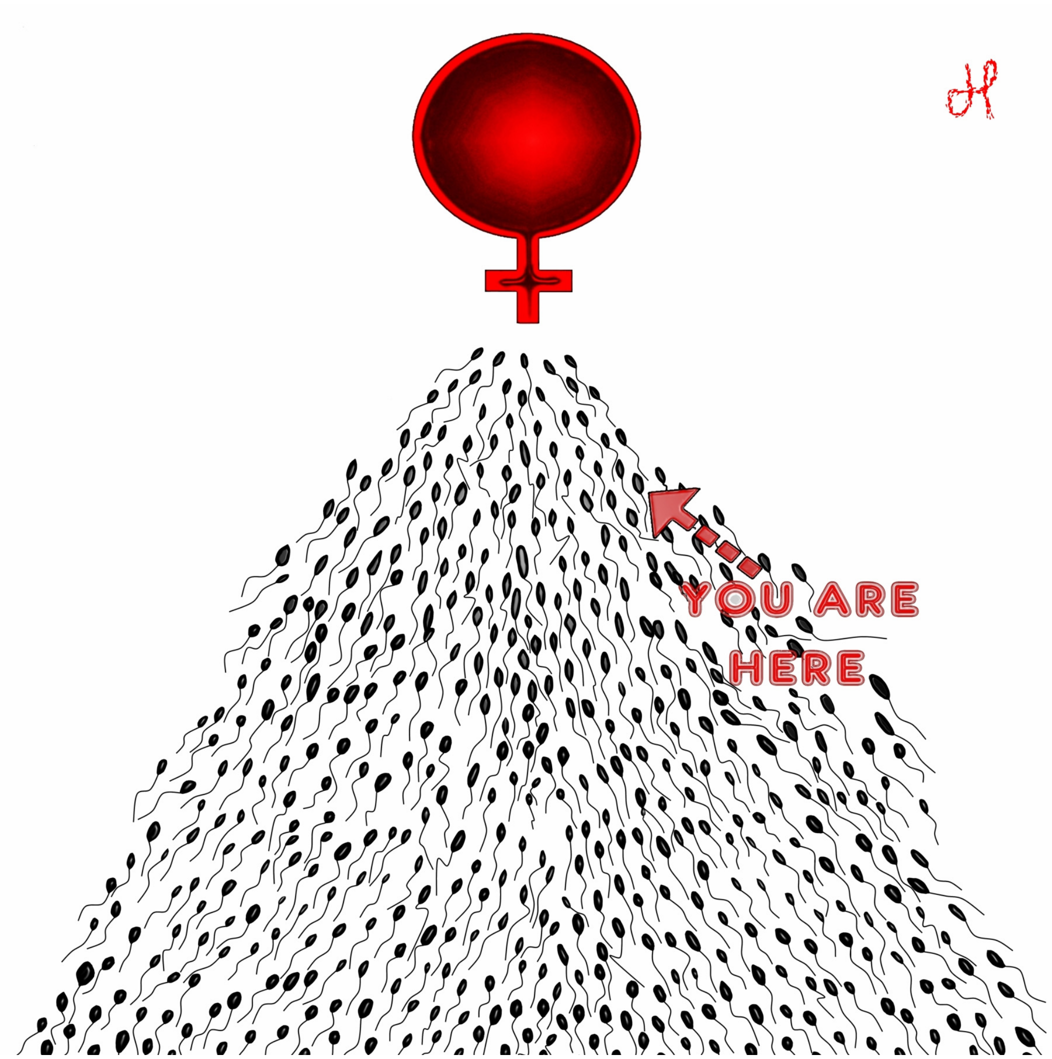
BIOINFORMATICS
Action Labs

Jean-Louis Lassez, Ryan Rossi, and Stephen Sheel

Preface

Bioinformatics is the application of computational techniques and tools to analyze and manage biological data. This book provides an introduction to bioinformatics through the use of Action Labs. These labs allow students to get experience using real data and tools to solve difficult problems. The book comes with supplementary software tools and papers. The labs use data from Breast Cancer, Liver Disease, Diabetes, SARS, HIV, Extinct Organisms, and many others. The book has been written for first or second year computer science, mathematics, and biology students. The supplementary software and papers can be found at http://www.kibazen.com/binf



Jean-Louis Lassez: "Life is Pachinko" at the Kinsey Institute Museum

Table of Contents

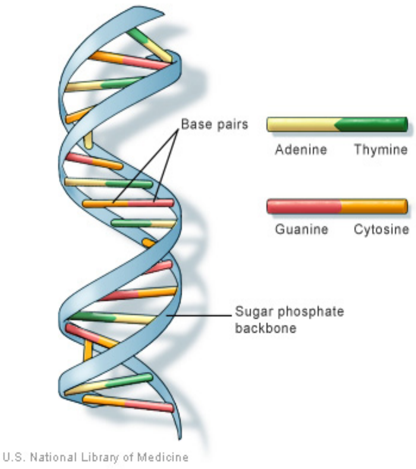
The software and supplementary papers are located at http://www.kibazen.com/binf

Chapter 1

Introduction to Bioinformatics

What is Bioinformatics

Background:



U.S. National Library of Medicine

What is Bioinformatics? It depends on who you are talking to. A geneticist, a biologist, a mathematician, a CEO of a pharmaceutical company and a computer scientist all would have related, but different, opinions as to what Bioinformatics is.

Purpose: This lab introduces various aspects of Bioinformatics, its scientific basis, its techniques and its applications.

Resources: There are many excellent resources on Bioinformatics that can be found on the web. Visit, for instance, the tutorial located at: *http://www.ebi.ac.uk/2can/bioinformatics/bioinf_what_1.html*

Key Terms:

- Genome
- Gene
- Protein
- Amino Acid
- DNA
- Codon
- Prokaryote
- Eukaryote
- Archaea
- RNA

Directions: Read the tutorial in the resources (or equivalent) thoroughly.

Exercises:

1. Give a concise, yet precise, definition of Bioinformatics.

2. What are the biggest challenges facing Bioinformatics? Why do you think this is the case?

3. Give a list of the main biological databases that can be accessed on the internet.

4. What are the differences in the functions of the various biological databases?

5. Name the categories of the major data analysis tool.

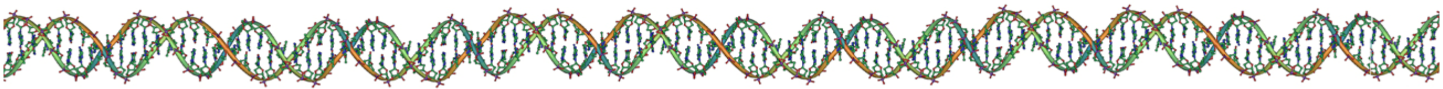
6. How are the sequence analysis tools used in Bioinformatics?

7. Make a list of the most important real-world applications for Bioinformatics. Rank your choices from 1-10 and ***justify why***, in your view, the application received its ranking (As the ranking is subjective and tied to your taste or expertise, what matters most is not the ranking you choose but the justifications you give).

References:

1. *European Molecular Biology Laboratory (EMBL)."What is Bioinformatics?". <http://www.ebi.ac.uk/2can/bioinformatics/bioinf_what_1.html>.*
2. *Genetic Home Reference: Your Guide to Understanding Genetic Conditions [Internet]. Bethesda, MD: United States National Library of Medicine, National Institute of Health [modified: 2009 July 31]. [Illustration], DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone.;[cited 2007 July][about 3 screens]. Available from: http://ghr.nlm.nih.gov/handbook/basics/dna.*

Exploring Frameshifts



Background:

A **frameshift mutation** (also called a **frameshift** or a **framing error**) is a genetic mutation that inserts or deletes a number of nucleotides that are not evenly divisible by three from a DNA sequence. Due to the triplet nature of gene expression by codons, the insertion or deletion disrupts the reading frame, or the grouping of the codons, resulting in a completely different translation from the original. The earlier in the gene the deletion or insertion occurs, the more altered the gene product will become.

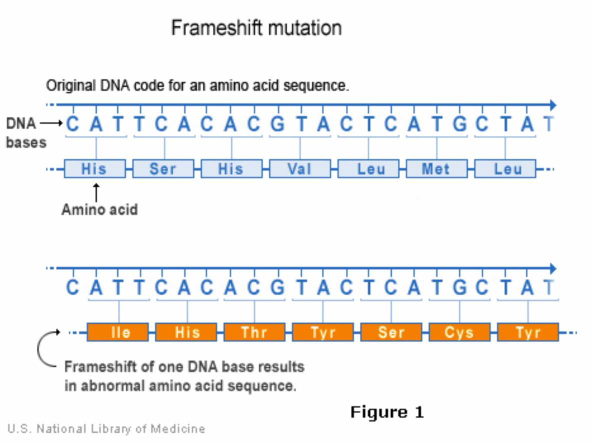

Frameshift mutation
Original DNA code for an amino acid sequence.
DNA bases
C A T T C A C A C G T A C T C A T G C T A T
His Ser His Val Leu Met Leu
Amino acid
C A T T C A C A C G T A C T C A T G C T A T
Ile His Thr Tyr Ser Cys Tyr
Frameshift of one DNA base results in abnormal amino acid sequence.

Figure 1

U.S. National Library of Medicine

Frameshift mutations frequently result in severe genetic diseases.

Purpose:

This lab is intended to analyze how different mutations affect sequences.

Resources:

BLAST: http://www.ebi.ac.uk/blast
Transeq: http://www.ebi.ac.uk/emboss/transeq/

Key Terms:

- Frameshift mutation
- Codon
- Insertion
- Deletion

Directions:

Make sure you have an understanding of the keywords above, and then complete the exercises below.

Exercises:

1. How many ways can we parse this DNA subsequence into a potential coding frame?

………TACGGAAGTTCACTGCAATCAGTTGACTGAGGACTG……

2. Assume that the coding frame for the subsequence is in fact:

TAC/GGA/AGT/TCA/CTG/CAA/TCA/GTT/GAC/TGA/GGA/CTG

Translate this subsequence into a sequence of amino acids. (You can do it by hand using the table for the genetic code, but using the **Transeq** program will be easier and faster.)

3. Now an insertion mutation has happened resulting in the following sequence:

TACGGTAAGTTCACTGCAATCAGTTGACTGAGGACTG

Translate this new sequence into a sequence of amino acids.

4. Next divide the sequence, which has a deletion mutation, into codons:

```
TACGAAGTTCACTGCAATCAGTTGACTGAGGACTG
```

Translate this new sequence into a sequence of amino acids.

5. Are there significant changes in the translation? Explain the reason for the differences in the translation from questions 3 and 4.

6. Run the **BLAST** program on the three DNA sequences above. Do the frameshifts cause a misclassification in the organisms identified by **BLAST** when compared to the original DNA sequence?

7. Visit the site: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=442341. Read the abstract for the article. Summarize the authors' main point.

References:

1. *Schach,B.G., Yoshitake,S. and Davie,E.W.," Hemophilia B (factor IXSeattle 2) due to a single nucleotide deletion in the gene for factor IX", The Journal of Clinical Investigation, no. 4(1987), <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=442341>.(12 September 2006).*
2. *The European Bioinformatics Institute (EBI)."Blast @ EBI".< http://www.ebi.ac.uk/blast>.*
3. *The European Bioinformatics Institute (EBI). "EMBOSS Transeq". <http://www.ebi.ac.uk/emboss/transeq/>.*
4. *Genetic Home Reference: Your Guide to Understanding Genetic Conditions [Internet]. Bethesda, MD: United States National Library of Medicine, National Institute of Health [modified: 2009 July 31]. [Illustration], Frameshift mustation.;[cited 2007 July][about 3 screens]. Available from: http://ghr.nlm.nih.gov/handbook/basics/dna.*

Bioinformatics Tools

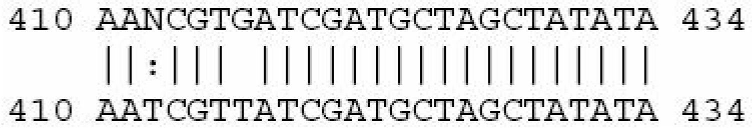
Background:

Molecular Sequence Alignment Tool

The Right Tool for the Right Job!

Sequence similarity is assessed, in a first instance, by comparing the first, and then second, then third, etc. letters from each sequence and scoring positive points when there is a match and negative points when there is no match. The problem becomes more complex when we have gaps, which occur when one sequence may have been subjected to one or more insertion or deletion mutations. This lab provides an introduction to sequence alignment, which is the first fundamental tool in the study of biosequences.

Here is an example of alignment:



```
410 AANCGTGATCGATGCTAGCTATATA 434
    ||:||| ||||||||||||||||||
410 AATCGTTATCGATGCTAGCTATATA 434
```

The numbers at each end of the sequences correspond to the nucleotide number in the original sequence. The (|) means a match, while (:) means a gap and no connector means a substitution, as we see on the seventh pair.

Purpose:

This lab introduces Molecular Sequence Alignment tools.

Resources:

For this exercise use the software located at: http://xylian.igh.cnrs.fr/bin/align-guess.cgi.

Key Terms:

- Genome
- Sequence Alignment
- Mutation
- Insertion/deletion/substitution
- Gap Penalty
- E-score
- Nucleotide

Directions:

As will often happen with online bioinformatics resources, links, such as the one in the resources may or may not work. It is part of this lab to train you in searching the net until you find the appropriate information. Once you are at the website, or another equivalent one, run the alignment tool with the sequences below.

First Sequence:
AACGCCCAGGGTTTCCCAGTCACGACGTTGTAAAAGCGACGGCCAGTGCCA

Second Sequence:
AACGCCAGGGTTTTCCCAGTCACGACGTTGTAAAACGACGGCCAGTGCCA

Exercises:

1. What percentage identity do these two sequences have?

2. What is the gap penalty and where is/are the gap(s) in the alignment?

3. What is the score of the alignment?

The next exercises make use of an ORF finder and the sequence below.

The link to ORF Finder is: http://www.ncbi.nlm.nih.gov/gorf/gorf.html

Give the following sequence as input to the program:

```
TTGCTGTGTGAGGCAGAACCTGCGGGGGCAGGGGCGGGCTGGTTCCCTGGCCAGCCATTGGCAGAGTCCGCAGGCTAGGG
CTGTCAATCATGCTGGCCGGCGTGGCCCCGCCTCCGCCGGCGCGGCCCGCCTCCGCCGGCGCACGTCTGGGACGCAAGGC
GCCGTGGGGGCTGCCGGGACGGGTCCAAGATGGACGGCCGCTCAGGTTCTGCTTTTACCTGCGGCCCAGAGCCCCATTCA
TTGCCCCGGTGCTAGCGGCGCCGCGAGTCGGCCCGAGGCCTCCGGGGACTGCCGTGCCGGGCGGGAGACCGCCATGGCGA
CCCTGGAAAAGCTGATGAAGGCCTTCGAGTCCCTCAAGTCCTTCCAGCAGCAGCAGCAGCGCAGCAGCAGCAGCAGCAGC
AGCAGCAGCAGCAGCAGCAGCAGCAGCAACAGCCGCCACCGCCGCCGCCGCCGCCGCCGCCTCCTCAGCTTCCTCAGCCG
CCGCCGCAGGCACAGCCGCTGCTGCCTAGCCGCAGCCGCCCCCGCCGCCGCCCCCGCCGCCACCCGGCCCGGCTGTGGCT
GAGGAGCCGCTGCACCGACCGTGAGTTTGGGCCCGCTGCAGCTCCCTGTC
```

4. What do the colored bars represent in the frames?

5. Which frame does not contain an open reading frame?

6. Which frame has the longest open reading frame?

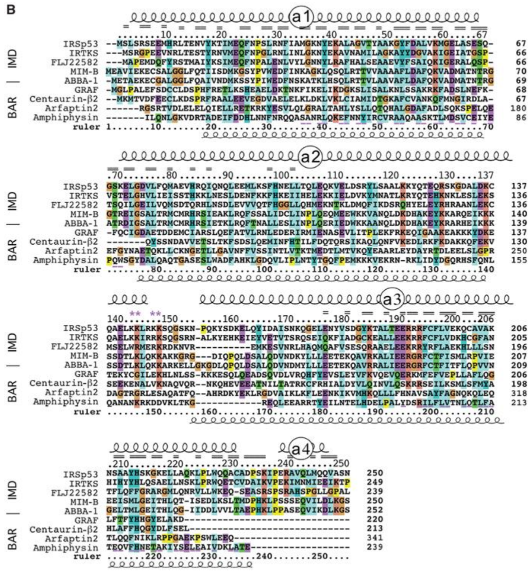
7. Which of these ORF's, if any, correspond to a known gene?

References:

1. *Institut de Génétique Humaine. "ALIGN Query using sequence data". <http://xylian.igh.cnrs.fr/bin/align-guess.cgi>.*
2. *National Center for Biotechnology Information (NCBI). "ORF Finder (Open Reading Frame Finder)". <http://www.ncbi.nlm.nih.gov/gorf/gorf.html>.*

Chapter 2

Introduction to BLAST and FASTA



Database Searching Options

Statistical matrices allow a query sequence to be aligned with matching sequences in the database. The less complex, faster matrices sacrifice a certain degree of match significance. The matrix together with the choice of the program essentially determines the search sensitivity and speed.

Filtering masks regions of the query sequence that has repeats or other low compositional complexity areas. Masking is achieved by replacing the repeats with N's, the IUB code for any base.

The three main public molecular databases are EMBL(Europe), GenBank(US), and DDBJ(Japan). These three databases update each other with new sequences collected from each region, every 24 hours.

Every entry into the database requires a unique identifier that never changes and a version number.

A redundant database is a database where more than one copy of each variant of a sequence may be found. The advantage of a redundant database is that it's much more likely to contain recently discovered sequences. The disadvantage is that the biologically significant results are more likely to be hidden among the large number of reported matches.

Sequence Alignment Programs

BLAST – BLAST is the fastest, but compromises some degree of sensitivity for speed.

FASTA – FASTA is slower, but more sensitive then BLAST.

BLITZ – BLITZ also provides a very sensitive search but is very slow to run.

BLAST and FASTA are the most commonly used sequence alignment programs.

BLAST is the algorithm used by a family of five programs that will align a query sequence against sequences in a molecular database.

BLASTP - Compares a nucleotide query sequence against a nucleotide sequence database.

BLASTN – Compares an amino acid query sequence against a protein 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.

Reading BLAST Output

All BLAST programs produce a similar output consisting of:

- Colored bars are distributed in a way to reflect the region of alignment onto the query sequence. The color legend represents the significance of the alignment scores.
- **E value:** the expect value is the probability that the associated match is due to randomness; the lower the E value, the more specific/significant the match.
- Sequences that are significantly aligned to our query sequence will appear. Their corresponding line descriptions are listed in order of lowest to highest E value.
- Identifiers for the database sequences appear in the first column and are hyperlinked to the associated GenBank entry.
- The Score (bits) is a sum value calculated for alignments using the scoring matrix; the higher the score value, the better the alignment.
- The percent identity is the percent of exact matches between the query sequence and the database sequence, this value also gives the number of nucleotide bases or amino acid residues that are matched in the database sequence versus the query sequence.

Understanding BLAST

Scoring: The alignment for each base in the word is scored: if a nucleotide in the query word exactly matches a nucleotide at the same position in the database word (e.g. A with A), then a positive score is awarded. If the match is good, but not perfect, then a lower score is awarded. The sum score is used to determine the degree of similarity.

PAM matrices: These matrices are most sensitive for alignments of sequences with evolutionary related homologs. The greater the number in the matrix name, the greater the expected evolutionary (mutational) distance, i.e. PAM30 would be used for alignments expected to be more closely related in evolution than an alignment performed using the PAM250 matrix.

BLOSUM matrices: These matrices are most sensitive for local alignment of related sequences typically chosen when trying to identify an unknown nucleotide sequence.

Expect option: Ten is used as a default. This means that 10 matches are expected to be found by chance.

Score Option: The scoring of an alignment can be set by the user. The M parameter is the score awarded when a pair matches; must be a positive integer. The N parameter is the score awarded when a pair doesn’t match. The ratio of M:N determines the degree of evolution that is accepted. The default values for M are 5 and N is 4. Ratio is 5:4 or 1.25.

Directions: Using the information above or similar materials found on the web, answer the following questions.

Exercises:

1. What are the differences between molecular databases and the other databases that you might be more familiar with?

2. What are the three main public molecular databases and what is significant about all three of them?

3. What are the different identifier codes for molecular database entries? How are they useful?

4. What are the advantages and disadvantages of using redundant databases?

5. List the three different sequence alignment programs described above. Explain the advantages and disadvantages of each one.

6. What are the five different BLAST alignment programs? Give some examples as to when they would be used.

7. What are the most significant factors when reading the output of a search? How are they useful?

8. How does the scoring of alignments work?

9. What are the major differences between PAM and BLOSSUM matrices?

References:

1. *Altschul, Stephen, Gish, Warren, Miller,Webb, Myers, Eugene and Lipman, David. "Basic Local Alignment Search Tool". The Journal of Molecular Biology, no. 215 (1990). <http://blast.wustl.edu/doc/blast.html>. (9 September 2007).*
2. *National Center for Biotechnology Information (NCBI). "BLAST Information)". <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html>. (9 January 2007).*
3. *Urbach, Jonathan, "BLAST: Basic Local Alignment Search Tool". Seminar Presentation <http://pga.mgh.harvard.edu/Parabiosys/education/seminars/blast.pdf>. (10 September 2007).*

Exploring BLAST #1

Background: Sequence similarity is used to help find if two sequences have the same ancestor or similar function.

Purpose: This lab is intended to provide a basic understanding of the use of the **BLAST** search engine. **BLAST** is used to analyze both protein sequences and DNA sequences. BLASTN will be used in this lab.

```
TDGTAVVNYAQWESEQAYRVNFG---ADPRSAE-LREALSSLPGLMGPPKAVFMTPRGAILPS-------
RDGEQVVNYAQWRSEADFRAMHADPRLQPHFDY-CRSVSRPKPIFCEVTHSFGATSPEGA---------- 45   59
NNGRHVVNYAQWESEETFRSMHALPELQEHFAF-CRGIATPLSVPCAVAQVFEEAAG------------- 43   57
VDTPGTLNYIQWRSLADLEARYQGQKFQKKTVPLFDQLATSVKLLKTEL-------------------- 25   32
DVRPQYVNIAVWDDEASFRAAVAHPQFPAHAAV-LRALSTSEPTLYRSRQIRVAPGAPAMSRPEGRTT-- 23   30
ERPGQYVNVAEWRDLASFRAAVSHDGFRPHADA-LRALSESRPELYLVR-LRRE-GAPGLDGPASEGEEI 22   27
DKDNSYVNIAVWTDHDAFRRALAQPGFLPHATA-LRALSTSEHGLFTARQTLPEGGDTTGSGHR------ 23   27
DARPQYVNIAVWDDEASFRAAVAHPEFPAHAAA-LRALSTSEPTLYRHRQIRVAPDVPAVSGPGGRTT-- 22   27
HDGSTLLHHSQWASEQAYEAFVKTHRQERVDEIDTAVPGIERLGLGRYRRYRSAAREDR----------- 22   24
TDGTRVLNYAEWESAQAHLDALAAPGDGVGSTTPQWERVQNWPGLTGGGRVSRYDHALGLVPR------- 35   51
LDGTRVVNYAQAQDQAAMQRVFEHLRGNGFLDR-NRALGQAHPGLYEVALTVE----------------- 34   46
EDGQHVLNYTCWRSREDCERAWLAREDAQGPLSAGVWRLGAKSVRFETFLVDAEGC-------------- 26   41
```

Resources: A few **BLAST** sites you can use:
http://www.ebi.ac.uk/Tools/blast2/nucleotide.html
http://www.ebi.ac.uk/Tools/blastall/nucleotide.html
http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- BLASTN
- EMBL
- Bit-score
- ESTs
- Gapped Alignment
- E-value

Directions: Read the sequence analysis notes and complete the exercises below.

1. Copy the query sequence given below:

```
AAGCTTTTGTAAGAATTGCAACTTCTCATATCATACAACCCTAGAAACATCCAATACACCAAAACTAGGAGATGTCAATC
TATTAATCACTTCCATTTAAACTGTCCTCCACAGAAATGCCATCATTAGTCTATCGCGGAAAATATCTCAAAATACCAGC
AAGATTCTCTTCAAGCCAAGTTAGATCACATGACAAGCTGATGGAAAATGCATGCTAATAAAAGCTGCTAAAAAGGCTTT
GCTCCTTGACCGCTGACACTTTCTGGCACCACATTGACACACTCCAGCAACAACGTGTAGAGTCAGTGGCAGTGGATTTA
ATGACTTGAGTACAGTGTGCTGACAGTGCAGGTGTTTGGCCAGCTCTGATGTGCAGTGACTCGATTTAGCAATACGGCTC
TTATTTCTAACACCCAGTTC
```

2. Go to http://www.ch.embnet.org/software/bBLAST.html

3. Select the program: **BLASTN**. This is the **BLAST** program that will compare a nucleotide query sequence against a nucleotide database.

4. Select the database: **EMBL** without Expressed Sequence Tags (ESTs). This is the main EMBL nucleotide database.

5. Ignore the matrix option. It is not used by **BLASTN**.

6. Select sequence input format: Plain Text. The nucleotide sequence above is in plain text or raw format, i.e. it does not contain any header information.

7. Select the following: Gapped Alignment: ON; **BLAST** filter: ON; Graphic Output: ON. These are all ON by default.

8. Paste the query sequence into the specified area.

9. Hit the button: Run **BLAST**.

10. Wait as your query is processed by the server.

11. Examine the output.

Exercises:

1. Based on the report what do you believe the sequence represents?

2. Examine the graph. What does the graph represent? Explain the significance of the colored bars. Which graphical alignment most likely corresponds to the sequence we are looking for?

3. What is the bit-score of our alignment and how is the bit-score calculated?

4. What is the E-value and what does it represent?

5. Place the sequence below that contains the first 142 nucleotides of our original sequence into **BLAST**. Determine the best alignment. Contrast this alignment with the alignment obtained from our original sequence alignment. How do the alignments differ? Explain your answer.

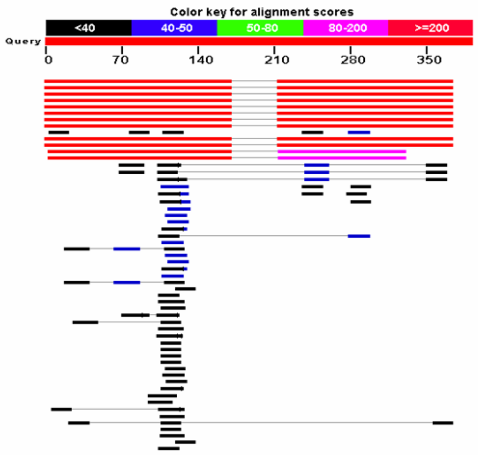
```
AAGCTTTTGTAAGAATTGCAACTTCTCATATCATACAACCCTAGAAACATCCAATACACCAAAACTAGGAGATGTCAATC
TATTAATCACTTCCATTTAAACTGTCCTCCACAGAAATGCCATCATTAGTCTATCGCGGAAA
```

References:

1. *DNA Data Bank of Japan(DDBJ).”BLAST Version 2.2.15”. <http://blast.ddbj.nig.ac.jp/top-e.html>*
2. *National Center for Biotechnology Information (NCBI). “BLAST”. <http://www.ncbi.nlm.nih.gov/blast/Blast.cgi>.*
3. *Swiss EMBnet.org. “Basic BLAST”.<http://www.ch.embnet.org/software/bBLAST.html>.*
4. *Washington University in St. Louis School of Medicine.”WU-BLAST”.<http://blast.wustl.edu/>*

Exploring BLAST #2

Background:



GenBank is a leading nucleotide sequence repository. It is maintained as a consortium between the United States National Center for Biotechnology Information (**NCBI**), the European Molecular Biology Laboratory (**EMBL**), and the DNA Data Bank of Japan (**DDBJ**).

Purpose:

This lab is intended to introduce a more advanced feature of **BLAST.**

Resources:

A few **BLAST** sites you can use:
http://www.ebi.ac.uk/Tools/blast2/nucleotide.html
http://www.ebi.ac.uk/Tools/blastall/nucleotide.html
http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- GenBank
- NCBI
- DDBJ
- Accession Number
- EMBL

Directions:

Use BLAST to answer the exercises below.

Exercises:

1. Find the sequence taken from a **Swiss-Prot** entry whose accession ID is Q96QB1. This can be done using the BLAST server.

Now use BLAST to identify the sequence that corresponds to the accession ID. What is the score and E-value?

2. Copy the sequence given below and use it to run a **BLAST** search:

```
GATTGGCTACGGGCAACTGGTTTCCCCCAGTATGCACAGCTTTATGAAGATTTCCTGTTC
CCCATCGATATTTCCTTGGTCAAGAGAGAGCATGATTTTTTGGACAGAGATGCCATTGAG
GCTCTATGCAGGCGTCTAAATACTTTAAACAAATGTGCGGTGATGAAGCTAGAAATTAGT
CCTCATCGGAAACGAAGTGACGATTCAGACGAGGATGAGCCTTGTGCCATCAGTGGCAAA
TGGACTTTCCAAAGGGACAGCAAGAGGTGGTCCCGGCTTGAAGAGTTTGATGTCTTTTCT
CCAAAACAAGACCTGGTCCCTGGGTCCCCAGACGACTCCCACCCGAAGGACGGCCCCAGC
CCCGGAGGCACGCTGATGGACCTCAGCGAGCGCCAGGAGGTGTCTTCCGTCCGCAGCCTC
AGCAGCACTGGCAGCCTCCCCAGCCACGCGCCCCCCAGCGAGGATGCTGCCACCCCCCGG
ACTAACTCCGTCATCAGCGTTTGCTCCTCCAGCAACTTGGCAGGCAATGACGACTCTTTC
```

```
GGCAGCCTGCCCTCTCCCAAGGAACTGTCCAGCTTCAGCTTCAGCATGAAAGGCCACGAA
AAAACTGCCAAGTCCAAGACGCGCAGTCTGCTGAAACGGATGGAGAGCCTGAAGCTCAAG
```

What do you notice when you examine the results? Was the first answer correct? If not, what accounts for the change in output? Why did the score change?

3. Copy the sequence below, and run a **BLAST** search. Examine the results.

```
CAGATCAACTGCCAGTCTGTGGCCCAGATGAACCTGCTGCAGAAATACTCGCTCCTAAAG
TTAACTGCCCTGCTGGAGAAATACACACCCTCTAATAAGCATGGTTTTAGCTGGGCTGTG
CCCAAGTTCATGAAAAGGATCAAGGTTCCAGACTACAAGGACCGGAATGTATTCGGGGTC
CCTCTGACAGTCAATGTGCA
```

How would you identify this sequence? If the last 10 letters of this sequence were deleted, do the results change in any way? Why or why not?

4. The sequence in the last alignment is clearly unique. What would happen if only the first 15 bases were known? The first 15 bases are `CAGATCAACTGCCAG`.

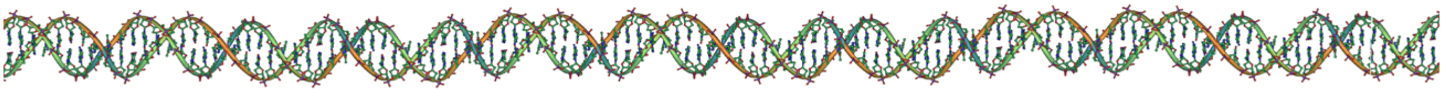
Use **BLAST** to align the above sequence. What are the results? The resulting output may occur because the search criteria needs to be optimized for the search to work effectively. Go back to **BLAST** and click on the "Advanced BLAST" button. Use the default parameters, except you should select the **EMBL** database and the **nr** from the list of **EMBL** divisions. Adjust the E-value to 100. Remember that doing so will reduce the specificity of the search by accepting more chance alignments. Also turn off the **Xblast-repsim** filter. Failure to do so will mask almost the entire sequence with N's. Finally, click the run button.

What are the results and how do you think using advanced **BLAST** helped align our sequence?

References:

1. *Swiss EMBnet.org. "Basic BLAST".<http://www.ch.embnet.org/software/bBLAST.html>.*

Exploring FASTA



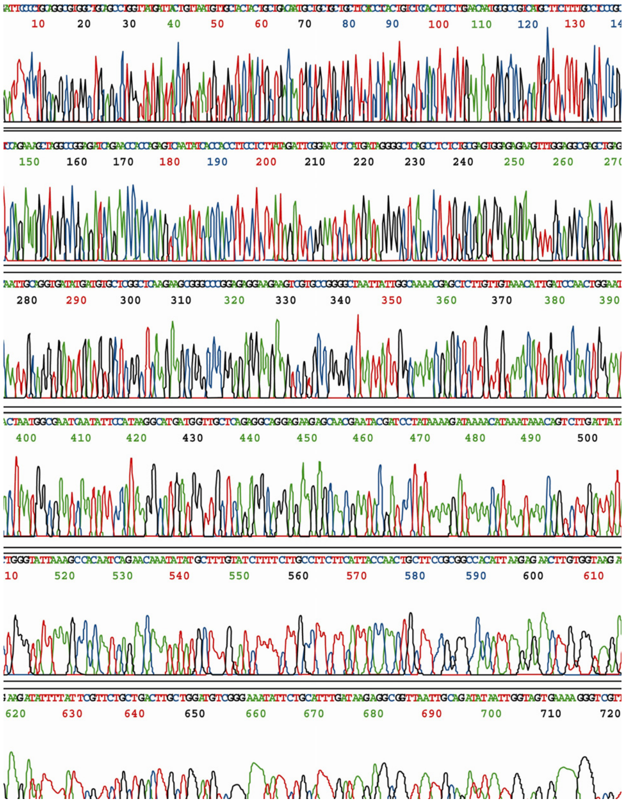
Background:

The FASTA algorithm and family of programs are similar to BLAST in that they both align a query sequence against all of the sequences in a database and return the most significant matches. Whereas BLAST relies on the sum match probability for each local alignment for the sequence, FASTA scores only *exact* matches. FASTA allows gapped searches to be made. Like BLAST, FASTA is heuristic, sacrificing some speed for sensitivity. FASTA comes in several flavors, depending upon the nature of your query, the most appropriate program should be chosen when searching. Some of the FASTA programs are:

fasta3: A DNA query sequence is aligned against a DNA sequence database. A protein query sequence will be aligned against a protein database.

tfasta3: Aligns a protein query sequence against a DNA sequence database, translating the DNA sequences 'on-the-fly'.

fastx3: Aligns a DNA query sequence against a protein sequence database, comparing the translated DNA sequence in three frames.



Purpose:

This lab is intended as an elementary introduction to the **FASTA** search engine.

Resources:

A few **FASTA** sites you can use:
http://fasta.bioch.virginia.edu/fasta_www2/fasta_list2.shtml
http://www.ebi.ac.uk/Tools/fasta33/nucleotide.html

Key Terms:

- FASTA
- SwissProt
- RefSeq
- Entrez
- NCBI

Directions:

Follow the instructions to identify the sequence below. Answer the exercises at the end of the lab.

1. Copy the sequence:
CCAGATCCTGGACAGAGGACAATGGCTTCCATGCAATTGGGCAGATGTGTGAGGCACCTGTGGTGACC

2. Go to the GeneStream FASTA server (or equivalent).

3. Paste the sequence into the query sequence window.

4. Select the database: **gbpri P** (GenBank Primates).

5. Hit the **Perform Search…** button.

6. Be patient!

Exercises:

1. Identify the sequence given above.

2. Compare and contrast the FASTA results layout with the BLAST results layout. How are they similar? How do the two layouts differ?

3. Looking at the results, what happens when one of the alignment links is clicked? How is this useful?

4. What are the advantages of using FASTA over BLAST?

5. Which alignment program, BLAST or FASTA, can align shorter sequences better? Explain why?

References:

1. FASTA Sequence Comparison at the University of Virginia. "FASTA Server". <http://fasta.bioch.virginia.edu/fasta_www2/fasta_list2.shtml>.

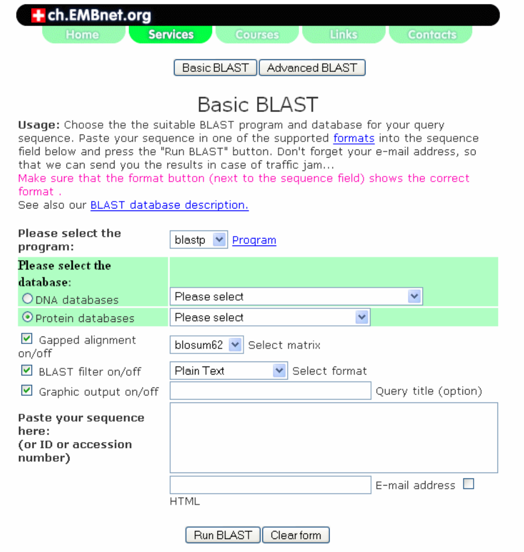
2. U.S. Department of Energy Human Genome Program, Genome Management Information System, Oak Ridge National Laboratory (Image).

Chapter 3

BLAST Analysis and Applications

Understanding the BLAST Programs

Background:



Basic BLAST

Usage: Choose the the suitable BLAST program and database for your query sequence. Paste your sequence in one of the supported formats into the sequence field below and press the "Run BLAST" button. Don't forget your e-mail address, so that we can send you the results in case of traffic jam...

Make sure that the format button (next to the sequence field) shows the correct format .

See also our BLAST database description.

Please select the program: blastp Program

Please select the database:

○ DNA databases — Please select

● Protein databases — Please select

☑ Gapped alignment on/off — blosum62 Select matrix

☑ BLAST filter on/off — Plain Text Select format

☑ Graphic output on/off — Query title (option)

Paste your sequence here:
(or ID or accession number)

E-mail address ☐ HTML

BLAST is a useful tool that allows a user to compare a query against protein and nucleotide sequence databases. The results from this tool can provide valuable functional and holomological information.

Purpose:

This lab is intended to contrast the five different **BLAST** programs in order to understand their function and appropriate applications.

Resources:

Use the **BLAST** site at (or equivalent): http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- BLASTN
- BLASTP
- BLASTX
- TBLASTN
- TBLASTX

Directions:

Use the five BLAST programs to answer the exercises below.

Exercises:

1. Each **BLAST** program compares a query sequence against a database sequence. For each of the five programs, identify the type of the query sequence and the type of the database sequence.

BLAST Programs	Query Type	Database Type
BLASTP		
BLASTN		
BLASTX		
TBLASTN		
TBLASTX		

2. Using **BLASTN,** find the organism that the sequence below comes from:

```
ATGGGAAAAAGGACTTCCCTAGAAGTGAGTCTTGGGGAGTTGGGGGGAGAAAAGTGTCGA
GGAGGGCGTCGGAGTTTCCCACCGCTGGCTGCTTCCCGGCCCGCACGCCCGGGAGGGTGG
CGGTGGGCGCGCAGAGATCTTTGCAAAACAGCGTCCAGGGCGGAAAACAACTCACAGGCC
TGCCGCCCCCAAAGGCGGGCAGGTCCGGACGCGCTGGGCCCTGGTCCCTTCGGCCGCAAA
CGGCGCAAGTCACGCACTGCGTTCACCGCGCAACAGGTGCTGGAGCTGGAGCGGCGCTTC
GTCTTCCAGAAGTACCTGGCGCCGTCCGAGCGAGACGGGCTAGCTACGCGACTCGGCCTG
GCCAACGCGCAGGTGGTCACTTGGTTCCAGAACCGGCGAGCCAAGCTCAAGCGCGATGTG
GAGGAGATGCGCGCCGACGTCGCCTCGCTACGCGCGTTGTCCCCGGAAGTCCTGTGCAGC
TTAGCACTGCCCGAAGGCGCTCCAGATCCCGGCCTCTGCCTCGGCCCTGCCGGCCCTGAC
TCCCGGCCCCACCTGTCAGACGAGGAGATACAGGTGGACGATTGA
```

3. Use the same sequence with **BLASTX.** Does it appear to be the same organism that was obtained when **BLASTN** was used for alignment?

4. In your opinion which **BLAST** program, **BLASTN** or **BLASTX**, produces the best results?

Next, use the same sequence with **TBLASTX**. Compare and contrast these results with the results obtained using **BLASTN** and **BLASTX**.

5. Does the **TBLASTX** program better align the DNA sequence?

6. As a result of **TBLASTX**, has your identification of the organism changed?

The following data is the amino acid sequence for the nucleotide sequence used in the questions above. Use this sequence below to answer questions 7 and 8.

```
MGKRTSLEVSLGELGGEKCRGGRRSFPPLAASRPARPGGWRWARRDLCKTASRAENNSQA
CRPQRRAGPDALGPGPFGRKRRKSRTAFTAQQVLELERRFVFQKYLAPSERDGLATRLGL
ANAQVVTWFQNRRAKLKRDVEEMRADVASLRALSPEVLCSLALPEGAPDPGLCLGPAGPD
SRPHLSDEEIQVDD
```

7. Using **BLASTP,** align the amino acid sequence above. Contrast these results with the results achieved using the previous **BLAST** programs.

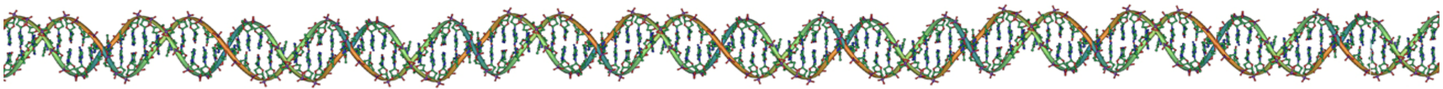
8. Using the above amino acid sequence run **TBLASTN**. Compare these results with the results obtained using the other **BLAST** programs. Explain any differences and similarities found. Does the **TBLASTN** program better align the sequence?

9. After using all the five **BLAST** programs, is it possible to positively identify the organism which the two sequences represent? Explain why or why not.

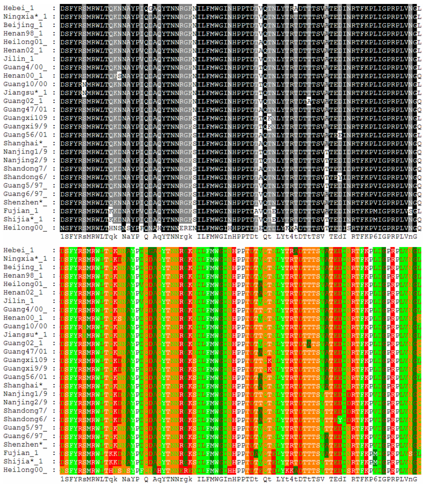
References:

1. *Swiss EMBnet.org. "Basic BLAST". <http://www.ch.embnet.org/software/bBLAST.html>.*

Sequence Information Searching



Background:



SWISS-PROT is a manually curated biological database of protein sequences. SWISS-PROT provides reliable protein sequences with a high level of detailed information. Such as function of protein, structure, variants, etc.

Genecards is a human gene database that includes genomic, proteomic, orthologies, disease relationships, SNP's, gene expressions and gene functions.

KEGG is a database of genes, proteins, biological systems, chemical building blocks, molecular wiring diagrams, reaction networks and hierarchies of biological objects.

Purpose:

This lab is intended to explore the different sequence databases and tools.

Resources:

Websites that will be useful:

- SWISS-PROT @ expasy - http://www.expasy.ch/
- ORF Finder - http://www.ncbi.nlm.nih.gov/gorf/gorf.html
- IHOP - http://www.ihop-net.org/UniPub/iHOP/
- EBI - http://www.ebi.ac.uk/index.html
- GeneCards - http://www.dkfz-heidelberg.de/GeneCards/
- KEGG **-** http://www.genome.ad.jp/kegg/

Directions:

Complete the exercises below.

Exercises:

1. Using one of the sequence databases search for the protein sequence **Hexokinase 1**? What is its purpose?

2. What protein accession numbers are associated with **Hexokinase 1**?

3. What accession numbers are associated with the yeast form of **hexokinase** protein? Typically these protein sequences are used for what functions?

4. What is the nucleotide sequence for the **yeast hexokinase A** gene.

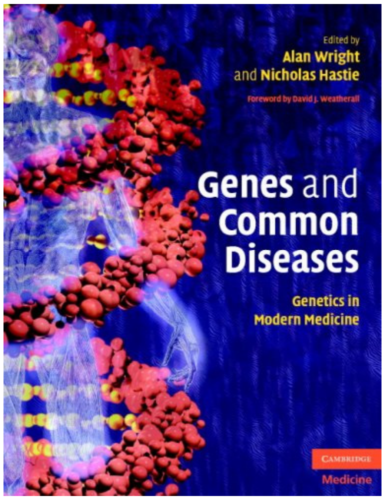
5. Obtain the genomic DNA of **human hexokinase 4**. What disease is caused by a mutation of this protein?

References:

1. *European Bioinformatics Institute (EBI). <http://www.ebi.ac.uk/index.html>. (17 December 2007).*
2. *ExPASy Proteomics Server. <http://www.expasy.ch/>.(11 December 2007).*
3. *GeneCards. <http://www.dkfz-heidelberg.de/GeneCards/>. (18 December 2007).*
4. *Information Hyperlinked Over Proteins (IHOP). <http://www.ihop-net.org/UniPub/iHOP/>. (18 December 2007).*
5. *Kyoto Encyclopedia of Genes and Genomes (KEGG). <http://www.genome.ad.jp/kegg/>. (18 December 2007).*
6. *National Center for Biotechnology Information (NCBI). "ORF Finder (Open Reading Frame Finder)". <http://www.ncbi.nlm.nih.gov/gorf/gorf.html>.*

Genes and Diseases

Background:


Edited by
Alan Wright
and Nicholas Hastie
Foreword by David J. Weatherall
Genes and Common Diseases
Genetics in Modern Medicine
Cambridge
Medicine

The study of Bioinformatics will continue improving the drug industry by providing ways to cure or prevent genetic diseases.

Purpose: This lab is intended to discover how Bioinformatics can help prevent and protect individuals from genetic diseases.

Resources: Genes and Diseases tutorial: http://www.ebi.ac.uk/2can/disease/

Key Terms:

- Genetic Disease
- Mutation
- Inheritance
- Chromosome
- SNP

Directions: Find on the internet the answers to the following exercises. A website is suggested in the resources.

Exercises:

1. How many base pairs are in the human genome and how many genes are there?

2. What percent of the human genome contains genes?

3. What is a major function of genes?

4. If the human genome only contains 2% genes, what is the purpose of the remaining 98% of the genome?

5. What are the three types of mutations and what can they lead to?

6. At what rate do mutations usually happen and what does our body do to repair this?

7. What are the problems/benefits with this repair system?

8. How do mutations cause disease?

9. What kind of environmental stresses can cause mutations?

10. In the mutation case study for Cystic Fibrosis, how many mutations in the specific gene cause this disease and how many base pairs are in the gene?

11. What are single nucleotide polymorphisms (SNPs)?

12. What percentage of our genes is identical to any other human?

13. What is pharmacogenetics and how is it useful?

14. How could polymorphic studies be used in target validation?

15. What are some of the databases that contain genetic mutations and associated diseases?

16. How has the pharmaceutical industry embraced the Bioinformatics revolution?

17. How did the process of medicine development change because of the Bioinformatics revolution?

18. How many genes does the current drug market target?

19. How does this present a huge opportunity for drug discovery?

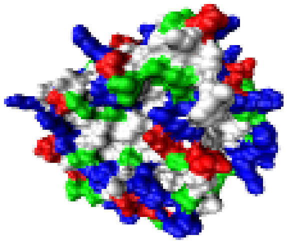
20. How did Bioinformatics become a major part of drug discovery?

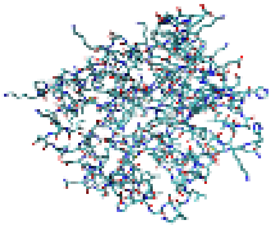
References:

1. *EMBL-EBI. "Genes and Diseases".<http://www.ebi.ac.uk/2can/disease/>. (19 December 2007).*

Protein Database Searching

Background:





Protein database searching is between two and five times more sensitive than DNA database searching for several reasons:

- The DNA alphabet is smaller (4 letters), yielding less information for each position (there are 20 possible amino acids at each position in a protein).
- The genetic code is redundant — several DNA codon triplets code for the same amino acid.
- The protein sequence similarity is more conserved through time than is the DNA sequence similarity.

There are two major protein databases that you will frequently encounter: PIR and SWISS-PROT. Unlike the three major nucleotide databases, the entries in PIR and SWISS-PROT are not mirrored (copied).

Purpose: This lab is intended to use the SWISS-PROT report to analyze a particular sequence.

Resources: See the pages linked from the SWISS-PROT report. All the help needed to answer the following questions should be on the first page of the results returned by the BLAST program.

The BLAST program: http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- SWISS-PROT Database
- PIR Database

Directions: Using BLASTP complete the exercises below.

Exercises:

Run this sequence against the SWISS-PROT Database. Adjust the BLAST options for this search and select a protein BLAST program. Run the search and identify the protein. Use the link provided to examine the SWISS-PROT report.

```
MSTAVLENPGLGRKLSDFGQETSYIEDNCNQNGAISLIFSLKEEVGALAKVLRLFEENDVNLTHIESRPSRLKKDEYEFF
THLDKRSLPALTNIIKILRHDIGATVHELSRDKKKDTVPWFPRTIQELDRFANQILSYGAELDADHPGFKDPVYRARRKQ
FADIAYNYRHGQPIPRVEYMEEEKKTWGTVFKTLKSLYKTHACYEYNHIFPLLEKYCGFHEDNIPQLEDVSQFLQTCTGF
RLRPVAGLLSSRDFLGGLAFRVFHCTQYIRHGSKPMYTPEPDICHELLGHVPLFSDRSFAQFSQEIGLASLGAPDEYIEK
LATIYWFTVEFGLCKQGDSIKAYGAGLLSSFGELQYCLSEKPKLLPLELEKTAIQNYTVTEFQPLYYVAESFNDAKEKVR
NFAATIPRPFSVRYDPYTQRIEVLDNTQQLKILADSINSEIGILCSALQKIK
```

2. What is the name of the given entry from SWISS-PROT?

3. What is the primary accession number for this protein?

4. What is the most common name of the protein?

5. What is the gene called?

6. Which year was the crystal structure of the catalytic domain determined? Name the first author.

7. Does the enzyme require a co-factor to function? If so, what?

8. Name the most common disease that arises as a result of deficiency of this enzyme.

9. Which cytogenetic locus does the gene reside at? (e.g. 13p10.1)

10. What is the PAHdb?

11. How many amino acid residues are there in the protein?

12. What is the molecular weight of the protein?

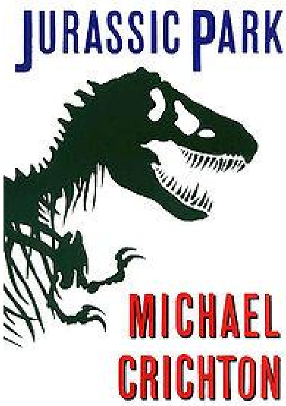
13. Using the Cross-references listed for UniProtKB/Swiss-Prot entry P00439, go to the 3D structure databases and find the 3D image for the first entry listed. Copy and paste the image below. Go to the Organism-specific databases, and click the link to the GeneCards PAH and find the 3D structure for 1DMW. Explore different options by clicking on the image. What is the description of 1DMW?

References:

1. *EMBL-EBI. "UniProtKB/Swiss-Prot". < http://www.ebi.ac.uk/swissprot/>. (19 December 2007).*
2. *Protein Information Resource (PIR).< http://pir.georgetown.edu/>. (19 December 2007).*
3. *Swiss EMBnet.org. "Basic BLAST".<http://www.ch.embnet.org/software/bBLAST.html>.*
4. Opabinia regalis, Created from PDB entry 1TIM using the freely available visualization and analysis package VMD (18 August 2006).

Jurassic Park

Background:



The ***Jurassic Park*** movie addressed the issue of recovering dinosaur DNA from mosquitoes trapped in fossilized tree sap. From the dinosaur DNA they clone the dinosaurs. In the movie they show a DNA sequence that allegedly comes from a dinosaur.

Purpose: This lab is intended to use BLAST to identify the DNA sequence in the movie Jurassic Park.

Resources: http://www.ebi.ac.uk/blastall/nucleotide.html (or equivalent)

Key Terms:

- BLAST
- Protein Function
- E-value
- Patented DNA

Directions: Use BLAST to run the DNA sequence presented in the Jurassic Park Movie. Complete the exercises below.

Exercises:

1. Take the sequence given below and align it against a Nucleic Acid database using the program BLASTN.

```
>DinoDNA from Jurassic Park p. 135 nr
GAATTCCGGAAGCGAGCAAGAGATAAGTCCTGGCATCAGATACAGTTGGAGATAAGGACG
GACGTGTGGCAGCTCCCGCAGAGGATTCACTGGAAGTGCATTACCTATCCCATGGGAGCC
ATGGAGTTCGTGGCGCTGGGGGGGCCGGATGCGGGCTCCCCCACTCCGTTCCCTGATGAA
GCCGGAGCCTTCCTGGGGCTGGGGGGGGGGCGAGAGGACGGAGGCGGGGGGCTGCTGGCC
TCCTACCCCCCCTCAGGCCGCGTGTCCCTGGTGCCGTGGGCAGACACGGGTACTTTGGGG
ACCCCCCAGTGGGTGCCGCCCGCCACCCAAATGGAGCCCCCCCACTACCTGGAGCTGCTG
CAACCCCCCCGGGGCAGCCCCCCCCATCCCTCCTCCGGGCCCCTACTGCCACTCAGCAGC
GGGCCCCCACCCTGCGAGGCCCGTGAGTGCGTCATGGCCAGGAAGAACTGCGGAGCGACG
GCAACGCCGCTGTGGCGCCGGGACGGCACCGGGCATTACCTGTGCAACTGGGCCTCAGCC
TGCGGGCTCTACCACCGCCTCAACGGCCAGAACCGCCCGCTCATCCGCCCCAAAAAGCGC
CTGCTGGTGAGTAAGCGCGCAGGCACAGTGTGCAGCCACGAGCGTGAAAACTGCCAGACA
TCCACCACCACTCTGTGGCGTCGCAGCCCCATGGGGGACCCCGTCTGCAACAACATTCAC
GCCTGCGGCCTCTACTACAAACTGCACCAAGTGAACCGCCCCCTCACGATGCGCAAAGAC
GGAATCCAAACCCGAAACCGCAAAGTTTCCTCCAAGGGTAAAAAGCGGCGCCCCCCGGGG
GGGGGAAACCCCTCCGCCACCGCGGGAGGGGGCGCTCCTATGGGGGGAGGGGGGGACCCC
TCTATGCCCCCCCCGCCGCCCCCCCCGGCCGCCGCCCCCCCTCAAAGCGACGCTCTGTAC
GCTCTCGGCCCCGTGGTCCTTTCGGGCCATTTTCTGCCCTTTGGAAACTCCGGAGGGTTT
TTTGGGGGGGGGGCGGGGGGTTACACGGCCCCCCCGGGGCTGAGCCCGCAGATTTAAATA
ATAACTCTGACGTGGGCAAGTGGGCCTTGCTGAGAAGACAGTGTAACATAATAATTTGCA
CCTCGGCAATTGCAGAGGGTCGATCTCCACTTTGGACACAACAGGGCTACTCGGTAGGAC
```

```
CAGATAAGCACTTTGCTCCCTGGACTGAAAAAGAAAGGATTTATCTGTTTGCTTCTTGCT
GACAAATCCCTGTGAAAGGTAAAAGTCGGACACAGCAATCGATTATTTCTCGCCTGTGTG
AAATTACTGTGAATATTGTAAATATATATATATATATATATATATCTGTATAGAACAGCC
TCGGAGGCGGCATGGACCCAGCGTAGATCATGCTGGATTTGTACTGCCGGAATTC
```

2. Based on the score report provided which sequence do you think aligned the best? (You will want to leave the score report open for the duration of the lab)

3. What is the E-Value, score and percent identity of the sequence?

4. Click on the DB:ID link for the description of the aligned sequence. What is the function of this sequence?

5. Looking at the descriptions, can you tell what organism this sequence came from?

6. How is the information we just found in this sequence related to the Jurassic park movie? If you do not remember, ask a student sitting next to you, or http://en.wikipedia.org/wiki/Jurassic_Park and read the "Plot summary."

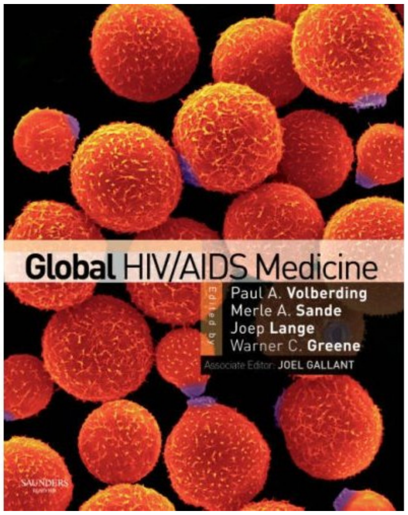
References:

1. ***Jurassic Museum*** - 2004-05-22. The Museum of Jurassic Technology in Los Angeles. *http://pdphoto.org/PictureDetail.php?pg=8089. (27 December. 2007).*
2. *EMBL-EBI. "Nucleotide Database Query". <http://www.ebi.ac.uk/blastall/nucleotide.html>. (26 December 2007).*
3. *Jurassic Park. Wikipedia.org. "Plot Summary". <http://en.wikipedia.org/wiki/Jurassic_Park>. (26 December 2007).*

HIV Patient Research

Background:

The HIV epidemic is one of the world's most destructive health crises. In 2006 there were an estimated 39.5 million people were living with HIV. HIV can be detected by testing DNA.



Purpose: This lab is intended to analyze a particular DNA sequence using BLAST and determine whether or not the individual has HIV.

Resources: A few **BLAST** sites you can use:
http://www.ebi.ac.uk/Tools/blast2/nucleotide.html
http://www.ebi.ac.uk/Tools/blastall/nucleotide.html
http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- HIV
- BLAST

Directions: Suppose a doctor informs an acquaintance of yours, as a result of a blood test, that they have HIV. They categorically deny it. Use **BLAST** to analyze the sequence below to determine whether or not the medical doctors are correct in their diagnosis.

Exercises:

Run the following alleged HIV Sequence using different BLAST programs:

```
GCGCAACAGCATATGTTGCAACTCACAGTCTGGGGCATCAAGCAGCTCCAGGCAAGAGTC
CTGGCTGTAGAAAGATACCTAAAGGATCAACAGCTCCTGGGGATTTGGGGCTGCTCTGGA
AAACTCATTTGCACCACTGCTGTGCCTTGGAATGCTAGTTGGAATAATACATCTATAGAT
GATATTTGGGATAACATGACCTGGATGGAGTGGGAAAGAGAAATTGACAATTATACAGGC
TACATATACAGCTTAATTGAAGAATCGCAGAACCAGCAAGAAAAGAATGAACAAGACTTA
TTGGAATTTGATATATGGGCAAATTTGTGGAATTGGTTTAACATAACAAATTGGCTGTGG
TATATA
```

1. Does this sequence appear to indicate HIV?

2. How closely does the sequence align with HIV?

After the sequence has been verified, experiment with the strength of the sequence identification. For instance, there could have been errors in sequencing that affected parts of the nucleotide sequence. You can check the strength of the sequence identification by modifying a letter and determining whether or not the sequence still indicates HIV.

3. How strong does this sequence identification appear to be? Explain.

Next, delete a few letters and run your new sequence on **BLAST**.

4. Does this modified sequence indicate HIV?

5. What can be concluded from the results of your experiment?

6. Delete an increasing amount of letters until the sequence is not aligned as HIV. Explain your findings.

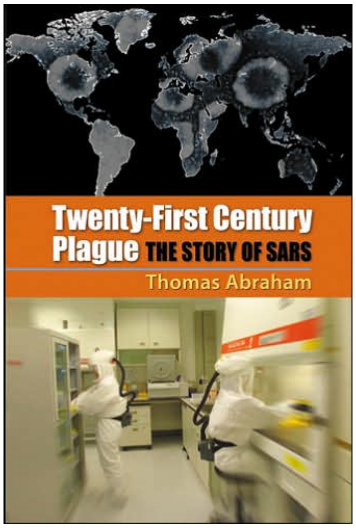
7. After running all of the experiments and logging all of the results, does the diagnosis of HIV appear to be correct? Explain why or why not.

References:

1. *Swiss EMBnet.org. "Basic BLAST". <http://www.ch.embnet.org/software/bBLAST.html>.*

SARS Exploration

Background: Severe acute respiratory syndrome (SARS) is a very contagious form of pneumonia caused by a virus belonging to a family of viruses known as coronavirus, which cause mild to moderate upper-respiratory illness in humans and is associated with respiratory, gastrointestinal, liver and neurological disease in animals. SARS was first reported in February 2003. Over the next few months the illness spread to more than two dozen countries in North America, South America, Europe, and Asia before the SARS global outbreak of 2003 was contained.



Purpose: This lab is intended to analyze a particular DNA sequence and determine whether or not the individual could be diagnosed with SARS.

Resources: A few **BLAST** sites you can use:
http://www.ebi.ac.uk/Tools/blast2/nucleotide.html
http://www.ebi.ac.uk/Tools/blastall/nucleotide.html
http://www.ch.embnet.org/software/bBLAST.html

Key Terms:

- SARS
- BLAST

Directions: Use BLAST to identify the sequence. Complete the exercises below.

Exercises:

1. Identify the sequence below by running it on **BLAST**.

```
CTTTAAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTAGTGCACCTACGCAGTATAAACAATAATAAATTTTACTGTCG
TTGACAAGAAACGAGTAACTCGTCCCTCTTCTGCAGACTGCTTACGGTTTCGTCCGTGTTGCAGTCGATCATCAGCATAC
CTAGGTTTCGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTTCTTGGTGTCAACGAGAAAACACACGTCCAACT
CAGTTTGCCTGTCCTTCAGGTTAGAGACGTGCTAGTGCGTGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCAC
GTGAACACCTCAAAAATGGCACTTGTGGTCTAGTAGAGCTGGAAAAAGGCGTACTGCCCCAGCTTGAACAGCCCTATGTG
TTCATTAAACGTTCTGATGCCTTAAGCACCAATCACGGCCACAAGGTCGTTGAGCTGGTTGCAGAAATGGACGGCATTCA
GTACGGTCGTAGCGGTATAACACTGGGAGTACTCGTGCCACATGTGGGCGAAACCCCAATTGCATACCGCAATGTTCTTC
TTCGTAAGAACGGTAATAAGGGAGCCGGTGGTCATAGCTATGGCATCGATCTAAAGTCTTATGACTTAGGTGACGAGCTT
GGCACTGATCCCATTGAAGATTATGAACAAAACTGGAACACTAAGCATGGCAGTGGTGCACTCCGTGAACTCACTCGTGA
GCTCAATGGAGGTGCAGTCACTCGCTATGTCGACAACAATTTCTGTGGCCCAGATGGGTACCCTCTTGATTGCATCAAAG
ATTTTCTCGCACGCGCGGGCAAGTCAATGTGCACTCTTTCCGAACAACTTGATTACATCGAGTCGAAGAGAGGTGTCTAC
TGCTGCCGTGACCATGAGCATGAAATTGCCTGGTTCACTGAGCGCTCTGATAAGAGCTACGAGCACCAGACACCCTTCGA
AATTAAGAGTGCCAAGAAATTTGACACTTTCAAAGGGGAATGCCCAAAGTTTGTGTTTCC
```

2. What are the scores of the top five alignments?

3. The sequence below is the first five lines of the original sequence. Run it in **BLAST.**

```
CTTTAAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTAGTGCACCTACGCAGTATAAACAATAATAAATTTTACTGTCG
TTGACAAGAAACGAGTAACTCGTCCCTCTTCTGCAGACTGCTTACGGTTTCGTCCGTGTTGCAGTCGATCATCAGCATAC
CTAGGTTTCGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTTCTTGGTGTCAACGAGAAAACACACGTCCAACT
CAGTTTGCCTGTCCTTCAGGTTAGAGACGTGCTAGTGCGTGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCAC
GTGAACACCTCAAAAATGGCACTTGTGGTCTAGTA
```

Does this sequence still align as SARS?

4. Contrast the two runs, what changes resulted in the alignment of this sequence compared to the first sequence?

5. This sequence has a 20 letter insertion in the middle. The insertion is represented by the bolded C's inserted in the middle of the sequence. Run it on **BLAST**.

```
CTTTAAAATCTGTGTAGCTGTCGCTCGGCTGCATGCCTAGTGCACCTACGCAGTATAAACAATAATAAATTTTACTGTCG
TTGACAAGAAACGAGTAACTCGTCCCTCTTCTGCAGACTGCTTACGGTTTCGTCCGTGTTGCAGTCGATCATCAGCACCC
CCCCCCCCCCCCCCCCCGTGACCGAAAGGTAAGATGGAGAGCCTTGTTCTTGGTGTCAACGAGAAAACACACGTCCAACT
CAGTTTGCCTGTCCTTCAGGTTAGAGACGTGCTAGTGCGTGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCGGAGGCAC
GTGAACACCTCAAAAATGGCACTTGTGGTCTAGTA
```

Did the insertion of the 20 letters affect the alignment? If so, how?

6. Did the E-value change from the second run as compared to the first run? List the E-values and discuss their significance.

7. What is noticeable about the graph of the second run? How is this different from the alignment without the insertion of the 20 letters?

8. The sequence below contains the 20 letter insertion from above and some rearranged letters (insertions, deletions or substitutions). Again, run it using **BLAST**.

```
TTTTACAATCTGTCTAGCTGTCGCTCGGCTGCATGCCTAATGCACCTACGCAGTATAAACAATAATAAATTTTACTGTCG
TTAACAAGAAACGAGTAACTCGTCCCTCTTCTTCAGACTGCTTTCGGTTTCGTCCGTGTTGCAGTTGATCATCAGCACCC
CCCCCCCCCCCCCCCCCGTGACCGAAAGGTAAGATGTAGAGCCTTGTTCTTGGTGTCAACGATAAAACACACGTCCAACT
CAGTTTGCCTCTCCTTCAAGTTAGAGAGGTGCTAGTGCGTGGCTTCGGGGACTCTGTGGAAGAGGCCCTATCTGAGGCAC
GTGAACACCTTAACAATGGCACTTGTGGTCTAGTA
```

How did the mutations combined with the insertion of C's affect the alignment? Is the sequence still aligned as SARS?

References:

1. *Swiss EMBnet.org. "Basic BLAST".<http://www.ch.embnet.org/software/bBLAST.html>.*

Extinct Organism Research

Background:

Geneticists are now sequencing the genomes of long-extinct organisms. In 2005, scientists announced that they have completely sequenced the DNA for cave bears that roamed the Austrian Alps some 40,000 years ago.

Purpose: This lab uses **BLAST** to identify extinct organisms and related living animals.

Resources: **BLAST** Servers:
http://www.ncbi.nlm.nih.gov/BLAST
http://www.ch.embnet.org/software/bBLAST.html
http://www.ch.embnet.org/software/aBLAST.html

Key Terms:

- GenBank
- BLAST

Directions: The sequences below are from extinct organisms. Choose a **BLAST** server and test these sequences using the appropriate **BLAST** programs and answer the exercises below.

Exercises:

1. What organism(s) does the following DNA sequence belong to?

```
TTTATCGATTATAGAACAGGCTCCTCTAGAGGGATGTAAAGCACCGCCAAGTCCTTTGAGTTTTAAGCTGTTGCTAGTAG
TTCTCTGGCGGATAGTTTTGTTTAGGGTAACTATCTAAGTTTAGGGCTAAGC
```

2. After you have identified the organism, look at the results and identify the organisms that are most closely related to this organism. Place them in order from the most related to the least.

Do you find this anomalous? Explain your answer.

3. Using the accession number, Y08503, retrieve its sequence. What is this organism?

4. List the five organisms that are most closely related to the organism with the accession number: Y08503.

5. Using **Basic Blast** at www.ch.embnet.org, determine what organism the protein sequence below belongs to?

```
MFLINVLTVTLPILLAVAFLTLVERKALGYMQLRKGPNVVGPYGLLQPIADAIKLFTK
EPVYPQTSSKFLFTIAPILALTLALTVWAPLPMPYPLINLNLSLLFILAMSSLMVYSI
LWSGWASNSKYALMGALRAVAQTISYEVSMTTIILSMVLMNGSFTLTAFATTQEHLWL
IFPMWPLMMMWFTSTLAETNRAPFDLTEGESELVSGFNVEYSAGPFALFFMAEYANII
MMNALTVILFMGTSCNPQMPEISTINFVVKTMILTICFLWVRASYPRFRYDQLMYLLW
KNFLPLTLALCMWHISILISLACIPPQA
```

6. Using the accession number: Q8W9N6, retrieve the sequence. What is this organism?

7. How closely is this organism related to the organism identified with the protein sequence above?

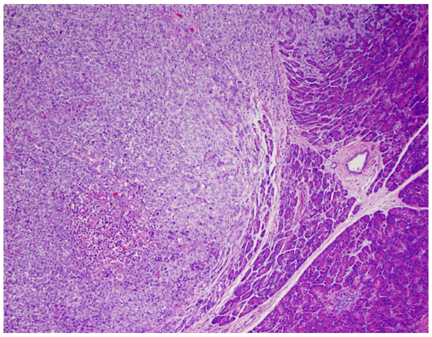
8. List the five organisms that are most closely related to the organism with the accession number: Q8W9N6.

References:

1. *NCBI / BLAST Home.*<http://www.ncbi.nlm.nih.gov/BLAST>. (07 July 2007).
2. *Ch.embnet.org, Basic Blast.* <http://www.ch.embnet.org/software/bBLAST.html>. (07 July 2007).
3. *Ch.embnet.org, Advanced Blast.* <http://www.ch.embnet.org/software/aBLAST.html>. (07 July 2007).

Cancer BLAST Experiment

Background:



Susceptibility to some forms of cancer is hereditary. This can sometimes be indicated in nucleotide and protein sequences.

Purpose:

The purpose of this lab is to become familiar with a tool Transeq and use it in combination with **BLAST** to analyze DNA and protein sequences in a real-world application.

Resources:

To answer the following lab questions use the **BLAST** site (or equivalent): http://www.ch.embnet.org/software/bBLAST.html and the Transeq site: http://www.ebi.ac.uk/emboss/transeq/

Key Terms:

- BLASTX
- BLASTN
- Oncology
- Transeq
- Cancer

Directions:

The oncologist informs you that you might be at risk for cancer. The test results contain the DNA sequence listed below. Use the bioinformatics tools **BLAST** and **Transeq**, to determine whether or not this DNA sequence might put you at risk for cancer. As you proceed, carefully analyze each of the searches conducted and answer the questions in detail below.

Exercises:

BLASTN is used to compare a nucleotide query sequence against known nucleotide sequence databases. Using the DNA sequence from your test results, run **BLASTN**, to determine possible similarities between the test sequence below and known sequences.

>DNA Sequence Test Results:

```
ATGGTCCAGAGGCTGTGGGTGAGCCGCCTGCTGCGGCACCGGAAAGCCCAGCTCTTGCTGGTCAACCTGCTAACCTTTGG
CCTGGAGGTGTGTTTGGCCGCAGGCATCACCTATGTGCCGCCTCTGCTGCTGGAAGTGGGGGTAGAGGAGAAGTTCATGA
CCATGGTGCTGGGCATTGGTCCAGTGCTGGGCCTGGTCTGTGTCCCGCTCCTAGGCTCAGCCAGTGACCACTGGCGTGGA
CGCTATGGCCGCCGCCGGCCCTTCATCTGGGCACTGTCCTTGGGCATCCTGCTGAGCCTCTTTCTCATCCCAAGGGCCGG
CTGGCTAGCAGGGCTGCTGTGCCCGGATCCCAGGCCCCTGGAGCTGGCACTGCTCATCCTGGGCGTGGGGCTGCTGGACT
TCTGTGGCCAGGTGTGCTTCACTCCACTGGAGGCCCTGCTCTCTGACCTCTTCCGGGACCCGGACCACTGTCGCCAGGCC
TACTCTGTCTATGCCTTCATGATCAGTCTTGGGGGCTGCCTGGGCTACCTCCTGCCTGCCATTGACTGGGACACCAGTGC
CCTGGCCCCCTACCTGGGCACCCAGGAGGAGTGCCTCTTTGGCCTGCTCACCCTCATCTTCCTCACCTGCGTAGCAGCCA
CACTGCTGGTGGCTGAGGAGGCAGCGCTGGGCCCCACCGAGCCAGCAGAAGGGCTGTCGGCCCCCTCCTTGTCGCCCCAC
TGCTGTCCATGCCGGGCCCGCTTGGCTTTCCGGAACCTGGGCGCCCTGCTTCCCCGGCTGCACCAGCTGTGCTGCCGCAT
GCCCCGCACCCTGCGCCGGCTCTTCGTGGCTGAGCTGTGCAGCTGGATGGCACTCATGACCTTCACGCTGTTTTACACGG
ATTTCGTGGGCGAGGGGCTGTACCAGGGCGTGCCCAGAGCTGAGCCGGGCACCGAGGCCCGGAGACACTATGATGAAGGC
GTTCGGATGGGCAGCCTGGGGCTGTTCCTGCAGTGCGCCATCTCCCTGGTCTTCTCTCTGGTCATGGACCGGCTGGTGCA
GCGATTCGGCACTCGAGCAGTCTATTTGGCCAGTGTGGCAGCTTTCCCTGTGGCTGCCGGTGCCACATGCCTGTCCCACA
```

```
GTGTGGCCGTGGTGACAGCTTCAGCCGCCCTCACCGGGTTCACCTTCTCAGCCCTGCAGATCCTGCCCTACACACTGGCC
TCCCTCTACCACCGGGAGAAGCAGGTGTTCCTGCCCAAATACCGAGGGGACACTGGAGGTGCTAGCAGTGAGGACAGCCT
GATGACCAGCTTCCTGCCAGGCCCTAAGCCTGGAGCTCCCTTCCCTAATGGACACGTGGGTGCTGGAGGCAGTGGCCTGC
TCCCACCTCCACCCGCGCTCTGCGGGGCCTCTGCCTGTGATGTCTCCGTACGTGTGGTGGTGGGTGAGCCCACCGAGGCC
AGGGTGGTTCCGGGCCGGGGCATCTGCCTGGACCTCGCCATCCTGGATAGTGCCTTCCTGCTGTCCCAGGTGGCCCCATC
CCTGTTTATGGGCTCCATTGTCCAGCTCAGCCAGTCTGTCACTGCCTATATGGTGTCTGCCGCAGGCCTGGGTCTGGTCG
CCATTTACTTTGCTACACAGGTAGTATTTGACAAGAGCGACTTGGCCAAATACTCAGCGTAG
```

1. Does the result appear to be verify that you might be at risk for cancer?

2. What does this sequence appear to be?

Next run **BLASTX** on the DNA sequence above. **BLASTX** compares a nucleotide query sequences against a known protein sequence database.

3. How do the results differ from the previous search? Do you still think you know what cancer this sequence indicates?

Note: This sequence is difficult to identify, you will need to carefully analyze the resulting list of sequences. Find more information by clicking on each of the sequences identified.

Now run the sequence using the **Transeq** tool which translates nucleic acid sequences to their corresponding peptide sequence, *http://www.ebi.ac.uk/emboss/transeq/.*

This tool can translate any or all of the three forward or three reverse sense frames. Choose the first protein sequence and then some other protein sequences and run them with different protein **BLAST** programs. When running these searches, click on the actual sequence that is the closest in resemblance to the sequence being tested. This will give you more information on the gene.

4. What type of cancer does this appear to be? Explain your answer.

5. Does the nucleotide sequence and protein sequence produce the same results?

6. Which program do you think aligned DNA sequence the best?

References:

1. The Swiss node of EMBnet. <http://www.ch.embnet.org/software/bBLAST.html>.
2. *The European Bioinformatics Institute (EBI). "EMBOSS Transeq". <http://www.ebi.ac.uk/emboss/transeq/>.*
3. *NHGRI Press Photos Gallery: Science Photos OtherGenetic [Internet]. Bethesda, MD: United States National Library of Medicine, National Institute of Health. [Photo], Metastatic Melanoma Cells [cited 2009 August 19][about 3 screens]. Available from: <http://www.genome.gov/17516873>*

Scoring Matrices

Background: In Bioinformatics scoring matrices for computing alignment scores are often based on observed substitution rates, derived from the substitution frequencies seen in multiple alignments of sequences. Every possible identity and substitution is assigned a score based on the observed frequencies of such occurrences in alignments of related proteins. The score is calculated from the frequency of occurrence of a match of the two individual amino acids in evolutionarily related sequences, and provides a measure of a chance alignment of the two amino acids. This score will also reflect the frequency that a particular amino acid occurs in nature, as some amino acids are more abundant than others. Higher scores indicate that the probability that those two amino acids aligned by chance is very small, and lower scores indicate a high probability the two amino acids aligned by chance, and are evolutionarily unrelated. Thus, identities are assigned the most positive scores. Frequently observed substitutions also receive positive scores, but matches that are unlikely to have been a result of evolution, and are more likely indicative of no relation at that position, are given negative scores. Matrices with scoring schemes based on observed substitution rates are superior to simple identity scores, or scores based solely on sidechain moiety similarity. The two most commonly used types of scoring matrices are the PAM matrices and the BLOSSUM matrices.

Purpose: This lab is intended to compare and become familiar with the scoring matrices.

Resources: http://www.bioinformatics.nl/tools/pam.html

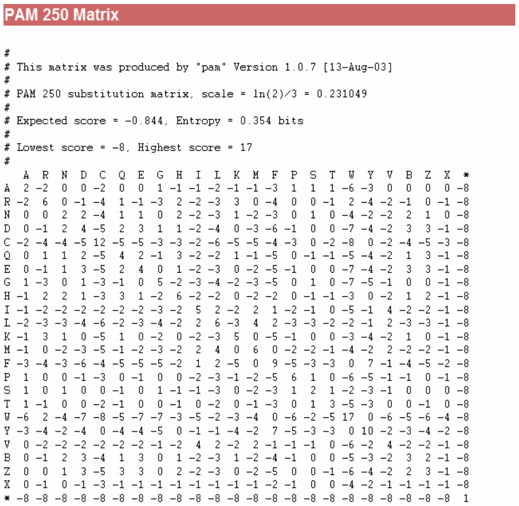
Key Terms:

- Scoring Matrix
- BLOSUM
- PAM
- Evolutionary Distance

Directions: A Biomolecular Informatics website sponsored by the University of Nijmegen, allows online computation of PAM matrices. Use this resource to view some PAM matrices.

The default is a PAM 250 matrix; calculate this matrix and observe the results. This PAM 250 matrix has a built-in gap penalty of -8, as seen in the * column. There are 24 rows and 24 columns. The first 20 are the amino acids, represented by the one letter code.

B represents the case where there is ambiguity between aspartate or asparigine.



PAM 250 Matrix

```
#
# This matrix was produced by "pam" Version 1.0.7 [13-Aug-03]
#
# PAM 250 substitution matrix, scale = ln(2)/3 = 0.231049
#
# Expected score = -0.844, Entropy = 0.354 bits
#
# Lowest score = -8, Highest score = 17
#
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
A  2 -2  0  0 -2  0  0  1 -1 -1 -2 -1 -1 -3  1  1  1 -6 -3  0  0  0  0 -8
R -2  6  0 -1 -4  1 -1 -3  2 -2 -3  3  0 -4  0  0 -1  2 -4 -2 -1  0 -1 -8
N  0  0  2  2 -4  1  1  0  2 -2 -3  1 -2 -3  0  1  0 -4 -2 -2  2  1  0 -8
D  0 -1  2  4 -5  2  3  1  1 -2 -4  0 -3 -6 -1  0  0 -7 -4 -2  3  3 -1 -8
C -2 -4 -4 -5 12 -5 -5 -3 -3 -2 -6 -5 -5 -4 -3  0 -2 -8  0 -2 -4 -5 -3 -8
Q  0  1  1  2 -5  4  2 -1  3 -2 -2  1 -1 -5  0 -1 -1 -5 -4 -2  1  3 -1 -8
E  0 -1  1  3 -5  2  4  0  1 -2 -3  0 -2 -5 -1  0  0 -7 -4 -2  3  3 -1 -8
G  1 -3  0  1 -3 -1  0  5 -2 -3 -4 -2 -3 -5  0  1  0 -7 -5 -1  0  0 -1 -8
H -1  2  2  1 -3  3  1 -2  6 -2 -2  0 -2 -2  0 -1 -1 -3  0 -2  1  2 -1 -8
I -1 -2 -2 -2 -2 -2 -2 -3 -2  5  2 -2  2  1 -2 -1  0 -5 -1  4 -2 -2 -1 -8
L -2 -3 -3 -4 -6 -2 -3 -4 -2  2  6 -3  4  2 -3 -3 -2 -2 -1  2 -3 -3 -1 -8
K -1  3  1  0 -5  1  0 -2  0 -2 -3  5  0 -5 -1  0  0 -3 -4 -2  1  0 -1 -8
M -1  0 -2 -3 -5 -1 -2 -3 -2  2  4  0  6  0 -2 -2 -1 -4 -2  2 -2 -2 -1 -8
F -3 -4 -3 -6 -4 -5 -5 -5 -2  1  2 -5  0  9 -5 -3 -3  0  7 -1 -4 -5 -2 -8
P  1  0  0 -1 -3  0 -1  0  0 -2 -3 -1 -2 -5  6  1  0 -6 -5 -1 -1  0 -1 -8
S  1  0  1  0  0 -1  0  1 -1 -1 -3  0 -2 -3  1  2  1 -2 -3 -1  0  0  0 -8
T  1 -1  0  0 -2 -1  0  0 -1  0 -2  0 -1 -3  0  1  3 -5 -3  0  0 -1  0 -8
W -6  2 -4 -7 -8 -5 -7 -7 -3 -5 -2 -3 -4  0 -6 -2 -5 17  0 -6 -5 -6 -4 -8
Y -3 -4 -2 -4  0 -4 -4 -5  0 -1 -1 -4 -2  7 -5 -3 -3  0 10 -2 -3 -4 -2 -8
V  0 -2 -2 -2 -2 -2 -2 -1 -2  4  2 -2  2 -1 -1 -1  0 -6 -2  4 -2 -2 -1 -8
B  0 -1  2  3 -4  1  3  0  1 -2 -3  1 -2 -4 -1  0  0 -5 -3 -2  3  2 -1 -8
Z  0  0  1  3 -5  3  3  0  2 -2 -3  0 -2 -5  0  0 -1 -6 -4 -2  2  3 -1 -8
X  0 -1  0 -1 -3 -1 -1 -1 -1 -1 -1 -1 -1 -2 -1  0  0 -4 -2 -1 -1 -1 -1 -8
* -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8 -8  1
```

Z is the case where there is ambiguity between glutamate or glutamine.

X represents an unknown or nonstandard amino acid.

Exercises:

1. In the PAM 250 scoring matrix, where can the highest scores for each amino acid be found? Why?

2. Would the answer for the above question be true for all scoring matrices?

3. What row and column combination gives the highest score and the second highest score in the PAM 250 scoring matrix? (Specify the score values).

4. Why are some scores for amino acid identities higher than others?

5. Use the back button on the browser, and calculate a PAM 100 scoring matrix. Are the two highest scoring matches the same combination of row and column as in the PAM 250 scoring matrix? Why?

6. Explain any differences in the gap penalties of the PAM 250 scoring matrix versus the PAM 100 scoring matrix.

7. If given an unknown sequence, what would be the preferable scoring matrix to use?

References:

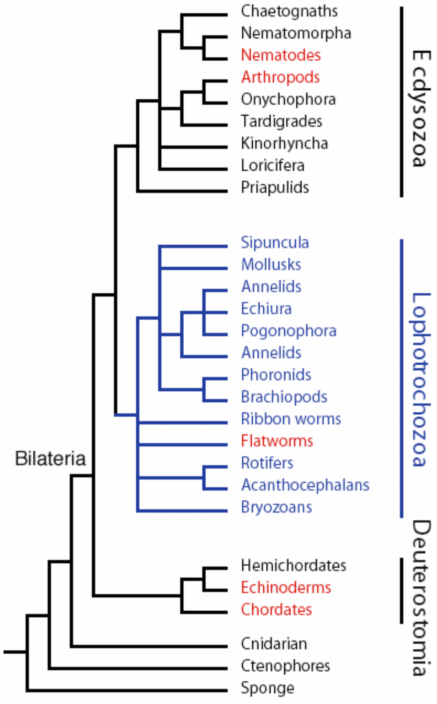
1. *Bioinformatics.nl. "Calculate PAM Matrix". < http://www.bioinformatics.nl/tools/pam.html>. (3 August 2004).*

Chapter 4

Advanced Bioinformatics Tools

Introduction to CLUSTAL

Background: Multiple sequence alignment techniques are used most often with protein sequences. These techniques, however, can be used with DNA sequences as well. In general, multiple sequence alignment, when applied to three or more nucleotide or protein sequences, allows researchers to search for possible relationships that reflect sequence function, sequence homology, and sequence structure.



Multiple sequence alignment is one of the most powerful procedures used in Bioinformatics. As in pair wise alignment analysis of two sequences, the results of multiple sequence alignment are not necessarily correct or incorrect, i.e. the resulting sequential alignments may or may not reflect known biological properties. However, proteins with closely related functions tend to be similar in structure and function from organism to organism.

Comparative modeling attempts to predict the structure of a target nucleotide or protein sequence with related structures found in curated databases. While there are a number of steps, the first step is to identify sequences for known organisms with the target sequence; **BLAST** can be used for that purpose. The next step is to perform a multiple sequence alignment of our target sequence with related sequences found using **BLAST.** The process does not stop here. The over aim is ultimately to construct a 3-D model of the amino acid sequence for the resultant protein and understand its function.

The **Clustal** algorithm begins by relating the closest sequences and then adding the more divergent sequences. The **Clustal** program builds a phylogenetic tree that gives some hint to the degree of similarity among the sequences being aligned.

Purpose: This lab is intended to introduce multiple sequence alignment and to use **Clustal** programs to analyze multiple sequence alignments and to introduce the **Jalview** tool for examining **Clustal** program output.

Resources:
ClustalW: http://www.ebi.ac.uk/clustalw/
ClustalX: ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalX/
Jalview: http://www.ebi.ac.uk/jalview/

Key Terms:

- ClustalX
- ClustalW
- Homologs
- Cladogram
- Multiple Sequence Alignment
- Phylogenetic tree
- Similarity
- Comparative Modeling

Directions: Use **Clustal** to complete the exercises below.

Exercises:

1. What is the purpose of **Clustal**?

2. How does **ClustalX** differ from **ClustalW**?

3. What are the differences between homology and similarity?

4. How is **Clustal** used in Comparative Modeling?

Suppose you have the following five mythical protein sequences:

1) INTEMPERATE
2) IRRADIATED
3) ATTEMPTED
4) TEMPLATE
5) GYRATED

5. In your estimation, which proteins are closely related?

6. Assuming that these five protein sequences are somehow related, using the numbers 1 through 5, make a guess at what their family tree might look like.

Format the above five protein sequences into the format and save your file. For example the first entry should be:

```
>#1 intemperate
INTEMPERATE
```

Paste the **FASTA** formatted file contents into the data input window in the **ClustalW** submission form located at: http://www.ebi.ac.uk/clustalw/

7. Using the output from **ClustalW**, what is the phylogenetic relationship between the five protein sequences? Use **Jalview** to create the phylogenetic tree. How does this compare with your estimates in exercise #6?

8. What protein(s) of interest are conserved in the sequences?

9. **Clustal** tends to give better results when run with protein sequences than with nucleotides sequences. Explain why this might be the case.

Consider the protein sequence below:

```
MTNIRKSHPLIKILNNSFIDLPTPVNISSWWNFGSLLGACLIIQILTGLFLAMHYTSDTL
TAFSSVTHICRDVNYGWIIRYLHANGASMFFLCLYAHIGRGIYYGSYLYPETWNIGIVLL
LTVMATAFMGYVLPWGQMSFWGATVITNLLSAIPYIGTNLVEWVWGGFSVDKATLTRFFA
```

```
LHFILPFIVTALVMVHLLFLHETGSNNPTGLISDSDKIPFHPYYSVKDLLGLFLLILVLL
LLTLFSPDMLGDPDNYTPANPLNTPPHIKPEWYFLFRYAILRSIPNKLGGVLALVLSILI
LALLPLLHTSKQRSLSFRPLSQCLFWILVADLITLTWIGGQPVEHPYIIIGQLASILYFS
IILIFMPIAGLIENHLLKW
```

Using **BLAST**, identify the organisms that are most closely associated with the sequence. Choose four of the organisms and use **ClustalW** to create a phylogenetic tree and save your results.

10. What areas among the species are conserved?

Consider the following DNA sequence:

```
ATATTTGAAAGCTGTGTCTGTAAACTGATGGCTAACAAAACcttaGATTTTGGTCACTTC
TAAAATGGAACATTTAAAGAAAGCTGACAAAATATTAATTTTGCATGAAGGTAGCAGCTA
TTTTTATGGGACATTTTCAGAACTCCAAAATCTACGGCCAGACTTTAGCTCAAAACTCAT
GGGATGTGATTCTTTCGACCAATTTAGTGCAGAAAGAAGAAATTCAATCCTAACTGAGAC
CTTACGCCGATTCTCATTAGAAGGAGATGCTCCTGTCTCCTGGACAGAAACCAATCTTTT
AAACAGACTGGAGAGTTTGGGGAAAAAAGGAAGAATTCTATTCTCAATCCAATCAACTCT
ATACGAAAATTTTCCATTGTGCAAAAGACTCCCTTACAAATGAATGGCATCGAAGAGGAT
TCTGATGAGCCATTCAttGAGAAGGCTGTCCTTAGTACCAGATTCTGAGCAGGGAGAGGC
GATACTGCCTCGCATCAGCGTGATCAGCACTGGCCCACGCTTCAGGCACGAAGGAGGCAG
TCTGTTCTGAACCTGATGACACACTCAGCTCTACCAAGGTCAGAACATTCACCGAAAGAC
AACAGCATCCACACGAAAAGTGTCACTGGCCCCTCAGGCAAACTTGACTGAACTGGATAT
ATATTCAAGAAGGTTATCTCAAGAAACTGGCTTGGAAATAAGTGAAGAAATTAACGAAGA
AGACTTAAAGG
```

11. Run **BLAST** on the sequence and attempt to identify the sequence. What is the closest match?

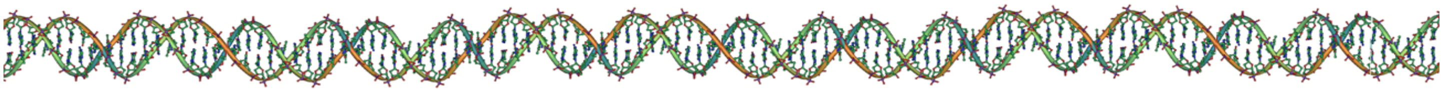
12. If you were successful in identifying the sequence above, extract DNA from three related organisms and run **ClustalW** on the five sequences. Use **Jalview** to examine the results. Use print screen to save screen shots of the aligned sequences and the phylogenetic tree.

13. Using the web as a resource, what are some general guidelines for using **ClustalW**?

References:

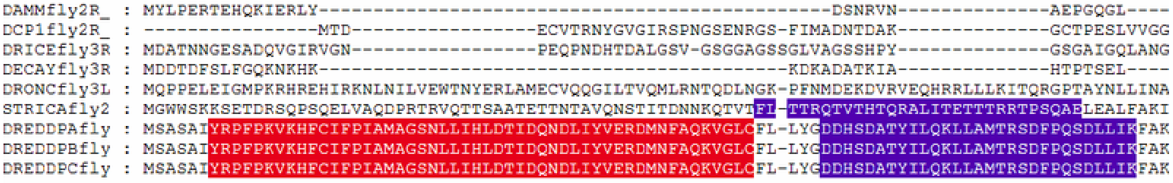
1. *ClustalX. "ftp: ClustalX". <ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalX/>.*
2. *EMBL-EBI. "ClustalW". <http://www.ebi.ac.uk/Tools/clustalw2/index.html>.*
3. *University of Dundee. "JalView: A Multiple Alignment Editor". <http://www.jalview.org/>.*

Introduction to PSI-BLAST



Background:

PSI-BLAST - (Position-Specific Iterated **BLAST**) is a tool that produces a position-specific scoring matrix constructed from a multiple alignment of the top-scoring **BLAST** responses to a given query sequence. This scoring matrix produces a profile designed to identify the key positions of conserved amino acids within a motif. When a profile is used to search a database it can often detect subtle relationships between proteins that are distant structural or functional homologs. These relationships are often not detected by a **BLAST** search with a sample sequence query.



```
DAMMfly2R_  : MYLPERTEHQKIERLY-----------------------------------------------DSNRVN--------------AEPGQGL----
DCP1fly2R_  : ----------------MTD-----------------ECVTRNYGVGIRSPNGSENRGS-FIMADNTDAK--------------GCTPESLVVGG
DRICEfly3R  : MDATNNGESADQVGIRVGN-----------------PEQPNDHTDALGSV-GSGGAGSSGLVAGSSHPY--------------GSGAIGQLANG
DECAYfly3R  : MDDTDFSLFGQKNKHK-------------------------------------------KDKADATKIA--------------HTPTSEL----
DRONCfly3L  : MQPPELEIGMPKRHREHIRKNLNILVEWTNYERLAMECVQQGILTVQMLRNTQDLNGK-PFNMDEKDVRVEQHRRLLLKITQRGPTAYNLLINA
STRICAfly2  : MGWWSKKSETDRSQPSQELVAQDPRTRVQTTSAATETTNTAVQNSTITDNNKQTVTFL-TTRQTVTHTQRALITETTTRRTPSQAELEALFAKI
DREDDPAfly  : MSASAIYRPFPKVKHFCIFPIAMAGSNLLIHLDTIDQNDLIYVERDMNFAQKVGLCFL-LYGDDHSDATYILQKLLAMTRSDFPQSDLLIKFAK
DREDDPBfly  : MSASAIYRPFPKVKHFCIFPIAMAGSNLLIHLDTIDQNDLIYVERDMNFAQKVGLCFL-LYGDDHSDATYILQKLLAMTRSDFPQSDLLIKFAK
DREDDPCfly  : MSASAIYRPFPKVKHFCIFPIAMAGSNLLIHLDTIDQNDLIYVERDMNFAQKVGLCFL-LYGDDHSDATYILQKLLAMTRSDFPQSDLLIKFAK
```

PSI-BLAST was engineered to identify distant relationships between sequences that are too subtle to discover with a regular **BLAST** search. In the first round **PSI-BLAST** is just like a normal **BLAST**; it finds sequence homologs. In the second round or "iteration" of **PSI-BLAST**, it figures out which residues tend to be conserved by creating a custom profile for each position of the sequence from a multiple alignment. Then another **BLAST** is performed using the profile to produce a position-specific scoring matrix based on which positions evolution has conserved versus which positions evolution has allowed to vary. The sequences found after the first round are added to the profile, allowing **PSI-BLAST** to detect more distant homologs in each iteration.

Purpose:

This lab is intended to introduce the use of **PSI-BLAST**.

Resources:

http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html
http://biocluster.ucr.edu/~hli/tishunter.html

Key Terms:

- PAM
- BLOSSUM
- Profile
- Homology

Directions:

Use **PSI-BLAST** to complete the exercises below.

Exercises:

1. Using **PSI-BLAST** run the protein sequence below. How many hits are returned?

```
XDEDETTALVCDNGSGLVKAGFAGDDAPRAVFPSIVGRPRHQGVMVGMGQKDSYVGDEAQSKRGILTLKYPIEHGIITNW
DDMEKIWHHTFYNELRVAPEEHPTLLTEAPLNPKANREKMTQIMFETFNVPAMYVAIQAVLSLYASGRTTGIVLDSGDGV
THNVPIYEGYALPHAIMRLDLAGRDLTDYLMKILTERGYSFVTTAEREIVRDIKEKLCYVALDFENEMATAASSSSLEKS
YELPDGQVITIGNERFRCPETLFQPSFIGMESAGIHETTYNSIMKCDIDIRKDLYANNVMSGGTTMYPGIADRMQKEITA
LAPSTMKIKIIAPPERKYSVWIGGSILASLSTFQQMWITKQEYDEAGPSIVHR
```

2. What is, by far, the most common protein shown as a hit in the list of scores?

Click on one of the "Run **PSI-BLAST** Iteration 2" buttons to run a 2nd iteration of **PSI-BLAST**. The format window will still be open from the 1st round of **BLAST**. To view the 2nd iteration of **PSI-BLAST** results, return to the format window and select "Format!". View the results window. Above the graphical display reads "Results of **PSI-Blast** iteration 2". Each new iteration of **PSI-BLAST** requires going back and forth between the format and results windows. Be careful. Using the back and forward arrows will NOT necessarily allow you to view previous iterations of **PSI-BLAST**. In some browsers, **PSI-BLAST** will proceed to the next iteration when using the back and forward arrows. Keep track of what the next iteration number should be, and that it matches the one that **PSI-BLAST** displays. When the results appear, view the legend under the color alignment graph. A yellow starburst containing "NEW" indicates a new sequence identified as a result of the most recent iteration. A green dot indicates a sequence already present prior to the most recent iteration.

3. How many results were returned for the second iteration?

4. Did the second iteration locate any new sequence within the threshold E-value? If so, how many?

5. Are there new members of the Actin-like ATPase domain superfamily in the section entitled “Sequences with E-value worse than threshold”? If so, name one.

6. How many new sequences are identified as a result of the third iteration?

Finally, perform the first iteration of the **PSI-BLAST** search with the same query sequence as above, but with the database set to "nr".

7. How many **BLAST** hits on the query sequence are returned? Why do you think this is?

References:

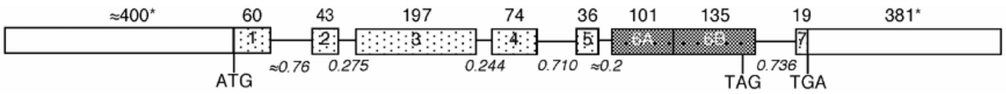
1. *NCBI. “PSI-BLAST Tutorial”. <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html>.*

Translation Initiation Sites

Background:

Translation Initiation Sites contain the start signals of a gene. These start signals are represented by specific subsequences of nucleotides. However these specific subsequences may have a number of occurrences, only one of which represents the actual start signal.

The tool we are going to use to find the actual start signal is **TISHunter**. This software uses Support Vector Machines trained with many other sequences to make an accurate prediction of the real start signal.



Purpose:

The purpose of this lab is to have a basic understanding of translation initiation sites and how TISHunter helps in finding these sites.

Resources:

"Tishunter",<http://bioinfo.ucr.edu/~hli/>
http://www.ncbi.nlm.nih.gov/gorf/gorf.html

Key Terms:

- TISHunter
- ORF Finder
- Gene
- Translation
- Exon
- Intron

Directions:

Using **TISHunter** complete the exercises below.

Exercises:

1. Using **TISHunter** run the sequence below. What occurred when running this sequence?

```
ATGTCCGAAG ATATCTTTGA CGCCATCATC GTCGGTGCAG GGCTTGCCGG TTCGGTTGCC GCACTGGTGC
TCGCCCGCGA AGGTGCGCAA GTGTTAGTTA TCGAGCGTGG CAATTCCGCA GGTGCCAAGA ACGTCACCGG
CGGGCGTCTC TATGCCCACA GTCTGGAACA CATTATTCCT GGTTTCGCCG ACTCCGCCCC CAUGGTAGAA
CGCCTGATCA CCCATGAAAA ACTCGCGTTT ACGGAAAAGT CAGCGATGAC TATGGACTAC TGCAATGGTG
ACGAAACCTC GCCATCCCAG CGTTCTTACT CCGTTTTGCG CAGTAAATTT GATGCCTGGC TGATGGAGCA
GGCCGAAGAA GCGGGCGCGC AGTTAATTAC CGGGATCCGC GTCGATAACC TCGTACAGCG CGATGGCAAA
GTCGTCGGTG TAGAAGCCGA TGGCGATGTG ATTGAAGCGA AAACGGTGAT CCTTGCTGAT GGGGTGAACT
CCATCCTTGC CGAAAAATTG GGGATGGCAA AACGCGTCAA ACCGACGGAT GTGGCGGTTG GCGTGAAGGA
ACTGATCGAG TTACCGAAGT CGGTTATTGA AGACCGTTTT CAGTTGCAGG GTAATCAGGG GGCGGCTTGC
CTGTTTGCGG GATCACCCAC CGATGGCCTG ATGGGCGGCG GCTTCCTTTA TACCAATGAA AACACCCTGT
CGCTGGGGCT GGTTTGTGGT TTGCATCATC TGCATGACGC GAAAAAATCG GTGCCGCAAA TGCTGGAAGA
TTTCAAACAG CATCCGGCCG TTGCACCGCT GATCGCGGGC GGCAAGCTGG TGGAATATTC CGCTCACGTA
GTGCCGGAAG CAGGCATCAA CATGCTGCCG GAGTTGGTTG GTGACGGCGT ATTGATTGCC GGTGATGCCG
CCGGAATGTG TATGAACCTC GGTTTTACCA TTCGCGGTAT GGATCTGGCG ATTGCCGCCG GGGAAGCCGC
AGCAAAAACC GTGCTTTCAG CGATGAAAAG CGACGATTTC AGTAAGCAAA AACTGGCGGA ATATCGTCAG
CATCTTGAGA GTGGTCCGCT GCGCGATATG CGTATGTACC AGAAACTACC GGCGTTCCTT GATAACCCAC
GCATGTTTAG CGGCTACCCG GAGCTGGCGG TGGGTGTGGC GCGTGACCTG TTCACCATTG ATGGCAGCGC
GCCGGAACTG ATGCGCAAGA AAATCCTCCG CCACGGCAAG AAAGTGGGCT TCATCAATCT AATCAAGGAT
GGCATGAAAG GAGTGACCGT TTTATGA
```

2. Using **TISHunter** run the smaller sub-sequence below. Why did the program not find any start signals, even though there is one?

```
ATGTCCGAAG ATATCTTTGA CGCCATCATC GTCGGTGCAG GGCTTGCCGG TTCGGTTGCC GCACTGGTGC
TCGCCCGCGA AGGTGCGCAA GTGTTAGTTA TCGAGCGTGG CAATTCCGCA GGTGCCAAGA ACGTCACCGG
CGGGCGTCTC TATGCCCACA GTCTGGAACA CATTATTCCT GGTTTCGCCG ACTCCGCCCC CAUGGTAGAA
CGCCTGATCA CCCATGAAAA ACTCGCGTTT
```

In the sequence below, there is a potential start signal inserted, which is the portion highlighted in red. Run the sequence using **TISHunter**.

```
ATGTCCGAAG ATATCTTTGA CGCCATCATC GTCGGTGCAG GGCTTGCCGG TTCGGTTGCC GCACTGGTGC
TCGCCCGCGA AGGTGCGCAA ATGTTAGTTA TCGAGCGTGG CAATTCCGCA GGTGCCAAGA ACGTCACCGG
CGGGCGTCTC TATGCCCACA GTCTGGAACA CATTATTCCT GGTTTCGCCG ACTCCGCCCC CAUGGTAGAA
CGCCTGATCA CCCATGAAAA ACTCGCGTTT
```

3. Explain the outcome. Why are the results the way they are?

4. Use the previous sequence and run it in **ORF Finder**. Looking at the first frame only, what do you notice? How is **ORF Finder** different from **TISHunter**?

5. Why is it so important to find the "real" start signals?

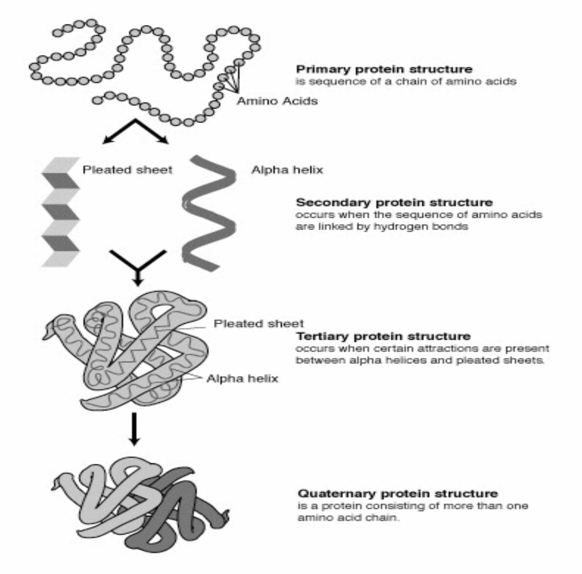
References:

a. *Systomics Network Bioinformatics Core. "TISHunter",< http://biocluster.ucr.edu/~hli/tishunter.html>.*

b. *National Center for Biotechnology Information (NCBI). "ORF Finder (Open Reading Frame Finder)". <http://www.ncbi.nlm.nih.gov/gorf/gorf.html>.*

Protein Structure

Background: Proteins are an important class of biological macromolecules present in all biological organisms made up of elements such as carbon, hydrogen, nitrogen, oxygen, and sulfur. All proteins are polymers of amino acids. The polymers, also known as polypeptides, consist of 20 different amino acids. For chains under 40 residues the term peptide is frequently used instead of protein. To be able to perform their biological function, proteins fold into one, or more, specific spatial conformations, driven by a number of noncovalent interactions such as hydrogen bonding, ionic interactions, Van der Waals' forces and hydrophobic packing. In order to understand the functions of proteins at a molecular level, it is often necessary to determine the three dimensional structure of proteins.


Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.
Alpha helix
Quaternary protein structure
is a protein consisting of more than one amino acid chain.

Purpose: This lab is intended to get a basic understanding of protein structures and why they are important.

Resources: http://www.ebi.ac.uk/2can/tutorials/structure/index.html

Key Terms:

- Protein Structure
- Peptides
- Macromolecular Structure
- Amino Group

Directions: Using the internet and MSD complete the exercises below.

Exercises:

1. How is discovering the structure of proteins important?

2. How many three dimensional protein structure coordinates are known and where are they stored?

3. How does a protein determine its physical structure?

Go to this website: http://www.ebi.ac.uk/msd-srv/msdpro/index.html, then click on “View MSDPro Demo”. This website contains information about how to use/search the Macromolecular Structure Database. After watching the demo, continue with this lab.

Run the same search as performed in the movie. The movie can be paused or rewound at any time using the movie control bar above the flash movie. After running the same search do not expect the same results. This database is always updated with new structures.

4. When searching how many results did the query return?

5. Click on the PDB ID code that displays 1dg8. A window opens up. What is the name of this protein? (Go back to the software after retrieving the search results. Click on the resolution tab to sort all the resolutions. Select all of the results with the resolution 2. Then click view from the bottom buttons.) Explain what happens after you click view?

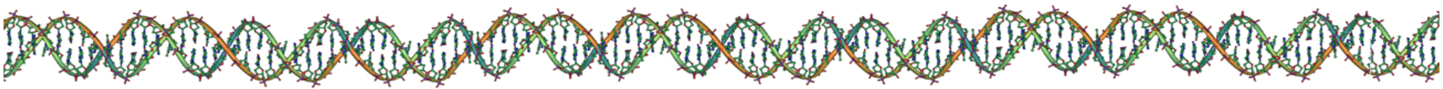
6. What is noticeable about the protein sequences at the bottom of **AstexViewer** and the overall view of the protein(s) in the above window?

7. When zooming out you should notice that there is a region in the protein(s) three dimensional view that does not overlap. By looking at the protein sequences below, pinpoint where this might be in the sequences. Hint: Examine the places that do not match bases at the same position with the other proteins. It is acceptable to click on the base to get a close up view of it.

References:

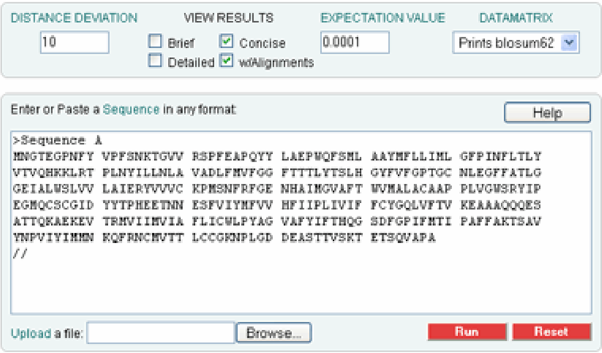
1. *EMBL-EBI. "2Can Support Portal – Protein Structure". <http://www.ebi.ac.uk/2can/tutorials/structure/index.html>.*
2. *EMBL-EBI. "Macromolecular Structure Database (MSDPro)". <http://www.ebi.ac.uk/msdsrv/msdpro/index.html>.*

Protein Function



Background:

Like other biological macromolecules such as polysaccharides and nucleic acids, proteins are essential parts of organisms and participate in every process within cells. Many proteins are enzymes that catalyze biochemical reactions, and are vital to metabolism. Proteins also have structural or mechanical functions, such as actin and myosin in muscle, and the proteins in the cytoskeleton, which forms a system of scaffolding that maintains cell shape. Other proteins are important in cell signaling, immune responses, cell adhesion, and the cell cycle. Proteins are also a necessary part of our diet, since animals cannot synthesise all the amino acids and must obtain essential amino acids from food. Through the process of digestion, animals break down ingested protein into free amino acids that can be used for protein synthesis.



DISTANCE DEVIATION: 10

VIEW RESULTS: □ Brief ☑ Concise □ Detailed ☑ w/Alignments

EXPECTATION VALUE: 0.0001

DATAMATRIX: Prints blosum62

Enter or Paste a Sequence in any format

```
>Sequence A
MNGTEGPNFY VPFSNKTGVV RSPFEAPQYY LAEPWQFSML AAYMFLLIML GFPINFLTLY
VTVQHKKLRT PLNYILLNLA VADLFMVFGG FTTTLYTSLH GYFVFGPTGC NLEGFFATLG
GEIALWSLVV LAIERYVVVC KPMSNFRFGE NHAIMGVAFT WVMALACAAP PLVGWSRYIP
EGMQCSCGID YYTPHEETNN ESFVIYMFVV HFIIPLIVIF FCYGQLVFTV KEAAAQQQES
ATTQKAEKEV TRMVIIMVIA FLICWLPYAG VAFYIFTHQG SDFGPIFMTI PAFFAKTSAV
YNPVIYIMMN KQFRNCMVTT LCCGKNPLGD DEASTTVSKT ETSQVAPA
//
```

Purpose:

The purpose of this lab is to get a basic understanding of the **FingerPRINTScan** program and how to use it to get information about protein functions.

Resources:

http://www.ebi.ac.uk/2can/tutorials/function/FingerPRINTScan.html
http://www.ebi.ac.uk/printsscan/

Key Terms:

- Protein Function
- FingerPRINTScan
- Codon
- Prokaryote

Directions:

Using the internet and **FingerPRINTScan** complete the exercises below.

Exercises:

1. What is the **FingerPRINTScan** program and how is it used?

2. What is the software particularly good at finding?

3. What is a motif?

4. What is the question we are looking to answer when we use **FingerPRINTScan**?

5. How does the **FingerPRINTScan** work? What is the "method"?

6. What are the four parameters that you can change when running a **FingerPRINTScan** search?

Use the sequence below with the **FingerPRINTScan** software to answer the following questions.

```
ATGAGCATCCCGCACTGGTGGGACCAGCTGCGGGCTGGCAGCTCGGAGGTGGACTGGTGC
GAGGACAACTACACCATCGTGCCTGCCATCGCCGAGTTCTACAACACGATCAGCAACGTC
TTATTTTTCGTTTTACCGCCCATCTGCATGTGCTTGTTTCGCCAGTATGCAACATGCTTC
AACAGTGGCATCTACCTAATATGGACTCTCTTAGTTGTAGTGGGAATTGGATCCGTCTAC
TTCCATGCAACCTTAAGTTTCTTGGGTCAGATGCTTGATGAACTTGCAATCCTTTGGGTT
CTGATGTGTGCCCTGGCCATGTGGTTCCCCAGAAGGTATCTACCAAAGGTCTTTCGGAAT
GACAGGGGCAGGTTCAAGGCGGTGGTCTGCGTCCTTTCTGCAGTTACAACATGCCTGGCG
TTTGTCAAGCCTGCCATCAACAACATCTCTCTGATGACCCTGGGGGTTCCATGCACTGTG
CTGCTCATTGCAGAGCTAAGGAGGTGTGACAATGTGCGTGTCTTTAAACTGGGCCTCTTC
TCTGGCCTCTGGTGGACCCTCGCGCTCTTCTGCTGGATCAGTGACCGAGCCTTCTGTGAG
CTGCTGTCCTCCTTCCACTTTCCCTACCTGCATTGCGTGTGGCACATCCTCATCTGCCTT
GCTGCCTACCTGGGCTGCGTATGCTTTGCCTACTTTGATGCTGCTTCCGAGATCCCCGAG
CAGGGCCCCGTCATCAAGTTCTGGCCCAGCGAGAAATGGGCCTTCATCGGGGTCCCCTAC
GTGTCCCTCCTGTGTGCCAGCAAGAAGTCACCGGTCAAGATCACGTGA
```

7. After you have run the search give the alignment report (screenshot).

8. What "FingerPrint" returned with the best score and what was the score?

9. What organism does this sequence appear to be from?

10. Now re-run the search and rearrange some of the parameters. Record the results. Explain what changed from the original results and why it did. Be sure to list every detail of the searches so they can be reproduced if needed.

References:

1. *EMBL-EBI. "2Can Support Portal – Protein Function". <http://www.ebi.ac.uk/2can/tutorials/function/FingerPRINTScan.html>.*
2. *EMBL-EBI. "FingerPRINTScan". < http://www.ebi.ac.uk/printsscan/>.*

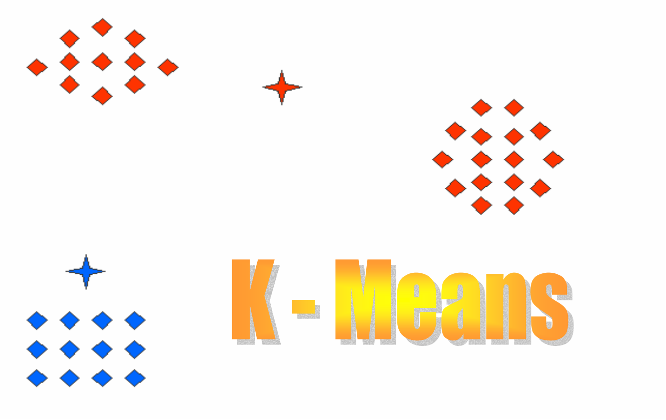
Chapter 5

Classification and Pattern Recognition

K-means Clustering

Background:

K-means is one of the simplest unsupervised machine learning algorithms. The algorithm clusters objects based on attributes into k groups. It has a vast amount of applications in many fields. One of the applications of K-means in Bioinformatics is the analysis of gene expression data.



Purpose:

This lab is intended to get a basic understanding of K-means clustering and the fundamental notions behind this algorithm used in machine learning.

Resources:

K-means Tutorial:
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

Use the K-means Java applet below:
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

Key Terms:

- Unsupervised Learning
- Clustering
- Objective Function
- Centroid

Directions:

Using the internet and the K-means java applet complete the exercises below.

Exercises:

1. Explain briefly how the K-means algorithm works.

2. What applications is it used for?

3. What are it's advantages and disadvantages?

Look at the K-means tutorial referenced above. Go to the K-means Java applet and read the getting started section at the bottom of the applet. Now, run an arbitrary simulation.

4. Describe what is happening. What specific things do you observe?

5. Describe happens when you add more clusters? Describe what happens when you add more data points.

6. Next run an arbitrary simulation with the default values of data=100 and clusters=3. What do you observe when you change the metric from Euclidean to Manhattan?

7. Change the metric back to Euclidean and run simulations again by changing the location of the clusters and/or data points. Explain what you observe when you do this. Why do you think this is happening in terms of what you learned in the tutorial?

8. Using the internet or your own creativity (preferably): How could K-means Clustering be useful in Bioinformatics?

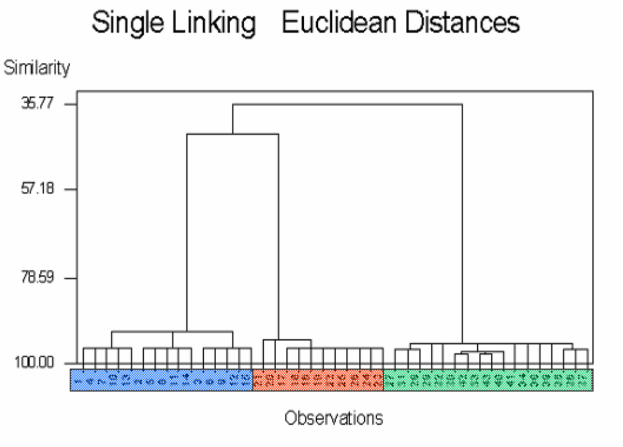
References:

1. *Clustering - K-means. "A Tutorial on Clustering Algorithms". <http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html>.*
2. *K-means – Interactive demo. "A Tutorial on Clustering Algorithms". <http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html>.*

Hierarchical Clustering

Background:

In Hierarchical clustering the data is separated into groups by a series of partitions which may run from one cluster to n clusters. Hierarchical clustering is used in various fields such as Bioinformatics, networking, physics and many more.



Purpose:

This lab is intended to get a basic understanding of Hierarchical clustering and the fundamental notions behind the method.

Resources:

Hierarchical Clustering Tutorial:
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

Use the Hierarchical Clustering Java Applet:
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html

Key Terms:

- Single-linkage
- Complete-linkage
- Average-linkage
- Agglomerative

Directions:

Read the tutorial thoroughly. Define each of the key terms above and then answer the following questions.

Exercises:

1. How many clusters do you start with when using Hierarchical Clustering? And do you have to define the amount of clusters as you did with K-means? What are the advantages and/or disadvantages of hierarchical clustering?

2. Now go to the site for the Hierarchical clustering applet listed above and run the applet on an example using the default parameters. Briefly explain what is happening and why.

3. Using the same example, try moving some points around on the line and step through the example again. What do you observe and why (if you do not observe a change, make more changes until you do)? Try changing the parameters as well.

4. Contrast K-Means and Hierarchical Clustering. What are their similarities and what are their differences? Do these methods provide the same information about the data?

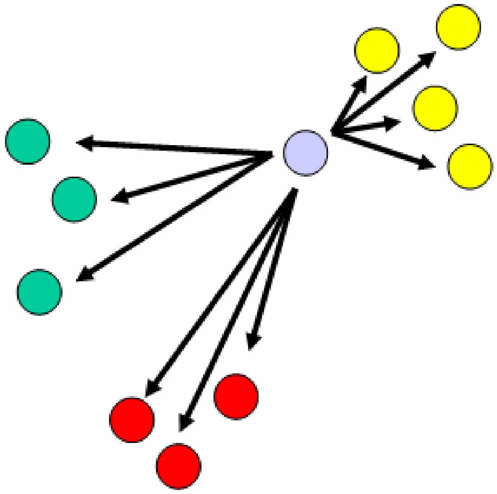
5. Using the internet or your own creativity (preferably): How can Hierarchical Clustering be applied in Bioinformatics?

References:

1. *Clustering – Hierarchical. "A Tutorial on Clustering Algorithms". <http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html>.*
2. *Hierarchical Clustering – Interactive demo. "A Tutorial on Clustering Algorithms". <http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html>.*

Introduction to Maxim Toy and Data Modifier

Background:



Maxim is a simple data classification algorithm which has many attractive properties and performs essentially as well as or better than support vector machines in some cases[1].

Purpose: This lab is intended to use and understand **Maxim Toy** and the **Data Modifier**.

Resources: The software and paper can be found at: http://www.kibazen.com/binf/

Key Terms:

- Supervised Learning
- Data Classification
- Data Modifier
- Objective Function
- Maxim Toy

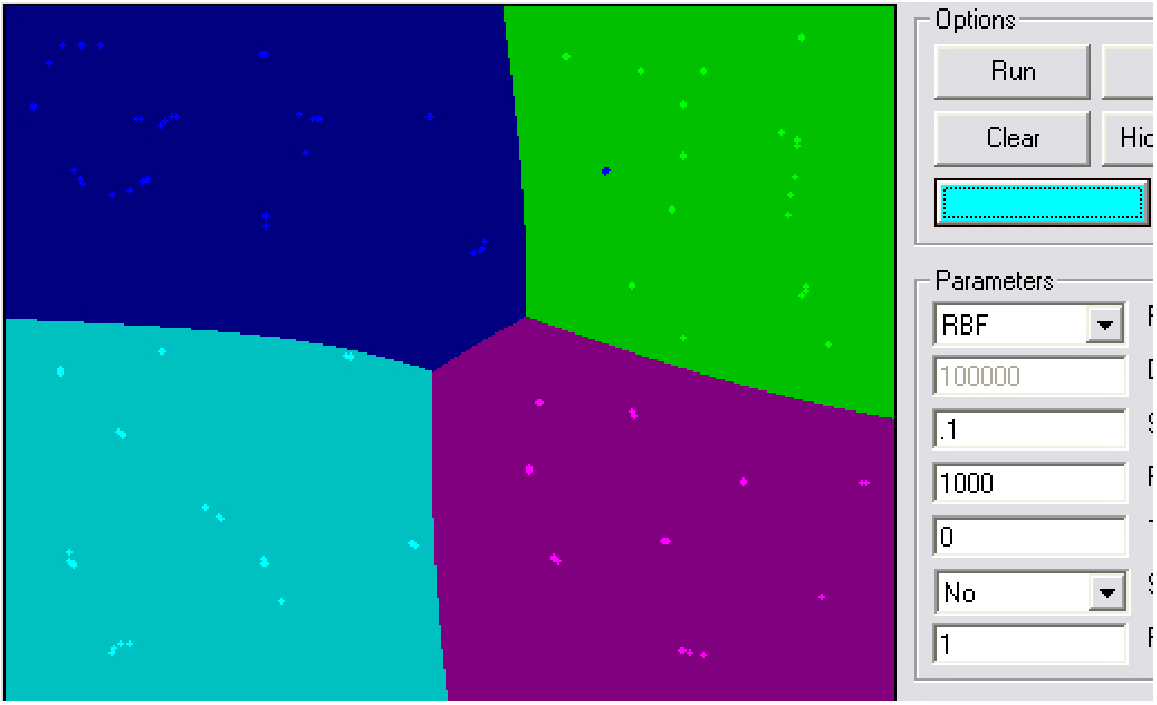
Directions: Using the Maxim Toy software and data modifier complete the tutorial below.

Exercises:

In this tutorial we explore some of the concepts of data classification using the Maxim Toy.

1. Open the **Maxim Toy** software. Click on the drawing space provided and create a few dots with a particular color. Now switch to another color and do the exact same thing. These dots in essence represent data; every color represents a different class of data. Hence, we now have two classes of data represented by the two colors. You should now click run. Examine the output drawing created by the **decision function**, how do you interpret it?

This is an example of how **Maxim Toy** might look after you run it using four colors of dots. *Note: It does not matter where you place the dots; this is just a toy example to get a basic understanding of the concepts in data classification.*



RBF

100000

.1

1000

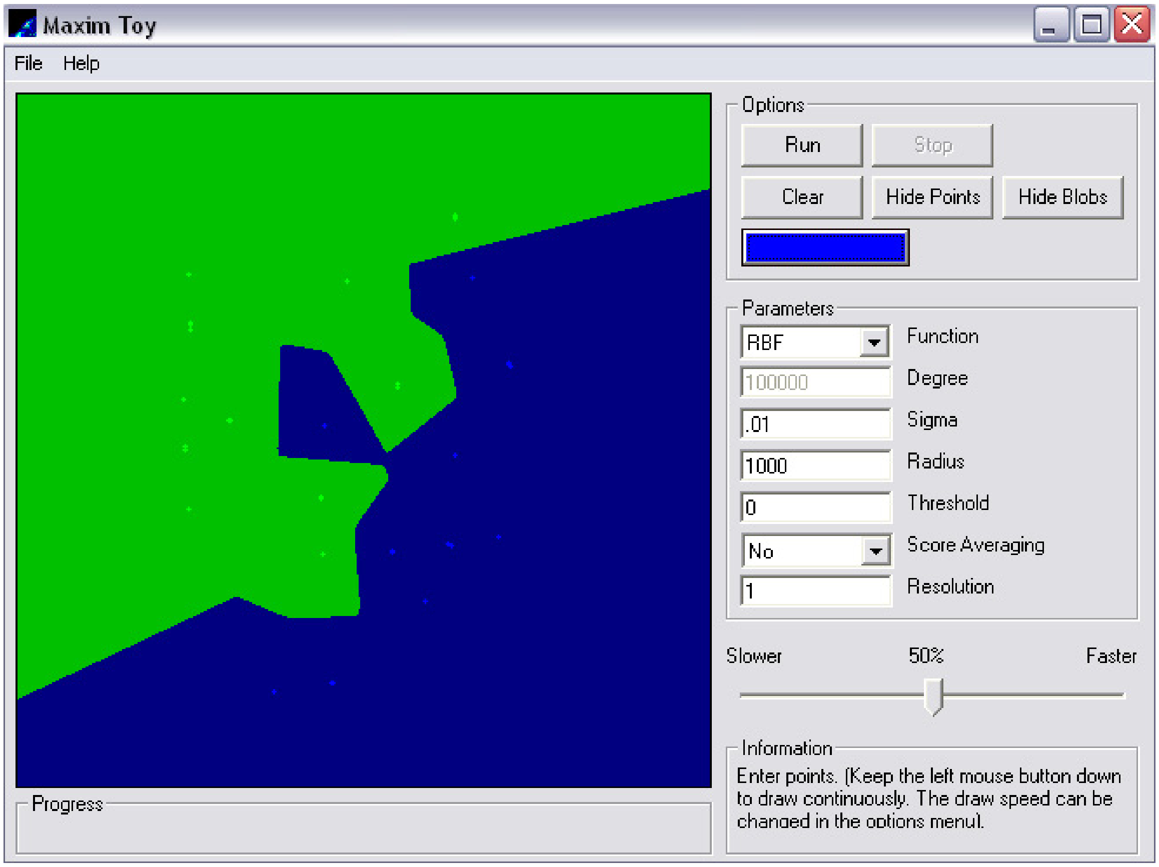
0

No

1

2. Run the software again setting the function parameter to RBF, and play around with some of the different parameters, such as Sigma. Record your findings.

3. Now let us look at the specific example below. If we had an unknown data object in the green area, which class do you think this item should be classified in?



Function: RBF

Degree: 100000

Sigma: .01

Radius: 1000

Threshold: 0

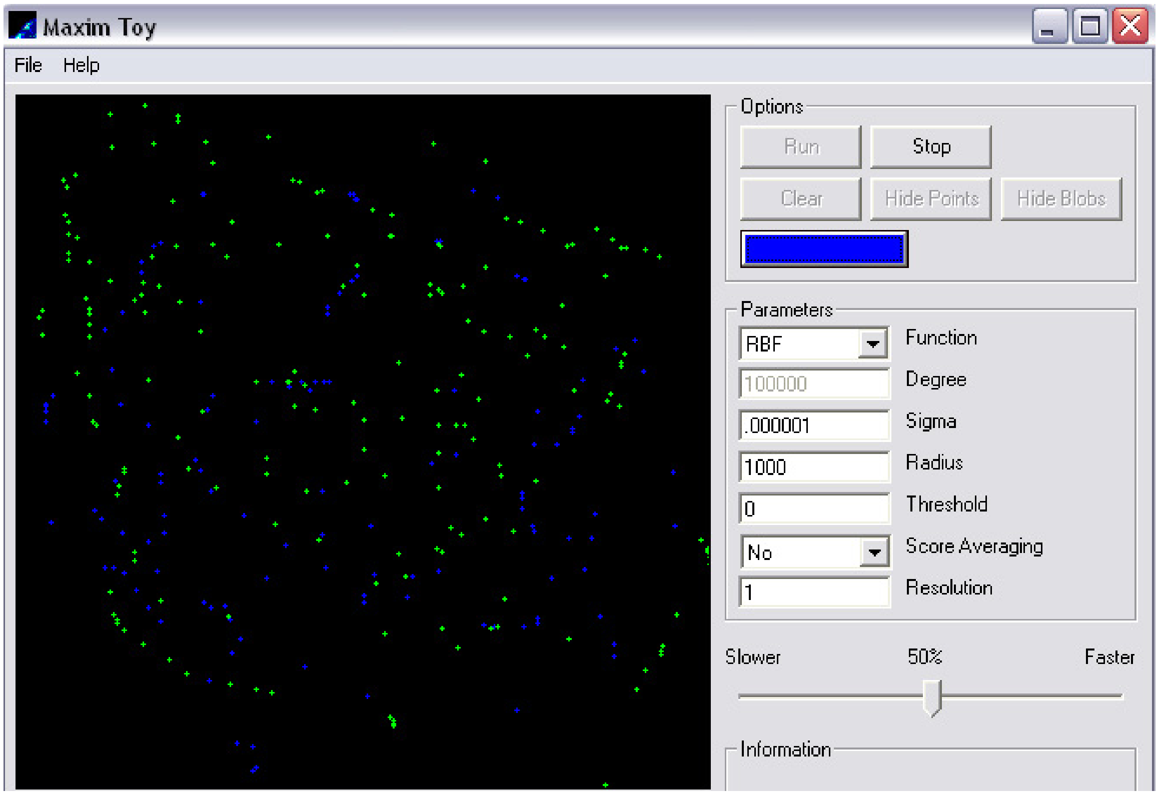
Score Averaging: No

Resolution: 1

Enter points. (Keep the left mouse button down to draw continuously. The draw speed can be changed in the options menu).

4. We will now examine two very important concepts in data classification. These concepts are called overfitting, and underfitting[1]. Click on the clear button. You should now have a blank screen. Enter some more dots on the screen just like you did before using two different colors. Make sure the function parameter is set to RBF. You will also want to set your sigma parameter very low, try something in the range of .000001. Click run; examine the results, why do you think this happened?

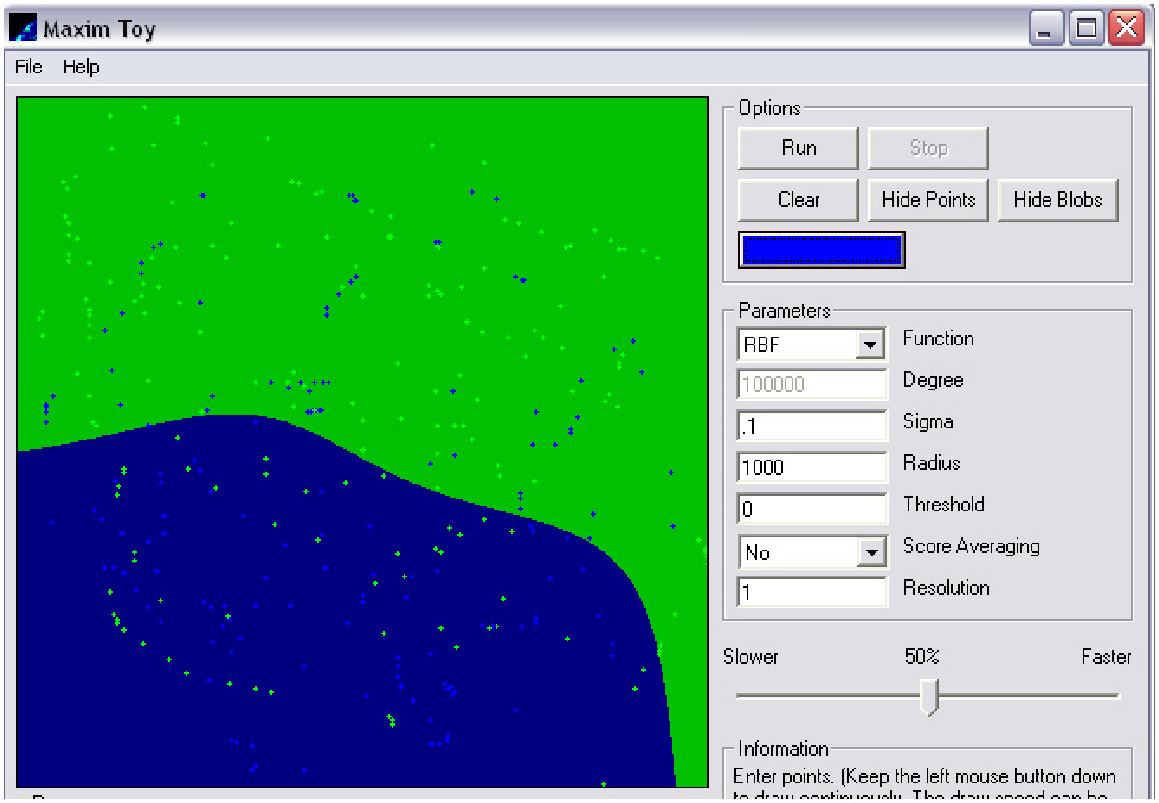
This concept is called overfitting. It is when the decision function generalizes very little. This happened because of the very low sigma. Here is an example of over fitting below:



RBF Function
100000 Degree
.000001 Sigma
1000 Radius
0 Threshold
No Score Averaging
1 Resolution

5. Now add random dots everywhere using the two colors similar to the screen above; make sure some of them are intersecting. Now with the function parameter still set to RBF, increase the sigma parameter to .1; this may be sufficient for your classification. If not, increase it a bit more. What do you notice about this classification?

The concept illustrated by your output is called underfitting. The decision function does not discriminate enough causing classification errors. You clearly see a lot of blue points in the green area and vise versa.



RBF	Function
100000	Degree
.1	Sigma
1000	Radius
0	Threshold
No	Score Averaging
1	Resolution

Enter points. (Keep the left mouse button down

Using the Data Modifier

We provide this tutorial for students who are not familiar with programs for data manipulation. You will need to learn how to use the data modifier (or equivalent program) to complete the rest of the data classification projects. This is a brief tutorial on how to use the data modifier tool. For this tutorial do not pay any attention to the Change Class field, as we will not be explaining it in this tutorial. *Note: If the data modifier does not run on your computer you will need to update your .NET framework.*

1. The **data modifier** is a very useful tool when preparing data for use with classification tools such as **Maxim**. It will allow you to remove an attribute from both the training and testing files in one run. We will examine how this is done.

2. To run **data modifier**, you will need information about your data files such as the size of the training and testing sets, the number and description of the data attributes and the number of classes. In this example we will use real-world data dealing with liver cancer; the info file reads:

   BUPA Liver Disorders

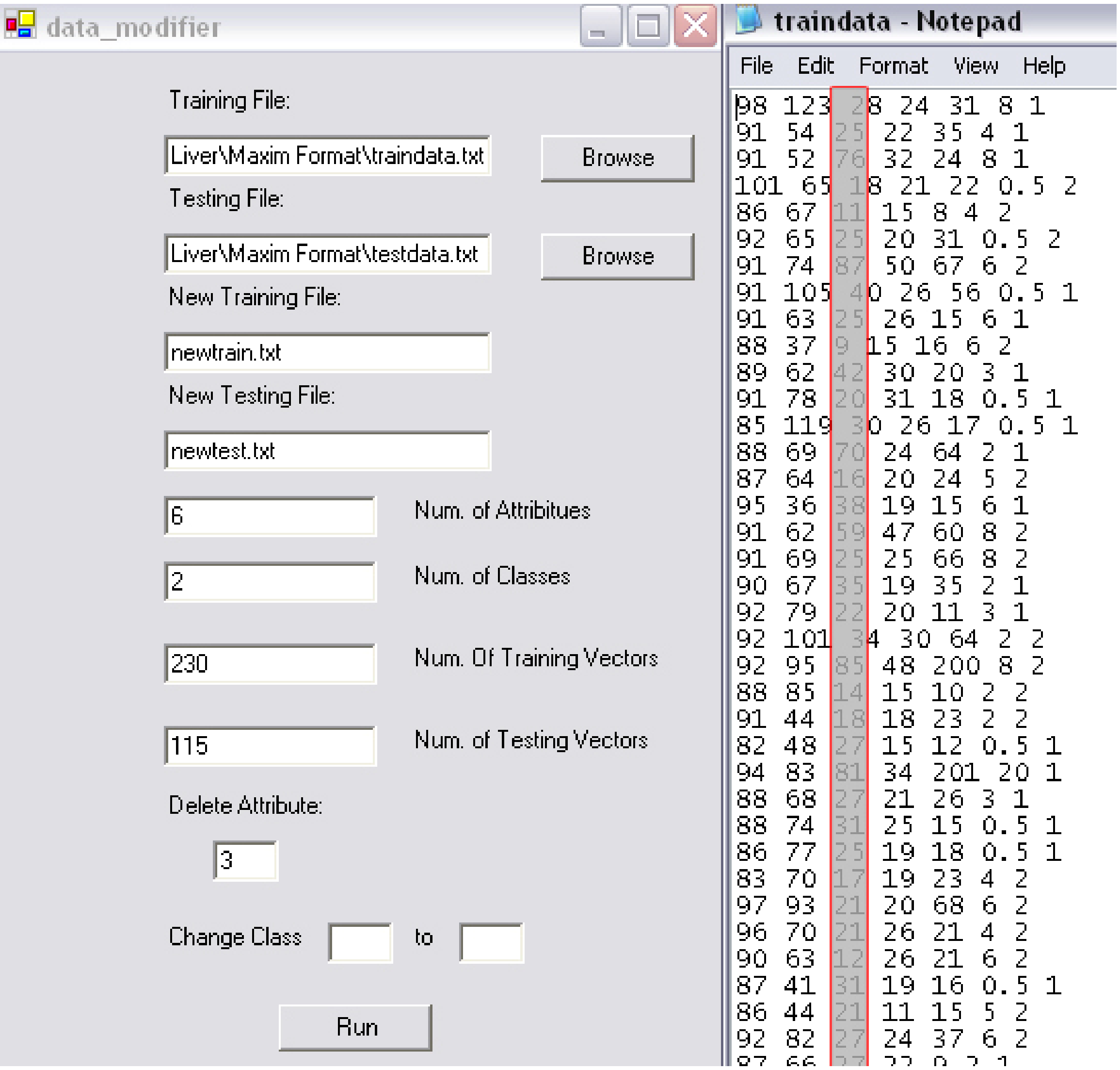
   Train Size: 230
   Test Size: 115
   Attributes: 6
   Classes: 2

3. Examine the screen shot for the data modifier below. The first entry field is the location of the training file and the second entry is the location of the testing file. In this example the files are located in the same folder.

4. The next two fields are used for naming the new training and the new testing files. Every time you modify your files you should change the filenames. Otherwise your original data file will be overwritten by the new updated files.

5. The next four fields contain data about the training and testing files. When using the *BUPA Liver Disorders* data set, the information is located in the file: *info file* and is also described in step#2 above.

6. The next field is used for the column number of the attribute field to be deleted. In the example below, the highlighted attribute number three is to be deleted.



Training File: Liver\Maxim Format\traindata.txt

Testing File: Liver\Maxim Format\testdata.txt

New Training File: newtrain.txt

New Testing File: newtest.txt

6 Num. of Attributes

2 Num. of Classes

230 Num. Of Training Vectors

115 Num. of Testing Vectors

Delete Attribute: 3

Change Class to

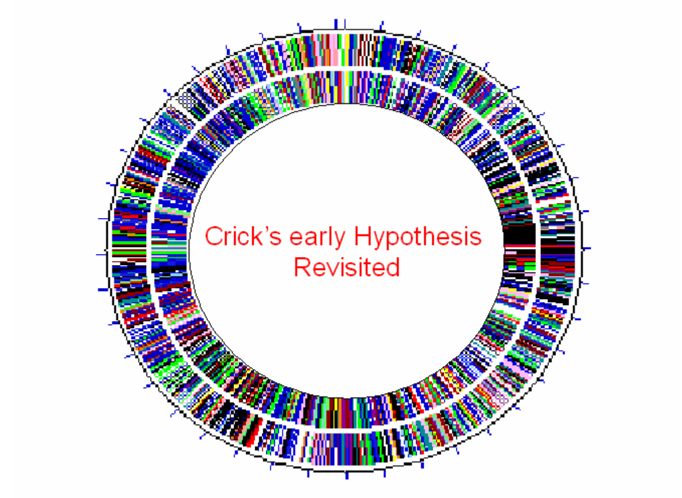
```
98 123 28 24 31 8 1
91 54 25 22 35 4 1
91 52 76 32 24 8 1
101 65 18 21 22 0.5 2
86 67 11 15 8 4 2
92 65 25 20 31 0.5 2
91 74 87 50 67 6 2
91 105 40 26 56 0.5 1
91 63 25 26 15 6 1
88 37 9 15 16 6 2
89 62 42 30 20 3 1
91 78 20 31 18 0.5 1
85 119 30 26 17 0.5 1
88 69 70 24 64 2 1
87 64 16 20 24 5 2
95 36 38 19 15 6 1
91 62 59 47 60 8 2
91 69 25 25 66 8 2
90 67 35 19 35 2 1
92 79 22 20 11 3 1
92 101 34 30 64 2 2
92 95 85 48 200 8 2
88 85 14 15 10 2 2
91 44 18 18 23 2 2
82 48 27 15 12 0.5 1
94 83 81 34 201 20 1
88 68 27 21 26 3 1
88 74 31 25 15 0.5 1
86 77 25 19 18 0.5 1
83 70 17 19 23 4 2
97 93 21 20 68 6 2
96 70 21 26 21 4 2
90 63 12 26 21 6 2
87 41 31 19 16 0.5 1
86 44 21 11 15 5 2
92 82 27 24 37 6 2
```

7. After you have run this program it will create the files: newtrain.txt and a newtest.txt. To delete another attribute, simply enter the name and location of newtrain.txt and newtest.txt in the first and second fields, adjust the number of attributes from 6 to 5 since we previously deleted the 3rd attribute and change the names of your output files.

References:

1. Axel Bernal, Tayfun Karadeniz, Karen Hospevian, Jean-Louis Lassez, **"Similarity Based Classification"** *Advances in Intelligent Data Analysis V* 2810, Springer Berlin, 2003, 187-197.
2. Tayfun Karadeniz & Jean-Louis Lassez: ***"Maxim Toy"***. 2 August 2002.
3. UC Irvine Machine Learning Repository. ***"Liver Disorders Data Set"***. <http://archive.ics.uci.edu/ml/datasets/Liver+Disorder>. (28 Aug. 2007).

Introduction to Universal Frame Finder



Background: In 1957 Crick hypothesized that the genetic code was a comma free code. This property would imply the existence of a universal coding frame and make the set of coding sequences a locally testable language. As the link between nucleotides and amino acids became better understood, it appeared clearly that the genetic code was not comma free. Crick then adopted a radically different hypothesis: the "frozen accident". However, the notions of comma free codes and locally testable languages are now playing a role in DNA Computing, while circular codes have been found as subsets of the genetic code. We revisit Crick's 1957 hypothesis in that context. We show that coding sequences from a wide variety of genes from the three domains, eukaryotes, prokaryotes and archaea, have a property of testable by fragments, which is an adaptation of the notion of local testability to DNA sequences. These results support the existence of a universal coding frame, as the frame of a coding sequence can be determined from one of its fragments, independently from the gene or the organism.

Purpose: This lab is intended for one to become familiar with, and practice using, the universal frame finder tool, which can be used to find the frame for the coding sequence of any organism.

Resources:

- NCBI has databases which contain a large amount of genomes and genes from a various range of organisms. ftp://ftp.ncbi.nlm.nih.gov/genomes/.

The software and paper can be found at: http://www.kibazen.com/binf/

Key Terms:

- Open Reading Frame
- Universal Frame Finder
- Trinucleotide
- FASTA Format

Directions: The three sets (**T_0** , **T_1** , & **T_2**) of trinucleotides are defined as follows:

T_0 = {AAA, AAC, AAT, ACC, ATC, ATT, CAG, CTC, CTG, GAA, GAC, GAG, GAT, GCC, GGC, GGT, GTA, GTC, GTT, TAC, TTC, TTT}

T_1 = {ACA, ATA, CCA, TCA, TTA, AGC, TCC, TGC, AAG, ACG, AGG, ATG, CCG, GCG, GTG, TAG, TCG, TTG, ACT, TCT, CCC}

T_2 = {CAA, TAA, CAC, CAT, TAT, GCA, CCT, GCT, AGA, CGA, GGA, TGA, CGC, CGG, TGG, AGT, CGT, TGT, CTA, CTT, GGG}

Assume that the sequence:
ATGTATCTTAATCGTCCAGTTAATGTCAAAACAGAAGTTAAAAGATTAAGAACAACTTAA

represents a gene. This sequence can be translated into three sets of trinucleotides, which can be represented using 0's, 1's and 2's based on which of the sets T_0, T_1 and T_2 the trinucleotide belongs.

ATG-TAT-CTT-AAT-CGT-CCA-GTT-AAT-GTC-AAA-ACA-GAA-GTT-AAA-AGA-TTA-AGA-ACA-ACT-TAA translates to frame

F_0: 1 2 2 0 2 1 0 0 0 0 1 0 0 0 2 1 2 1 1 2

A-**TGT-ATC-TTA-ATC-GTC-CAG-TTA-ATG-TCA-AAA-CAG-AAG-TTA-AAA-GAT-TAA-GAA-CAA-CTT**-AA translates to frame

F_1: 2 0 1 0 0 0 1 1 1 0 0 1 1 0 0 2 0 2 2

AT-**GTA-TCT-TAA-TCG-TCC-AGT-TAA-TGT-CAA-AAC-AGA-AGT-TAA-AAG-ATT-AAG-AAC-AAC-TTA**-A translates to frame

F_2: 0 1 2 1 1 2 2 2 2 0 2 2 2 1 0 1 0 0 1

Next, a sliding window of a fixed length is used to generate overlapping windows. Using the numerical sequences for the frames above the following testing vectors F_0, F_1 and F_2 are generated using a window of size 10.

F_0	**F_1**	**F_2**

1 2 2 0 2 1 0 0 0 0
2 2 0 2 1 0 0 0 0 1
2 0 2 1 0 0 0 0 1 0
0 2 1 0 0 0 0 1 0 0
2 1 0 0 0 0 1 0 0 0
1 0 0 0 0 1 0 0 0 2
0 0 0 0 1 0 0 0 2 1
0 0 0 1 0 0 0 2 1 2
0 0 1 0 0 0 2 1 2 1
0 1 0 0 0 2 1 2 1 1
1 0 0 0 2 1 2 1 1 2
	2 0 1 0 0 0 1 1 1 0
	0 1 0 0 0 1 1 1 0 0
	1 0 0 0 1 1 1 0 0 1
	0 0 0 1 1 1 0 0 1 1
	0 0 1 1 1 0 0 1 1 0
	0 1 1 1 0 0 1 1 0 0
	1 1 1 0 0 1 1 0 0 2
	1 1 0 0 1 1 0 0 2 0
	1 0 0 1 1 0 0 2 0 2
	0 0 1 1 0 0 2 0 2 2
		0 1 2 1 1 2 2 2 2 0
		1 2 1 1 2 2 2 2 0 2
		2 1 1 2 2 2 2 0 2 2
		1 1 2 2 2 2 0 2 2 2
		1 2 2 2 2 0 2 2 2 1
		2 2 2 2 0 2 2 2 1 0
		2 2 2 0 2 2 2 1 0 1
		2 2 0 2 2 2 1 0 1 0
		2 0 2 2 2 1 0 1 0 0
		0 2 2 2 1 0 1 0 0 1

Once the testing vectors F_0, F_1 and F_2 are generated, the training vectors C_0, C_1, and C_2 are generated in the same manner from a curated gene sequence. Only the two sets of training vectors C_0 and C_2 are used. $C_{1,}$ exhibiting properties of C_0 and C_2, is not used because it has the potential to generate noise.

The program used in this lab is called the Universal Frame Finder. This program is used as follows:

1. The first step is to find a few coding sequences to be used as input. NCBI has databases which contain a large amount of genomes and genes from a various range of organisms: ftp://ftp.ncbi.nlm.nih.gov/genomes/

   Look for gene files in the FASTA format, which have the extension ".ffn".

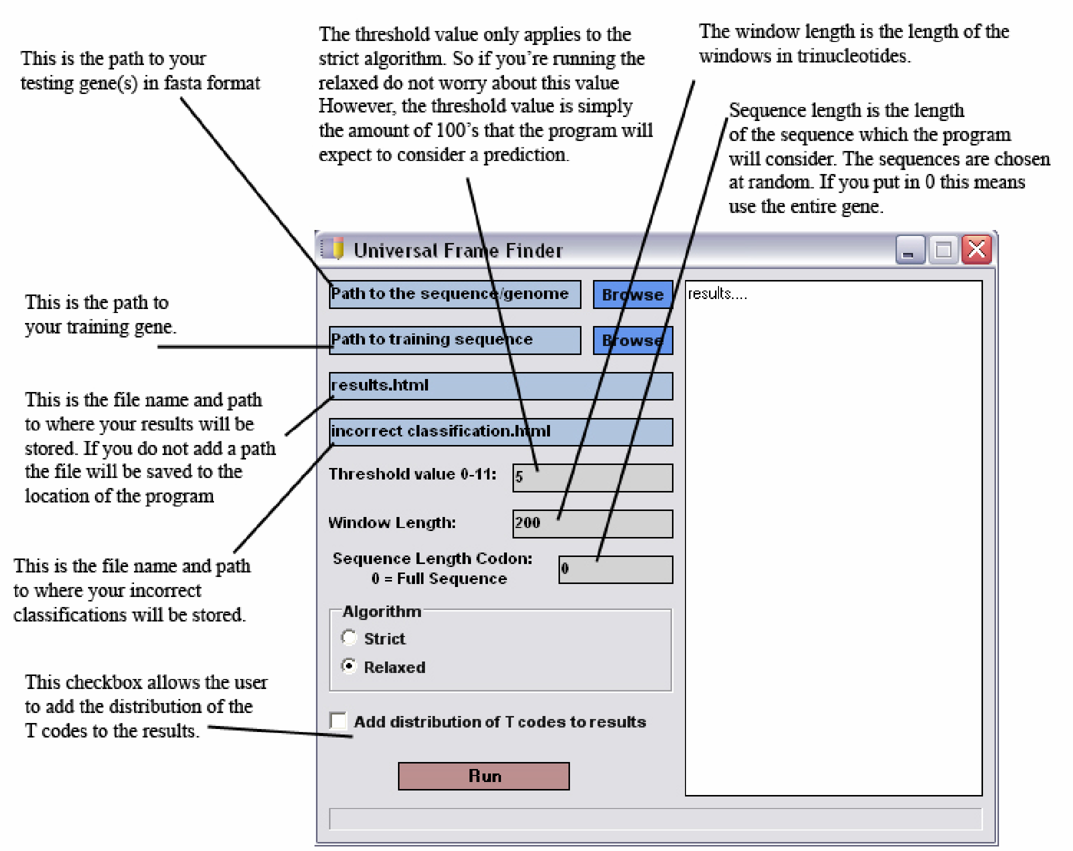
2. The next step is to input the coding sequences file into the program by clicking the browse button on the right side of the textbox, which is labeled: **"Path to sequence/genome."** Select the file with the coding sequences that were saved in step 1.

3. Now select a coding sequence from which a training set will be built. In the program window click the browse button on the right side of the textbox which is labeled **"Path to training sequences."** Next, select the file with the coding sequence, which will be used to create the training set.

4. The name of the output file can be changed by entering a new name in the text box labeled "**results.html"**. The only constraint is that the file name must end with .html in order to view the results properly.

5. Leave the rest of the settings on default. Next, click **"Run"** and wait for the results. The operating system may report that the "program is not responding", but if you wait long enough you should get results. The results are stored in results.html, unless you have changed it. The file is saved in whatever directory the program lies.

Note: When you first start you will only want to run the program with just a few genes because the computational time increases with each gene you add. You might also want to choose a small gene to use as a training set.



This is the path to your testing gene(s) in fasta format

The threshold value only applies to the strict algorithm. So if you're running the relaxed do not worry about this value However, the threshold value is simply the amount of 100's that the program will expect to consider a prediction.

The window length is the length of the windows in trinucleotides.

Sequence length is the length of the sequence which the program will consider. The sequences are chosen at random. If you put in 0 this means use the entire gene.

This is the path to your training gene.

This is the file name and path to where your results will be stored. If you do not add a path the file will be saved to the location of the program

This is the file name and path to where your incorrect classifications will be stored.

This checkbox allows the user to add the distribution of the T codes to the results.

Universal Frame Finder

Path to the sequence/genome

Path to training sequence

results.html

incorrect classification.html

Threshold value 0-11: 5

Window Length: 200

Sequence Length Codon: 0 = Full Sequence 0

Algorithm

○ Strict

● Relaxed

□ Add distribution of T codes to results

results....

Exercises:

1. Using the **URL** referenced above, or an alternative source, select genes from 6 different organisms. Make sure the entry for each gene is in the FASTA format where the first line always begins with ">". Place 5 genes into one text file. The last gene will be used to make the training set.

   Next, record what organisms the genes came from and their FASTA headers. After you have finished, follow the steps given above and run these genes using the **Universal Frame Finder** program.

   Note: The program uses the FASTA format header to determine the beginning and ending of each gene.

2. How many frames are correctly classified and how many are undecided?

3. Examine closely the results html file. What do you notice?

4. Switch the training set to another arbitrary gene and run the same five genes again. What do you notice and what is your explanation?

5. Now choose a sequence length between 10-100 and a window size. The window size must be less than the sequence size. For instance, the sequence size could be 100 and window size could be 50. Generally a 2:1 ratio seems to work fairly well.

   Try 10 different combinations and record the results. What can you conclude from your experiments?

6. What is the smallest sequence size and window size which gives good results? Why do you think this is?

7. Try increasing the sequence size and window size by increments of 10-20, what trend do you notice when the sequence size and window size becomes relatively large? **Hint:** You may get a better feeling for the results by looking at the html results file.

8. Now run the gene that is given below.

```
> TP53 Gene Homo Sapiens - Altered
ATGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCCCCTCTGAGTCAGGAAACATTTTCAG
ACCTATGGAAACTACTTCCTGAAAACAACGTTCTGTCCCCCTTGCCGTCCCAAGCAATGG
ATGATTTGATGCTGTCCCCGGACGATATTGAACAATGGTTCACTGAAGACCCAGGTCCAG
ATGAAGCTCCCAGAATGCCAGAGGCTGCTCCCCGCGTGGCCCCTGCACCAGCAGCTCCTA
CACCGGCGGCCCCTGCACCAGCCCCCTCCTGGCCCCTGTCATCTTCTGTCCCTTCCCAGA
AAACCTACCAGGGCAGCTACGGTTTCCGTCTGGGCTTCTTGCATTCTGGGACAGCCAAGT
CTGTGACTTGCACGTACTCCCCTGCCCTCAACAAGATGTTTTGCCAACTGGCCAAGACCT
GCCCTGTGCAGCTGTGGGTTGATTCCACACCCCCGCCCGGCACCCGCGTCCGCGCCATGG
CCATCTACAAGCAGTCACAGCACATGACGGAGGTTGTGAGGCGCTGCCCCCACCATGAGC
GCTGCTCAGATAGCGATGGTCTGGCCCCTCCTCAGCATCTTATCCGAGTGGAAGGAAATT
TGCGTGTGGAGTATTTGGATGACAGAAACACTTTTCGACATAGTGTGGTGGTGCCCTATG
AGCCGCCTGAGGTTGGCTCTGACTGTACCACCATCCACTACAACTACATGTGTAACAGTT
```

```
CCTGCATGGGCGGCATGAACCGGAGGCCCATCCTCACCATCATCACACTGGAAGACTCCA
GTGGTAATCTACTGGGACGGAACAGCTTTGAGGTGCGTGTTTGTGCCTGTCCTGGGAGAG
ACCGGCGCACAGAGGAAGAGAATCTCCGCAAGAAAGGGGAGCCTCACCACGAGCTGCCCC
CAGGGAGCACTAAGCGAGCACTGCCCAACAACACCAGCTCCTCTCCCCAGCCAAAGAAGA
AACCACTGGATGGAGAATATTTCACCCTTCAGATCCGTGGGCGTGAGCGCTTCGAGATGT
TCCGAGAGCTGAATGAGGCCTTGGAACTCAAGGATGCCCAGGCTGGGAAGGAGCCAGGGG
GGAGCAGGGCTCACTCCAGCCACCTGAAGTCCAAAAAGGGTCAGTCTACCTCCCGCCATA
AAAAACTCATGTTCAAGACAGAAGGGCCTGACTCAGACTGA
```

Looking at the html results what do you notice that is different with this gene and why do you think this is?

9. Take this gene and run it with **BLAST** and try to figure out why it might give you these results.

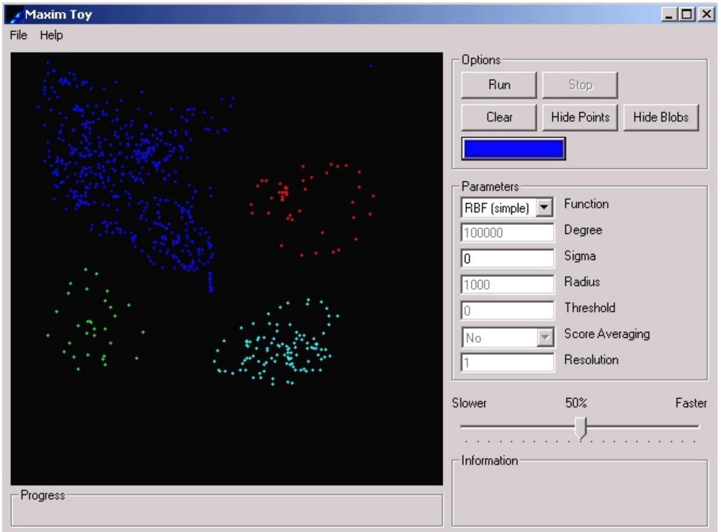
References:

1. *Jean-Louis Lassez, Ryan Rossi, Axel Bernal:* "Crick's Hypothesis Revisited: The Existence of a Universal Coding Frame." AINA/BLSC, 745-751 (2007)
2. Jean-Louis Lassez: "Circular Codes and Synchronization." IJCIS 5, 201-208 (1976)
3. D. G. Arquès, C. J. Michel, "A Complementary Circular Code in the Protein Coding Genes." *J. T. Biol.* 182, 45-58 (1996)
4. *National Center for Biotechnology Information (NCBI). "Directory of a List of Genomes".* <ftp://ftp.ncbi.nlm.nih.gov/genomes/>*.*

Maxim Parameter Project

Background:

Maxim is a simple data classification algorithm designed Dr. Jean-Louis Lassez. Maxim has many attractive properties and performs essentially as well as or better than support vector machines.



RBF (simple) Function
100000 Degree
0 Sigma
1000 Radius
0 Threshold
No Score Averaging
1 Resolution

Purpose:

This lab is intended for the user to explore the different parameters of maxim (such as the kernel function, degree, sigma and radius) using a real world data set located at the University of California-Irvine Machine Learning Data Repository. For this lab we will uses Diabetes Data Set. This data originated from the National Institute of Diabetes and Digestive and Kidney Diseases. The database was compiled by John Hopkins University. The data has been used as benchmark data for years to help classify the data mining software. This specific data set has eight attributes and two classes. The best achieved accuracy with the data set is 76%.

Resources:

The software and maxim paper can be found at: http://www.kibazen.com/binf/
http://archive.ics.uci.edu/ml/datasets.html

Key Terms:

- Underfitting
- Overfitting
- Kernel Functions
- Radial Basis Function

Directions:

Use **Maxim** to complete the exercises below. Be sure to record ALL the experiments that you perform in tables.

Exercises:

The company you are working for has assigned you the task of analyzing the Pima Indian Diabetes Data Set using the Maxim Software. You have two goals in mind: 1) get a better understanding of the Maxim software and 2) determine whether or not the Data Set can be used in machine learning and consequently be used to make predictions whether or not an individual might have diabetes.

First examine the data set and determine what the attributes are and what the two classes stand for. Look for missing values. If you find a line with a missing value delete the entire line from your data set. Next you will need to divide the data set up into two groups one used for training and one used for testing. Generally, you can use a 70%-30% ratio between your training set and your testing data set. Before the two sets are formed from the data set make sure that the lines of data are randomized. You need to get a representative sample of both of the two classes in the training set and the testing set. Save the training set as DiabetesTrain.txt and the testing set as DiabetesTest.txt.

A) Using the Maxim software and the Radial Basis Function (RBF) parameter, complete the entries for the following chart:

Data Set: Pima Indian Diabetes Set					
Test Number	**Function Used**	**Sigma**	**Radius**	**Training Vs. Training**	**Training Vs. Testing**
1.	RBF	.000001	1000		
2.	RBF	.00001	1000		
3.	RBF	.0001	1000		
4.	RBF	.001	1000		
5.	RBF	.01	1000		
6.	RBF	.1	1000		
7.	RBF	1	1000		
8.	RBF	10	10000		
9.	RBF	100	10		
10.	RBF	1000	1000		

Contrasting training vs. training and training vs. testing, examine your results and answer the following questions:

1. What percentage of correct classification would you expect when training set is tested against the training set?

2. Which test(s) give you the best results for training vs. testing?

3. Which test(s) give you the best results for training vs. training?

4. How do the varying values of Sigma affect the test results?

B) Using the Maxim software and the Polynomial Function (POLY) parameter, complete the entries for the following chart:

Data Set: Pima Indian Diabetes Set					
Test Number	**Function Used**	**Radius**	**Degree**	**Training Vs. Training**	**Training Vs. Testing**
1.	POLY	15000	2		
2.	POLY	10000	2		
3.	POLY	7500	2		
4.	POLY	5000	2		
5.	POLY	3000	2		
6.	POLY	1000	2		
7.	POLY	500	2		
8.	POLY	100	2		
9.	POLY	10	2		
10.	POLY	1	2		

5. What percentage of correct classification would you expect when training set is tested against the training set?

6. Which test(s) give you the best results for training vs. testing?

7. Which test(s) give you the best results for training vs. training?

8. How do the varying values of the Radius with the Degree constant at affect the test results?

9. Overall, which Maxim function gives the better results for this data set? What is the best classification achieved for this data set?

10. Identify any tests using either function which result in overfitting or underfitting. Hint: It might be helpful to review underfitting and overfitting by examining graphical test results using the Maxim Toy software.

References:

1. Tayfun Karadeniz & Jean-Louis Lassez: ***"Maxim"***. 2 August 2002.
2. UC Irvine Machine Learning Repository. ***"Diabetes Data Set".***
   <. http://archive.ics.uci.edu/ml/datasets/Diabetes>. (28 Aug 2007).
3. UC Irvine Machine Learning Repository. ***"Letter Recognition Data Set".***
   <http://archive.ics.uci.edu/ml/datasets/Letter+Recognition>. (28 Aug. 2007).
4. UC Irvine Machine Learning Repository. ***"Liver Disorders Data Set".***
   <http://archive.ics.uci.edu/ml/datasets/Liver+Disorder>. (28 Aug. 2007).
5. UC Irvine Machine Learning Repository. ***"Wine Data Set".***
   < http://archive.ics.uci.edu/ml/datasets/Wine>. (28 Aug. 2007).

Breast Cancer Analysis

Background:

Excluding skin cancer, breast cancer in the most common cancer found in women. According to the American Cancer Society, it's estimated that about 178,480 women in the United States will be found to have invasive breast cancer in 2007.

Purpose: This lab explores data classification using the Maxim software tool to classify breast cancer data.

Resources: The software and maxim paper can be found at: http://www.kibazen.com/binf/
http://archive.ics.uci.edu/ml/datasets.html

Key Terms:

- Maxim
- Cancer
- Breast Cancer

Directions: Use Maxim to complete the exercises below.

Exercises:

In this lab we will analyze breast cancer data and determine which attributes actually have an impact on breast cancer. You have been given two files, one is the training and one is the testing data sets. These files contain information about patients. Every row of data denotes a different patient. The ultimate goal of this project is for you to find out which attributes have a significant impact on breast cancer. You will need to record all your results. At the end of this lab there is an example on how you *might* want to do this.

0. Before answering the questions below, try running Maxim with the breast cancer data as provided in the training and testing data sets. What are the results?

1. Note, some of the patients are missing data, so you will need to go through the files and delete any patient with missing data. The missing data items in the file are represented by a question mark (?). *Hint: to find the missing data pieces you can use the 'find' function of a text editor.*

Now examine the attribute list which is below. Do you think all of the attributes are necessary?

2. Which attribute might be causing Maxim to produce results with lower accuracy? Determine which attribute is causing the noise and consequently needs to be deleted. When you find the attribute, use the data modifier program to delete the attribute.

Number of Training vectors: 499
Number of Testing vectors: 200
Number of Classes: 2
Number of Attributes: 10

There are 16 missing values denoted with a question mark.

#	*Attribute*	*Domain*
1.	*Sample code number*	*id number*
2.	*Clump Thickness*	*1 - 10*
3.	*Uniformity of Cell Size*	*1 - 10*
4.	*Uniformity of Cell Shape*	*1 - 10*
5.	*Marginal Adhesion*	*1 - 10*
6.	*Single Epithelial Cell Size*	*1 - 10*
7.	*Bare Nuclei*	*1 - 10*
8.	*Bland Chromatin*	*1 - 10*
9.	*Normal Nucleoli*	*1 - 10*
10.	*Mitoses*	*1 - 10*
11.	*Class:*	*(2 for benign, 4 for malignant)*

3. After you have corrected the two data files by deleting the column containing the attribute causing noise and deleting the row for any patient with missing data, use Maxim to analyze the breast cancer data. You will want to record all your results in tables, with the settings you used. There is an example of how you might do this below. The first thing you will want to do is test and record the Breast Cancer data unmodified.

4. Since the goal of this project is to determine which attributes have the most significance dealing with breast cancer. You should now begin deleting attributes that you think would have the most significance dealing with breast cancer. After testing and/or deleting each attribute, record your results and describe your observations. At the end of your project, report what you have concluded from your experiments.

Remember, you can delete any combination of attributes. You could even delete every single attribute and just test one attribute at a time. The decision is strictly up to you. If you have any questions, do not hesitate to ask your lab instructor.

5th Attribute Deleted

Data Tested	Function	Sigma	Training vs. Testing
Breast Cancer	RBF Simple	.000001	84.42%
Breast Cancer	RBF Simple	.00001	84.42%
Breast Cancer	RBF Simple	.0001	84.42%
Breast Cancer	RBF Simple	.001	84.42%
Breast Cancer	RBF Simple	.01	84.42%
Breast Cancer	RBF Simple	.1	84.42%
Breast Cancer	RBF Simple	1	82.41%
Breast Cancer	RBF Simple	4	79.40%
Breast Cancer	RBF Simple	10	77.89%
Breast Cancer	RBF Simple	100	77.89%
Breast Cancer	RBF Simple	1000	77.89%

Observations when testing the new data set.

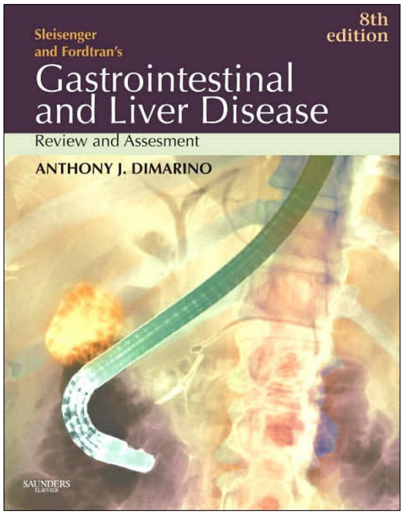
References:

1. *Axel E. Bernal, Karen Hospevian, Tayfun Karadeniz, Jean-Louis Lassez: "Similarity Based Classification". IDA 2003: 187-197.*
2. *Breast Cancer Home Page - National Cancer Institute,< http://www.breastcancer.org/>. (26 December 2007).*
3. *UC Irvine Machine Learning Repository, "Breast Cancer Wisconsin (Original) Data Set". <http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>. (28 August 2007).*

Liver Disease Analysis

Background:

Your liver is an incredibly hard working organ. Its main purpose is to filter and thereby clean liquids. Liver disease has many causes including alcoholism; however it has been estimated by the American Liver Foundation that 1 in 10 Americans suffer from liver disease.



Purpose: This lab explores the use of the Maxim software tool to classify data in a real world setting.

Resources: The software and maxim paper can be found at: http://www.kibazen.com/binf/ http://www.ics.uci.edu/~mlearn/databases/liver-disorders/bupa.names

Key Terms:

- Maxim
- Liver Disease
- Classification

Directions: Use Maxim to complete the exercises below.

Exercises:

The company you are working for has been contracted to find out the amount of drinks which would have a significant effect on liver disease. As an expert witness, you will have to stand trial and present your results. Your experiments will have a big impact on the court's decision.

You are to take the liver disease data file which has sixteen classes and merge them until you get the "best" accuracy. After you achieve the best accuracy you will want to do attribute testing. Try deleting different attributes, specifically the drinks consumed attribute and compare your results again. Report ALL your results in tables and make sure you give enough information that the experiment you do can be reproduced. After you perform your experiments and achieve the "best" accuracy, you will then explain what you have observed while performing these experiments. Remember you are going to trial, so your paper will need to be professional and reliable. Prove to your audience that the data sets are reliable (UC Irvine).

The data set you need for this exercise is found at the University of California, Irvine.
This data was created by BUPA Medical Research Ltd. The data has been used as benchmark data for years to help classify data mining software. This specific data set has six attributes and sixteen

classes. The training size is 191 and the testing size is 155. The best achieved accuracy with this data set is around 33%.

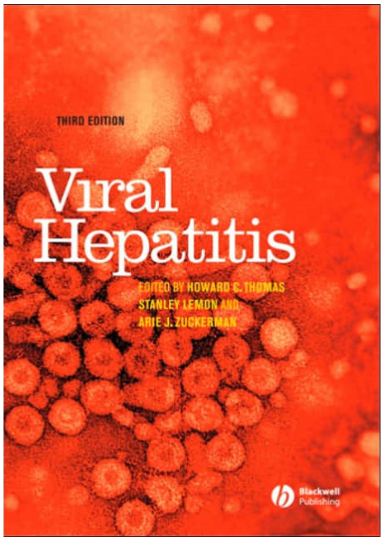
Attribute Information:		
1.	MCV	Mean Corpuscular Volume
2.	AlkPhos	Alkaline Phosphotase
3.	SGPT	Alamine Aminotransferase
4.	SGOT	Aspartate Aminotransferase
5.	GammaAGT	Gamma-Glutamyl Transpeptidase
6.	DRINKS	Number of half-pint equivalents of alcoholic beverages drunk per day
7.	CLASS	Value indicates what class

References:

1. *Axel E. Bernal, Karen Hospevian, Tayfun Karadeniz, Jean-Louis Lassez: "Similarity Based Classification". IDA 2003: 187-197.*
2. *American Liver Foundation. <http://www.liverfoundation.org/>. (26 December 2007).*
3. *UC Irvine Machine Learning Repository, "Liver Disorders Data Set". <http://www.ics.uci.edu/~mlearn/databases/liver-disorders/bupa.names>. (29 August 2007).*

Hepatitis Analysis

Background:



Hepatitis is an inflammation of the liver. It may be caused by bacterial or viral infection, parasitic infestation, alcohol, drugs, toxins, or transfusion of incompatible blood. Although many cases of hepatitis are not a serious health threat, the disease can become chronic and sometimes lead to liver failure and death.

You are employed by a small hospital, where 50 patients are treated for Hepatitis. The data from these patients will be used for the training set.

The hospital keeps an electronic record for each patient, which includes the following attributes:

```
 Attribute information:
 1. AGE: 10, 20, 30, 40, 50, 60, 70, 80
 2. SEX: male, female
 3. STEROID: no, yes
 4. ANTIVIRALS: no, yes
 5. FATIGUE: no, yes
 6. MALAISE: no, yes
 7. ANOREXIA: no, yes
 8. LIVER BIG: no, yes
 9. LIVER FIRM: no, yes
10. SPLEEN PALPABLE: no, yes
11. SPIDERS: no, yes
12. ASCITES: no, yes
13. VARICES: no, yes
14. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
15. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250
16. SGOT: 13, 100, 200, 300, 400, 500,
17. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
18. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90
19. HISTOLOGY: no, yes


Class: DIE, LIVE
```

After a natural disaster occurred in a neighboring state, most of the hospitals were left without running water and electricity. Therefore, the hospital where you are employed has to accept 30 additional Hepatitis patients (testing set).

Some of the patients are in critical condition (high probability of death), and have to be placed in the intensive care unit upon arrival. Unfortunately, the hospital is currently understaffed and doesn't have enough medical personnel to thoroughly check every patient upon arrival in order to determine the exact state of his/her condition.

Purpose:

This lab shows how classification tools such as Maxim and SVM can be used to analyze and classify data sets.

Resources: Liver Data Set, Maxim Software, SVM Light
The software and maxim paper can be found at: http://www.kibazen.com/binf/

Key Terms:

- Maxim
- Liver Disease
- Classification

Directions: Use Maxim and SVM to complete the exercises below.

Exercises:

Fortunately, the previous hospitals of the transferred patients also kept the electronic records of their patients with the attributes described above. Your goal is to automatically classify new patients and to determine, based on classification results, whether the person is in critical condition. Remember, you don't know whether all of the attributes in the patients' records are important for the classification. Therefore, you need to run an experiment on each attribute and determine whether it contains important information for the classification or is creating noise. The key is to delete each attribute and to classify the testing set without this attribute. Based on whether the accuracy of the classification increased, decreased or stayed the same you can determine whether this attribute was important for the classification.

Remember: this is a matter of life or death!

Make sure you achieve the best accuracy possible. Present the results of your experiments in a clear readable form, preferably in tables, followed by your comments. Your report should be written in such way that the director of the hospital could reproduce your experiments, should he/she doubt their integrity.

References:

1. *Axel E. Bernal, Karen Hospevian, Tayfun Karadeniz, Jean-Louis Lassez: "Similarity Based Classification". IDA 2003: 187-197.*
2. *"SVM_Light". < http://www.cs.cornell.edu/People/tj/svm_light/>. (29 Aug 2007).*

Chapter 6

Advanced Topics

HMM and Protein Sequence Analysis

Andrey Andreyevich Markov
1856 - 1922

Background: A **hidden Markov model** (**HMM**) is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example in pattern recognition applications. One application of HMM's to the field of bioinformatics is protein sequence analysis. Where you input a protein sequence into an HMM and as output you get a score representing the state probabilities of that sequence through the Hidden Markov Model.

Purpose: This lab provides a basic understanding of hidden markov models and their use in protein sequence analysis.

Resources: http://www.soe.ucsc.edu/research/compbio/ismb99.handouts/KK185FP.html

Key Terms:

- Statistical Profile
- Finite State Machine
- Overfitting
- HMM Scoring
- Markov Processes

Directions: Read the tutorial above thoroughly (or equivalent) and complete the exercises below.

Exercises:

1. What are two common applications of Hidden Markov Models?

2. What are proteins and how many amino acids are in the protein alphabet?

3. How does a protein in the parent cell change when the cell goes through mitosis?

4. What is the model shown in Figure 6?

5. How are the probabilities of the amino acids calculated?

6. What are the probabilities of CCGSS?

7. What is the actual probability of CCGSS?

8. What is the score of the sequence CCGSS?

9. What are some of the other factors the statistical model takes into account?

10. What are two good alternative scoring methods used in HMM?

11. What does the score mean in HMM?

12. Why is global scoring sometimes misleading?

13. How can you build a HMM for protein sequence analysis?

14. In sequence weighting, what is overspecializing?

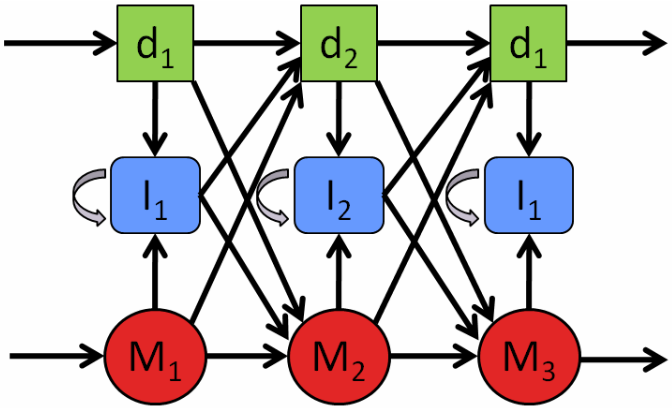
15. Explain the problem of overfitting in HMM?

16. What are the two known flaws with Hidden Markov Models?

References:

1. *Russian Academy of Sciences [Internet].Moscow, Russian [modified: 2002 May 12]. [Photo],* Markov Andrei Andreevich*; [cited: 2007 December 26][about 1 screen].Available from:* *http://www.ras.ru/win/db/show_per.asp?P=.id-53175.ln-en.dl-.pr-inf.uk-0.*
2. *Genetic Home Reference: Your Guide to Understanding Genetic Conditions [Internet]. Bethesda, MD: United States National Library of Medicine, National Institute of Health [modified: 2009 July 31]. [Illustration], DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone.; [cited 2007 July][about 3 screens]. Available from: http://ghr.nlm.nih.gov/handbook/basics/dna.*
3. <http://www.soe.ucsc.edu/research/compbio/ismb99.handouts/KK185FP.html>. (16 septembre 2007).
4. Jean-Louis Lassez, Ryan Rossi, Kumar Jeev: "Ranking Links on the Web: Search and Surf Engines," Lecture Notes of Artificial Intelligence, IEA/AIE, 199-208 (2008).

Introduction to Pfam



Background:

Pfam is a database of multiple alignments of protein domains or conserved protein regions. The alignments represent some evolutionary conserved structure which has implications for the protein's function. Profile hidden Markov models (profile HMMs) built from the Pfam alignments can be very useful for automatically recognizing that a new protein belongs to an existing protein family, even if the homology is weak. Unlike standard pairwise alignment methods (e.g. BLAST, FASTA), Pfam HMMs deal sensibly with multidomain proteins

Pfam is formed in two separate ways. Pfam-A are accurate human crafted multiple alignments, whereas Pfam-B is an automatic clustering of the rest of a nonredundant protein database derived from the PRODOM database.

Pfam is a large collection of **multiple sequence alignments** and hidden Markov models covering many common protein families. Pfam version 19.0 (December 2005) contains alignments and models for 8183 protein families, based on the Swissprot 48.1 and SP-TrEMBL 31.1 protein sequence databases.

Purpose:

This lab uses the PFAM tool to analyze protein sequences.

Resources:

http://pfam.janelia.org/

Key Terms:

- HMM
- Protein Families
- Taxonomy
- SWISS-PROT
- TrEMBL

Directions:

Use Pfam to complete the exercises below.

Exercises:

1. Run a search on the sequence below in the PFAM tool. What are the results.

```
MTELPAPLSYFQNAQMSEDNHLSNTVRSQNDNRERQEHNDRRSLGHPEPLSNGRPQGNSR
QVVEQDEEEDEELTLKYGAKHVIMLFVPVTLCMVVVVATIKSVSFYTRKDGQLIYTPFTE
DTETVGQRALHSILNAAIMISVIVVMTILLVVLYKYRCYKVIHAWLIISSLLLLFFFSFI
YLGEVFKTYNVAVDYITVALLIWNFGVVGMISIHWKGPLRLQQAYLIMISALMALVFIKY
LPEWTAWLILAVISVYDLVAVLCPKGPLRMLVETAQERNETLFPALIYSSTMVWLVNMAE
GDPEAQRRVSKNSKYNAESTERESQDTVAENDDGGFSEEWEAQRDSHLGPHRSTPESRAA
VQELSSSILAGEDPEERGVKLGLGDFIFYSVLVGKASATASGDWNTTIACFVAILIGLCL
TLLLLAIFKKALPALPISITFGLVFYFATDYLVQPFMDQLAFHQFYI
```

2. What is the name of the sequence that was aligned?

3. What is the e-value and score of the sequence?

4. What does the green/silver bar represent?

5. After you have analyzed the output. Find out what disease this protein can cause. **Hint**: click on the sequence links for more information about the sequence.

6. Now try to search using the accession number. The accession number is: PF03835. Use this link: http://pfam.janelia.org/search/ to search. What is the name of the protein?

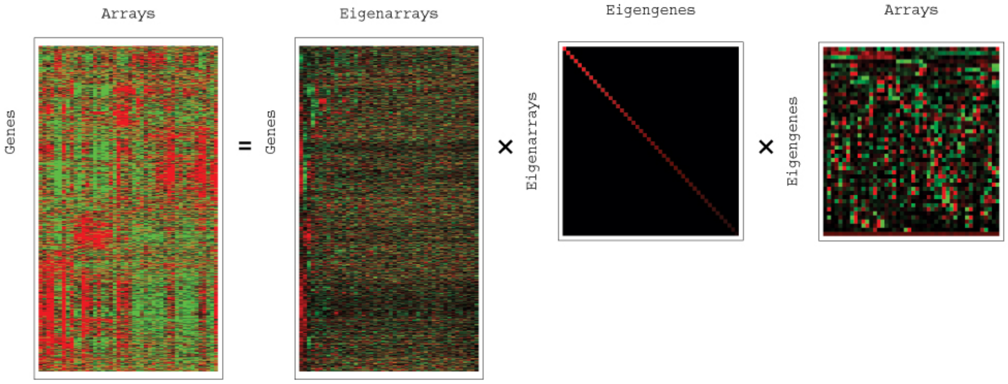
7. Based on the protein description. Explain how this protein works.

References:

1. *Howard Hughes Medical Institute (HHMI): <Pfam: clans, web tools and services>: R.D. Finn, J. Mistry, B. Schuster-Böckler, S. Griffiths-Jones, V. Hollich, T. Lassmann, S. Moxon, M. Marshall, A. Khanna, R. Durbin, S.R. Eddy, E.L.L. Sonnhammer and A. Bateman Nucleic Acids Research (2006), Database Issue 34:D247-D251. (3 September 2007).*
2. *Howard Hughes Medical Institute (HHMI): Janelia Farm Research Campus. <http://pfam.janelia.org/>. (3 September 2007).*

Singular Value Decomposition and Latent Semantic Analysis

Background: Latent Semantic Analysis (LSA) has been successfully used in applications such as Speech Recognition, Natural Language Processing, Cognitive Modeling, Document Classification and Cross Language Information Retrieval. LSA is based on Singular Value Decomposition (SVD) which has many applications. In particular an early and major use of SVD is in noise removal and dimension reduction. In latent semantic analysis we extract hidden information about the relationships between objects as they change when we set all, but the most significant, singular values to zero.



Purpose: The purpose of this lab is to get a basic understanding of the singular value decomposition (and LSA) and to learn how these can be applied.

Resources: MATLAB: http://www.mathworks.com
SCILAB: http://www.scilab.org
http://www.bluebit.gr/matrix-calculator/

Directions: Read the tutorial thoroughly and complete the exercises along the way.

Exercises:

Let $M \in \Re^{nxm}$, we decompose M into three matrices using Singular Value Decomposition.

where $U \in \Re^{nxm}$, $S \in \Re^{mxm}$ and $V^T \in \Re^{mxm}$. The matrix S contains the singular values located in the [i,i]1,..,n cells in decreasing order of magnitude and all other cells contain zero. The eigenvectors of MMT make up the columns of U and the eigenvectors of MTM make up the columns of V. The matrices U and V are orthogonal, unitary and span vector spaces of dimension n and m, respectively. The inverses of U and V are their transposes.

$$\underset{U}{\begin{bmatrix} | & | & & | \\ d_1^f & d_2^f & \cdots & d_k^f \\ | & | & & | \end{bmatrix}} \underset{S}{\begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_k \end{bmatrix}} \underset{V^T}{\begin{bmatrix} — & d_1^c & — \\ — & d_2^c & — \\ & \vdots & \\ — & d_k^c & — \end{bmatrix}}$$

The columns of U are the principal directions of the columns of the original dataset and the rows of V^T are the principal directions of the rows of the original dataset. The principal directions are ordered according to the singular values and therefore according to the importance of their contribution to M.

The singular value decomposition is used by setting some singular values to zero, which implies that we approximate the matrix M by a matrix:

$$M_k = U_k S_k V_k^T$$

A fundamental theorem by Eckart and Young states that M_k is the closest rank-k least squares approximation of M. This theorem can be used in two ways. To reduce noise by setting insignificant singular values to zero or by setting the majority of the singular values to zero and keeping only the few influential singular values in a manner similar to principal component analysis.

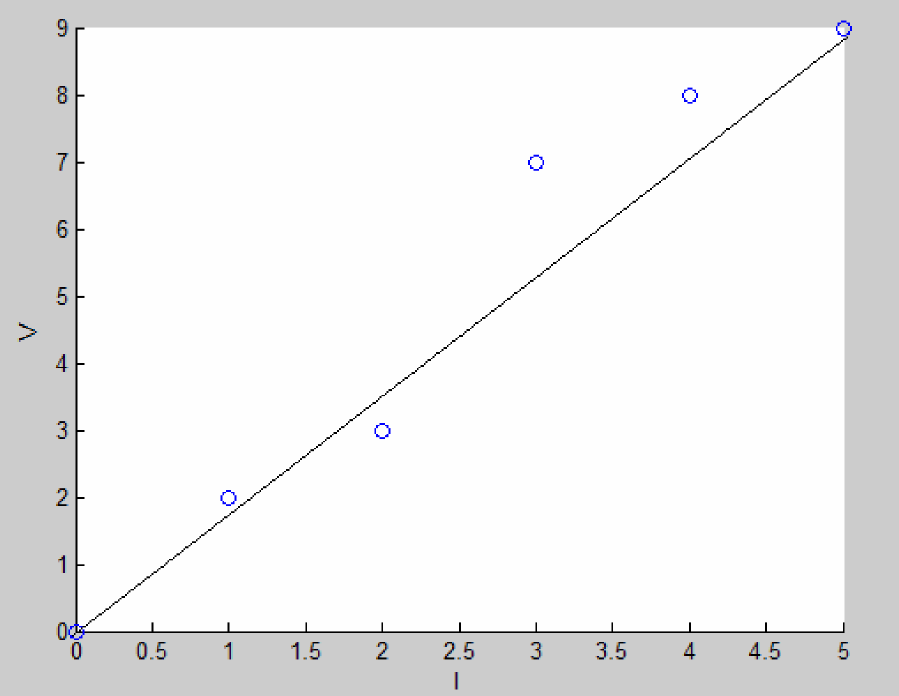
In latent semantic analysis we extract information about the relationships between calls and features as they change when we set all, but the most significant, singular values to zero. The singular values in S provide contribution scores for the principal directions in U and V^T.

We use the terminology "principal direction" for the following reason. In zoomed clusters it was shown that (assuming unit vectors) the principal eigenvector is an "iterated centroid" that is a version of the notion of centroid, where outliers are given a lower weight. Furthermore, in text analysis it is usual to consider that the main information is provided by the direction of the vectors rather than by their length.

Ohm's law Example: V = RI

M =

I	V
0	0
1	2
2	3
3	7
4	8
5	9



We first decompose M into three matrices using the singular value decomposition.

$$M = USV^T$$

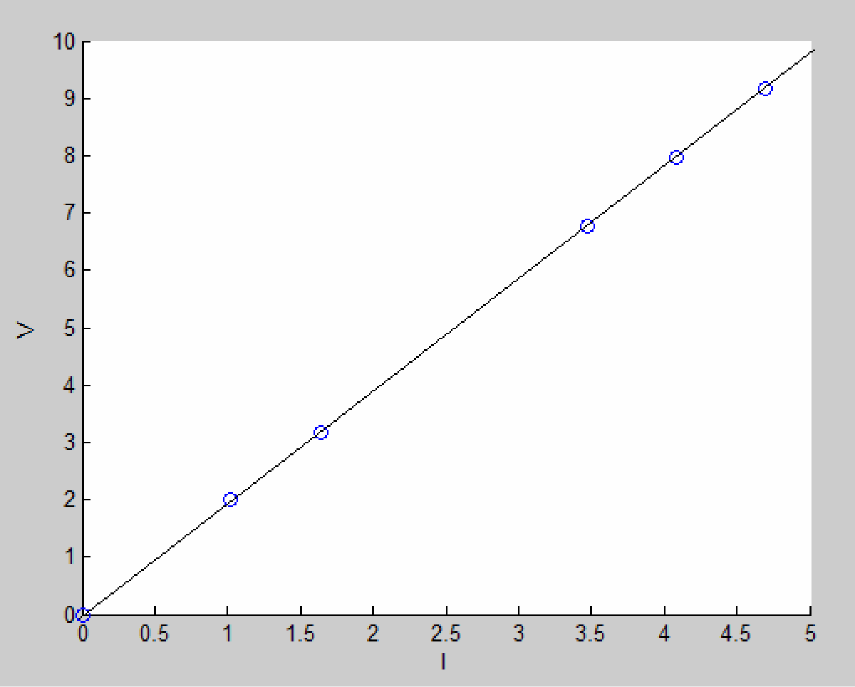
$$\mathbf{U} = \begin{array}{cc} 0 & 0 \\ -0.1383 & -0.0318 \\ -0.2216 & 0.5415 \\ -0.4699 & -0.7004 \\ -0.5531 & -0.1272 \\ -0.6364 & 0.4461 \end{array} \qquad \mathbf{S} = \begin{array}{cc} 16.1688 & 0 \\ 0 & 0.7549 \end{array} \qquad \mathbf{V}^{\mathbf{T}} = \begin{array}{cc} -0.4568 & -0.8896 \\ 0.8896 & -0.4568 \end{array}$$

From looking at S, we can see that the first singular value is much higher than the 2nd singular value. By only considering the first singular value and the corresponding eigenvectors of U and V^T we can find a best fit least squares approximation.

$$M^{'} = U_1 S_1 V_1^T$$

$$\mathbf{U} = \begin{array}{c} 0 \\ -0.1383 \\ -0.2216 \\ -0.4699 \\ -0.5531 \\ -0.6364 \end{array} \qquad \mathbf{S} = 16.1688 \qquad \mathbf{V}^{\mathbf{T}} = \begin{array}{cc} -0.4568 & -0.8896 \end{array}$$

$$\mathbf{M'} = \begin{array}{cc} 0 & 0 \\ 1.0214 & 1.9890 \\ 1.6364 & 3.1867 \\ 3.4704 & 6.7585 \\ 4.0854 & 7.9561 \\ 4.7004 & 9.1538 \end{array}$$



Noise removal

$$
M = \begin{array}{cccccccccc}
0.283 & 0.839 & 0.350 & 0.603 & 0.239 & 0.691 & 0.500 & 0.500 & 0.724 & 0.384 \\
0.701 & 0.781 & 0.383 & 0.745 & 0.603 & 0.623 & 0.771 & 0.766 & 0.653 & 0.636 \\
0.583 & 0.659 & 0.279 & 0.648 & 0.427 & 0.604 & 0.658 & 0.631 & 0.688 & 0.446 \\
0.829 & 0.129 & 0.080 & 0.462 & 0.577 & 0.218 & 0.657 & 0.593 & 0.330 & 0.395 \\
0.742 & 0.527 & 0.232 & 0.645 & 0.520 & 0.535 & 0.731 & 0.688 & 0.645 & 0.476 \\
0.528 & 0.668 & 0.313 & 0.611 & 0.458 & 0.533 & 0.604 & 0.603 & 0.556 & 0.491 \\
0.028 & 1.001 & 0.461 & 0.549 & 0.183 & 0.676 & 0.343 & 0.406 & 0.584 & 0.442 \\
0.577 & 0.533 & 0.223 & 0.571 & 0.395 & 0.528 & 0.610 & 0.579 & 0.622 & 0.386 \\
0.327 & 0.926 & 0.418 & 0.655 & 0.338 & 0.699 & 0.545 & 0.572 & 0.685 & 0.514 \\
0.710 & 0.260 & 0.127 & 0.485 & 0.492 & 0.315 & 0.617 & 0.571 & 0.409 & 0.388 \\
0.283 & 0.839 & 0.350 & 0.603 & 0.239 & 0.691 & 0.500 & 0.500 & 0.724 & 0.384 \\
0.701 & 0.781 & 0.383 & 0.745 & 0.603 & 0.623 & 0.771 & 0.766 & 0.653 & 0.636
\end{array}
$$

$$M = USV^T$$

We decompose M using the singular value decomposition…

$$
\mathbf{S} = \begin{array}{cccccccccc}
5.8001 & & & & & & & & & \\
 & 1.4282 & & & & & & & & \\
 & & 0.2733 & & & & & & & \\
 & & & 0.0195 & & & & & & \\
 & & & & 0.0095 & & & & & \\
 & & & & & 0.0075 & & & & \\
 & & & & & & 0.0064 & & & \\
 & & & & & & & 0.0054 & & \\
 & & & & & & & & 0.0031 & \\
 & & & & & & & & & 0.0006
\end{array}
$$

and set the insignificant singular values in S to zero, since these represent noise.

$$
\mathbf{S} = \begin{array}{cccccccccc}
5.8001 & & & & & & & & & \\
 & 1.4282 & & & & & & & & \\
 & & 0 & & & & & & & \\
 & & & 0 & & & & & & \\
 & & & & 0 & & & & & \\
 & & & & & 0 & & & & \\
 & & & & & & 0 & & & \\
 & & & & & & & 0 & & \\
 & & & & & & & & 0 & \\
 & & & & & & & & & 0
\end{array}
$$

$$
M' = \begin{array}{cccccccccc}
0.272 & 0.862 & 0.384 & 0.598 & 0.286 & 0.655 & 0.486 & 0.509 & 0.644 & 0.446 \\
0.713 & 0.757 & 0.344 & 0.752 & 0.557 & 0.663 & 0.780 & 0.758 & 0.732 & 0.576 \\
0.577 & 0.674 & 0.306 & 0.642 & 0.459 & 0.579 & 0.652 & 0.637 & 0.631 & 0.491 \\
0.834 & 0.117 & 0.064 & 0.463 & 0.553 & 0.235 & 0.660 & 0.593 & 0.367 & 0.371 \\
0.736 & 0.541 & 0.249 & 0.646 & 0.544 & 0.515 & 0.726 & 0.691 & 0.603 & 0.500 \\
0.535 & 0.651 & 0.295 & 0.609 & 0.429 & 0.555 & 0.613 & 0.601 & 0.602 & 0.465 \\
0.036 & 0.986 & 0.436 & 0.552 & 0.149 & 0.699 & 0.351 & 0.401 & 0.641 & 0.403 \\
0.565 & 0.554 & 0.253 & 0.571 & 0.436 & 0.493 & 0.603 & 0.584 & 0.551 & 0.439 \\
0.328 & 0.922 & 0.412 & 0.657 & 0.330 & 0.708 & 0.548 & 0.569 & 0.702 & 0.492 \\
0.711 & 0.262 & 0.126 & 0.483 & 0.493 & 0.315 & 0.616 & 0.568 & 0.416 & 0.380 \\
0.272 & 0.862 & 0.384 & 0.598 & 0.286 & 0.655 & 0.486 & 0.509 & 0.644 & 0.446 \\
0.713 & 0.757 & 0.344 & 0.752 & 0.557 & 0.663 & 0.780 & 0.758 & 0.732 & 0.576
\end{array}
$$

So from M, we can see that M’ is the best fit least squares approximation of M.

$$M = \begin{matrix}
0.283 & 0.839 & 0.350 & 0.603 & 0.239 & 0.691 & 0.500 & 0.500 & 0.724 & 0.384 \\
0.701 & 0.781 & 0.383 & 0.745 & 0.603 & 0.623 & 0.771 & 0.766 & 0.653 & 0.636 \\
0.583 & 0.659 & 0.279 & 0.648 & 0.427 & 0.604 & 0.658 & 0.631 & 0.688 & 0.446 \\
0.829 & 0.129 & 0.080 & 0.462 & 0.577 & 0.218 & 0.657 & 0.593 & 0.330 & 0.395 \\
0.742 & 0.527 & 0.232 & 0.645 & 0.520 & 0.535 & 0.731 & 0.688 & 0.645 & 0.476 \\
0.528 & 0.668 & 0.313 & 0.611 & 0.458 & 0.533 & 0.604 & 0.603 & 0.556 & 0.491 \\
0.028 & 1.001 & 0.461 & 0.549 & 0.183 & 0.676 & 0.343 & 0.406 & 0.584 & 0.442 \\
0.577 & 0.533 & 0.223 & 0.571 & 0.395 & 0.528 & 0.610 & 0.579 & 0.622 & 0.386 \\
0.327 & 0.926 & 0.418 & 0.655 & 0.338 & 0.699 & 0.545 & 0.572 & 0.685 & 0.514 \\
0.710 & 0.260 & 0.127 & 0.485 & 0.492 & 0.315 & 0.617 & 0.571 & 0.409 & 0.388 \\
0.283 & 0.839 & 0.350 & 0.603 & 0.239 & 0.691 & 0.500 & 0.500 & 0.724 & 0.384 \\
0.701 & 0.781 & 0.383 & 0.745 & 0.603 & 0.623 & 0.771 & 0.766 & 0.653 & 0.636
\end{matrix}$$

Now we will move towards latent semantic analysis:

In our collection we have the three documents below. From these documents, we construct a term by document matrix M by only considering the meaningful words in every document.

d1: **Survey** of **ordered trees**
d2: **Graph** of **paths** in **trees**
d3: **Survey** of **paths** in **spanning trees**

$$\begin{array}{l} \\ \text{Survey} \\ \text{Ordered} \\ \text{Trees} \\ \text{Graph} \\ \text{Paths} \\ \text{Spanning} \end{array} \quad M = \begin{array}{ccc} d1 & d2 & d3 \\ \hline 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}$$

Usually a weighting scheme is applied to the term by document matrix such as the log entropy as seen below:

$$M_{i,j} = \log(f_{i,j} + 1) \ \times \ \left(1 - \sum_{j=1}^{m} \left(\frac{f_{i,j}}{g_i} \log\left(\frac{f_{i,j}}{g_i}\right)\right) / \log N \right)$$

Where $F_{i,j}$ is the word frequency i in sequence j, g_i is the total number of times word i occurs in the sequences and N is the number of sequences in the corpus.

However, in this small example we are only concerned with understanding the concepts of LSA, so we will simply use the frequency of the words in the documents with no weighting scheme applied.

As our query, we will use the document q mapped into the same space as our training documents:

q: **spanning trees** of a **graph**

$$q = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

We will now decompose the matrix M using the Singular Value Decomposition. This can be done using online software such as Matlab or the Bluebit Matrix Calculator: http://www.bluebit.gr/matrix-calculator/

$$M = USV^T$$

$$\mathbf{U} = \begin{matrix} -0.461 & 0.500 & 0.191 \\ -0.191 & 0.500 & -0.461 \\ -0.653 & -0.000 & -0.270 \\ -0.191 & -0.500 & -0.461 \\ -0.461 & -0.500 & 0.191 \\ -0.270 & 0.000 & 0.653 \end{matrix} \qquad \mathbf{S} = \begin{matrix} 2.613 & 0 & 0 \\ 0 & 1.414 & 0 \\ 0 & 0 & 1.082 \end{matrix} \qquad \mathbf{V}^{\mathbf{T}} = \begin{matrix} -0.500 & -0.500 & -0.707 \\ 0.707 & -0.707 & 0.000 \\ -0.500 & -0.500 & 0.707 \end{matrix}$$

By keeping the first two columns of U and V and the first two columns and rows of S we have a rank-2 approximation of M.

$$M' = U_k S_k V_k^T$$

$$\mathbf{U} = \begin{matrix} -0.461 & 0.500 \\ -0.191 & 0.500 \\ -0.653 & -0.000 \\ -0.191 & -0.500 \\ -0.461 & -0.500 \\ -0.270 & 0.000 \end{matrix} \qquad \mathbf{S} = \begin{matrix} 2.613 & 0 \\ 0 & 1.414 \end{matrix} \qquad \mathbf{V}^{\mathbf{T}} = \begin{matrix} -0.500 & -0.500 & -0.707 \\ 0.707 & -0.707 & 0.000 \end{matrix}$$

In this new space, we have that the rows of U are the coordinates of the terms and the columns of V^T are the coordinates of the documents. We also have that the columns of U are the principal direction of the documents and the rows of V^T are the principal direction of the terms.

Terms	Term Coordinates		Document Coordinates		
Survey	-0.461	0.500	d1	d2	d3
Ordered	-0.191	0.500	-0.500	-0.500	-0.707
Trees	-0.653	-0.000	0.707	-0.707	0.000
Graph	-0.191	-0.500			
Paths	-0.461	-0.500			
spanning	-0.270	0.000			

Now we find the new query coordinates with the reduced 2-dimensional space using $q = q^T U_k S_k^{-1}$

$$q = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.461 & 0.5 \\ -0.191 & 0.5 \\ -0.653 & 0 \\ -0.191 & -0.5 \\ -0.461 & -0.5 \\ -0.270 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2.613} & \\ & \frac{1}{1.414} \end{bmatrix} = \begin{bmatrix} -0.4268 & -0.3536 \end{bmatrix}$$

We use cosine scoring:

$$sim(q_1, q_2) \;=\; q_1^T q_2 / \sqrt{q_1^T q_1 \times q_2^T q_2}$$

So from V we find the document coordinates, or they can be found also by $d = d^T U_k S_k^{-1}$

$d_1(-0.5,0.707)$
$d_2(-0.5,-0.707)$
$d_3(-0.707,0)$

as demonstrated above, we find the query coordinates by $q = q^T U_k S_k^{-1}$

$q(-0.4268,-0.3536)$

We can now use these coordinates for cosine scoring as follows:

$$sim(q, d_1) \;=\; q^T d_1 / \sqrt{q^T q \;\times d_1^T d_1} \;=\; -0.0763$$

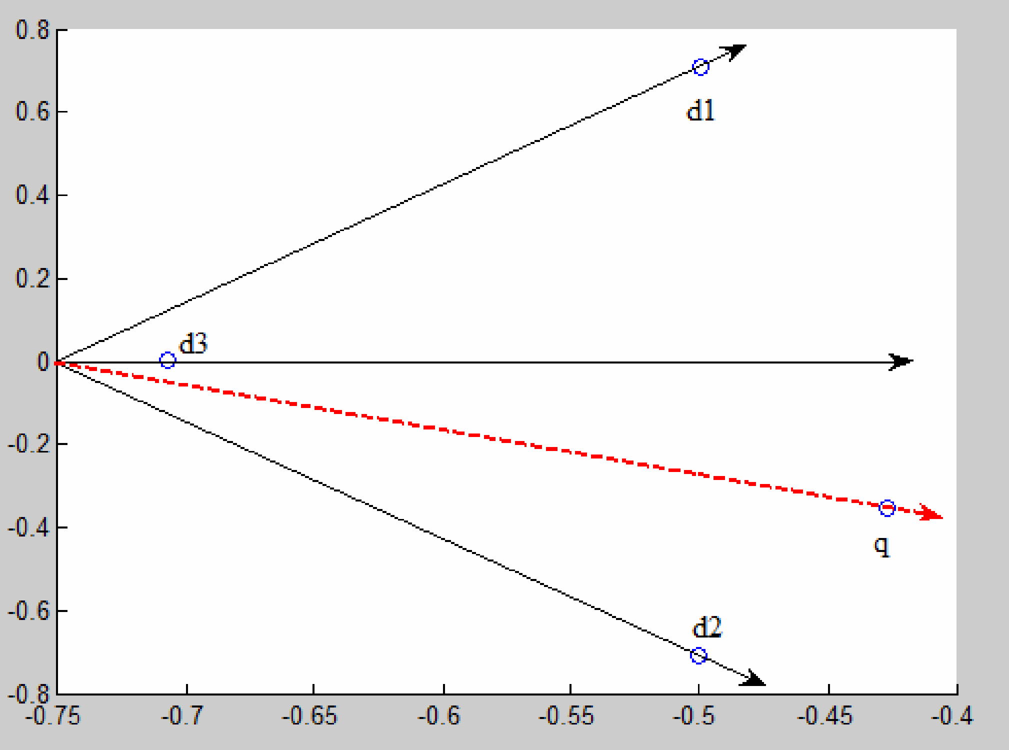
$$sim(q, d_2) \;=\; q^T d_2 / \sqrt{q^T q \;\times d_2^T d_2} \;=\; 0.9655$$

$$sim(q, d_3) \;=\; q^T d_3 / \sqrt{q^T q \;\times d_3^T d_3} \;=\; 0.7701$$

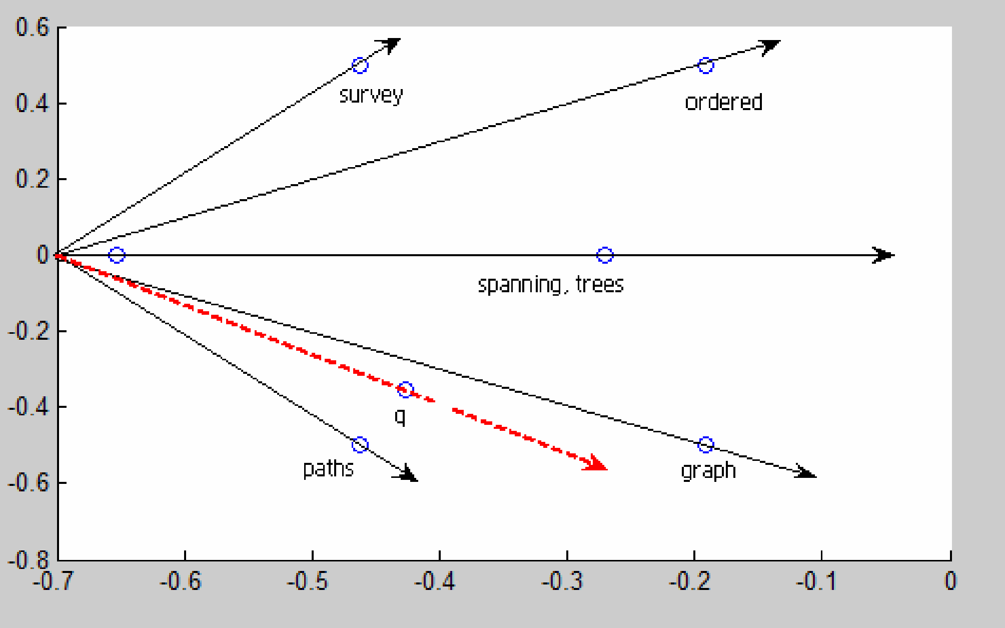
So from this, we can rank the documents by largest to smallest and find that d_2 is most similar to our query.

$d_2 > d_3 > d_1$

But for this example in 2 dimensions we can look at the plotted vectors of the documents:



Similarly, we can look at the plotted vectors of the terms:



$$\mathbf{M'} = \begin{matrix} 1.1036 & 0.1036 & 0.8536 \\ 0.7500 & -0.2500 & 0.3536 \\ 0.8536 & 0.8536 & 1.2071 \\ -0.2500 & 0.7500 & 0.3536 \\ 0.1036 & 1.1036 & 0.8536 \\ 0.3536 & 0.3536 & 0.5000 \end{matrix}$$

$$\mathbf{M} = \begin{matrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix}$$

1. What terms / documents are most similar to our query?

2. What seems to be the reason behind this similarity?

3. Why do you think LSA groups the words: spanning and trees together?

4. What can be learned by looking at the columns of U and V?

5. Now find or make a collection of three documents like we did in this example. After you have the three documents, complete the LSA steps in this tutorial as we did in the example above. Record all the details and write up your findings. Show screenshots of the plotted vectors.

6. Find the paper "Singular value decomposition for genomic-wide expression data processing and model." Briefly describe what this article is about and how it relates to this tutorial.

7. Go on Google Scholar and list a few recent applications of SVD/LSA in bioinformatics.

References:

1. Alter,O., Brown,P.O. and Botstein,D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl Acad. Sci. USA*, **97**, 10101–10106.
2. *Bluebit Matrix Calculator. <http://www.bluebit.gr/matrix-calculator/>. (29 August 2007).*
3. *Google Scholar. <http://scholar.google.com/>. (29 August 2007).*

Appendix

Supplementary Papers

Crick's Hypothesis Revisited: The Existence of a Universal Coding Frame

Jean-Louis Lassez*, Ryan A. Rossi
Computer Science Department, Coastal Carolina University
jlassez@coastal.edu, raross@coastal.edu

Axel E. Bernal
Computer Science Department, University of Pennsylvania
abernal@seas.upenn.edu

Abstract

In 1957 Crick hypothesized that the genetic code was a comma free code. This property would imply the existence of a universal coding frame and make the set of coding sequences a locally testable language. As the link between nucleotides and amino acids became better understood, it appeared clearly that the genetic code was not comma free. Crick then adopted a radically different hypothesis: the "frozen accident". However, the notions of comma free codes and locally testable languages are now playing a role in DNA Computing, while circular codes have been found as subsets of the genetic code. We revisit Crick's 1957 hypothesis in that context. We show that coding sequences from a wide variety of genes from the three domains, eukaryotes, prokaryotes and archaea, have a property of testable by fragments, which is an adaptation of the notion of local testability to DNA sequences. These results support the existence of a universal coding frame, as the frame of a coding sequence can be determined from one of its fragments, independently from the gene or the organism the coding sequence comes from.

1. Introduction

In the early stages of the discovery of the genetic code, Crick hypothesized that the genetic code's structure was endowed with specific information theoretic properties [1].

This would be not only most satisfying intellectually speaking, but would also help explain the extraordinary fact that the genetic code is essentially the same for all organisms. This property would imply that the frame of a fragment of a coding region from any gene or organism can be determined independently from the start or stop codons and independently from the gene or the organism it comes from. We refer to this as the universal frame property of coding regions.

When the mapping from codons to amino acids was better understood and the genetic code appeared to be not comma free, Crick abandoned his early hypothesis to adopt a radically opposite one, the "frozen accident": the structure of the genetic code and its uniqueness are due to an accident in evolution rather than being due to its functionality from an information theoretic point of view. As a consequence, at present, the frame is determined by careful statistical analysis taking into account the specific origin of the organism. In an interesting historical record [2], a notion of comma free codes was credited as being the prettiest wrong idea in all of the 20th century science. However, researchers still pursue, along different lines, the hypothesis that the genetic code's structure is not accidental [3], and in the exciting and growing area of DNA computing, researchers are not dealing with "accidental" codes, they build them according to good information theoretic properties, and comma free codes are again under study [4]. Furthermore, significant results have been found regarding evolutionary aspects of the genetic code [5,6] as well as new techniques to detect the coding regions and the coding frames [7,8,9], when the notion of comma free code is replaced by the more appropriate notion of circular code [10,11].

In this work, we revisit Crick's hypothesis and reformulate it with the hindsight of 50 years of progress in biology, formal languages and coding theories. We argue that the appropriate notion to study the structure of the genetic code and coding sequences is not the notion of comma free or the notion of circular code, but the notion of testable by fragment, which we introduce as a variant of the notion of locally testable [12], as it is more suitable to the analysis of genomic sequences.

To this effect, we show that from a single arbitrarily chosen gene, DKEYP-117E10.6, from the zebra-fish, we can infer by similarity the coding frame of 95% of 2939 genes in the three domains, prokaryotes, eukaryotes and archaea. We then show how one can infer the coding frame of a gene from a fragment of its coding sequence with a certain probability as a function of the length of the segment and independently of the genome or isochore to which the gene belongs. We demonstrate how these results support the existence of a universal coding frame by using a relaxed version of Crick's hypothesis, in which more than one codon is needed to retrieve the coding frame. We also stress the significant role played by the partitioning of the genetic code into three subsets, the T codes [5,6], related to early evolutionary models of the genetic code [13, 14].

In the conclusion we mention research directions that arise from our studies. We believe in particular that the methods we introduced can be used for the analysis of other genomic features, such as pseudo genes, gene complements, and UTR's.

Further information and results can be found at: http://cs.coastal.edu/ucf/

2. Comma Free to Testable by Fragment

Here we present the motivations and intuitions that lead to the reformulation of Crick's hypothesis.

From a formal language/coding theory point of view (but not necessarily from a biological point of view), one can think of the coding sequence of a gene as a sequence of words written in the alphabet {A,C,G,T} having special properties. Each word is translated into a symbol representing an amino acid. There is no special symbol separating the words. What should be the properties of this set of words?

In order to avoid ambiguities in translation we do not want to have a set of words such as: {AC, TGAC, ACTG} because the message ACTGAC could be parsed in different ways: AC/TGAC and ACTG/AC, leading to an ambiguity: which of the corresponding translations is the intended one? As a consequence we need a set of words that forms a code, that is a set of words such that any message can be parsed into code words in a unique way, leading to a unique possible translation. If all words have the same length the problem is solved trivially because there is a unique way to parse messages from such a code. However, a form of ambiguity still exists: Consider the code {ACT, TAG, CTT, AGA}, the sequence ACTTAG can be parsed unambiguously into ACT/TAG, so we know that the words ACT and TAG will be those to translate. But if the sequence is extracted from a longer sequence whose extremities are unknown, say ……ACTTAG….. then we cannot be sure, because the subsequence CTT is also a code word and could be a candidate for translation depending on the frame. What is worse, assuming that the intended frame is the one that corresponds to …ACT/TAG…, an error in transmission that would drop the first letter (A) would cause a frame-shift CTT/AG…. and a non intended translation.

Crick's early hypothesis was that the genetic code is comma free; in that case it ought to have a very strong property: no single trinucleotide in a frame-shift can be translated, because no trinucleotide in a frameshift belongs to the code. As a consequence, the occurrence of a single code word in a coding region defines the frame, regardless of the start or stop codons, the gene or the organism it comes from. Hence, Crick's hypothesis can be reinterpreted as claiming the existence of a universal frame, with the comma free property as means of establishing this hypothesis.

We now know of course, that the notion of comma free is far too drastic, indeed it implies that we can determine the frame given any fragment of length 5 in the coding region; still, that does not necessarily rule out the existence of a universal frame. We first relax the notion of comma free codes. In [11] the definition of parasite sub-messages was introduced. If {ACT, TAG, CTT, AGA} is the code and ACTTAG the intended message, then CTT is a parasite sub-message. A comma free code is a code without parasites. A code with bounded parasitism allows parasite sub-messages of at most length d code words. If the code has bounded parasitism we need to see a sequence of d+1 code words in order to determine the frame. So if our goal is to test the universal frame hypothesis it is reasonable to consider such codes rather than the most restrictive comma free.

We now consider another way of addressing the universal frame hypothesis: local testability [12]. Informally, if we can decide that a sequence belongs to a language L by analyzing independently all its factors of a given length, the language L is called locally testable. It is easy to create examples of locally testable languages; for example, consider the set L of sequences that do not contain the subsequence ATA, testing all sub-words of length 3 in the sequence for equality to ATA allows us to determine if the sequence belongs to L. A great example of "something" not locally testable is provided by Escher, who was followed by a number of (creative) imitators in MAD magazine, with their drawings of "impossible" objects. Look at his famous "endless staircase" (image to be found on the site http://cs.coastal.edu/ucf/), if there is a window that allows us to see only four steps at a time, each view is compatible with a regular staircase. But when you have global view of the whole, you realize it is not a staircase. This conflict between local and global has been used systematically in a number of Escher's other drawings. We know that (finite) codes with bounded parasitism generate sets of messages that are locally testable and conversely [11].

However if we are interested in the universal frame hypothesis, the fact that the codes are comma free, or have bounded parasitism is only of secondary importance if the set of messages is locally testable. Indeed local testability may allow us to find the frame, even if the underlying code does not have the above-mentioned properties. The reason is simple: there could be rules that restrict the generation of parasite sub-sequences, for instance rules that restrict long repeats of AAA, CCC, GGG and TTT. We then still should be able to verify the universal frame hypothesis despite the eventual lack of properties of the underlying genetic code.

More formally let G be a code and G* the set of all messages that it can generate and let L, strict subset of G* a language defined by some grammatical rules. The definition of the frame of words in L could very well come from the rules rather than from the properties of the code G. Therefore, in order to verify the universal frame hypothesis, we can relax Crick's hypothesis from G comma free to G being a code with bounded parasitism, to G* being locally testable, to L being locally testable. All these notions are very closely related in a formal way, described in the next section; however it is by using the most appropriate one that the problem's solution will become apparent. In that respect we will consider two further adaptations of the mathematical formalism to our situation. In coding theory as well as in formal language theory, two words are considered different if they are not syntactically identical. This is too strict for our purpose; we will use the notion of similarity between words rather than identity. Furthermore, the formal definition of a locally testable language is far more restrictive than what its intuitive and informal motivation infers. We will then use a more appropriate variant of this formal definition that is still very much in the spirit of the informal one. For this reason, we will not use the terminology "locally testable" but instead the terminology "testable by fragments". We now can reformulate Crick's hypothesis: What is the length of the shortest fragment of coding sequence, if it exists, that will allow us to determine the frame, independently of the gene or the organism it comes from?

3. Preliminary Definitions and Results

A set of words S is a code if and only if any message, that is any word of S*, can be parsed in a unique way into words of S.

The results we give now can be derived from well known more general theorems [10,11]. However we are in a situation in which they can be established in a simple and intuitive way, when we *only consider codes X whose words have the same length k.*

Let ***m*** be a message from a code **X**, that is a sequence of words from the code **X**. If a subsequence ***p*** of **X**, in a shifted frame, is also made of words of **X**, ***p*** is called a *parasite sub-message* and ***m*** will be referred to as the *intended message.*

A *comma free* code is a code that does not admit any parasite sub-messages. As a consequence the frame is determined by any occurrence of a code word in a message.

Remark 1. The genetic code is not comma free as any sequence of length 3 in a gene is a code word, regardless of the frame in which it occurs.

A code **X** has *bounded parasitism* of degree *d* if there are parasite sub-messages in words of **X*** made of at most *d* words of **X**.

A code has *spread parasitism* if one can find messages with parasite sub-messages of arbitrary length.

As a consequence we have:

Proposition 1. A code **X** has bounded parasitism of degree *d* if and only if the code $\mathbf{X}^{d+1}$ is comma free.

Hence, if the **X** code has bounded parasitism of degree *d*, any occurrence of a sequence of *d*+1 words of **X** determines the frame.

We will relate these notions to the concept of locally testable. The set **X*** of messages from a code **X** is *strictly locally testable* if and only if we can decide if a word w belongs to **X*** in the following way: there exists a number *d* such that the prefix of *w* of length *kd* is in **X***, as well as the suffix of *w* of length *kd*, and all factors of *w* of length *kd* are factors of words of **X***.

In other words we can decide if *w* is a message from **X** by sliding a window of a given length along *w* and independently analyse the properties of each window.

Theorem 1. A Comma Free code **X** generates a set of messages **X*** which is strictly locally testable.

Proof (informal). Let w be a word of **X*** it is straightforward to see that it satisfies the conditions. What we have to show is that if *w* does not belong to **X***, then some condition will not be satisfied. First case, *w*'s length is not a multiple of *k*. Assuming that all the other conditions are met, the suffix of *w* of length 2*k* cannot satisfy the condition because it would imply that a word of **X** appears in a shifted frame, in contradiction with the fact that **X** is comma free. Second case, *w* is of length multiple of *k*. Then one of the *k*-uples in the coding frame does not belong to **X**. This will be found immediately if it is one of the first two. Assume it is the third. Then the sequence made of the second and the third triplets cannot be a factor of words of **X*** because the third triplet does not belong to **X**, and if it was a shifted factor it would imply that **X** is not comma free. Now if the third triplet belongs to **X** we can shift the argument to the next triplet and repeat the argument.

Theorem 2. If **X*** is strictly locally testable then X has bounded parasitism

Proof (informal). If **X** does not have bounded parasitism, then we can have parasite sub-messages of arbitrary length. We can then make a word that does not belong to **X***, but has arbitrarily long prefixes and suffixes that belong to **X***. As a consequence windows of fixed size cannot discriminate between the two competing frames and the word w will be accepted as a word of **X***.

So we have established the links between bounded parasitism, comma free and local testability, we will now briefly mention how circular codes [11] are related. Circular codes have applications in dynamical systems, coding and automata theory, combinatorics [15], and more recently in theoretical biology [5,6,7,8,9], as we will point out in the next section. They are of relevance here because in the finite case they are identical to codes with bounded parasitism, and it is this property that has been used in the applications in biology, not the circularity. The "circularity" aspect of circular codes might be more relevant in DNA computing where one computes with plasmids [16].

4. T-representations and Similarities

The following circular codes (which are in fact codes with bounded parasitism) have been found as subsets of the genetic code:

> **X_0** = {AAC, AAT, ACC, ATC, ATT, CAG, CTC, CTG, GAA, GAC, GAG, GAT, GCC, GGC, GGT, GTA, GTC, GTT, TAC, TTC}
>
> **X_1** = {ACA, ATA, CCA, TCA, TTA, AGC, TCC, TGC, AAG, ACG, AGG, ATG, CCG, GCG, GTG, TAG, TCG, TTG, ACT, TCT}
>
> **X_2** = {CAA, TAA, CAC, CAT, TAT, GCA, CCT, GCT, AGA, CGA, GGA, TGA, CGC, CGG, TGG, AGT, CGT, TGT, CTA, CTT}

These codes have remarkable properties, and have been used to help identify coding regions for prokaryotes and eukaryotes [7]; other circular codes have been found for archaea [8], and yet others are used to find the frame in bacterial coding regions [9]. But it is the codes T_0=X_0U{AAA, TTT}, T_1=X_1U{CCC} and T_2=X_2U{GGG} that we will consider as their union forms the whole genetic code. These codes [5,6] when translated in the two letter genetic alphabet {R,Y} (R = purine, that is A or G, while Y = pyrimidine, that is C or T) allow to retrieve a codon model for primitive protein coding genes [13,14].

The issue is then the analysis of the distribution of these codes in genes. For this purpose, we associate three T-Representations to any coding sequence u:

The first representation, T, is obtained by replacing each codon by 0 if it belongs to T_0, 1 if it belongs to T_1 and 2 if it belongs to T_2. This representation corresponds to the coding frame, while the two others represent the shifted frames. The second representation T^+ is obtained by elimination of the first letter of u and applying the preceding construction. Finally, the third representation T^{++} is obtained by eliminating a second letter from u and again applying the same construction.

We then build the sets C, C^+ and C^{++} of all windows of length k of respectively T, T^+ and T^{++}. The set C represents the coding frame while the two others represent shifted frames of the coding frame. Consider the similar sets of windows, F, F^+ and F^{++} associated to another gene. The question is: does the set F which also represents a coding frame exhibit more similarity to the set C than it does to the sets C^+ or C^{++}?

To answer this question we use a simple similarity test based on the radial basis function, which was shown to perform essentially as well as the SVM for this type of problems [17] and which will allow us to derive more information on the structure of the data.

The similarity between two windows X and Y is defined as:

$$S(X,Y) = e^{-\frac{\|X-Y\|^2}{2\sigma^2}}$$

Where σ represents the "tightness" of the similarity measure [17]. The similarity of a vector X to a set V of vectors is defined as the average of the similarities of X to each vector in V.

Given three sets of previously classified training vectors, C which represents a coding frame, C^+ and C^{++}, which represents the shifts of the coding frame, a vector X will be predicted as being a coding vector if it is more similar to C than it is to C^+ or C^{++}. Else it will be predicted as a non-coding vector. When σ decreases, it creates conditions leading to overfitting as two vectors need to be closer in order to have a non-negligible value for the measure of their similarity. In general automatic classification works poorly in case of overfitting, we will see here an interesting example of its use.

5. Comparing Frames

The full results mentioned in that section, as well as the programs used are to be found on the site (http://cs.coastal.edu/ucf/).

We arbitrarily selected the coding sequence of a well curated gene, DKEYP-117E10.6 a gene from the zebrafish. From the T representations of this coding sequence, we derived the three sets of windows C, C^+ and C^{++}. We tested sets of windows F, F^+ and F^{++} derived from the T representations of a few other coding sequences from the same organism. As a starting point we used the representations of the entire coding sequences, and chose the window size as k = 200.

We initially found some confusion where windows from F^+ were seen to be more similar to windows from C, C^+ or C^{++}, nevertheless it seemed that there was a trend, and in particular none of the windows from F was more similar to windows from C^{++}. In order to analyze this further we decided to remove C^+ from the training set.

We then saw something very striking and consistent over the few examples that we ran (see an example in table 1). First when testing F, the set corresponding to the coding frame of the gene, there is a 100% success, we have no false negatives. Furthermore this success rate is maintained up to very small values of sigma, implying that all windows of F are very close to the windows of C, as the results resist the move towards overfitting. On the other hand the results for F^{+} and F^{++} varied, and decreased as the value of sigma decreased, indicating more widely distributed vectors.

Sigma	F Similarity	F^{+} Similarity	F^{++} Similarity
.4	100%	77.96%	18.79%
.2	100%	77.96%	18.79%
.1	100%	77.96%	18.79%
.01	100%	77.66%	17.01%
.006	100%	73.96%	16.12%
.0058	100%	73.96%	16.12%
.005	100%	72.93%	14.79%
.003	100%	55.33%	7.1%
.002	100%	43.93%	10.06%

Table 1. Percentage of windows from the frames of the fimD gene of yersinia pestis KIM that are similar to windows from the coding frame of the gene DKEYP-117E10.6 from the zebrafish.

This led us to define a first algorithm, that we call the *strict algorithm*,

Strict Algorithm:

We predict that the set F represents the coding frame if and only if

1: for a full range of values of σ all windows of F are more similar to the set C of windows in the coding frame of the training set than they are similar to the windows in the set C^{++} which represents a twice shifted coding frame

2: there exist windows in the twice shifted frame F^{++} that are more similar to the windows in C^{++} than to those in C.

We are simplifying the algorithm by not analyzing the similarity with C^{+}. The justification, besides being empirical, is based on the following argument: As we require that F be most strongly similar to C, if F^{+} is not as similar to C it can be ignored. The case remains where F is also most strongly similar to C. In that case both F and F^{+} are most dissimilar to C^{++}, but if we assigned F^{+} to the coding frame we would have to assign F to C^{++}, but F exhibits 0% similarity to C^{++}.

We then selected coding sequences from 34 prokaryotes, 12 eukaryotes and 13 archaea. From each of these organisms we selected randomly an average of forty coding sequences. This allowed us to see similarities between coding sequences in the same organism as well as similarities between coding sequences from different domains. We also added 100 genes from KEGG and the Weizmann Institute, which are particularly well-studied and curated coding sequences. Finally we took 953 coding sequences from a wide variety of mammalian organisms, and with a wide range of GC content, which were previously used as benchmark test sets for gene-finding by the bioinformatics group at the University of Pennsylvania, these three subsets gave us a total of 2939 testing sequences.

The results were striking: 95% of the T- representations of the coding frames of these 2939 coding sequences are more similar to the T representation of the coding sequence of the gene DKEYP-117E10.6 than they are similar to the T representation of its coding sequence shifted twice. Furthermore the strictness of the algorithm requiring no false negative (100% score in the first column) for a range increasingly small values for σ indicates that all these representations are indeed very similar. One factor is that the number of occurrences of codons from T0 is higher in the coding frame, which is consistent with prior results concerning prokaryotes and eukaryotes [5], but as we will see later, it is not the only factor. The failed predictions were found to be mostly concentrated in a few specific organisms, such as Saccharomyces cerevisiae and Caenorhabditis elegans, rather than being randomly distributed, nevertheless the predictions were correct for the vast majority of the other genes in these organisms. We also experimented with other coding sequences for our training set, such as the human TP53 gene, e-coli metE gene and the Pyrococcus abyssi PAB0437 gene and obtained essentially similar results. These training genes are from the three different families and have substantially different DNA sequences.

Now we address the problem of the relevance to comma free codes, codes with bounded parasitism, circular codes, and the notion of testable by fragment. There are a number of striking instances where we find that not only all windows from the coding frame of the tested gene are similar to the windows of the coding frame of the training gene, but none other are. This property would be consistent with the existence of a comma free code made of words of length at most 600 nucleotides, as we have windows of length 200 in the T representations. Or equivalently this would be consistent with the existence of a code made of shorter words, not comma free but having the property of bounded parasitism. This property would limit the possibility of alternative splicing. We also find examples where both F and F^{+} show extreme similarity with the windows of the coding frame of the training gene. This is consistent with the eventual possibility of alternative splicing, and corresponds to the notion of spread parasitism: two valid translations are possible. These are important problems that will require our attention, but at present they are beyond the scope of our study, as they leads us to look for special methods to determine the frame related to subfamilies, while we are concerned here with universality.

But in all cases our results support the argument that in the set of coding sequences and shifted coding sequences, the language of coding sequences is testable by fragments. This is because we analyze all windows, and can make decisions solely from this analysis.

We can now address Crick's revised hypothesis: what is the length of the shortest fragment of a coding region that will allow us to predict the frame, independently of the gene or the organism it comes from?

We will have to perform a double fragmentation: first generate a random fragment from a coding sequence, and then fragment again by creating windows as we did previously. But we will change the algorithm, indeed the selection of small fragments implies smaller windows, and this will violate the non false negative requirement: small windows from the coding frame of the tested fragment might be similar to small windows from the twice shifted coding frame of the training set. Furthermore the robustness to overfitting that was displayed previously might not occur as systematically: all scores may vary with decreasing values of σ . Nevertheless it may still be possible to correctly predict the frame, but with less accuracy. So we will use a relaxed form of the algorithm:

Relaxed Algorithm:

We will predict that F is the set of windows extracted from the coding frame if and only if the following conditions are met:

1: $$F_S{}^{+} - F_S \le F_S - 50$$

2: $$F_S{}^{++} < F_S > 50$$

Where F_S, F^+_S, and F^{++}_S are the average scores of respectively F, F^+ and F^{++} for a range of sigma values.

Here instead of requiring that all windows of F be similar to those of C we only require that at least 50% be similar to those of C. Then we require that the windows of F be more similar to those of C than the windows of the twice shifted frame F^{++}. Finally we use a heuristic which is a relaxed version of the preceding one. It is also justified pragmatically, even if its supporting argument is somewhat weaker. The larger F_S is, the less likely F is to be associated with the twice shifted coding frame even if F^+_S is larger than F_S.

Once the size of the fragments to test is chosen, we randomly generate a fragment of that size for each of the 2939 sequences. The relaxed algorithm allows us to predict the correct frame in 75% of the cases, for a fragment length of ten trinucleotides and a window size of two trinucleotides. The relaxed algorithm also allows us to predict the correct frame in 90% of the cases, for a fragment length of sixty trinucleotides and a window of twenty-five trinucleotides. Due to the randomness of the selection, minor variations in the success rate occur when repeating the process. For these fragments that are substantially smaller than the whole coding sequence, the distribution of the codons from T0 does not necessarily favor as strongly the coding frame. Now even for small window sizes we still see, not as drastically as with windows of size 200, the phenomenon of robustness with respect to overfitting, indicating that windows in the coding frames of most of the genes considered have a very tight relationship.

6. Conclusion

Provided that we replace the notion of comma-free by the related notion of testable by fragment, Crick's 1957 hypothesis seems vindicated: our results support the existence of a universal frame based on a simple mathematical model. Now it is very tempting to try our method on non coding parts of genomes. But one should realize that when we work within the coding region, we know that there exists a coding frame. Outside of the coding region, we will of course find one frame that will be more similar to a coding frame than the two other shifted frames. So one has to adapt our method to a far more complex situation, and this will be a major undertaking. We can nevertheless see indications that it can be useful. For instance preliminary results show that it is sensitive to (obviously) pseudo genes and gene complements, but also seems sensitive to some UTR's.

References

1. F. H. C. Crick, J. S. Griffith, L. E. Orgel, "Codes Without Commas", *Proc. Natl. Acad. Sci. U.S.A.* 43, 1957, 416-421.
2. H. Brian, "The invention of the genetic code", *American Scientist* 86, 1998, 8-14.
3. R. D. Knight, S. J. Freeland, L. F. Landweber, "Selection, History and Chemistry: The Three Faces of the Genetic Code", *Trends Biochem. Sci.* 24, 1999, 241-247.
4. M. Arita, *Aspects of Molecular Computing,* 2950 Springer Berlin, 2004, 23-35.
5. D. G. Arquès, C. J. Michel, "A Complementary Circular Code in the Protein Coding Genes", *J. Theor. Biol.* 182, 1996, 45-58.

6. D. G. Arquès, C. J. Michel, “A Code in the Protein Coding Genes”, *BioSystems* 44, 1997, 107-134.
7. D. G. Arquès, J. Lacan, C. J. Michel, “Identification of protein coding genes in genomes with statistical functions based on the circular code”, *BioSystems* 66, 2002, 73-92.
8. G. Frey, C. J. Michel, “Circular Codes in Archaeal Genomes”, *J. Theor. Biol.* 223, 2003, 413-431.
9. G. Frey, C. J. Michel, “Identification of circular codes in bacterial genomes and their use in a factorization method for retrieving the reading frames of genes”, *Comp. Biol. & Chem.* 30, 2006, 87-101.
10. J-L. Lassez, “On the Structure of Systematic Prefix Codes”, *Int. J. Comp. Math.* 3, 1972, 177-188.
11. J-L. Lassez, “Circular Codes and Synchronization”, *Int. J. Comp. & Infor. Sci.* 5, 1976, 201-208.
12. T. Head, “Splicing Representations of Strictly Locally Testable Languages”, *Discrete Appl. Math.* 87, 1998, 139-147.
13. F. H. C. Crick, S. Brenner, A. Klug, G. Pieczenik, “A Speculation on the Origin of Protein Synthesis”, *Origins of Life* 7, 1976, 389-397.
14. M. Eigen, P. Schuster, *Naturwissenschaften* 65, Springer Berlin, 1978, 341-369.
15. Google scholar [circular codes]
16. L. Kari, M. Daley, G. Gloor, R. Siromoney, L. F. Landweber, *Foundations of Software Technology and Theoretical Computer Science* 1738, Springer Berlin, 1999, 269-282.
17. A. E. Bernal, T. Karadeniz, K. Hospevian, J-L. Lassez, *Advances in Intelligent Data Analysis V* 2810, Springer Berlin, 2003, 187-19.

Automated Discovery of Translation Initiation Sites and Promoter Sequences in Bacterial Genomes

Axel Bernal[1], Karen Hovsepian[2], JianHua Yang[2], Jean-Louis Lassez[1*]
1: Integrated Genomics
2: New Mexico Institute of Technology

ABSTRACT

Support Vector Machines have recently been used to efficiently detect Translation Initiation Sites. Here we use Support Vector Machines to identify potential regulatory patterns (both activators and inhibitors) around predicted Translation Initiation Sites. We analyze how the prediction of these sites and the prediction of promoter sequences can be enhanced, by using various heuristics, Singular Value Decomposition and Latent semantic Analysis.

We tested our system on known Translation Initiation Sites for the E. coli genes, and generated a list of potential regulatory patterns. These patterns were matched and analyzed against known promoter sub-sequences from RegulonDB. We conclude our paper with a brief mention of possible applications of these techniques to related problems, such as finding signatures in protein families.

1. INTRODUCTION

Once a genome has been sequenced and assembled, the next major task is to identify its coding regions (i.e. genes). In order to accomplish this task, different methods based on codon usage statistics, mutual information or Markov models have been developed. All of them in one way or another assume that long and short correlations between nucleotides differ from coding to non-coding regions. Whereas the end of a coding region is unambiguously defined, it is not the case for it's beginning, and for a given Open Reading Frame (potential gene) there are several candidate Translation Initiation Sites (TIS). TIS recognition is a more difficult task and has recently been attacked by Machine Learning methods, first Neural Nets (Pedersen and Nielsen, 1997) and then Support Vector Machines (Zien et al., 2000) with specially engineered kernels, which gave the best results.

The aforementioned techniques do not address the related problem of finding regulatory patterns around Translation Initiation Sites. Such patterns represent potential promoters, activators and inhibitors, which control gene expression. Because of this, discovering such patterns is of great biological significance. In this paper we represent translation initiation sites by sets of patterns. We are then in a framework of processing documents, and we know that Support Vector Machines are particularly efficient for such tasks, as shown by Thorsten (Thorsten, 1998), Dumais (Dumais et al., 1998), Vapnik (Vapnik, 1995), even with the simpler linear kernels. We can also apply other successful techniques used in search engines to classify documents and address problems of synonymy, polysemy and more prosaically noise removal, with Latent Semantic Analysis and Singular Value Decomposition (Berry et al., 1995; Deerwester et al. 1990; Dumais et al. 1988).

This paper is organized as follows: the first part reviews briefly the mathematical and algorithmic principles that form the foundation of Support Vector Machines. There are a number of SVM programs available on the net (see http://www.kernel-machines.org for a list). For this paper we used the latest version of "SVM Light" by Joachim Thorsten .
The second part explains how we obtained our training and testing sets and the pattern extraction methods used. In the third part, results of various experiments under different conditions are shown. Later, we analyze the effect of different parameter values and heuristics over the classification results. In particular we show that the pattern based approach and the second best heuristic enhance substantially the efficiency of SVM's.
The fourth part focuses on the discovery of potential regulatory patterns. We show how we can classify patterns according to the importance they play in the classification, and analyze the relation between these patterns and known promoter regions from RegulonDB. As opposed to the case of the Reuters database described by Thorsten (Thorsten, 1998), we show that in our situation only comparatively few patterns are needed to obtain an efficient classification.

In conclusion we discuss how our results can be improved, particularly with the use of Singular Value Decomposition and Latent Semantic Analysis, Finally, and mention other areas where the techniques described here could be applied.

2. SUPPORT VECTOR MACHINES

With a wide range of applications: face recognition, isolated handwritten digit recognition, speaker identification, charmed quark detection, face detection, see Isabelle Guyon's site (Guyon) and most recently classification of micro-array gene expression data (Brown et al., 1999), support vector machines have become a major tool in Machine Learning.

We will give here an outline of the design of support vector machines and highlight the key elements that make them outperform competitive approaches for so many applications. The reader is referred to Vapnik's books (Vapnik, 1995; Vapnik, 1998) for a study of their deep underlying theory, to Burges' tutorial (Burges, 1998) for an introduction (in fact quite substantial), and to Cristianini and Shawe-Taylor's book (Cristianini and Shawe-Taylor, 2000) for a thorough treatment. Our aim in this outline is to introduce the unfamiliar reader to the basic notions and make this paper a little bit more self-contained. It can be safely skipped otherwise.

The goal of SVM is to compute a function, which will serve to predict whether an arbitrary object belongs to a given family or not. Thus we would like to find a function F, such that the sign of F(x) tells us when x belongs to the family (positive), and when x does not belong (negative). The magnitude of the absolute value of F(x) is a measure of the strength of the prediction. This function is called the decision function.

In a first phase of the SVM, called the learning phase, the decision function is inferred from a set of objects. For these objects the classification is known a-priori. The objects of the family of interest are called the positive objects and the objects from outside the family, the negative objects.

In a second phase, called the testing phase, the decision function is applied to arbitrary objects in order to determine, or more accurately predict, whether they belong to the family under study, or not.

LINEAR CASE

The objects are represented by vectors in R^n , where each coefficient represents a feature of the object: weight, size, etc…

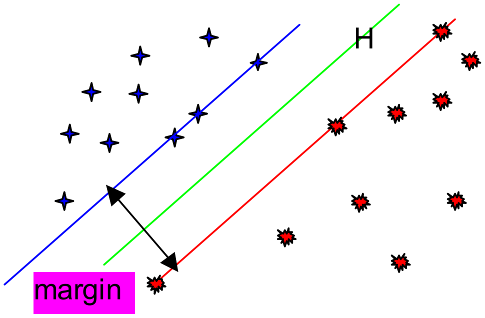


Figure 1

The positive examples form a cloud of points, say the blue cloud, while the negative examples form another cloud of points, say, the red cloud. The aim is to find a hyper-plane H separating, when possible, the two clouds of points in some optimal way. We will address later the case when the two clouds are not separable.

DEFINITION OF MARGIN AND MAXIMAL MARGIN

Let H be a separating hyper-plane, H_b a separating hyper-plane parallel to H and passing through the blue points closest to H, H_r a separating hyper-plane parallel to H and passing through the red points closest to H.

The margin, γ, is the distance between the two parallel separating hyper-planes Hb and Hr. In the above figure, the margin is the distance between the red line and the blue line. Vapnik's theory of risk minimization shows that hyper-planes for which γ is maximum have better generalization potential than others, and so the problem of linear SVM is to find a separating hyper-plane with maximum margin.

There are many ways to represent mathematically such an optimization problem. We have here two concerns. The first is to find a formulation that can be handled by standard optimization techniques (Quadratic programming in our case). The second concern is of major significance from an application point of view: it is indeed possible to find a formulation that will allow us to construct non linear separating surfaces while remaining in the previous computational framework of linear separation.

QUADRATIC PROGRAMMING

A constrained optimization problem consists of two parts: a function to optimize, and a set of constraints to be satisfied by the variables. Constraint satisfaction is typically a hard combinatorial problem; while for an appropriate choice of function, optimization is a comparatively easier analytical problem. Hence, we choose the design of formulations where the constraints are simple linear constraints and we use duality to move expressions from the set of constraints to the function we seek to optimize.

In the maximum margin case, we want to maximize a distance.

In order to express distances of points to a hyper-plane **W*X + b = 0**, we request that $\|\mathbf{W}\|^2 = \mathbf{1}$. Equivalently we can divide the expression by $\|\mathbf{W}\|^2$.

However either formulation gives us a non-linear constraint, which does not lead to efficient computation. We will therefore choose a formulation of the problem, which moves the non-linear constraint into the function to optimize.

Let **W* X + b = 0** be the equation of a separating hyper-plane, situated halfway between the two sets, so that for some **t > 0** we have the blue points $\mathbf{X_b}$ on one side, the red points $\mathbf{X_r}$ on the other:

$$\frac{W * X_r + b}{\|W\|^2} \geq \frac{t}{\|W\|^2}$$

and

$$\frac{W * X_b + b}{\|W\|^2} \leq \frac{t}{\|W\|^2}$$

And there exist blue and red points for which the inequalities are replaced by equalities.

Consequently we have the margin:

$$\gamma = \frac{2t}{\|W\|^2}$$

Assume the maximum margin is reached for $\mathbf{W = W_0, b = b_0, t=t_0}$. Dividing $\mathbf{W_0}$ and $\mathbf{b_0}$ by $\mathbf{t_0}$ shows that the maximal margin is reached for hyper-planes such that $\gamma = \frac{2}{\|W\|^2}$

Without loss of generality we may therefore assume that **t = 1**. And the problem of maximizing γ is replaced by the problem of minimizing the norm of **W**,

$$\frac{1}{2}*\langle W,W\rangle \qquad (1)$$

under the linear constraints:

$$Y_i*(W*X_i+b)\geq 1 \quad \text{for any i} \qquad (2)$$

where $\mathbf{X_{i,}}$ is a data point and $\mathbf{Y_i}$ is the label of the data point, equal to **1** or **–1** depending on whether the point is a positive or negative example.

We now have a typical quadratic programming problem and we will change this formulation with non-linear separability in mind.

NON LINEAR SEPARATION

It can be shown that if you have fewer points than the number of dimensions, then any two sets are separable. It is therefore tempting, when the two sets are not linearly separable, to map the problem into a higher dimension where it will become separable. There is however a price to pay, as quadratic programming problems are quite sensitive to high dimensions. Support Vector Machines handle this problem in a brilliant way, by simulating in the original space a computation in an arbitrarily higher (even infinite) dimensional space.

Consider the following diagram of a mapping Φ from the original space to a higher dimensional space.

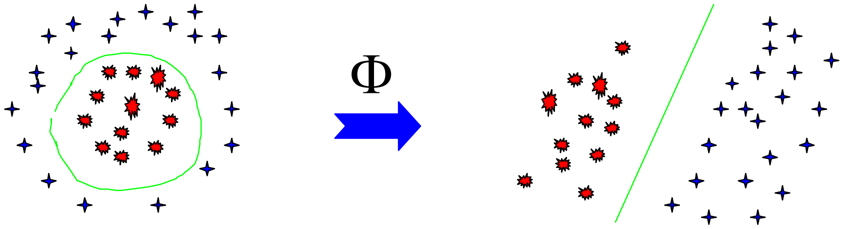


Figure 2

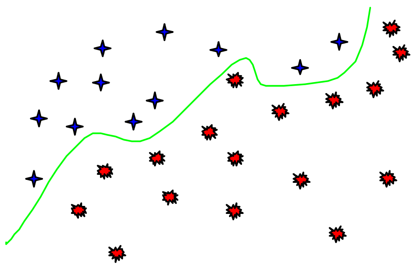


Figure 3

Φ linearizes the problem in higher dimensions, and the effect on the initial data is a non-linear separation.

For this to be possible two conditions have to be met:

1: find a formulation such that the data appears only as vector dot products

2: find an appropriate function K, called a Kernel function, such that

$$\langle \Phi(V), \Phi(W) \rangle = K(V, W)$$

In such a case there is no need to represent the vectors in high dimension as the computation is performed by K in the original space. This might superficially appear as a contrived trick, so the reader is referred to Vapnik's books (Vapnik, 1995) in order to realize that there is in fact a very deep theory behind the design of kernel functions.

WOLFE'S DUAL

The preceding formulation can be transformed by duality, which has the advantage of simplifying the set of constraints, but more importantly, Wolfe's dual gives us a formulation where the data appears only as vector dot products. As a consequence we can handle non-linear separation.

Minimizing $\frac{1}{2} * \|W\|^2$ under the constraint of (2) is equivalent to maximizing the dual Lagrangian obtained by computing variables from the stationarity conditions and replacing them by the values so obtained in the primal Lagrangian. Details can be found in (Cristianini and Shawe-Taylor, 2000).

$$Y_i * (W * X_i + b) \geq 1 \quad \text{for any i} \qquad (2)$$

The primal Lagrangian is:

$$L = \frac{1}{2}\|W\|^2 - \sum_{i=1} \left[\alpha_i * (Y_i(W * X_i + b) - 1\right] \qquad (3)$$

Where α_i is a Lagrange multiplier and Y_i is the label of the corresponding data point under the constraint:

$$\alpha_i > 0 \quad \text{for all i.} \qquad (4)$$

The stationarity conditions are:

$$\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = \sum_{i=1} \alpha_{\mathrm{i}} * Y_i = 0 \qquad (5)$$

$$\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = \mathrm{W} - \sum_{i=1} \alpha_{\mathrm{i}} * Y_i * X_i = 0 \qquad (6)$$

Substituting the value of W from (6) in the primal Lagrangian (3) gives us the Wolfe Dual Lagrangian:

$$W(\alpha) = \sum_{i=1} \alpha_i - \frac{1}{2} \sum_{i=1} \left[\alpha_i * \alpha_j Y_i * Y_j * \left\langle X_i, X_j \right\rangle \right] \qquad (7)$$

This must be maximized, subject to the constraints (4) and (5). It is a standard quadratic programming formulation different from the original in that the data appears only as inner products.

It is then straightforward to implement non-linearity by simply replacing the vector products by kernel functions.

SUPPORT VECTORS

It is clear geometrically that the maximum margin is defined by only a subset of points called support vectors. Indeed from (7) we know that

$$W = \sum_{i=1} Y_i * \alpha_i * X_i \, ,$$

and that the data points $\mathbf{X}_{\mathbf{i}}$, whose coefficients α_i equal **0,** are irrelevant to the definition of the separating surface.

OVERFITTING AND SOFT MARGIN TRADE OFF

Figure 5 shows an example where, by choosing a kernel of sufficient degree we can find a surface complex enough to separate the two clouds of points. When there is noise we can make this surface extremely complex in order to fit the data. This phenomenon is called over-fitting, as maybe some of these noisy points should be in fact ignored, leading to a simpler surface of separation. In figure 5 we have a trade off: a simple linear separation rather than a complex one at the cost of a training error, which could in fact be noise or eventually erroneous data.

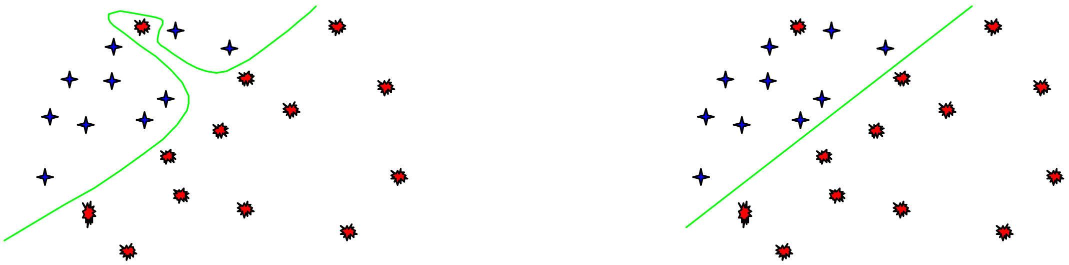


Figure 4

So there exists a trade-off between the degree of the kernel function and the extent to which training errors are allowed. This has a very important consequence, as the algorithm we described will not work when the sets are not separable. Therefore to have the algorithm work we must increase the non-linearity, and therefore the complexity of the surface, and therefore the risk of grossly over-fitting.

The problem is solved by relaxing the conditions in such a way that a certain degree of misclassification is allowed, leading to simpler solutions at the cost of some erroneous, or potentially erroneous predictions.

We refer the reader to (Cristianini and Shawe-Taylor, 2000) for the description of the techniques involved, we just remark that this extension to non-separable cases follows the same transformations as in the separable case, that lead to a quadratic programming formulation suitable for the use of kernel functions. The user has to define the value of a parameter that controls the extent of misclassification allowed. This value is heavily dependent on the data at hand.

3. METHODS

SYSTEM FRAMEWORK

We use SVMs to classify TIS. For this, an original representation of the nucleotides around TIS (window) based on patterns is used. Previous representations of this information include the use of sparse coding (Pedersen and Nielsen, 1997) and specially engineered kernels based on the statistical distribution of the nucleotides (Zien et al., 2000).

Our representation requires a previous extraction of patterns around potential TIS by using techniques described in (Brazma et al., 1998). There are two advantages of this approach, the first one is merely computational, as we map the problem into one of document processing for which SVMs are known to be very efficient (Drucker et al.,1999; Guyon; Thorsten, 1998). The second one however is of more importance for biologists, as some of these patterns will play a more significant role in the classification, and therefore are more likely to be associated to promoters. As a result, in the process of finding true TIS we are also determining potential regulatory sites.

GENE CONSTRAINTS

It is important to notice that after the SVM classification has been finished, it is necessary to guarantee that each gene has one and only one TIS assigned to it. Thus, as in (Zien et al., 2000) we use the ranking of the data points, provided by SVM, and we choose the TIS with the highest value as the predicted one.

4. EXPERIMENTAL RESULTS

DATA SETS

Our data comes from carefully curated genes of the E. Coli genome, filtered from entries in GeneBank. We blasted the sequences against the GeneBank database and only took sequences that showed strong and aligned similarities in the 5’ upstream region to other genes in other organisms. In this way we tried to avoid using earlier or late calls for the start codons, since this situation could have been easily spotted when analyzing the alignments individually. The resulting set consists of 149 sequences of which 101 were used for training and 48 for testing. From this set of genes, around 726 windows and 816 patterns were derived. This careful selection certainly accounts for the quality of our results, since both training and testing data are carefully curated. This allows us to evaluate more confidently the proposed techniques.

PARAMETERS

When running an SVM experiment we are faced with a substantial number of decisions regarding the data representation: Which window size should we take? Should we have a binary representation or not? Should we normalize or not? What Kernel should we choose? What should be the value of the ratio over-fitting/misclassification (C)? In practice we are told that these decisions heavily depend on the specific experimental data, and that we should try various combinations before we settle on a particular one.

As there are infinitely many possible kernels and the C ratio can take infinitely many values, it becomes clearly impossible to consider all combinations and authors usually report only the best results they obtained.

We will report here a number of typical experiments from which we draw useful conclusions regarding the effect of changing the value of the parameters over the quality of the classification by SVMs.

As for the size of the window around any potential TIS, we tried (results not shown) several sizes and positions relative to the TIS. The best results were obtained with windows of size 30, upstream of the TIS.

Table 1 contains typical results obtained from experiments, in which the various combinations of the parameters discussed above were tried. Specifically the experiments were run with binary/non-binary data, normalized/non-normalized data and various values of C and kernel degrees of 1, 2 and 3. The column, which is labeled BF gives us the number of misclassifications.

So for instance, the first row tells us that for non-normalized, non-binary data, with a coefficient C set at 0.002 and a linear kernel, 5 TIS have been misclassified out of 44.

Norm	Bin	C	Degree	BF
		0.002	1	5/44
Y		0.002	1	5/44
	Y	0.002	1	5/44
Y	Y	0.002	1	5/44
		0.004	1	5/44
Y		0.004	1	6/44
	Y	0.004	1	5/44
Y	Y	0.004	1	5/44
		0.008	1	5/44
Y		0.008	1	4/44
	Y	0.008	1	2/44
Y	Y	0.008	1	4/44
		1.000	1	5/44
Y		1.000	1	5/44
	Y	1.000	1	9/44
Y	Y	1.000	1	5/44
		0.002	2	5/44
Y		0.002	2	5/44
	Y	0.002	2	5/44
Y	Y	0.002	2	5/44
		0.004	2	6/44
Y		0.004	2	5/44
	Y	0.004	2	5/44
Y	Y	0.004	2	5/44
		0.008	2	4/44
Y		0.008	2	5/44
	Y	0.008	2	2/44
Y	Y	0.008	2	4/44
		1.000	2	5/44
Y		1.000	2	5/44
	Y	1.000	2	9/44
Y	Y	1.000	2	5/44
		0.002	3	5/44
Y		0.002	3	5/44
	Y	0.002	3	5/44
Y	Y	0.002	3	5/44
		0.004	3	6/44
Y		0.004	3	5/44
	Y	0.004	3	5/44
Y	Y	0.004	3	5/44
		0.008	3	4/44
Y		0.008	3	5/44
	Y	0.008	3	2/44
Y	Y	0.008	3	4/44
		1.000	3	5/44
Y		1.000	3	5/44
	Y	1.000	3	9/44
Y	Y	1.000	3	5/44

Table 1. Experimental results for different SVM parameters

From the above table we see that the Best Fit (BF) rates of most misclassifications are around 11%, with an occasional high of 20% and occasional low of 4.5%. It is not clear what combination of parameters should be used to find an optimal solution.

Tables 2 and 3, extracted from Table 1 as typical cases illustrate that binary weights and normalization do not seem to have substantial effect on the performance of SVM. For the experiments in table 4, binary and non-normalized data was used. This table shows that, as previously reported, C has a major effect on the performance of SVM.

	Non Normalized C=0.002	Normalized C=0.002	Non-Normalized C=1.000	Normalized C=1.000
Best Fit	5/44	5/44	5/44	5/44

Table 2. Effects of normalization over the SVM predictions

	Non-Binary C=0.002	Binary C=0.002	Non-Binary C=0.004	Binary C=0.004
Best Fit	5/44	5/44	6/44	5/44

Table 3. Effects of data representation over the SVM predictions

C	0.002	0.004	0.008	1.000
Best Fit	5/44	5/44	2/44	9/44

Table 4. Effect of the ratio over-fitting / misclassification over the SVM predictions

SECOND BEST FIT (SBF)

A question arises, whether our misclassifications are indeed errors in prediction. In fact we see that in most of the cases, our prediction is very close to being correct. Indeed, for a broad range of parameters values, when our first choice is not correct, the second best will be. This makes the system far more practical than we initially thought as in case of misclassification the user is given a second choice that is most probably correct. Furthermore, the first and second choices that we predict are often so close that both could be valid. This situation usually happens as the result of having two TIS that are very close to each other, in which case it is preferable to report both starts for further manual inspection. Table 5 shows the effect of considering the SBF as part of our set of predictions.

Norm	Bin	C	Degree	BF	SBF
		0.002	1	5/44	1/44
Y		0.002	1	5/44	1/44
	Y	0.002	1	5/44	1/44
Y	Y	0.002	1	5/44	1/44
		0.004	1	5/44	1/44
Y		0.004	1	6/44	1/44
	Y	0.004	1	5/44	1/44
Y	Y	0.004	1	5/44	1/44
		0.008	1	5/44	1/44
Y		0.008	1	4/44	0/44
	Y	0.008	1	2/44	0/44
Y	Y	0.008	1	4/44	1/44
		1.000	1	5/44	0/44
Y		1.000	1	5/44	1/44
	Y	1.000	1	9/44	1/44
Y	Y	1.000	1	5/44	1/44
		0.002	2	5/44	1/44
Y		0.002	2	5/44	1/44
	Y	0.002	2	5/44	1/44
Y	Y	0.002	2	5/44	1/44
		0.004	2	6/44	1/44
Y		0.004	2	5/44	1/44
	Y	0.004	2	5/44	1/44
Y	Y	0.004	2	5/44	1/44
		0.008	2	4/44	0/44
Y		0.008	2	5/44	0/44
	Y	0.008	2	2/44	0/44
Y	Y	0.008	2	4/44	1/44
		1.000	2	5/44	1/44
Y		1.000	2	5/44	1/44
	Y	1.000	2	9/44	1/44
Y	Y	1.000	2	5/44	1/44
		0.002	3	5/44	1/44
Y		0.002	3	5/44	1/44
	Y	0.002	3	5/44	1/44
Y	Y	0.002	3	5/44	1/44
		0.004	3	6/44	1/44
Y		0.004	3	5/44	1/44
	Y	0.004	3	5/44	1/44
Y	Y	0.004	3	5/44	1/44
		0.008	3	4/44	0/44
Y		0.008	3	5/44	1/44
	Y	0.008	3	2/44	1/44
Y	Y	0.008	3	4/44	1/44
		1.000	3	5/44	1/44
Y		1.000	3	5/44	1/44
	Y	1.000	3	9/44	1/44
Y	Y	1.000	3	5/44	1/44

Table 5. Effect of SBF Algorithm over the SVM predictions

5. POTENTIAL PROMOTER SEQUENCES

In his study of documents of the Reuters database, Thorsten (Thorsten,1998) makes the point that all words, even those with little information content, play a role in the classification by SVM. For our particular application we will show that a very substantial number of patterns can be discarded without affecting the classification. The biological explanation for this is that only certain patterns are related to activation or inhibition sites (promoter/regulatory sites), while the rest could be thought as 'background noise'. Using our approach, we are able to find these important patterns while predicting TIS.

To accomplish this, we simply use the weight function provided by the SVM classification and rank the patterns according to their weight. Patterns with weight close to zero will have little effect on the classification. The presence of patterns with a high positive weight in a site will increase the ranking of the site, while patterns with a low negative weight will decrease the ranking of the site.

A first experiment to confirm the usefulness of this classification of patterns is run using the parameter settings which gave us the best results: Non binary, Non normalized data, using C=0.0080, in which case we achieved a 2/44 misclassification performance for BF and 0/44 for SBF.

In this experiment we eliminated 20, then 40 then 60 of the activators, inhibitors and neutrals. As expected we see that the performance decreases when activators or inhibitors are eliminated, while the elimination of the neutrals does not affect the performance.

However we note that the second best fit algorithm is less sensitive to changes than the best-fit algorithm.

Pattern category	20		40		60		100	
	BF	SBF	BF	SBF	BF	SBF	BF	SBF
Positive	4/44	1/44	4/44	1/44	5/44	0/44	7/44	1/44
Negative	4/44	1/44	5/44	1/44	5/44	1/44	7/44	1/44
Neutral	2/44	0/44	2/44	0/44	2/44	0/44	2/44	0/44

Table 6. Effect of removing activators, inhibitors and neutral patterns over the SB and SBF predictions

Another experiment was run eliminating substantially more neutral patterns.

Pattern category	100		200		318		320		600		700	
	BF	SBF	BF	SBF	BF	SBF	BF	SBF	BF	SBF	BF	SBF
Neutral	2/44	0/44	2/44	0/44	2/44	0/44	3/44	1/44	3/44	1/44	5/44	1/44

Table 7. Effect of removing more neutral patterns over the SBF and BF prediction

We see that we can eliminate over 300 patterns without any significant loss in the quality of the classification, and that we can eliminate up to 600 patterns out of 816 if we accept to compromise the performance in a minimal way. Again we see how robust the second best fit algorithm is.

This ranking of patterns is meaningful biologically as it helps to determine which patterns play a role in determining the correct TIS. In another series of experiments to test this conclusion, we generated a set of weights for the patterns using the SVM optimal parameters found in the previous section, and compared them to sigma promoter entries in RegulonDB (Araceli et al., 1998). We only took promoters with a score of 4.0 or more. Figure 1 plots each pattern versus the annotated SVM weight and its frequency of occurrence in the database of promoters.

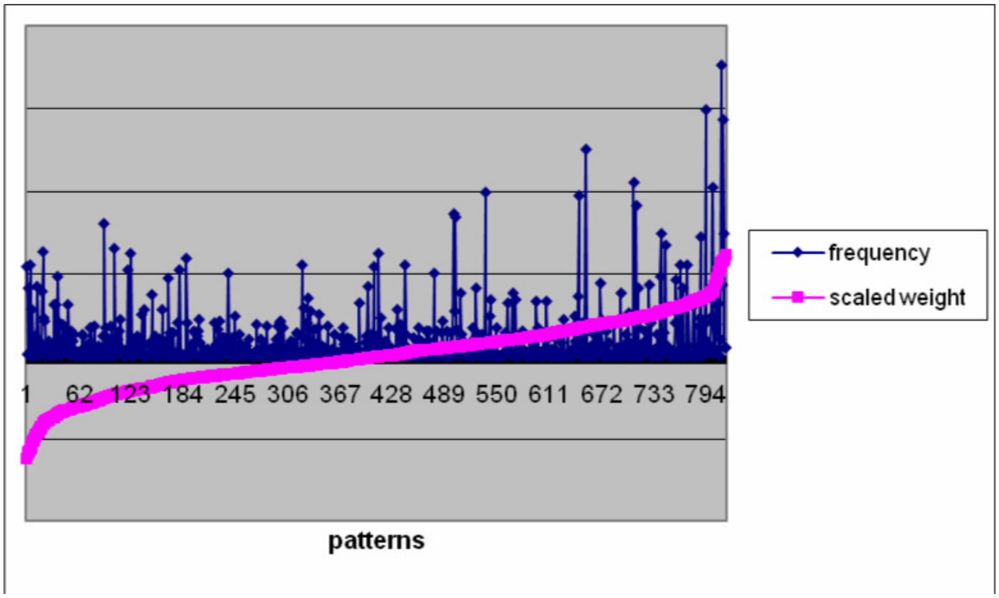


Figure 1. Results of our predictions, when comparing RegulonDB entries to the list of ranked patterns obtained from the experiment.

An important observation drawn from Figure 1 is that there seems to be a significant amount of background 'noise' introduced by patterns that have a high random probability of occurrence, because either they are very short or have a very loose structure (many positions filled with 'generic' nucleotides). Almost 85% of the set of patterns have 20 occurrences or less. Nevertheless it is clear that patterns that have higher weight have significantly more matches (extreme right of the graph). It is a bit less clear that the patterns with essentially zero weights occur only with a background noise frequency (in the 250/320 zone on the horizontal axis).

So we found some indication that the most (in)significant patterns from an SVM classification point of view have also a (low)high number of matches to RegulonDB promoter entries. We view this as a justification for a more systematic study.

Now let us consider the reverse problem: do the promoter entries in RegulonDB have an effect on SVM classification?

So, in another set of experiments we chose as input in our set of patterns those patterns that have a minimum threshold frequency of successively 5, 10, 15 and 20 in the RegulonDB promoter entries. We also varied the SVM parameters. The detailed results are to be found in the appendix. In summary what we found was:

1: The sizes of the various inputs vary from 300 to 100, and similarly to the previous experiments where we had similar input sizes we see a corresponding decrease in performance. It is however more pronounced and that can be understood because not all the patterns we are removing are "background noise", among them, we are also removing patterns classified as promoters in RegulonDB (Araceli et al., 1998).

2: The results here depend more on variations of the parameters. In particular linear SVM's are not sufficient to obtain the best classification. This is due to the fact that the number of patterns (the dimension) becomes less than the number of points.

3: The best fits are significantly poorer than in our previous experiments. Still the worst success rates are around 78% while the best are around 89%. The second best fit heuristic results are also in general poorer than in the previously reported experiments, however, playing with the parameters one can still find success up to 98%.

Here, we have no doubt, even though more thorough experiments are warranted, that the patterns found in RegulonDB can be used for classification by SVM, strengthening the claim that SVM can play a role in detecting and evaluating potential promoter patterns.

6. CONCLUSION and FUTURE WORK

One can view the work presented here as a pilot study motivating further systematic studies of the model words/documents used with such success in text classification.

Here we have studied patterns/windows, but one could have easily studied patterns/proteins, or genes/genomes, and in that context one can make a systematic use of support vector machines or other tools such as Singular Value Decomposition and Latent Semantic Analysis that we addressed in the next section:

SINGULAR VALUE DECOMPOSITION

There are several aspects of interest in singular value decomposition. The main ones are noise removal, latent semantic analysis, principal component analysis, and pseudo inverse. Here, we will only consider the first two aspects.

A main aim of SVD/LSA (Berry et al., 1995; Deerwester et al. 1990; Dumais et al. 1988; Berry and Brown, 1999) is to improve retrieval rates from queries in search engines. One way to accomplish this is to compute an approximation of the data by using a truncated singular value decomposition to remove the noise, and then apply the SVM software on the cleaner input. The difficulties of this approach are mainly computational, because running the singular value decomposition on the original matrix is a very costly procedure. But, from the previously described experiments we knew that we could use SVM to eliminate the neutral patterns, 'the background noise'. This leads to a substantially smaller matrix that can be used in a second step as input to the singular value decomposition method.

A typical experimental result is shown in table 8. We chose, on purpose, cases where the parameters' settings do not give us optimal results for the SVM. We can then see more clearly that applying a truncated SVD to the data prior to running the SVM software indeed removes noise and improves the SVM performance. In these experiments, all but the largest 200 singular values were set to zero.

	Non-Binary Non-Normalized C=0.008	Binary Non-Normalized C=0.01	Non-Binary Non-normalized C=0.01
No SVD	4	5	6
With SVD	2	4	4

Table 8. Effect of SVD over the number of misclassifications produced by SVM using BF

As in search-engine applications, it is clear that SVD has a positive effect overall. We must note though, that we did not study the more involved aspects of latent semantic analysis such as the handling of polysemy and synonymy, although they deserve to be studied thoroughly, given that some types of mutations create a wealth of synonyms.

An interesting starting point might be that the weight vector produced by the SVM algorithm plays the role of an "ideal" query, one that helps retrieve all the documents of interest and only those. Synonymy and polysemy could be inferred from this weight vector. Two synonyms will have similar subcontexts, these can be evaluated by studying the weights of the words in the contexts, which must compensate for the absence of one of the words in each document. Conversely polysemy will be reflected by different subcontexts, whose words must have significant weights.

Another view of this work, of course, is that we have produced specific results regarding the automated discovery of TIS and promoter sequences. Regarding SVM applications, we have confirmed, via a different route, the results of Zien (Zien et al., 2000). More specifically we have shown that using a pattern based representation of the potential TIS sites, allowed us to gather useful information regarding potential promoters. The use of additional heuristics (such as the second best fit) improved the prediction and practicality of our system; there exist other heuristics though, that could be used in conjunction and/or instead of SBF, such as asymetric distribution or relative abundance of start codons. This is something that will be incorporated in the near future.

7. REFERENCES

Araceli M.H., Salgado,H., Thieffry,D. and Collado-Vides,J. (1998) RegulonDB: A Database on Transcriptional Regulation in -Escherichia Coli. 1998 Oxford University Press Nucleic Acids Research, 26(1), 55-59.

Berry M.W. and Brown M. (1999) Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM Book Series: Software, Environments, and Tools.Philadelphia.

Berry,M.W., Dumais,S.T. and O'Brien,G.W. (1995) Using Linear Algebra For Intelligent Information Retrieval. SIAM Review, 37(4), 573-595.

Brown,M., Grundy,W., Lin,D., Cristianini,N., Sugnet,C., Furey,T., Ares,M. and Haussler,D. (1999) Knowledge-based Analysis of Microarray Gene Expression Data Using Support Vector Machines. Proceedings of the National Academy of Science, 97(1), 262-267.

Brazma,A., Jonassen,I., Eidhammer,I., Gilbert,D. (1998) Approaches to Automatic Discovery of Patterns in Biosequences, Journal of Computational Biology, 5(2), 277-303.

Burges,C. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 121-167.

Cortes.C. and Vapnik,V.N. (1995) Support-vector networks. Machine Learning, 20, 273-297.

Cristianini,N. and Shawe-Taylor,J. (2000) An Introduction to Support Vector Machines. Cambridge University Press. Cambridge

Deerwester,S., Dumais,S.T., Landauer,T.K., Furnas,G.W. and Harshman,R.A. (1990) Indexing by latent semantic analysis. Journal of the Society for Information Science, 41(6), 391-407.

Drucker,H., Wu,D. and Vapnick,V.N. (1999) Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10(5), 1048-1054.

Dumais,S.T., Furnas,G.W., Landauer,T.K. and Deerwester,S. (1988)Using latent Semantic analysis to improve information retrieval. In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.

Dumais,S.T., Platt,J., Heckerman,D. and Sahami,M. (1998). Inductive Learning Algorithms and Representations for Text Categorization. Proceedings of ACM-CIKM98, 148-155.

Guyon,I. http://www.clopinet.com/isabelle.

Thorsten J. http://ais.gmd.de/~thorsten/svm_light.

Thorsten J. (1998) Text Categorization with Support Vector Machine: Learning With Many Relevant Features. European Conference on Machine Learning.

Klinkenberg R. and Thorsten J. (2000) Detecting Concept Drift with Support Vector Machines. Proceedings of the Seventeenth International Conference on Machine Learning (ICML), Morgan Kaufman.

Pedersen,A.G. and Nielsen,H. (1997) Neural Network Prediction of Translation Initiation Sites in Eukaryotes: Perspectives for EST and Genome analysis. ISMB'97, 226-233.

Vapnik,V.N. (1995) The Nature of Statistical Learning Theory. SpringerVerlag. New York.

Vapnik,V.N. (1998) Statistical Learning Theory. John Wiley & Sons.

Zien,A., Rätsch,G., Mika,S., Schölkopf,B., Lemmen,C., Smola,A., Lengauer,T. and Müller,K.R. (2000) Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. Bioinformatics, 16 (9), 799-807

Glossary

Accession number is a unique number or combination of letters and numbers assigned to each record in a database.

Agglomerative Hierarchical Clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. An example of this is species taxonomy.

Alpha helix is one of two types of protein secondary structures. An alpha helix is a tight helix that results from the hydrogen bonding of the carboxyl (CO) group of one amino acid to the amino (NH) group of another amino acid.

Amino acids are the basic building block of proteins. There are 20 different amino acids that link together in various orders to form proteins.

Amino group is a functional group with one nitrogen and two hydrogen atoms.

Archaea are single-celled organisms without nuclei and with membranes different from all other organisms.

Average-linkage is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group.

BLAST is the fastest sequence analysis program, but compromises some degree of sensitivity for speed.

BLASTP compares a nucleotide query sequence against a nucleotide sequence database.

BLASTN compares an amino acid query sequence against a protein sequence database.

BLASTX compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

BLITZ also provides a very sensitive search but is very slow to run.

BLOSUM matrices are calculated matrices, most sensitive for local alignment of related sequences, ideal when trying to identify an unknown nucleotide sequence.

Cancer is any malignant growth or tumor caused by abnormal and uncontrolled cell division.

cDNA or complementary DNA is synthesized in the laboratory from a messenger RNA template.

CDS is the coding sequence or the portion of a nucleotide sequence that makes up the triplet codons that actually code for amino acids.

Centroid is the average value of a group of objects in a cluster.

Chromosome is a threadlike linear strand of DNA and associated proteins in the nucleus of animal and plant cells that carries the genes and functions in the transmission of hereditary information.

Cladogram is a dichotomous phylogenetic tree that branches repeatedly, suggesting the classification of molecules or organisms based on the time sequence in which evolutionary branches arise.

Clustal W is a general-purpose program for multiple alignments of DNA and protein sequences developed by Thompson, et. al., in 1994.

Clustal X is a new windows interface for the ClustalW multiple sequence alignment program.

Clustering is the process of organizing objects into groups whose members are similar in some way.

Codon is a set of three adjoined nucleotides (triplet) that codes for an amino acid or a termination signal.

Complete-linkage also called farthest neighbor, clustering method is the opposite of single linkage. Distance between groups is now defined as the distance between the most distant pair of objects, one from each group.

Conservation is when a substitution of one amino for another preserves the physico-chemistry properties of the original residue.

Contig is a group of cloned (copied) pieces of DNA representing overlapping regions of a particular chromosome

Deletion is the loss, as through mutation, of one or more nucleotides from a chromosome.

DDBJ - (DNA Databank of Japan) is the sole DNA data bank in Japan, which is officially certified to collect DNA sequences from researchers and to issue the internationally recognized accession number to data submitters.

DNA is the material inside the nucleus of cells that carries genetic information. The scientific name for DNA is deoxyribonucleic acid.

Domain is a discrete portion of a protein assumed to fold independently of the rest of the protein and possessing its own function

EBI (European Bioinformatics Institute) is a non-profit academic organisation that forms part of the European Molecular Biology Laboratory (EMBL).

EMBL (EuropeanNucleotide Sequence Database) constitutes Europe's primary nucleotide sequence resource.

Entrez is a molecular sequence retrieval system, which contains an integrated view of all publically available nucleotide and protein databases.

ESTs (Expressed Sequence Tags) are pieces of the genetic sequence for genes that are turned "on," ie, actively being transcribed into messenger RNA in the cell.

Eukaryote is a cell or organism with membrane-bound, structurally discrete nucleus and other well-developed subcellular compartments.

E-value is the number of hits in a database search that we expect to see by chance with this score or better.

Evolutionary Distance in phylogenetic trees is the sum of the physical distance on a tree separating organisms; this distance is inversely proportional to evolutionary relatedness.

Exon is the protein-coding DNA sequence of a gene. Compare intron

Expect option - ten is default. This means that 10 matches are expected to be found by chance.

FASTA is an alignment program created by Pearson and Lipman in 1988. The program is one of the many heuristic algorithms proposed to speed up sequence comparison.

FASTA Format is a text-based format for representing either nucleic acid sequences or protein sequences.

Filtering is the process of hiding regions of (nucleic acid or amino acid) sequence having characteristics that frequently lead to spurious high scores. Also known as masking.

FingerPRINTScan is a tool that classifies sequences using the family definitions that are present in the PRINTS database. It is particularly good at detecting distant evolutionary relationships.

Finite State Machine (FSM's) are machines which proceed in clearly separate and discrete steps from one to another of a finite number of configurations or states

Frameshift mutation is an alteration of DNA where insertion or deletion of sequence occurs that is not a multiple of three base pairs, thus disrupting the gene/protein

Gap is a space introduced into an alignment to compensate for insertions or deletions in one sequence relative to another

Gene is the basic biological unit of heredity; a segment of deoxyribonucleic acid (DNA) needed to contribute to a function.

Gene locus is a gene's position on a chromosome or other chromosome marker; also, the DNA at that position.

Gene Ontology is a controlled vocabulary of terms relating to molecular function, biological process, or cellular components.

GenBank is a large database and data repository of nucleic acid and protein sequences at the National Library of Medicine (USA).

Genecards offers information about human genes and their mouse homologs, with a focus on cellular functions and involvement in diseases.

Genetic Disease is a disease caused by a genetic mutation that is either inherited or arises spontaneously.

Genome is all of the genetic information or hereditary material possessed by an organism; the entire genetic complement of an organism.

Global alignment is when two nucleic acid or amino acid sequences are lined up along their entire length.

H is the relative entropy of the target and background residue frequencies.

HIV (human immunodeficiency virus) is the causative agent of Acquired Immunodeficiency Syndrome (AIDS).

Homolog is a gene related to a second gene by descent from a common ancestral DNA sequence.

Homology is the structural similarity due to descent from a common ancestor or form.

HMM (Hidden Markov Model) a probabilistic model used to align and analyze sequence datasets by generalization from a sequence profile.

HMM Scoring is related to the statistical significance of the alignment. A score of zero is marginal; according to the model's statistics, it's 50% likely that the alignment is a real match to the model, and 50% likely that it's not. The higher the score, the better.

Inheritance - attributes acquired via biological heredity from the parents.

Insertion is a chromosome abnormality in which material from one chromosome is inserted into another nonhomologous chromosome; a mutation in which a segment of DNA is inserted into a gene or other segment of DNA, potentially disrupting the coding sequence.

Intron - "intervening sequence," a stretch of nucleic acid sequence spliced out from the primary RNA transcript before the RNA is transported to the cytoplasm as a mature mRNA; can refer either to the RNA sequence or the DNA sequence that from which the RNA is transcribed. See also exon.

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a bioinformatics resource for linking genomes to life and the environment.

K-means Clustering is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assumes k clusters) fixed a priori.

Local alignment is the alignment of portions (rather than the entire sequence length) of two nucleic acid or amino acid sequences

Low Complexity Region (LCR) are regions of biased composition including homopolymeric runs, short-period repeats, and more subtle overrepresentation of one or a few residues.

Macromolecular Structure is the structure of larger biologically important organic molecules known as carbohydrates, lipids, proteins, and nucleic acids.

Markov Processes are useful for analyzing dependent random events - that is, events whose likelihood depends on what happened last.

Masking is the removal of repeated or low complexity regions from a sequence so that sequences are compared

Motif is a discrete portion of a protein assumed to fold independently of the rest of the protein and possessing its own function

Multiple sequence Alignment is a bioinformatics tool that compares multiple DNA or amino acid sequences and aligns them to highlight their similarities.

Mutation is a change in the number, arrangement, or molecular sequence of a gene.

NCBI is the National Center for Biotechnology Information, a division of the NIH, is the home of the BLAST and Entrez servers.

Objective function (optimality criterion) is a function that defines how well data fit a particular hypothesis (as, for instance, a particular phylogenetic tree).

Oncology is the branch of medicine devoted to the diagnosis and treatment of cancer.

Open Reading Frame (ORF) contains a series of triplets coding for amino acids without any termination codons; sequence is (potentially) translatable into protein.

ORF Finder is a graphical analysis tool which finds all open reading frames of a selectable minimum size in a user's sequence or in a sequence already in the database.

Orthologous sequences are homologous sequences in different species that result from a common ancestral gene during speciation. Orthologous genes may or may not have similar functions.

Overfitting is caused by using too many covariates for the number of outcome events in a multivariable predictive model; can lead to spurious or incorrect associations.

PAM (Percent Accepted Mutation) is a unit to quantify the amount of evolutionary change in a protein sequence.

PAM matrices are predicted matrices, most sensitive for alignments of sequences with evolutionary related homologs.

Paralogous sequences are homologous sequences within a single species that arose by gene duplication.

Peptides are two or more amino acids chained together by a bond called a "peptide bond."

Percent identity (called "Identities" is given as a percent) is the percent of exact matches between your query sequence and the database sequence.

Phylogenetic tree is tree-like diagram that depicts the evolutionary relationships between different organisms.

PIR is the Protein Identification Resource, a database of protein sequences.

Polypeptide chain is a chain of peptides or amino acids. A polypeptide chain usually consists of 100 or fewer amino acids. A protein is made up of one or several polypeptide chains.

Primary structure is the amino acid sequence of a polypeptide chain. This is the most basic protein structure.

Profile is an analysis (often in graphical form) representing the extent to which something exhibits various characteristics

Prokaryote is a cell or organism lacking a membrane-bound, structurally discrete nucleus and other subcellular compartments. Bacteria are prokaryotes.

Proteomics is the systematic analysis of protein expression of normal and diseased tissues that involves the separation, identification and characterization of all of the proteins in an organism.

Quaternary structure is the interconnection and arrangement of polypeptide chains within a protein. Only proteins with more than one polypeptide chain can have quaternary structure.

RefSeq database is a curated database of Genbank's genomes, mRNAs and proteins.

RNA is a nucleic acid molecule similar to DNA but containing ribose rather than deoxyribose.

SARS is a severe form of pneumonia, caused by a virus, that appeared in outbreaks in 2003.

Score (bits) is a sum value calculated for alignments using the scoring matrix; the higher the score value, the better the alignment.

Scoring: if a nucleotide in the query word exactly matches a nucleotide at the same position in the database word (e.g. A with A), then a positive score is awarded, ….

Scoring Matrix - A matrix that defines scores for amino acid substitutions, reflecting the similarity of physicochemical properties, and observed substitution.

Score Option is the ratio of M:N determines the degree of evolution that is accepted. The default values for M are 5 and N is 4.

Secondary structure is the folded, coiled, or twisted shape of a polypeptide that results from hydrogen bonding between parts of a molecule. There are two types of secondary structure: alpha helix and a beta pleated sheet.

Similarity is defined in this book as how well one sequence matches another determined by calculation by an alignment program of identical and conserved residues.

Single-linkage is one of the simplest agglomerative hierarchical clustering methods, also known as the nearest neighbor technique. The defining feature of the method is that distance between groups is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each group are considered.

Substitution Matrix contains values proportional to the probability that amino acid i mutates into amino acid j for all pairs of amino acids.

Taxonomy is the study of the general principles of scientific classification, especially the orderly classification of plants and animals according to their presumed natural relationships.

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.

Tertiary structure is the three-dimensional structure of a polypeptide chain that results from the way that the alpha helices and beta pleated sheets are folded and arranged

TISHunter is a highly accurate predictor for translation initiation sites in human mRNAs.

Transeq translates nucleic acid sequences to the corresponding peptide sequence.

Translation is the process whereby genetic information coded in messenger RNA directs the formation of a specific protein at a ribosome in the cytoplasm.

TREMBL is the supplement of SWISS-PROT that contains all the translations of EMBL nucleotide sequence entries not yet integrated in SWISS-PROT.

Trinucleotide is a polymer made up of three mononucleotides.

Unitary Matrix is a scoring system in which only identical characters receive a positive score. Also known as the identity matrix.

Universal Frame Finder is a tool which can be used to find the frame for the coding sequence of any organism.

Unsupervised learning methods do not assume any a priori knowledge. The computer has to learn from the data set by itself.

Index