Gene Finding
Slides by Carl Kingsford
Genome of the Cow

a sequence of 2.86 billion letters
enough letters to fill a million pages of a typical book.

TATGGAGCCAGGTGCCCTGGGGCAACAAAAGACTGTGGTCACTGAATTCTCCTTCTTTGGTCTAACAAGAGAGACATAG
AACTGCAATCCATCTCTTTTGGCATCTTCTCTTTTGCTATGTGATCACGACGGGCAACTTGAGTATCCTGCCGCCATCTTTGTGGAGCCCAAACTCCACACCCCCATGTACTACTTCCTGGGGAACCTTTCTCTGCTGGACAT
TGGGTGCATCAGTGTCAACCATTCTCCCCCATGCTGCGCCTCTCGACATCGCCAGGGCTCCCTATGCAGCCTGCACATCACAGCTCTTCTTTTTCCACCTCCTGGCCTGGACTGTCACCTCCTGACAGCCATGGCCAC
GACCGCTACCTGGCCATTGCCAGCCCTCTCACCTATAGCATCGGCTACGCTGGGCTCCGTGTCATCTCGGTATCCTATGCAACGTGGGCTACGGGCTACCTGGCCATTTGCCAGCCCCTCACCTATAGCATCCGCATGAGCCGTGACGTCCAGGGAGCCCTGGTGCGTCCGCTGCTCCATTCCCCATCTCCTTCATCAATGCTCTGACCCACAGTGGCTGTGTCTGTGCTGGACTTCTGCGGCCCTAACGTGGTCAACCACTTCTACTGTGACCTCCCGCCCCTTTTCCAGCTCTCCTGCTCCAGCATCCACCTCAACGGGCAGCTACTTTTCGTGGGGGCCACCTTCATGGGGGTGGTCCCCATGGTCTTCATCTCGGTATCCTATGCACGTGGCAGCCGCAGTCCTGCGGATCCGCTCGGCAGAGGGCAGGAAGAAAGCCTTCTCCACGTGTGGCTCCCACCTCACCGTGGTCTGCATCTTTTATGGAACCGGCTTCTTCAGCTACATGCGCCTGGGCTCCGTGTCCGCCTCA
GACAAGGACAAGGGCATTGGCATCCTCAACACTGTCATCAGCCCCATGCTGAACCCACTCATCTACAGCCTCCGGAAACCCTGATGTGCAGGGCGCCCTGAAGAGGTTGCTGACAGGGGAAGCGGCCCCCGGAGTG...
"Central Dogma" of Biology

DNA =
- double-stranded, linear molecule
- each strand is string over \{A,C,G,T\}
- strands are complements of each other (A ↔ T; C ↔ G)
- substrings encode for genes
- most of which encode for proteins

mRNA (T → U)

Translation

Genome
The Genetic Code

- There are 20 different amino acids & 64 different codons.
- Lots of different ways to encode for each amino acid.
- The 3rd base is typically less important for determining the amino acid.
- Three different “stop” codons that signal the end of the gene.
- Start codons differ depending on the organisms, but AUG is often used.
Estimates for the # of Human Genes

Before human genome sequence was available, many (but not all) estimates for # of genes were high (> 80,000).

Now estimates are < 23,000.
Subsequences of DNA are “genes” that encode (mostly) for proteins.

# of genes in various organisms still not definitely know (because finding genes in the sequence is a hard problem that we will talk about).

But there are reasonably good estimates.
The Prokaryotic Gene Finding Problem

• Genes are subsequences of DNA that tell the cell how to make specific proteins.

• How can we find which subsequences of DNA are genes?

Start Codon: ATG
Stop Codons: TGA, TAG, TAA

Challenges:
• The start codon can occur in the middle of a gene (where it encodes for the amino acid methionine)
• The stop codon can occur in nonsense DNA between genes.
• The stop codon can occur “out of frame” inside a gene.
• Don’t know what “phase” the gene starts in.
A Simple Gene Finder

1. Find all stop codons in genome

2. For each stop codon, find the in-frame start codon farthest upstream of the stop codon, without crossing another in-frame stop codon.

   GGC $\text{TAG}$ $\text{ATG}$ $\text{AGG}$ $\text{GCT}$ $\text{CTA}$ $\text{ACT}$ $\text{ATG}$ $\text{GGC}$ $\text{GCG}$ $\text{TAA}$

   Each substring between the start and stop codons is called an ORF “open reading frame”

3. Return the “long” ORF as predicted genes.

   3 out of the 64 possible codons are stop codons ⇒ in random DNA, every 22nd codon is expected to be a stop.
Gene Finding as a Machine Learning Problem

- Given training examples of some known genes, can we distinguish ORFs that are genes from those that are not?

- **Idea**: can use distribution of codons to find genes.
  - every codon should be about equally likely in non-gene DNA.
  - every organism has a slightly different bias about how often certain codons are preferred.
  - could also use frequencies of longer strings (k-mers).
# Bacillus anthracis (anthrax) codon usage

<table>
<thead>
<tr>
<th>Codon</th>
<th>Amino Acid</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUU</td>
<td>F</td>
<td>0.76</td>
</tr>
<tr>
<td>UUC</td>
<td>F</td>
<td>0.24</td>
</tr>
<tr>
<td>UUA</td>
<td>L</td>
<td>0.49</td>
</tr>
<tr>
<td>UUG</td>
<td>L</td>
<td>0.13</td>
</tr>
<tr>
<td>CUU</td>
<td>L</td>
<td>0.16</td>
</tr>
<tr>
<td>CUC</td>
<td>L</td>
<td>0.04</td>
</tr>
<tr>
<td>CUA</td>
<td>L</td>
<td>0.14</td>
</tr>
<tr>
<td>CUG</td>
<td>L</td>
<td>0.05</td>
</tr>
<tr>
<td>AUU</td>
<td>I</td>
<td>0.57</td>
</tr>
<tr>
<td>AUC</td>
<td>I</td>
<td>0.15</td>
</tr>
<tr>
<td>AUA</td>
<td>I</td>
<td>0.28</td>
</tr>
<tr>
<td>AUG</td>
<td>M</td>
<td>1.00</td>
</tr>
<tr>
<td>GUU</td>
<td>V</td>
<td>0.32</td>
</tr>
<tr>
<td>GUC</td>
<td>V</td>
<td>0.07</td>
</tr>
<tr>
<td>GUA</td>
<td>V</td>
<td>0.43</td>
</tr>
<tr>
<td>GUG</td>
<td>V</td>
<td>0.18</td>
</tr>
<tr>
<td>UCU</td>
<td>S</td>
<td>0.27</td>
</tr>
<tr>
<td>UCC</td>
<td>S</td>
<td>0.08</td>
</tr>
<tr>
<td>UCA</td>
<td>S</td>
<td>0.23</td>
</tr>
<tr>
<td>UCG</td>
<td>S</td>
<td>0.06</td>
</tr>
<tr>
<td>UCC</td>
<td>S</td>
<td>0.08</td>
</tr>
<tr>
<td>UAC</td>
<td>Y</td>
<td>0.23</td>
</tr>
<tr>
<td>UAA</td>
<td>*</td>
<td>0.66</td>
</tr>
<tr>
<td>UAG</td>
<td>*</td>
<td>0.20</td>
</tr>
<tr>
<td>UGU</td>
<td>C</td>
<td>0.73</td>
</tr>
<tr>
<td>UGC</td>
<td>C</td>
<td>0.27</td>
</tr>
<tr>
<td>UGA</td>
<td>*</td>
<td>0.14</td>
</tr>
<tr>
<td>UGG</td>
<td>W</td>
<td>1.00</td>
</tr>
<tr>
<td>CGU</td>
<td>R</td>
<td>0.26</td>
</tr>
<tr>
<td>CAC</td>
<td>H</td>
<td>0.21</td>
</tr>
<tr>
<td>CAA</td>
<td>Q</td>
<td>0.78</td>
</tr>
<tr>
<td>CAG</td>
<td>Q</td>
<td>0.22</td>
</tr>
<tr>
<td>CGU</td>
<td>R</td>
<td>0.26</td>
</tr>
<tr>
<td>AAC</td>
<td>N</td>
<td>0.24</td>
</tr>
<tr>
<td>AAA</td>
<td>K</td>
<td>0.74</td>
</tr>
<tr>
<td>AAG</td>
<td>K</td>
<td>0.26</td>
</tr>
<tr>
<td>GGU</td>
<td>G</td>
<td>0.30</td>
</tr>
<tr>
<td>GGC</td>
<td>G</td>
<td>0.09</td>
</tr>
<tr>
<td>GGA</td>
<td>G</td>
<td>0.41</td>
</tr>
<tr>
<td>GGG</td>
<td>G</td>
<td>0.20</td>
</tr>
</tbody>
</table>
An Improved Simple Gene Finder

• Score each ORF using the product of the probability of each codon:

\[ \text{GFscore}(g) = \text{Pr(codon}_1\text{)} \times \text{Pr(codon}_2\text{)} \times \text{Pr(codon}_3\text{)} \times \ldots \times \text{Pr(codon}_n\text{)} \]

But: as genes get longer, GFscore(g) will decrease.

So: we should calculate GFscore(g[i...i+k]) for some window size k.

The final GFSCORE(g) is the average of the Scores of the windows in it.
Glimmer
Salzberg et al., NAR, 1998

- Score ORFs using 6 HMMs:
  - 1 model for each reading frame (3 forward, 3 reverse)
- ORFs for which the correct reading frame is the highest score are saved as candidates.

- Use “Interpolated Markov models” to adapt to data availability
- Handle overlapping ORFs
Interpolated HMMs

\[ P(S|M) = \sum_{x=1}^{n} IMM_8(S_x) \]

IMM score is a linear combination of 8th, 7th, …, 0th order models:

\[ IMM_k(S_x) = \lambda_k(S_{x-1}) \cdot P_k(S_x) + [1 - \lambda_k(S_{x-1})] \cdot IMM_{k-1}(S_x) \]

- **Sequence**
- **Length of the sequence**
- **String ending at position x**
- **Model**
- **Weight of the k-mer ending at position x-1**
- **Probability of letter at position x from a kth-order model**
Setting Parameters

• If # of occurrences of context k-mer ≥ 400, \( \lambda = 1 \)

• Otherwise compare the following with a \( \chi^2 \) statistic:

\[
\lambda_i(S_{x-1}) = \begin{cases} 
\frac{c}{400} \sum_{b \in \{acgt\}} f(s_1 s_2 \ldots s_i b) & \text{if } c < 0.50 \\
0.0 & \text{if } c \geq 0.50
\end{cases}
\]
Setting Parameters

- If # of occurrences of context k-mer ≥ 400, \( \lambda = 1 \)
- Otherwise compare the following with a \( \chi^2 \) statistic:

\[
\begin{align*}
\lambda_i(S_{x-1}) &= \begin{cases} 
0.0 & \text{if } c < 0.50 \\
\frac{c}{400} \sum_{b \in \{acgt\}} f(s_1s_2\ldots s_ib) & \text{if } c \geq 0.50
\end{cases}
\end{align*}
\]

When \( c \) is large, distribution of length \( i \) frequencies differs from that predicted by the \( i-1 \) order IMM. The more they differ, the more we weight them.
## IMM vs. 5th Order HMM

<table>
<thead>
<tr>
<th>Model</th>
<th>Genes found</th>
<th>Genes missed</th>
<th>Additional genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLIMMER IMM</td>
<td>1680 (97.8%)</td>
<td>37</td>
<td>209</td>
</tr>
<tr>
<td>5th-Order Markov</td>
<td>1574 (91.7%)</td>
<td>143</td>
<td>104</td>
</tr>
</tbody>
</table>

The first column indicates how many of the 1717 annotated genes in *H.influenzae* were found by each algorithm. The ‘additional genes’ column shows how many extra genes, not included in the 1717 annotated entries, were called genes by each method.

Salzberg et al., NAR, 1998
Overlaps

Scored separately with the two IMMs for the reading frames for the two genes

Discard the shorter gene if the longer gene’s reading frame scores higher
Eukaryotic Genes & Exon Splicing

Prokaryotic (bacterial) genes look like this:

```
ATG
```

Eukaryotic genes usually look like this:

```
ATG  exon  intron  exon  intron  exon  intron  exon  TAG
```

Introns are thrown away

mRNA:

```
AUG  exon  intron  exon  intron  exon  intron  exon  UAG
```

Exons are concatenated together

This spliced RNA is what is translated into a protein.
Hypothetical Eukaryotic Gene Parse Tree
A (Bad) Eukaryotic Gene Finder

Arrows show transitions with non-zero probabilities

What are some reasons this HMM gene finder is likely to do poorly?

Finite State Machine
Bad Eukaryotic Gene Finder

Why is it so bad?

- The positions in the codons are treated independently: the probability of emitting a base can’t depend on which previous base was emitted.

- Only one strand of the DNA is considered at once.

- Length distributions of introns and exons are not considered.
Genscan


- Explicitly double stranded
- One of the first to handle sequences with ≥ 1 gene in them

---

![Diagram of Genscan](Image: Zhang, 2002)
Generalized HMMs

- Each state can emit a sequence of symbols.

- In the diagram on the previous slide, each state emitted a complete gene feature (e.g. an entire exon):

\[
\max_{\pi} \prod_{i=1}^{n} \Pr(x_i \ldots x_{i+d_i} \mid \pi_i, d_i) \Pr(d_i \mid \pi_i) \Pr(\pi_i \rightarrow \pi_{i+1})
\]

- Probability of transitioning to the next state
- Probability that the state will emit \(d_i\) symbols.
- Probability of emitting the string of length \(d_i\).
Generalized HMMs

- Each state can emit a sequence of symbols.
- In the diagram on the previous slide, each state emitted a complete gene feature (e.g. an entire exon):

\[
\max_{\pi} \prod_{i=1}^{n} \Pr(x_i \ldots x_{i+d_i} | \pi_i, d_i) \Pr(d_i | \pi_i) \Pr(\pi_i \rightarrow \pi_{i+1})
\]

- Probability that the state will emit \(d_i\) symbols.
- Probability of emitting the string of length \(d_i\).
- Probability of transitioning to the next state.

This probability could itself be computed by an HMM or a Markov chain, etc.
Components Needed

- Probability distribution of initial state
  - the fraction of known genome corresponding to each state, divided into groups by GC content.

- State transition probabilities
  - the probability X follows Y in known genes

- Length distributions for each state
  - For exons: estimated from empirically observed distribution (next slide)
  - For introns: geometric distribution with parameter $q_{gc}$, where is the best fit parameter for regions with a given GC content.

- Sequence models for each state/length

  for states with strong motifs:
Feature Length Distributions

(a) Introns

(b) Initial exons

(c) Internal exons

(d) Terminal exons
Sequence Profiles (PSSM)

Amino Acid

Motif Position

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

A C D E

T V W Y

Color ≈ Probability that the $i^{th}$ position has the given amino acid $= e_i(x)$.  

$\sum = 1$
Sequence Generators

**Exons:** 3 different 5th-order Markov models:

- 1 model for each base of a codon
- Sequence generated by repeatedly applying model 1, then 2, then 3, and so on.
- Separate models for regions with GC content < 43%

**Non-coding states:** (F, T, I)

- 5th-order Markov model
- Separate model for regions with GC content < 43%

**Acceptor / donor sites:** a more complicated model that accounts for dependencies between positions.
GlimmerHMM

Majoros et al, 2004

Differences:
Interpolated HMM for coding sequences
New splicing model
GlimmerHMM

Differences:
Interpolated HMM for coding sequences
New splicing model

Majoros et al, 2004
## GlimmerHMM Performance

<table>
<thead>
<tr>
<th></th>
<th>Nuc Sens</th>
<th>Nuc Prec</th>
<th>Nuc Accur</th>
<th>Exon Sens</th>
<th>Exon Prec</th>
<th>Exact Genes</th>
<th>Size of test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>D. rerio</td>
<td>93%</td>
<td>78%</td>
<td>86%</td>
<td>77%</td>
<td>69%</td>
<td>24%</td>
<td>549 genes</td>
</tr>
<tr>
<td>C. elegans</td>
<td>96%</td>
<td>95%</td>
<td>96%</td>
<td>82%</td>
<td>81%</td>
<td>42%</td>
<td>1886 genes</td>
</tr>
<tr>
<td>Arabidopsis</td>
<td>97%</td>
<td>99%</td>
<td>98%</td>
<td>84%</td>
<td>89%</td>
<td>60%</td>
<td>809 genes</td>
</tr>
<tr>
<td>Cryptococcus</td>
<td>96%</td>
<td>99%</td>
<td>98%</td>
<td>86%</td>
<td>88%</td>
<td>53%</td>
<td>350 genes</td>
</tr>
<tr>
<td>Coccidioides</td>
<td>99%</td>
<td>99%</td>
<td>99%</td>
<td>84%</td>
<td>86%</td>
<td>60%</td>
<td>503 genes</td>
</tr>
<tr>
<td>Brugia</td>
<td>93%</td>
<td>98%</td>
<td>95%</td>
<td>78%</td>
<td>83%</td>
<td>25%</td>
<td>477 genes</td>
</tr>
</tbody>
</table>

% of true gene nucleotides that GlimmerHMM predicts as part of genes.

% of true exons that GlimmerHMM found.

% of predicted gene nucleotides that are correct.

% of predicted exons that are true exons.

% of genes perfectly found.
Compare with GENSCAN

- On 963 human genes:

<table>
<thead>
<tr>
<th></th>
<th>Nuc Sens</th>
<th>Nuc Prec</th>
<th>Nuc Acc</th>
<th>Exon Sens</th>
<th>Exon Prec</th>
<th>Exon Acc</th>
<th>Exact Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlimmerHMM</td>
<td>86%</td>
<td>72%</td>
<td>79%</td>
<td>72%</td>
<td>62%</td>
<td>67%</td>
<td>17%</td>
</tr>
<tr>
<td>Genscan</td>
<td>86%</td>
<td>68%</td>
<td>77%</td>
<td>69%</td>
<td>60%</td>
<td>65%</td>
<td>13%</td>
</tr>
</tbody>
</table>

- Note that overall accuracy is pretty low.
Generalized Pair HMMs

Use: find genes simultaneously in 2 genomes increased signal b/c the structure of homologous genes is often very similar.

- Pair: Each state emits two symbols, one for each sequence
- Generalized Pair: a pair of lengths d, e is drawn from a joint probability distribution and a pair of sequences X, Y of length d, e, respectively, are generated at each state.

Generalized Pair HMMs

Use: find genes simultaneously in 2 genomes increased signal b/c the structure of homologous genes is often very similar.

- Pair: Each state emits two symbols, one for each sequence
- Generalized Pair: a pair of lengths $d, e$ is drawn from a joint probability distribution and a pair of sequences $X, Y$ of length $d, e$, respectively, are generated at each state.


Image: Zhang, 2004

Reverse strand: mirror reflection of above
Combining Several Predictors

- Use each programs exon probability scores (probability that exon is included in the parse).

- Example: keep disagreeing exons only if score is above a threshold.

Rojic et al. *Bioinformatics* 18(8) 2002
Recap

• Simple gene finding approaches use codon bias and long ORFs to identify genes.

• Many top gene finding programs for Eukaryotes are based on generalizations of Hidden Markov Models because multiple types of signals are present in a gene (intron, exon, etc.)

• Basic HMMs must be generalized to emit variable sized strings.