Introduction to Bioinformatics

Sequence logos and operon prediction

Lecturer: Jan Baumbach
Teaching assistant(s): Diogo Marinho

Lecture outline

- Part I: Logos
  - Sequence logos
  - HMM logos
- Part II: Operon prediction
  - Introduction
  - Operon prediction
    - Previous work
  - A novel method for accurate operon prediction
    - Principles
    - Features
    - Statistical inference
    - Results
  - Summary

Part I – Sequence logos

Sequence Logos (I)

- Profile P_{ij}, $i=1..L, j = 1..|AAs|$, i.e. a probability distribution of AAs for each position (L – length of a sequence)
- The *uncertainty* or *entropy* of distribution P_i at the i-th position of the profile

$$H(P_i) = -\sum_{j \in AAs} P_{ij} \log_2 P_{ij}$$

$H(P_i) = 0,$ if only one residue is found at that position (no uncertainty)
$H(P_i)$ is $max,$ if frequencies of each AA are equal

Sequence Logos (2)

- The *information content* of position i

$$I(P_i) = \log_2 |AAs| - H(P_i)$$

- *Background frequencies* of AAs. (e.g. tryptophan occurs much less than leucine)
  - count AA occurrences in all known proteins, or only in the proteins of the superfamily under consideration.

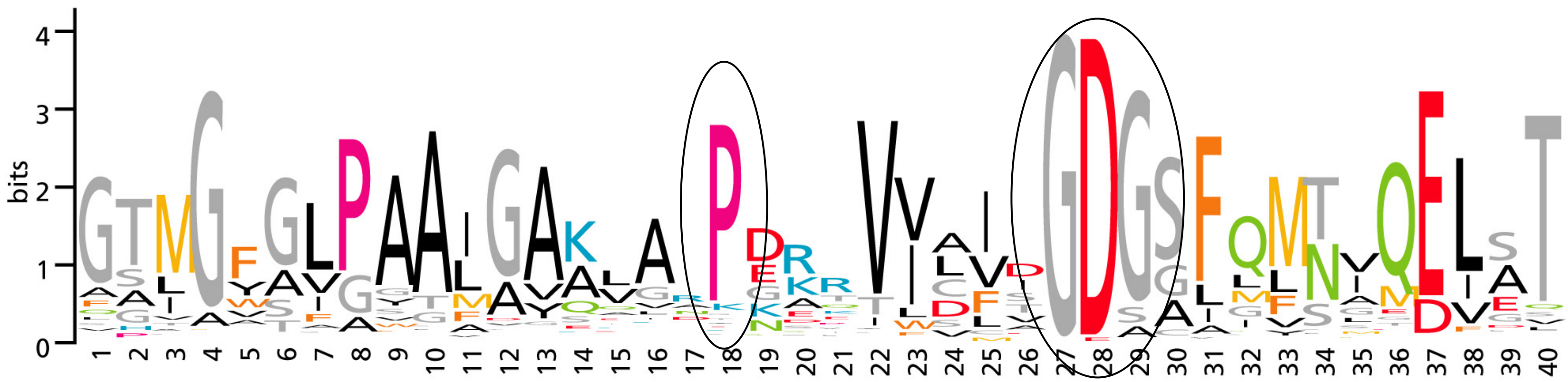
- *Relative entropy*

$$H(P_i \| \pi) = -\sum_{j \in AAs} P_{ij} \log_2 (P_{ij} / \pi_j)$$

π_j – background frequency of AA j

Sequence Logos (3)

- contribution of a residue: $P_{ij} \cdot H(P_i \parallel \pi)$
- heights of residues in a logo are proportional to their contributions
- stack height is proportional to the relative entropy
- colors highlight different properties of different AAs.



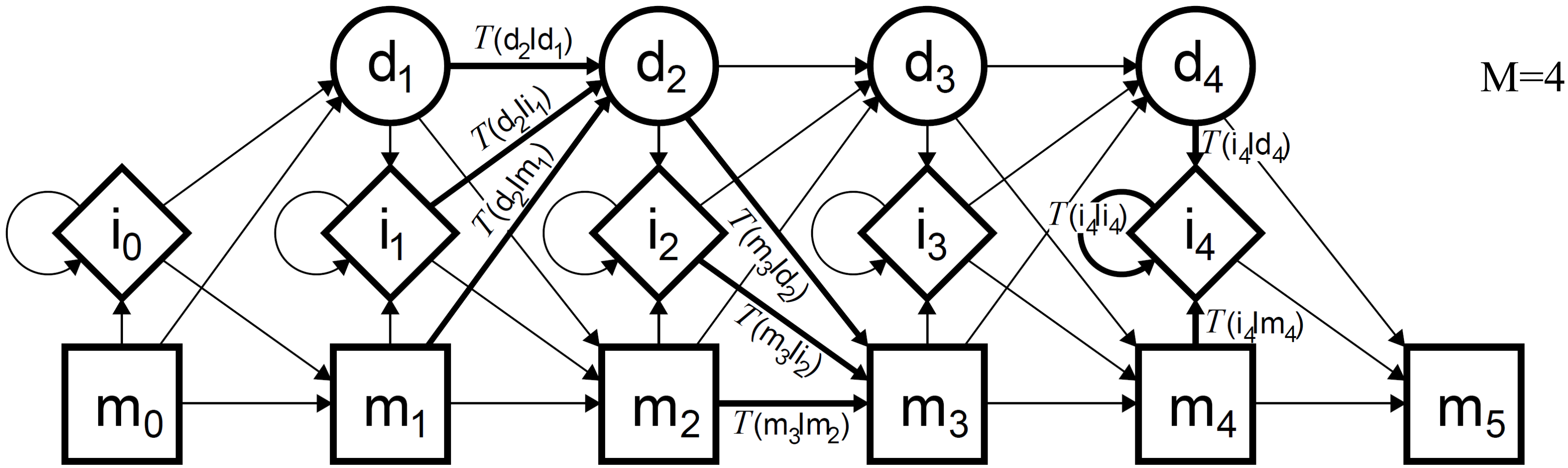
Example of a protein sequence alignment logo, taken from the BLOCKS database. This is block IPB000399E, constructed from an alignment of 186 sequences of TDP (thymine diphosphate)-binding enzymes. This region of sequence is also present in PROSITE (entry PS00187), which reports a TPP (thiamine pyrophosphate)-binding pattern [LIVMF]-[GSA]-x(5)-P-x(4)-[LIMFYW]-x-[LIVMF]-x-G-D-[GSA]-[GSAC].Note that this pattern is only found in a subset of 44 of the 186 sequences whose alignment is shown here. The conserved proline of this pattern in column 18, and G-D-[GSA] at columns 27-29. http://weblogo.berkeley.edu

HMM logos

HMMs in Bioinformatics

- Coding and non-coding regions in DNA determination
- Modeling of protein-binding sites in DNA
- Modeling of protein superfamilies
- Protein secondary structure prediction
- Transmembrane protein prediction
- Protein Structure Prediction
- Multiple sequence alignment
- Gene prediction

HMMs for protein sequence generation: architecture



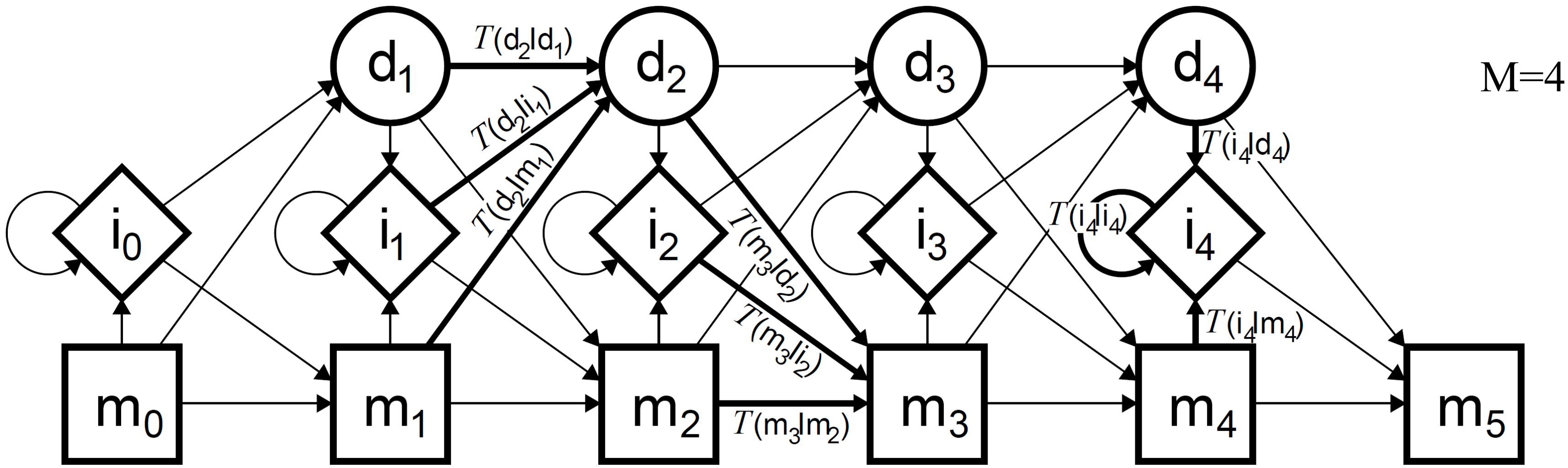
m – match states (columns in multiple alignment)

i – insert states

d – delete states

$P(x\|q)$	Probability of a letter *x* in state *q* *(emission probability)*
$T(r\|q)$	Probability of transition from state *q* to state *r* *(transition probability)*

HMMs for protein sequence generation: assumptions



Markov assumption

- the next state depends only on the current state

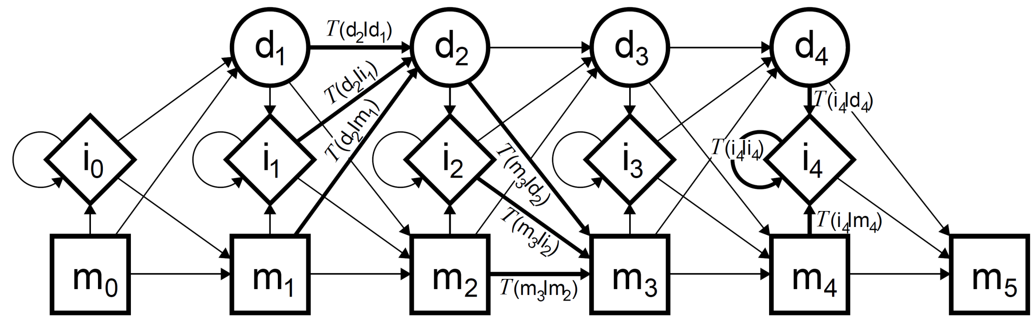
Stationary assumption

- state transition probabilities are independent on time

Output independence assumption

- the current observation is independent of previous observations

HMM Notation



x	Amino acid (AA)
s	Sequence of AAs ($s=x_1,\ldots x_L$)
L	Length of sequences
q, r	State in an HMM
path	A sequence of states
N	Number of states in a path ($N \geq L$)
M	Length of model
m, i, d	Match, insert and delete states
m_0, m_{M+1}	Begin and end states
$P(x\|q)$	Probability distribution of AAs in state q *(emission probability)*
$T(r\|q)$	Probability of transition from state q to r *(transition probability)*
$l(i)$	Index in the sequence $x_1,\ldots x_L$ of AAs produced in state q_i if q_i is a match or insert state
$s(1),\ldots,s(n)$	Training set of sequences

model

An HMM of certain length M and all transition and emission probabilities

Prob($x_1,\ldots,x_L, q_0,\ldots,q_{N+1}$|model)

The probability of the event that the path $q_0,\ldots,q_{N+1}$ is taken and the sequence $x_1, \ldots,x_L$ is generated

Prob($x_1,\ldots,x_L$|model)

The probability of any sequence $x_1,\ldots,x_L$ of AAs

Prob(sequences|model)

The probability of a set of training sequences s(1),…,s(n)

Sequence Probabilities

The probability of the event that the path $q_0, \ldots, q_{N+1}$ is taken and the sequence $x_1, \ldots, x_L$ is generated:

$$\mathrm{Prob}(x_1 \ldots x_L, q_0 \ldots q_{N+1} \mid \mathrm{model}) = T(m_{N+1} \mid q_N) \times \prod_{i=1}^{N} T(q_i \mid q_{i-1}) P(x_{l(i)} \mid q_i)$$

where $P(x_{l(i)} \mid q_i) = 1$ if q_i is a delete state

$$\mathrm{Prob}(x_1 \ldots x_L \mid \mathrm{model}) = \sum_{paths\ q_0 \ldots q_{N+1}} \mathrm{Prob}(x_1 \ldots x_L, q_0 \ldots q_{N+1} \mid \mathrm{model})$$

The probability of any sequence $x_1, \ldots, x_L$ of AAs is sum over all possible paths that could produce that sequence.

Parameter Estimation: Maximum Likelihood

Maximum Likelihood (ML) of the model:

Given a set of training sequences s(1),…,s(n), find a model

$$\mathrm{Prob}(\mathrm{sequences} \mid \mathrm{model}) = \prod_{j=1}^{n} \mathrm{Prob}(s(j) \mid \mathrm{model}) \quad \longrightarrow \quad \max$$

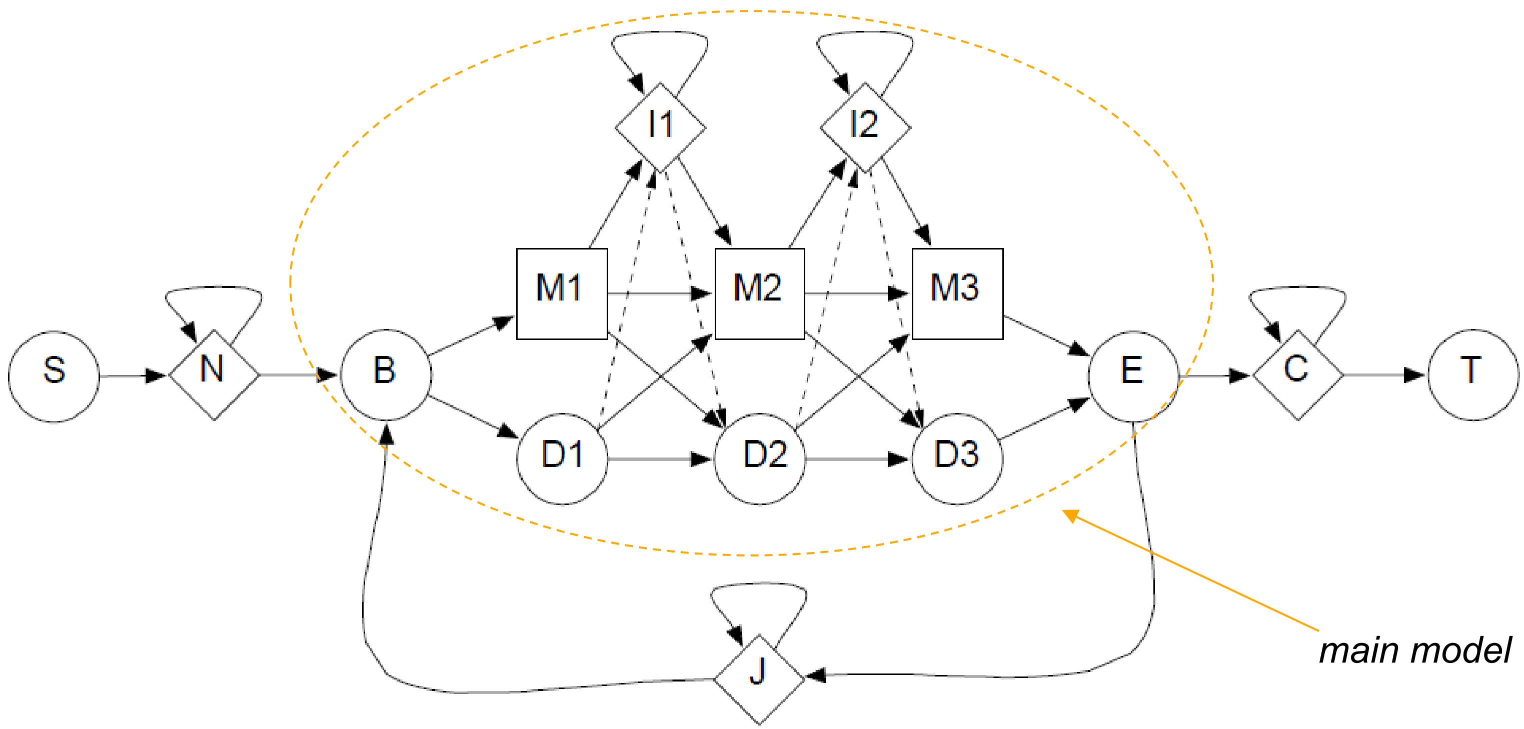
Negative Log Likelihood – as a measure of fitness of the model

$$\mathrm{NLL\text{-}score} = -\log \mathrm{Prob}(\mathrm{sequences} \mid \mathrm{model})$$

HMMer – One tool to rule them all

- A profile HMM of length 3 according to the HMMER software package:

  B – begin, E – end, Di – delete, Mi – match, Ii – insert (the other states are not relevant for HMM Logos)



HMM Logos: Main idea (1)

- Method to visualize central aspects of protein families represented by HMM profile
- Incorporates both emission and transition probabilities of an HMM
- An extension of Sequence Logos

HMM Logos: Main idea (2)

- Stack height – deviation of the position's AA emission frequencies from the background frequencies
- Stack width for both
  - probability of reaching the state (the hitting probability)
  - the expected number of AAs the state emits during a pass through (the state's expected contribution)
- Highlight differences between homologous subfamilies

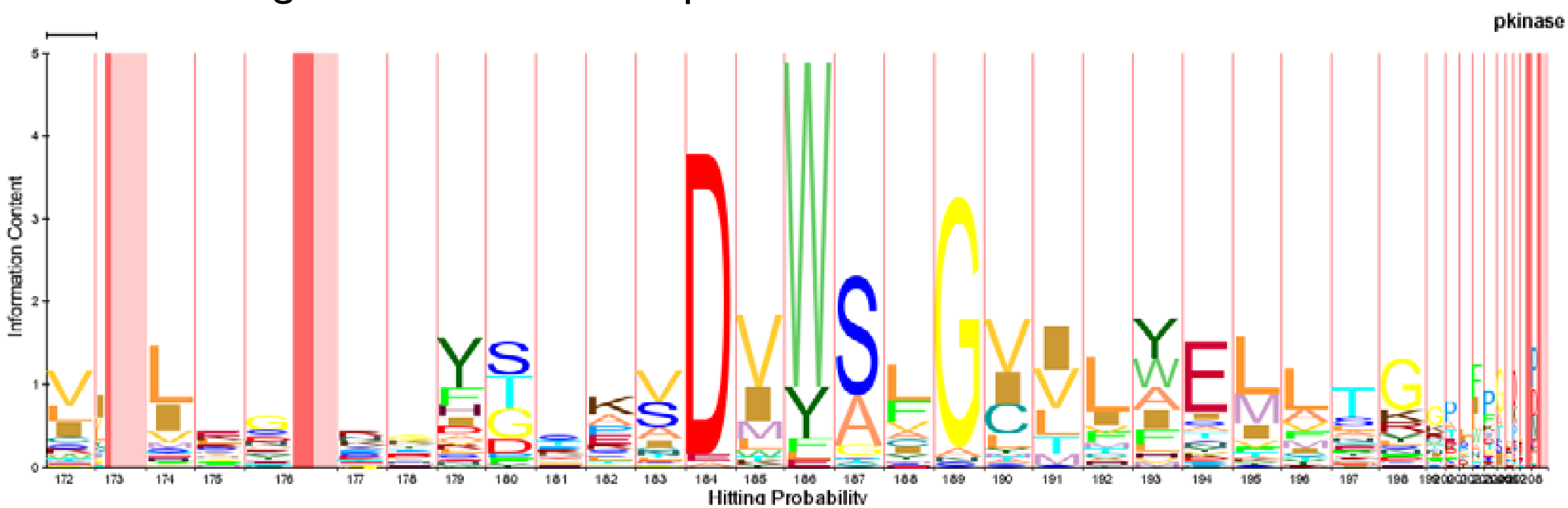
HMM Logos: Definitions

- s – a state of the main model
- $h(s)$ – hitting probability of state s from **B** following any possible path

  =sum of probabilities of visiting state *s* starting in state **B**
- $C(s)$ – contribution:
  - insert state: number of emitted gaps along a path B→…→E
  - match state: 1 (if *s* reached) or 0 (otherwise)
- $c(s) := \mathrm{E}[C(s)]$ – expected contribution of state s:
  - insert state: h(s) * expected number of gaps
  - match state: h(s)

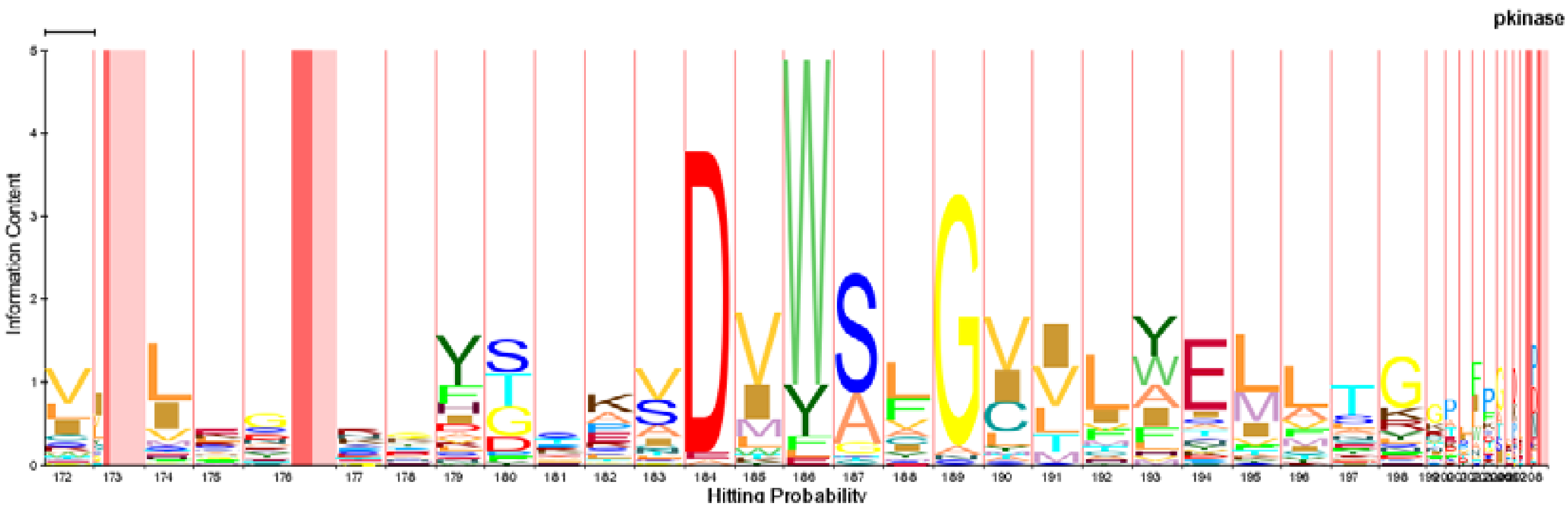
HMM Logos (1)

- Alternating stacks for match and insert states for all positions $1,\ldots,L$ in the profile
- The total height of a stack is the relative entropy $H(e||\pi)$ between the state's emission distribution e and the background distribution π obtained from state N.
- The relative height of a letter within the stack is proportional to its emission probability e_j.
- The largest letter is on the top of the stack.



HMM Logos (2)

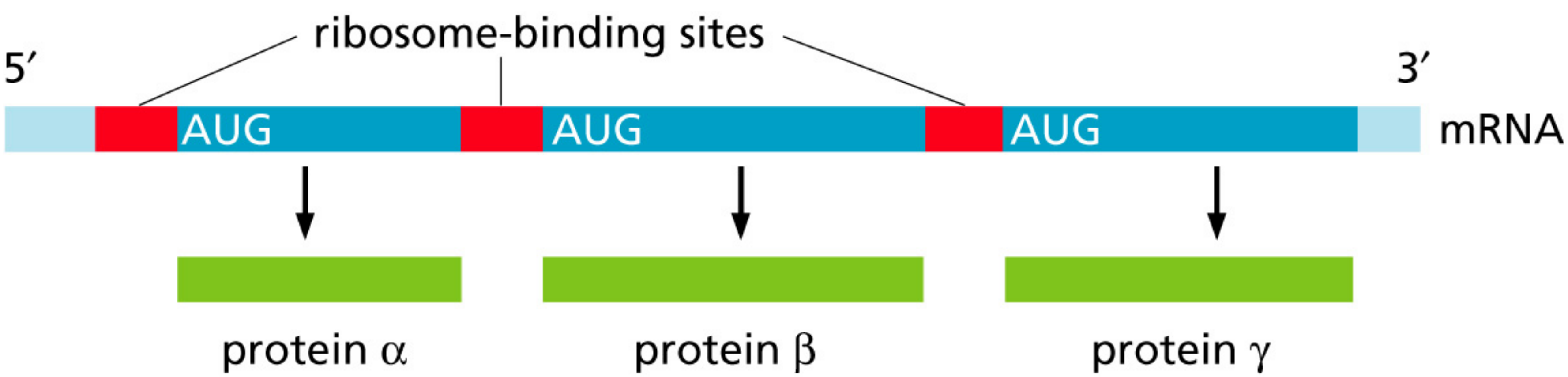
- Width of a stack s: expected contribution *c(s)*.
- Background of an insert state's stack is shaded in two different colors:
  - *h(s)* shaded with a medium-red background.
  - *c(s) - h(s)* shaded with a lighter red.
- letters in different colors – structural or functional similarity



Part II – Operon Prediction

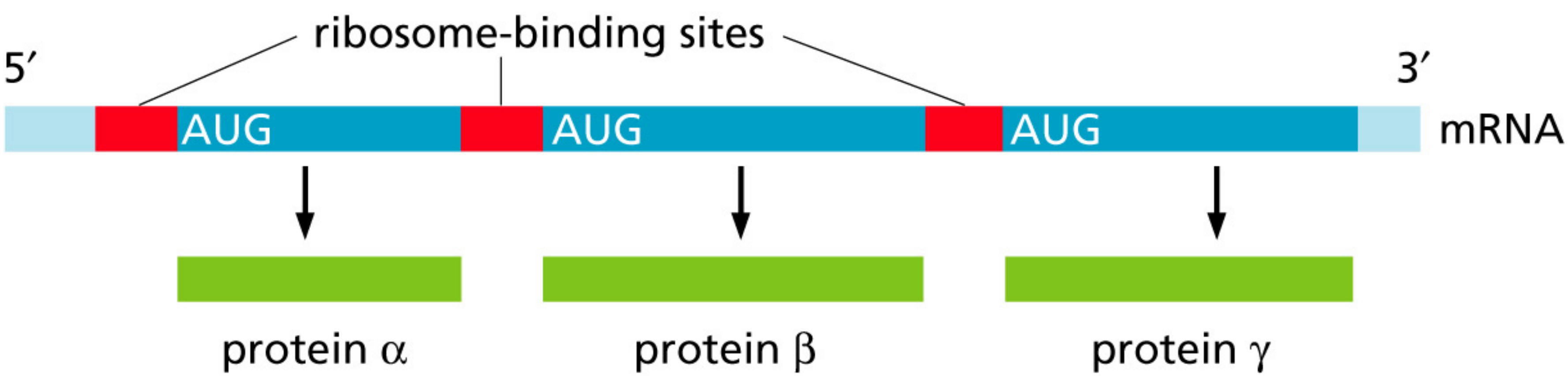
Operon (1)

- Operon
  - is a segment of a genome
  - consists of several consecutive genes whose expression is controlled as a single unit.
  - is transcribed into a single mRNA molecule that encodes several proteins.



Operon (2)

- Operons
  - require only one control region to activate the simultaneous expression.
  - encode functionally related protein sequences
  - are rarely found in eukaryotes



Problem and aims:

- More and more genome data
  - ⇒ characterization of transcriptional regulation
    - ⇒ Automated methods for prediction of regulatory interactions are required

=> Enhance our knowledge of gene regulation and function.

Previous work:

- Methods relying on DBs of experimentally identified transcripts for training and for validation
  - Supervised
  - Available only for few organisms
  - Difficult to judge accuracy on new genomes

  - **Conservation of operons in multiple species**
    - Idea: Adjacent and evolutionarily conserved genes are likely to be in one operon
    - Confident prediction
    - But: Operon annotation missing for most bacteria

Previous work:

- Methods relying on DBs of experimentally identified transcripts for training and for validation
  - Supervised
  - Available only for few organisms
  - Difficult to judge accuracy on new genomes

  - **Conservation of operons in multiple species**
    - Idea: Adjacent and evolutionarily conserved genes are likely to be in one operon
    - Confident prediction
    - But: Operon annotation missing for most bacteria

  - **Distance models**
    - Idea: genes in the same operon have fewer base pairs of DNA in between, than just adjacent genes
    - But: distance varies from species to species

Previous work:

- Methods relying on DBs of experimentally identified transcripts for training and for validation
  - Supervised
  - Available only for few organisms
  - Difficult to judge accuracy on new genomes

  - **Conservation of operons in multiple species**
    - Idea: Adjacent and evolutionarily conserved genes are likely to be in one operon
    - Confident prediction
    - But: Operon annotation missing for most bacteria

  - **Distance models**
    - Idea: genes in the same operon have fewer base pairs of DNA in between, than just adjacent genes
    - But: distance varies from species to species

→ Unsupervised methods necessary

A Novel Method for Accurate Operon Predictions in All Sequenced Prokaryotes

Price M.N., Huang K.H., Alm E.J., Arkin A.P.

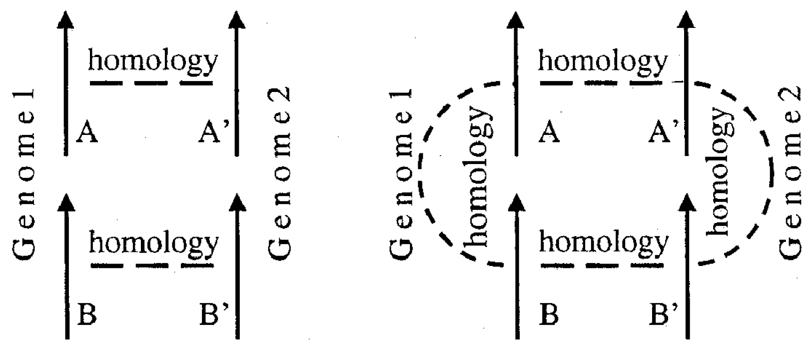
Nucleic Acids Research 33:880-892, 2005

Accurate operon prediction

- Estimation of the likelihood that two adjacent genes are contained within the same transcriptional unit (TU).
  - Based on genome sequence
  - Free from parameters
  - Validation by comparison with microarray expression profiles

Definitions

A *conserved gene pair* is defined as two adjacent genes (A,B) for which a homologous gene pair (A`, B`) can be found in another genome, such that A is homologous to A`, B is homologous to B`, and the pair (A`,B`) are adjacent.



A pair is *not* considered *conserved* if the similarity between A and B is higher than the similarity between A and A' or B and B'.

Genes in conserved same-strand pairs are candidates for membership in the same operon.

How to estimate probability that genes in a conserved same-strand pair belong to the same operon?

Principles

- The key elements
  - use both comparative and distance information
  - infer a genome-specific distance model from preliminary comparative-only predictions

- The Key assumption:
  - The greater conservation of adjacency for genes on the same-strand of DNA, compared to opposite-strand pairs, is entirely due to operons.
  - i.e. not-operon pairs and opposite-strand pairs have the same distribution of values for the comparative and functional features

Features for adjacent genes

- For each pair of adjacent genes on the same strand calculate:
  - *distance*
    - the number of base pairs separating the two genes,

Features for adjacent genes

- For each pair of adjacent genes on the same strand calculate:
  - *distance*
    - the number of base pairs separating the two genes,
  - *comparative features*
    - how often their orthologs are near each other (within 5 kb) in other genomes,

Features for adjacent genes

- For each pair of adjacent genes on the same strand calculate:
  - *distance*
    - the number of base pairs separating the two genes,
  - *comparative features*
    - how often their orthologs are near each other (within 5 kb) in other genomes,
  - *functional similarity*
    - whether their predicted functions are in the same category [from COG (Clusters of Orthologous Groups)]

Features for adjacent genes

- For each pair of adjacent genes on the same strand calculate:
  - *distance*
    - the number of base pairs separating the two genes,
  - *comparative features*
    - how often their orthologs are near each other (within 5 kb) in other genomes,
  - *functional similarity*
    - whether their predicted functions are in the same category [from COG (Clusters of Orthologous Groups)]
  - *similarity of CAI*
    - the similarity of their codon adaptation index (CAI), a measure of synonymous codon usage

Features for adjacent genes

- For each pair of adjacent genes on the same strand calculate:
  - *distance*
    - the number of base pairs separating the two genes,
  - *comparative features*
    - how often their orthologs are near each other (within 5 kb) in other genomes,
  - *functional similarity*
    - whether their predicted functions are in the same category [from COG (Clusters of Orthologous Groups)]
  - *similarity of CAI*
    - the similarity of their codon adaptation index (CAI), a measure of synonymous codon usage

- (Calculate features separately for closely and distantly related genomes)

Some definitions

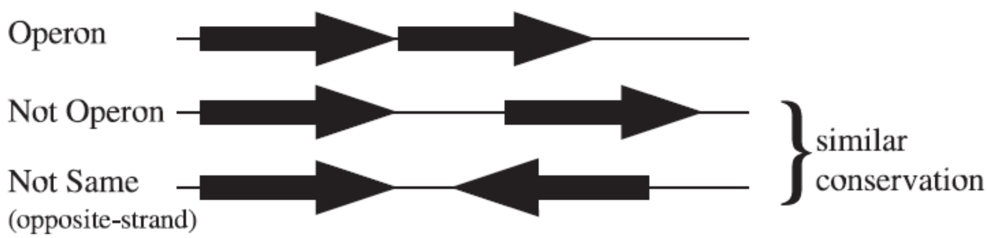
- $P(\text{Operon}|\text{Value})$ – probability that two adjacent genes are in the same operon given the value of the feature
- $P(\text{Operon}|\text{Same})$ – proportion of operon pairs on the same-strand
- $P(\text{Same}|\text{Value})$ – probability of having two genes at the same strand, given a feature
- …

Statistical Inference (1)

- Estimate probability that two adjacent genes are in the same operon given their sequences and the values of the features;
- Use assumptions to infer distributions of the comparative and functional features for operon and not-operon pairs (adjacent genes)
  - mixture of the distributions:
    - not-operon pairs (all adjacent genes from opposite strands)
    - operon pairs (some adjacent genes from same strand; unknown)

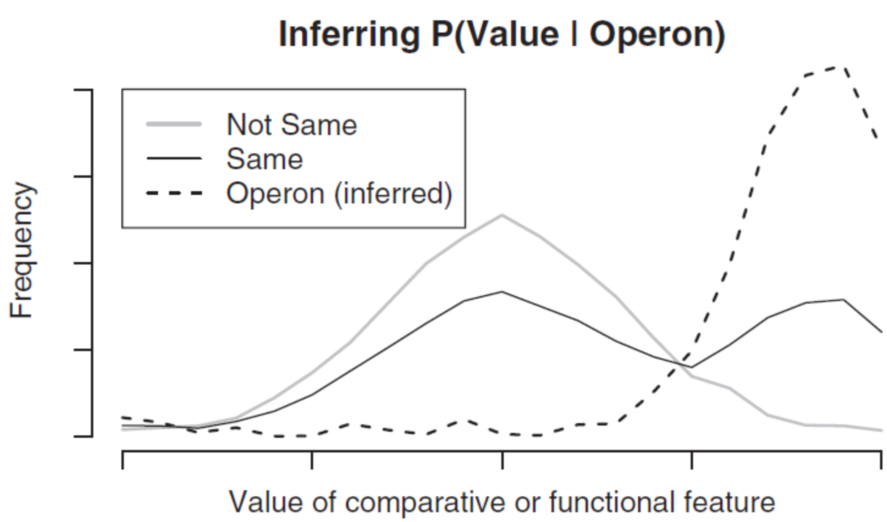
Estimation of P(Value|Operon)

- Estimate P(Value|Operon) for operon pairs
  - "subtract" out the contribution from not-operon pairs



Three types of pairs of adjacent genes and the key assumption.

Inferring P(Value|Operon) from P(Value|Same) and P(Value|NotSame).



Likelihood ratio estimation

Assumption: $P(\text{Values} \mid \text{NotOperon}) \approx P(\text{Values} \mid \text{NotSame})$

$$P(\text{Values} \mid \text{Same}) = P(\text{Values} \mid \text{Operon}) + P(\text{Values} \mid \text{NotOperon})$$

Same – same-strand vs. opposite-strand pairs
Value – a comparative/ functional feature

Likelihood ratio estimation

Assumption: $P(\text{Values} \mid \text{NotOperon}) \approx P(\text{Values} \mid \text{NotSame})$

$$P(\text{Values} \mid \text{Same}) = P(\text{Values} \mid \text{Operon}) + P(\text{Values} \mid \text{NotOperon})$$

Same – same-strand vs. opposite-strand pairs
Value – a comparative/ functional feature

Application of a long sequence of mathematical reformulations (mainly Bayes Theorem)…

$$\frac{P(\text{Values} \mid \text{Operon})}{P(\text{Values} \mid \text{NotOperon})} \approx \frac{\frac{P(\text{NotSame})}{P(\text{Same})} \cdot \frac{P(\text{Same} \mid \text{Values})}{P(\text{NotSame} \mid \text{Values})} - P(\text{NotOperon} \mid \text{Same})}{P(\text{Operon} \mid \text{Same})}$$

A genome-specific distance model

- Split the pairs into those with *high* and *low* comparative/functional likelihood ratios
  - treat these as preliminary operon predictions
  - false positive error rate of the predictions equals the fraction of opposite-strand gene pairs ‘predicted’ to be in the same operon

$$P(\text{High} \mid \text{NotOperon}) \approx P(\text{High} \mid \text{NotSame})$$

A genome-specific distance model

- Split the pairs into those with *high* and *low* comparative/functional likelihood ratios
  - treat these as preliminary operon predictions
  - false positive error rate of the predictions equals the fraction of opposite-strand gene pairs 'predicted' to be in the same operon

$$P(\mathrm{High} \mid \mathrm{NotOperon}) \approx P(\mathrm{High} \mid \mathrm{NotSame})$$

Bayes…

$$P(\mathrm{NotOperon} \mid \mathrm{High}) \approx \frac{P(\mathrm{High} \mid \mathrm{NotSame}) \cdot P(\mathrm{NotOperon} \mid \mathrm{Same})}{P(\mathrm{High} \mid \mathrm{Same})}$$

$$P(\mathrm{Operon} \mid \mathrm{Same}) = P(\mathrm{Operon} \mid \mathrm{High}) \cdot P(\mathrm{High} \mid \mathrm{Same}) + P(\mathrm{Operon} \mid \mathrm{Low}) \cdot P(\mathrm{Low} \mid \mathrm{Same})$$

Overall prediction

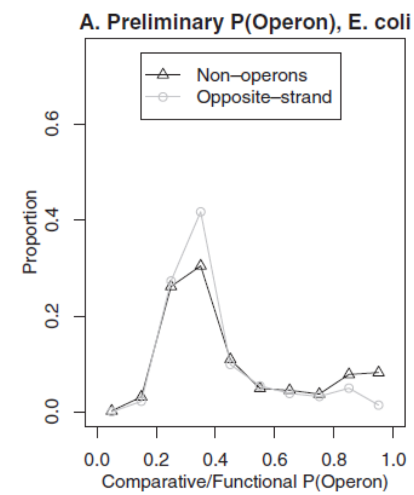
$$\frac{P(\text{Operon} \mid \text{AllFeatures})}{P(\text{NotOperon} \mid \text{AllFeatures})} =$$

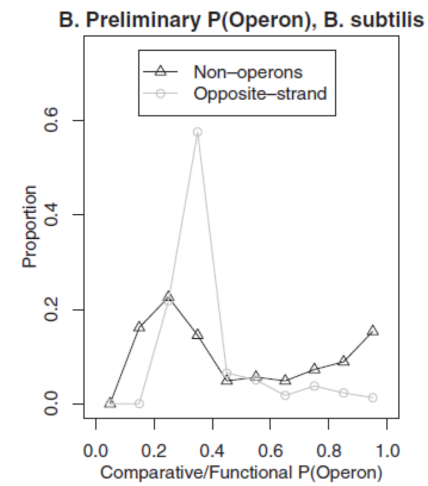
$$\frac{P(\text{Operon} \mid \text{Same})}{P(\text{NotOperon} \mid \text{Same})} \cdot \frac{P(\text{Values} \mid \text{Operon})}{P(\text{Values} \mid \text{NotOperon})} \cdot \frac{P(\text{Distance} \mid \text{Operon})}{P(\text{Distance} \mid \text{NotOperon})} \cdot \frac{P(\text{CAI} \mid \text{Operon})}{P(\text{CAI} \mid \text{NotOperon})}$$

(The assumption of conditional independence of the comparative/functional features and the similarity of CAI is approximately true.)

Test of key assumption

- Conservation of 'known' not-operon pairs.
  - The distribution of preliminary estimates of P(Operon), using only the comparative and functional features, for opposite-strand pairs and 'known' not-operon pairs in (A) *E. coli K12* and (B) *B. subtilis*.
  - In *B. subtilis* there is a predominance of highly conserved genes (present in many other genomes) in this small data set used. → an explanation of the peak at 0.25 for known not-operons
- The modest deviations from the assumption are due to co-transcription of the 'known' not-operon pairs





Accuracy for known transcripts

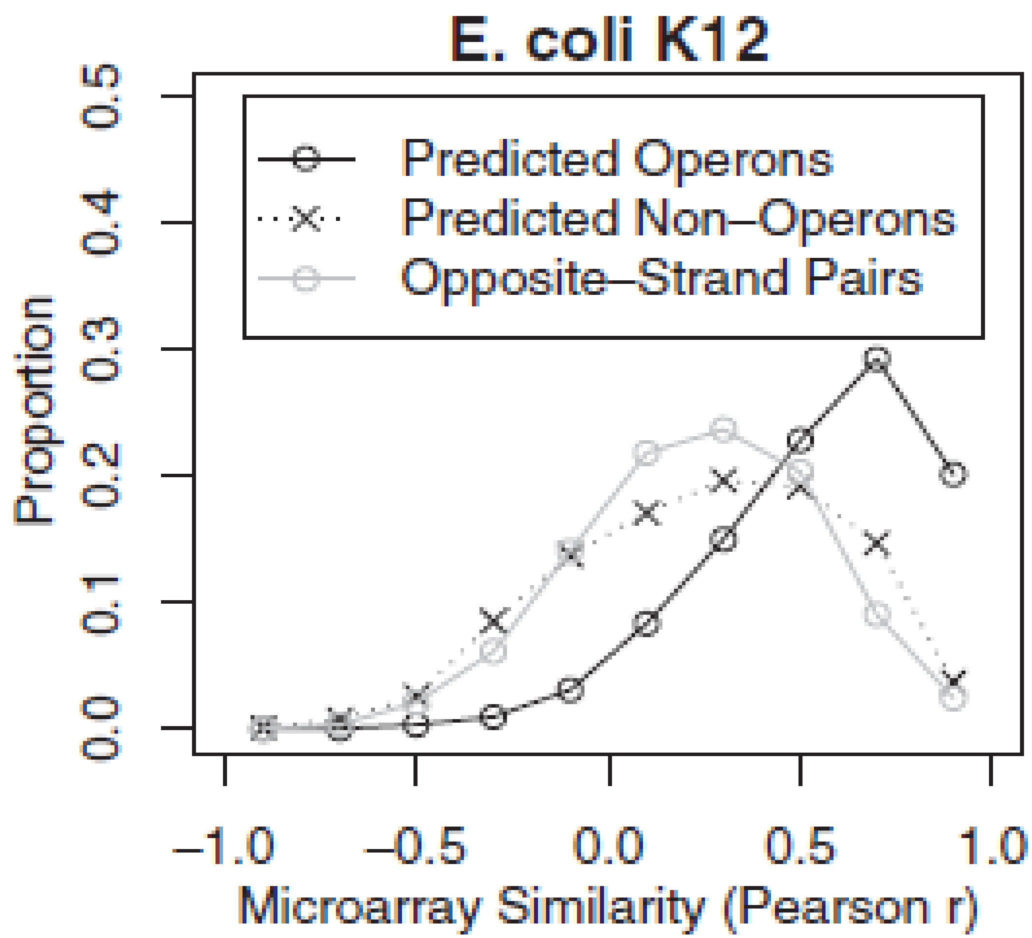
- Threshold:
  - P(Operon|AllFeatures) > 0.5

	E.coli	B.Subtilis
Sensitivity (TP), %	88.3	79.9
Specificity (TN), %	90.9	71.0
AOC unsupervised	0.920	0.919
AOC supervised	0.888	0.907

Accuracy for microarray data (1)

6 species

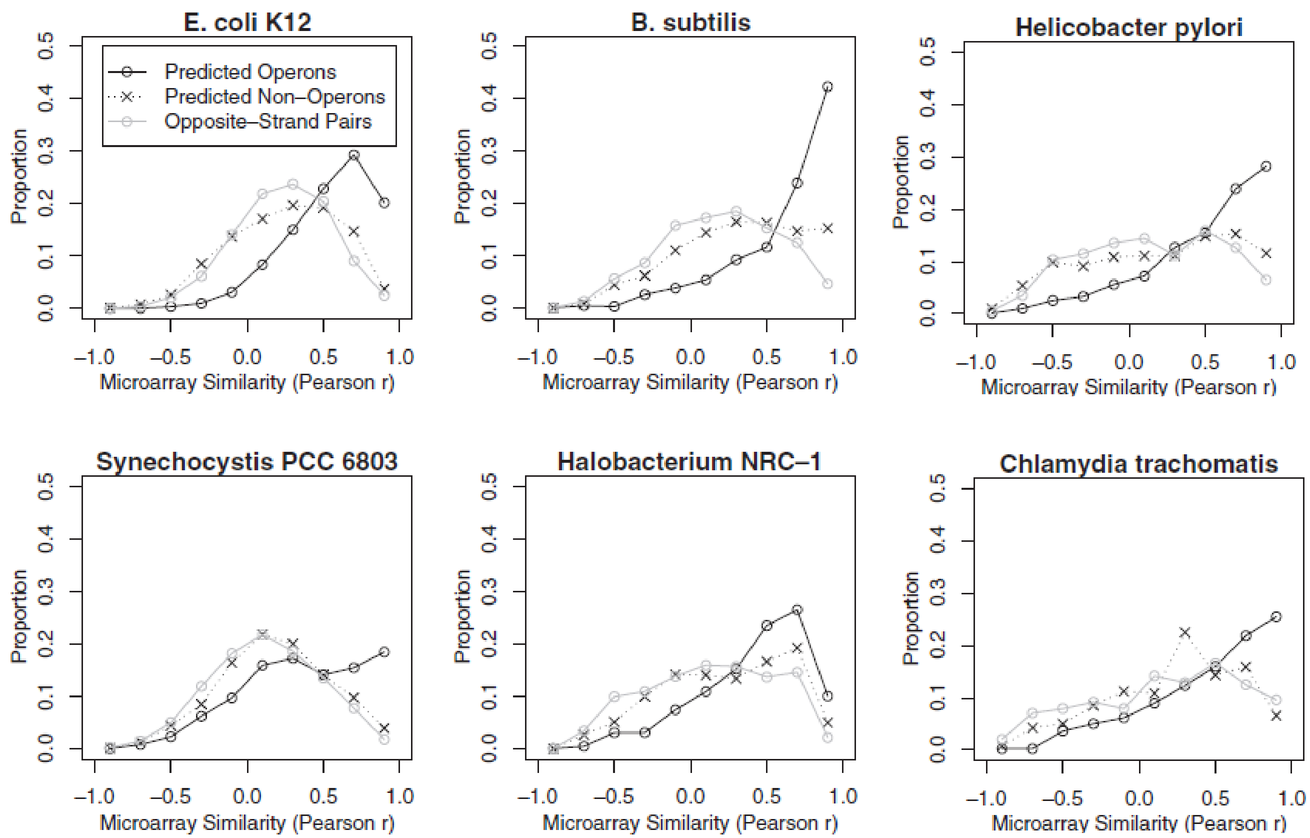
- distribution of adjacent gene pairs having a certain Pearson correlation
- strong co-expression, relative to other adjacent pairs on the same strand, for predicted operon pairs
- little co-expression for predicted not-operon pairs (similar to opposite-strand pairs)



Accuracy for microarray data (I)

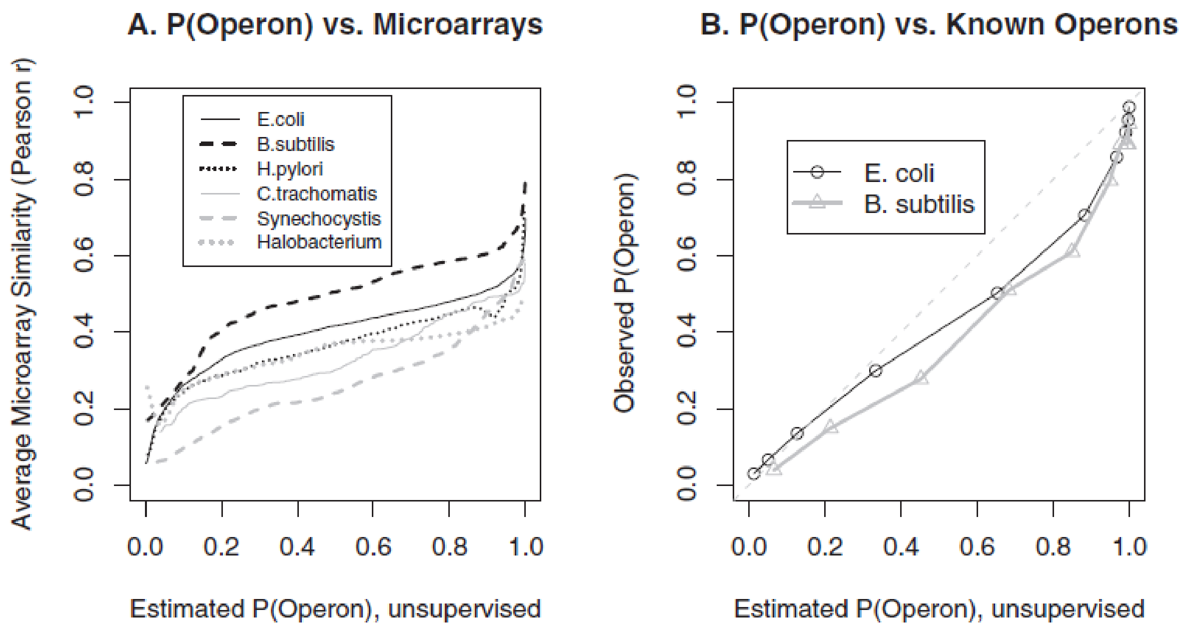
6 species

- distribution of adjacent gene pairs having a certain Pearson correlation
- strong co-expression, relative to other adjacent pairs on the same strand, for predicted operon pairs
- little co-expression for predicted not-operon pairs (similar to opposite-strand pairs)



Accuracy for microarray data (2)

- *P(*Operon|AllFeatures*)* is consistent with microarray data and with known operons



Feature contribution

- Genome-specific models – the majority of the agreement between predictions and gene expression data

Genome	Distance	Comparative	All features
E. coli K12	0.406	0.401	0.494
B. subtilis	0.420	0.335	0.461
H. pylori	0.275	0.231	0.343
C. trachomatis	0.260	0.167	0.303
Synechocystis	0.159	0.222	0.268
Halobacterium	0.198	0.159	0.215

- Distance prediction identifies new operons, in comparison with comparative genomics alone
- CAI – little effect on the final predictions (not shown)

Some results and findings:

- Accurate unsupervised prediction of operons
- Combination of comparative genomics and genome-specific distance models

- Accuracy of 85% for *E. coli* and 83% for *B. subtilis*
- *H. pylori* has many operons, contrarily to previous reports
- *Bacillus anthracis* has an unusual number of pseudogenes within conserved operons
- *Synechocystis PCC 6803* has many operons even though it has unusually wide spacings between conserved adjacent genes

References:

[Schuster-Böckler et al., 2004] Schuster-Böckler B., Schultz J. Rahmann S. HMM Logos for visualization of protein families, BMC Bioinformatics 2004, 5:7doi: 10.1186/1471-2105-5-7

[Price et al., 2005] Price M.N., Huang K.H., Alm E.J., Arkin A.P. A Novel Method for Accurate Operon Predictions in All Sequenced Prokaryotes. Nucleic Acids Research 33:880-892, 2005

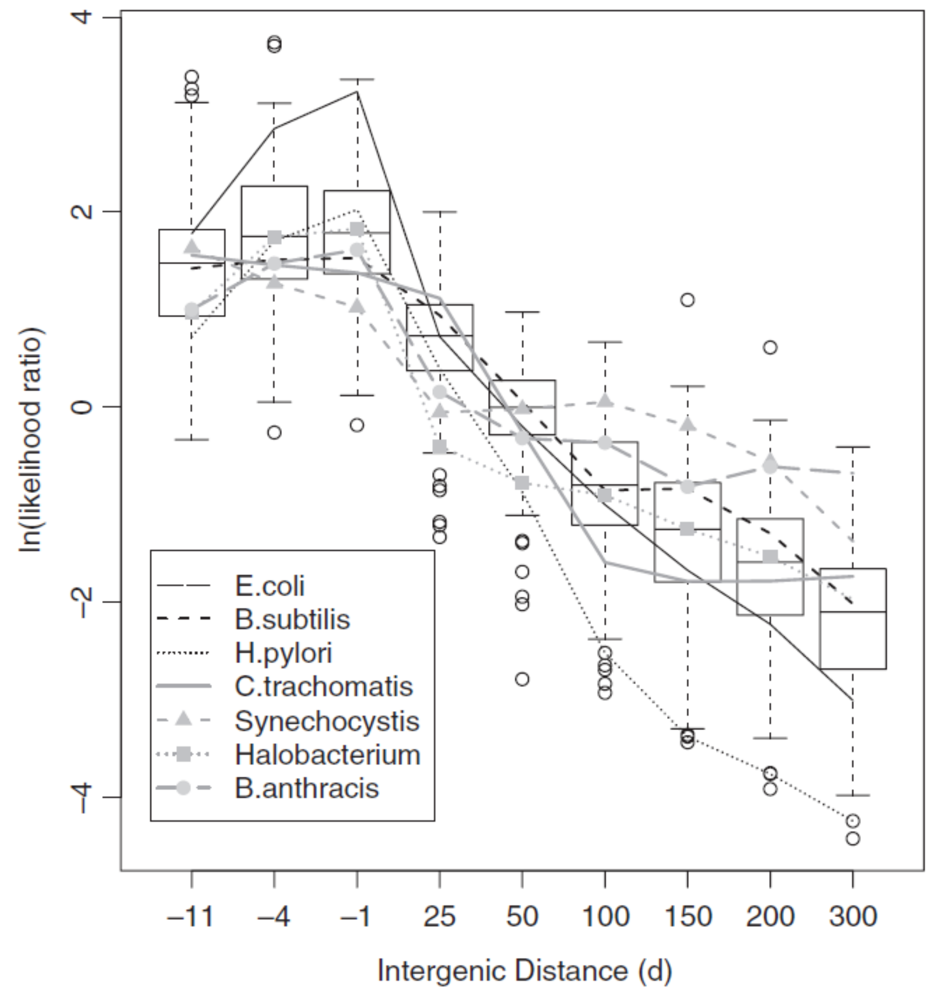
[Ermolaeva et al., 2001] Ermolaeva, M. D., White O., Salzberg S. L. Prediction of operons in microbial genomes. Nucleic Acids Res. 29:1216–1221, 2001

Request www.google.com for “wikipedia”

Thank you!

Distance models across 124 genomes

- Boxplots, across 124 genomes, of the genome-specific log-likelihoods *ln(*P(Distance|Operon*)/ P(*Distance|NotOperon*))* at the indicated distances.
  - where the log likelihood is zero, operon and not-operon pairs are predicted to be equally likely to have that distance.
  - the boxes show quartiles and medians
  - whiskers extend up to 1.5 times the interquartile range from the box,
  - dots show outlying genomes
  - the non-linear x-axis highlights the sharp peak around the common separations of -1 and -4
  - distance models for a few specific genomes are shown with lines
- Although most genomes follow the same trend of more operons at lower separations, significant differences are seen in the shape and magnitude of their distance models



Parameter Estimation: Maximum *a posteriori*

Maximum *a posteriori* (MAP) of the model:

$$\mathrm{Prob}(\mathrm{model} \mid \mathrm{sequences}) = \frac{\mathrm{Prob}(\mathrm{sequences} \mid \mathrm{model})\, \mathrm{Prob}(\mathrm{model})}{\mathrm{Prob}(\mathrm{sequences})}$$

Find a most likely model (the best one):

$$\boxed{\mathrm{Prob}(\mathrm{sequences} \mid \mathrm{model})\, \mathrm{Prob}(\mathrm{model}) \quad \rightarrow \quad \max}$$

Parameter Estimation: ML and MAP

Forward Backward (or Baum–Welch) algorithm

version of Expectation Maximization algorithm

- Main idea: iterative adaptation of the model to fit the data

Parameter Estimation: ML and MAP (3)

(1) Initialization: assign $T(r|q)$ and $P(x|q)$ for each x, q, r; prior knowledge, "model surgery"

(2) Get new estimate (by Brute Force or Dynamic Programming) of

- $T(r|q)$: by counting the number of times a transition is made from state q to r for all paths and all sequences
- $P(x|q)$: by counting the number of times the AA x is aligned to the state q

(3) Replace old estimates by the new ones

(4) Repeat steps (2) and (3) until convergence

Multiple Alignment: the Viterbi Algorithm (1)

Calculate negative logarithm of the probability of the single most likely path for the sequence:

$$\text{Given a model:}\quad -\log \max_{paths} \text{Prob}(\text{s}, \text{path} \mid \text{model})$$

$$\text{dist}(s, \text{model}) = \min_{paths}\{-\log \text{Prob}(s, \text{path} \mid \text{model})\}$$

$$= \min_{paths} \sum_{i=1}^{N+1} [-\log T(q_i \mid q_{i-1}) - \log P(x_{l(i)} \mid q_i)]$$

- For multiple alignment, align each sequence to the model by the Viterbi algorithm.

Multiple Alignment: the Viterbi Algorithm (2)

$$\text{dist}(s, \text{model}) = \min_{paths} \sum_{i=1}^{N+1} [-\log T(q_i \mid q_{i-1}) - \log P(x_{l(i)} \mid q_i)]$$

Positions dependent penalties:

Transition from

match to delete state = **gap-initiation penalty**
delete to delete state = **gap-extension penalty**
match to insert state = **insertion-initiation penalty**
insert to insert state = **insertion-extension penalty**

and

penalty for aligning the AA to the position

An HMM example (I)

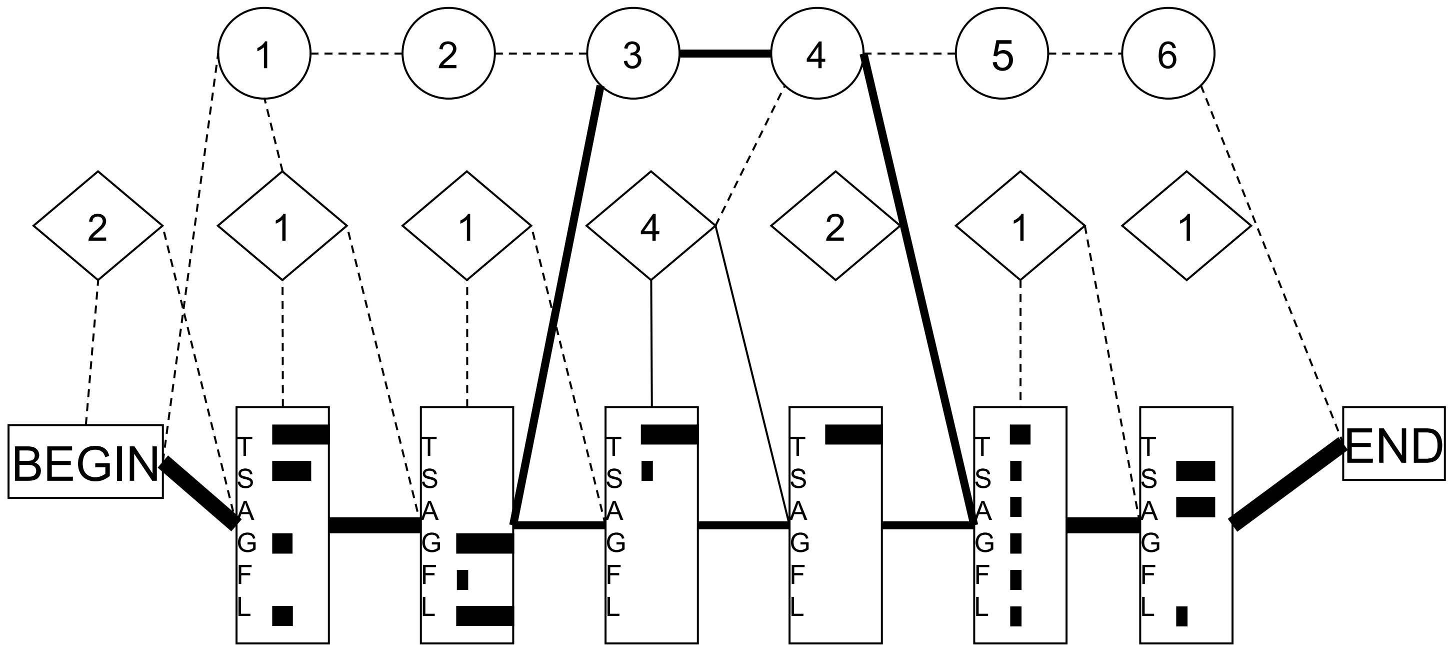
An artificial HMM model.

Thickness of a line indicates what fraction of the training sequences made that transition or used that particular AA.

A broken line indicates that less than 5% of the sequences used that transition.

Numbers in the insertion states shows the average length of an insertion beginning at that position.

Numbers in the deletion states shows the index number of that position.



An HMM example (2)

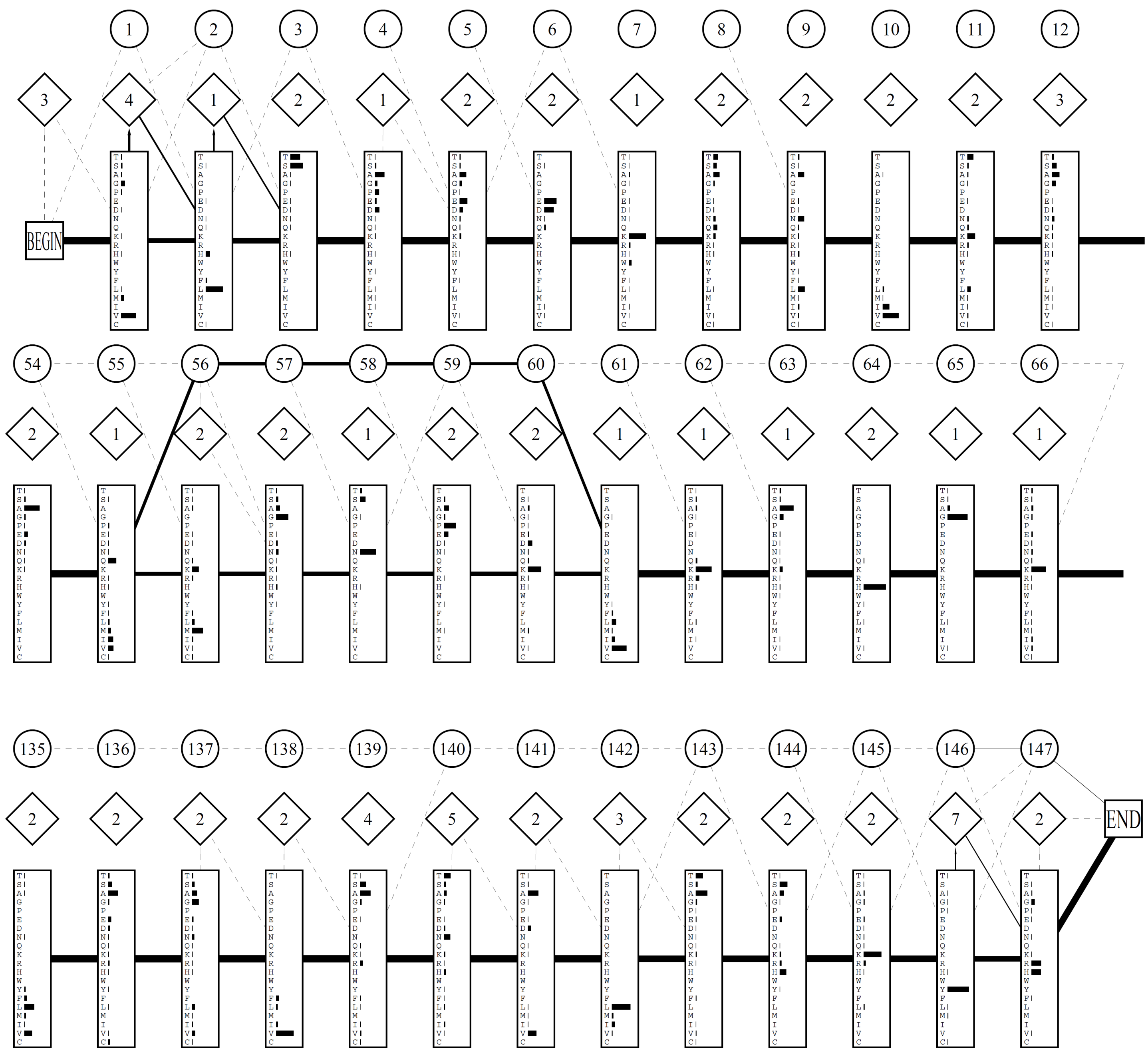
Parts of a globin model alignment.

Thickness of a line indicates what fraction of the 400 training sequences made that transition or used that particular AA.

A broken line indicates that less than 5% of the sequences used that transition.

Numbers in the insertion states shows the average length of an insertion beginning at that position.

Numbers in the deletion states shows the index number of that position.



HMM Logos: example

Visualization of subfamily-specific sites:

Comparison of the HMM Logos of the small GTPases Ras and Rab from SMART.

Arrows indicate subfamily specific sites RabF2 to RabF5.