Biological Sequence Analysis
and Motif Discovery

Introductory Overview Lecture
Joint Statistical Meetings 2001, Atlanta

Jun Liu
Department of Statistics
Harvard University

http://www.fas.harvard.edu/~junliu
jliu@stat.harvard.edu
Topics to be covered

• Basic Biology: DNA, RNA, Protein; genetic code.

• Biological Sequence Analysis
  – Pairwise alignment --- dynamic programming
    • Needleman-Wunsch
    • Smith-Waterman
    • Blast …
  – Multiple sequence alignment
    • Heuristic approaches
    • The hidden Markov model
  – Motif finding in DNA and protein sequence
Central Paradigm of Bioinformatics

Genetic Information → Molecular Structure → Biochemical Function → Symptoms (Phenotype)

Courtesy of Doug Brutlag
Buliding Blocks of Biological Systems:  
*nucleotides and amino acids*

- **DNA** (nucleotides, 4 types): information carrier/encoder.
- **RNA**: bridge from DNA to protein.
- **Protein** (amino acids, 20 types): action molecules.
- **Genetic code**: deciphering genetic information.

```
ATGAATCGTA GGGTTTGAA CGCTGGCAAT ACGATGACTT CTCAAGCGAA
CATTGACGAC GCCAGCTGGA AGGCGGTCTC CGAGGGCGGA ......  
```

```
MNRRGLNAGNTMTSQANIDDGWSWKAVSEGG ......  
```
From DNA to Protein

The diagram illustrates the process of protein synthesis starting from DNA. The DNA in the nucleus is transcribed into mRNA, which then moves to the cytoplasm. The mRNA is translated by ribosomes, which use tRNAs to bring amino acids and assemble them into a growing protein chain. This process converts genetic information from DNA to functional proteins.
## Genetic Code

<table>
<thead>
<tr>
<th>First Position</th>
<th>Second Position of Codon</th>
<th>Third Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>TTT Phe [F]</td>
<td>CAT His [H]</td>
</tr>
<tr>
<td></td>
<td>TTC Phe [F]</td>
<td>CAC His [H]</td>
</tr>
<tr>
<td></td>
<td>TTA Leu [L]</td>
<td>CCA Pro [P]</td>
</tr>
<tr>
<td></td>
<td>TTG Leu [L]</td>
<td>CCG Pro [P]</td>
</tr>
<tr>
<td></td>
<td>CTT Leu [L]</td>
<td>CCT Pro [P]</td>
</tr>
<tr>
<td></td>
<td>CTC Leu [L]</td>
<td>CCC Pro [P]</td>
</tr>
<tr>
<td></td>
<td>CTA Leu [L]</td>
<td>CCA Pro [P]</td>
</tr>
<tr>
<td></td>
<td>CTG Leu [L]</td>
<td>CCG Pro [P]</td>
</tr>
<tr>
<td></td>
<td>ATT Ile [I]</td>
<td>ACT Thr [T]</td>
</tr>
<tr>
<td></td>
<td>ATC Ile [I]</td>
<td>ACC Thr [T]</td>
</tr>
<tr>
<td></td>
<td>ATA Ile [I]</td>
<td>ACA Thr [T]</td>
</tr>
<tr>
<td></td>
<td>ATG Met [M]</td>
<td>ACG Thr [T]</td>
</tr>
<tr>
<td></td>
<td>GTT Val [V]</td>
<td>GCT Ala [A]</td>
</tr>
<tr>
<td></td>
<td>GTC Val [V]</td>
<td>GCC Ala [A]</td>
</tr>
<tr>
<td></td>
<td>GTA Val [V]</td>
<td>GCA Ala [A]</td>
</tr>
<tr>
<td></td>
<td>GTG Val [V]</td>
<td>GCG Ala [A]</td>
</tr>
</tbody>
</table>

## Notes
- TTT, TTC, TTA, TTG code for Leu [L], and TAT, TAC, TAA, TAG code for Ter [end].
- CGT, CGC, CCG, CGG code for Arg [R].
- GGT, GGC, GCG, GGG code for Gly [G].
- GTA, GTC, GTT code for Val [V].
- GCT code for Ala [A].
- GAT code for Asp [D].
- GGC, GGG code for Gly [G].
- GAA, GAG code for Glu [E].
Main Resources for Data

- **NCBI** --- national center for biotechnology information.
  - GenBank maintenance
  - BLAST searching and servers
  - Entrez database
  - Taxonomy database
  - Structure database
  - Bankit and SEQUIN submission software

- **EMBL** --- European equivalent of NCBI.
  - Web access [http://www.embl-heidelberg.de](http://www.embl-heidelberg.de)
  - EBI --- outstation of EMBL. [http://www.ebi.ac.uk/](http://www.ebi.ac.uk/)
Finding “Patterns” in Biological Sequences
Everything is related to everything else in a logical way ....
A Motif
Brute-force String Search

A STRING SEARCHING EXAMPLE CONSISTING OF ...
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
STING
A STRING SEARCHING EXAMPLE CONSISTING OF ...
Boyer-Moore String Search
Consensus in Sequences

• A database for protein families and patterns:

Pattern: \([\text{LIVM}]-[\text{ST}]-\text{A}-[\text{STAG}]-\text{H}-\text{C}\)

Interpretation:

\(<\text{A}-x-[\text{ST}(2)]-x(0,1)-\text{V}-\{\text{LI}\}>\)

This pattern, which must be in the N-terminal of the sequence (\(<\)’), means:
Ala-any-[Ser or Thr]-[Ser or Thr]-
(any or none)-Val-(any but Leu & Ile)
Pattern Finding Programs

  - **ScanProsite** - Scan a sequence against PROSITE or a pattern against SWISS-PROT
  - **ProfileScan** - Scan a sequence against the profile entries in PROSITE

- **Motif web programs** ([http://motif.stanford.edu/identify/](http://motif.stanford.edu/identify/))
  - **IDENTIFY**
  - **SCAN**

- **GCG SEQWEB programs** (commercial)
  - **Stringssearch**
  - **Findpatterns**
  - **Motifs**
Pairwise Sequence Alignment Methods
The Sequence Alignment problem

Given a pair of sequences, how do we view them?

A better representation?

Or

What determines the choice? Gap & mismatch penalties
Sequence Alignment and Typical Objective Function

\[
\begin{array}{cccccc}
X & 220 & 230 & 240 & 250 & X \\
F & \text{---SGGNT}\text{HIYMNHVEQCKEILRREPKE}\text{LCELVISGLPYKF}\text{RYLSTKE-QLK-Y} \\
| & \vdots & \vdots & \vdots & \vdots & \vdots \\
GDFIHTLGDAHYLNHNIE\text{PLKIQLQREPRPRFPPKLRILRKE}\text{VKIDDFKAEDFQIEGY}N \\
X & 260 & 270 & 280 & 290 & X 
\end{array}
\]

\[
\text{Score} = \sum_{\text{Region Start}} \text{Region End} \left( \text{Similarity-weights} - \sum_{\text{Region Start}} \text{Penalties} \right)
\]

where:

\[
\text{Penalty} = \text{Gap-penalty} + \text{Size-of-gap} \times \text{Gap-size-penalty}
\]
Global: Needleman-Wunsch Algorithm

- Finding the optimal alignment via dynamic programming
- Example alignment

\[
\begin{align*}
\text{C} & \quad \text{A} & \quad \text{T} & \quad \text{T} & \quad \text{G} & \quad \ldots \\
\text{T} & \quad \text{C} & \quad \text{A} & \quad \text{T} & \quad \\
\text{A} & \quad \text{T} & \quad \text{C} & \quad \text{G} & \quad \\
\text{T} & \quad \text{G} & \quad \text{C} & \quad \text{A} & \quad \\
& \quad \ldots & \quad \ldots & \quad \ldots & \quad \ldots \\
\text{C} & \quad \text{A} & \quad \text{T} & \quad \text{T} & \quad \text{G} & \quad \\
\text{T} & \quad \text{C} & \quad \text{A} & \quad \text{T} & \quad \text{G} & \quad \\
\end{align*}
\]
Alignment Recursion

\[
F(i, j) = \max \left\{ F(i-1, j-1) + s(x_i, y_j), \quad F(i-1, j) - \gamma, \quad F(i, j-1) - \gamma \right\}
\]

E.g., \( s(x, y) = 2I_{\{x=y\}}, \quad \gamma = 1 \)

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>T</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>-3</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>-4</td>
<td>-1</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>G</td>
<td>-5</td>
<td>-2</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Substitution/Scoring Matrices

- Pam matrices (*Dayhoff et al. 1978*) --- phylogeny-based.

PAM1: expected number of mutation = 1%

PAM250 matrix, log-odds representation
**BLO**(ck)**SU**(bstitution)**M**(atrix) (Henikoff & Henikoff 1992)

- Derived from a set (2000) of aligned and ungapped regions from protein families; emphasizing more on chemical similarities (versus how easy it is to mutate from one residue to another). BLOSUM\(x\) is derived from the set of segments of \(x\)% identity.

### BLOSUM62 Matrix, log-odds representation

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

• **Linear score:** $\gamma_o(g) = -gd$
  
  – Typically: $d=8$, in unit of half-bits ($=4\log_2$)

• **Affine score:** $\gamma_o(g) = -d - (g-1)e$
  
  – $d$: gap opening penalty; $e$: gap extension penalty
  
  – Typical $d=12$, $e=2$ (in unit of half-bits)

• Gap penalty corresponds to log-probability of opening a gaps. For example, under the standard linear score, $P(g=k) = \exp(-dk) = 2^{-4k}$
Local Alignment:

**Smith-Waterman Algorithm**

\[
F(i, j) = \max \begin{cases} 
0 \\
F(i-1, j-1) + s(x_i, y_j) \\
F(i-1, j) - d \\
F(i, j-1) - d 
\end{cases}
\]
Sequence comparison and data base search

Similarity search (BLAST, FASTA)

Recognize by matching two sequences:

Query sequence

Target sequence
BLAST
(Altschul et al. 1990)

- Create a word list from the query;
  - word length = 3 for protein and 12 for DNA.
- For each listed word, find “neighboring words” (~ 50), $S(W, W') > T$
- For each sequence in the database, search exact matches to each word in the set.
- Extend the hits in both directions until score drops below $X$
- No gap allowed; use Karlin-Altschul statistics for significance
- New versions (>1.4) of BLAST gives gapped alignments.
- Compute Smith-Waterman for “significant” alignments
- BLASTP (protein), BLASTN (DNA), BLASTX (pr → DNA).
BLAST 2.0

- Two word hits must be found within a window of A residues in order to trigger extension
- Gapped extension from the middle of ungapped HSPs

- Position-specific iterative (PSI-) BLAST.
  - Profile constructed on the fly and iteratively refined.
  - Begin with a single query, profile constructed from those significant hits; use the profile to do another search, and iterate the procedure till “convergence”
A Bayesian Model for Pairwise Alignment

**Missing data --- Alignment matrix**

\[ A_{i,j} = 1 \text{ if residue } i \text{ of sequence 1 aligns with residue } j \text{ of sequence 2, } 0 \text{ otherwise.} \]

**Observed data pair of sequence } R^{(1)}, R^{(2)} \]

\[ P(R_i^{(1)}, R_j^{(2)} \mid A, \Theta) = \Theta_{R_i^{(1)}} + \Theta_{R_j^{(2)}} + A_{i,j} \Psi_{R_i^{(1)},R_j^{(2)}} \]

\[ \Psi_{R^1,R^2} \text{ PAM or Blosum} \]

\[ \sum_i A_{i,j} \leq 1 \quad \sum_j A_{i,j} \leq 1 \]
Multiple Alignment

How we develop Prosite patterns!
ClustalW Step 1: Distance Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hbb_Human</td>
<td></td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hbb_Horse</td>
<td>.17</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hba_Human</td>
<td>.59</td>
<td>.60</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hba_Horse</td>
<td>.59</td>
<td>.59</td>
<td>.13</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Myg_Phyca</td>
<td>.77</td>
<td>.77</td>
<td>.75</td>
<td>.75</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Glb5_Petma</td>
<td>.81</td>
<td>.82</td>
<td>.73</td>
<td>.74</td>
<td>.80</td>
<td>-</td>
</tr>
<tr>
<td>Lgb2_Luplu</td>
<td>.87</td>
<td>.86</td>
<td>.86</td>
<td>.88</td>
<td>.93</td>
<td>.90</td>
</tr>
</tbody>
</table>

Pairwise alignment: Calculate distance matrix

ClustalW Step 2: bBuild the Tree

Rooted Neighbor Joining tree (guide tree)
ClustalW Step 3: Progressive Alignment

Progressive alignment: Align following the guide tree
However ....

- No explicit model to guide for the alignment --- heuristic driven.
- Tree construction has problems.
- Overall, not sensitive enough for remotely related sequences.
The Hidden Markov Model

For given $z_s$, $y_s \sim f(y_s \mid z_s, \theta)$, and the $z_s$ follow a Markov process with transition $p_s(z_s \mid z_{s-1}, \phi)$.

$\pi_t(z_t) = p(z_1, \ldots, z_t \mid y_1, \ldots, y_t; \phi, \theta)$

“The State Space Model”
What Are Hidden in Sequence Alignment?

- **HMM Architecture**: transition diagram for the underlying Markov chain.
The *Path*

$$\delta_i = \# \text{ of deletions}$$

$$G_i = \begin{cases} 
0 & \text{if generated by insertion} \\
1 & \text{if generated by a match}
\end{cases}$$
Pfam alignment view (http://pfam.wustl.edu/browse.shtml)
More Restricted Models

(Liu et al. 99, Neuwald et al. 97)

- **Block Motif Propagation Model:** limit the total number of gaps but no deletions allowed.

  Motif 1    Motif 2    Motif 3    Motif 4
Finding Repetitive Patterns

Biological Sequence Analysis
Motif Alignment Model

**Motif**

\[ a_1 \quad \text{Motif} \quad a_2 \]

width = \( w \)

\[ a_k \]

length \( n_k \)

*The missing data:* Alignment variable: \( A=\{a_1, a_2, \ldots, a_k\} \)

- Every **non-site positions** follows a common multinomial with \( p_0=(p_{0,1}, \ldots, p_{0,20}) \)
- Every position \( i \) in the motif element follows probability distribution \( p_i=(p_{i,1}, \ldots, p_{i,20}) \)
The Algorithm

- Initialized by choosing random starting positions
  \[ a_1^{(0)}, a_2^{(0)}, \ldots, a_K^{(0)} \]

- Iterate the following steps many times:
  - Randomly or systematically choose a sequence, say, \textit{sequence k}, to exclude.
  - Carry out the \textit{predictive-updating} step to update \( a_k \)

- Stop when no more observable changes in likelihood

- Available Programs:
  - Gibbs motif sampler (http://www.wadsworth.org/res&res/bioinfo)
  - Bioprospector (http://bioprospector.stanford.edu)
1. Compute predictive frequencies of each position $i$ in motif

\[ c_{ij} = \text{count of amino acid type } j \text{ at position } i. \]
\[ c_{0j} = \text{count of amino acid type } j \text{ in all non-site positions.} \]
\[ q_{ij} = \frac{(c_{ij} + b_j)}{(K-1+B)}, \quad B = b_1 + \cdots + b_K \text{ “pseudo-counts”} \]

2. Sample from the predictive distribution of $a_k$.

\[ P(a_k = l + 1) \propto \prod_{i=1}^{w} \frac{q_{i,R_k(l+i)}}{q_{0,R_k(l+i)}} \]
Using MACAW
Idea 2: Mixture modeling

- View the dataset as a long sequence with $k$ motif types:

- **Idea**: partition the input sequence into segments that correspond to different (unknown) motif models.
- It is a mixture model (unsupervised learning).
- Implement a predictive updating scheme.
Special Case: Bernoulli Sampler

- **Sequence data:** \( R = r_1 r_2 r_3 \ldots \ldots r_N \)
- **Indicator variable:** \( \Delta = \delta_1 \delta_2 \delta_3 \ldots \ldots \delta_N \)

\[
\delta_i = \begin{cases} 
1, & \text{if it is the start of an element} \\
0, & \text{if not.} 
\end{cases}
\]

- **Likelihood:** \( \pi(R, \Delta \mid \Theta, \varepsilon) \), \( \varepsilon \) is the prior prob for \( \delta_i = 1 \)

- **Predictive Update:**

\[
\frac{\pi(\delta_k = 1 \mid \Delta_{[\sim k]}, R)}{\pi(\delta_k = 0 \mid \Delta_{[\sim k]}, R)} = \frac{\hat{\varepsilon}}{1 - \hat{\varepsilon}} \prod_{i=1}^{w} \left( \frac{\hat{p}_{i,r_{k+i-1}}}{\hat{p}_{0,r_{k+i-1}}} \right)
\]

parameter for the motif model
References (self)