Hidden Markov Models

Selecting the initial model parameters

Using HMMs for (simple) gene finding
HMMs as a generative model

A HMM *generates a sequence of observables* by moving from latent state to latent state according to the transition probabilities and *emitting an observable* (from a discrete set of observables, i.e. a finite alphabet) from each latent state visited *according to the emission probabilities* of the state ...

For a HMM that generates finite strings (e.g. a HMM with an end-state), the language $L = \{X | p(X) > 0\}$ is regular ...
Selecting initial model parameters

The initial selection of transition and emission probabilities, i.e. $A$, $\pi$, $\Phi$, should model (how we see) the underlying structure of the observations, i.e. the syntax of possible sequences of observations, recall that the language $L = \{x \mid P(x \mid \theta) > 0\}$ is regular.

The initial selection of parameters is essential just to decide which parameters are 0 (or 1), i.e. to decide which transitions of emission should never (or always) be possible ...
Example – Gene finding

Each protein is encoded in a stretch of DNA. A gene ...

Which is expressed when the protein is needed ...

Important problem

Locating genes on the genome and determining how they get expressed ...

Recognizing the patterns that indicates a gene ...
<table>
<thead>
<tr>
<th>UCA</th>
<th>GAG</th>
<th>UCG</th>
<th>CAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHE</td>
<td>SER</td>
<td>TYR</td>
<td>CYS</td>
</tr>
<tr>
<td>PHE</td>
<td>SER</td>
<td>TYR</td>
<td>CYS</td>
</tr>
<tr>
<td>LEU</td>
<td>SER</td>
<td>OCHRE</td>
<td>OPAL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCU</th>
<th>GAG</th>
<th>UCG</th>
<th>CAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRO</td>
<td>HIS</td>
<td>ARG</td>
<td>ARG</td>
</tr>
<tr>
<td>PRO</td>
<td>HIS</td>
<td>ARG</td>
<td>ARG</td>
</tr>
<tr>
<td>PRO</td>
<td>GLU</td>
<td>ARG</td>
<td>ARG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCU</th>
<th>GAG</th>
<th>UCG</th>
<th>CAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILEU</td>
<td>THR</td>
<td>ASPN</td>
<td>SER</td>
</tr>
<tr>
<td>ILEU</td>
<td>THR</td>
<td>ASPN</td>
<td>SER</td>
</tr>
<tr>
<td>ILEU</td>
<td>THR</td>
<td>LYS</td>
<td>ARG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UCG</th>
<th>GAG</th>
<th>UCG</th>
<th>CAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL</td>
<td>ALA</td>
<td>ASP</td>
<td>GLY</td>
</tr>
<tr>
<td>VAL</td>
<td>ALA</td>
<td>ASP</td>
<td>GLY</td>
</tr>
<tr>
<td>VAL</td>
<td>ALA</td>
<td>GLU</td>
<td>GLY</td>
</tr>
</tbody>
</table>

**KEY**
- PHE: Phenylalanine
- GLU: Glutamic Acid
- ASP: Aspartic Acid
- SER: Serine
- THR: Threonine
- ILE: Isoleucine
- LEU: Leucine
- VAL: Valine
- ALA: Alanine
- GLY: Glycine
- MET: Methionine
- PRO: Proline
- LYS: Lysine
- CYS: Cysteine
- ARG: Arginine
- TRP: Tryptophan
- HIS: Histidine
- TRA: Trinque
- GLN: Glutamine
- ASN: Asparagine
- ASN: Asparagine
Design a HMM that models the syntax of genes
Gene structure

 Depends on the organism (eucaryote or procaryote)

Smaller genomes and high coding density.

Large genomes. Intron/exon structure and low coding density.
Gene structure in eukaryotes

Eukaryotic gene structure in more details
Gene structure in procaryotes

**Biological facts**
- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-code `atg`
- The gene ends with a stop-codon `taa`, `tag`, or `tga`
- The number of nucleotides in a gene is a multiple of 3

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]

**Z:** NNNCCCCCCCCCCCCNNNNNNNNNNCCCCCCCCCCCCCCCCCCCCCCCCCCCCCNNNNNNNNNNNNNN

**X:** acgatgcgcctaatatgtccgatgacgtgagcataagcgcacatgcag
Gene structure in procaryotes

- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon `atg`

Z: NNNCCCCCCNNNNNNNNNNCCCCCCCCCCCCCCCCCCCCCCCCCNNNNNNNNNNNNNN

X: acgatgcgctaatatgtccgatgacgtgagcataaagcgacatgcag

C: coding

N: non-coding

$\pi_N = 1$
$\pi_C = 0$
Gene structure in procaryotes

Biological facts
- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon \textbf{atg}

\[
\begin{align*}
Z: & \quad \text{NNNCCCCCCCCCCCCNNNNNNNNNNNNNNNNNNNNNNNNNNN} \\
X: & \quad \text{acgatgcgctaatatgtccgatgacgtgagcataagcgacat}
\end{align*}
\]

\[
\begin{align*}
\pi_N &= 1 \\
\pi_C &= 0
\end{align*}
\]
Gene structure in procaryotes

**Biological facts**
- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon `atg`
- The gene ends with a stop-codon `taa, tag` or `tga`

**DNA**
- **mRNA**
- **Translation**

**Z:** NNNCCCCCCCCCCNNNNNNNNNCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCNNNNNNNNNNNNN

**X:** acgatgcgcctaataatgttcgatgacgtgagcataagcgacatg

\[
\begin{align*}
\pi_N &= 1 \\
\pi_C &= 0
\end{align*}
\]

A: >0  C: 0  G: 0  T: 0  N: non-coding
A: 1   C: 0  G: 0  T: 0
A: 0   C: 0  G: 0  T: 1
A: 0   C: 0  G: 1  T: 0
A: >0  C: >0  G: >0  T: >0  C: coding
Gene structure in procaryotes

- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon **atg**
- The gene ends with a stop-codon **taa**, **tag** or **tga**

\[\pi_N = 1\]
\[\pi_C = 0\]
Gene structure in procaryotes

- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon **atg**
- The gene ends with a stop-codon **taa, tag** or **tga**
- The number of nucleotides in a gene is a multiple of 3

N: non-coding

- A: >0
- C: >0
- G: >0
- T: >0

C: coding

- A: >0
- C: >0
- G: >0
- T: >0

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Gene structure in procaryotes

- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon **atg**
- The gene ends with a stop-codon **taa**, **tag** or **tga**
- The number of nucleotides in a gene is a multiple of 3

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Gene structure in procaryotes

From “An Introduction to HMMs for Biological Sequences”, A. Krogh, 1998
Gene structure in procaryotes

N: non-coding

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Gene structure in procaryotes

Gene finding

- Select initial model structure (e.g. as done here)

- Select model parameters by training. Either “by counting” from examples of (X,Z)'s, i.e. genes with known structure, or by EM- or Viterbi-training from examples of X, i.e. sequences which are known to contain a gene.

- Given a new sequence X, predict its gene structure using the Viterbi algorithm for finding the most likely sequence of underlying latent states, i.e. its gene structure
Example – Gene finding

Gene finding

* Select initial model structure (e.g. as done here)
* Select model parameters by training. Either “by counting” from examples of \((X,Z)\)'s, i.e. genes with known structure, or by EM- or Viterbi-training from examples of \(X\), i.e. sequences which are known to contain a gene.

Even more biology

* There can be genes in both directions (and overlapping)
* There are more possible start-codons \(\text{atg, gtg, and ttg}\)
* Internal codons cannot be start- or stop-codons
* And a lot more ...

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
DNA

AGT  GAT  AAT  GTA
\(e'_1, e'_2, e'_3\)  \(s'_1, s'_2, s'_3\)

TTA  CAT  ATG  TAA
CTA  TGA  TAG  TGA

ATG  TAA  TAG  TGA

5' end

Adenine  Thymine

5' end

Phosphate-deoxyribose backbone

Sugar phosphate backbone

3' end

U.S. National Library of Medicine
Even more biology

There can be genes in both directions

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Example – 7-state HMM

Observable: \{A, C, G, T\}, States: \{0, 1, 2, 3, 4, 5, 6\}

\[
\begin{array}{cccccccc}
0.00 & 0.00 & 0.90 & 0.10 & 0.00 & 0.00 & 0.00 & 0.00 \\
1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.05 & 0.90 & 0.05 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.10 & 0.90 & 0.00 & 0.00 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
\end{array}
\]

\[
\begin{array}{cccccc}
0.30 & 0.25 & 0.25 & 0.20 & 0.20 & 0.35 \\
0.20 & 0.35 & 0.15 & 0.30 & 0.20 & 0.25 \\
0.40 & 0.15 & 0.20 & 0.25 & 0.25 & 0.25 \\
0.25 & 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\
0.20 & 0.40 & 0.30 & 0.10 & 0.30 & 0.20 \\
0.30 & 0.20 & 0.30 & 0.20 & 0.30 & 0.20 \\
0.15 & 0.30 & 0.20 & 0.35 & 0.30 & 0.20 \\
\end{array}
\]

Graph representation of the 7-state HMM with transition probabilities between states and emission probabilities for observable nucleotides.
This model is also applicable for gene finding.

It does not model start- and stop-codons explicitly, but models that genes in both directions are a sequence of triplets.
Problem: From annotation to Z

DNA → TRANSCRIPTION → mRNA → TRANSLATION → protein

**Biological facts**
- The gene is a substring of the DNA sequence of A,C,G,T's
- The gene starts with a start-codon *atg*

Z: NNNCCCCCC...NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN

X: acgatgcgc...gacgcga

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]

A: >0  C: >0  G: >0  T: >0  N: non-coding
A: 1   C: 0   G: 0   T: 0   C: coding
A: 0   C: 0   G: 0   T: 1
A: 0   C: 0   G: 1   T: 0
A: >0  C: >0  G: >0  T: >0
Problem: From annotation to Z

Problem: The string $Z=\text{NNNCCCCC...}$ is not a proper sequence of states in the illustrated HMM, but it can easily be converted into one (because there in this case is a 1-1 matching between a sequence of Ns and Cs and a sequence of states).

$Z: \text{NNNCCCCC...}$

$X: \text{acgatgcgc...}$

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Problem: From annotation to Z

Problem: The string \( Z = \text{NNNCCC} \ldots \) is not a proper sequence of states in the illustrated HMM, but it can easily be converted into one (because there is a 1-1 matching between a sequence of Ns and Cs and a sequence of states).

\[
\begin{align*}
\text{Z: } & \quad \text{NNNCCC} \ldots \\
\text{X: } & \quad \text{acgatgcgc} \ldots \\
\end{align*}
\]

\[
\begin{align*}
\pi_N &= 1 \\
\pi_C &= 0 \\
\end{align*}
\]
Evaluating performance

**Nucleotide Level**

![Diagram of nucleotide level evaluation]

**REALITY**

- TN (True Negative)
- FN (False Negative)
- TP (True Positive)
- FP (False Positive)

**PREDICTION**

- TN (True Negative)
- FN (False Negative)
- TP (True Positive)
- FP (False Positive)

**Formulas**

- **Sensitivity**
  
  \[ S_n = \frac{TP}{TP + FN} \]

- **Specificity**
  
  \[ S_p = \frac{TP}{TP + FP} \]

- **Correlation Coefficient**
  
  \[ CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}} \]

- **Approximate Correlation**
  
  \[ AC = (ACP - 0.5) \times 2 \]

**References**

- Burset and Guigo, 1996
compare_anns.py

Genome 6
Cs   (tp=757332, fp=164766, tn=305197, fn=57217): Sn = 0.9298, Sp = 0.8213, AC = 0.6213
Rs   (tp=715865, fp=127462, tn=304830, fn=57584): Sn = 0.9255, Sp = 0.8489, AC = 0.6603
Both (tp=1473197, fp=292228, tn=247613, fn=114801): Sn = 0.9277, Sp = 0.8345, AC = 0.4520

Genome 7
Cs   (tp=868820, fp=236008, tn=517048, fn=79049): Sn = 0.9166, Sp = 0.7864, AC = 0.6285
Rs   (tp=815026, fp=226580, tn=511963, fn=84134): Sn = 0.9064, Sp = 0.7825, AC = 0.6205
Both (tp=1683846, fp=462588, tn=268917, fn=163183): Sn = 0.9117, Sp = 0.7845, AC = 0.4529

Genome 8
Cs   (tp=705403, fp=137180, tn=351159, fn=74782): Sn = 0.9041, Sp = 0.8372, AC = 0.6424
Rs   (tp=607762, fp=169829, tn=351738, fn=74203): Sn = 0.8912, Sp = 0.7816, AC = 0.5865
Both (tp=1313165, fp=307009, tn=276956, fn=148985): Sn = 0.8981, Sp = 0.8105, AC = 0.4166

Genome 9
Cs   (tp=776640, fp=203664, tn=340882, fn=88415): Sn = 0.8978, Sp = 0.7922, AC = 0.5550
Rs   (tp=759048, fp=219786, tn=336181, fn=93116): Sn = 0.8907, Sp = 0.7755, AC = 0.5270
Both (tp=1535688, fp=423450, tn=276956, fn=181531): Sn = 0.8943, Sp = 0.7839, AC = 0.3122

Genome 10
Cs   (tp=612457, fp=106124, tn=253878, fn=88014): Sn = 0.8744, Sp = 0.8523, AC = 0.5872
Rs   (tp=371869, fp=138143, tn=291605, fn=50287): Sn = 0.8809, Sp = 0.7291, AC = 0.5707
Both (tp=984326, fp=244267, tn=203591, fn=138301): Sn = 0.8768, Sp = 0.8012, AC = 0.3640
Even more biology
There can be genes in both directions

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]
Analysis of some genomes

Start-codon in normal genes:
ATG [8423, 'NCCC']
ATC [3, 'NCCC']
ATA [1, 'RCCC']
GTG [713, 'NCCC']
ATT [3, 'NCCC']
CTG [2, 'NCCC']
GTT [1, 'NCCC']
CTC [1, 'NCCC']
TTA [1, 'NCCC']
TTG [1020, 'NCCC']

Stop-codon in normal genes:
TAG [1949, 'CCCN']
TGA [1531, 'CCCN']
TAA [6686, 'CCCN']

Reversed stop-codon in reversed genes:
TTA (reverse-complement: TAA) [6596, 'NRRR']
CTA (reverse-complement: TAG) [2014, 'NRRR']
TCA (reverse-complement: TGA) [1148, 'NRRR']

Reversed start-codon in reversed genes:
TAT (reverse-complement: ATA) [2, 'RRRN']
ATG (reverse-complement: CAT) [1, 'RRRN']
GAT (reverse-complement: ATC) [1, 'RRRN']
CAT (reverse-complement: ATG) [8077, 'RRRN']
AAT (reverse-complement: ATT) [4, 'RRRN']
TAC (reverse-complement: GTA) [1, 'RRRN']
CAC (reverse-complement: GTG) [715, 'RRRN']
CAA (reverse-complement: TTG) [953, 'RRRN']
CAG (reverse-complement: CTG) [4, 'RRRN']

Length of genome1: 1852441 (1852441)
Length of genome2: 2211485 (2211485)
Length of genome3: 2499279 (2499279)
Length of genome4: 1796846 (1796846)
Length of genome5: 2685015 (2685015)
Length of genome6: 2127839 (2127839)
Length of genome7: 2742531 (2742531)
Length of genome8: 2046115 (2046115)
Length of genome9: 2388435 (2388435)
Length of genome10: 1570485 (1570485)
Length of genome11: 2096309 (2096309)