

Check "status" and alter the menu to "known"

Hit the "Count" button on thee right hand side of the page – there should be 774 entries that pass current filters.

PROTEIN: Limit to "Interpro IDs" and copy the two Interpro IDs of the pax6 gene into the box

SNP: Check "coding"

Select "SNPs with HGBASE IDs"

Leave all other parameters untouched and proceed to the **next** page. Only 21 entries pass filters.

This is the "Output" page and on the "Features" tab, enter the following data:

REGION: Check the "Chromosome Name"

GENE: Check Ensembl IDs

EXTERNAL REFERENCES: Check Uniprot/SWISSPROT IDs

DISEASE ATTRIBUTES: Check OMIM ID
Check Disease Description

PROTEIN: Check Interpro ID

Check the "UniProt/SwissProt" IDs and the "Disease Description". Leave the output selection as "HTML".

You will have 21 disease entries that involve proteins with similar domains to the PAX6 gene and have filtered them down from the 34,583 entries in the entire ENSEMBL human database. To do this by hand would have taken you an awful lot longer and is an illustration of how computers can aid detection of specific information. [You will also note that of these entries, there are several duplications. This is often because each gene has several criteria or can be mapped to several phenotypes].

If you are doing more work with proteins, you might want to investigate the **Expert Protein Analysis System (ExPASy)** held at the Swiss Institute of Bioinformatics. This site not only holds the SwissProt and TrEMBL[20] database sections of UniProt, but also offers many tools for the user to analyse their protein sequences. We will return to ExPASy later on in the course, but for the moment, we will query the protein database for our Pax6 protein.

Go to http://www.expasy.org and select "SWISSPROT and TrEMBL from the left hand side of the page. Type EITHER the accession number OR the identifier you retrieved using SwissProt into the "Search" field at the top of the page and click the GO button. Examine the entry. Although the layout of the data in the NiceProt entry may be slightly different, the information is exactly the same as you saw using NiceView via SRS at the EBI.

You will also notice that this view offers the possibility of some analysis – in particular the inclusion of this protein in a BLAST search. Following this option will display a textual output of a sequence homology search of the databases SwissProt and TrEMBL. We will be looking at BLAST searching later in the course.

More databases

[20] TrEMBL is a database which holds peptide sequences resulting from automatic translation and annotation of the EMBL nucleotide sequences -hence Translated EMBL. It is generally seen as a precursor to SwissProt which contains manually annotated peptide sequences.

We have only been able to include a few of the many databases available on the Internet in this case study. You might like to try exploring a few others if you have the time, using either PAX6 or a gene or protein you are interested in.

For example, you might like to try:

Reactome

Have a look at a new project which tries to detail pathway information at http://www.reactome.org Expert biologists and computer scientists have come together to create a highly curated database of reaction pathways. Currently only the Human pathways are created, and data in other organisms is inferred if there is enough evidence.

KEGG

The primary objective of KEGG is to computerise the current knowledge of molecular interactions; namely, metabolic pathways, regulatory pathways and molecular assemblies. At the same time, KEGG maintains gene catalogs for all the organisms that have been sequenced, and links each gene product to a component on the pathway. It also contains a database of all chemical compounds in living cells and links each component to a pathway component http://www.genome.ad.jp/kegg/kegg.html

If you are not working on Human genes, you may have to search further for information. The NCBI is a good place to start for completed genomes. These include many bacterial genomes at http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome.

The **Institute of Genome Research (TIGR)** is also a good place to look. TIGR also offer Gene Indices of organisms that have not yet been completely sequenced. Try TIGR at http://www.tigr.org.

The ArkDB is a repository for sequence information from various animals generally seen in Britain as Poultry and Livestock. The database also offers information on animals such as the cat and salmon. http://www.thearkdb.org/ also has rather nice little cartoons of the animals in question! It is curated at the Roslin Institute just outside Edinburgh in Scotland using funds provided by the BBSRC[21].

Plants are represented in a variety of databases. The model genome for the plant species, Arabidopsis thaliana is hosted in an Ensembl type database at http://atensembl.arabidopsis.info/. The Gramene database at http://www.gramene.org/ is an open source, curated website for the comparison of grass genomes.

Also remember WWW search engines. http://www.google.com is probably one of the

[21] The Government funded Biotechnology and Biological Sciences Research Council.

best, but use your favourite for a text search.

EMBOSS – A data analysis package

During this module, you will use the available PAX6 sequences to perform some of the tasks researchers need to do very frequently. You will learn how to perform pairwise alignments, how to make restriction maps and design PCR primers, and more importantly, you will start to become familiar with the tools that are available to you and will know how to look for programs that will help you further.

The exercises in this chapter and many of those later in the course are based on a freely available sequence analysis package called **EMBOSS**. This is not the only sequence package available to you - you can also use **GCG**, **Staden** and various others. You should be aware that EMBOSS is actively being developed and new applications are frequently added. Please contact support on emboss-bug@ emboss.open-bio.org if you experience any problems using EMBOSS.

A list of current and proposed applications is maintained at http://emboss.sourceforge.net/Apps/. We'll run through a few examples of simple EMBOSS programs to get you used to the interface we are going to use, and then move on to sequence alignment.

Graphical User Interface

The interface we will be using is called Jemboss[22], and it has been written at the RFCGR by T. Carver in collaboration with the EMBOSS team. It has been designed to run on both PC and UNIX systems[23]. It runs with the help of a Java plug in called Java Web Start. This will have been done on the machines in the room.

Go to the Jemboss homepage in your browser, and select the link "Launch Jemboss". You may have to wait a few moments for the program to load when you first do this.

After a pause of a few seconds, during which time you will be asked to type in your username and password[24]. Click on "OK" and the full Jemboss interface should appear on your screen. On the left hand side you will see analysis groups. Click on these to display other menus and programs belonging to that group. You will see the name of the program, along with its one line description. This should aid you in making a

[22] So called because it has been written in the programming language Java. This language allows programs to function on any java enabled computer platform.

[23] Unfortunately, the old Mac operating system does not support Java and unless you have MacOSX (purchase since 2001), you must use Jemboss via VNC.

[24] For the purposes of this course, you will be assigned a username and password. Otherwise you should use the identification you received when you registered with the RFCGR.

decision about the correct program to use. Alternatively, programs are listed in alphabetical order underneath these groups. If you know which program you wish to use, you may scroll down the menu and select it by clicking once on the program name. Alternatively, you may start typing in the name of the program in the "GoTo" box, and when it appears in the menu, just click on it.

The central panel is where the program form will appear. In red at the top of the form the name of the program is displayed, followed by its one-line description. Below this, there are the sequence entry options[25].

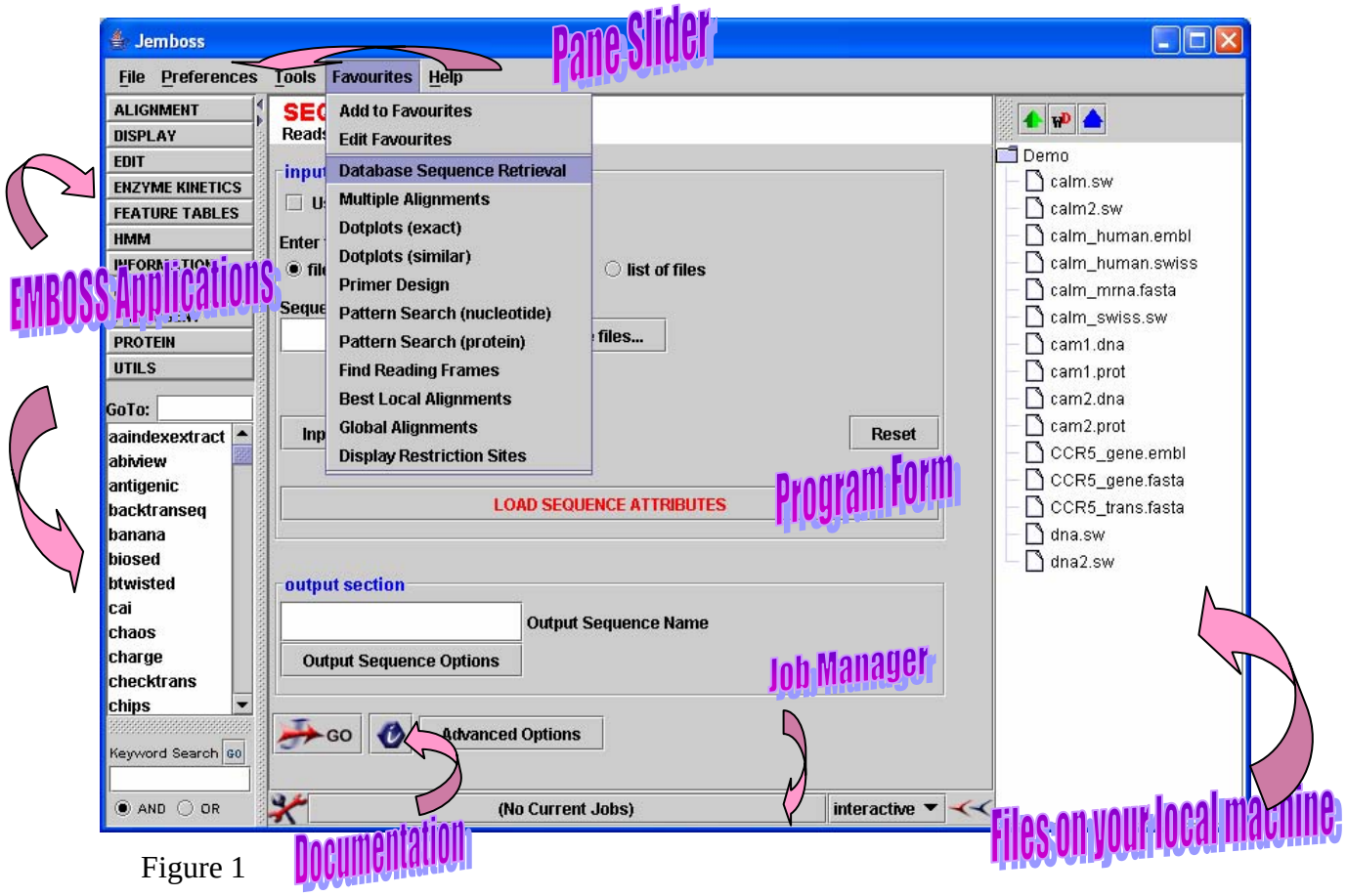


Figure 1

Sequences may be added to the program in a number of ways. They may be entered straight into the filename box using the conventional EMBOSS Universal Sequence Address (USA)[26]; files may be transferred from local or remote accounts by "drop & dragging" the relevant file into the filename box; sequences may be pasted into the filename box by selecting the "paste" option. For programs requiring multiple *sequence options*, files may be added separately. Sequence properties may also be entered using

[25] Some programs will not require a sequence, and in these cases there is, obviously, no sequence entry field.
[26] The sequence address communicates to the program which sequence you would like to retrieve from which database. It is written in the form **database:sequence** where sequence represents the sequence identifier or accession number.

the "Input Sequence Options" menu. This latter option allows you to choose the database from which you would like to retrieve your sequence, but you must ensure that you TYPE the sequence identifier or accession number INTO THE SEQUENCE FILENAME BOX[27].

Different applications will have various options available on the program form. Select, or fill in these options as required.

If an "Output Sequence Name" is entered, your output will automatically be saved onto the server, using that name. If you do not enter one, however, the file is automatically saved onto the server using a default output name, and you may retrieve it at a later date.

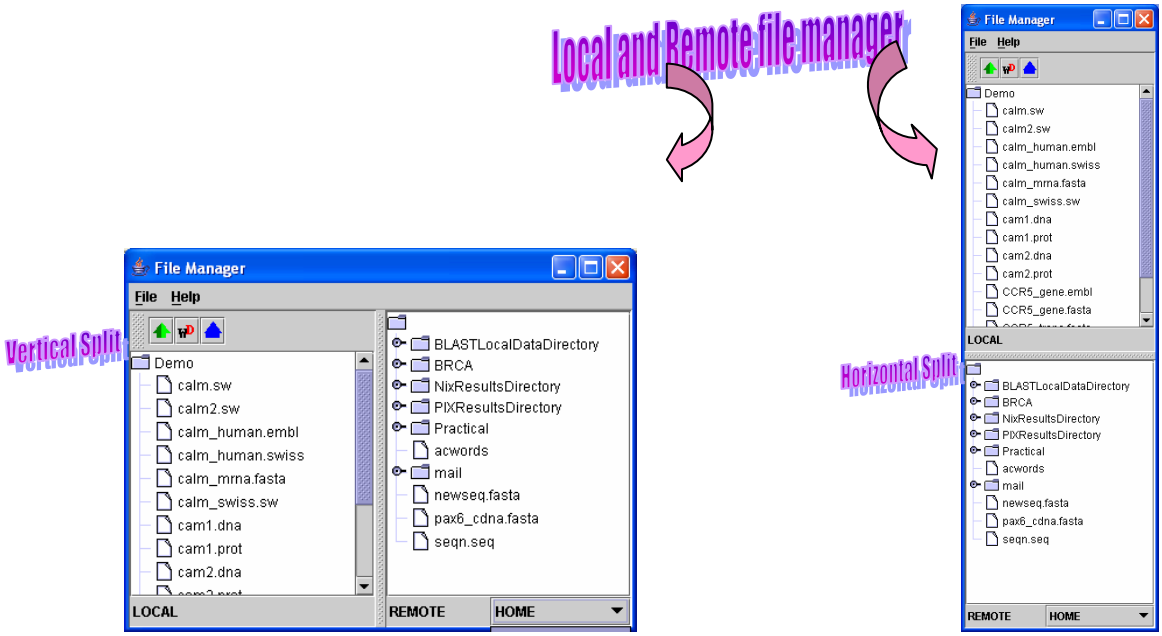


Figure 2

Both the remote and local file managers can be accessed using the "File" menu of the Jemboss interface and selecting "Show Local and Remote Files". An elongated window (figure 2) will pop up on your screen and you may split the screen horizontally or vertically. Files may be dragged between local and remote computers by picking up the file to move, and placing it over the name of a file in the relevant manager. You may also drag & drop files from this window to the sequence input field. The drop down menu at the top of this window is to allow you to toggle between files that are stored in your home directory, and your scratch directory.

Jemboss has an inbuilt facility to allow you to transfer files between your local disk and your server account. Just drag and drop the required files from one list to another to move them between accounts. Each file manager also has an options menu associated with it. By clicking the right hand mouse button once a menu will pop up, offering the option to refresh the list of files. This is particularly useful if you have just created a

[27] If you do not type in a sequence identifier, the program will try to download the whole of your specified database, when using programs like "seqret", for example.

new file or folder. Options to delete a file or folder are also on the menu, together with renaming a file, or creating a new folder.

Hit the "Go" button to run an application. If you do not understand the significance of a parameter option, use the "mouse over" facility. If the parameter has a specific help text attached to it, it will appear on the screen. Alternatively, click on the  button, which will take you to the program documentation.

Your results will be presented to you in a new tabbed window on the screen (Figure 3). The first tab should contain the results of the program that you ran. The name on the tab is the default output name, or a filename specified by you. You may save your results to your local computer. A second tab (**cmd**) will give you the command line[28]. If you are saving graphical output, you may save it as a ".png" file (the default). This format may be inserted as a picture into Microsoft Word documents. Other tabs contain any sequences you may have pasted into the "input sequence" window.

Figure 3

You can then view your file using Jemboss, by double-clicking on the filename[29]. Alternatively, you can save the file to your local machine and then view the output using a WWW browser. The file manager on Jemboss should automatically update to show your saved files. When you have finished with the window, it can be closed.

If you have entered your input sequence and filled in any extra sequence parameters you wish to use in the "Input Sequence Options" box, you may then click the "LOAD SEQUENCE ATTRIBUTES" button[30]. This will then display all remaining sequence parameters for the program that are pertinent to you sequence only. Parameters that are not appropriate for your sequence will be greyed out. If you would prefer redundant (for a particular sequence) parameters to be absent on the screen, de-

[28] As EMBOSS runs on a UNIX operating system, the optimal way of executing a program is to write out all the commands you wish the program to do on one line – the command line. The interface will actually do this for you, but if you work with the UNIX system, or would like to start, it may be useful for you to know the exact command line used.

[29] Files may also be edited in this way.

[30] It is important to input all sequence options first, otherwise the sequence attributes will be overwritten.

select “Shade unused parameters” from the “Preferences” menu of the Jemboss interface.

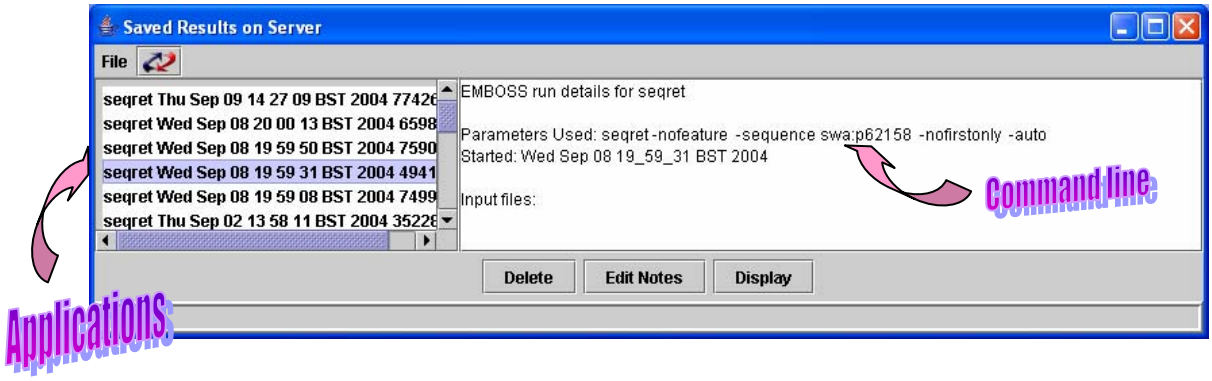


Figure 4

Saved results can be accessed by using the “Saved Results” option on the pull-down “File” menu in the Jemboss window. This will display all previous applications you have run. Details are displayed in the right hand pane in command line format to offer a reminder of exact parameters used for a particular analysis. Results can be displayed by selecting the appropriate entry and clicking on the “Display” button. Information and results files can be discarded by clicking first on the program and then on the “Delete” button.

Programs may be run interactively, or in batch mode. This allows the analysis to be carried out in the background, whilst you get on with doing something else. The job manager is located at the bottom of the Jemboss window, and will keep a tally of all batch programs running, and all those completed. For each session, the job manager will also record all analyses run, and you may recall and display them as you would using the program manager. The job manager will automatically refresh every 15 seconds, to let you know whether your analysis is still running, or whether it has finished. This time frame may be changed, by choosing the “Advanced Options” of the Jemboss “File” menu and altering the options in the pull down job manager menu.

Jemboss Practical

Here are a couple of simple programs to familiarise you with Jemboss.

Use the keyword search at the bottom of the page by typing “translat” in the field. Click on the “Go” button and note the number of entries. Do the same thing with “restrict” and note the number of entries. Now try putting both words in to the keyword search – first using the AND button, and then the OR button.

A window will appear giving details of all the programs in EMBOSS which have the relevant words in their one line description. The AND option will combine all words in the search, the OR option allows either one or the other to be allowed in the one line description, but not both. Although Jemboss has grouped programs according to functionality, if you are new to the package, it isn’t always obvious which program to use.

Re-use the keyword search, this time typing “databases” into the search box.

Towards the bottom of the list is a program called “showdb”. According to its one line description, this program displays information on the currently available databases. We will use this program now.

Use the “GoTo” box in the middle of the left hand margin and type in as many letters as you need of the program “showdb” for it to be highlighted in turquoise in the scroll menu. Hit <enter> to select the program. Do not fill anything in the database field, and leave the “protein databases” and “nucleic acid databases” options ticked. Click the “Advanced Options” button to display further parameters, select the “Display Column Headings” and click on “Go”.

The left hand column of this list details all the databases that are available for use with EMBOSS[31]. The names contained in this list represent the identifier of each database to EMBOSS applications.

EMBL identifiers[32]

……..……

Swissprot identifiers ……

Letters in the “Type” column indicate whether the database contains **P**rotein or **N**ucleotide sequences and an “OK” in the subsequent three columns “ID”, “Qry” and “All”[33] represent whether you may query the database for a single sequence, a set of sequences, or all sequences in that database respectively. The “Comment” column provides a description of the database. If you do not need all the information provided in the output, you may select the “Display specified columns” option and choose the information you require.

You do not need this window anymore, so close it down.

None of the Jemboss windows will close automatically, so in order to keep your desktop tidy and retain computer speed, it is best to close down windows as you no longer need them.

[31] Like SRS, this may change depending on where you are using the service. If your university is providing EMBOSS as a service, the database selection may be more limited than a large national bioinformatics centre.

[32] These are the names, in the first column, that the databases may be called. These names may then be used as part of the Uniform Sequence Address (USA) of a database entry.

[33] For more than one sequence, you can use a wildcard. This is represented by an asterisk inserted after any identifying characters. So searching for all pax sequences in Swissprot, you could type **sw:pax*** into the sequence field of an application. If you wanted to search all the files in this database, you would enter **swissprot:*** into the field.

Introductory Sequence Analysis

There are various entry points to analysing your data. If you already have the sequence, or the gene has been relatively well characterised, you may use some of the entry points we looked at in the previous section. If you know nothing about the piece of DNA that has just come off the sequencer, then you will want to perform a range of analytical tasks on the data to try and establish what it is. This initial analysis would probably commence with a BLAST search. However, for the purposes of this course, and explaining things in a sequential order, this type of database searching will be covered later on in the course.

You have your own sequence, called pax6_cdna.fasta. You think you may have a mutated form of the human pax6 gene, but would like to make sure before you write your paper. From your database searches, you know that there is a genomic clone that should contain the pax6 gene, so we will start off by aligning your cDNA fragment against the full genomic sequence[34]

Use the "Favourites" menu on the Jemboss toolbar to call up the "Database Sequence Retrieval" option. This loads the program "seqret" into the program form.

The Favourites menu has eleven classic programs listed. These can be edited and the list extended with programs you may use more often. Any of the eleven you do not need may be deleted.

Ensure "file/database entry" is selected and then call up the "input sequence options" menu. The genomic sequence is in the EMBL database, so select **embl** from the "Databases available" menu. You should see your selection appear in the "Sequence filename" box of the interface window. Click on the "OK" button at the bottom of the panel. After the colon in this box, type in the accession number of your genomic sequence[35]. Click on "Go" to run the program.

Your results will appear in a tabbed window called "HSA1280.fasta". This is the default output name assigned by the program, as we didn't specify one. Note the composition of the fasta format. The first line contains a greater than sign (>) followed by a description of the sequence. The sequence commences on the second line of the file.

Save the sequence file to your local (PC) account as **pax6_genomic.fasta** and close the results window.

[34] You identified this during you database searching with SRS.
[35] This is incredibly important, otherwise you will start to download the entire EMBL database.

Sequence Comparison

There is no unique, precise, or universally applicable notion of similarity. An alignment is an arrangement of two sequences, which shows where the two sequences are similar, and where they differ. An optimal alignment, of course, is one that exhibits the most similarities, and the least differences. Broadly, there are three categories of methods for sequence comparison.

- **Segment methods** compare all overlapping segments of a predetermined length (e.g., 10 amino acids) from one sequence to all segments from the other. This is the approach used in dotplots.
- **Optimal global alignment** methods allow the best overall score for the comparison of the two sequences to be obtained, including a consideration of gaps. These programs align sequences over their whole length.
- **Optimal local alignment** algorithms seek to identify the best local similarities between two sequences also including explicit consideration of gaps. Alignment may only be over a short span of sequence.

Dotplots

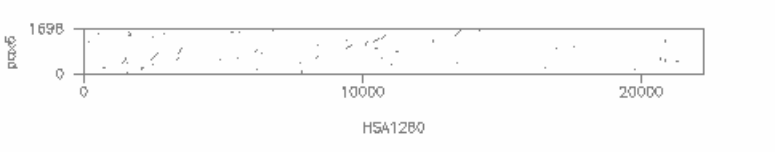
The most intuitive representation of the comparison between two sequences is using dotplots. One sequence is represented on each axis and significant matching regions are distributed along diagonals in the matrix.
There are two different algorithms that are commonly used in creating dotplots. The first method involves matching identical regions of sequence and plotting a dot in these areas. The second involves using "sliding windows" to compare two sequences using a threshold score[36] value. A window size is selected as a run of adjacent nucleotide or amino acid residues, and a score chosen to reflect the degree of similarity of sequence required. Each window of sequence A is compared to each window of sequence B, and a dot is only placed in that region if the match scores or exceeds the set threshold level

It should be immediately obvious which sections of sequence align with each other and further investigation of specific areas of the sequence can be conducted.

Select the program "dottup" from the scroll menu or under the "Alignment/Dot Plots" menu (or Dotplots (exact) from Favourites menu). Drag you genomic sequence into the first sequence input box, and your cDNA sequence into the second box. Leave the "word size" as the default (10) and hit "Go".

Your results should be displayed in a rectangular box with the genomic sequence along the bottom axis, and the cDNA sequence on the vertical axis. It should look like this:

[36] A score is calculated between two sequences using a two dimensional array of numbers called a matrix. The matrix for DNA is relatively simple, with each exact base match scoring 5 and each mismatch scoring –4. Various other scores represent matches between ambiguity codes. Matrices for amino acids are more complex, and are discussed later in the chapter.

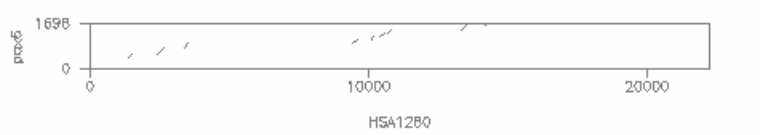


Due to the size discrepancies between the two sequences, the result appears in a long, thin oblong, that is difficult to see. If you wish to see more detail, try selecting the "stretch axes" option before you run the analysis.

Accepting the default word size of ten means that the sliding region that is compared against the two sequences is only ten nucleotides long and as you can see, we get a lot of extra dots (noise) The reason for this is background matching – areas of random identity between the two sequences. We can reduce this by selecting a larger window for the sliding comparison.

Re-run "dottup", but this time, select a window size of 50.

This is the result. Nine larger regions of identity run between the two sequences.



We are comparing genomic sequence with cDNA, so it would be reasonable to assume that this plot represents the nine exons found in the pax6 gene. Does this tally with the number of exons you can find using the information in Ensembl (www.ensembl.org)? The background noise has been cut out completely, but other features, such as repeats may have been lost, which may not necessarily be what you want in all cases. It is often best to make several dotplots using different parameters.

	A	T	G	C	S	W	R	Y	K	M	B	V	H	D	N	U
A	**5**	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2	-4
T	-4	**5**	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2	5
G	-4	-4	**5**	-4	1	-4	1	-4	1	-4	-1	-1	-4	-1	-2	-4
C	-4	-4	-4	**5**	1	-4	-4	1	-4	1	-1	-1	-1	-4	-2	-4
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-1	-1	-3	-3	-1	-4
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-3	-3	-1	-1	-1	1
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-3	-1	-3	-1	-1	-4
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-1	-3	-1	-3	-1	1
K	-4	1	1	-4	-2	-2	-2	-2	-1	-4	-1	-3	-3	-1	-1	1
M	1	-4	-4	1	-2	-2	-2	-2	-4	-1	-3	-1	-1	-3	-1	-4
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-1	-2	-2	-2	-1	-1
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-2	-1	-2	-2	-1	-4
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-2	-2	-1	-2	-1	-1
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-2	-2	-2	-1	-1	-1
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2
U	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2	5

If you know that your sequences are not exactly alike, but should be relatively similar, you may want to choose the second dotplot method. We can use the EMBOSS program **dotmatcher** instead. Remember that with this algorithm, a match score is calculated for each sliding window, using a scoring matrix. In this case we are looking at nucleotide sequences, but amino acid sequences may also be aligned. Consider the sequences ACGTACGT and AGGTACCT. Comparing the two we see that the first nucleotide in each sequence is an A - so we have a match at this position. If we look at the scoring matrix given above, we see that matching A against A adds 5 to our score.

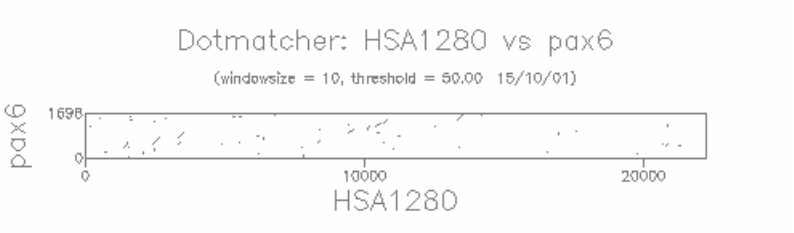
The second position is C in one sequence and G in the other - and we see from our matrix that this subtracts 4 from the match score. The program continues to score the sequences, adding and subtracting, and if the total score for the window exceeds a threshold that you set, a dot is placed in the output and the window slides on.

Select “dotmatcher” from the scroll menu (dotplots (similarity) from Favourites menu) or the “Alignment/Dot Plots” menu. Again fill the sequence boxes with the genomic and cDNA. Accept the default values for “window size” and “threshold”. Leave the matrix field blank. This will ensure that the default matrix EDNAFULL[37] is used. Hit “Go”.

You will be rewarded with what looks like a black box and a few white dots. The default threshold is 23, which allows for 3 mismatches out of ten nucleotides, so with our input sequences that is going to be quite a lot!! The white areas are those where there was no match.

Leave the default value for “window size”, alter the “threshold” to 50 and hit “Go”.

There is some background of dots - but remember that we are scoring over a window of length 10 and our threshold is set at 50. A quick glance back at the scoring matrix should convince you that you don't need many matching nucleotides to push the score up!

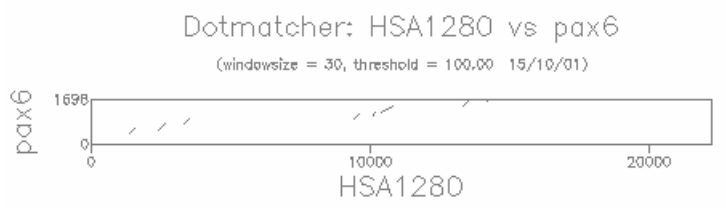


However, remember that from our matrix, the maximum score we can get with a window size of 10 is 50, so there isn't any point in increasing the threshold further - and in fact, to do so would mean that we would lose the matches we see now. What we could do instead is to increase the window size - remember, our exons are going to be much longer than 10 nucleotides!

[37] This is the matrix depicted above, and includes all DNA ambiguity codes.

Re-run dotmatcher with a window size of 30. At the same time, increase the threshold to 100.

The resulting plot should look like this:



Again, you should note that we are emphasising "stronger" features at the expense of smaller ones.

The dotplot can be very useful in giving a rapid overview of the level of similarity between two sequences, but it doesn't give us any detailed sequence information. For this, we need to use different programs.

Sequence alignment

We'll move on now to some alignments that let you see more detail of the sequences you are using. The algorithms we will be using are more rigorous than those used for searching databases; so even if you have retrieved a sequence from a database using something like **BLAST** (we'll be looking at this later), it will be well worth your while performing a careful pairwise alignment afterwards.

The basic idea behind the sequence alignment programs is to align the two sequences in such a way as to produce the highest score - a scoring matrix is used to add points to the score for each match and subtract them for each mismatch. The matrices commonly used for scoring protein alignments are more complex than the simple match/mismatch matrices used for DNA sequences such as the one we saw earlier; the scores that form the protein matrices are designed to reflect similarity between the different amino acids rather than simply scoring identities. We will describe some of the available matrices in the next chapter where we will explore the best choice of matrix for various situations.

Over time various mutations occur in sequences; the scoring matrices attempt to cope with mutations, but insertions and deletions require some extra parameters to allow the introduction of gaps in the alignment. There are penalties both for the creation of gaps and for the extension of existing ones; the default gap parameters given in alignment programs have been found to be empirically correct with test sequences but you should experiment with different gap penalties, as we will see.

Global sequence alignment

A global alignment is one that compares the two sequences over their entire lengths, and is appropriate for comparing sequences that are expected to share similarity over the whole length. The alignment maximises regions of similarity and minimises gaps using the scoring matrices and gap parameters provided to the program. The EMBOSS program **needle** is an implementation of the Needleman-Wunsch algorithm[i] for global alignment; the computation is rigorous and **needle** can take a minute or so to run on the remote machines if the sequences are long. First, we'll run **needle** using the default parameters:

Select “needle” from the scroll menu or the “Alignment/Global” menus. Drop & drag first the genomic sequence, and then the cDNA sequence into the “sequence filename” entry fields. Click the “LOAD SEQUENCE ATTRIBUTES” option for both entry fields, to bring up the default values for the program. Now select the “input sequence options” box ONLY for your genomic sequence and specify sequence start and end points of 1bp and 15,000bp respectively.

As this was a global alignment, none of the sequence has been discarded. It is also obvious, that the cDNA is not long enough to line up with the entire genomic sequence, so initially, you will have to scroll down your results to where the alignment starts.

```
Global: HSA1280 vs pax6
Score: 3096.00

HSA1280         1        gatccggagcgacttccgcctatttccagaaattaagctcaaact 45

pax6

HSA1280         46       tgacgtgcagctagttttattttaaagacaaatgtcagagaggct 90

pax6

HSA1280         91       catcatattttccccctcttctatatttggagcttatttattgc 135

pax6
```

This is around 7,500bp, however, this alignment is patchy.

```
HSA1280         7741     cccccagccaagcgccctaaatagcacggagg...cgc.ccgctc 7781
                                                 | ||||    ||  ||||
pax6            1                                cagaggtcaggcttcgcta 19

HSA1280         7782     ttcggacagtgattaatgatagcagagcagagg..gg........ 7816
                          | || ||||||  |  |   |  |||  ||||  ||
pax6            20       atgggccagtgaggagcg...gtggaggcgaggccggcgccgcac 61

HSA1280         7817     .......ttaacacacttca............ctgaaaag...tc 7839
                               |||||||||| |            | | | ||   ||
pax6            62       acacacattaacacacttgagccatcaccaatcagcataggaatc 106
```

Scroll down further to where the alignment begins to look more convincing.

Start of first alignment region (base pair) .....

How does this compare to the information in Ensembl and the EMBL entry for the pax6 gene? Scroll down further and count the number of regions of identity.

dottup and **dotmatcher** showed us that there were nine regions of identity in this gene, and **needle** seems to only have found four! The reason for this lies in the default gap parameters used in **needle** - the cost for creating and extending gaps is rather high and since all the program is doing is arithmetically producing the highest score it can, it sometimes turns out that the alignment it gives you is not very sensible. The gap creation penalty is the amount that is subtracted from the score for initially opening a gap, and the gap extension penalty is the cost for extending the length of a gap by one position. In this case, **needle** has found that adding in the odd stretch of 4 aligned bases here, 5 aligned bases there will give a higher score than incorporating longer gaps and aligning the exons correctly.

Run **needle** again, this time choosing a value of 0.1 for the gap extension penalty, but keeping the gap creation penalty at 10

Scroll down the results and count the number of closely matching regions. Are all nine exons displayed this time? Do the alignments start and finish at the same positions as those already documented in EMBL and Ensembl?

If you compare the two alignments, you'll notice that the first one picked out only exons 4, 5, 6, and 7, and matched the remainder of the cDNA to the genomic sequence by sporadically aligning runs of nucleotides. This makes more sense if we consider the size of the introns by looking at the second alignment - between exons 3 and 4, there is an intron that is around 6kb - at a cost of 0.5 per extended position, that subtracts a great deal from the score and matching the 900 nucleotides of the cDNA in short runs turns out to be "cheaper" for the algorithm. The introns between exons 4 and 5, 5 and 6, and 6 and 7 are only 500, 150 and 50 nucleotides respectively so do not reduce the score so much. The intron following exon 7 is 2500 nucleotides, so again a better score is achieved by matching small runs of the remaining 250 nucleotides rather than inserting a large gap and aligning the exons "properly".

This illustrates a valuable point - whenever you are running bioinformatics programs, it's unwise to blindly accept the default parameters and expect the programs to always give sensible results. As with wet biology, you should always think carefully about the results you have achieved and decide whether they make sense.

Local sequence alignment

As we mentioned above, global sequence alignment algorithms align sequences over their entire lengths. You do need to think about whether that type of alignment makes sense for your sequences. For our example, where we expect each exon to be represented in the sequences and in the same order, it has worked well - however, how well do you think this approach would work with, for example, multidomain

proteins that share one domain but not others, or sequences where there have been regions of duplication? A second comparison method, local alignment, searches for regions of local similarity and need not include the entire length of the sequences.
The EMBOSS program **matcher** is a rigorous local alignment program, based on the "lalign"[ii] program written by which is an implementation of an algorithm described by M.S. Waterman and M. Eggert[iii].

Select "matcher" from the scroll menu or the "Alignment/Local" menus (or Best Local Alignments from favourites menu). Drag & drop the genomic and cDNA sequences into the sequence input fields and click the "LOAD SEQUENCE ATTRIBUTES" button to obtain the default values. Hit "Go".

This time the alignment starts immediately. Remember, this is a local alignment, so the unmatching ends are thrown away.

```
           2430         2440      2450      2460      2470
HSA128 CACTTCCCCTAT---GCAGGTGTCCAACGGATGTGTGAGTAAAATTCTGGGCAGGTATTA
       :: :::::  ::   ::::::::::::::::::::::::::::::::::::::::::::
  pax6 CATTTCCCGAATTCTGCAGGTGTCCAACGGATGTGTGAGTAAAATTCTGGGCAGGTATTA
     540       550       560       570       580       590

     2480      2490      2500      2510      2520      2530
HSA128 CGAGACTGGCTCCATCAGACCCAGGGCAATCGGTGGTAGTAAACCGAGAGTAGCGACTCC
       ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  pax6 CGAGACTGGCTCCATCAGACCCAGGGCAATCGGTGGTAGTAAACCGAGAGTAGCGACTCC
     600       610       620       630       640       650

     2540      2550      2560      2570      2580      2590
HSA128 AGAAGTTGTAAGCAAAATAGCCCAGTATAAGCGGGAGTGCCCGTCCATCTTTGCTTGGGA
       ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
  pax6 AGAAGTTGTAAGCAAAATAGCCCAGTATAAGCGGGAGTGCCCGTCCATCTTTGCTTGGGA
     660       670       680       690       700       710
```

And this, unfortunately is the only matching region displayed. Not very much aligned, you might think, and certainly nowhere near the nine exons we saw using the dotplots and, eventually, **needle**. At first you might think that the default gap penalties used by this program must be incorrect for our alignment. This alignment does, however, correspond to the alignment of the second exon found using needle. But! We are also using a program that has been designed to give the best alignments – the default number being one, so if it had found all nine exons, we wouldn't know unless we specified an output of at least nine alignments – or worked out the gap penalties in such a way, that the inter-exon gaps were less expensive than the exon alignments, thus aligning the region containing all nine exons.

Re-run matcher, but this time, specify ten alignments in the "number of alternative matches" field. Change also the "gap penalty" to 20. Leave the "gap length penalty" at 4 and hit "Go".

Again, scan backwards and forwards to find the exons. This time, you should see that the nine exons have been correctly predicted. They have not, however, been predicted in the correct order. Why[38]?

[38] Remember, that this program gives you the best alignments first, and the "best alignment" for the computer involves not only percentage identity, but length of alignment too.

You should be aware that local alignment methods only report the single best match between two sequences - there may be a large number of alternative local alignments that do not score as highly. If two proteins share more than one common region, for example one has a single copy of a particular domain while the other has two copies, it may therefore be possible to "miss" the second and subsequent alignments. You'll be tipped off to this type of occurrence if you have done a dotplot and your local alignment does not show all the features you expected to see, as in this case. It can be sensible to restrict the search to defined regions of the sequence - we'll see how to tell EMBOSS programs to use only a portion of a sequence later on in this chapter.

If you have some extra time, why not try rerunning **needle** and/or **matcher** with different gap penalties - the numbers we have used here won't always work, and you may have to experiment, as we did when designing this exercise.

EMBOSS contains other pairwise alignment programs - **stretcher** and **water** are global and local alignment programs. Stretcher is less rigorous than **needle** and **water** slightly more rigorous than **matcher**. Less rigorous algorithms run more quickly; and they may be useful for database searching. **supermatcher** is designed for local alignments of very large sequences and is even less rigorous in its implementation. If you really want to align a cDNA sequence to a genomic sequence, as we did here, you can also use **Est2Genome**. The documentation pages for all these programs can be found at: http://emboss.sourceforge.net

ORF Identification and Translation

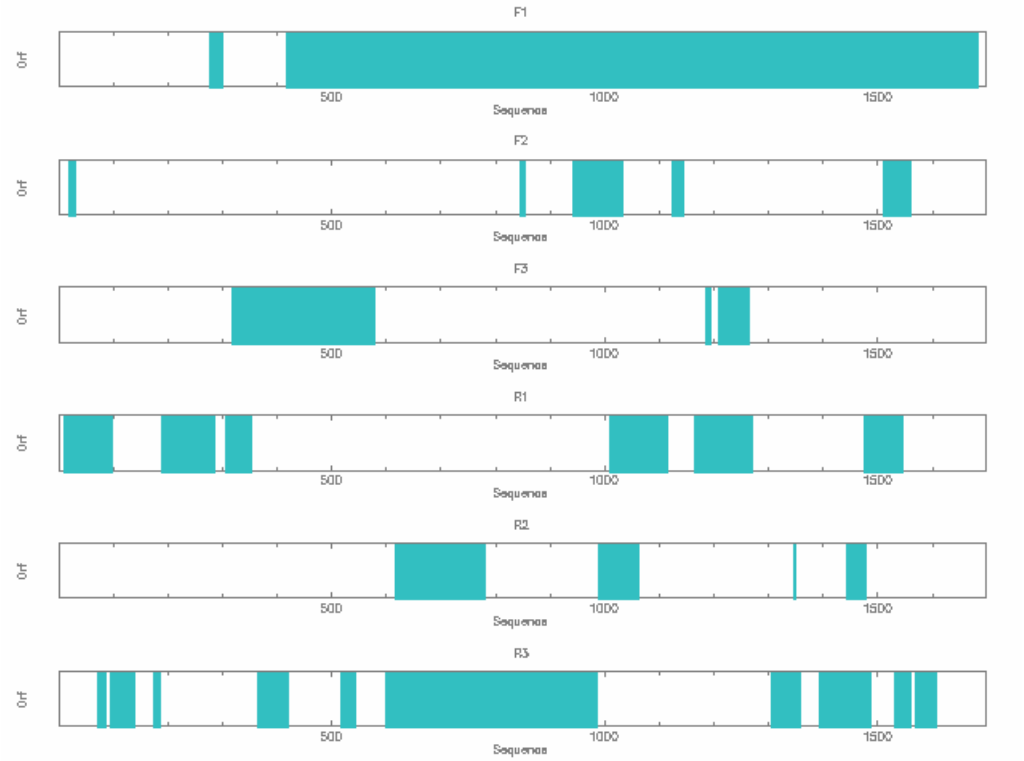
Finally in this section we'll show you some EMBOSS applications for translating your cDNA sequence into protein. Later on today we'll see some methods for predicting exon/intron boundaries with our genomic sequence and hopefully you'll start to realise that gene structure prediction is a tough problem.

First, we need to identify our open reading frame. We can get a rapid visual overview of the distribution of ORFs in the six frames of our sequence using the EMBOSS program **plotorf.**

Select "plotorf" from the scroll menu, or the "Nucleic/Gene Finding" menus and insert your cDNA sequence into the "Sequence Filename" field. This is a human sequence, so leave the start and stop codons as the default entries. Save as **pax6_cDNA.png**[39] and close the output window.

You will see a graphical output that shows the potential open reading frames (ORF) in all six-frames. You'll notice that the longest ORF occurs in F1 (*i.e.* forward sense, frame 1) starting at around 400 bases and ending around 1700.

[39] You must save your graphical files with a .png extension, otherwise you will not be able to open and view them as graphics.



Now we have a feel for the area we're expecting to need to translate, we need to know the exact start and end points for our translation. To do this, we can use the EMBOSS program **getorf**.

Select "getorf" from the scroll menu, or the "Nucleic/Gene Finding" menus and insert your cDNA sequence into the "Sequence Filename" field. Ensure that the " code to use" box had "standard" selected Use you results from **plotorf** to select a minimum size for translation of ORFs[40]. Select the "Translation of regions between START and STOP codons" option for the "type of output" and hit "Go".

plotorf is just a graphical representation of the textual information produced by **getorf**. Since we asked for all ORFs above a minimum size to be reported, **getorf** is telling us about a number of potential ORFs. We know from **plotorf** that our ORF will be in the region 400 to 1700, so scroll through the output until you identify this. What are the actual start and end positions?

Close the output window.

From this you should have found that the translation is from 418 to 1683. The next step is to translate this region into the protein sequence. The EMBOSS program that translates nucleotide sequences is called **transeq**[41]**.**

Select "transeq" from the scroll menu, or the "Nucleic/Translation" menus and insert your cDNA sequence into the "Sequence Filename"

[40] You do not want to omit all potential ORFs, so a size in the region of 150 will allow you to examine several potential proteins.

[41] This program may not be necessary if you have already obtained the correct protein by using "getorf".

field. Select "Input Sequence Options" Enter 418 and 1683 into the "begin" and "end" input fields respectively. Select "nucleotide" and click "OK". We know that our ORF is in frame 1, so enter **1** into the "Frames to Translate" field. Ensure that the "Code to use" is set to "standard" and hit "Go". Edit the comment line in the sequence to read **Pax6 conceptual translation from mutant cDNA** and save the output as **pax6.pep**. Close the output window.

```
>pax6 conceptual translation from mutant cDNA
MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELPHSGARPCDISRILQVSNGCVSKILGRY
YETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVCTNDNIPSV
SSINRVLRNLASEKQQMGADGMYDKLRMLNGQTGSWGTRPGWYPGTSVPGQPTQDGCQQQ
EGGGENTNSISSNGEDSDEAQMRLQLKRKLQRNRTSFTQEQIEALEKEFERTHYPDVFAR
ERLAAKIDLPEARIQVWFSNRRAKWRREEKLRNQRRQASNTPSHIPISSSFSTSVYQPIP
QPTTPVSSFTSGSMLGRTDTALTNTYSALPPMPSFTMANNLPMQPPVPSQTSSYSCMLPT
SPSVNGRSYDTYTPPHMQTHMNSQPMGTSGTTSTGLISPGVSVPVQVPGSEPDMSQYWPR
LQ
```

You are now in a position to compare your sequence with the official database sequence of pax6. This is best done using a sequence alignment, where you can instantly see any discrepancies between sequences.

Align your translated pax6.pep sequence with the database entry. Use the SwissProt accession number that you found whilst searching SRS. Remember the "Input Sequence Options" for retrieving a sequence from a database!

You should note that there is a single amino acid difference between the two sequences.

Amino acid difference ……

Residue at which difference occurs ……

There is another program in EMBOSS which will identify single point mutations. **diffseq** uses the sliding window methodology to align sequences and note differences between them.

Select "diffseq" from the scroll menu or the "Alignment/Differences" menus. Input your pax6.pep sequence in the first "Sequence Filename " box, and choose the "Input Sequence Options" for the second sequence. Choose the database "swissprot" from the "Databases available" menu. You should see that **swissprot:** has been filled in the "Sequence Filename" box. Select "fasta" format from the "Sequence Format". Select the "protein" option and click on "OK". Return to the "Sequence Filename" entry field and type in **p26367**. Leave the word size as 10 and hit "Go". Select the "pax6.diffseq" tab on the output window:

Does the A → P substitution at residue 33 agree with the information you obtained by aligning the protein sequences? If it doesn't, it probably means something has gone wrong somewhere in the process. Check back to see if you can identify what it is[42]

[42] It is always worthwhile checking your analysis at each stage, to ensure that you are still on the right track.

Close the results output window.

Restriction Maps

If you look closely at the alignment produced by **needle** or **matcher** you can see that the alignment is not perfect.

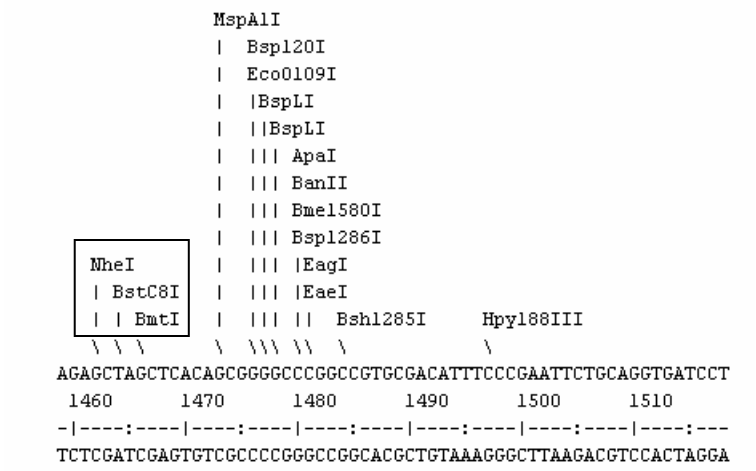
Select “Saved Results” from the “File” menu of the Jemboss window. Select your final “needle” (or matcher) analysis and click “display” to show you the results once again.

Restriction mapping and PCR primer design can be created using computational technology. First we will map a region between the 1459 and 1503 base pair regions**.**

As you know, several hundred restriction enzymes (RE) have been isolated (mostly from bacteria) and their nucleotide recognition sites characterised. The motifs recognised by RE are mostly palindromes and vary in length from DNA tetramers (frequent cutters) to 26-mers (BaeI). We will use the EMBOSS program **remap** to make a restriction map of each sequence in the area where the difference is seen in the alignment. Comparison of the restriction map of the cDNA clone with the map of the genomic fragment will indicate whether RE digestion would be suitable for investigating the discrepancy between the two sequences.

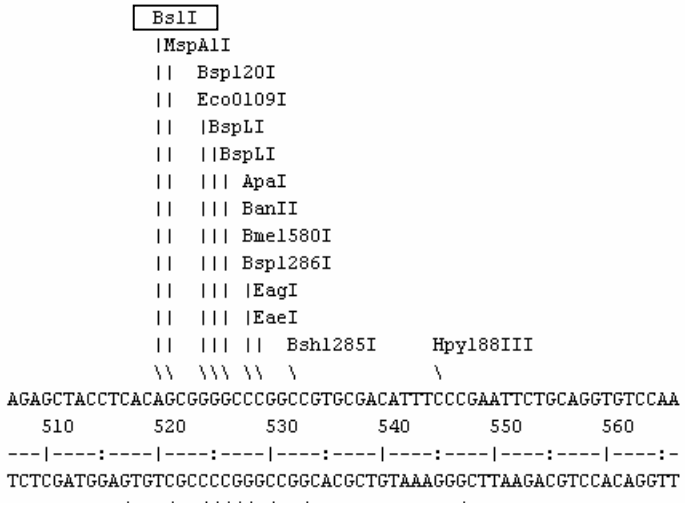
Select ”remap” from the scroll menu or the “Nucleic/Restriction” menus and drag the pax6_genomic.fasta file into the correct field. Select “Input Sequence Options” and enter 1459 and 1503 into the “begin” and “end” fields respectively. Leave the “enzyme list” as **all** and set the “Minimum recognition site length” to **6**. You already know the translation, so de-select “Display translation” and “Display cut sites and translation of reverse sense”. Leave the ORF size as it is, and change the “Code to use” to **standard**. Let everything else remain as default and hit “Go”.

The person who blindly plods on will end up wasting a lot of time later, when designing laboratory experiments to corroborate the computational evidence.



Re-run **remap** for the corresponding piece of cDNA (positions 507-551), using the same parameters.

So, as you can see, the changed sequence removes three restriction sites and creates another one (boxed). We can find more information about these restriction enzymes, such as their recognition sites and where they can be bought from the REBASE web site. REBASE is a publicly available database of restriction enzyme data.



If the restriction enzymes mentioned here differ slightly from your results, check them in the isoschizomer list below the graphic.

Go to http://rebase.neb.com/rebase/rebase.html and type the name of an enzyme into the “or go directly to enzyme” search box and hit the button marked "**Go**". You'll be shown a table of information about the enzyme, including links to isoschizomers and suppliers.

If you want to search for specific restriction enzymes on your own sequences, you can narrow down the search using remap, either by restricting the type of cutters you wish to find, or restricting the enzymes the program searches for.

Primer Design

Another approach to screening the discrepancy would take advantage of DNA amplification using the polymerase chain reaction. Designing oligos for PCR can be done by hand, but experience shows that you can save time, aggravation & funds by using a computer to avoid bad primers. Bad primer pairs have melting temperatures that are too different, are not specific enough to your target, hybridise together or form internal hairpin loops, amongst other things.

One further advantage of using bioinformatics to help in designing PCR primers, is that algorithms usually use more accurate methods for T_m calculations than the rule of thumb used at the bench: (A+T) = 2 and (G+C) = 4 degrees centigrade.

We want to design primers to amplify the region of our cDNA centred around position 514, where our potential mutation lies. There are several programs in EMBOSS that you can use for primer design, but the one with the most options is called **eprimer3**. This was written[43] at the Massachusetts Institute of Technology (MIT) in the United States, and has been adapted for use with our package. One of the variables used for the calculation of melting and annealing temperatures of the primer pairs is their CG content.

It is sensible to use the genomic sequence to amplify the region containing the suspected mutation, so review the results obtained with matcher, and identify the region in which the mutation would be found on the genomic sequence.

Select “eprimer3” from the scroll menu or the “Nucleic/Primer” menus. Drag & drop your **pax6_genomic.fasta** file into the “Sequence Filename” box and click on the “LOAD SEQUENCE ATTRIBUTES” button. Click on the “advanced options” button. Specify a “Target Region” of between 1450 and 1480 by typing **1450,1480** in the appropriate field. Leave the other parameters in their default settings.

You will have noticed that the advanced options on this program involve many parameters. Certainly more than we need for this course, and possibly more than you will need for your analysis. The best idea is to go through the parameters yourself, and decide which ones are necessary for your particular requirements.

[43] If you prefer to use the web version, see http://www-genome.wi.mit.edu/cgi-bin/primer/primer3_www.cgi

```
EPRIMER3 RESULTS FOR HSA1280

                    Start  Len   Tm     GC%   Sequence

 1 PRODUCT SIZE: 200
   FORWARD PRIMER    1378   20  60.27  55.00  AGGTCACAGCGGAGTGAATC

   REVERSE PRIMER    1558   20  59.95  50.00  ATGAAGAGAGGGCGTTGAGA


 2 PRODUCT SIZE: 202
   FORWARD PRIMER    1378   20  60.27  55.00  AGGTCACAGCGGAGTGAATC

   REVERSE PRIMER    1560   20  59.81  45.00  AAATGAAGAGAGGGCGTTGA


 3 PRODUCT SIZE: 203
   FORWARD PRIMER    1378   20  60.27  55.00  AGGTCACAGCGGAGTGAATC

   REVERSE PRIMER    1561   20  59.81  50.00  GAAATGAAGAGAGGGCGTTG


 4 PRODUCT SIZE: 198
   FORWARD PRIMER    1380   20  59.42  55.00  GTCACAGCGGAGTGAATCAG

   REVERSE PRIMER    1558   20  59.95  50.00  ATGAAGAGAGGGCGTTGAGA


 5 PRODUCT SIZE: 200
   FORWARD PRIMER    1380   20  59.42  55.00  GTCACAGCGGAGTGAATCAG

   REVERSE PRIMER    1560   20  59.81  45.00  AAATGAAGAGAGGGCGTTGA
```

The results commence with a summary of the variables entered. Below this summary, each primer pair is defined. The forward primer represents a sequence complimentary to the reverse strand, and will create copies of the forward strand when extended. Reverse primers are complimentary to the forward strand, and create copies of the reverse strand. Position numbering is relative to the forward strand for both primers. The melting temperature of the primer is displayed, together with the GC content (in brackets). Below this, is the length of the primer, and further down the annealing temperature of each primer

The GC content, melting temperature and length of the product is also included in the final results.

Save the output file as **pax6_genomic.prima**, and close the output window.

Should you wish to check whether your designed primer would anneal anywhere else in the genome, you may use **fuzznuc**, which will return all other occurrences of a single nucleotide series.

Select “fuzznuc” from the scroll menu, or from “Nucleic/Motifs” menus. Select the pax6_genomic.fasta sequence to enter into the “Sequence Filename” field. Double click on **pax6_genomic.prima** in your file

manager and highlight the forward primer. Copy it using ^C[44]. Return to the Jemboss window and insert the primer sequence you have just copied by clicking with your mouse once in the “Search Pattern” field and pressing ^V[45]. Alter the number of mismatches to two (2) and press "Go".

Only one region should match in your genomic sequence, and it should be the same position as your forward primer.

Re-run **fuzznuc** using the reverse primer. This time you must remember to also click on "Advanced Options" and "search complimentary strand".

Again, only one sequence is found, suggesting your primer will not anneal in several positions. To be extra cautious, you may wish to try this again specifying a greater number of mismatches.

If you have designed a primer yourself, and wish to check the melting temperature, there is an EMBOSS program called **dan** to do just that.

[i] Needleman, S. B. and Wunsch, C. D. (1970) J. Mol. Biol., 48, 443.
[ii] Huang X. and Miller W. (1991) Adv. Appl. Math. 12 337-357

[iii] Waterman M.S. and Eggert M. (1987) J. Mol Biol 197 723-728

[44] Hold down the control key as you press the “c” key.
[45] Hold down the control key as you press the “v” key. If you using a unix system, you need not bother with this, and can cut just by highlighting the sequence with your left mouse button, and paste using the middle mouse button.

Advanced Sequence Analysis

The aim of this module is to explore some of the more complex bioinformatics tools for sequence analysis and to give some hints on how to evaluate the results from these analyses. The approaches we'll consider now are more demanding on the user, the computer hardware, or both, and include database mining software, an integrated comprehensive interface to general DNA analysis and the use of gene prediction software.

Sequence Alignment Scores

Most computer programs for sequence alignments try to answer the question "Which alignment will give me the highest score(s)?" Some sequence alignment algorithms adopt a rigorous approach (e.g. the Smith-Waterman algorithm as implemented by EMBOSS' **water**) assessing every possible alignment and reporting the most statistically likely, whilst others, particularly those used for database searching, use simpler approaches to improve speed.

The scores are better interpreted as statistics and we will come onto that later.

Scoring Matrices

Scoring matrices are used internally by all sequence alignment programs. The simplest form of scoring matrix is an identity matrix where a score is assigned only to a pairwise comparison between identical residues. This is the approach taken with many nucleic acid scoring matrices as we saw in the previous chapter; the concept may be extended to include comparisons for IUB nucleotide ambiguity codes. However, amino acid matrices are central to polypeptide sequence comparisons, containing scores for every one of the 210 possible pairs of amino acids (i.e. 190 pairs of different amino acids + 20 pairs of identical amino acids).

Since the first protein sequences were obtained, many different types of scoring scheme have been devised. Understanding how these are built and how they are used can help in making sensible use of sequence alignment software and database mining tools.

GENETIC CODE SCORING

Genetic code scoring as introduced by Fitch[i] considers the minimum number of DNA/RNA base changes (0, 1, 2 or 3) that would be required to inter-convert the codons for any two amino acids. The scheme has been used both in the construction of phylogenetic trees and in the determination of homology between protein sequences having similar three-dimensional structures[ii].

Chemical similarity scoring

Chemical similarity scoring schemes give greater weight to the alignment of amino acids with similar physico-chemical properties, since major changes in amino acid type could reduce the ability of a protein to perform its biological role and hence cause it to be selected against during the course of evolution. McLachlan[iii] developed a scoring scheme that classifies amino acids on the basis of polar or non-polar character, size, shape and charge. Feng et al.[iv] extended the scheme to include considerations of the redundancy of the genetic code.

Observed substitutions – Dayhoff's matrix

These schemes use multiple sequence alignments to assess the frequency with which amino acids substitute for one another. Possibly the most widely used scheme for scoring amino acid pairs is that developed by Dayhoff and co-workers[v] who examined global alignments of closely similar sequences, observing the frequency with which amino acid substitutions occurred. A complete picture of the mutation process including those amino acids which were not observed to change was determined by calculating the average ratio of the number of changes a particular amino acid type underwent to the total number of amino acids of that type present in the database. This was combined with the point mutation data to give a mutation probability matrix. This matrix is specific for a particular evolutionary distance, but may be used to generate matrices for greater evolutionary distances by multiplying it repeatedly by itself. The mutation data matrix is based on the idea of the Point Accepted Mutation (PAM); 1 PAM represents an evolutionary distance in which 1% of the amino acids have been changed. This doesn't mean that 100PAM represents a 100% change in amino acids composition, as we need to remember that changes can revert, and certain positions may undergo more than one substitution.

There is a series of PAM matrices, each appropriate for use with sequences separated by different evolutionary distances; the higher numbered PAM matrices are appropriate for sequences that are more evolutionarily divergent than the lower PAM matrices. The most commonly used PAM matrix is PAM250; at an evolutionary distance of 256PAM approximately 80% of the amino acid positions are observed to have changed[vi], though it should be noted that the amino acids vary in their mutability. At this distance, 48% of the tryptophans, 41% of the cysteines and 20% of the histidines are unchanged, but only 7% of serines would remain.

PAM250 scoring matrix:

	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Z
A	**2**	0	-2	0	0	-4	1	-1	-1	-1	-2	-1	0	1	0	-2	1	1	0	-6	-3	0
B	0	**2**	-4	3	2	-5	0	1	-2	1	-3	-2	2	-1	1	-1	0	0	-2	-5	-3	2
C	-2	-4	**12**	-5	-5	-4	-3	-3	-2	-5	-6	-5	-4	-3	-5	-4	0	-2	-2	-8	0	-5
D	0	3	-5	**4**	3	-6	1	1	-2	0	-4	-3	2	-1	2	-1	0	0	-2	-7	-4	3
E	0	2	-5	3	**4**	-5	0	1	-2	0	-3	-2	1	-1	2	-1	0	0	-2	-7	-4	3
F	-4	-5	-4	-6	-5	**9**	-5	-2	1	-5	2	0	-4	-5	-5	-4	-3	-3	-1	0	7	-5
G	1	0	-3	1	0	-5	**5**	-2	-3	-2	-4	-3	0	-1	-1	-3	1	0	-1	-7	-5	-1
H	-1	1	-3	1	1	-2	-2	**6**	-2	0	-2	-2	2	0	3	2	-1	-1	-2	-3	0	2
I	-1	-2	-2	-2	-2	1	-3	-2	**5**	-2	2	2	-2	-2	-2	-2	-1	0	4	-5	-1	-2
K	-1	1	-5	0	0	-5	-2	0	-2	**5**	-3	0	1	-1	1	3	0	0	-2	-3	-4	0
L	-2	-3	-6	-4	-3	2	-4	-2	2	-3	**6**	4	-3	-3	-2	-3	-3	-2	2	-2	-1	-3
M	-1	-2	-5	-3	-2	0	-3	-2	2	0	4	**6**	-2	-2	-1	0	-2	-1	2	-4	-2	-2
N	0	2	-4	2	1	-4	0	2	-2	1	-3	-2	**2**	-1	1	0	1	0	-2	-4	-2	1
P	1	-1	-3	-1	-1	-5	-1	0	-2	-1	-3	-2	-1	**6**	0	0	1	0	-1	-6	-5	0
Q	0	1	-5	2	2	-5	-1	3	-2	1	-2	-1	1	0	**4**	1	-1	-1	-2	-5	-4	3
R	-2	-1	-4	-1	-1	-4	-3	2	-2	3	-3	0	0	0	1	**6**	0	-1	-2	2	-4	0
S	1	0	0	0	0	-3	1	-1	-1	0	-3	-2	1	1	-1	0	**2**	1	-1	-2	-3	0
T	1	0	-2	0	0	-3	0	-1	0	0	-2	-1	0	0	-1	-1	1	**3**	0	-5	-3	-1
V	0	-2	-2	-2	-2	-1	-1	-2	4	-2	2	2	-2	-1	-2	-2	-1	0	**4**	-6	-2	-2
W	-6	-5	-8	-7	-7	0	-7	-3	-5	-3	-2	-4	-4	-6	-5	2	-2	-5	-6	**17**	0	-6
Y	-3	-3	0	-4	-4	7	-5	0	-1	-4	-1	-2	-2	-5	-4	-4	-3	-3	-2	0	**10**	-4
Z	0	2	-5	3	3	-5	-1	2	-2	0	-3	-2	1	0	3	0	0	-1	-2	-6	-4	**3**

OBSERVED SUBSTITUTION - BLOSUM MATRICES

Dayhoff-like matrices derive their initial substitution frequencies from global alignments of very similar sequences. Henikoff and Henikoff[vii] took the approach of analysing multiple local alignments of more distantly related sequences so that the substitutions being considered are those occurring in relatively conserved regions of the proteins rather than over the entire length of the sequence. Local alignments of related sequences were used to populate a database, and within alignment, the sequences were clustered into groups where the sequences are similar at a certain threshold value of percentage identity. Each group was then used to calculate substitution frequencies for all pairs of amino acids and this used to calculate a BLOSUM (blocks substitution matrix) matrix. Different matrices are obtained by varying the clustering threshold, for example, the BLOSUM 80 matrix was derived using a threshold of 80% identity, BLOSUM 62 has a threshold value of 62% and so on. Thus, lower numbered BLOSUM matrices (e.g. BLOSUM30) are appropriate for sequences that are highly evolutionarily divergent, while higher BLOSUM matrices (e.g. BLOSUM90) are suitable for less divergent sequences.

MATRICES DERIVED FROM TERTIARY STRUCTURE ALIGNMENTS

In future, we will probably use matrices derived from alignments based on structural

considerations, as distantly related proteins may be aligned more accurately. The most reliable protein sequence alignments may be obtained when all the proteins have had their tertiary structures experimentally determined. Some structure derived matrices have been produced; for example, Risler et al.[viii] derived substitution frequencies from 32 proteins structurally aligned in 11 groups, and Overington et al.[ix] aligned 7 families for which 3 or more proteins of known three dimensional structure were known and derived a series of substitution matrices. Bowie et al.[x] have also derived substitution tables specific for different amino acid environments and secondary structures.

WHICH MATRIX SHOULD I USE?

Ideally, when you are comparing two sequences, either as a single pairwise comparison or as a database sequence similarity search, you should have some idea of the evolutionary distance divergence of the two sequences so that you can use an appropriate scoring matrix. Obviously for very similar sequences the matrix used would matter less to find alignments, than for when you want to do very sensitive database searches. When comparing similar sequences, scoring based on identity could be sufficient. There is a trade off between sensitivity, computational time and your personal result analysis time.

The general consensus is that the PAM and BLOSUM substitution matrices are superior to identity, genetic code or physical property matrices. However, there are Dayhoff matrices of different PAM values and BLOSUM matrices of different percentage identity - which of these should you use for a particular application?

Various groups have studied this question:

- Schwartz and Dayhoff[xixii,] recommended PAM250 for pairwise comparisons of proteins known to be distantly related. This matrix gave a consistently higher significance score than other matrices in the range 0-750 PAM. The scores were also better than those[xiii] achieved with chemical similarity, the genetic code or identity matrices.
- Altschul[xiv] also concluded that a matrix of 200 PAMS is most appropriate when the sequences to be compared are thought to be related. However, when you search a database with a sequence, you do not know in advance what the relationship between the sequences will be, i.e. whether or not they are homologous. In this situation, he concluded that a 120 PAM matrix was the best compromise. Further, he suggested that three matrices be used when performing local alignments: PAM40, PAM120 and PAM250. The lower PAM matrices will tend to find short alignments of highly similar sequences, while higher PAM matrices will find longer, weaker local alignments.
- Collins and Coulson[xv] also advocate the compromise PAM100 matrix, and support the use of multiple PAM matrices to allow detection of local similarities of all types.
- Henikoff and Henikoff[xvi] compared the BLOSUM matrices to various others by evaluating how effectively they detect known members of a protein family from a **BLAST** search of a sequence. They conclude that overall the BLOSUM 62 matrix is the most effective. However, other substitution matrices investigated perform better than BLOSUM 62 for a proportion of the families.

The overall conclusion is that no single matrix is the complete answer for all sequence comparisons. As more protein three-dimensional structures are determined,

substitution tables derived from structure comparison will probably give the most reliable data. A good strategy may be to use one matrix to find ``similar'' sequences and produce an initial alignment, then view the results and do more rigorous pairwise alignments with more **appropriate** matrices.

Alignments will also depend upon the sensitivity of the algorithm you have used, the gene model if chosen and parameters. You should always use your biological common sense when analysing computational results.

Much of the above information was adapted from Geoff Barton's chapter in Protein Structure Prediction – A Practical Approach[xvii].

Database mining

SEARCHING FOR SEQUENCE SIMILARITIES IN DATABASES

The growing size and diversity of the public sequence databases makes them invaluable resources for molecular biologists. When investigating a novel DNA sequence, a fast, cheap and potentially very rewarding analysis involves scanning EMBL, or GenBank, for sequences with homology to your own sequence. Database searching is one of the first and most important steps in analysing a new sequence. If your unknown sequence has a similar copy already in the databases, a search will quickly reveal this fact and if the copy is well annotated you will have various clues to help you in further studying your sequence. Database searches usually provide the first clues of whether the sequence belongs to an already studied and well known protein family. If there is a similarity to a sequence that is from another species, then they may be homologous (i.e. sequences that descended from a common ancestral sequence). Knowing the function of a similar/homologous sequence will often give a good indication of the identity of the unknown sequence[46].

Many programs for database searching already exist, but still many more are being developed. We'll talk about **BLAST** today but there are others that you should be aware of and that you might want to look at if time allows:

> **BLAST** (Basic Local Alignment Search Tool) performs fast database searching combined with rigorous statistics for judging the significance of matches.
>
> **FASTA** can be used to compare either protein or DNA sequences and hence the name, which stands for Fast-All.
>
> **BLITZ** is an automatic electronic mail server for the **MPsrch** program. **MPsrch** allows you to perform sensitive and extremely fast comparisons of your protein sequences against Swiss-Prot protein sequence database using the Smith and Waterman best local similarity algorithm.

[46] You should bear in mind that in order to identify homologous sequences, searches involving the protein sequence are approximately 5 times more sensitive at finding matches

BLAST

BLAST (Basic Local Alignment Search Tool) is a heuristic method to find the highest scoring locally optimal alignments between a query sequence and a database. A gapped **BLAST** search allows gaps (deletions and insertions) to be introduced into the alignments that are returned. Allowing gaps means that similar regions are not broken into several segments. The scoring of these gapped alignments tends to reflect biological relationships more closely.

The **BLAST** algorithm and family of programs rely on work on the statistics of local sequence alignments by Altschul *et al*[xviii]. The statistics allow us to estimate the probability of obtaining an alignment with a particular score. The **BLAST** algorithm permits nearly all sequence matches above a cutoff[47] to be located efficiently in a database.

The algorithm operates as follows:

1. **BLAST** scans the database for *words* (typically 3-mers for proteins) that score at least *T* (a designated threshold value) when aligned with a word in the query sequence – such aligned pairs are called *hits*.
2. If a second non-overlapping hit is found within a distance *A* of the first and on the same diagonal, the first hit is extended between the database and query sequences in both directions. Extension continues, scoring all the time, until the running score drops below the maximum score seen so far by a value *X*. The resulting local alignment is called an *HSP* (high-scoring segment pair) or *MSP* (maximum scoring segment pair).
3. If the alignment score of the HSP exceeds a given value Sg (the gapped score), then a gapped extension of the HSP is initiated.

Earlier versions of **BLAST** looked only for single hits and extended them all; however, the extensions did not incorporate gaps and thus missed some potentially interesting matches. The gapped extension currently used, takes much longer to execute, but speed is improved overall by the requirement for two non-overlapping close hits before the initial extension is triggered, and the value of Sg is chosen so that only about one extension is triggered per 50 database sequences.

These modifications to **BLAST** mean that it now runs three times faster than earlier versions and in trials it found more statistically significant alignments than the old **BLAST**.

BLAST FAMILY OF PROGRAMS

The **BLAST** family of programs allows all combinations of DNA or protein query sequences with searches against DNA or protein databases. (Most of the time use of these is behind an interface.)

- **blastp**: compares an amino acid query sequence against a protein sequence

[47] This cutoff is usually generated by the BLAST program, based on the parameters you have selected.

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 (both strands) against a protein sequence database.
- **tblastn**: compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands).
- **tblastx**: compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database.
- **PSI-Blast**: Position-Specific Iterated **BLAST**. This is potentially a very sensitive method to pull out significant hits in a protein-protein database search. This first performs a gapped BLAST database search and then uses the information from any significant alignments returned to construct a position-specific score matrix, which replaces the query sequence for the next round of database searching. **PSI-Blast** may be iterated until no new significant alignments are found. We'll look at this tomorrow when we do some protein analysis.

For more information on the **BLAST** algorithm, read our **BLAST** FAQ, or look at the NCBI WWW site: http://www.ncbi.nlm.nih.gov/BLAST/

Go to the NCBI homepage at http://ncbi.nlm.nih.gov and select the BLAST option on the top menu. Click on the link for "Nucleotide-nucleotide BLAST (blastn)".

Double click on your **pax6_cDNA.fasta** sequence in the Jemboss file manager, and copy the contents of the file[48]. Paste this sequence into the BLAST web page.

Leave the database as "nr" – this is a non-redundant set of all nucleotide sequence that NCBI holds. Leave all other options the same and hit the BLAST! button.

A page will appear telling you your results have been delivered into the BLAST queue and an estimate of how long it should take to retrieve results.

Hit the Format! Button and a separate window will appear. If your results are ready, they will be displayed in this window. If they need a little longer, then you will be told and the window will automatically refresh at regular intervals until your results are ready.

The output of a search is broken down into several sections.
The following results do not represent those which should you should see after the exercise, but **merely an example** to explain the various sections.

Header

The **BLAST** report starts with some header information that lists the type of program (here **BLASTX**), the version (here 2.2.1), and a release date. Also listed is a reference

[48] It is not critical if you copy the sequence description in addition to the sequence itself.

to the **BLAST** program, the query definition line, and summary of the database used.

```
BLASTX 2.2.1 [Apr-13-2001]

Reference:
Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

RID: 1004974810-22070-30849

Query=
         (22,253 letters)

Database: Non-redundant SwissProt sequences
           100,395 sequences; 36,513,100 total letters
```

Summary

One-line descriptions of the database matches found are presented next. These include a database sequence identifier, the corresponding definition line, as well as the score (in bits) and the statistical significance ('E value') for this match (please see the section on statistics for an explanation of bits and significance).

Sequences producing significant alignments:		Score (bits)	E Value
gi|417450|sp|P32117|PAX6_MOUSE	PAIRED BOX PROTEIN PAX-6 (OC...	723	0.0
gi|2495315|sp|P55864|PAX6_XENLA	PAIRED BOX PROTEIN PAX-6	705	0.0
gi|1352720|sp|P47238|PAX6_COTJA	PAIRED BOX PROTEIN PAX-6 (P...	702	0.0
gi|129651|sp|P26630|PAX6_BRARE	PAIRED BOX PROTEIN PAX[ZF-A]...	691	0.0
gi|3914281|sp|O73917|PAX6_ORYLA	PAIRED BOX PROTEIN PAX-6	678	0.0
gi|1352719|sp|P47237|PAX6_CHICK	PAIRED BOX PROTEIN PAX-6	330	5e-90
gi|12643549|sp|O18381|PAX6_DROME	PAIRED BOX PROTEIN PAX-6 (...	252	1e-66
gi|3914276|sp|O43316|PAX4_HUMAN	PAIRED BOX PROTEIN PAX-4	244	2e-64

The identifiers shown here are all from SwissProt, so they all have 'sp' in the first field, followed by the accession, and then a Locus name. The syntax of these identifiers is discussed in more detail in the appendices of ftp://ftp.ncbi.nlm.nih.gov/blast/db/README. As this search has been done at the NCBI, and entries that would be in the European database EMBL are in the American

version Genbank and notated by *gb* in the list.

The definition lines are taken from the definition line in the database, with three dots (...) indicating that the definition line was too long for the space available.

What is listed in the NCBI summary as PAX6_MOUSE also includes pax6 human on the alignment table (see below).

Alignments

The sequence identifier, the full definition line and the length of the database sequence precede each alignment. The next lines display the score (both in bits and the raw score) and the statistical significance of the match, followed by the number of identities and positive matches according to the scoring system (e.g., BLOSUM62) and, if applicable, the number of gaps in the alignment. Finally the actual alignment is shown, with the query on top and the database match labelled as 'Sbjct'. Between the two sequences the residue is shown if it is identical, and a '+' if it is a conservative substitution. One or more dashes, "-", indicate insertions or deletions. X indicates that the residue has been masked out (see below).

The example below is the summary of the first NCBI hit, pax6 mouse:

```
>gi|417450|sp|P32117|PAX6_MOUSE PAIRED BOX PROTEIN PAX-6 (OCULORHOMBIN)
 gi|6174889|sp|P26367|PAX6_HUMAN   PAIRED   BOX   PROTEIN   PAX-6   (OCULORHOMBIN)
(ANIRIDIA, TYPE II PROTEIN)
          Length = 422


 Score =  723 bits (1865), Expect = 0.0
 Identities = 362/422 (85%), Positives = 362/422 (85%)
 Frame = +1


Query: 418  MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRY 597
            MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRY
Sbjct: 1    MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRY 60


Query: 598  YETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVCTNDNIPSV 777
            YETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVCTNDNIPSV
Sbjct: 61   YETGSIRPRAIGGSKPRVATPEVVSKIAQYKRECPSIFAWEIRDRLLSEGVCTNDNIPSV 120


Query: 778  SSINRVLRNLASEKQQMGADGMYDKLRMLNGQTGSWGTRPGWYPGTSVPGQPTXXXXXXX 957
            SSINRVLRNLASEKQQMGADGMYDKLRMLNGQTGSWGTRPGWYPGTSVPGQPT
Sbjct: 121  SSINRVLRNLASEKQQMGADGMYDKLRMLNGQTGSWGTRPGWYPGTSVPGQPTQDGCQQQ 180


Query: 958  XXXXXNTNSISSNGEDSDEAQMXXXXXXXXXXNRTSFTQEQIEALEKEFERTHYPDVFAR 1137
                 NTNSISSNGEDSDEAQM          NRTSFTQEQIEALEKEFERTHYPDVFAR
```

```
Sbjct: 181  EGGGENTNSISSNGEDSDEAQMRLQLKRKLQRNRTSFTQEQIEALEKEFERTHYPDVFAR 240


Query: 1138 ERLAAKIDLPEARIQVWFSNRRAKWRREEKLRNQRRQASNXXXXXXXXXXXXXXVYQPIP 1317
            ERLAAKIDLPEARIQVWFSNRRAKWRREEKLRNQRRQASN              VYQPIP
Sbjct: 241  ERLAAKIDLPEARIQVWFSNRRAKWRREEKLRNQRRQASNTPSHIPISSSFSTSVYQPIP 300


Query: 1318 QPTTPVSSFTSGSMLGRTDTALTNTYSALPPMPSFTMANNLPMQPPVPSQTSSYSCMLPT 1497
            QPTTPVSSFTSGSMLGRTDTALTNTYSALPPMPSFTMANNLPMQPPVPSQTSSYSCMLPT
Sbjct: 301  QPTTPVSSFTSGSMLGRTDTALTNTYSALPPMPSFTMANNLPMQPPVPSQTSSYSCMLPT 360


Query: 1498 SPSVNGRSYDTYTPPHMQTHMNSQPMXXXXXXXXXLIXXXXXXXXXXXXXXXDMSQYWPR 1677
            SPSVNGRSYDTYTPPHMQTHMNSQPM         LI               DMSQYWPR
Sbjct: 361  SPSVNGRSYDTYTPPHMQTHMNSQPMGTSGTTSTGLISPGVSVPVQVPGSEPDMSQYWPR 420
```

Statistics

The last section lists specifics about the database searched as well as statistical and search parameters used:

```
Database: swissprot
    Posted date:  Nov 4, 2001  2:23 AM
  Number of letters in database: 37,368,625
  Number of sequences in database:  101,737

Lambda     K      H
   0.318    0.135    0.401

Gapped
Lambda     K      H
   0.270   0.0470    0.230


Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Number of Hits to DB: 62125662
Number of Sequences: 101737
Number of extensions: 1161069
Number of successful extensions: 3183
Number of sequences better than 10.0: 57
Number of HSP's better than 10.0 without gapping: 37
Number of HSP's successfully gapped in prelim test: 0
Number of HSP's that attempted gapping in prelim test: 3122
Number of HSP's gapped (non-prelim): 56
length of query: 566
length of database: 37,368,625
effective HSP length: 53
effective length of query: 512
effective length of database: 31,976,564
effective search space: 16372000768
effective search space used: 16372000768
frameshift window, decay const: 50,  0.1
T: 12
A: 40
X1: 16 ( 7.3 bits)
X2: 38 (14.8 bits)
```

```
X3: 64 (24.9 bits)
S1: 41 (21.7 bits)
S2: 68 (30.9 bits)
```

These are also incorporated after the last alignment in the BLAST output.

We can judge the results of a **BLAST** search by two numbers. One is the 'bit' score; the expression of the score in terms of bits makes it independent of the scoring system used (i.e. the matrix). The other is the Expect value, which estimates the statistical significance of the match, specifying the number of matches with a given score that are expected in a search of a database of this size. It decreases exponentially with the Score (S) that is assigned to a match between two sequences. Essentially, the E value describes the random background noise that exists for matches between sequences.

An E value of two would indicate that two matches with this score are expected purely by chance. The E value is the most intuitive way to rank results or compare the results of one query run against two different databases. The E value is used as a convenient way to create a significance threshold for reporting results. When the E value is increased from the default value of 10, a larger list with more low-scoring hits can be reported. If your E value is high, it may not necessarily mean a bad alignment. It is also possible that your match is between a short section of sequence.

A WORD ABOUT FILTERING

In your **BLAST** results, you may see areas where residues have been replaced by NNNNN (for nucleotide sequences) or XXXXX (for protein sequences). This indicates that a region of low complexity sequence has been detected in this region by programs that are run on your sequence prior to the database search, and has been masked out to prevent artefactual hits. Low-complexity regions can result in high scores that reflect compositional bias rather than significant position-by-position alignment. Often repeat regions are also masked out – your sequence is searched against a database of human and rodent repeat (ALU, KPN, LINES, B1, MER, etc.) sequences. Any matches are masked out of your query sequence and the resulting masked sequence is then used to search against your chosen databases.

BLAST at NCBI filters low complexity regions by default. This can be altered to also filter repeat regions, or indeed the entire filtering system can be turned off. All options are under the “Options for advanced blasting” section.

Repeat this search using different options. For example, what happens when you switch the repeat filtering on, or the low complexity filtering off?

This is a DNA sequence being searched against a nucleotide database so it doesn’t make much sense to alter the scoring matrix; but if you chose to search against a protein database, it would be something you should think about.

Gene Identification Software

A relatively recent advance in bioinformatics has been the development of software specialised in gene identification (gene ID). These programs are used to recognise and extract the functional genetic information encoded in novel DNA sequences. Rather than collecting data for use in the laboratory, gene ID software assists investigators in the characterisation of such diverse genetic features as promoters, splice sites, coding versus non-coding regions, polyadenylation signals etc.

The individual features of a DNA sequence can be integrated to form a complete gene model. The efficiencies of the various gene ID algorithms are closely tied to our current understanding of molecular genetics. Although many different approaches are used to pick out specific features in unknown DNA sequences, all rely on past experience of the programmer (or program) with the feature under study. For example, when looking for splice sites in genomic DNA, some programs use the best currently accepted consensus (e.g. AG dinucleotide at intron/exon boundary) to scan the unknown sequence, while other programs are trained with a set of splice sites in known sequences before they are asked to evaluate an unknown sequence (neural network approach). Indeed, as with typical biology experiments, gene ID applications often have non negligible false positive and false negative rates in their predictions. The specific predictions (true positive) rates fluctuate between 50% and 95%, depending on the programs used and the DNA sequences submitted. Recent significant advances in the development of bioinformatics tools for gene identification have taken advantage of so called Hidden Markov Models (HMM) which are based on sound statistical models (e.g. **GENSCAN**).

There are many GeneID programs on the WWW. One of the best single programs for *de novo* gene prediction is **GENSCAN**, which uses HMM techniques (see next section).

Go to the Genscan server at http://genes.mit.edu/GENSCAN.html. Then follow the subsequent "Nucleic Acids"; "Gene Identification" and "Genscan". Scroll down the page until you reach the entry fields for your input sequence. Accept the default for organism and cutoff, and enter a description for your sequence. Select "Predicted CDS and peptides". Copy and paste your **pax6_genomic.fasta** sequence from the WWW[49] or Jemboss file manager into the data entry field. Enter your email address and press the "Run GENSCAN" button.

Take a look at the results. The predicted exons are displayed in a table where "intr" signifies an internal exon, "Term" signifies a terminal exon, and "PlyA" describes the position of a poly adenosine tail. "Sngl" defines a single exon gene, and "Prom" a promoter region.

GENSCAN is one of the gene finding programs that will recognise if there is more than one potential gene in a sequence, and will also make a good attempt at recognising partial genes. Each predicted gene is numbered (in this example, there are three predicted genes) and each of the features of that gene is also numbered (for example, 1.01, 1.02, and so on). The gene number is displayed, to the left of the feature

[49] This is available by following the "WWW menu" link from the RFCGR homepage. Select "Utilities", File Management" to the "Simple WWW file manager". Click on your file selection, and it will be displayed to you in the web browser from where you can copy it.

definition (Type) followed by the strand (S) on which the feature has been found. The start (Begin) and end (End) positions of the features are then displayed, together with the length (Len).

Various scores are presented, and you can read about exactly what they are on the GENSCAN web page. Each predicted exon is assigned a probability, P, which is *"the estimated probability under GENSCAN's model of genomic sequence structure that the exon is correct. This probability depends in general on global as well as local sequence properties."* Trials with sets of test data suggest that very high probability exons (P > 0.99) are nearly always correct, those with 0.50 < P < 0.99 are correct most of the time, while those with P < 0.50 are not reliable. GENSCAN has been designed primarily for vertebrate sequences and may be less accurate for non-vertebrates.

Predicted genes/exons:

Gn.Ex	Type	S	.Begin	...End	.Len	Fr	Ph	I/Ac	Do/T	CodRg	P....	Tscr..
1.01	Intr	+	1078	1194	117	2	0	71	52	101	0.610	4.32
1.02	Intr	+	1380	1510	131	1	2	141	58	68	0.979	8.59
1.03	Intr	+	2438	2653	216	1	0	104	94	184	0.995	18.38
1.04	Intr	+	3358	3523	166	0	1	76	93	110	0.633	8.91
1.05	Intr	+	3608	3675	68	0	2	88	34	28	0.386	-4.89
1.06	Intr	+	4425	4496	72	2	0	72	49	73	0.412	0.48
1.07	Intr	+	6243	6332	90	2	0	138	61	28	0.776	4.37
1.08	Intr	+	7135	7293	159	0	0	46	68	172	0.796	10.26
1.09	Intr	+	8347	8440	94	0	1	52	51	4	0.656	-8.28
1.10	Intr	+	9426	9584	159	1	0	108	54	233	0.717	20.94
1.11	Intr	+	10100	10182	83	0	2	29	86	120	0.971	4.34
1.12	Intr	+	10412	10562	151	1	1	72	58	73	0.983	1.51
1.13	Intr	+	10661	10776	116	0	2	113	108	88	0.949	12.55
1.14	Intr	+	13168	13504	337	0	1	0	99	244	0.536	10.87
1.15	Term	+	14195	14280	86	0	2	78	42	36	0.556	-5.16
1.16	PlyA	+	14994	14999	6							1.05
2.02	PlyA	-	15281	15276	6							1.05
2.01	Sngl	-	17114	16698	417	2	0	72	41	225	0.652	12.26
2.00	Prom	-	18687	18648	40							-5.35
3.02	PlyA	-	18984	18979	6							1.05
3.01	Term	-	20802	20690	113	1	2	58	50	131	0.532	4.04

Three genes have been predicted (defined by numbers 1, 2 and 3 on the left hand side of the results[50]).

Number of Exons (gene 1) ……

Start and finish of each ……

……

Does this correspond with the information in Ensembl? Why not[51]?

[50] The numbering is reversed for predictions 2 and 3 as they have been found on the anti-sense strand.

[51] Ensembl uses Genscan as only one of it's prediction tools. Genes are built using several sources of information. Also, some of the exons have been found on another clone, the sequence of which is no contained in Z83307.

[i] Fitch, W. M. (1966) J. Mol. Biol. 16, 9-16
[ii] Cohen, F. E., Novotny, J., Sternberg, M. J. E., Campbell, D. G., and Williams, A. F. (1981) Biochem. J. 195, 31-40.
[iii]McLachlan A.D. (1972) J. Mol. Biol. 64, 417-37
[iv] Feng, D. F., Johnson, M. S., and Doolittle, R. F. (1985) J. Mol. Evol. 21, 112-125
[v]
[vi]
[vii] Henikoff, S. and Henikoff, J. G. (1992) Proc. Nat. Acad. Sci. 89, 10915-10919
[viii] Risler, J. L., Delorme, M. O., Delacroix, H., and Henaut, A. (1988) J. Mol. Biol. 204, 1019-1029
[ix] Overington, J., Johnson, M. S., Sali, A., and Blundell, T. L. (1990) Proc. R. Soc. Lond. B. 241, 132-145
[x] Bowie, J. U., Luthy, R., and Eisenberg, D. (1991) Science 253, 164-170
[xi] Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978) A model of evolutionary change in proteins. matrices for detecting distant relationships In M. O. Dayhoff, (ed.), Atlas of protein sequence and structure, volume 5, pp. 345-358 National biomedical research foundation Washington DC
[xii] Schwartz, R. M. and Dayhoff, M. O. (1978) In M. O. Dayhoff, (ed.), Atlas of protein sequence and structure, volume 5, pp. 353-362 National biomedical research foundation Washington DC.
[xiii]
[xiv] Altschul, S. (1991) J. Mol. Biol. 219 (3), 555-565
[xv] Collins, J. F., Coulson, A. F. W., and Lyall, A. (1988) Comp. App. Biosci. 4, 67-71.
[xvi] Henikoff, S. and Henikoff, J. G. (1993) Proteins 17, 49-61
[xvii] Sternberg 1996, Protein Structure prediction - a practical approach, Edited by M. J. E.Sternberg, IRL Press at Oxford University Press, 1996, ISBN 0 19 963496 3
[xviii] Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997) Nucleic Acids Res. 25:3389-3402

Computational Protein Analysis

A protein sequence can be obtained from gene predictions, papers, database matches or even peptide sequencing. Having a protein sequence we will now want to try to identify it and find out something about its function. You may already have ideas about the physiological function, so you could use bioinformatics methods to support or refute these. It is important to use and critically evaluate information from different sources (including the nucleotide analysis), all of which will aid the determination of possible function.

To do this we can look for:

- Similar sequences in the sequence databases.
- Distinctive patterns/domains associated with protein function.
- Functionally important residues.
- Secondary and tertiary structure which provides more insights than the primary sequence.
- Physical properties, e.g. potential hydrophobicity/hydrophilicity and isoelectric point.

Protein sequence analysis is often more accurate than nucleotide sequence analysis because:

- It usually contains a higher signal to "junk" ratio.
- Database similarity searches are about five times more sensitive.
- The 3-D structure of similar proteins may be known.
- Evolutionary relationships are sometimes more visible.
- Annotation of protein sequence and related databases is often comprehensive.

Having found the sequence of the human PAX6 protein, in this chapter you will use bioinformatics tools to find out as much as you can about its structure and function. In these exercises, you search for known motifs and domains, predict secondary structure, construct a multiple alignment and examine structures of related proteins.

When using bioinformatics tools, you should check out every option of a program and read about what the program is really doing and what the output means. If you have any questions during this course on the nature of the theory behind the programs, then do ask. If you have a protein sequence, then it is sensible to do similarity searches, multiple sequence analysis, motif searches and possibly phylogeny on one or more sequences. If a 3-D structure of your protein exists, that's superb; if not, you currently have a 70-80% chance of predicting the overall structure correctly.

Always check your predictions using wet biology! Do not be overly trusting of computers, and always corroborate the results they give you.

Protein Sequence Analysis

We'll start with an introduction to the range of approaches you can take when trying to characterise your protein.

LOOK FOR KNOWN DISTINCTIVE PATTERNS

You can get a variety of clues by looking for patterns and motifs in your sequence:

- These are often derived from multiple sequence alignments.
- Conserved protein domains or regions can be very useful in trying to determine which protein family a sequence belongs to, catalytic sites, carbohydrate binding sites etc.
- Various research groups have created their own databases and search tools; it might be worth using a variety of these.

FIND HOMOLOGOUS (PARALOGOUS AND ORTHOLOGOUS) SEQUENCES

Using a database similarity search can give you a great deal of information:

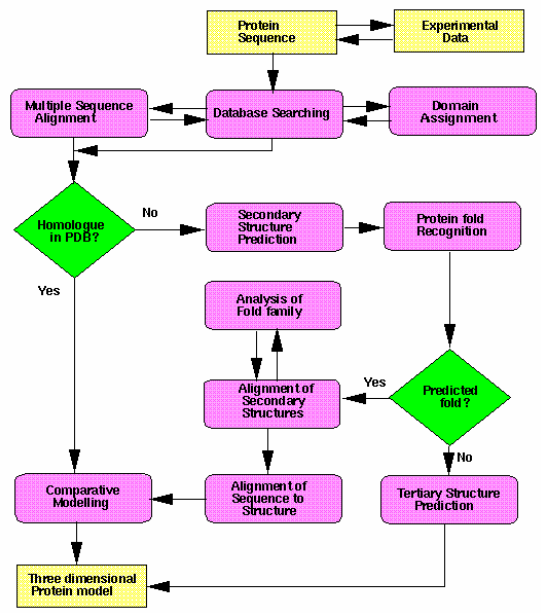
- Homologues may be well annotated and their function documented in the literature.
- Simply comparing your sequence with homologues can tell you a lot.
- Phylogenetic analysis may reveal evolutionary relationships between proteins and help you decide which family or super family a protein belongs to.

N.B. Be aware of convergent evolution.

HAVING SOME IDEA OF STRUCTURE MAY HELP YOU PREDICT POSSIBLE FUNCTIONS

Knowing the protein fold(s) together with conserved domains (or even residues) may tell you what type of functions this protein could have.

See the structure prediction flow chart figure.
Is comparative modelling by homology an option?



Structure Prediction Flow chart by Robert Russell

Search against sequences or structures in tertiary structure databases

Searching against databases such as PDB can help you to:
Predict possible secondary structure.
Predict tertiary structure de novo, best methods in the 70-80% accuracy range for folds[52].

We are going to have a brief look at the first two of these approaches. The second two are extremely complex and beyond the scope of this course.

Primary and Secondary Structure

Primary Structure

The "primary structure" of a protein is simply another term for its amino acid sequence.

Secondary Structure

Short regions of polypeptide chain often associate into regular patterns. The twist angles defining the conformation of the protein backbone will have similar values in these regions. The two most common types of secondary structure are the alpha helix and the beta strand.

The **Alpha Helix** is the "classic" element of protein structure. Its stability was predicted by Linus Pauling, seven years before helices were first observed in the myoglobin structure. An alpha helix is a tight coil held together by hydrogen bonding between backbone NH and CO groups. Helices are very stable; at least one is found in the great majority of protein structures.

The **Beta Strand** is a regular, extended section of polypeptide chain. Single strands are not stable, but a number of strands can associate into a **beta sheet**, which is stabilised by hydrogen bonds between the backbone of adjacent strands (which can be parallel or anti-parallel). Sheets can be all-parallel, all-anti-parallel, or mixed; all-anti-parallel sheets are the most common.

Protein Properties

There is a great deal of information about a protein to be gained from its amino acid sequence alone. Many programs will characterise various sequence properties. These may offer some clues as to the functionality of individual regions of the protein.

Basic statistic on the protein sequence may be gathered in textual form by the use of **pepstats**.

[52] Function and structure are not always perfectly coupled – although structural homologues are still a useful source of information.

Open **pepstats** in Jemboss from the scroll menu or the "protein" and "composition" menus. Drag and drop your protein sequence into the sequence filename field and hit "Go".

The results include information on the composition of the peptide sequence together with a calculated molecular weight and isoelectric point of the protein. The Dayhoff stats represent the molar percentage concentration of that residue within the protein divided by Dayhoff statistic (relative occurrence per 1000 residues, normalised to a percentage value) the.

Molecular Weight ……

Isoelectric point ……

If you wish to see a range of charges on the protein, as the pH values changes (relevant for purification techniques, perhaps), then the EMBOSS application iep will display this information in graphical format.

Select **iep** in Jemboss from the scroll menu or the "protein" and "composition" menus. Input your protein sequence and click LOAD SEQUENCE ATTRIBUTES. Tick the "Plot vs pH" option and hit "Go"

As with all bioinformatics applications, it is not a good idea to rely on just one program. The above values may therefore be tested using the Expert Protein Analysis System located at he Swiss Institute of Bioinformatics in Geneva.

Go to the ExPASy site at http://www.expasy.org and select the "Proteomics Tools" link on the right hand side of the page. Follow the link to the "PeptideMass" program, and paste in your protein sequence. Hit the "perform" button to run the application.

The results will appear almost instantly, headed by the sequence pasted into the program. This is followed by information of the type of digest selected (in this case, the default values) and a table of the individual regions digested, together with their respective molecular masses. The molecular mass of the entire macromolecule is noted above this table, together with the isoelectric point.

Molecular Weight ……

Isoelectric point ……

Are there any differences in the values calculated by the **pepstats** program. Why might this be?

Have a look at the ProteinView for each pax6 transcript in Ensembl and compare the values for molecular mass and isoelectric point. Are there any differences – why do you think that is? (Hint: use a pairwise alignment program to align the results of each transcript translation with the protein in the database)

Hydropathy

One of the first options that may be considered is the presence, or absence of transmembrane domains. These are regions of hydrophobicity within the protein, generally between 20-25 residues in length. Specific programs look for potential transmembrane regions, but they may also be spotted by constructing an hydropathy profile. Other regions of hydropathy may represent a hydrophobic protein core.

The sliding windows technique is used across the sequence, to create a plot representing the respective hydrophobicity and hydrophilicity of various regions. A "window" is selected for the sequence, and the hydropathy scores of each of the residues within that window are added together and the sum total divided by the number of residues in the window. The scores themselves have been devised experimentally by several researchers according to their specific criteria.

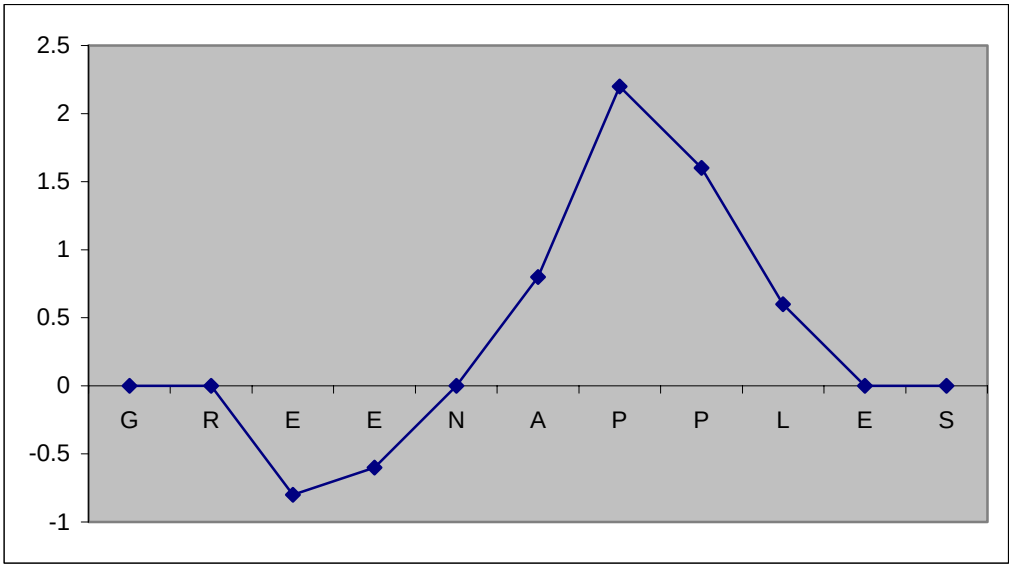
Thus, from the sequence GREENAPPLES and a hydropathy scoring matrix of

G	3	N	1
R	-2	A	4
E	-3	P	1
L	5	S	-1

A window size of five would result in addition of the hydropathy scoring of residues 1-5 (3 + (-2) + (-3) + (-3) + 1) divided by five.

The resulting score for that window would be –4/5 = -0.8. The window then slides along one residue to calculate the score for residues 2-6, then residues 3-7 etc, plotting the relevant score on a graph each time. The resulting graph would look like this:

If the zero line of the horizontal axis indicated the cutoff for hydrophobicity, this profile would suggest two potential hydrophobic regions, which may represent transmembrane domains.



We will now look at some of the applications used to search for regions of hydropathy within a protein. Before we start the analysis, you may want to make a separate directory either on your local computer, or your remote server to house your new analyses in.

Select the program **pepinfo** from the scroll bar or the "protein" and "composition" menus. Type **sw:pax6_human** into the "input sequence" field. Click the "LOAD SEQUENCE ATTRIBUTES" button and accept the defaults displayed for you. Press "Go" and wait for the results to be displayed. The default view for the results output should be **pepinfo1.png**, which will display results in their graphical format.

The distribution of various residues is displayed in this first view. The top graph represents small amino acid residues in the sequence, the next displaying tiny ones. The following graphs display the distribution of aliphatic; aromatic, non-polar; polar; charged; positive; negative residues in descending order. For documentation to describe the characteristics of each distribution grouping, see appendix III at the back of this book.

Pepinfo2.png is the graphical output of the hydropathy plots using the algoriothms developed by Kyte & Doolittle[xix] and Sweet & Eisenberg[xx]. The bottom graph is a consensus plot using the method of Eisenberg *et al*.[xxi] Regions of particularly high hydrophobicity may indicate transmembrane segments, though there are a variety of other programs that specifically predict transmembrane regions.

After retrieving information on the make up of the protein sequence, a logical following step is to look at secondary structure predictions. The EMBOSS program **Garnier** uses one of the simplest algorithms[xxii] for predicting secondary structure. It was devised in 1978 when the selection of 3 dimensional structures to use as a basis for algorithm development was much more limited than it is now, but it is still in use as a primitive prediction tool.

Select **garnier** from the scroll menu or the "protein" and "2D structure" menus. Type **sw:pax6_human** into the sequence input field and select "Go".

As you will see from the output, this program tries to predict four types of secondary structure: α-helices (H); β-sheets (E); coils (C) and turns (T), and their positions within the protein sequence.

```
GARNIER plot of PAX6_HUMAN, 422 aa; DCH = 0, DCS = 0

 Please cite:
 Garnier, Osguthorpe and Robson (1978) J. Mol. Biol. 120:97-120

           .   10    .   20    .   30    .   40    .   50    .   60
       MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVSKILGRY
 helix                            HH     H
 sheet EE       EEEEEEEE          EEEEE         EEEEEE     EEEEEEEE
 turns                       T          TTTT TTT      TTTTT
 coil    CCCCCCC        CCCCC CC            C
```

The first thing you might note from this output, is that certain features are composed of only one or two residues. Particularly in the case of larger structures such as helices or sheets, these results must be interpreted very carefully.

This early algorithm is not expected to predict with more than about 65% accuracy. More modern programs for secondary structure prediction have not improved much since the days of Garnier, Osguthorpe and Robson, now achieving perhaps 75% accuracy. The newest programs tend to use a consensus of several methods, and to work with families of proteins rather than single sequences. JPRED, which is available at the EBI at Hinxton, is one of the most accurate of these programs.

Jpred

Jpred is a web-based application that takes either a single sequence or a multiple sequence alignment (see later), uses a variety of algorithms (including GOR as implemented in **garnier**) to predict protein secondary structure and presents a consensus result.

You cannot use database entries as input to **Jpred**, so you will have to cut and paste your sequence in from another window. We'll use SRS to find the sequence we need. You'll need to have two browser windows open at once to do this efficiently.

Go to the EBI homepage at http://www.ebi.ac.uk. Type **pax6_human** into the "Database Search" box at the top of the page. Alter the pull down menu to read "Protein Sequences" and click on the "Go" button.

The Uniprot entry will be displayed on screen.

From the right hand menu above the full entry select the "fasta" link which will display the sequence in fasta format. Highlight this sequence

and copy it. (You could just as easily have used seqret on Jemboss to retrieve the sequence)

We now have the sequence we require for our Jpred analysis. This is a method that can be employed for retrieving any sequence to paste into the field of an analysis program. Multiple sequences may also be retrieved and copied in this manner.

Go to the Jpred web site at http://www.compbio.dundee.ac.uk and follow the link to the "Jpred" site and then the "server" link. Select the "Prediction" option. Enter your email address into the field in Step 1. Paste your sequence into the appropriate field in Step 2, and check you have not copied any of the sequence description line by mistake. Leave the defaults set in Step 3. Click the button marked "Run secondary structure prediction now!" to start your analysis.

Your run should stop almost immediately, as there are several hits to three-dimensional structures in the Protein DataBank (PDB). These are obviously going to be a lot more accurate than any prediction. However, for the purposes of this run, we wish to disregard them.

Use your browser's "Back" button to return to the Jpred for and scroll down to Step 4 and tick the box. This allows the user to bypass any scan of the PDB. Re-run the analysis. You will be advised by email when this has finished.

When given the option, the easiest way to view the results is as HTML. Occasionally the Jpred server can be a bit temperamental. We've included here part of the results we got from this analysis in case that happens to you today.

Predictions for request pax6_3669

```
YourSeq:          MQNSHSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGA
O93374 :          HGGLNQLGGMFVNGRPLPEVIRQRIVDMAHQGVRPCD
O57416 :          TPLGQGRVNQLGGVFINGRPLPNHIRHKIVEMAHHGI
PAX2_HUMAN-02 :   .G..HGGVNQLGGVFVNGRPLPDVVRQRIVELAHQGV
O61616           : .SGGHGGVNQLGGVYVNGRPLPDVVRHRIVELAHQG
PX8D_HUMAN       : ...GHGGLNQLGGAFVNGRPLPEVVRQRIVDLAHQG
O57419           : TPLGQGRVNQLGGVFINGRPLPNHIRHKIVEMAHHG
: 1---------11--------21--------31--------41--------51--------61--------71--------81--------91--------101-------111-------121-------131-------141-------151-------161-------171-------181-------191-------201-------211-------221-------231-------241-------251-------261-------271-------281-------291-------301-------311-------321-------331-------341-------351-------361-------371-------381-------391-------401-------411-------- :

jalign           : -------------EE----------HHHHHHHHH-----------EE-----EEEEE--EE--
jfreq            : ---------E--EEEE--------HHHHHHHHHH--------EEEEEE---EEEEEE-EEE--
jhmm             : ----------------------HHHHHHHHHHHH-----------------------------
jnet             : ---------E---EEE------HHHHHHHHHHHH-------HHHHHH----HHHHHHHH----
jpssm            : ---------E------------HHHHHHHHHHHH------HHHHHHHH-HHHHHHHHHHHH-

Jpred            : ---------E---EE-------HHHHHHHHHHHH-----------------HHHHH------
MCoil            : ------------------------------------------------------------
MCoilDI          : ------------------------------------------------------------
MCoilTRI         : ------------------------------------------------------------
Lupas 21         : ------------------------------------------------------------
Lupas 14         : ------------------------------------------------------------
Lupas 28         : ------------------------------------------------------------
```

```
Jnet_25         : -------BBBBBBBBBBBBBB--BBB-BBBBBB--BB-BBBBB--B-BB-BBBBBBBB-B-B-
Jnet_5          : ----------B--BB------------B------------B---B------B--B-------
Jnet_0          : ------------------------------------------------------------
JnetRel         :789999840014436399988626899999999707687441344321055145666100046 9
```

First you see your sequence aligned with a range of sequences retrieved using a PSI-Blast search (we'll see PSI-Blast later on). The idea behind this is that secondary structure predictions are more accurate if based on a multiple alignment of similar proteins than on a single sequence. Jpred presents the results of running various secondary structure prediction algorithms on the alignment, representing potentially helical regions as H, and predicted sheet regions as E, and a consensus prediction is presented. Algorithms are also run to predict coil regions. The key to all the abbreviations is given below the results.

Sequence Motifs and Domains

Sequence motifs and domains are derived from sequence alignments of related proteins. The distinction between a "motif" and a "domain" is not clear-cut, but the term "motif" is generally used to refer to a short sequence pattern and "domain" to a longer region of sequence similarity. Domains are almost always represented by statistical models, known as profiles, which describe the probability of finding each amino acid at each position in the domain.

MOTIFS

The main database containing protein motifs is PROSITE. The sequence motifs in this database are described using a simple pattern description language. They include some very common, simple motifs, many only a few residues long, that indicate possible sites for post-translational modifications (e.g. glycosylation or phosphorylation).

Go to http://www.expasy.org and select "PROSITE" from the "Databases" section. Select "ScanProsite" from the "Tools for PROSITE" section of the page. Enter **pax6_human** in the entry field under "Scan PROSITE for patterns, profiles and rules…..." and tick the "exclude patterns with a high probability of occurrence". Use the "Quick scan" button to start the search. It should only take a few seconds, then examine the output.

The results should take only a few seconds to be returned, and you should see that only two patterns have been returned. The homeobox, and the paired box signatures both appear in the results section, together with the respective fragments of sequence found in your query. Following the link to the number beginning with "PS" will show you the prosite entry for each domain. The link starting with "PDOC" will display entries to give you more information about the domain, and proteins containing it.

Type of nuclear binding in Homeobox domain ……..

Homeobox domain consensus pattern ……..……

……

…….

Re-run the query, but this time, deselect the ““exclude pattens with a high probability of occurrence” option. How many domains are you now presented with? Why[53]?

You should already know from the database entries that PAX6 is a DNA binding protein. Many (but by no means all) of these proteins bind DNA with a simple structural motif consisting of two alpha helices joined by a short loop (the so-called helix-turn-helix motif). A simple EMBOSS utility, **helixturnhelix**, can be used to look for the pattern of amino acid residues associated with this motif.

Select **helixturnhelix** from the scroll menu, or from the “Protein” and “2D structure” menus. Drag and drop **pax6.pep** into the sequence filename field, accept the defaults and press “Go”.

Your results should display one fragment of the sequence indicated by the program to be a possible helix turn helix motif.

```
#=======================================
#
# Sequence: PAX6_HUMAN     from: 1   to: 422
# HitCount: 1
#
# Hits above +2.50 SD (972.73)
#
#=======================================

Maximum_score_at at "*"

(1) Score 1109.000 length 22 at residues 238->259
           *
 Sequence: FARERLAAKIDLPEARIQVWFS
           |                    |
         238                  259
 Standard_deviations: 2.96

#---------------------------------------
#---------------------------------------
```

Have a look at the predicted helix turn helix motif. Does it correspond to the pattern you identified as a homeobox domain in prosite[54]?

DOMAINS

[53] This type of search will includes such domains as glycosylation, or myristoylation sites, which, although biologically relevant to your protein, may not offer as many clues about the function of it.

[54] It should not be exactly the same, as the homeobox domain is obviously made up of much more than just one motif, but you should find the HTH pattern within this domain.

There are many protein domain databases available on the Web. One of the most widely used is **Pfam**, which is available at the Sanger Institute. Pfam is a collection of protein families and domains, and contains multiple protein alignments and profile-HMMs of these families. Over 50% of the entries in SwissProt have a match to one of the families in Pfam.

There are two types of protein family in Pfam:
Pfam-A families are curated; a seed alignment of representative sequences is inspected. A profile (see later) representing this seed alignment is used to search SwissProt and matching sequences are automatically aligned to the seed alignment to create the full alignment. There is usually extensive annotation for a Pfam-A family.
Pfam-B contains sequences that are not contained in Pfam-A. The alignments are produced completely automatically and may not be of such high quality as the Pfam-A alignments. The alignments used here come straight from the ProDom[55] database.

Go to http://www.sanger.ac.uk/Software/Pfam and select "Protein Search". Enter the SwissProt identifier **pax6_human** into the top field (the query box) and press the "Submit Query" button to start the search.

You should see the two domains you discovered using prosite have now been picked out in Pfam.

Start and End of PAX domain .....

Start and End of homeobox domain.....

Are the regions, which these domains cover the same as you discovered in prosite?

Examine each domain entry by clicking on the domain icon to find out more about it. Click on the "View Graphic" button in the "Domain organisation" box. This will give you a graphical list of all other SwissProt proteins containing these domains.

Another database that defines functional protein families in a similar way is **PRINTS**. In this database, each domain is identified by a number of short, particularly well-conserved sequences. A full match to one of these "fingerprints" will match all the relevant short sequences in the correct order. A partial match is recorded if some are missing or if they occur in an incorrect order.

The PRINTS database can be searched using the **fingerPRINTscan** program.

Go to http://umber.sbs.man.ac.uk/fingerPRINTScan/ and select the FPScan link (or alternatively you can use the mirror at the EBI http://www.ebi.ac.uk/printsscan/) and paste in the pax6.pep sequence in raw format. Leave all defaults and hit the "Send Query" button.

[55] ProDom is an uncurated database held in Toulouse in France. It aims to align all fragments of sequence involved in a particular domain. See http://prodes.toulouse.inra.fr/prodom/doc/prodom.html

There are two fingerprints (pairedbox and hthrepressor) which show matches of all elements with our pax6 protein sequence. We know from previous analysis that the paired box motif exists, but the second motif is not mentioned anywhere else.

Click on the “graphic” link next to the hthrepressor fingerprint to see the graphical display of the hits.

The graphic shows that the first element has matched the pax6 sequence in several places, but the second element only matches in one place. This is in the same region as our homeobox motif.

How do the results of this scan compare with those from Prosite and Pfam? Are there any false positive matches?

There are many more pattern and motif databases; a full analysis of these resources is outside the scope of this course. Each uses a different approach and contains a different collection of protein families. Some are particularly useful for a certain type of protein; for example, the SMART database specialises in domains that occur in signalling proteins. Therefore, it is always good practice to search a range of databases with your sequence.

You should have found that the databases agree that PAX6_HUMAN contains two distinct domains. The C-terminal domain is a homeobox (Prosite finds both characteristic homeobox motifs); the N-terminal one is a PAX domain, which contains the longer “paired box” motif.

Multiple Sequence Alignment

All entries in motif and domain databases are constructed (whether manually or automatically) from multiple sequence alignments. A multiple sequence alignment is self-explanatory: an alignment of many related protein or DNA sequences. Residues that are equivalent in all members of the family are aligned.

Why?

There are many reasons why you might want to construct a multiple sequence alignment. These include:-

To highlight regions of similarity, divergence and mutations.
To provide more information than a single sequence. (e.g. for an even more sensitive search to find other, more distant, family members.)
Creating a consensus will highlight functionally important domains or residues.
It could reveal errors in protein sequence prediction (or even in sequencing)
Secondary structure and other predictions improve with multiple alignments
Evolutionary analysis (phylogeny).
To find novel motifs (e.g. using Hidden Markov Model techniques).
To select appropriate primers for a gene family.

Multiple Sequence Alignment Methods

Automatic:
Programs include the Clustal family; HMMer and AMPS.
Good when > 50 % identity.
Semi-automatic:
Hidden Markov Model techniques
Hand craft the first alignments, then automatic
Fully hand crafted:
Cinema
Slow!

Almost everyone will want to start a multiple sequence alignment project using one of the automatic methods. However, this has some disadvantages. The programs treat protein (and gene) sequences as sequences of letters only: they "know no biology". You should never think that the result of running a program like Clustal once, with a random selection of proteins from a particular family, as "the" correct answer.

How do I get a better alignment?
Are your sequences actually homologous?
Alignments should be done on appropriate sequences, the sequences you choose to align will depend upon what you wish to achieve; you may wish to analyse a representative spread of a gene family or compare a specific gene in various organisms etc.
Choosing just sequence fragments, which share a common region, may well give a better alignment than full-length sequences.
Once you have an approximate alignment, it is then best to edit this using your knowledge about the proteins
The sequences may have errors, insertions, deletions etc.
Structural information is often more conserved than sequence, so use this if it's available.
Additionally you should try altering program parameters.

Rigorous optimal alignments would require huge computing resources, so various heuristic approaches are used. **Clustering** is a good popular method, which is used in the widely used **Clustal** series of programs. **Clustal** has many advantages: it is fast, reliable and easy to use. It is also available, free, for a wide range of computer platforms. Here's an outline of its approach:

Clustering
1. Align every pair of sequences with each other, recording the pairwise similarity scores.
2. Align the most similar pairs of sequences to produce clusters.
3. Repeat 1 and 2, using increasingly dissimilar sequences or clusters of sequences, until you have produced a tree.

We will align PAX6 with a small number of other human PAX (paired box) proteins. We will use the EMBOSS program **emma**, which is an interface to ClustalW.

We're going to use SRS to get some sequences for the alignment.

Go to http://srs.ebi.ac.uk and select the "Library Page". Select the SwissProt database and choose the "standard" query form. In the first text box type **pax** and set the associated menu to "Description". In the

second box type **human**, and choose “Organism Name” from that menu. Ensure that the “Use View” menu is set to “SeqSimpleView” and the “Use Wildcards” is selected. Click on the “Search” button.

You should get 11 entries displayed in a simple table format where each row in the table is a SwissProt entry. Nine of the entries are obviously paired box protein sequences, while two are not. Why were these sequences picked up by the search? (**Hint**: SRS did a text-based search for PAX). If your text searches also reveal several proteins which obviously do not belong to the group you wish to align, you must ensure they are excluded from and subsequent operations.

We now want to save the entries we have found to do some multiple sequence alignments.

Find the paxillin entries. These are obviously not pax proteins, so tick the box to the left hand side of this entry. Below the “Apply options to”, ensure that the “unselected results only” option is chosen. Set the pull down menu below the “View” button to “names only” and press the yellow “Save” button. Ensure the “Save table as ASCII” box is ticked, and the “Output options” set to “text/html (to browser window)” before pressing “Save” once again. The accession numbers of ten sequences should now be listed in a manner which will allow you to copy them into an EMBOSS application.

You may now save this information as a file using something like notepad. Save it in the directory you have also specified as your Jemboss local home directory (You can check the exact path by looking in **Preferences/Advanced Options** on the Jemboss toolbar).

[A sneakier way of doing it is to open up an existing file in Jemboss, delete the contents, paste in the pax information and then save it – remembering to rename it!]

We have created what is known as a list file – i.e. a file containing a list of identifiers, which we can use as input to several EMBOSS applications. We will now run emma using this new list file, but bear in mind, that this file can just as easily be created on your laboratory or home PC and dragged *via* Jemboss into any program form (if you are doing this, however, you MUST ensure that is gets saved as a txt file and not as html otherwise EMBOSS will not recognise it).

Select **emma** from the scroll menu or the “Alignment” and “multiple” menus. Go to the “File” menu in the top left hand corner of the Jemboss interface window and select “Show remote files”. Find the pax.sw file that you transferred and drag it into the “Sequence Filename” field. As this is a list file, it must be pre-pended with an @ sign – so type **@** at the beginning of the file path name so the correct input should read something like **@C:\Windows\Bioinformatics\pax.sw** (or similar, depending on the path for your local directory). Leave the default setting as they are, and press “Go” to run the program.

A multiple sequence alignment will require rather a lot more computing power than something such as seqret. At the bottom of the Jemboss window, you will see that the job manager is automatically turned on as the process is run in “batch” mode – i.e. in

the background and not interactively. This enables you to carry on doing something else whilst one process is running. When the program has finished, the job manager with automatically update to tell you that the job has been "completed" and is not still "running".

When the job manager window says that your process is complete, click on the job manager bar and select the process that you just ran[56]. Click on the "display" button to display the results. Look for a **pax.aln** file – this is the result of your alignment. Save this file to your account[57].

This file may look like several fasta formatted files with gaps, and it is certainly not easy to determine which areas are aligned, and which are not. Currently a second program is needed to view this alignment in a more meaningful format.

Select the program **prettyplot** from the scroll menu or the "Display" menu. Drop and drag the pax.aln file into the correct field and hit the "LOAD SEQUENCE ATTRIBUTES" button. Leave the defaults as they are, and "Go".

A graphical window should now appear on your screen. The nine sequences of the alignment are blocked together and broken into blocks over three pages – use the tabs to move from one page to the next. The red coloured residues are conserved residues in the alignment. The green coloured residues are where similar residues occur in a particular position in the alignment. You should be able to see one long stretch in which the sequences are highly conserved. From your knowledge of the sequences, which domain do you think the conserved region corresponds to?

We now want to take a closer view of the alignment of the conserved paired box domain by loading it into the Java based graphical editor called Jalview.

Choose the "Tools" menu from the interface toolbar, and select the multiple sequence alignment editor. Drag and drop your .aln file into the correct field, and change the "file type" to **fasta**. "Launch" the viewer.

The first thing you may notice underneath your alignment is a consensus pattern. You may also notice that the pax (paired box) domain is very well conserved. Scroll through to residue position 150. Can you see that there has been a bit of an alignment problem here? Look for the two serine residues in the sequences – are they aligned in all 11 sequences?

Jalview lets you edit the alignment. One of it' option allows you to colour the alignment. This may make it easier to spot mismatched regions. Position the mouse arrow cursor over the residue you want to slide, press down the left mouse button and drag the sequence along to the right. Gap characters appear automatically. Introduce a two residue gap in all the sequences except PAX7_human. Create a two residue gap in all the sequences except PAX7_human so that the residues VSS now line up properly. At this point it is worthwhile calculating the new consensus

[56] If you do this too early, and the program is still running, the right hand side of the job manager window will state that the results are "pending".

[57] When you save a file, ensure that you have actually activated the tab in the results window before saving, otherwise you will create a file with mystery content!

score. Look at the graph under the sequences – there is a "hole" in it around the area we have just been altering. Select the menu **Calculate**, and click on **consensus** and you should see a hole appear where your insertion is.

There is an additional multiple sequence alignment viewer that is unique to Jemboss.

Highlight the pax.aln file and then access the file menu with your right hand mouse button. Select the "Open with" option and then "Jemboss Alignment Editor".

You can do the same manipulations with this editor.

If you have a spare minute, go back to Pfam at http://www.sanger.ac.uk/Software/Pfam and look at the PAX entry by using the option to "Browse Pfam". Enter **PAX** in the query entry box and click on "Search". The top hit will take you to the PAX 'paired box' domain entry. You will see the alignment option on the left. Click on the radio button next to the "Full (235)" option, then click the "Get Alignment" box underneath. Look for the PAX7_human entry (hint: use Edit -> 'Find in page'). Scroll to the very right of the view. The alignment shows you the "GL" residues of PAX7_human are an insertion.

PSI-BLAST

PSI-Blast is a new version of the BLAST algorithm that allows you to search for remote homologues of your protein sequence. The idea is well explained on the PSI-Blast web site at http://www.ncbi.nlm.nih.gov/blast/index.html (choose the psi-blast option).

1. PSI-Blast takes as an input a single protein sequence and compares it to a protein database, using the gapped BLAST program

2. The program constructs a multiple alignment, and then a profile, from any significant local alignments found. The original query sequence serves as a template for the multiple alignment and profile.

3. The profile is compared to the protein database, again seeking local alignments.

4. PSI-Blast estimates the statistical significance of the local alignments found.

5. Finally, PSI-Blast iterates, by returning to step (2), an arbitrary number of times or until convergence.

PSI-Blast is an extremely useful facility for identifying new members of a protein family, and is frequently the tool of choice of researchers involved in "domain hunting". However, you do need to be careful with PSI-Blast; if you just blindly let it run till convergence there is a danger that unrelated sequences can creep in and contaminate your profile. Once this happens, even more unrelated sequences can be brought in at the next iteration and you may be misled about the nature of your protein. We'll illustrate this with a practical. Running PSI-Blast with the entire PAX6 protein sequence can be slow and will pull in hundreds of paired box proteins. We'll use just one of the regions annotated in the SwissProt entry for PAX6:

In one browser window start an SRS session and use the Quick Search option to pull up the entry for **pax6_human**. Scan down to the bottom of the sequence to find the section marked "Features" and find the line

COMPBIAS 279 422 PRO/SER/THR-RICH.

Click on the link to "Domain" in that line. You now see a page with a portion of the PAX6 sequence, representing a domain we haven't yet found any information about. Use your mouse to highlight this portion of the sequence, and then use your browser's **Edit** menu and select **Copy**. Use a second browser window to go to the NCBI PSI-Blast page at http://www.ncbi.nlm.nih.gov/blast/index.html and select the **PSI- and PHI-BLAST** link. Restrict the search to just SwissProt to save time; so press the pull down menu marked **nr** at the top of the page and from it select **swissprot**.

Click in the text box marked "Search**"** and then use **Edit**->**Paste** to paste your sequence into this box. Leave all other option and hit the **Blast!** Button. Now wait for a few moments while PSI-Blast does its first Blast search with your sequence. Click on the Format button to display the graphical view in a separate window.

At the top of the results page is a link to the PSI-Blast reference, followed by information about the search you performed and a graphical blast hit viewer much like the one you saw earlier in the nucleotide BLAST. Below this is a list of the hits, divided into two sections:

- **Sequences with E-value BETTER than threshold**: the default threshold for PSI-Blast is E=0.005. The sequences in this region are those that gave E-values better (i.e. closer to zero) than this.
- **Sequences with E-value WORSE than threshold**.

PSI-Blast will use hits from this list to construct a profile for re-searching the database. By default, all the hits with E-values better than the threshold will be used to construct the profile, though you can select and deselect sequences as you wish.

Make sure all the hits with E-values better than the threshold and none of those with E-values worse than the threshold are selected. Press the button marked **Run PSI-Blast iteration 2** and wait for the results. You will be expected to go back to the original PSI-BLAST window (the one you started from) and press "Format" again. Then your results will appear in the second window.

This time the results page says **"No new sequences were found above the 0.005 threshold!"** This means that PSI-Blast didn't find any additional proteins with E-values better than the threshold and has converged. The same seven PAX6 proteins are found.

Now, suppose you were searching with a sequence that you don't yet know anything about. The first hit that falls below the threshold, **Nicotinamide mononucleotide adenylyltransferase,** has an E-value of 0.018, which is a marginal hit, but it may not be unreasonable to want to include it in the next round of searches (**NB** From time to

time, NCBI make adjustments to PSI-Blast. If **Nicotinamide mononucleotide adenylyltransferase** is not in the results list, just choose another sequence to illustrate our point[58]).

Tick the box to the left of "Nicotinamide mononucleotide adenylyltransferase" and press the **Run PSI-Blast iteration 3** button**.** Remember to use both browser windows!

This time, note that the inclusion of just one additional sequence has pulled in a few additional sequences, which may or may not be related to our Pro-Ser-Thr rich domain. In this case, these sequences have relatively large E-values, but depending on the sequence, they may have very small E-values, which would normally mark them out to you as being highly significant hits – in particular, note that the E-value for **Nicotinamide mononucleotide adenylyltransferase** is now reported as much smaller than the original 0.018.

This illustrates several valuable points about PSI-Blast:

- As with all similarity searches, you must evaluate the results carefully. Would you be justified in characterising your original sequence as a Swiss Cheese protein subunit on the basis of this search?

- Be very careful with the E-values you get from PSI-Blast. Note that the E-value for **Nicotinamide mononucleotide adenylyltransferase** as much smaller after the second iteration, and it's easy to assume that this means it has suddenly become a very significant hit – at any iteration the E-value merely reflects the significance of the match to the sequence set chosen in the previous iteration. All it's really telling you in this case is that **Nicotinamide mononucleotide adenylyltransferase** is very similar to itself, not to your original sequence.

- Once a sequence is included in a PSI-Blast iteration, all its friends come in on the next iteration, often with what appear to be highly significant E-values – you need to be very careful about the sequences you include.

Don't just assume that because something comes above your threshold E-value it "must" be relevant – you have the power to set the threshold for inclusion in PSI-Blast iteration to any value you choose, and the default of 0.005 is a very conservative value. Admittedly, in this case our PAX6 hits all had E-values significantly smaller than the 0.018 of the additional sequence we selected, but that won't necessarily be the case with your proteins. It certainly wouldn't be unreasonable to set the threshold to 0.01 – would you ignore a normal BLAST result if the top hit had an E-value of 0.06?

PSI-Blast is a very powerful tool and is widely used both by researchers and by other software applications. It can be very useful, but you do need to be careful with it, possibly more than with any of the other applications we have shown you on this course.

[58] As sequences are added to the database, it is possible that there will be new sequences that resemble our domain to a greater extent than the protein mentioned. Due to the rise in sequences in the database, the statistics change, and the E value may also differ from the text.

Protein Tertiary Structure

The functional properties of proteins depend on their three-dimensional structures. Their linear polypeptide chains fold into a wide variety of shapes, locating the functional groups of essential amino acids in positions where they can interact with ligands - such as in an enzyme active sites - or receptors. The first protein structure - that of myoglobin - was determined by John Kendrew and his colleagues in 1958 using X-ray crystallography. For about a quarter of a century after that, the number of protein structures known grew very slowly, reaching a couple of hundred by the mid-1980s. Improvements in techniques, particularly the influence of recombinant DNA technology, has led to an explosion in the number of known structures. X-ray diffraction is still the most common technique used, but the proportion of structures determined using multi-dimensional NMR is increasing, and structures of a few proteins - particularly membrane bound proteins - have been determined to fairly high resolution using electron diffraction.

The Protein DataBank (PDB) has been the repository for all publicly available protein structure co-ordinates since the mid-70s. The first release had fourteen entries; in November 2001 the PDB held over 16,500 structures, with dozens released every week. This does not, however, mean that the structures of over 16,500 different proteins are known, for example the database contains several hundred different lysozyme structures.

Until 1999 the PDB was based in Brookhaven, New York. It has recently been taken over by a consortium of three US based groups known as the Research Collabatory for Structural Bioinformatics. All structures are available via a Web interface; the main site is currently at http://www.rcsb.org and mirror sites are being established in many countries. It is very useful to use a local mirror, particularly if you are likely to be downloading large co-ordinate entries. A few sites still use the search engines developed for use with the original PDB. There is a full mirror of the new PDB at the Cambridge Crystallographic Data Centre (CCDC). This mirror is located at http://pdb.ccdc.cam.ac.uk**.** There is also one held at the EMBL outstation at Hinxton. The URL for this site is http://pdb-browsers.ebi.ac.uk/.

Numerous graphics programs are available for viewing protein structures, ranging from complex (and very expensive) molecular simulation packages to simple programs for PCs such as the acclaimed, and free, **Rasmol**, whose web site is at http://www.umass.edu/microbio/rasmol There are many different ways of representing structure, which emphasise different aspects of the structures. Three common ones are:

> **Stick mode**: bonds are represented by lines and atoms are not shown. This represents the chemical details of binding sites accurately, but cannot show the overall fold of a protein clearly.
> **Space filling**: each atom is represented as a sphere, with its radius proportional to the Van der Waals radius of the atom type. This is the most "realistic" way of representing a protein structure, but the least clear.
> **Ribbon or cartoon drawings**: a smooth ribbon is drawn through the protein backbone. This is the clearest way of representing protein folds. The familiar cartoons with alpha helices shown as coils and beta strands as

arrows were first introduced by Jane Richardson. Helices can also be shown as cylinders.

Protein structures are described using a hierarchical terminology. This can be quite confusing, especially as terms such as "motif" and "domain" have been introduced into the original classification scheme. We outline the hierarchy here:

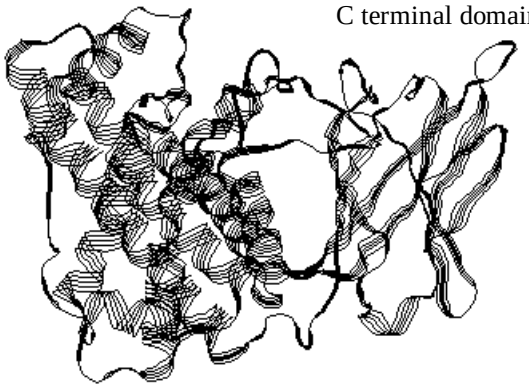
SUPERSECONDARY STRUCTURE; MOTIFS

Small groups of secondary structure units often associate together to form definite recognised patterns. Simple examples include: the helix-turn-helix motif, the beta hairpin (which consists of two anti-parallel beta strands joined by a short loop) and the beta-alpha-beta motif (where two parallel strands are joined by a single alpha helix). These units are found in many proteins with different functions, and there is often no sequence similarity between similar structural units. These groups of helices and strands are termed supersecondary structures or structural motifs.
Some specific types of motif may have either a distinct function, an associated characteristic sequence pattern, or both. For example, one specific type of helix-turn-helix motif, with a characteristic angle between the helices and characteristic amino acid sequence pattern, is known to bind calcium. A helix-turn-helix motif with a different geometry will bind DNA. Larger and more complex combinations of secondary structure elements can also be associated with specific functions, such as the Rossman fold - a beta-alpha-beta-alpha-beta motif - which binds mononucleotides.
Protein functions are often characterised by sequence motifs where there is no known association with a particular structure.

DOMAINS

A domain is defined as a compact globular unit within a protein, which is stable as an independent unit (i.e. it has its own hydrophobic core). A single protein chain may contain one or more domains. In multi-domain proteins, approximate domain boundaries are usually easily determined by eye; determining the exact boundaries between domains is often fairly subjective. Domains often have specific functions: in a multi-domain enzyme, one domain - the catalytic domain - will contain most or all of the active site residues. Some proteins, such as immunoglobulins and gamma crystallin, consist of two or more copies of similar domains.

The structure of alpha-toxin from *Clostridium perfringens* - the key determinant of gas gangrene - is readily seen to be divided into two domains. The N terminal domain has phospholipase C activity; the C terminal domain, which is similar in structure to eukaryotic C2 domains, binds phospholipid.



The structure of *C. perfringens* -toxin, a two-domain protein

Proteins containing many domains are termed mosaic proteins. Folds are generally classified at the domain level: i.e. each domain in a multi-domain protein is given a separate fold classification.

TERTIARY AND QUATERNARY STRUCTURE

The term tertiary structure is used to refer to the fold of any single polypeptide chain, which may consist of one or more domains.
The biologically functional form of some proteins is as an aggregation of several polypeptide chains. The association of two or more chains together to form a functional protein is referred to as the quaternary structure of the protein. Sometimes the association is between multiple copies of the same chain (or very similar chains), as is the case with haemoglobin. Other proteins consist of many different polypeptides, such as cytochrome c oxidase (13 chains).

Although the structures of several thousand proteins are known, there are only a few hundred different protein folds. Several groups of workers have developed hierarchical models for classification of protein folds. These are not always 100% identical, although the principles adopted are very similar.

In all classification schemes, folds are first classified into large groupings - classes - depending on the number and type of secondary structure units they contain.

The main classes are

- All-alpha
- All-beta
- Alpha and beta

Some classification schemes distinguish between alpha and beta structures where the helices and beta strands alternate and those where they tend to cluster into regions. In the SCOP database, described below, the former sub-class is named “alpha/beta” and the latter one “alpha & beta”. A minority of protein folds do not fit into any of these

classes. Many of these are small proteins are stabilised by disulphide bridges or by interaction with metal ions.

The main classes are subdivided further according to the way the secondary structure units are arranged - the **architecture** - and, further, into the number and order of those units - the **topology** or fold. For example, the immunoglobulin fold - seven anti-parallel beta strands arranged in a particular order - is one example of a beta-barrel.

FOLD: MAJOR STRUCTURAL SIMILARITY

Major folds are determined by number, order, and topology of secondary structure elements. Proteins sharing a common fold will not necessarily share a common evolutionary origin, as some ways of packing secondary structure elements together are particularly energetically favourable.
A few folds are very common, with many different examples (often with different functions); these have been termed super-folds. An example of a super-fold is the TIM barrel fold (first observed in triose phosphate isomerase) which contains 8 parallel beta sheets arranged in a barrel like pattern.

Proteins grouped into a **superfamily** have sufficient topological equivalence for a common evolutionary origin to be considered probable. Superfamily members rarely have significant sequence homology, but usually have a similar (if not an identical) function. For example, actin, the ATPase domain of heat shock protein, and hexakinase are grouped into the same superfamily.

Proteins in the same family (sometimes termed a **homologous family**) are similar enough for a clear evolutionary relationship to be assumed. They usually have very similar, if not identical, function. Almost always the evolutionary relationship may be detected from sequence similarity. One exception to this rule is the globin family, where pairwise sequence identity between members may be as low as 15% (which is insufficient for any evolutionary relationship to be assumed).

ALPHA-DOMAIN FOLDS

The smallest of the main classes is the alpha-domain, or all-alpha, class. As its name implies, it contains those proteins where all, or almost all, secondary structure units are alpha helices.
The alpha-domain class can be subdivided into the following architectures:-

Bundles, where the helices lie approximately parallel or anti-parallel to each other
Non-bundles, where the angles between adjacent helices are neither approximately 0° or approximately 180°. Globins have this architecture.
Few secondary structures, an architecture comprising proteins with only one or two helices.

ALL-BETA FOLDS

This class comprises proteins where all, or almost all, secondary structure units are beta strands. There may be one or even more alpha helices on the periphery of the structures. Most of the strands form anti-parallel beta sheets.

There are many all-beta architectures; some of the most common are:-

Barrels, which contain a single anti-parallel beta-sheet folded into a closed "barrel-like" structure. These may contain from 6 to 16 strands and are subdivided according to the topology of the connections between the strands. Types of barrel include

- up-and-down barrels (the simplest topology)
- Greek key barrels (which includes the immunoglobulin fold)
- "Jelly roll" barrels

Sandwiches with two beta-sheets packed together in a layered arrangement
Less commonly found architectures include **prisms**, **trefoils**, **propellers**, and the **beta-helix**.

Alpha and Beta Folds

The largest of the three main classes consists of proteins with a reasonable proportion of both helices and strands.

Alpha / Beta Folds

These proteins consist of alternating helices and strands, with the strands forming parallel or mixed sheets. Many enzymes and nucleotide binding proteins belong to this class. Architectures belonging to this category include

- Alpha/beta barrels, which consist of an inner beta-barrel surrounded by an outer ring of helices, very approximately collinear with the beta strands. The common "TIM" barrel fold is a subdivision of this family.

Alpha + Beta Folds

These proteins contain both helices and strands, but these are spatially separated. The strands are more likely to form anti-parallel sheets. Common proteins with folds in this category include lysozyme and cysteine proteases.

Other Folds

About 5% of all proteins fall outside these main classes. Most of these are small proteins or those with few secondary structure elements. Small proteins may not have a stable hydrophobic core, in which case it is likely to be held together by co-ordination to metal ions or by disulphide bonds.

You are interested in finding out whether the structure of the PAX6 protein is known. Even if it is not, structures of related proteins, or of proteins homologous to a single domain of PAX6, may be available.

Go to the PDB site at http://www.rcsb.org Click on "SearchLite**"** (in the panel on the right hand side of the screen, underneath the text box). Enter **pax** in the text box and click "Search**".** This should pull out structures of proteins related to human pax6. There may be some "false positive" matches.

How many of these matches are false positives? How accurate are was this search criterion?

Go back to the "Search" screen and enter **homeobox** in the text box. This should give more useful matches. If you scroll right down to the bottom you will see our friend PAX6. Press the red "Explore**"** link next to it.

Four-character PDB code of pax6 ……..

Examine the information in the main entry. What information, apart from the co-ordinates themselves, can be retrieved from this first screen? Does this structure file contain bound DNA?

We're going to download the structure file for PAX6. The four letter PDB code for human pax6 is **6PAX**. We'll download the structure file and view it using RasMol.

Select "Download/Display File**"** from the menu on the left hand side of the Protein Explorer screen. Save the file to disk, in a convenient location, as an uncompressed file complete with coordinates (select the relevant box in the download table – should be a text file). Remember where you put it! A good trick is to save it in the same directory as you will be saving RasMol into.

If you are using a local installation of RasMol, you can skip over the next section. However, if you do not have it installed on your machine, you must download it.

Go to the Rasmol homepage at http://www.umass.edu/microbio/rasmol/index2.htm and scroll down to approximately the middle of the page. Follow the link to "Getting and installing Rasmol or Chime…" This will take you "RasMol, the program itself". Follow the link for "Getting and Installing Rasmol". For a machine running Windows, and "Download RasWin for PC/Windows". At the top of the page follow the link to "Get 32 bit RasWin" and save it onto your desktop.

An icon representing a molecule of three atoms should appear. Double click on this to see a viewer window and a command line tab. Drag and drop your 6PAX.pdb file into the black viewer window to see the protein molecule.

Use the display and colour menus to investigate different representations of the protein. Rotate the molecule by using different combinations of the mouse buttons.

Different mouse button combinations do different things. Holding the left button down will rotate your protein. Holding the right one down will move it up and down. Holding the shift key and left mouse button will allow you to zoom in and out.

SCOP, or **S**tructural **C**lassification **o**f **P**roteins, is a hierarchical database of protein structure, which aims to provide a broad overview of all known folds. It can be used to determine the close relatives of any protein and can be found at

http://scop.mrc-lmb.cam.ac.uk/scop/. The SCOP classification treats the alpha / beta and alpha + beta categories as separate top-level classes.

Go back to the main PDB Structure Explorer page for the structure you have been examining visible in your browser. Go to the "Structural Neighbours**"** page (linked from the left-hand menu) and select "SCOP**".** Explore the classification of your protein. What other structures have similar folds?

The CATH (Class, Architecture, Topology, Homology) database, developed by Janet Thornton and her colleagues at University College London, is automatically derived and can be found at http://www.biochem.ucl.ac.uk/bsm/cath_new/index.html. It uses four top level classes: alpha, beta, alpha and beta, and "other". Professor Thornton's group has derived a diagrammatic representation of the number of protein folds represented in each class, architecture and topology. This shows that, overall, the alpha and beta class is the largest; and the "other" class by far the smallest. However, the fold distribution is different for proteins of different functional types:-

- **DNA binding proteins** are more likely (than average) to have alpha-domain folds
- **Carbohydrate binding proteins** are more likely (than average) to have all-beta folds
- **Enzymes** are more likely (than average) to have alpha / beta folds

Go back to the Structural Neighbours page. Select **CATH** and Explore the place of your protein in the CATH structural hierarchy. Are there differences between the classification of this protein fold in **SCOP** and in **CATH**?

You know that a single point mutation of one alanine residue in the N terminal domain of PAX6 is responsible for some eye defects. What can you learn, from the PAX6 structure, about how a mutation at that position will affect the function of this protein? The sequence you downloaded and worked on in the sequence alignments chapter carried a mutation in codon 33.

Firstly, use the PAX6 Mutations Database to find out more about this mutation.

Go to http://www.hgu.mrc.ac.uk/Softdata/PAX6/ and select "Tables and Stats of Allelic Variants" Select the table giving information about single point mutations (substitutions). Scroll down the table until you come to a mutation involving A33 (look in the "trivial name" column). What amino acid is this residue mutated into? What types of eye disease does this mutation cause? What else can you learn about this mutation? How does this compare with what you discovered from the SNPs in ENSEMBL[59]?

This mutation is in the larger, N-terminal paired box (PAX) domain of PAX6. You should have already examined the structure of this domain bound to DNA: its PDB code is 6PAX. We will highlight this alanine residue in Rasmol to determine its position and predict the likely effect of the mutation.

[59] This wasn't registered as a SNP in Ensembl, because, it is, in fact, a disease causing mutation.

Residue numbering schemes in the PDB are not always identical to those in SwissProt. Entries in sequence databases are quite often longer precursor sequences. Conversely, the N and C termini of protein chains are often difficult to determine. If you compare the sequence of the protein in PDB entry 6PAX with the SwissProt entry PAX6_HUMAN you will see that residue A33 in the protein sequence is equivalent to residue 30 in the PDB file. Of course, we will focus on this residue!

Display your pax6 structure as a "ribbon" or a "cartoon". Go to the Rasmol command line window and type **select ala30A**[60]**,** hit return and type **colour red.**

Examine the position of residue 30. How close is it to the protein's DNA binding site? What secondary structure element is this residue in? Why do you think that mutating this residue to proline would disrupt the structure and function of PAX6?

There is much more help available on using Rasmol, it's an extremely powerful program. If you have some extra time, go to the Rasmol home page at http://www.umass.edu/microbio/rasmol and follow through some of the tutorial material there.

If you have time left then please take a look at: http://www.ebi.ac.uk/msd/education/Tutorial.html

That's the end of our tour of protein structure tools. We hope you have enjoyed it and found it useful. We have only been able to show you a few of the tools available, but we encourage you to explore on your own, perhaps following links from some of the web pages we have shown you.

[xix] Kyte J. , Doolittle R..F., (1982) J. Mol Biol 157, 105-132

[xx] Sweet R.M., Eisenberg D., (1983) J. Mol. Biol. 4 79-488

[xxi] Eisenberg D., Weiss R.M., Terwilliger T.C., (1982) Nature 299 371-374

[xxii] Garnier, Ogusthorpe, Robson (1978) J. Mol Biol 120 97-120

[60] This tells RasMol that the residue we are interested in is residue number 30, which is an alanine residue, and it is in chain A of the protein.

Further Practicals

Now to implement you knowledge! If you have a sequence of your own, then you may have been working on it throughout the course, or you may want to work on it now. If you do not, however, you may like to take the "secret.seq" file that has been placed in your training account and work on that.

You will find a file "secret.seq" in your user account in the Practical directory. This contains part of a nucleotide sequence. Your task is to try and find out:

- *The name of the gene*
- *The location of the gene*
- *The protein encoded by this gene*
- *The function of this protein within the cell.*
- *The organism to which this gene belongs*

Does the final protein (or any part of it) have a structure that has been solved?

Has a deficiency in this gene been reported as causing disease?

Does it exist in other organisms?

- *Design primers to amplify a suitable domain in the protein*

Do they have a high GC content?

What temperature do they melt at?

- *Align this protein sequence with others from the database*

How much of is conserved across organisms?

Are there any regions of high conservation? Did you expect any?

Remember when you are doing this exercise, trusting one program is not always a good idea, so try to use several. You may work in pairs or groups if you find it easier.

Appendix I: Sequence symbols

Nucleotide symbols, their complements, and the standard one-letter amino acid symbols are shown below in separate lists. The letter codes for amino acid codes and nucleotide ambiguity were proposed by IUB (Nomenclature Committee, 1985, Eur. J. Biochem. 150; 1-5)

NUCLEOTIDES

The meaning of each symbol and its complement are shown below.

Nucleotide notations

IUB/GCG	Meaning	Complement
A	A	T
C	C	G
G	G	C
T/U	T	A
M	A or C	K
R	A or G	Y
W	A or T	W
S	C or G	S
Y	C or T	R
K	G or T	M
V	A or C or G	B
H	A or C or T	D
D	A or G or T	H
B	C or G or T	V
X/N	G or A or T or C	X
.	not G or A or T or C	.

AMINO ACIDS

Here is a list of the standard one-letter amino acid codes and their three-letter equivalents. The synonymous codons and their depiction in the IUB codes are shown. You should recognise that the codons following semicolons (;) are not sufficiently specific to define a single amino acid even though they represent the best possible back-translation into the IUB codes!

Amino Acid NOTATION

Symbol	3-letter	Meaning	Codons	IUB Depiction
A	Ala	Alanine	GCT,GCC,GCA,GCG	!GCX
B	Asp,Asn	Aspartic, Asparagine	GAT,GAC,AAT,AAC	!RAY
C	Cys	Cysteine	TGT,TGC	!TGY
D	Asp	Aspartic	GAT,GAC	!GAY
E	Glu	Glutamic	GAA,GAG	!GAR
F	Phe	Phenylalanine	TTT,TTC	!TTY
G	Gly	Glycine	GGT,GGC,GGA,GGG	!GGX
H	His	Histidine	CAT,CAC	!CAY
I	Ile	Isoleucine	ATT,ATC,ATA	!ATH
K	Lys	Lysine	AAA,AAG	!AAR
L	Leu	Leucine	TTG,TTA,CTT, CTC,CTA,CTG	!TTR,CTX,YTR;YTX
M	Met	Methionine	ATG	!ATG
N	Asn	Asparagine	AAT,AAC	!AAY
P	Pro	Proline	CCT,CCC,CCA,CCG	!CCX
Q	Gln	Glutamine	CAA,CAG	!CAR
R	Arg	Arginine	CGT,CGC,CGA, CGG,AGA,AGG	!CGX,AGR,MGR;MGX
S	Ser	Serine	TCT,TCC,TCA, TCG,AGT,AGC	!TCX,AGY;WSX
T	Thr	Threonine	ACT,ACC,ACA,ACG	!ACX
V	Val	Valine	GTT,GTC,GTA,GTG	!GTX
W	Trp	Tryptophan	TGG	!TGG
X	Xxx	Unknown		!XXX
Y	Tyr	Tyrosine	TAT, TAC	!TAY
Z	Glu,Gln	Glutamic, Glutamine	GAA,GAG,CAA,CAG	!SAR
*	End	Terminator	TAA, TAG, TGA	!TAR,TRA;TRR

Appendix II: list files

A list file is a text file of sequence file names or database references. It is an excellent and flexible way for you to produce a specific mini-database, e.g. to do ***fasta*** searches against or look for patterns in.

Each entry is on a separate line. These can be sequences either in your Unix account, in a sequence database or even in another list file. You can even put in comments if you preface them with, e.g. a trivial example:

```
embl:hsfau          ! an entry in the embl database
rabbit.seq          ! a file in your directory
swissprot:pax6*     ! all SwissProt entries whose identifier starts with "pax6"
@hedgehog.list    ! another list file called "hedgehog.list"
```

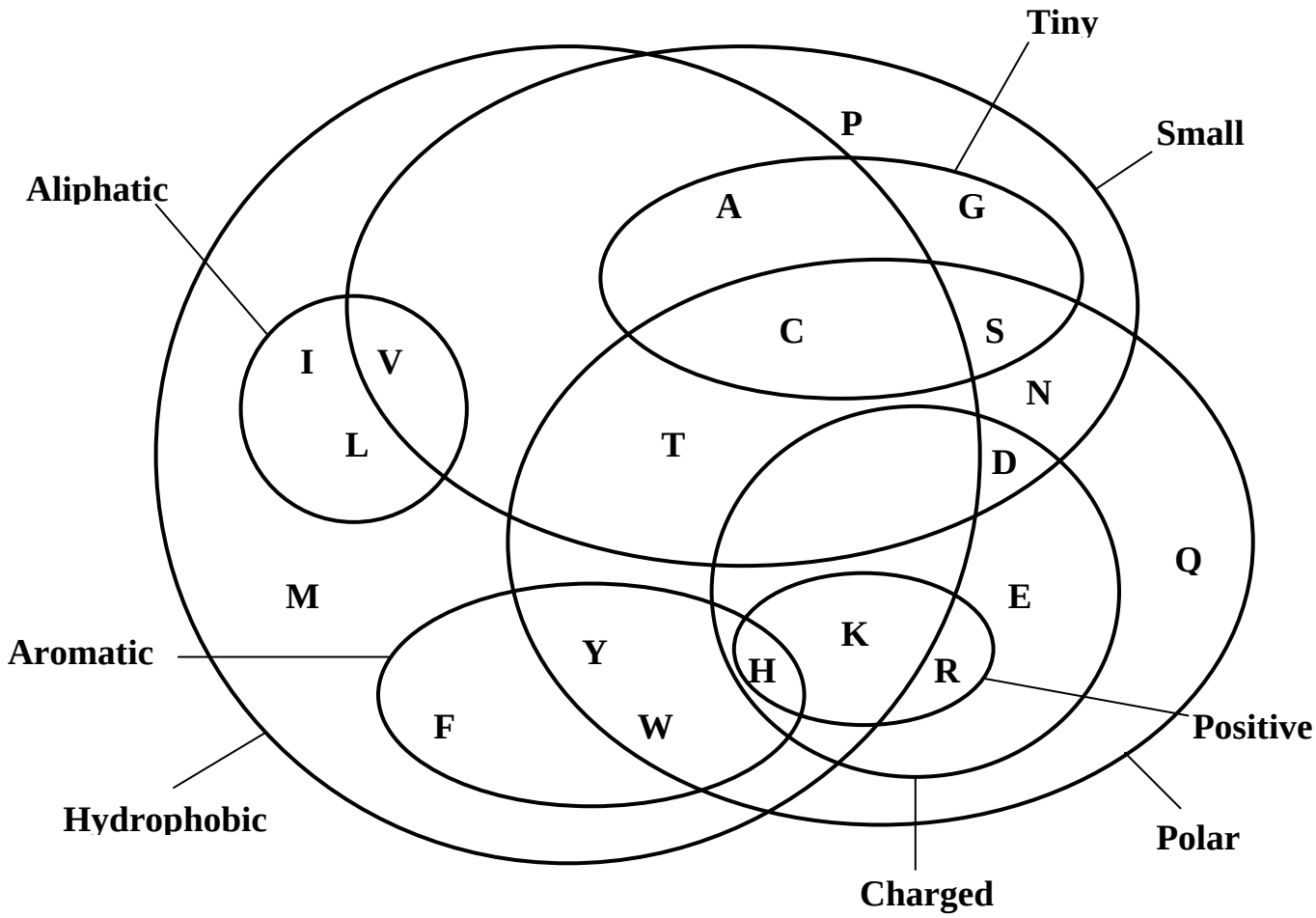
You can use a text editor like ***pico*** or ***nedit*** to create and edit your list file. You may also edit a file in Jemboss. You can make use of simple Unix commands to add the sequences in your directory to a list file:

Create a list file:	***ls *.seq > new.list***
Add sequences to an existing file:	***ls ins*.seq >> insulin.list***
Or from another directory:	***ls /people/nobel/*.seq > prize.list***

The **@** symbol in front of your list file tells EMBOSS programs that the file is a file of filenames and not a sequence. e.g. **seqret @listfile**

Appendix III: Amino Acid Properties

Relationship of amino acid properties, which are important in the determination of protein tertiary structure.



Appendix IV: UNIX Commands

UNIX is an operating system found on many large processors, and is the language used to activate many bioinformatics programs. ***Directories*** are set up in UNIX as a tree, in much the same way that folders are organised in Windows. Each directory is a part (or a ***subdirectory***) of a larger directory and all directories are initiated from a main administration type directory (or ***root***, generally denoted by a slash ***/*** in front of the name of the first or top directory in the tree).

The directory you are working in is known as your working directory. To find out your current working directory, type **pwd** (print working directory) at the unix% prompt.

Every time you log on with your username and password, you will automatically start your UNIX session in your home directory. This is also known as your ***account***.

You may wish to move around directories and will need the **cd** (change directory) command. You may move up the directory tree using the **cd** command and two dots (**..**), or down again by typing **cd** followed by the name of the directory you wish to move into. Each directory is separated by a forward slash (***/***). A single dot (**.**) indicates the directory you are currently in.

PLEASE NOTE: *There must always be a space between the* ***command*** *and its* ***argument*** *(e.g. cd* ***..*** *or cd directory name). This is true for all UNIX commands, whatever you are doing. UNIX is also case sensitive, which means that if you type CD, UNIX will respond with an error message and will not carry out your command.*

The order (or ***string***) of directories and subdirectories from root that you need to type to access your file specifies its ***path***. The actual filename is the last name in this string and is separated from the directory name by a slash (***/***).

By default, you start off in your home directory, which is generally something like /people/mrnaxx, and represents the trunk of your directory tree - with all your subdirectories being created within it. If you get lost on your travels through several working directories, simply type **cd** and you will be transferred back into your home directory.

Listing files

Directories do not just contain subdirectories. They also contain files to house your important documents and data. These are analogous to the files contained within folders in Windows. You may look at which files you have in a directory by typing **ls** (list). This will give you several columns containing the names of the files you have in your working directory. If you know your files well, or have only a few, this may be sufficient information for you. You may, however, wish to know more. This can be done by adding extra commands (called ***options*** or ***switches*** or ***flags***) to the command line.

There are two ways of listing the files held in a certain directory. You may either move to the directory in question and list the files using the **cd** and **ls** commands, or you could list the files by giving the ***pathname*** of the directory as an option.

If you wish to see details on a specific file, you may include a filename (on its own or as part of the pathname) on the command line after the switch. Don't forget a space!

File Permissions

These can be viewed by the ls –l command for either files or directories. A file is denoted by a dash (-) at the front of the permissions guide, and a directory is denoted by d. The three characters xwr refer to execute (the ability to run (***execute***) files), ***write*** (the ability to edit or delete a file) and ***read*** (the ability to read the contents of a file) permissions (or ***privileges***) respectively. The first group of three letters represents the user's (i.e. your) permissions. The next three indicate the privileges for a specified group, of which you are a member. UNIX provides facilities for several people to collaborate on a specific project and everyone is a member of the same group. If you want to know which group you are in, you could ask the computer to display the environment variable **GROUP** by typing **echo $GROUP**. Everyone must be a member of at least one group, and the members of this group can be given access to files that remain unauthorised for users outside a particular group. The third grouping of three letters represents the permission for the rest of the world.

File permissions are often used as a means of providing security for your data.

Changing file permissions

You may alter permissions for files that you own. To do this, you must use the **chmod** (change mode) command together with which permission you wish to alter (u=user; g=group; o=other; a=all) plus (or minus) what you would like to add (take away) access for (i.e. read (r); write (w) or execute (x) permission. The final command on the line must be the name of the file you wish to change permission for.

Creating and Removing Directories

You will not always wish to put your files into your home directory, in which case you must create your own subdirectories. This is especially important for organising your work. This is done by typing in **mkdir** (make directory) as a command, followed by a space and the name you want to call your new directory. Alternatively, if you wish to create your directory in a different directory to your current working directory (i.e. the one you are in), you must give the full pathname of where the new directory should go

Each time a space is used UNIX understands a new command or ***argument***. Therefore names of files or directories must either be one word; or several words separated by an underscore (_).

Should a directory be empty or old, you may wish to remove it. If a directory still has files in it, they must be removed first (see section 4.4.3), and then you may remove the directory using the **rmdir** (remove directory) command.

Removing files

Just as with directories, files can also be manipulated. They can be removed using the **rm** (remove) command followed by a space and the filename, or pathname of the file you wish to remove. Before removing a directory, all files in that directory must be removed first (see section 4.3). You may use a switch for this command. Typing **rm –i** filename will prompt you for a final decision on whether you want to delete the file. This is a safety option, as it is very easy to delete files in UNIX and there is no way to recover them after deletion.

Copying files

You may also copy files. This may be because you want to copy a file from one directory to another, or because you want to duplicate a file in order to use it as a template for something else. The command for this is **cp** (copy) followed by a space, then the name of the file to be copied (or the ***first argument***), followed by a space and then the name you wish to call the copied file (or the ***second argument***). If you are duplicating a file in a second directory, you may wish to retain the same name. If you are duplicating the file in the same directory, however, it may be wise to change the name of it to something else as UNIX will overwrite files of the same name without alerting you to the fact.

List files

The majority of files contain data that may be considered a single entity. The information they contain may be for your information or may be a single input into another program (e.g. when using EMBOSS). Files may contain various extensions (e.g. .txt; .seq; .fasta) and contains a specific set of data.

There is, however, another group of files, generally called list files. They contain lists of specific information, for example several restriction enzymes, or EMBL accession codes. List files are sometimes needed as input for programs such as GCG and EMBOSS and the filename is preceded by @. Thus to input the list of restriction enzymes saved in the file enz.dat, you would type **@enz.dat** as an argument on the command line. However, these files are just text files as far as UNIX is concerned, and are only useful as input for those programs named above.

Appendix V: Websites

Search and Retrieval

http://srs.ebi.ac.uk	SRS
http://www.ncbi.nlm.nih.gov/Entrez/	Entrez
http://www.expasy.ch/	Expert Protein Analysis System

Bibliographic Databases

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed	PubMed
http://www.gen.tcd.ie/pubcrawler/	PubCrawler

Databases

http://www3.ncbi.nlm.nih.gov/Omim/	OMIM – Online Mendelian Inheritance in Man
http://gdbwww.gdb.org/	GDB – Genome Database
http://bioinfo.weizmann.ac.il/cards/	GeneCards
http://www.ensembl.org	Ensembl
http://www.ebi.ac.uk/asd	Alternative Splicing Database

Clinical Databases

http://www.hgmd.org	Human Gene Mutation Database

More Databases

http://www.genome.ad.jp/kegg/kegg.html	KEGG (Kyoto Encyclopaedia of Genes and Genomes)
http://wit.mcs.anl.gov/WIT2/	WIT (What Is There?)
http://www.tigr.org	The Institute of Genome Research
http://www.thearkdb.org	The ARK database
http://www.reactome.org	Reactome Database

Protein Databases

http://www.sanger.ac.uk/Software/Pfam/	Pfam – Protein Domains
http://smart.embl-heidelberg.de/	SMART – cell signalling
http://www.ebi.ac.uk/interpro/scan.html	INTERPRO
http://www.rcsb.org/	Protein Data Bank
http://scop.mrc-lmb.cam.ac.uk/scop/	SCOP – Structural Classification of Proteins
http://www.biochem.ucl.ac.uk/bsm/cath/	CATH – Class, Architecture, Topology, Homology Database

EMBOSS

http://emboss.sourceforge.net	EMBOSS homepage

Miscellaneous

http://www.ebi.ac.uk/genomes/mot/	Genome Monitoring Table
http://www.compbio.dundee.ac.uk	Jpred