Review: Pair HMMs

- Consider this special case:

  \[ V_M(i, j) = P(x_i, y_j) \max \]
  \[ V_I(i, j) = Q(x_i) \max \]
  \[ V_J(i, j) = Q(y_j) \max \]
  \[ (1 - 2\delta) V_M(i - 1, j - 1) \]
  \[ (1 - \epsilon) V_I(i - 1, j - 1) \]
  \[ (1 - \epsilon) V_J(i - 1, j - 1) \]
  \[ \delta V_M(i - 1, j) \]
  \[ \epsilon V_I(i - 1, j) \]
  \[ \delta V_M(i, j - 1) \]
  \[ \epsilon V_J(i, j - 1) \]

- Similar for **forward/backward** algorithms
  - (see Durbin et al for details)

**QUESTION:** What’s the computational complexity of DP?
Connection to NW with affine gaps

\[ V_M(i, j) = \frac{P(x_i, y_j)}{Q(x_i) Q(y_j)} \max \]

\[ \begin{cases} 
(1 - 2\delta) V_M(i - 1, j - 1) \\
(1 - \varepsilon) V_i(i - 1, j - 1) \\
(1 - \varepsilon) V_j(i - 1, j - 1) 
\end{cases} \]

\[ V_i(i, j) = \max \begin{cases} 
\delta V_M(i - 1, j) \\
\varepsilon V_i(i - 1, j) 
\end{cases} \]

\[ V_j(i, j) = \max \begin{cases} 
\delta V_M(i, j - 1) \\
\varepsilon V_j(i, j - 1) 
\end{cases} \]

- Account for the extra terms “along the way.”
Connection to NW with affine gaps

\[
\log V_M(i, j) = \log \frac{P(x_i, y_i)}{Q(x_i) Q(y_i)} + \max \left\{ \log (1 - 2\delta) + \log V_M(i - 1, j - 1), \log (1 - \epsilon) + \log V_I(i - 1, j - 1), \log (1 - \epsilon) + \log V_J(i - 1, j - 1) \right\}
\]

\[
\log V_I(i, j) = \max \left\{ \log \delta + \log V_M(i - 1, j), \log \epsilon + \log V_I(i - 1, j) \right\}
\]

\[
\log V_J(i, j) = \max \left\{ \log \delta + \log V_M(i, j - 1), \log \epsilon + \log V_J(i, j - 1) \right\}
\]

- Take logs, and ignore a couple terms.
Connection to NW with affine gaps

\[ M(i, j) = S(x_i, y_j) + \max \left\{ \begin{array}{l}
M(i - 1, j - 1) \\
I(i - 1, j - 1) \\
J(i - 1, j - 1)
\end{array} \right. \]

\[ I(i, j) = \max \left\{ \begin{array}{l}
d + M(i - 1, j) \\
e + I(i - 1, j)
\end{array} \right. \]

\[ J(i, j) = \max \left\{ \begin{array}{l}
d + M(i, j - 1) \\
e + J(i, j - 1)
\end{array} \right. \]

• Rename!
Conditional random fields
Recall Likelihood $P(x, \pi)$

$P(x, \pi) = P(x_1, \ldots, x_N, \pi_1, \ldots, \pi_N) = a_{0\pi_1} a_{\pi_1\pi_2} \cdots a_{\pi_{N-1}\pi_N} e_{\pi_1}(x_1) \cdots e_{\pi_N}(x_N)$

- Enumerate all parameters $a_{ij}$ and $e_i(b)$; $n$ params
  
  $a_{0\text{Fair}} : \theta_1$; $a_{0\text{Loaded}} : \theta_2$; \ldots $e_{\text{Loaded}}(6) = \theta_{18}$

- Count the # of times each parameter $j = 1, \ldots, n$ occurs

$F(j, x, \pi) = \# \text{ parameter } \theta_j \text{ occurs in } (x, \pi)$

- (call $F(\cdot, \cdot, \cdot)$ the feature counts) Then,

$$P(x, \pi) = \prod_{j=1\ldots n} \theta_j^{F(j, x, \pi)} = \exp\left[\sum_{j=1\ldots n} \log(\theta_j) \times F(j, x, \pi)\right]$$
Conditional random fields - Recap

• Definition

\[
P(\pi \mid x) = \frac{\exp(\sum_{i=1}^{\lfloor x \rfloor} w^T F(\pi_i, \pi_{i-1}, x, i))}{\sum_{\pi'} \exp(\sum_{i=1}^{\lfloor x \rfloor} w^T F(\pi'_i, \pi'_{i-1}, x, i))}
\]

where

- \( F : (\text{state, state, observations, index}) \rightarrow \mathbb{R}^n \) “local feature mapping”
- \( w \in \mathbb{R}^n \) “parameter vector”

- Summation over all possible state sequences \( \pi'_1 \ldots \pi'_{\lfloor x \rfloor} \)
- \( a^T b \) for vectors \( a, b \in \mathbb{R}^n \) denotes inner product, \( \sum_{i=1}^{n} a_i b_i \)
For each component $w_j$, define $F_j$ to be a 0/1 indicator variable of whether the $j^{th}$ parameter should be included in scoring $x$, $\pi$ at position $i$:

$$w = \begin{bmatrix} 
\log a_{0}(1) \\
... \\
\log a_{0}(K) \\
\log a_{11} \\
... \\
\log a_{KK} \\
\log e_{1}(b_{1}) \\
... \\
\log e_{K}(b_{M})
\end{bmatrix} \in \mathbb{R}^{n}$$

$$F(\pi_{i}, \pi_{i-1}, x, i) = \begin{bmatrix} 
1\{i = 1 \land \pi_{i-1} = 1\} \\
... \\
1\{i = 1 \land \pi_{i-1} = K\} \\
1\{\pi_{i-1} = 1 \land \pi_{i} = 1\} \\
... \\
1\{\pi_{i-1} = K \land \pi_{i} = K\} \\
1\{x_{i} = b_{1} \land \pi_{i} = 1\} \\
... \\
1\{x_{i} = b_{M} \land \pi_{i} = K\}
\end{bmatrix} \in \mathbb{R}^{n}$$

Then, $\log P(x, \pi) = \sum_{i=1}^{\mid x \mid} w^{T} F(\pi_{i}, \pi_{i-1}, x, i)$
CRFS ≥ HMMs (continued)

- In an HMM, our features were of the form

\[ F(\pi_i, \pi_{i-1}, x, i) = F(\pi_i, \pi_{i-1}, x_i, i) \]

  - i.e., when scoring position \( i \) in the sequence, feature only considered the emission \( x_i \) at position \( i \).
  - Cannot look at other positions (e.g., \( x_{i-1}, x_{i+1} \)) since that would involve “emitting” a character more than once – double-counting of probability

- CRFs don’t have this restriction
  - Why? Because CRFs don’t attempt to model the observations \( x \)!
3 basic questions for CRFs

- **Evaluation:** Given a sequence of observations $x$ and a sequence of states $\pi$, compute $P(\pi \mid x)$

- **Decoding:** Given a sequence of observations $x$, compute the maximum probability sequence of states $\pi_{ML} = \arg \max_{\pi} P(\pi \mid x)$

- **Learning:** Given a CRF with unspecified parameters $w$, compute the parameters that maximize the likelihood of $\pi$ given $x$, i.e., $w_{ML} = \arg \max_{w} P(\pi \mid x, w)$
Viterbi for CRFs

- **Note that:**
  \[
  \arg\max_{\pi} P(\pi \mid x) = \arg\max_{\pi} \frac{\exp\left(\sum_{i=1}^{\mid x \mid} w^T_F(\pi_i, \pi_{i-1}, x, i)\right)}{\sum_{\pi'} \exp\left(\sum_{i=1}^{\mid x \mid} w^T_F(\pi'_i, \pi'_{i-1}, x, i)\right)}
  \]
  \[
  = \arg\max_{\pi} \exp\left(\sum_{i=1}^{\mid x \mid} w^T_F(\pi_i, \pi_{i-1}, x, i)\right)
  \]
  \[
  = \arg\max_{\pi} \sum_{i=1}^{\mid x \mid} w^T_F(\pi_i, \pi_{i-1}, x, i)
  \]

- **We can derive the following recurrence:**
  \[
  V_k(i) = \max_j \left[ w^T_F(k, j, x, i) + V_j(i-1) \right]
  \]

- **Notes:**
  - Even though the features may depend on arbitrary positions in $x$, $x$ is constant. DP depends only on knowing the previous state
  - Computing the partition function (denominator) can be done by a similar adaptation of the forward/backward algorithms
Given that we end up in state $k$ at step $i$, maximize score to the left and right.
Given that we end up in state k at step i, maximize score to the left and right.

X is fixed: => parse to the left of step i, given we end in state k, does not affect parse to the right of step i.
Learning CRFs

- Key observation: \( -\log P(\pi | x, w) \) is a differentiable, convex function of \( w \)

Any local minimum is a global minimum.
Learning CRFs (continued)

- Compute partial derivative of $\log P(\pi \mid x, w)$ with respect to each parameter $w_j$, and use the gradient ascent learning rule:

  Gradient points in the direction of greatest function increase
The CRF gradient

- It turns out that

\[
\frac{\partial}{\partial w_j} \log P(\pi | x, w) = F_j(x, \pi) - E_{\pi' \sim P(\pi' | x, w)} [ F_j(x, \pi') ]
\]

- This has a very nice interpretation:
  - We increase parameters for which the correct feature values are greater than the predicted feature values
  - We decrease parameters for which the correct feature values are less than the predicted feature values

- This moves probability mass from incorrect parses to correct parses
DNA Sequencing
DNA sequencing

How we obtain the sequence of nucleotides of a species

...ACGTGACTGAGGACCGTG
CGACTGAGACTGACTGGGT
CTAGCTAGACTACGTTTTA
TATATATATACGTCGTCGT
ACTGATGACTAGATTACAG
ACTGATTTAGATACCTGAC
TGATTTTAAAAAAATATT...
Human Genome Project

1990: Start

2000: Bill Clinton:

2001: Draft

2003: Finished

3 billion basepairs

$3 billion

“most important scientific discovery in the 20th century”

now what?
Which representative of the species?

Which human?

Answer one:

Answer two: it doesn’t matter

Polymorphism rate: number of letter changes between two different members of a species

Humans: ~1/1,000

Other organisms have much higher polymorphism rates

- Population size!
Why humans are so similar

A small population that interbred reduced the genetic variation

Out of Africa ~ 40,000 years ago

Heterozygosity: \( H \)

\[ H = \frac{4Nu}{1 + 4Nu} \]

\( u \sim 10^{-8}, N \sim 10^4 \)

\[ \Rightarrow H \sim 4 \times 10^{-4} \]
There is never “enough” sequencing

100 million species

Somatic mutations (e.g., HIV, cancer)

Sequencing is a functional assay

7 billion individuals
Sequencing Growth

Cost of one human genome

- 2004: $30,000,000
- 2008: $100,000
- 2010: $10,000
- 2015: $1,000
- ???: $300

How much would you pay for a smartphone?
Ancient sequencing technology – Sanger Vectors

Vector
Circular genome (bacterium, plasmid)

+ =

Known location
(restriction site)
Ancient sequencing technology – Sanger Gel Electrophoresis

1. Start at primer (restriction site)
2. Grow DNA chain
3. Include dideoxynucleoside (modified a, c, g, t)
4. Stops reaction at all possible points
5. Separate products with length, using gel electrophoresis
Fluorescent Sanger sequencing trace

Lane signal

(Real fluorescent signals from a lane/capillary are much uglier than this).

A bunch of magic to boost signal/noise, correct for dye-effects, mobility differences, etc, generates the ‘final’ trace (for each capillary of the run)

Trace
Making a Library (present)

- Shear to ~500 bases
- Put on linkers
- eventual forward and reverse sequence
- PCR to obtain preparative quantities
- Size selection on preparative gel
- Final library (~600 bp incl linkers) after size selection

Slide Credit: Arend Sidow
Library

- Library is a massively complex mix of -initially- individual, unique fragments

- Library amplification mildly amplifies each fragment to retain the complexity of the mix while obtaining preparative amounts
  - (how many-fold do 10 cycles of PCR amplify the sample?)
Fragment vs Mate pair (‘jumping’)

(Illumina has new kits/methods with which mate pair libraries can be built with less material)
Illumina cluster concept

5. Denature the double stranded molecules

6. Completion of amplification
   On completion, several million dense clusters of double stranded DNA are generated in each channel of the flow cell

Repeat cycles of solid-phase bridge amplification

Slide Credit: Arend Sidow
Cluster generation (‘bridge amplification’)
Clonally Amplified Molecules on Flow Cell
Reversible Terminators

**Detection**

1. Incorporate DNA with the fluorophore and cleavage site.
2. Deblock and cleave off the dye to form a free 3' end.
3. Ready for the next cycle.
Sequencing by Synthesis, One Base at a Time

Cycle 1: Add sequencing reagents
First base incorporated
Remove unincorporated bases
Detect signal

Cycle 2-n: Add sequencing reagents and repeat
Sequencing power for every scale.

Find the sequencing system that’s right for your lab.

Compare key specifications across the whole portfolio of Illumina sequencing systems. Understand the differences between the MiniSeq, MiSeq, NextSeq, HiSeq, and HiSeq X Series.

<table>
<thead>
<tr>
<th></th>
<th>MiniSeq System</th>
<th>MiSeq Series</th>
<th>NextSeq Series</th>
<th>HiSeq Series</th>
<th>HiSeq X Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Output</td>
<td>7.5 Gb</td>
<td>15 Gb</td>
<td>120 Gb</td>
<td>1500 Gb</td>
<td>1800 Gb</td>
</tr>
<tr>
<td>Maximum Reads per Run</td>
<td>25 million</td>
<td>25 million†</td>
<td>400 million</td>
<td>5 billion</td>
<td>6 billion</td>
</tr>
<tr>
<td>Maximum Read Length</td>
<td>2 × 150 bp</td>
<td>2 × 300 bp</td>
<td>2 × 150 bp</td>
<td>2 × 150 bp</td>
<td>2 × 150 bp</td>
</tr>
<tr>
<td>Run Time</td>
<td>4–24 hours</td>
<td>4–55 hours</td>
<td>12–30 hours</td>
<td>&lt;1–3.5 days (HiSeq 3000/HiSeq 4000) 7 hours–6 days (HiSeq 2500)</td>
<td>&lt;3 days</td>
</tr>
<tr>
<td>Benchtop Sequencer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>System Versions</td>
<td>• MiniSeq System for low-throughput targeted DNA and RNA sequencing</td>
<td>• MiSeq System for targeted and small genome sequencing</td>
<td>• NextSeq 500 System for everyday genomics</td>
<td>• HiSeq 3000 System for production-scale whole-genome sequencing</td>
<td>• HiSeq X Five System for production-scale whole-genome sequencing</td>
</tr>
<tr>
<td></td>
<td>• MiSeq FGx System for forensic genomics</td>
<td>• MiSeq System for everyday genomics</td>
<td>• NextSeq 550 System for both sequencing and cytogenomic arrays</td>
<td>• HiSeq 4000 Systems for production-scale genomics</td>
<td>• HiSeq X Ten System for population-scale whole-genome sequencing</td>
</tr>
<tr>
<td></td>
<td>• MiSeqDx System for molecular diagnostics</td>
<td>• HiSeq 2500 Systems for large-scale genomics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pacific Biosciences SMRT technology

The SMRT Sequencing advantage
SMRT Sequencing is ideal for a variety of research applications and offers many benefits, including:

- Longest average read lengths
- Highest consensus accuracy
- Uniform coverage
- Simultaneous epigenetic characterization
- Single-molecule resolution

An overview of SMRT Sequencing
Overview of SMRT Technology

Each of the four nucleotides is labeled with a different colored fluorophore
Oxford Nanopore

MinION
Portable, real-time biological analyses

MinION is a portable device for molecular analyses that is driven by nanopore technology. It is adaptable for the analysis of DNA, RNA, proteins or small molecules with a straightforward workflow. The MinION product specification is available here.

Simple workflows

Sample preparation

PromethION

Analyses

Simple sample preparation (Coming soon: automated sample preparation from Voltrax)

Pocket-sized MinION for analysis anywhere

Desktop PromethION for high throughput analysis

Real time analysis solutions from Metricor

Learn about Voltrax

Learn about MinION

Learn about PromethION

Learn about Metricor
Method to sequence longer regions

Get one or two reads from each segment

~900 bp

Genomic segment

cut many times at random (Shotgun)
Two main assembly problems

- De Novo Assembly
- Resequencing
Reconstructing the Sequence (De Novo Assembly)

Cover region with high redundancy

Overlap & extend reads to reconstruct the original genomic region
Definition of Coverage

Length of genomic segment: $G$
Number of reads: $N$
Length of each read: $L$

**Definition:** Coverage $C = \frac{N \times L}{G}$

How much coverage is enough?

*Lander-Waterman model:* $\text{Prob[ not covered bp ]} = e^{-C}$
Assuming uniform distribution of reads, $C=10$ results in 1 gapped region /1,000,000 nucleotides
Repeats

Bacterial genomes: 5%
Mammals: 50%

Repeat types:

- **Low-Complexity DNA** (e.g. ATATATATACATA…)

- **Microsatellite repeats** \((a_1…a_k)^N\) where \(k \sim 3-6\)
  (e.g. CAGCAGTAGCAGCACCAG)

- **Transposons**
  - **SINE**
    (Short Interspersed Nuclear Elements)
    e.g., ALU: ~300-long, \(10^6\) copies
  - **LINE**
    (Long Interspersed Nuclear Elements)
    ~4000-long, 200,000 copies
  - **LTR retroposons**
    (Long Terminal Repeats (~700 bp) at each end)
    cousins of HIV

- **Gene Families**
  genes duplicate & then diverge (paralogs)

- **Recent duplications**
  ~100,000-long, very similar copies
Sequencing and Fragment Assembly

50% of human DNA is composed of repeats

Error!
Glued together two distant regions

3x10⁹ nucleotides
What can we do about repeats?

Two main approaches:

• Cluster the reads

• Link the reads
What can we do about repeats?

Two main approaches:

• Cluster the reads

• Link the reads
What can we do about repeats?

Two main approaches:
• Cluster the reads
  
• Link the reads
Sequencing and Fragment Assembly

3x10^9 nucleotides

ARB, CRD

or

ARD, CRB?
Sequencing and Fragment Assembly

3x10^9 nucleotides
Fragment Assembly
(in whole-genome shotgun sequencing)
Fragment Assembly

Given N reads…
Where N ~ 30 million…
We need to use a linear-time algorithm
Steps to Assemble a Genome

Some Terminology

**read**  a 500-900 long word that comes out of sequencer

**mate pair**  a pair of reads from two ends of the same insert fragment

**contig**  a contiguous sequence formed by several overlapping reads with no gaps

**supercontig** (scaffold)  an ordered and oriented set of contigs, usually by mate pairs

**consensus sequence**  sequence derived from the multiple alignment of reads in a contig

..ACGATTACAATAGGTT..
1. Find Overlapping Reads

<table>
<thead>
<tr>
<th>Read</th>
<th>Position</th>
<th>Word</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaactgcag</td>
<td></td>
<td>gcagtct</td>
<td></td>
</tr>
<tr>
<td>aaactgcag</td>
<td></td>
<td>tc</td>
<td></td>
</tr>
<tr>
<td>aactgcagt</td>
<td></td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>actgcagta</td>
<td></td>
<td>gtc</td>
<td></td>
</tr>
<tr>
<td>gtcggtactc</td>
<td></td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>gggcccaaa</td>
<td></td>
<td>aactgcagta</td>
<td></td>
</tr>
<tr>
<td>gggcccaaa</td>
<td></td>
<td>acggatct</td>
<td></td>
</tr>
<tr>
<td>gccccaaact</td>
<td></td>
<td>ctactacac</td>
<td></td>
</tr>
<tr>
<td>actgcagta</td>
<td></td>
<td>gaactgcagta</td>
<td></td>
</tr>
<tr>
<td>actgcagta</td>
<td></td>
<td>ctactacac</td>
<td></td>
</tr>
<tr>
<td>gtcggtactc</td>
<td></td>
<td>tgcagta</td>
<td></td>
</tr>
<tr>
<td>tgcagta</td>
<td></td>
<td>tgcagta</td>
<td></td>
</tr>
<tr>
<td>ctactacac</td>
<td></td>
<td>ctactacac</td>
<td></td>
</tr>
<tr>
<td>tactacaca</td>
<td></td>
<td>tactacaca</td>
<td></td>
</tr>
<tr>
<td>tactacaca</td>
<td></td>
<td>tactacaca</td>
<td></td>
</tr>
</tbody>
</table>

(read, pos., word, orient.) (word, read, orient., pos.)
1. Find Overlapping Reads

- Find pairs of reads sharing a k-mer, k ~ 24
- Extend to full alignment – throw away if not >98% similar

\[
\begin{align*}
\text{TACA} & \quad \text{TAGATTACACAGATTACT} & \quad \text{GA} \\
\text{----} & \quad \text{------} & \quad \text{------} \\
\text{TAGT} & \quad \text{TAGATTACACAGATTACTAGA}
\end{align*}
\]

- Caveat: repeats
  - A k-mer that occurs N times, causes $O(N^2)$ read/read comparisons
  - ALU k-mers could cause up to $1,000,000^2$ comparisons

- Solution:
  - Discard all k-mers that occur “too often”
    - Set cutoff to balance sensitivity/speed tradeoff, according to genome at hand and computing resources available
1. Find Overlapping Reads

Create local multiple alignments from the overlapping reads

TAGATTACACACAGATTACTGA
TAGATTACACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACACAGATTACTGA
TAGATTACACACAGATTACTGA
TAGATTACACACAGATTACTGA
TAG TTACACAGATTATTGA
TAGATTACACACAGATTACTGA
TAGATTACACACAGATTACTGA
1. Find Overlapping Reads

- Correct errors using multiple alignment

   **TAGATTACACAGATTACTGA**
   **TAGATTACACAGATTACTGA**
   **TAGATTACACAGATTACTGA**
   **TAGTTACACAGATTACTGA**

   insert A

   replace T with C

   correlated errors—probably caused by repeats
   ⇒ disentangle overlaps

In practice, error correction removes up to 98% of the errors