Multiple Sequence Alignment (MSA)
BIOL 7711
Computational Bioscience

Biochemistry and Molecular Genetics
Computational Bioscience Program
Consortium for Comparative Genomics
University of Colorado School of Medicine
David.Pollock@uchsc.edu
www.EvolutionaryGenomics.com
Why a Hidden Markov Model?

- Data elements are often linked by a string of connectivity, a linear sequence
- Secondary structure prediction (Goldman, Thorne, Jones)
- CpG islands
- Models of exons, introns, regulatory regions, genes
- Mutation rates along genome
Why a Hidden Markov Model?

Complications?
- Insertion and deletion of states (indels)
- Long-distance interactions

Benefits
- Flexible probabilistic framework
  - E.g., compared to regular expressions
Multiple Sequence Alignment

- Generalize pairwise alignment of sequences to include more than two
- Looking at more than two sequences gives us much more information
  - Site-specific information
    - Sites are different!
    - E.g., which amino acids, coevolution
    - Process of change at a site
  - Evolutionary/phylogenetic relationships
    - Shorter branches, dissect compound indels
Sample MSA: cFOS

<table>
<thead>
<tr>
<th>Species</th>
<th>Sequence</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOS_RAT</td>
<td>MMFSGFNADYEASSSRCSSASPAGDLSYYHSPADSFS MSGPVNTQDFCADLSVSSANF</td>
<td>60</td>
</tr>
<tr>
<td>FOS_MOUSE</td>
<td>MMFSGFNADYEASSSRCSSASPAGDLSYYHSPADSFS MSGPVNTQDFCADLSVSSANF</td>
<td>60</td>
</tr>
<tr>
<td>FOS_CHICK</td>
<td>MMYQGFAGEYEAPSSRCSSASPAGDSLTYYPSPADSFS MSGPVNSQDFCTDLAVSSANF</td>
<td>60</td>
</tr>
<tr>
<td>FOSB_MOUSE</td>
<td>-MFQAFPGDYDS-GSRCSS-SPSAESQ--YLSSVDSFGSPPTAAASQE-CAGLGEMPGSF</td>
<td>54</td>
</tr>
<tr>
<td>FOSB_HUMAN</td>
<td>-MFQAFPGDYDS-GSRCSS-SPSAESQ--YLSSVDSFGSPPTAAASQE-CAGLGEMPGSF</td>
<td>54</td>
</tr>
<tr>
<td>FOS_RAT</td>
<td>IPTVTAISTSPDLQWLQPTLVSSLVSSVAPSO----------TRAPHYPGLPTPS-TGAYARAGVV</td>
<td>112</td>
</tr>
<tr>
<td>FOS_MOUSE</td>
<td>IPTVTAISTSPDLQWLQPTLVSSLVSSVAPSO----------TRAPHYPGLPTQS-AAGAYARAGMV</td>
<td>112</td>
</tr>
<tr>
<td>FOS_CHICK</td>
<td>VPTVTAISTSPDLQWLQPTLIISSVAPSO----------NRG-HPYGVFAAPAAYSPAVL</td>
<td>112</td>
</tr>
<tr>
<td>FOSB_MOUSE</td>
<td>VPTVTAITTSQDLQWLQPTLIISSMAQSQGQPPAQVDPYDMGTS----YSTPGLS</td>
<td>110</td>
</tr>
<tr>
<td>FOSB_HUMAN</td>
<td>VPTVTAITTSQDLQWLQPTLIISSMAQSQGQPPVVPVDPYDMGTS----YSTPGMS</td>
<td>110</td>
</tr>
<tr>
<td>FOS_RAT</td>
<td>KTMSGGRAQSIG-----------------------------------RRGKVQVELPSSPEEEERKRRRERRNKMAAA</td>
<td>152</td>
</tr>
<tr>
<td>FOS_MOUSE</td>
<td>KTVSGGRAQSIG-----------------------------------RRGKVQVELPSSPEEEERKRRRERRNKMAAA</td>
<td>152</td>
</tr>
<tr>
<td>FOS_CHICK</td>
<td>KAP-GGRGQSIG-----------------------------------RRGKVQVELPSSPEEEERKRRRERRNKMAAA</td>
<td>151</td>
</tr>
<tr>
<td>FOSB_MOUSE</td>
<td>AYSTGGASDSGQPSTSTTTSGPVSARPARPRPREEUTLTPSSPEEEERKRRVRRRERNLAAA</td>
<td>170</td>
</tr>
<tr>
<td>FOSB_HUMAN</td>
<td>GYSSGASGSGPPPSTSGTTSGPGPARPARPRPREEUTLTPSSPEEEERKRRVRRRERNLAAA</td>
<td>170</td>
</tr>
</tbody>
</table>
Optimal MSA

- Use Dynamic Programming?
- Optimal alignment algorithm exists, but is $O(2^n L^n)$
  - $n$ is the number of sequences
  - $L$ is the length of the longest sequence
- 10 sequences of length 100 take $2^{10}100^{10}$~$10^{23}$ operations, around 1 million years at 3GHz
- Exponential algorithms strike again
- So, approximation approaches?
Progressive MSA

- Start with pairwise alignments of closely related sequences
  - Add more distantly related sequences one at a time
  - Complexity proportional to $L^{2n}$
- Requires *a priori* assumptions about the phylogenetic relationships
  - Can be estimated from all pairwise comparisons
    - Unfortunate circularity to this approach
  - SATCHMO method tries to estimate both at once
- MSA score based on sum of pairwise scores
  - Can be weighted to reduce influence of similar sequences
Gaps in Progressive MSAs

How to score gaps in MSAs?

- Want to align gaps with each other over all sequences. A gap in a pairwise alignment that “matches” a gap in another pairwise alignment should cost less than introducing a totally new gap.
  - Possible that a new gap could be made to “match” an older one by shifting around the older pairwise alignment
  - Change gap penalty near conserved domains of various kinds (e.g. secondary structure, hydrophobic regions)

CLUSTALW2
http://www.ebi.ac.uk/Tools/clustalw2/
is the most widely used Progressive MSA program
Greedy Algorithms

- Best alignment of each new sequence to the existing alignment
  - Then never revisit the decision
- Even if changing an old decision (e.g. moving around the gaps in a previous alignment) could increase the score, this approach doesn't do it.
- Approach is called “greedy” because it uses the best solution at the current step, then moves on
  - Have to hope that the best solution to a part of the problem will be good solution for the whole problem
  - This is a common way to resolve exponential problems
Problems with Progressive MSA

- Depends *crucially* on the quality of the pairwise alignments, particularly among the closest matches (which are aligned first)
  - Small errors propagate to whole alignment
- There is no suitable resolution to the problem of *gap penalties over multiple sequences*
- Works reasonably well for closely related sequences.
  - Even then, *manual adjustments* are common
Iterative MSA Methods

Start with a reasonable approximation to the optimal MSA
  - e.g. by using a progressive method
  - Then “tweak” to improve it
  - Common CS idea, called “optimization”

Various optimization techniques tried
  - e.g., GAs and simulated annealing
  - Key is the scoring function for the whole MSA
  - Also, what steps (tweaks) to take that are likely to improve the score
Block Based Methods

- Start with short local alignments (blocks)
  - Then reduce the problem to aligning the regions between the blocks

- “Divide and conquer”
  - Another common CS approach to exponential problems

- How to find the blocks?
  - DALIGN (local alignment methods)
  - DCA (divide and conquer alignments)
  - Tmsa (identify patterns and use them to define blocks)
## Using Pseudocounts to Profile

<table>
<thead>
<tr>
<th>MSA</th>
<th>Counts</th>
<th>Add 1 Pseudocounts:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBB</td>
<td>A 2 0 1</td>
<td>A 3 1 2</td>
</tr>
<tr>
<td>ABC</td>
<td>B 1 4 1</td>
<td>B 2 5 2</td>
</tr>
<tr>
<td>ABD</td>
<td>C 1 0 1</td>
<td>C 2 1 2</td>
</tr>
<tr>
<td>CBA</td>
<td>D 0 0 1</td>
<td>D 1 1 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Profiles</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3</td>
<td>1 2 3</td>
</tr>
<tr>
<td>A</td>
<td>.5 0 .25</td>
<td>A .37 .12 .25</td>
</tr>
<tr>
<td>B</td>
<td>.25 1 .25</td>
<td>B .25 .63 .25</td>
</tr>
<tr>
<td>C</td>
<td>.25 0 .25</td>
<td>C .25 .12 .25</td>
</tr>
<tr>
<td>D</td>
<td>0 0 .25</td>
<td>D .12 .12 .25</td>
</tr>
</tbody>
</table>
MSA Databases

Once they have been calculated, they can be saved and shared

http://pfam.sanger.ac.uk

TigerFam: database of protein families curated for function, rather than homology
http://www.jcvi.org/cms/research/projects/tigrfams
Web sites offer multiple approaches to MSA

Interfaces to multiple programs
  - http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli

Main web-based MSA servers
  - ClustalW2
    (for proteins, see previous slide)
  - http://orangutan.math.berkeley.edu/fsa/
    (FSA: fast statistical alignment for genomic seqs)
  - http://www.charite.de/bioinf/strap/
    (structural alignments)
  - See course website for many more listings…
Protein motifs

Recall that local alignments can identify similar regions in non-homologous proteins.

These regions (sometimes called domains) often have shared structure and/or function.

Example: Zinc-finger DNA binding motif.

How to define them?

- Consensus sequence
- Regular expression
- Profile (probability for each amino acid at each position)
ProSite consensus sequences

Ribosomal protein L14 signature

**Description:**
Ribosomal protein L14 is one of the proteins from the large ribosomal subunit. In eubacteria, L14 is known to bind directly to the 23S rRNA. It belongs to a family of ribosomal proteins which, on the basis of sequence similarities [1], groups:

- Eubacterial L14.
- Algal and plant chloroplast L14.
- Cyanellae L14.
- Archaeobacterial L14.
- Yeast L17A.
- Mammalian L23.
- Caenorhabditis elegans L23 (B0336.10).
- Higher eukaryotes mitochondrial L14.
- Yeast mitochondrial Ynl138 (gene MFLP38).

L14 is a protein of 119 to 137 amino-acid residues. As a signature pattern, we selected a conserved region located in the C-terminal half of these proteins.

**Last update:**
November 1997 / Pattern and text revised.

**Technical section:**

**PROSITE method (with tools and information) covered by this documentation:**

**Consensus pattern:**

\[ QA \cdot [LIV]3 \cdot x(9,10) \cdot [DNS] \cdot G \cdot x(4) \cdot [FY] \cdot x(2) \cdot [NT] \cdot x(2) \cdot V \cdot [LIV] \]

**Sequences known to belong to this class detected by the pattern:**

ALL, except for pine L14 and for Acetabularia mitochondrial L14

**Other sequence(s) detected in Swiss-Prot:**

NONE

- Retrieve an alignment of Swiss-Prot true positive hits:
  - Clustal format, color, condensed view / Clustal format, color / Clustal format, plain text / Fasta format
- Retrieve the sequence logo from the alignment
- Taxonomic tree view of all Swiss-Prot/TrEMBL entries matching PS00049
- Retrieve a list of all Swiss-Prot/TrEMBL entries matching PS00049
- Scan Swiss-Prot/TrEMBL entries against PS00049
- view ligand binding statistics

**Matching PDB structures:** 1C04 1FFK 1GIY 1GS2 ... [ALL]
Recognizing ProSite patterns

- **L14 Ribosome pattern:**
  \([GA]-[LIV](3)\cdot x(9,10)-[DNS]-G\cdot x(4)-[FY]\cdot x(2)-[NT]\cdot x(2)-V-[LIV]\)

- **Some matching sequences:**
  1. GIIIACGHLIPQTNNGACRTYILNDRVV
  2. GVLLWQPKHCSNAADGAWAWFAATAAVL
  3. ALIVEANIIILSISGRATTFHATSAVI

- **ProSite patterns can be translated into regular expressions, although the bounded length patterns (e.g. \([LIV](3,5)\) are unwieldy to write down as regexps**
Regular expressions

Wide use in computer science. Basis of PERL language (see also BioPERL)

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG(AA</td>
<td>TT)GC</td>
</tr>
<tr>
<td>PXXP</td>
<td>Two Prolines separated by any two amino acids</td>
</tr>
<tr>
<td>^MSE</td>
<td>All peptide records that begin with MSE</td>
</tr>
<tr>
<td>[ST]X[VIL]$</td>
<td>PDZ binding sites. All matches must be at the end of a sequence.</td>
</tr>
<tr>
<td>AAAAAAAAAAN+AAAAAAA AAA</td>
<td>All records that contain two A(8) tracks separated by any number of bases.</td>
</tr>
<tr>
<td>[^C]CCCCCCCCC[^C]</td>
<td>All records that contain only C(8).</td>
</tr>
</tbody>
</table>

For proteins, a language like prosite patterns is more intuitive, but often equivalent.
Profiles

Rather than identifying only the “consensus” (i.e. most common) amino acid at a particular location, we can assign a probability to each amino acid in each position of the domain.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>C</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>D</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>E</td>
<td>0.4</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Applying a profile

Calculate score (probability of match) for a profile at each position in a sequence by multiplying individual probabilities.

Uses a “sliding window”

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.1</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>C</td>
<td>0.3</td>
<td>0.1</td>
<td>0.25</td>
</tr>
<tr>
<td>D</td>
<td>0.2</td>
<td>0.2</td>
<td>0.25</td>
</tr>
<tr>
<td>E</td>
<td>0.4</td>
<td>0.2</td>
<td>0.25</td>
</tr>
</tbody>
</table>

For sequence EACDC:
- EAC = .4 * .5 * .25 = .05
- ACD = .1 * .1 * .25 = .0025
- CDC = .3 * .2 * .25 = .015

To calculate a significance value, normalize by the probability of match to random sequence
Using motifs

- Great for annotating a sequence with no strong homologs
- INTERPRO is an uniform interface to many different motif methods and databases
  - ProSite
  - Prints (fingerprints = multiple motifs)
  - ProDom (like Pfam, but for domains)
  - SMART (mobile domains)
Interpro example

InterPro IPR001723 Steroid hormone receptor

<table>
<thead>
<tr>
<th>Accession</th>
<th>IPR001723 Str_hrmn_rcpt Matches: 1661 proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Family</td>
</tr>
<tr>
<td>Signatures</td>
<td>Database ID Name Proteins</td>
</tr>
<tr>
<td></td>
<td>PRINTS  PR00398 STRDHORMONER 166</td>
</tr>
<tr>
<td>Children</td>
<td>IPR002903 Retinoid X receptor IPR001728 Thyroid hormone receptor IPR003068 Transcription factor COUP IPR003069 Ecdysteroid receptor IPR003070 Orphan nuclear receptor IPR003074 Peroxisome proliferator-activated receptor IPR003078 Retinoic acid receptor IPR003079 Nuclear receptor ROR IPR012239 Oestrogen receptor</td>
</tr>
<tr>
<td>Contains</td>
<td>IPR000536 Nuclear hormone receptor, ligand-binding IPR001628 Nuclear hormone receptor, DNA-binding IPR008946 Nuclear receptor, ligand-binding IPR013629 Zinc finger, C4-type, C-terminal</td>
</tr>
<tr>
<td>Process</td>
<td>GO:0006355 regulation of transcription, DNA-dependent</td>
</tr>
<tr>
<td>Function</td>
<td>GO:0003677 DNA binding GO:0004879 ligand-dependent nuclear receptor activity</td>
</tr>
<tr>
<td>Component</td>
<td>GO:0005634 nucleus</td>
</tr>
</tbody>
</table>

Taxonomic coverage

Overlapping InterPro entries

<table>
<thead>
<tr>
<th>IPR001723</th>
<th>Numbers of overlapping proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPR000536</td>
<td>21 1640 489</td>
</tr>
<tr>
<td>IPR008946</td>
<td>24 1637 743</td>
</tr>
<tr>
<td>IPR018699</td>
<td>1603 58 1</td>
</tr>
</tbody>
</table>
InterPro example

Match the pattern to a protein

Q03181 Peroxisome proliferator-activated receptor delta (PPAR-delta) (PPAR- beta) (Nuclear hormone receptor 1)

More proteins

IPR000536 Nuclear hormone receptor, ligand-binding
IPR001628 Nuclear hormone receptor, DNA-binding
IPR001723 Steroid hormone receptor
IPR003074 Peroxisome proliferator-activated receptor
IPR003075 Peroxisome proliferator-activated receptor, beta
IPR008946 Nuclear receptor, ligand-binding

ModBase
CATH Domain
SCOP Domain
PDB Chain
How do we create motifs?

- General problem of inducing patterns from sequences is difficult
  - Classic language result (Gold)
    - Context-free grammars cannot be induced from only positive examples
  - Many patterns are compatible with any MSA
    - How to decide which elements are required?
- In general, we need positive examples (in the class) but also “near misses” sequences that are similar but not members of the class
- Not absolutely true for protein sequences
Finding Consensus Sequences

- Based on local MSAs
- ProSite consensus built from MSA on (Amos Bairoch's) biological intuition, tweaked by calculating sensitivity and specificity of the patterns over SwissProt
- True (False) positives defined by Bairoch's understanding
- Not an automatable procedure!
Creating profiles

- Given a local MSA, creating a profile is straightforward
  - Calculate frequency of each amino acid at each position
- What about zero frequencies?
  - Could be sampling errors, not real zero probabilities
  - Zero probabilities always make zero scores!
- Regularization
  - Pseudocounts
  - Dirichlet mixtures (blend in background frequencies)
Better regularizers

- Add one pseudocount is too large and too uniform in the small MSA case
- Instead, can add a fraction proportional to the overall frequency of occurrence of the amino acid
- Might want to add different pseudocounts depending on the actual count (add more to smaller counts, especially 0)
- Can use substitution matrices to estimate
Feature alphabets

- Amino acids can be grouped by their characteristics:
  - Size, hydrophobicity, ionizability, etc.
  - An amino acid is generally in more than one group
- Can set different regularizers (pseudocounts) for each different feature
- Most useful when there are multiple features
  - otherwise many amino acids get same pseudocount
Dirichlet mixture priors

- Fanciest (and near optimal) regularizer
- Allows several dimensions (like a feature, but not predefined), each of which has a different weight for each amino acid.
- Each pseudocount depends on the prior probability of seeing a particular distribution in a position
  - Add more “prior” to more unusual observations
- Pseudocount falls off with more observations
Alternative Motif Generation

- Finding fixed (but sparse) patterns
  - **IBM SPLASH**
    - Looks for occurrences of N of M letters of a word
    - Uses hashes to look at all words up to a fixed size
    - Empirical estimates of significance of matches.

- Probabilistic E/M search
  - **MEME**
    - Uses prior likelihood function to focus search in most promising parts of the space
    - Principled estimates of significance
    - More about this later...

- Hidden Markov Models
  - Next week!
Profiles: an Example

A 0.1
C 0.05
D 0.2
E 0.08
F 0.01

Gap

A 0.04
C 0.1
D 0.01
E 0.2
F 0.02

Gap

A 0.2
C 0.01
D 0.05
E 0.1
F 0.06

continue
insert
continue
insert
continue
insert
delete
Profiles, an Example: States

State #1

A  .1
C  .05
D  .2
E  .08
F  .01

State #2

Gap

A  .04
C  .1
D  .01
E  .2
F  .02

State #3

Gap

A  .2
C  .01
D  .05
E  .1
F  .06

delete
continue
continue
insert
insert
insert
Profiles, an Example: Emission

Sequence Elements (possibly emitted by a state)

State #1
- A: 0.1
- C: 0.05
- D: 0.2
- E: 0.08
- F: 0.01

State #2
- Gap
- A: 0.04
- C: 0.1
- D: 0.01
- E: 0.2
- F: 0.02

State #3
- Gap
- A: 0.2
- C: 0.01
- D: 0.05
- E: 0.1
- F: 0.06

Arrows indicate transitions and emission probabilities.
Profiles, an Example: Emission

Sequence Elements (possibly emitted by a state)

State #1
- A: 0.1
- C: 0.05
- D: 0.2
- E: 0.08
- F: 0.01

State #2
- Gap
- A: 0.04
- C: 0.1
- D: 0.01
- E: 0.2
- F: 0.02

State #3
- Gap
- A: 0.2
- C: 0.01
- D: 0.05
- E: 0.1
- F: 0.06

Emission Probabilities

continue
insert
delete
Profiles, an Example: Arcs

State #1

A .1
C .05
D .2
E .08
F .01

Gap

insert

transition

continue

insert

dele

State #2

A .04
C .1
D .01
E .2
F .02

Gap

State #3

A .2
C .01
D .05
E .1
F .06

continue

insert

continue
Profiles, an Example: Special States

**Self => Self Loop**

State #1:
- A: 0.1
- C: 0.05
- D: 0.2
- E: 0.08
- F: 0.01

State #2:
- A: 0.04
- C: 1.0
- D: 0.01
- E: 0.2
- F: 0.02

State #3:
- A: 0.2
- C: 0.01
- D: 0.05
- E: 0.1
- F: 0.06

**No Delete “State”**

- Transition: insert
- Loop: continue
- Delete: self-loop
A Simpler not very Hidden MM
Nucleotides, no Indels, Unambiguous Path

\[ P(D | M) = 0.7 \times 1.0 \times 0.4 \times 1.0 \times 0.3 \times 1.0 \]
A Simpler not very Hidden MM
Nucleotides, no Indels, Unambiguous Path

\[
\ln P(D \mid M) = \sum_{\text{states}} \ln P(E_D \mid \text{state}) + \sum_{\text{arcs}} \ln P(x \rightarrow y)
\]
A Toy not-Hidden MM
Nucleotides, no Indels, Unambiguous Path
All arcs out are equal

Example sequences: GATC ATC GC GAGAGC AGATTTC

\[ P(AGATTTC \mid M) = (0.5 \times 1.0)^{l=7} \]
A Simple HMM
CpG Islands; States are Really Hidden Now

\[
P(state_y^i \mid D < i) = \sum_x P(state_{x}^{i-1}) \times P(x \rightarrow y) \times P(E_D \mid state_y^i)
\]
The Forward Algorithm
Probability of a Sequence is the Sum of All Paths that Can Produce It

Non-CpG

CpG

G .3
C .3
A .2
T .2

G .1
C .1
A .4
T .4

G .3

.3*(.3*.8+ .1*.1) = .075

G .1

.1*(.3*.2+ .1*.9) = .015
The Forward Algorithm
Probability of a Sequence is the Sum of All Paths that Can Produce It

CpG

G  .3
C  .3
A  .2
T  .2

Non-CpG

G  .1
C  .1
A  .4
T  .4

\[ \begin{align*}
G &: 0.3 \\
C &: 0.3 \\
A &: 0.2 \\
T &: 0.2 \\
\end{align*} \]

\[ \begin{align*}
G &: 0.1 \\
C &: 0.1 \\
A &: 0.4 \\
T &: 0.4 \\
\end{align*} \]

\[ \begin{align*}
G &: 0.3 \times (0.3 \times 0.8 + 0.1 \times 1) = 0.075 \\
C &: 0.3 \times (0.075 \times 0.8 + 0.015 \times 1) = 0.0185 \\
G &: 0.1 \times (0.3 \times 0.2 + 0.1 \times 0.9) = 0.015 \\
C &: 0.1 \times (0.075 \times 0.2 + 0.015 \times 0.9) = 0.0029 \\
\end{align*} \]
The Forward Algorithm
Probability of a Sequence is the Sum of All Paths that Can Produce It

```
G: 0.3
C: 0.3
A: 0.2
T: 0.2
```

```
G: 0.1
C: 0.1
A: 0.4
T: 0.4
```

CpG

```
G: 0.8
0.2
0.1
0.9
```

Non-CpG

```
G: 0.3
C: 0.3
A: 0.2
T: 0.2
```

\[
\begin{align*}
G & \rightarrow 0.3 \times (0.3 \times 0.8 + 0.1 \times 0.1) = 0.075 \\
C & \rightarrow 0.1 \times (0.1 \times 0.9) = 0.015 \\
A & \rightarrow 0.4 \times (0.3 \times 0.2 + 0.1 \times 0.9) = 0.0029 \\
T & \rightarrow 0.4 \times (0.4 \times 0.2 + 0.1 \times 0.9) = 0.0011
\end{align*}
\]
The Forward Algorithm

Probability of a Sequence is the Sum of All Paths that Can Produce It

- **CpG**
  - G: 0.3
  - C: 0.3
  - A: 0.2
  - T: 0.2

- **Non-CpG**
  - G: 0.1
  - C: 0.1
  - A: 0.4
  - T: 0.4

\[
\text{Probability} = 0.3 \times (0.3 \times 0.8 + 0.1 \times 0.1) = 0.075
\]

\[
\text{Probability} = 0.2 \times (0.3 \times 0.2 + 0.1 \times 0.9) = 0.015
\]

\[
\text{Probability} = 0.4 \times (0.4 \times 0.8 + 0.1 \times 0.9) = 0.0029
\]

\[
\text{Probability} = 0.2 \times (0.2 \times 0.8 + 0.1 \times 0.1) = 0.003
\]

\[
\text{Probability} = 0.4 \times (0.4 \times 0.2 + 0.1 \times 0.9) = 0.0011
\]
The Viterbi Algorithm
Most Likely Path

CpG

G .3
C .3
A .2
T .2

Non-CpG

G .1
C .1
A .4
T .4

0.8
0.2
0.1
0.9

G .3
C .3
G .1
A .4
A .4

0.3\textbullet m(0.3\textbullet 0.8, 0.1\textbullet 0.1) = 0.072
0.1\textbullet m(0.3\textbullet 0.2, 0.1\textbullet 0.9) = 0.009
0.3\textbullet m(0.075\textbullet 0.8, 0.015\textbullet 0.1) = 0.0173
0.1\textbullet m(0.075\textbullet 0.2, 0.015\textbullet 0.9) = 0.0014
0.2\textbullet m(0.0185\textbullet 0.8, 0.0029\textbullet 0.1) = 0.0028
0.4\textbullet m(0.0185\textbullet 0.2, 0.0029\textbullet 0.9) = 0.0014
0.2\textbullet m(0.003\textbullet 0.8, 0.0025\textbullet 0.1) = 0.0044
0.4\textbullet m(0.003\textbullet 0.2, 0.0025\textbullet 0.9) = 0.0050
Forwards and Backwards

Probability of a State at a Position

CpG

\[
\begin{align*}
G &= 0.3 \\
C &= 0.3 \\
A &= 0.2 \\
T &= 0.2 \\
\end{align*}
\]

Non-CpG

\[
\begin{align*}
G &= 0.1 \\
C &= 0.1 \\
A &= 0.4 \\
T &= 0.4 \\
\end{align*}
\]
Forwards and Backwards
Probability of a State at a Position

\[ P(CpG \mid i = 4, D) \]

\[ = \frac{P(CpG)}{[P(CpG) + P(not - CpG)]} \]

\[ = \frac{0.0007}{0.0007 + 0.0009} = 0.432 \]
Homology HMM

- Gene recognition, identify distant homologs

Common Ancestral Sequence
- Match, site-specific emission probabilities
- Insertion (relative to ancestor), global emission probs
- Delete, emit nothing
- Global transition probabilities
Homology HMM
Homology HMM

Uses

- Score sequences for match to HMM
- Compare alternative models
- Alignment
- Structural alignment
Multiple Sequence Alignment HMM

- Defines predicted homology of positions (sites)
  - Recognize region within longer sequence
  - Model domains or whole proteins
- Can modify model for sub-families
- Ideally, use phylogenetic tree
  - Often not much back and forth
  - Indels a problem
Model Comparison

Based on

- For ML, take
  
  Usually to avoid numeric error

- For heuristics, “score” is

- For Bayesian, calculate

\[
P(D \mid \theta, M) = \frac{P(D \mid \theta, M) \cdot P(\theta) \cdot P(M)}{\sum P(D \mid \theta, M) \cdot P(\theta) \cdot P(M)}
\]
Parameters, $\theta$

- Types of parameters
  - Amino acid distributions for positions
  - Global AA distributions for insert states
  - Order of match states
  - Transition probabilities
  - Tree topology and branch lengths
  - Hidden states (integrate or augment)

- Wander parameter space (search)
  - Maximize, or move according to posterior probability (Bayes)
Expectation Maximization (EM)

- Classic algorithm to fit probabilistic model parameters with unobservable states

- Two Stages
  - Maximize
    - If know hidden variables (states), maximize model parameters with respect to that knowledge
  - Expectation
    - If know model parameters, find expected values of the hidden variables (states)

- Works well even with e.g., Bayesian to find near-equilibrium space
Homology HMM EM

- Start with heuristic (e.g., ClustalW)
- Maximize
  - Match states are residues aligned in most sequences
  - Amino acid frequencies observed in columns
- Expectation
  - Realign all the sequences given model
- Repeat until convergence
- Problems: Local, not global optimization
  - Use procedures to check how it worked
Model Comparison

Determining significance depends on comparing two models

- Usually null model, $H_0$, and test model, $H_1$
- Models are nested if $H_0$ is a subset of $H_1$
- If not nested
  - Akaike Information Criterion (AIC) [similar to empirical Bayes] or
  - Bayes Factor (BF) [but be careful]

Generating a null distribution of statistic

- Z-factor, bootstrapping, $\chi^2_n$ parametric
- bootstrapping, posterior predictive
Z Test Method

- **Database of known negative controls**
  - E.g., non-homologous (NH) sequences
  - Assume NH scores \( \sim N(\mu, \sigma) \)
    - i.e., you are modeling known NH sequence scores as a normal distribution
  - Set appropriate significance level for multiple comparisons (more below)

**Problems**

- Is homology certain?
- Is it the appropriate null model?
  - Normal distribution often not a good approximation
- Parameter control hard: e.g., length distribution
Bootstrapping and Parametric Models

- Random sequence sampled from the same set of emission probability distributions
  - Same length is easy
  - Bootstrapping is re-sampling columns
  - Parametric uses estimated frequencies, may include variance, tree, etc.
    - More flexible, can have more complex null
    - Pseudocounts of global frequencies if data limit

- Insertions relatively hard to model
  - What frequencies for insert states? Global?
Homology HMM Resources

- UCSC (Haussler)
  - SAM: align, secondary structure predictions, HMM parameters, etc.

- WUSTL/Janelia (Eddy)
  - Pfam: database of pre-computed HMM alignments for various proteins
  - HMMer: program for building HMMs
Increasing Asymmetry with Increasing Single Strandedness

e.g., \( P ( A\Rightarrow G) = c + \tau \)

\[ \tau = (D_{ssH} \ast \text{Slope}) + \text{Intercept} \]
2x Redundant Sites
4x Redundant Sites

![Graph showing relative substitution rate (A-->G) over time spent single-stranded for high, low, and complete states.](image-url)
Beyond HMMs

- Neural nets
- Dynamic Bayesian nets
- Factorial HMMs
- Boltzmann Trees
- Kalman filters
- Hidden Markov random fields
COI Functional Regions

$O_2 + \text{protons} + \text{electrons} = H_2O + \text{secondary proton pumping (ATP)}$