SOFT COMPUTING CHALLENGES
IN BIOINFORMATICS

K. R. PARDASANI
Professor & Former Head
DEPARTMENT OF MATHEMATICS,
BIOINFORMATICS & COMPUTER
APPLICATIONS
MAULANA AZAD NATIONAL INSTITUTE
OF TECHNOLOGY (MANIT)
BHOPAL - 462003
Email kamalrajp@rediffmail.com
Website: http://www.krpardasani.com
WHAT IS BIOINFORMATICS?

• Conceptualizing Biology in terms of molecules and applying informatics techniques (applied math's, computer science, and statistics) to understand and organize the information associated with these molecules on a large scale.

• MIS for molecular Biology

FUNDAMENTAL ISSUES:
How we describe, analyze, simulate and predict the dynamics of various biological processes using IT tools.

COMPLEXITY: due to massive amount of data obtained through numerous Biological experiments.

Managing and interpreting Biological data:
EMBL database: 17,807,926,047 nucleotides
Total entries: 15,851,373
BIOINFORMATICS

Biology, Computer Science & IT

Ultimate goal:- to enable discovery of new biological insights as well as create global perspective from which unifying principles of Biology can be discerned.

Job prospects:-

• $300 billion Pharmaceutical industries
• Fast growing biotech industry to support it
• Drug design going to be genomics related
• Shortage of Bioinformatics and molecular modeling specialists.
• Global market share for Bioinformatics is expected to cross US $ 60 billion by year 2006.

Salaries range:-

US $ 80,000/- to US $ 200,000/- per annum.
BIOINFORMATICS

• Science of using information to understand biological phenomena
• Part of Computational Biology

Bioinformatics consists of:

• **DNA Micro arrays:** technology to measure relative copies of *genetic message at different stages of development or disease.* (*levels of gene expression*)
• **Functional genomics:** large scale ways of identifying gene functions & associations.
• **Structural genomics:** attempts to crystallize/predict structures of all proteins.
• **Comparative genomics:** understand differences & similarities between all the genes of multiple species – evolution.
• **Medical informatics:** management of biomedical experimental data.
OBJECTIVES:

• Organizing data
• Develop tools & resources for analysis of data
• Analysis and interpretation of results

APPLICATIONS:

• Sequence analysis
• Primer design – short sequences to make many copies of a piece of DNA sequences
• Predict function of actual gene products
• Molecular modeling
• Crystallography – structural biology
• Genetic engineering
BIOINFORMATICS
(AN OVERVIEW)
THE CURRENT EXCITEMENT IN BIOLOGY
THE CHALLENGES OF BIOINFORMATICS.

TODAY, TOMORROW AND THE NEAR FUTURE

THE THREE E's

Extracting
Envisaging → THE BIOLOGICAL DATA
Elucidating
CENTRAL DOGMA

OF EARLY MOLECULAR BIOLOGY
DNA
↓↑
mRNA
↓
PROTEIN

Shift of paradigm

OF BIOINFORMATICS
SEQUENCE
↓
STRUCTURE
↓
FUNCTION
THE AGE OF BIOCOMPUTING

THE DNA COMPUTING

(The gene-protein assessment)
THE FUTURE NEXT OF BIOINFORMATICS

- Genome \( \text{GENOMICS} \)
- Protein \( \text{PROTEOMICS} \)
- Biochemical pathways \( \text{METABOLITES} \) (METABONOMICS ???)
- Phenotype (Function)

- FUNCTOMICS
- TRANSCRIPTOMICS

\( \sim \text{THE OMICS} \)
THE RELATED DISCIPLINES OF BIOINFORMATICS

Cheminformatics

Medical Informatics

Health Informatics

Medical computing

Nursing informatics
STRUCTURE PREDICTION : UNLOCKING BIOLOGICAL SECRETS (GENE IDENTIFICATION TOOLS)

- EXPASY
- SWISS MODEL
- GENO 3D
- CPH MODELS ETC...
Insilico APPROACHES OF THE DRUG DEVELOPMENT

ADME PROPERTIES

ABSORPTION
DISTRIBUTION
METABOLISM
ELIMINATION
WHAT METABONOMICS IS ALL ABOUT

ITS RELATIONSHIP USING BIOINFORMATICS TOOLS

ANABOLISM + CATABOLISM ---→ Metabolism

THE STUDIES USING BIOINFORMATICS......METABONOMICS

PATTERN RECOGNITION AND DATABASES

- REBASE
- PROSITE
- TRANSCRIPTION FACTOR DATABASE
- TRANSFAC
- EUCAARYOTIC PROMOTER DATABASE
BIOINFORMATICIST

BIOTECHNOLOGY CONCEPTS

+

INFORMATION TECHNOLOGY TOOLS

YAHOOO!!! I AM A HYBRID
THE EXTENDING SAGA OF BIOINFORMATICS

Human Genome Map for $250 only
BUY ONE AND GET MOUSE GENOME TRANSCRIPTOME FREE

WANT YOUR (SEQUENCE) MAP.??????
REGISTER
LIVING WITH BIOINFORMATICS??????

WAAHH!!!
MOUSEMAN....
WONDER HAVE THEY
JUST SEQUENCED
MOUSE GENOME MAP?
SEQUENCES

Sequences:- Viewed as strings of characters for convenience of understanding & performing Mathematical functions.

• Proteins & DNA may be similar with respect to their function, structure or primary sequence of amino or nucleic acids.
• Sequence determines shape, shape determines function.
• We study sequence similarity to discover similarity in shape & function.
Similarity in sequences

- Similarity measure i.e. two sequences show certain degree of similarity
- An alignment i.e. mutual arrangement of two sequences where two sequences are similar & where they differ

Quantitative

Optimal alignment – that exhibit most correspondences & least differences.

Qualitative
BIOLOGICAL MOTIVATIONS OF SEQUENCE ANALYSIS

- Large variety of biological problems involve sequences
- Sequence alignment – useful for discovering information related to functions, structure and evolution

Examples:
- Reconstructing long sequences of DNA from overlapping strings fragments.
- Determine physical and genetic maps from probe data under various experiments protocols.
- Storing, retrieving and comparing DNA strings.
- Comparing two or more strings for similarities to find related Proteins
- Exploring frequently occurring patterns of nucleotides.
- Finding informative elements in Proteins & DNA sequences.
- Identify an unknown sequence.
- Find other members of multigene families.
**Aim:** Learn functionality & structure of Protein without performing experiments & without physically constructing Protein itself.

**Basic idea:** Similar sequences produce similar proteins.

Predict characteristics of Proteins using its sequence data.

**Example:** Let two Protein sequences are identical at 25% of their positions. This association is found in Cancer and uncontrolled growth cells. Compare sequence of Cancer associated gene and sequence of Protein which influences cell growth.

- Correlation was very high.
- Proves connection between the two.
**Concepts:**

**Identical:** when corresponding character is shared between two species that character is said to be identical.

**Similar:** Degree to which two species or populations share identities.

**Homologous:** When characters are similar due to common ancestry.

**Analogous:** When characters are similar due to convergent evolution they are analogous.

**Orthologous:** When characters are homologous with conserved function.

**Paralogous:** When characters are homologous with divergent function.
SIMILARITY & DIFFERENCES

**Similarity:** Maximal sum of weights. Assign weights corresponding for resemblance.

- Occurred due to mutations – modifying DNA sequences
  - Insertion of letter/letters in a sequence.
  - Deletion of letter/letters in a sequence.
  - Substitution of letter by another.

- The notion of distance, assigning weights to each mutation.
- Distance between minimal sum of weights for a set of mutations.
MODELS FOR SEQUENCE ANALYSIS

- **Global alignment**
  
  **Input:** two strings S & T roughly of same length
  
  **Q:** What is the difference (similarity) between the two?
  
  It is done across entire sequence length to include as many matches as possible including sequence end.

- **Local alignment / similarity (more meaningful)**
  
  **Input:** Two strings S & T
  
  **Q:** What is maximum similarity (minimum difference) between substring of S & substring of T?
  
  **Q:** What are these most similar substrings?
  
  **Example:**
  
  S = a b c x d e x
  
  T = x x c - d e
  
  We give each match a value 2 & mismatch a value -1.
  
  \[ \alpha = c x d e \]
  
  \[ \beta = c \ - \ d e \]
  
  Have optimal alignment
Model for sequence analysis continued -

- **Ends free space alignment**
  
  **Input:** two strings of S & T of different lengths.
  
  **Q:** what is maximum similarity between substrings of S & T?
  
  **Given:** least one of these substrings must be prefix of the original string & one (not necessarily other) must be a suffix.

**Example:**

S = - - c a c - d b d v l

T = l t c a b d d b - - -

Two leading spaces at left end of alignment are free as well as three trailing spaces at right hand side.

- **Gap penalty**
  
  **Input:** Two strings S & T of different length.
  
  **Q:** Define gap as any maximal consecutive run of spaces, length of gap as the number of indel operations. What is the similarity between two strings, given weight function for gaps.
Example:

S = a t t c - - g a - t g g a c c
T = a - - c g t g a t t - - - c c

Four gaps of total eight spaces.

Then alignment would be described as

- Seven matches
- No mismatch
- Eight spaces

- Length of gap is No. of \textit{indel} operations.

- Concept of gap in alignment is important in many biological applications.

- Mutational events create gaps of varying sizes.
METHODS OF ALIGNMENT

• **DOT MATRIX** – useful for simple alignment, however does not show sequences or produce optimal alignment.

• **BRUTE FORCE** – produces alignments without gaps and has an N^2 complexity, where N is length of sequences.

• **DYNAMIC PROGRAMMING** – produces optimal alignment by starting an alignment from one end (as in dot matrix), then keeping track of all possible best alignments to that point.

• **HEURISTICS METHODS** – fast computational machine-based methods. May not be as accurate as dynamic programming.

**GRAPHIC SIMILARITY COMPARISONS:**

<table>
<thead>
<tr>
<th>GGCTTGACCGG</th>
<th>GGCTTGACCGG</th>
<th>GGCTTGACCGG</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥</td>
<td>≥</td>
<td>≥</td>
</tr>
<tr>
<td>GGATTTGACCCG</td>
<td>GGATTTGACCCG</td>
<td>GGATTTGACCCG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SIMILARITY VERUS DISTANCE

1. Elements of the matrices specify the weight to assign a given comparison by:
   - The cost of replacing one residue with another (distance); or
   - A measure of the similarity for the replacement.
2. Similarity is used for database searching.
3. Distance is more applicable for phylogenetic tree reconstruction.
4. Maximizing the similarity is fundamentally the same as minimizing a distance. Hence distance and similarity matrices are inter-convertible by some mathematical transformation appropriate for the given application.

<table>
<thead>
<tr>
<th>SIMILARITY</th>
<th>DISTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local alignment</td>
<td>Evolution &amp; phylogeny</td>
</tr>
<tr>
<td>Suited for comparing proteins</td>
<td>Triangle inequality</td>
</tr>
</tbody>
</table>
GLOBAL Vs LOCAL SIMILARITY

Global algorithms are not sensitive for highly diverged sequences, a better and faster method is local similarity. Three most widely used local similarity algorithms are – Smith-Waterman, BLAST, FASTA.

**Smith-Waterman:**
- it is a rigorous dynamic programming approach
- it does not make use of heuristic shortcuts

**FASTA:** *(developed by Lipman & Pearson in 1985)*
- considers exact matches between short substrings, for a given parameter
- allows to trade-off speed for precision: the larger we choose the parameter, the smaller is the number of exact matches
- Makes the program faster but loses precision

**BLAST:** *(developed by Altschul et al. in 1990)*
- it focuses on no-gap alignments of a certain, fixed length
- it uses a scoring function to measure similarity rather than distance
- It reports to the user all database entries which have a segment pair scores higher than the threshold parameter.
FASTA algorithm

(a) Sequence B

Sequence A

Find runs of identical words

(b) Sequence B

Sequence A'

Re-score using PAM matrix
Keep top scoring segments

(c) Sequence B

Sequence A

Join segments using gaps,
eliminate other segments

(d) Sequence B

Sequence A

Use dynamic programming to
create an optimal alignment
BLAST (Basic Local Alignment Search Tool) is a similarity search program developed by the research staff at NCBI/GenBank. It is available as a free service over the Internet that provides very fast, accurate, and database searching.

**BLAST goes through the following 3 steps**

- It takes each word from the query sequence (3 amino acids or 11 nucleotides).
- If similar words are found, **BLAST** tries to expand the alignment to the adjacent words.
- After all words are tested, a set of **HSPs (High-scoring Segment Pairs)** are chosen for that database sequence.
(1) Find the list of high scoring words $w$

Query sequence of length $L$

Maximum of $L-w+1$ words (typically $w = 3$ for proteins)

For each word from the query sequence find the list of words that will score at least $T$ when scored using a pair-score matrix [e.g. PAM 250]

(2) Compare the word list to the database and identify exact matches

Word list

Database sequences

Exact matches of words from word list

(3) For each word match, extend the alignment in both directions to find alignments that score greater than a threshold of value $S$

Maximal Segment Pairs (MSPs)
POPULAR SCORING MODELS FOR PROTEIN SEQUENCES

There are two popular scoring models for protein sequences – PAM and BLOSUM. PAM stands for Percent Accepted Mutation and BLOSUM stands for BLOcks SUbstitution Matrix.

**PAM is**
- based on explicit evolutionary model
- represents a specific evolutionary distance
- ranges from identical to completely random

**BLOSUM is**
- based on empirical frequencies
- always a blend of distances as seen in the database and PROSITE
- narrower range than PAM matrix
Representation of dot plots
GRAPHIC SIMILARITY COMPARISONS

• Uses the power of computer to present relationships between sequences

• Similarity between two sequences can be detected as a diagonal on an identity matrix

• To determine the similarity of sequences, we must compare all parts of one sequence with all parts of the other

• The alignment with the greatest number of identities would be the optimal alignment
**Graphic similarity comparisons**

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>T</th>
<th>G</th>
<th>A</th>
<th>C</th>
<th>C</th>
<th>G</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>G</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

Representation of scoring matrix
Representation of sequence $s$ & $t$
METHODS FOR OPTIMAL ALIGNMENT

**Global sequence alignment:**
- Here dynamic programming is used which is a method for breaking down the alignment of sequences into small parts.
- It is comparable to moving across a dot matrix and keeping track of all the matching pairs.
- Sequence alignment method predate dot-matrix searches and all of the alignment methods in use today.
- Over the course of evolution, some positions undergo base or amino acid substitution and bases or amino acids can be inserted or deleted.

**Local alignment:**
- Smith-Waterman dynamic programming algorithm is used for local alignment.
- The algorithm gives the highest-scoring local match between two sequences.
- The alignment are arrived at by starting at the highest-scoring positions in the scoring matrix and following a trace path up to a box that scores zero.
EXAMPLE:

- Calculate a dynamic programming matrix and alignment for the sequences ATT and TTC. How many optimal alignments are there?

Matrix:

```
0 1 2 3
1 1 2 3
2 1 1 2
3 2 1 2
```

Alignment:

```
ATT

TTC
```

The other optimal alignment is,

```
ATT-

| |

-TTC
```
Construction of the optimal alignment

\[ a[i,j] = \max \begin{cases} 
  a[i,j-1] - 2 \\
  a[i-1,j] + \rho(i,j) \\
  a[i-1,j-1] - 2 
\end{cases} \]
Hidden Markov Model

• HMMs derive from Markov chain that concentrate only on the sequence state.

• Since the early 1970s
  – Applied in speech recognition research

• The early 1990s
  – Introduced this model to the bioinformatics community
  – Sequence modeling, multiple alignment, protein structure prediction and profiling
Markov Chains

- Markov Property of order 1
- Formally
  \[ P(X_0, X_1, \ldots, X_t) = P(X_0)P(X_1 | X_0)P(X_2 | X_0, X_1) \cdots P(X_t | X_0, \ldots, X_{t-1}) \]
  \[ = P(X_0)P(X_1 | X_0)P(X_2 | X_1) \cdots P(X_t | X_{t-1}) \]

- **State** space = list of possible values for \( X \)
- **Transition matrix** = probability of moving from one \( X \) to another
- **Initial distribution** = initial value of \( X \)

- **CS intuition**
  - Stochastic finite automaton
Markovian Sequence

- States through which the chain passes from a sequence
- Example: \( S_0, S_1, S_1, S_1, S_0, S_1, \ldots \)
  \[
P(\text{seq}) = P(S_0, S_1, S_1, S_1, S_0, S_1, \ldots) = \pi(S_0)P(S_1 | S_0)P(S_1 | S_1) \ldots
\]

- Markov chain for generating DNA sequence
  \( S=\text{AGATCG} \ldots \)
  \[
P(\text{AGATCG}) = \pi(A)P(G | A)P(A | G)P(T | A) \ldots
\]
Hidden Markov Chains (HMMs)

- Observed sequence is a *probabilistic function* of underlying Markov chain
  - Example: HMM for a *(noisy)* DNA sequence
- True state sequence is unknown, but observation sequence gives us a clue

\[ \text{Obs} = \begin{cases} 
A & 0.7 \\
T & 0.3 \\
G & 0.8 \\
A & 0.3 \\
T & 0.7 \\
C & 0.6 \\
T & 0.2 \\
G & 0.8 \\
\end{cases} \]
**MSA (MULTIPLE SEQUENCE ALIGNMENT)**

It is a tool to determine levels of homology, and hence relatedness, between members of a series of globally related sequence.

**Tools for MSA:**

- Sum-of-pairs method
- Star alignment
- Two-step method (Clustal and Pileup approaches)
- Automated tools (Macaw, Meme etc.)
Global & Local MSA (multiple sequence alignment)
Example – **SP** (sum of pairs), method

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>Q</th>
<th>P</th>
<th>I</th>
<th>L</th>
<th>L</th>
<th>L</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>L</td>
<td>R</td>
<td>-</td>
<td>L</td>
<td>L</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>K</td>
<td>-</td>
<td>I</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td>P</td>
<td>P</td>
<td>V</td>
<td>L</td>
<td>I</td>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example – SP method

• The sum of pairs function scores each position in the protein, that is, each column, as the sum of the pairwise scores. For k sequences, there are k \((k-1)/2\) unique pairwise comparisons, excluding self comparisons. Here in column three, the score would be

\[
SP - \text{score} (I, -, I,V) = p (I, -) + p (I,I) + p (I, V) + p (-, I) + p (-, V) + p (I, V)
\]

Where \(p (a, b)\) is the pairwise score of two amino acids.
Optimal alignment between \( k = 3 \) sequence

Where \( K \) is the number of sequences
HMM for Multiple Alignment

- Match” states are alignment sequence positions
- Position-specific deletion penalties
- Position-specific insertion frequencies
- Path through states aligns sequence to model
Example of HMM Model

Transition probabilities ($T$) and emission probabilities ($e$)
Scoring in HMM model

- Score of accuracy along the path

\[ \log_e(0.4) + \log_e(0.3) + \log_e(0.46) + \log_e(0.6) + \log_e(0.97) + \log_e(0.5) + \log_e(0.015) + \log_e(0.73) + \log_e(0.01) + \log_e(1) = -13.25 \]
What is the most likely path through the model?

Prob(A in state I0) = 0.4*0.3 = 0.12

Prob(C in state I1) = 0.05*0.06*0.5 = 0.015

Prob(C in state M1) = 0.46*0.01 = 0.005

Prob(C in state M2) = 0.46*0.5 = 0.23

Prob(Y in state I3) = 0.015*0.73*0.01 = 0.0001

Prob(Y in state M3) = 0.97*0.23 = 0.22

Analogous to dynamic programming
Forward Algorithm

What is the probability of observing the sequence through the model?

Prob(A in state I0) = 0.4*0.3 = 0.12
Prob(C in state I1) = 0.05*0.06*0.5 = 0.015
Prob(C in state M1) = 0.46*0.01 = 0.005
Prob(C in state M2) = (0.005*0.97) + (0.015*0.46) = 0.012
Prob(Y in state I3) = 0.012*0.015*0.73*0.01 = 1.31x10^-7
Prob(Y in state M3) = 0.012*0.97*0.23 = 0.002

No trace-back necessary
HMM for Secondary Structure Prediction

- Available States: helix (H) or loop (L)
- Sequence of hidden states (transition probabilities)
- Generate observed AA sequence (emission probabilities)

\[ P(AA|H) \quad P(AA|H) \quad P(AA|H) \quad P(AA|L) \quad P(AA|L) \]

\[
\begin{align*}
V \quad A \quad W \quad K \quad N \\
A \quad A \quad V \quad N \quad N
\end{align*}
\]
## Introduction

- **Protein synthesis process:** DNA → RNA → Protein

<table>
<thead>
<tr>
<th>First Position (5' end)</th>
<th>Second Position</th>
<th>Third Position (3' end)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>UCU Ser</td>
<td>UAU Tyr</td>
</tr>
<tr>
<td>U</td>
<td>UCC Ser</td>
<td>UAC Tyr</td>
</tr>
<tr>
<td>U</td>
<td>UCA Ser</td>
<td>UAA Stop</td>
</tr>
<tr>
<td>U</td>
<td>UCG Ser</td>
<td>UAG Stop</td>
</tr>
<tr>
<td>C</td>
<td>CUU Leu</td>
<td>CAU His</td>
</tr>
<tr>
<td>C</td>
<td>CUC Pro</td>
<td>CAC His</td>
</tr>
<tr>
<td>C</td>
<td>CUA Pro</td>
<td>CAA Gln</td>
</tr>
<tr>
<td>C</td>
<td>CUG Pro</td>
<td>CGC Gln</td>
</tr>
<tr>
<td>A</td>
<td>AUU Ile</td>
<td>AAU Asn</td>
</tr>
<tr>
<td>A</td>
<td>AUC Thr</td>
<td>AAC Asn</td>
</tr>
<tr>
<td>A</td>
<td>AUA Thr</td>
<td>AAA Lys</td>
</tr>
<tr>
<td>A</td>
<td>AUG Met*</td>
<td>AAG Lys</td>
</tr>
<tr>
<td>G</td>
<td>GUU Val</td>
<td>GAU Asp</td>
</tr>
<tr>
<td>G</td>
<td>GUC Val</td>
<td>GAC Asp</td>
</tr>
<tr>
<td>G</td>
<td>GUA Val</td>
<td>GAA Gln</td>
</tr>
<tr>
<td>G</td>
<td>GUG Val</td>
<td>GCA Ala</td>
</tr>
</tbody>
</table>

**DNA:** ATGAGTAACGCG
**mRNA:** AUG AGU AAC GCG
**Protein:** Met Ser Asn Ala
Introduction

- **Protein**
  - Macro molecules built from 20 basic units

<table>
<thead>
<tr>
<th>AMINO ACID</th>
<th>SIDE CHAIN</th>
<th>AMINO ACID</th>
<th>SIDE CHAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspartic acid</td>
<td>Asp</td>
<td>D</td>
<td>negative</td>
</tr>
<tr>
<td>Glutamic acid</td>
<td>Glu</td>
<td>E</td>
<td>negative</td>
</tr>
<tr>
<td>Arginine</td>
<td>Arg</td>
<td>R</td>
<td>positive</td>
</tr>
<tr>
<td>Lysine</td>
<td>Lys</td>
<td>K</td>
<td>positive</td>
</tr>
<tr>
<td>Histidine</td>
<td>His</td>
<td>H</td>
<td>positive</td>
</tr>
<tr>
<td>Asparagine</td>
<td>Asn</td>
<td>N</td>
<td>uncharged polar</td>
</tr>
<tr>
<td>Glutamine</td>
<td>Gln</td>
<td>Q</td>
<td>uncharged polar</td>
</tr>
<tr>
<td>Serine</td>
<td>Ser</td>
<td>S</td>
<td>uncharged polar</td>
</tr>
<tr>
<td>Theonine</td>
<td>Thr</td>
<td>T</td>
<td>uncharged polar</td>
</tr>
<tr>
<td>Tyrosine</td>
<td>Tyr</td>
<td>Y</td>
<td>uncharged polar</td>
</tr>
<tr>
<td>Alanine</td>
<td>Ala</td>
<td>A</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Glycine</td>
<td>Gly</td>
<td>G</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Valine</td>
<td>Val</td>
<td>V</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Leucine</td>
<td>Leu</td>
<td>L</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Isoleucine</td>
<td>Ile</td>
<td>I</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Proline</td>
<td>Pro</td>
<td>P</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Phenylalanine</td>
<td>Phe</td>
<td>P</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Methionine</td>
<td>Met</td>
<td>M</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Tryptophan</td>
<td>Trp</td>
<td>W</td>
<td>nonpolar</td>
</tr>
<tr>
<td>Cysteine</td>
<td>Cys</td>
<td>C</td>
<td>nonpolar</td>
</tr>
</tbody>
</table>

Polar Amino Acids
Nonpolar Amino Acids
A hidden Markov model for predicting transmembrane helices in protein sequences

- A novel method to model and predict the location and orientation of alpha-helices in membrane-spanning proteins is presented.
- Introduce the probabilistic framework of the hidden Markov model (HMM) to transmembrane helix prediction.
- Using method: Hidden Markov Model (HMM)
Primary Structure

- The sequence of amino acids in the polypeptide chain
- Described as a string from alphabet $\Sigma_{aa}$
  - $|\Sigma_{aa}| = 20$
Secondary structure

- Every amino acid in the sequence belongs to one of the three structural motifs
  - α-helix (H)
  - β-sheet (E)
  - Loop or coil (C)
- flatted to a string from an alphabet
  - $\sum_{ss} = \{H, E, C\}$
Secondary structure

• \( \alpha \)-helix (H)
  – Built up from one continuous region in the sequence
  • Through the formation of hydrogen bonds between residues in position \( i \) and \( i+4 \)
Secondary structure

- **β-sheet (E)**
  - parallel β-sheet
    - Amino acids have
    - the same biochemical direction.
  - Anti parallel β-sheet
    - Amino acids
    - have alternating direction.
Secondary structure

• Loop or coil (C)
  – $\alpha$-helix and $\beta$-sheet are often connected by loop regions.

(a) DNA binding motif
(b) Ca++ binding motif
Tertiary structure

- The 3-dimensional organization of polypeptide chain atoms

- The result of the combinations of secondary structure elements
  - Due to interactions between the amino acids and the solvent
Quaternary structure

The complex spatial conformation of a protein composed of many distinct polypeptide chains (multimeric protein)
Protein Structure

Primary structure - polypeptide chain -

Secondary structure

Quaternary structure

Tertiary structure

Alpha-helix

Beta-sheet
Architecture of the HMM

• The basic architecture of transmembrane HMM (TMHMM)
  – There are three main locations of a residue
    • in the transmembrane helix core
      – in the hydrophobic trail region of the membrane
    • in the transmembrane helix caps
      – in head region of the membrane
    • in loops
Architecture of the HMM

Seven different states:

- One for helix core,
- Two for caps on either side,
- One for loops on the cytoplasmic side
- One each for short and long loops on the non-cytoplasmic side
- One for ‘globular domains in the middle of each loop
Training the HMM

• The first stage
  – Label protein sequence
  – Maximum likelihood estimation of HMMs

• The second stage
  – The first model was used to *relabel* the data.
  – Train from the relabeled sequences with no further unlabeling.

• The third stage
  – Train further by a method for ‘discriminative’ training
Training the HMM

TAL6_HUMAN

Correct
Unlabeled
Relabeled

Correct
Unlabeled
Relabeled

Correct
Unlabeled
Relabeled

Cytoplasmic by ‘i’ (inside)
Transmembrane helices by ‘M’
Non cytoplasmic by ‘o’ (outside)
Datasets

• Set 1
  – The set of 83 proteins originally compiled by (Jones, Taylor, & Thornton 1994)
  – 38 multi-spanning and 45 single-spanning proteins

• Set 2
  – The set of 160 proteins compiled by us
  – 108 multi-spanning and 52 single-spanning proteins

▶ These datasets were divided into 10 sets of about equal size.
  ◆ 9 sets – training set
  ◆ 1 set – sets for prediction
Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Training set size</th>
<th>Stage of training</th>
<th>Correct topology</th>
<th>Correct locations</th>
<th>Single TM sensitivity</th>
<th>Single TM specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMHMM</td>
<td>83</td>
<td>1</td>
<td>60 (72.3%)</td>
<td>62 (74.7%)</td>
<td>95.6%</td>
<td>96.4%</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>2</td>
<td>63 (75.9%)</td>
<td>65 (78.3%)</td>
<td>96.2%</td>
<td>96.5%</td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>3</td>
<td>64 (77.1%)</td>
<td>69 (83.1%)</td>
<td>96.2%</td>
<td>97.6%</td>
</tr>
<tr>
<td>MEMSAT</td>
<td>83</td>
<td></td>
<td>63 (75.9%)</td>
<td>67 (80.7%)</td>
<td>96.8%</td>
<td>94.6%</td>
</tr>
<tr>
<td>PHDhtm</td>
<td>83</td>
<td></td>
<td>(85.5%)</td>
<td>(88.0%)</td>
<td>98.8%</td>
<td>95.2%</td>
</tr>
<tr>
<td>TMHMM</td>
<td>160</td>
<td>1</td>
<td>106 (66.3%)</td>
<td>122 (76.3%)</td>
<td>95.4%</td>
<td>97.1%</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>2</td>
<td>120 (75.0%)</td>
<td>133 (83.1%)</td>
<td>96.8%</td>
<td>97.5%</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>3</td>
<td>123 (76.9%)</td>
<td>134 (83.8%)</td>
<td>97.1%</td>
<td>97.7%</td>
</tr>
<tr>
<td>MEMSAT</td>
<td>160</td>
<td></td>
<td>108 (67.5%)</td>
<td>118 (73.8%)</td>
<td>93.3%</td>
<td>95.6%</td>
</tr>
</tbody>
</table>

Correct topology: proteins for which all transmembrane segments and their orientations are correctly predicted.
Correct location: proteins for which all transmembrane segments and their orientations are correctly predicted, regardless of their orientation.
Single TM sensitivity: correctly predicted segments / true segment.
Single TM specificity: correctly predicted segments / total predicted segments.
Discussion

• The accuracy of the TMHMM is high
• Our HMM-based method embodies many conceptual and methodological aspects of previous methods.
• The main virtues
  – The model architecture maps closely to the biological system
  – Everything is done in the probabilistic framework of HMMs.
References


• Aik Choon Tan, David Gilbert “Machine Learning and its Application to Bioinformatics: An Overview”, Bioinformatics research center, university of Glasgow, August 31, 2001

• Protein structure http://home.postech.ac.kr/~smw1905/tema/structure/chapter2.htm

• Book: “Artificial Intelligence theory and practice”, Thomas dean
THANKS !