SAM-T98

Building and using an HMM framework for finding protein homologs and predicting protein structure

Melissa Cline, Christian Barrett, and Kevin Karplus

University of California, Santa Cruz

http://www.cse.ucsc.edu/research/compbio/

Acknowledgments

- SAM-T98 developed by Kevin Karplus and Christian Barrett Additional collaborators: Richard Hughey, David Haussler, Melissa Cline, Mark Diekhans, Leslie Grate, Kimmen Sjölander.
- SAM hidden Markov model software package developed by Richard Hughey and Anders Krogh [KBM+94, HK96].
- The sequence logo program used is by A. M. Amer, UCSC, based on the Sequence Logos program by Tom Schneider [SS90].
- Supported in part by NSF grant DBI-9408579, DOE grant DE-FG0395ER62112, GAANN fellowship program, and a grant from Digital Equipment Corporation.

Outline of tutorial

- External (web-site) view of SAM-T98. (Karplus)
- Overview of tutorial and SAM-T98. (Barrett)
- Entropy and sequence logos. (Cline)
- Introducing the running examples. (Barrett)
- Basics of profile HMMs. (Karplus)
- SHORT BREAK.
- Sequence weighting. (Cline)
- Column regularizers. (Karplus)
- Transition Regularizers. (Barrett)
- SHORT BREAK.
- Searching with the model. (Barrett)
- Null models. (Karplus)
- Stepping through the process. (Cline)
- Bidirectional scoring. (Barrett)
- Future directions and conclusions. (Barrett)

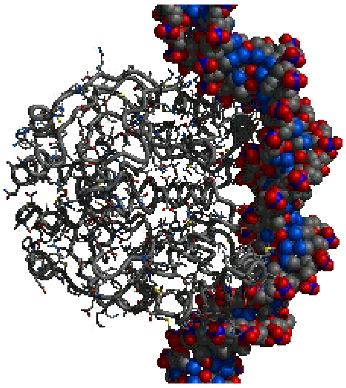
This talk is a "look under the hood" at the SAM-T98 method.

- Inputs: User supplies a *target* protein sequence or a trusted multiple alignment.
- Outputs:
  - Scores and pairwise alignments from aligning the target to a set of HMMs that we have built for known structures.
  - A multiple alignment of all similar sequences from a large non-redundant protein database.
  - A hidden Markov model (HMM) for searching databases or aligning sequences.
  - Search results and alignments for searching PDB (or Swissprot or NR) with the HMM.

SAM-T98 master web page

`http://www.cse.ucsc.edu/research/compbio/HMM-apps/HMM-applications.html`

UCSC HMM Applications



UCSC's SAM-T98 method for iterative SAM HMM construction and remote homology detection includes extensions to our CASP2 work (html) for our CASP3 predictions. A description of the method will appear in BioInformatics (html and postscript), and is favorably evaluated against PSI-BLAST and other methods in J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia, ''Sequence Comparisons Using Multiple Sequences Detect Twice as many Remote Homologues as Pairwise Methods'', JMB 284(4):1201-1210, 1998. Please cite SAM-T98 and this page as:

Kevin Karplus, Christian Barrett, Richard Hughey, "Hidden Markov Models for Detecting Remote Protein Homologies", Bioinformatics 14(10):846-856, 1998. WWW server available from http://www.cse.ucsc.edu/research/compbio.

Basic operations

Compare Sequence Against Protein Model Library
Search UCSC's protein structure HMM library with a single sequence. May reveal homologous structures.

Protein Query Against A Database
Protein homology search. This server creates a SAM-T98 multiple alignment from a single sequence, builds an HMM from it, and then scores the selected database using the HMM. The SAM-T98 alignment is not returned, but a selected number of hits from the database are returned as a multiple alignment.

Create or Tune Up a Multiple Alignment The user submits sequences or a multiple alignment and gets back a multiple alignment.

Compare Two Alignments
Submit two alignments in FASTA format and receive a score measuring the similarity of the alignments.

SAM-T98 search web page

Because the method can take a while for long proteins, or ones with many homologs, results are e-mailed back. They also don't fit on a slide—see T98-query.mail in handouts.

`http://www.cse.ucsc.edu/research/compbio/`
`HMM-apps/T98-query.html`

HMM-based Sequence Query

This page offers the ability to perform remote protein homolog searches using HMM technology. It is part of a larger suite
of UCSC's HMM applications that are available on the web. If you are familiar with BLAST searching, then you understand the functionality of this page.

Step 1: Enter email address and optional subject line

Please enter you email address (required) `karplus@cse.ucsc.edu`
Return email subject line (optional) `UCSC SAM-T98-query results`

Step 2: Enter one sequence

You may either cut and paste your sequence in the box below or upload it from a file on your local system. Sequences must be in a readseq-compatible format.
Cut and paste your sequence here, ...
`>2crd Charybdotoxin  XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS`

or specify the sequence-containing file on your system here.

Step 3: Choose database to search for homologs

PDB (quick)
SWISSPROT (a couple hours)
971 SCOP domains

Step 4: Choose threshold for reporting database hits

If you want to include more remote, or weaker scoring, sequences in the final report, choose a cutoff value that is less negative. The more negative the cutoff, the more strict you are being about what you will consider as a "hit".
NLL-NULL score cutoff: -9

A somewhat simpler page just computes the SAM-T98 multiple alignment and returns it.

`http://www.cse.ucsc.edu/research/compbio/`
`    HMM-apps/build-target98-alignment.html`

See build-target98-alignment.mail in the handouts.

Overview

What we are talking about today: practical application of hidden Markov models presented in the framework of SAM-T98.

- Criteria for accepting homologs
- Regularizers and weighting schemes
- Speeding up database searches
- Scoring and null models
- Library building

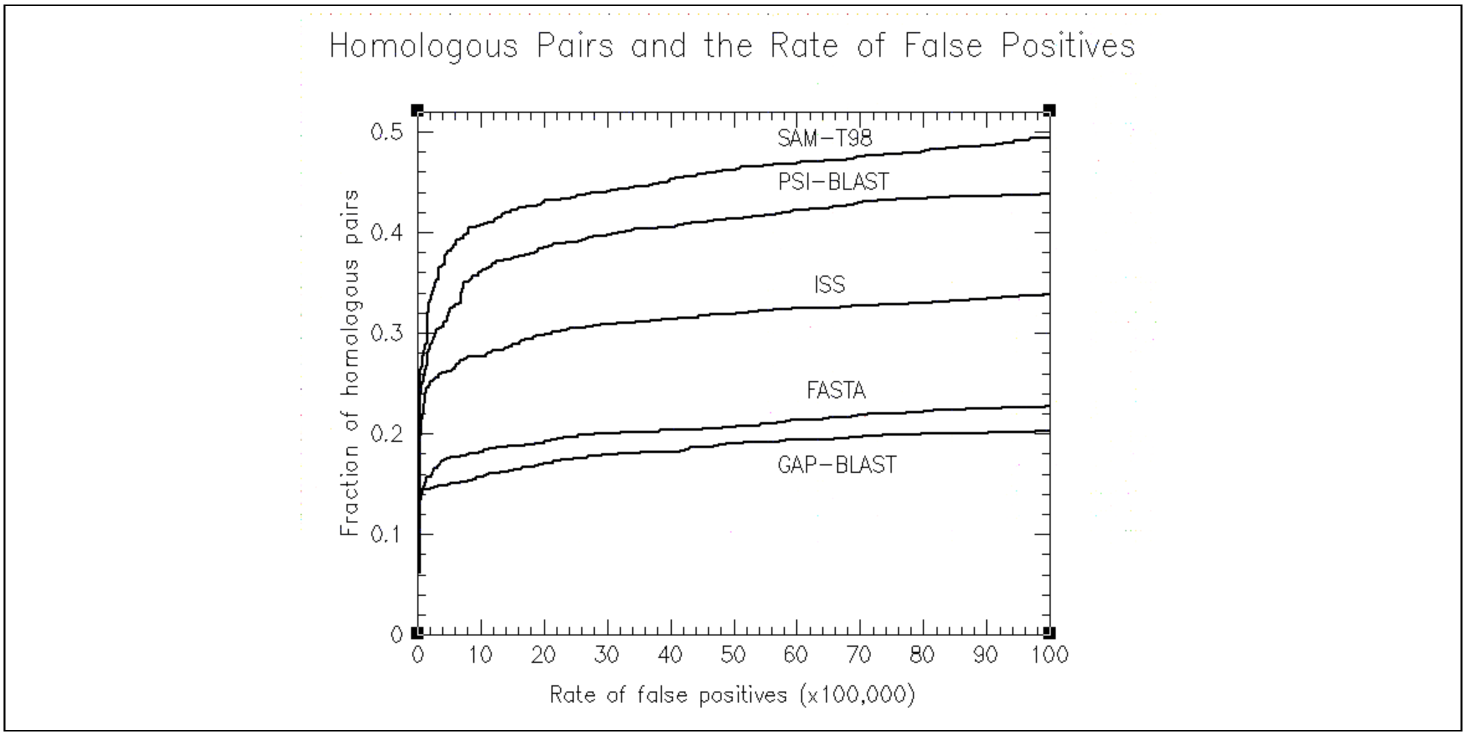
What we are **not** talking about today:

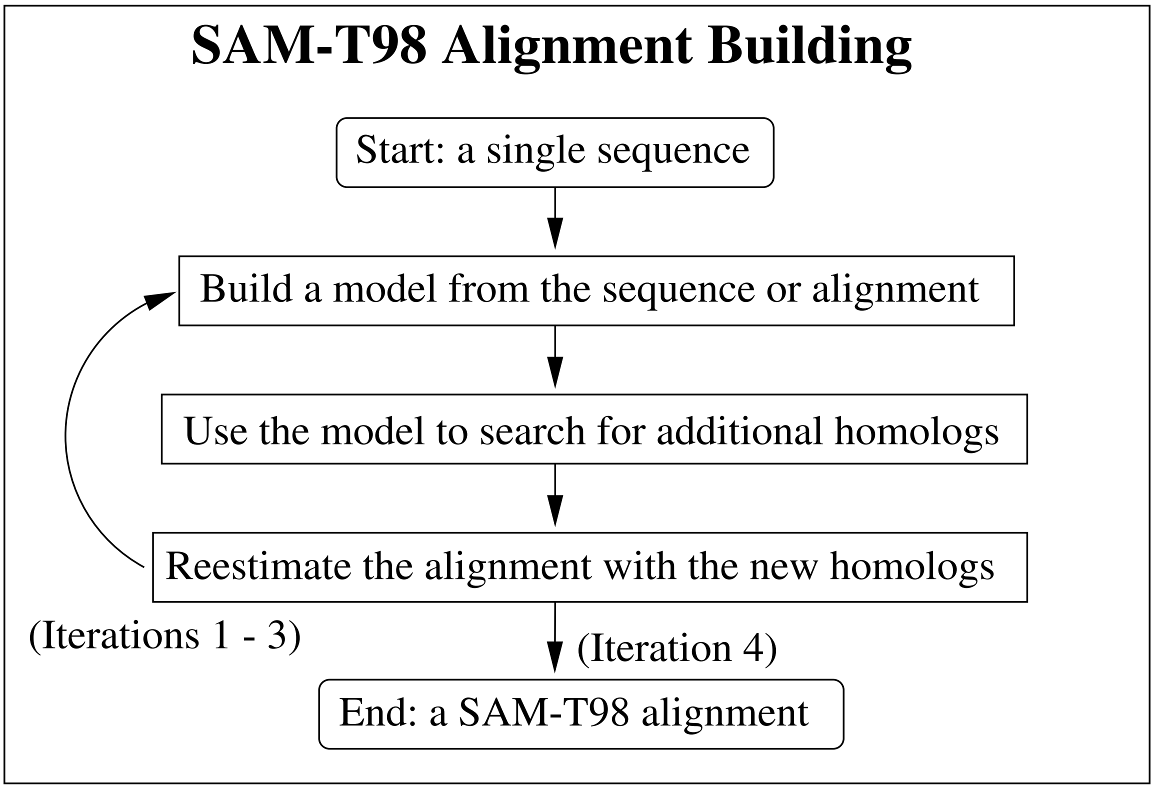
- Mathematical foundations of hidden Markov models
- Objectives of fold recognition
- Other applications of HMMs and stochastic modeling

What is SAM-T98?

SAM-T98 is an HMM-based homology detection method, similar to PSI-BLAST in function.

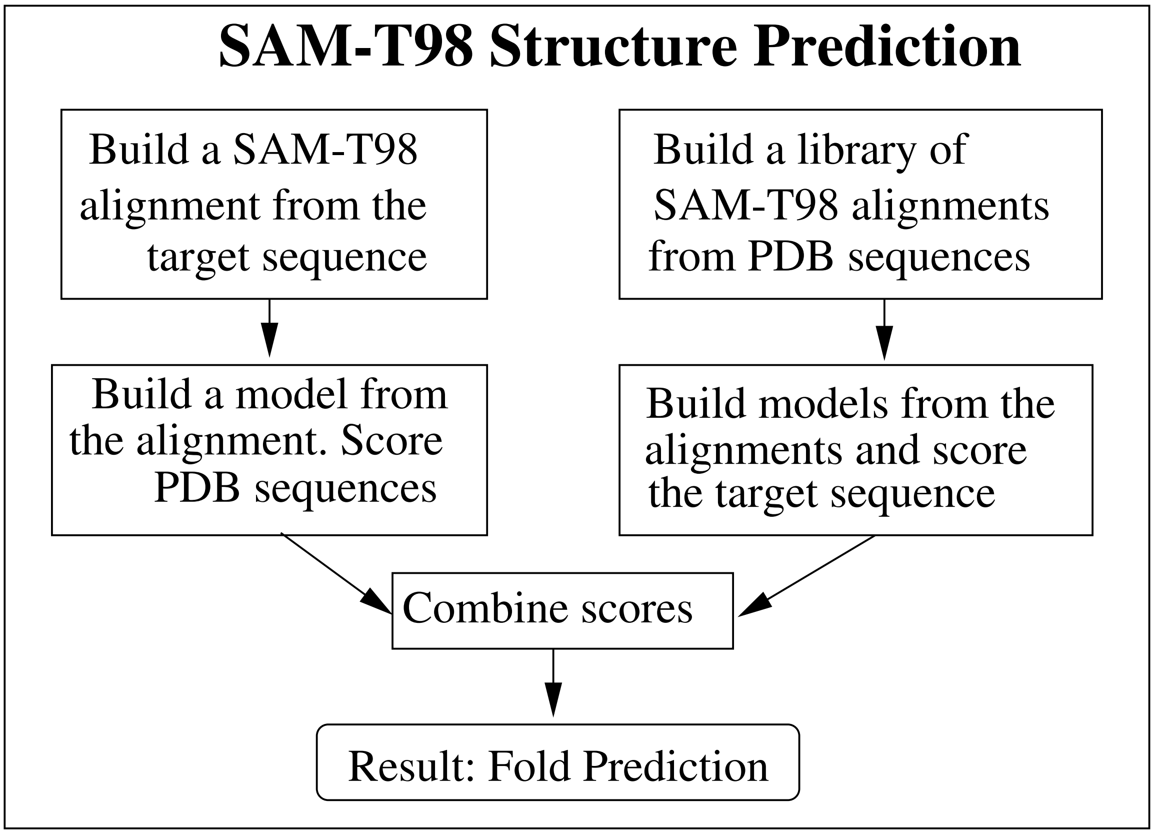
- SAM-T98 was the top fold recognition and homology modeling method at CASP3 of those using no direct structural information [KBC+99].
- The methodology that developed into SAM-T98 was one of the top fold recognition methods at CASP2 [KKB+97].
- In two detailed comparisons using SCOP [HMBC97], SAM-T98 was shown to outperform methods including PSI-BLAST, ISS, and Double BLAST at superfamily recognition [PKB+98, KBH98]





Starting point: A seed sequence or **reliable** seed alignment.

Goal: Build an alignment that encompasses the breadth of the seed's family.



Starting point: A target sequence of unknown structure.

Goal: Identify the structural family of the target sequence through identification of a homolog whose structure is known.

Review of Entropy

- In statistical thermodynamics, the entropy of a system describes the following:
  - Disorder
  - Log of the number of microstates in the system

  The second definition corresponds closely to the use of the term in information theory.

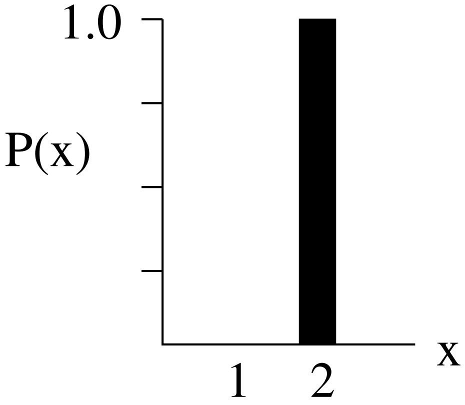
- The entropy of a random variable x is defined as $H(x) = -\Sigma_{x_i} P(x_i) \log_2 P(x_i)$ (bits).
- Uses of entropy include
  - Estimating the minimum expected size of a codeword needed for encoding or compressing data (*encoding cost*).
  - Measuring the information content or complexity of a probability distribution

- We will use entropy to measure the information in the columns of a multiple alignment, given their amino acid distributions.

Interpretation of entropy

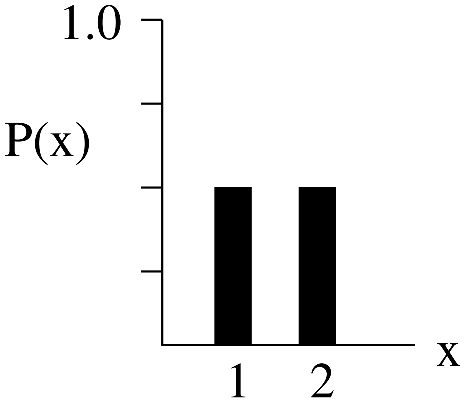
- Formula: $H(x) = -\sum_{x_i} P(x_i) \log_2 P(x_i)$ (bits).

- Entropy is minimized with a peaked distribution.



$$\begin{aligned} H(x) &= -\sum_{x_i} P(x_i) \log_2 P(x_i) \\ &= -0 \log_2(0) - 1 \log_2(1) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

- Entropy is maximized with a uniform distribution, achieving a maximum value of $\log_2(N)$, where N is the number of values that variable x can assume.



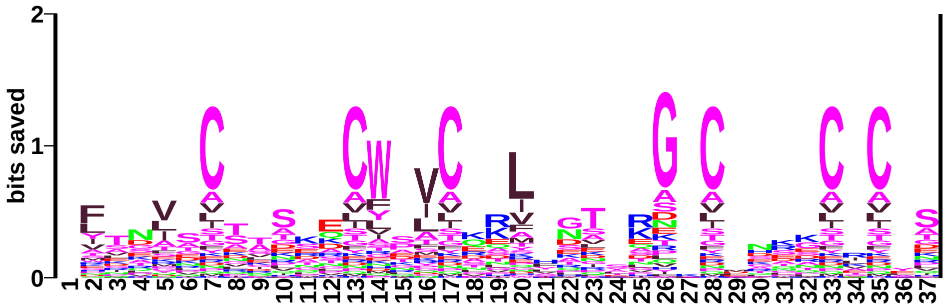
$$\begin{aligned} H(x) &= -\sum_{x_i} P(x_i) \log_2 P(x_i) \\ &= -(2 \times 0.5 \log_2(0.5)) \\ &= -2 \times 0.5 \times -1.0 \\ &= 1.0 \end{aligned}$$

Entropy of Alignment Columns

- Consider some alignment column c before c contains any amino acids.
  - Each amino acid is expected according to some *prior* or background probability P_0.
  - P_0 has some entropy H_0, also referred to as the *background entropy*.
- Consider column c when it contains some amino acids.
  - Column c has some *posterior* amino acid distribution P_c computed from the amino acid observations plus priors.
  - H_c, the entropy of column c, is computed from the posterior distribution P_c.
  - If column c is very conserved, then H_c approaches zero. If column c is very diverse, then H_c approaches H_0.
  - We refer to $H_0 - H_C$ as the *bits saved* at column c. A large savings indicates a strong alignment signal.

- The logos shown were created using a modified version of the program by Tom Schneider [SS90]—we compute the posterior probabilities differently.
- Each column's logo reflects its amino acid posterior distribution, computed with amino acid priors and sequence weighting.

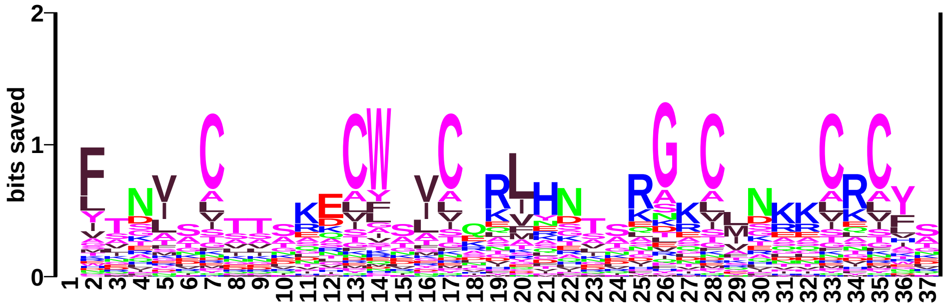
```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



Sequence Logos and entropy

- The height of each character is proportional to the log of the posterior probability of that amino acid in that column.
- The total height of each column represents the bits saved relative to the background entropy.
- The maximum possible savings at any column is approximately $\log_2(20) = 4.3$.
- The logo for an alignment reflects the signal that will be modeled by an HMM or other model trained on the alignment.
- The logo shown reflects the 2crd sequence using the Blosum62 regularizer. Its signal reflects what BLAST would search for when using Blosum62.

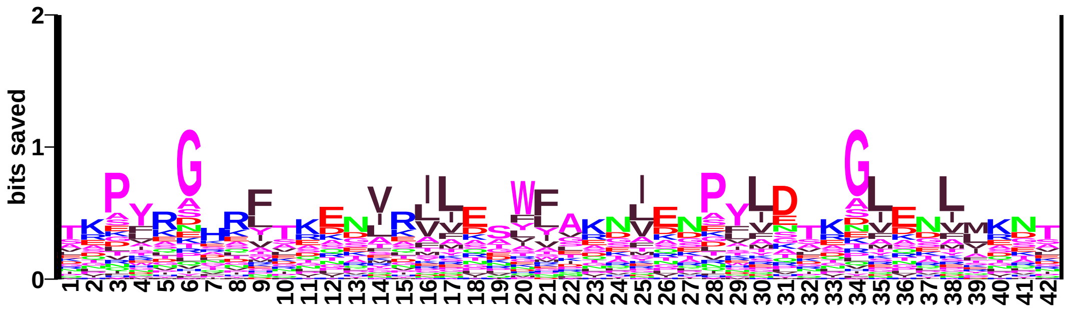
```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
```



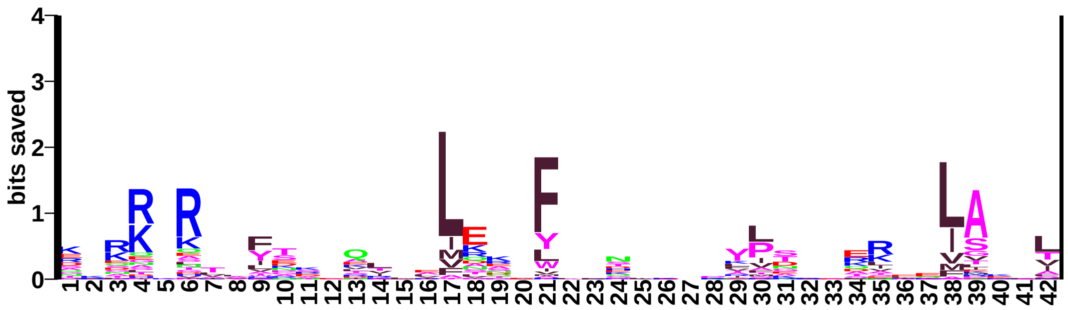
Interpreting Sequence Logos

- The logos shown reflect excerpts from alignments of 1yrnB. The upper logo shows the 1yrnB sequence only. The lower logo shows an alignment of approximately 2000 sequences.

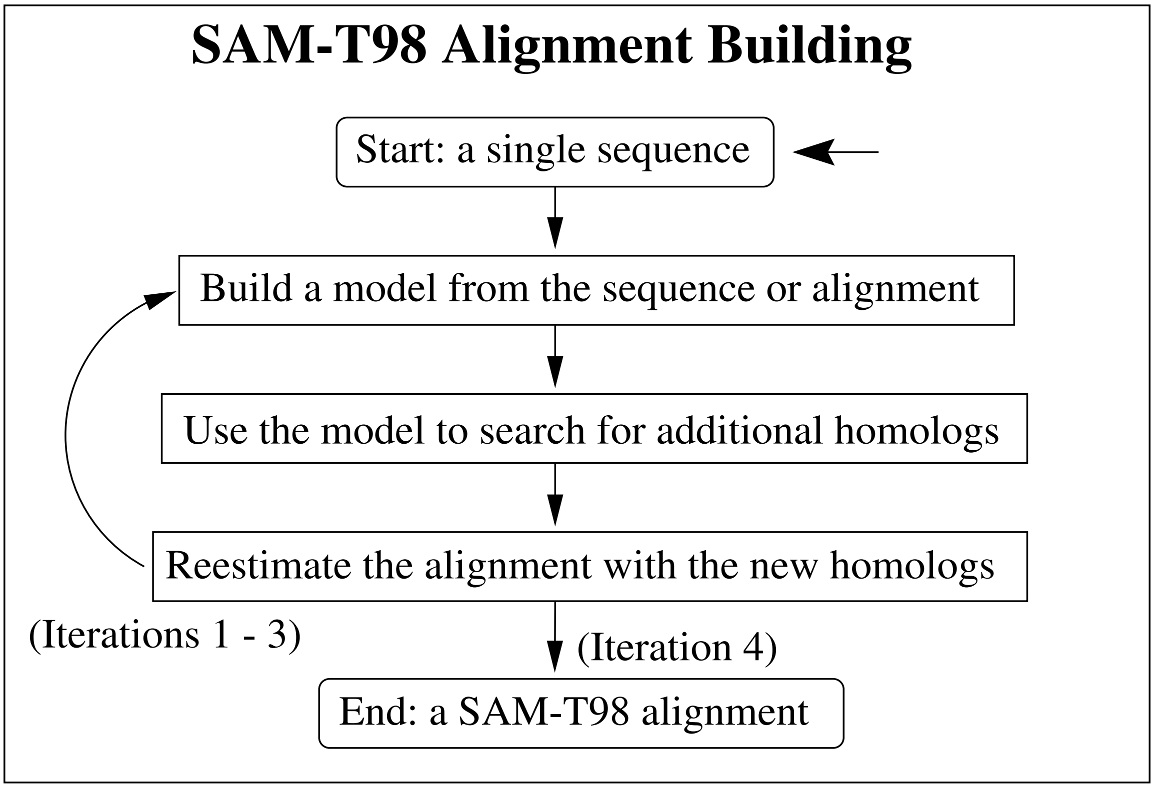
- If an alignment has a small number of sequences, the signal will be dominated by the priors, and most columns will have similar savings and a strong background signal.



- If an alignment contains numerous, diverse sequences, the observed data will dominate the signal. Some columns will show very strong savings, while others will show almost none.



Starting out



To illustrate the steps taken by SAM-T98, we will take two sequences through the full prediction process:

1. 2crd *Charybdotoxin*: small, cysteine-rich, neurotoxin that blocks K^+ channels.
2. 1yrnB *MAT α2 homeodomain* that regulates expression of cell type-specific genes in yeast.

These sequences have known structures, but we will ignore their structural information until the prediction process is complete.

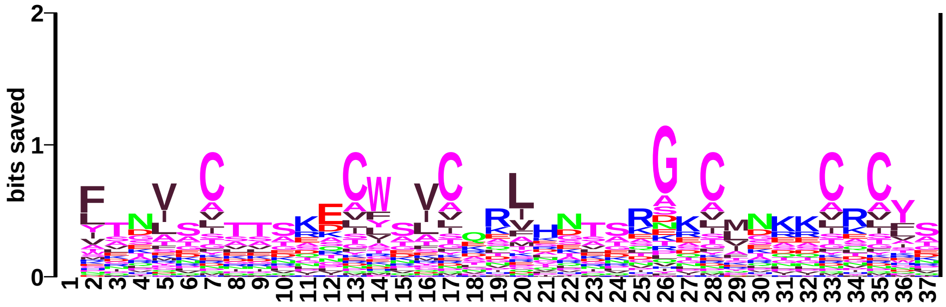
Example 1: 2crd

What information do we know initially?

- Sequence: `XFTNVSCTTS KECWSVCQRL HNTSRGKCMN KKCRCYS`

Other information, which we do not use:

- Is a member of a structural class of toxins that share the same motif but block different K^+/Na^+ channels.
- Disulfides C_7-C_{28}, C_{13}-C_{33}, C_{17}-C_{35} are critical for all class members.
- Activity is mediated through (different) positively charged residues that interact with negative charges on the channels.
- Logo below depicts the information in the regularized profile constructed from the single 2crd sequence.
- A BLAST search would be looking for approximately this signal.
- Through the tutorial, this signal will change as we make it more specific and sensitive.



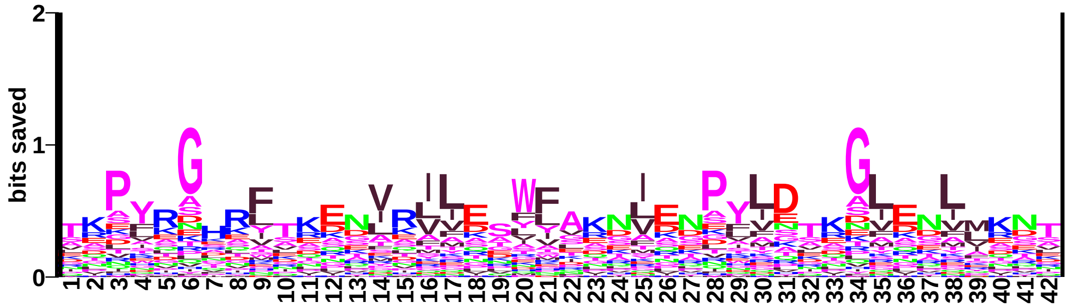
Example 2: 1yrnB

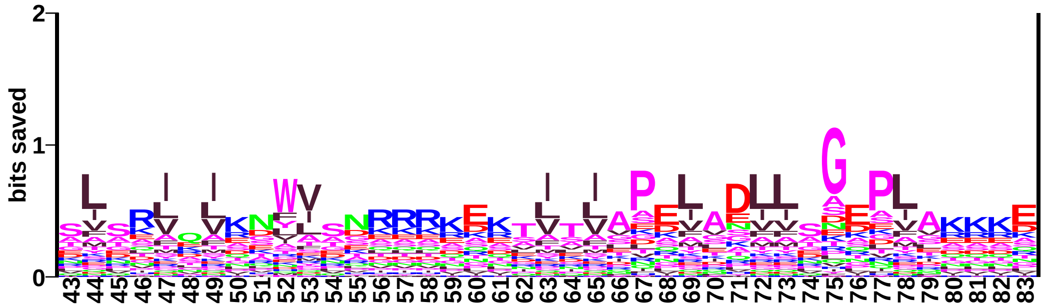
What do we know initially?

- Sequence: `TKPYRGHRFT KENVRILESW FAKNIENPYL DTKGLENLMK NTSLSRIQIK NWVSNRRRKE KTITIAPELA DDLSGEPLAK KKE`

Other information, which we do not use:

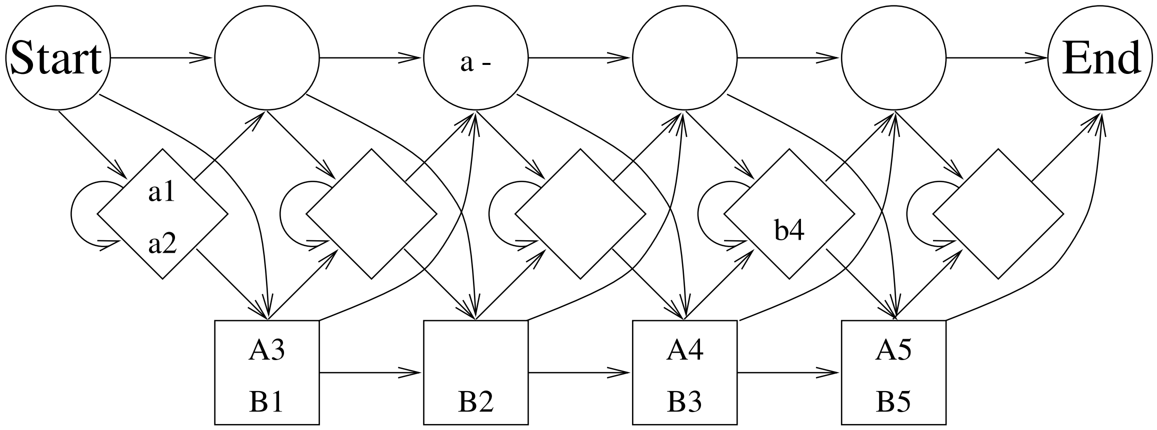
- Binds sequence-specific DNA through a helix-turn-helix motif and N-terminal tail, primarily through positively-charged residues and the sugar-phosphate backbone.
- Logo below depicts the information in the regularized profile constructed from the single 1yrnB sequence.





- A *hidden Markov Model (HMM)* is a finite-state machine with a probability for emitting each letter in each state, and with probabilities for making each transition between states.
- Probabilities of letters sum to one for each state.
- Probabilities of transitions out of each state sum to one for that state.
- We also include *null states* that emit no letters, but have transition probabilities on their out-edges.

Profile Hidden Markov Model




```
a1 a2 A3    –    A4  .  A5
 .  . B1   B2    B3 b4  B5
```

- Circles are null states.
- Squares are *match states*, each of which is paired with a null *delete state*. We call the match-delete pair a *fat state*.
- Each fat state is visited exactly once on every path from Start to End.
- Diamonds are *insert states*, and are used to represent possible extra amino acids that are not found in most of the sequences in the family being modeled.

Computing probabilities with an HMM

- Each path from Start to End generates as many letters as there are match and insert states on the path.
- Let path S be Start $\rightarrow s_1 \rightarrow \cdots \rightarrow s_n \rightarrow$ End, and let $x_1, \ldots, x_m$ be the letter-generating states of S ($m \leq n$).
- We can compute $\text{Prob}\,(A \mid M)$ as a sum over all paths that have $\text{len}\,(A)$ letter-generating states:

$$\text{Prob}\,(A \mid M) = \sum_{\text{paths } S} \text{Prob}\,(A \mid M, S)\, \text{Prob}\,(S \mid M) \ .$$

- The probability of the path depends only on the transition parameters:

$$\text{Prob}\,(S \mid M) = \prod_{i=0}^{n} \text{Prob}\,(s_{i+1} \mid M, s_i) \ .$$

- The probability of the sequence given the path depends only on the emission parameters of the letter-generating states used:

$$\text{Prob}\,(A \mid M, S) = \prod_{i=1}^{\text{len}(A)} \text{Prob}\,(A_i \mid M, x_i) \ .$$

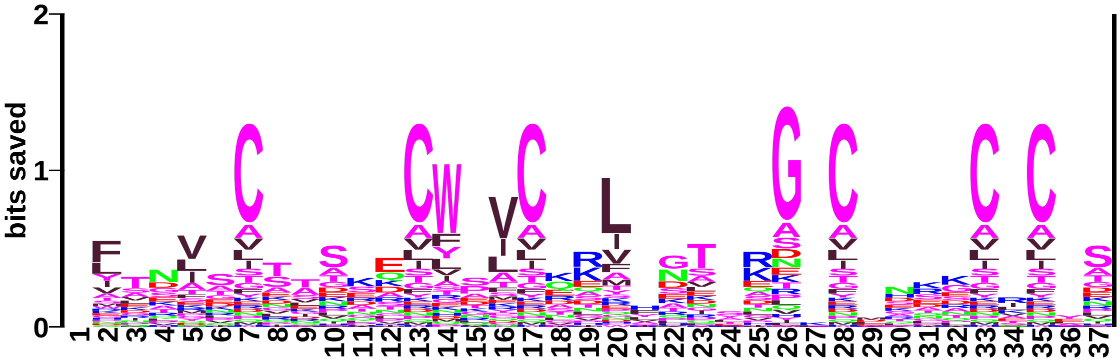
Generating a model from an alignment

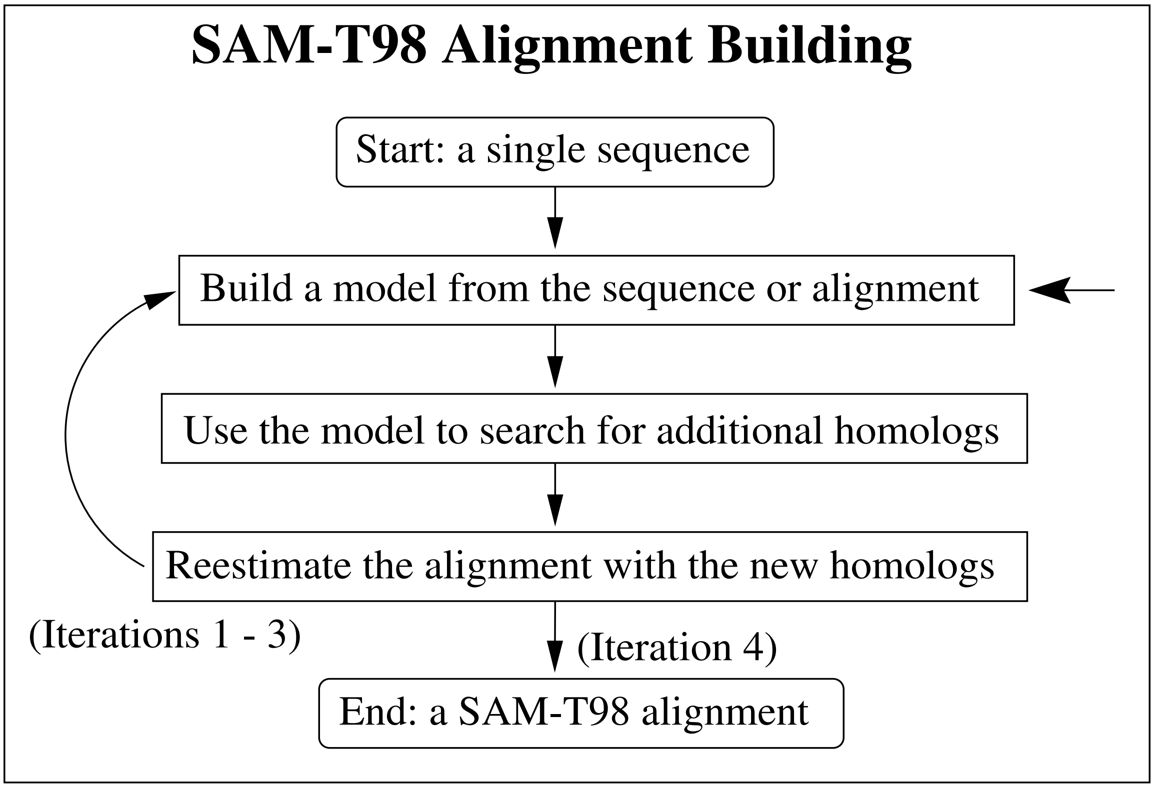
Goal: given an alignment, build a model to match the aligned sequences, and as-yet-unseen sequences similar to those aligned.

From this:

```
2crd XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```

To this:





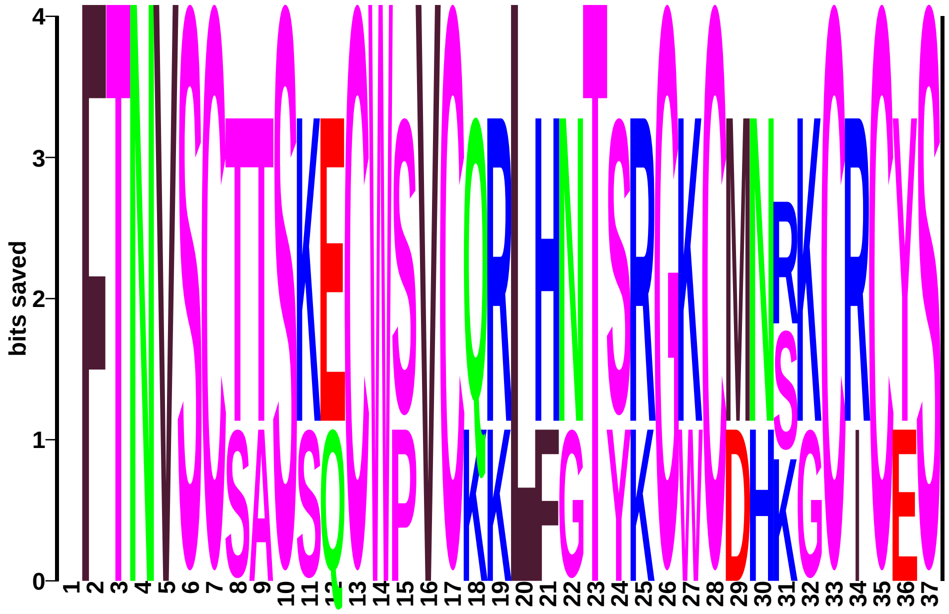
A model that is trained on an alignment should match the aligned sequences, plus other sequences similar to those aligned. This requires adding generality to the model.

Risks of not generalizing well

The logo shown is computed from the following alignment, with no generalizing. Remember that a logo reflects the signal that a model trained on the alignment will look for.

```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```

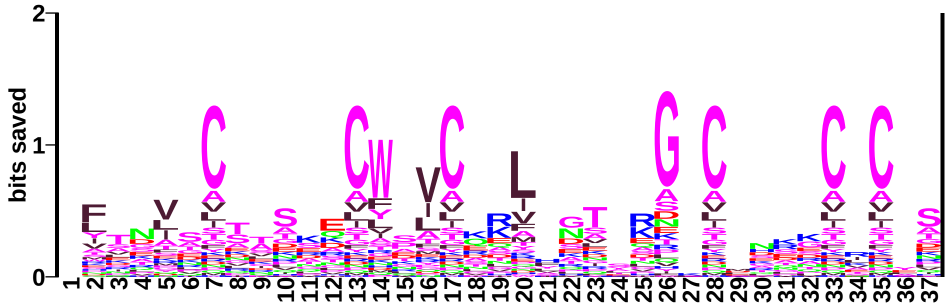
- All of the signal comes from the training alignment, even though this alignment consists of only three close homologs.
- In all columns, any amino acid not yet seen is expected with a probability of zero.
- Such a model would find only very close homologs at best.



Effects of generalizing

The logo shown below is computed from the same alignment as the previous logo, but using a more sensible generalizing scheme.

- The signal reflects the alignment, but the alignment does not dominate the signal.
- The conserved cysteines and the other structurally significant columns are emphasized.
- The amino acid posteriors reflect the amino acids seen plus their likely substitutions and structural significance.



SAM-T98 adds generality from three specific sources:

1. Sequence Weighting

2. Column Regularizers

3. Transition Regularizers

```
2crd  XFTNVSCTTSKECWSVCQRLHNTSRG.KCMNKKCRCYS
1bah  XFTNVSCTTSKEXWSVCQRLHNTSRG.KCMNKKXRCYS
1sxm  TIINVKCTSPKQCSKPCKELYGSSAGaKCMNGKCKCYNnx
1cmr  ------CTTSKECWSVCQRLHNTSKG.WCDHRGCICES
1txm  ----VSCTGSKDCYAPCRKQTGCPNA.KCINKSCKCYGcx
1lir  XFTQESCTASNQCWSICKRLHNTNRG.KCMNKKCRCYS
2bmt  XFTNVSCSASSQCWPVCKKLFGTYRG.KCMNSKCRCYS
1bkt  VGINVKCKHSGQCLKPCKDA-GMRFG.KCINGKCDCTPkx
1big  XFTDVKCTGSKQCWPVCKQMFGKPNG.KCMNGKCRCYS
```

Observe that in the alignment above, the first two sequences are nearly identical. Such redundancy is normal in protein families.

Problem: Training data tends to be imperfect. Training alignments can be homogeneous and skewed.

One Solution: Sequence weighting.

- First challenge: Selecting how much weight to assign each sequence relative to the others.
- Second challenge: Selecting how much total weight to assign to the sequences relative to the priors.

Henikoff Sequence Weights [HH94]

1. Each column receives a weight of 1, divided evenly between its distinct residues.
2. Within each column, the weight for each residue is divided evenly by its number of occurrences.
3. The weight of a sequence is the sum of the observation weights of its residues.

```
2crd XFT NVSC TTSKECWSVCQRLHNTSRG
1bah XFT NVSC TTSKEXWSVCQRLHNTSRG
1sxm TII NVKC TSPKQCSKPCKELYGSSAG
1cmr --- ---C TTSKECWSVCQRLHNTSKG
1txm --- -VSC TGSKDCYAPCRKQTGCPNA
1lir XFT QESC TASNQCWSICKRLHNTNRG
2bmt XFT NVSC SASSQCWPVCKKLFGTYRG
1bkt VGI NVKC KHSGQCLKPCKDA-GMRFG
1big XFT DVKC TGSKQCWPVCKQMFGKPNG
```

Residue(s)	N	Q	D	V	E	S	K	C
Counts	5	1	1	7	1	5	3	9
Residue Wt.	0.33	0.33	0.33	0.50	0.50	0.50	0.50	1.00
Observation Wt.	0.07	0.33	0.33	0.07	0.50	0.10	0.17	0.11

Henikoff Sequence Weights, Continued

1. Each column receives a weight of 1, divided evenly between its distinct residues.
2. Within each column, the weight for each residue is divided evenly by its number of occurrences.
3. The weight of a sequence is the sum of the observation weights of its residues.

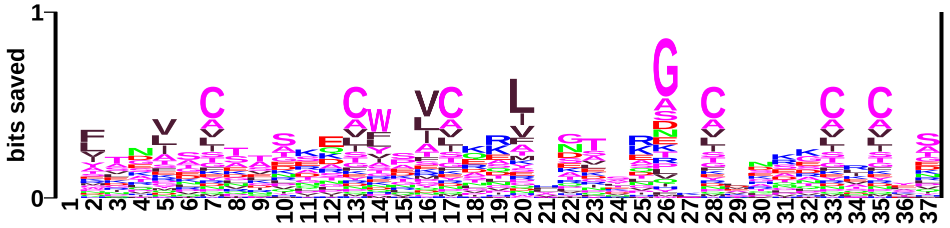
Residue(s)	N	Q	D	V	E	S	K	C
Counts	5	1	1	7	1	5	3	9
Residue Wt.	0.33	0.33	0.33	0.50	0.50	0.50	0.50	1.00
Observation Wt.	0.07	0.33	0.33	0.07	0.50	0.10	0.17	0.11

Sequence	Weight Computation	Absolute Weight	% Alignment Weight
2crd	0.07 + 0.07 + 0.10 + 0.11	0.35	8.6
1bah	0.07 + 0.07 + 0.10 + 0.11	0.35	8.6
1sxm	0.07 + 0.07 + 0.17 + 0.11	0.42	10.3
1cmr	0.11	0.11	2.7
1txm	0.07 + 0.17 + 0.11	0.35	8.6
1lir	0.33 + 0.50 + 0.10 + 0.11	1.04	25.6
2bmt	0.07 + 0.07 + 0.10 + 0.11	0.35	8.6
1bkt	0.07 + 0.07 + 0.17 + 0.11	0.42	10.3
1big	0.33 + 0.07 + 0.17 + 0.11	0.68	16.7

Effects of low total sequence weight

- The signal is dominated by the priors, and the data seen is given little emphasis over the data not seen.
  - Column 26 contains conserved glycines, but almost everything is considered likely.
  - Column 16 contains conserved valines, but leucine and isoleucine are almost as probable.
- At this total weight, most alignments will not have much emphasis on the model. Even if the alignment is diverse, its information will be muted.
- This total sequence weight is 0.71, and the average savings is 0.25 bits/column.

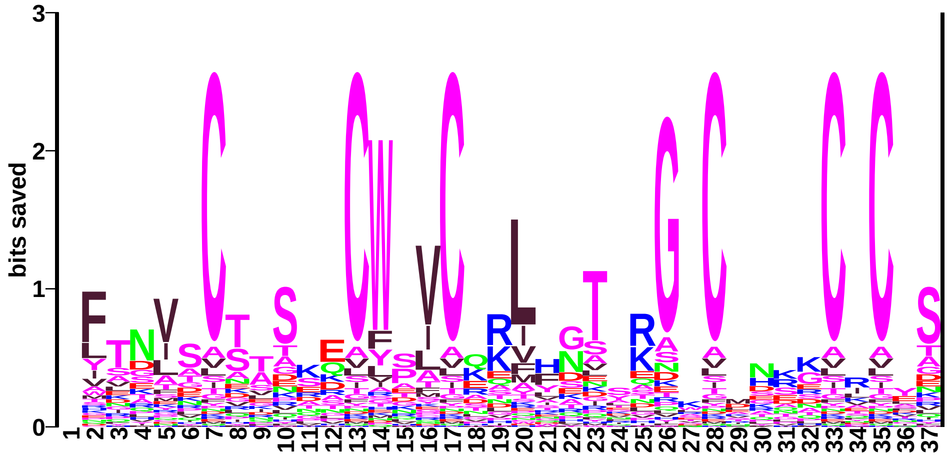
```
2crd  XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr  ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt  XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



Effects of high total sequence weight

- The signal is dominated by the training data, even though the training alignment is only three sequences.
- Tryptophan 14, Glycine 26, and the six cysteine columns are all expected with a very high probability to remain conserved, even though we can't yet say if they should be conserved.
- This total sequence weight is 1.85, and its average savings is one bit/column.

```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```

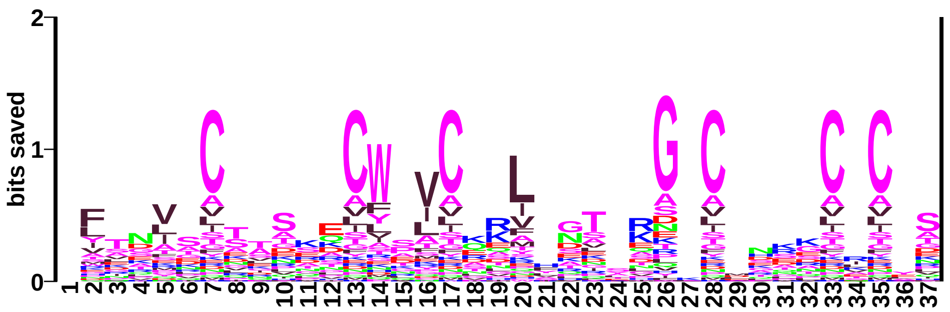


Selecting the total sequence weight

We have found our best results empirically by setting the total sequence weight to save half a bit on average relative to the background entropy H_0:

- If the sequences are all very similar, the average savings starts out high. The total sequence weight is set low to add more diversity to the signal.
- If the sequences are diverse, the average savings starts out low. The total sequence weight is set high to focus the signal more on the alignment and less on the priors.
- This total sequence weight is 1.11, and it saves an average of half a bit per column.

```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



Estimating weights to achieve the target savings

1. The sequence weights depend on the target savings.
2. The target savings depends on the column entropies.
3. The column entropies depend on the amino acid posteriors.
4. The amino acid posteriors depend on the sequence weights.

As one might suspect, an iterative procedure is required.

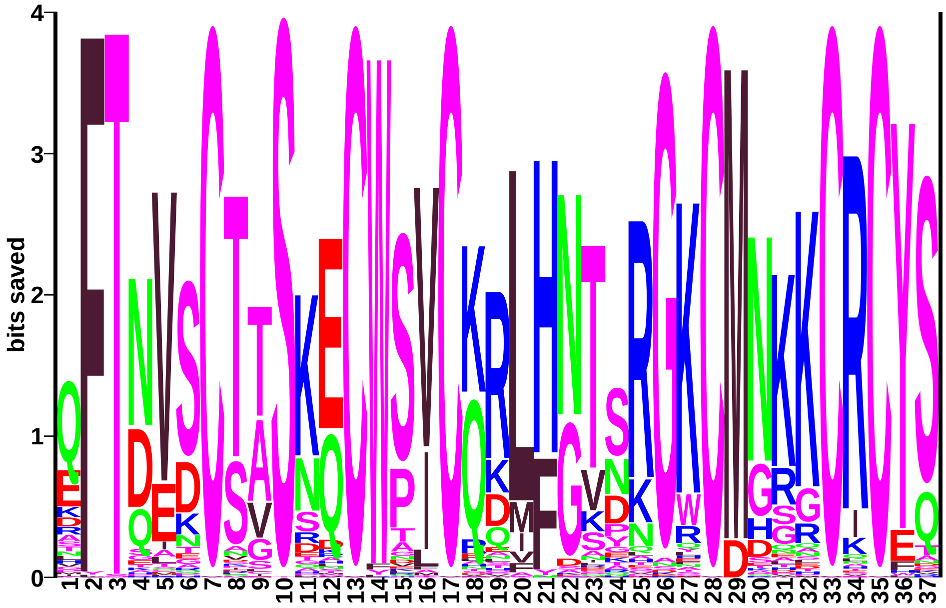
The total sequence weight is set by an algorithm that starts with an initial total weight, computes the average savings, and modifies the total weight to approach the target savings.

- All sequences are assigned a weight between 0.0 and 1.0.
- The target savings cannot always be achieved precisely, especially when the target savings is too high.

Effects of sequence weighting

Using the alignment below as input, the logo reflects the alignment with column regularizing but no sequence weighting.

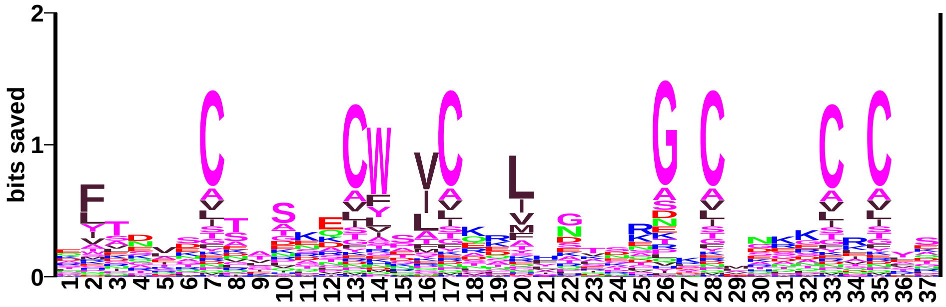
```
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....QFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....QFTQESCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....EFTDVDCSVSKECWSVCKDLFGVDRGKCMGKKCRCYQ
.....QFTDVDCSVSKECWSVCKDLFGVDRGKCMGKKCRCYQ
siegrQFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....-----SCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
.....XFTNVSCTTSKEXWSVCQRLHNTSRGKCMNKKXRCYS
.....XFTQESCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
.....XFTDVKCTGSKQCWPVCKQMFGKPNGKCMNGKCRCYS
```



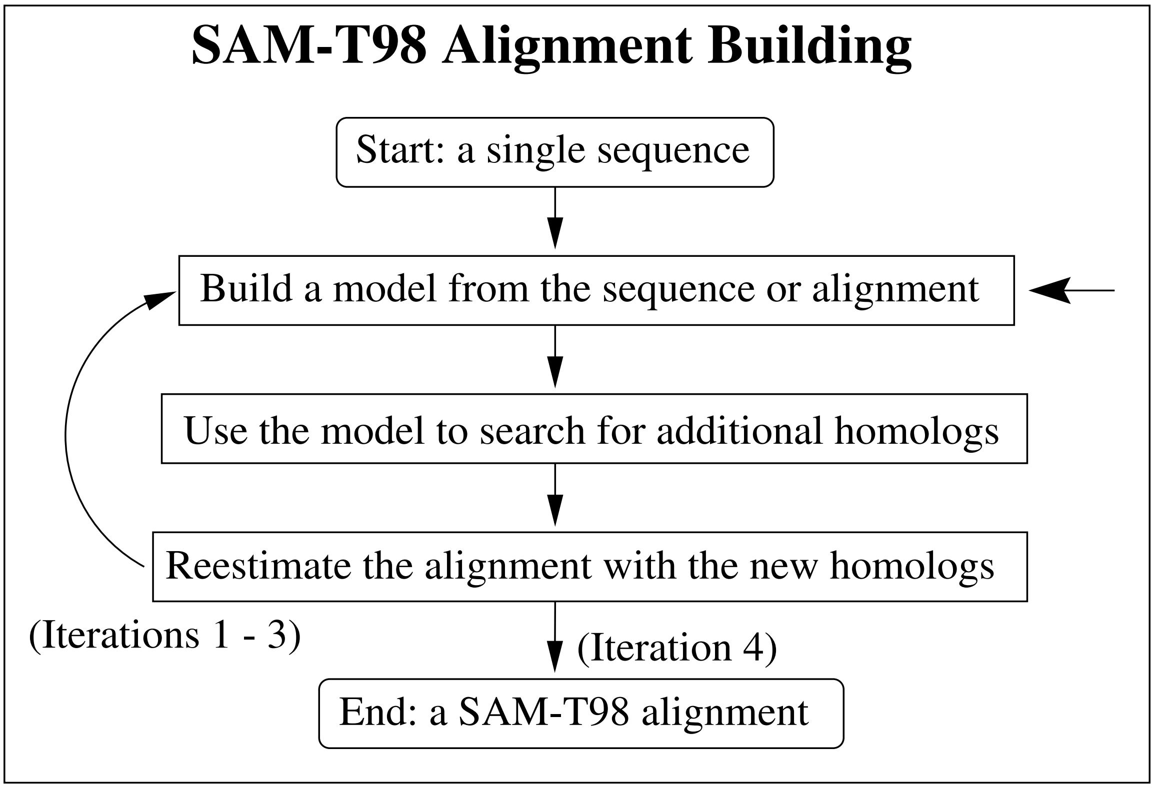
Effects of sequence weighting, continued

Using the same alignment and column regularizer as the previous logo, this logo reflects the alignment with Henikoff sequence weighting and half a bit saved on average over background entropy.

```
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....QFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....QFTQESCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....EFTDVDCSVSKECWSVCKDLFGVDRGKCMGKKCRCYQ
.....QFTDVDCSVSKECWSVCKDLFGVDRGKCMGKKCRCYQ
siegrQFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
.....-----SCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
.....XFTNVSCTTSKEXWSVCQRLHNTSRGKCMNKKXRCYS
.....XFTQESCTASNQCWSICKRLHNTNRGKCMNKKCRCYS
.....XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
.....XFTDVKCTGSKQCWPVCKQMFGKPNGKCMNGKCRCYS
```



Model building, Continued



1. Sequence Weighting

2. Column Regularizers

3. Transition Regularizers

Motivation: what would you consider the likelihood of anything other than Cysteine aligning in the indicated column in each of the two examples below?

Example 1

```
2crd XFTNVSCTTSKECWSVCQRLHNTSRG.KCMNKKCRCYS
```

Example 2

```
2crd XFTNVSCTTSKECWSVCQRLHNTSRG.KCMNKKCRCYS
1sxm TIINVKCTSPKQCSKPCKELYGSSAGaKCMNGKCKCYN
1cmr ------CTTSKECWSVCQRLHNTSKG.WCDHRGCICES
1bah XFTNVSCTTSKEXWSVCQRLHNTSRG.KCMNKKXRCYS
1txm ----VSCTGSKDCYAPCRKQTGCPNA.KCINKSCKCYG
1lir XFTQESCTASNQCWSICKRLHNTNRG.KCMNKKCRCYS
2bmt XFTNVSCSASSQCWPVCKKLFGTYRG.KCMNSKCRCYS
1bkt VGINVKCKHSGQCLKPCKDA-GMRFG.KCINGKCDCTP
1big XFTDVKCTGSKQCWPVCKQMFGKPNG.KCMNGKCRCYS
```

Desirable Properties in a Column Regularizer

A good column regularizer should do the following:

- Compute a set of column posteriors that is biologically realistic.
- Expect the amino acids not seen with nonzero probability.
- Respond to any pattern of conservation of residues or physiochemical property visible in the column.
- Take into account the size and weight of the alignment.
- Be inexpensive to compute and easy to understand.

Comparison of Column Regularizers

- The following slides show the logos produced using the same 3-sequence alignment, same weighting scheme, and different regularizers:
  - Maximum-likelihood estimate
  - Add-one pseudocount
  - Gribskov average score [GME87] with BLOSUM62 matrix [HH92]
  - Recode3.20comp Dirichlet mixture [SKB+96]

  Other regularizers are evaluated in the handouts [Kar95].
- All logos depict the savings at each column over the background entropy, where the background amino acid distribution P_0 is obtained by using the regularizer with no amino acid observations.

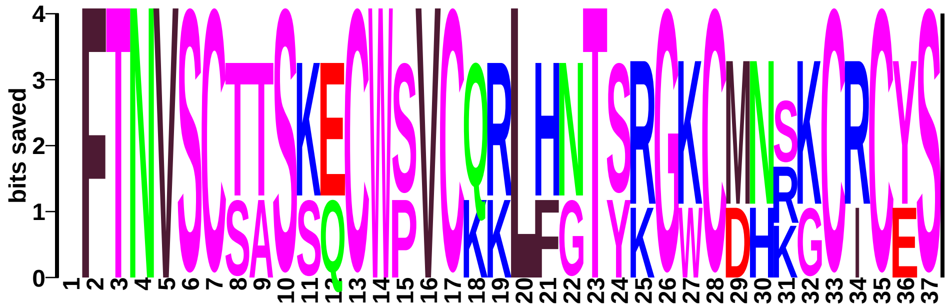
Maximum-Likelihood Estimate

Description: Given a set of observed column counts $\vec{O}$, the posterior probability for amino acid a, $\hat{P}(a|\vec{O})$, is computed as follows:

$$\hat{P}(a|\vec{O}) = \frac{O(a)}{\Sigma_{\text{amino acids } j}\, O(j)}$$

- Easy to compute.
- The distribution that gives the highest probability to the observed data.
- Gives infinite cost (zero probability) to amino acid that hasn't already been seen.
- Performs well when $\vec{O}$ is very, very large, but terribly in real cases.

```
2crd  XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr  ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt  XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



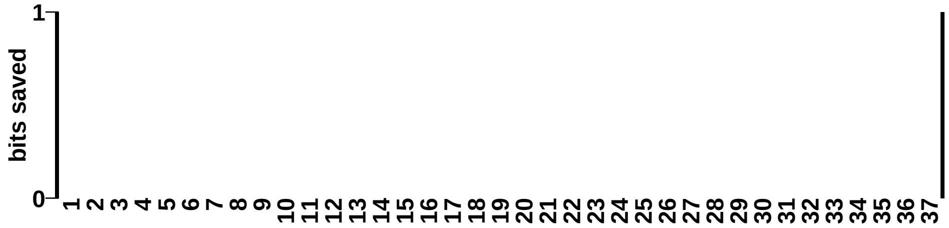
Add-One Pseudocount

Description: Given a set of observed column counts $\vec{O}$, the posterior probability for amino acid a, $\hat{P}(a|\vec{O})$, is computed as follows:

$$\hat{P}(a|\vec{O}) = \frac{O(a)+1}{\Sigma_{\text{amino acids } j} O(j)+1}$$

- Easy to compute.
- No knowledge—all distributions equally likely a priori.
- Performs well when $\vec{O}$ is very large, but poorly in most other cases.

```
2crd  XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr  ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt  XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



The logo above is not erroneous. You see no pronounced signal at any column because there is none. The low weights combined with the large pseudocounts make the posterior probabilities close to uniform.

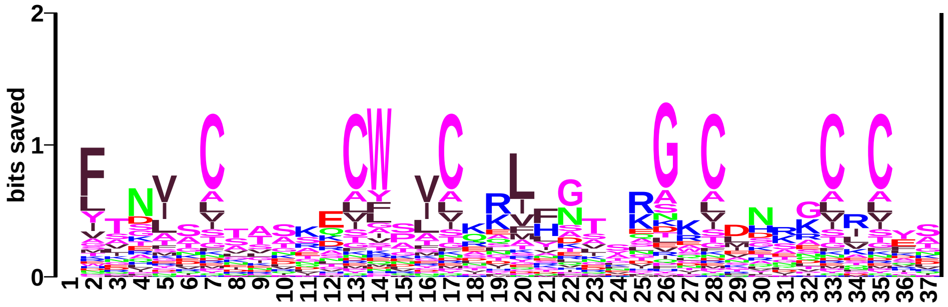
Gribskov Average Score [GME87] with BLOSUM62 Matrix [HH92]

Description: The posterior probability of some residue a given the set of counts $\vec{O}$ observed is defined according to a set of background probabilities P_0, a substitution matrix M:

$$\hat{P}(a|\vec{O}) \leftarrow P_0(a)e^{\frac{\sum_b M_{a,b} O(b)}{|\vec{O}|}}$$

- Easy to compute.
- Achieves excellent performance with one sequence.
- Does not change as the total sequence weight changes.
- Does not make optimal use of multiple sequences.

```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



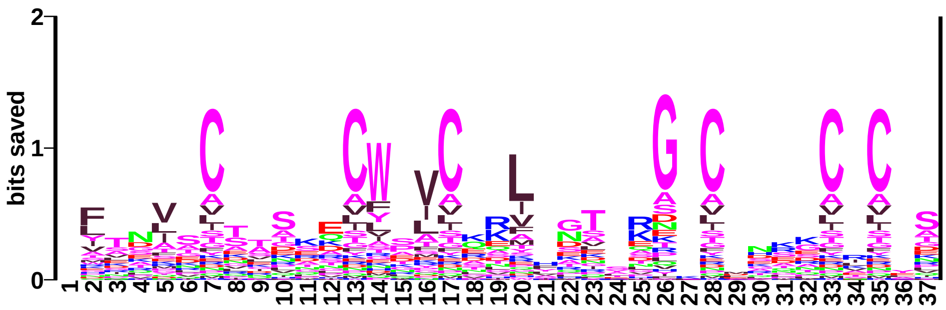
Dirichlet Mixtures [SKB+96]

Description: Several different pseudocounts (*components*) are fit to patterns of conservation and substitution within amino acid columns. To compute the posterior distribution of an alignment column, probability of each component is computed and used to mix together the posterior probabilities given the components.

$$\hat{P}(a|\vec{O}) = \sum_c P(c|\vec{O}) \frac{O(a) + \alpha_{c,a}}{\sum_{\text{amino acids } b} O(b) + \alpha_{c,b}}$$

- Can be hard to understand.
- Performs well with a broad range of alignment sizes.

```
2crd   XFTNVSCTTSKECWSVCQRLHNTSRGKCMNKKCRCYS
1cmr   ------CTTSKECWSVCQRLHNTSKGWCDHRGCICES
2bmt   XFTNVSCSASSQCWPVCKKLFGTYRGKCMNSKCRCYS
```



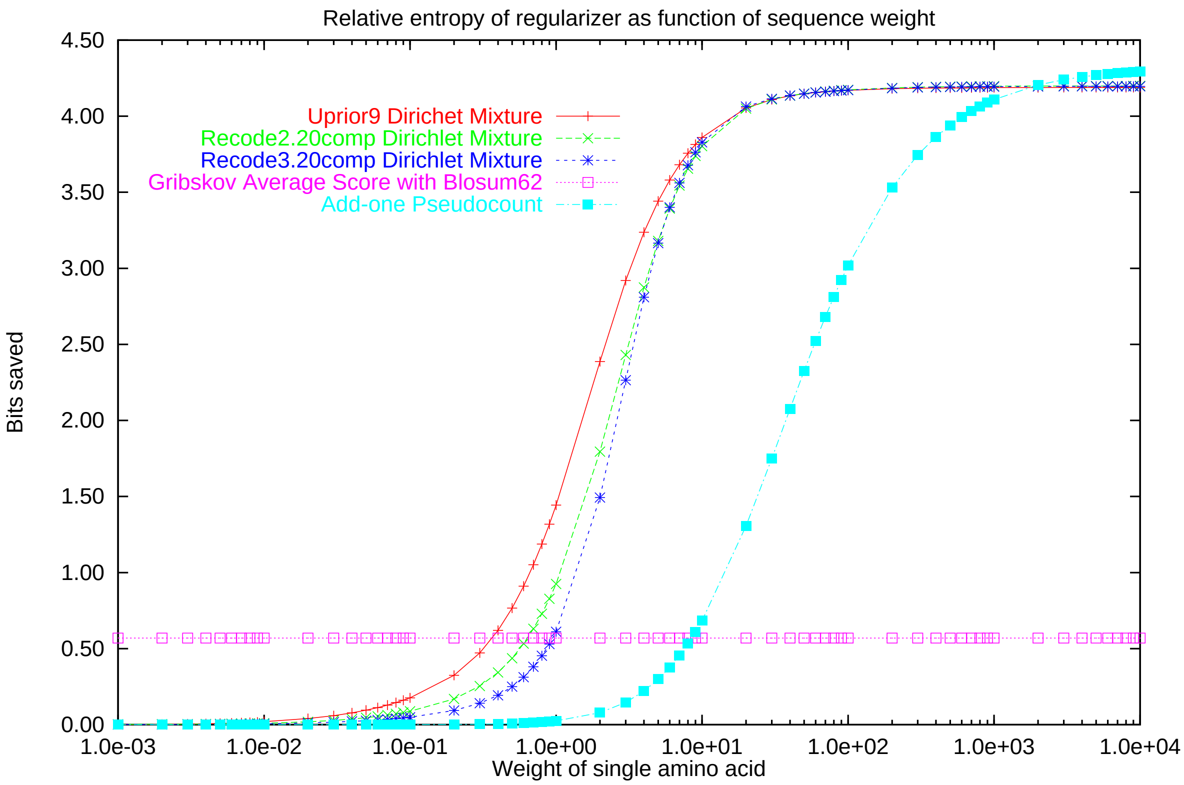
Regularizer Performance at Various Sequence Weights

To assess the performance of the regularizers at various total weights, below we compare their expected savings given one observation of amino acid b at various weights.

Expected column entropy $E[H_c]$ is computed as shown, where $\hat{P}(a,b)$ is the posterior probability of amino acids a and b in column c, and $P(a)$ and $P(b)$ are their prior probabilities:

$$E[H_c] = \sum_{\text{amino acids a,b}} \hat{P}(a,b) \log \frac{\hat{P}(a,b)}{P(a)P(b)}$$

$$= \sum_{\text{amino acids a,b}} \hat{P}(a|b) P(b) \log \frac{\hat{P}(a|b)}{P(a)}$$

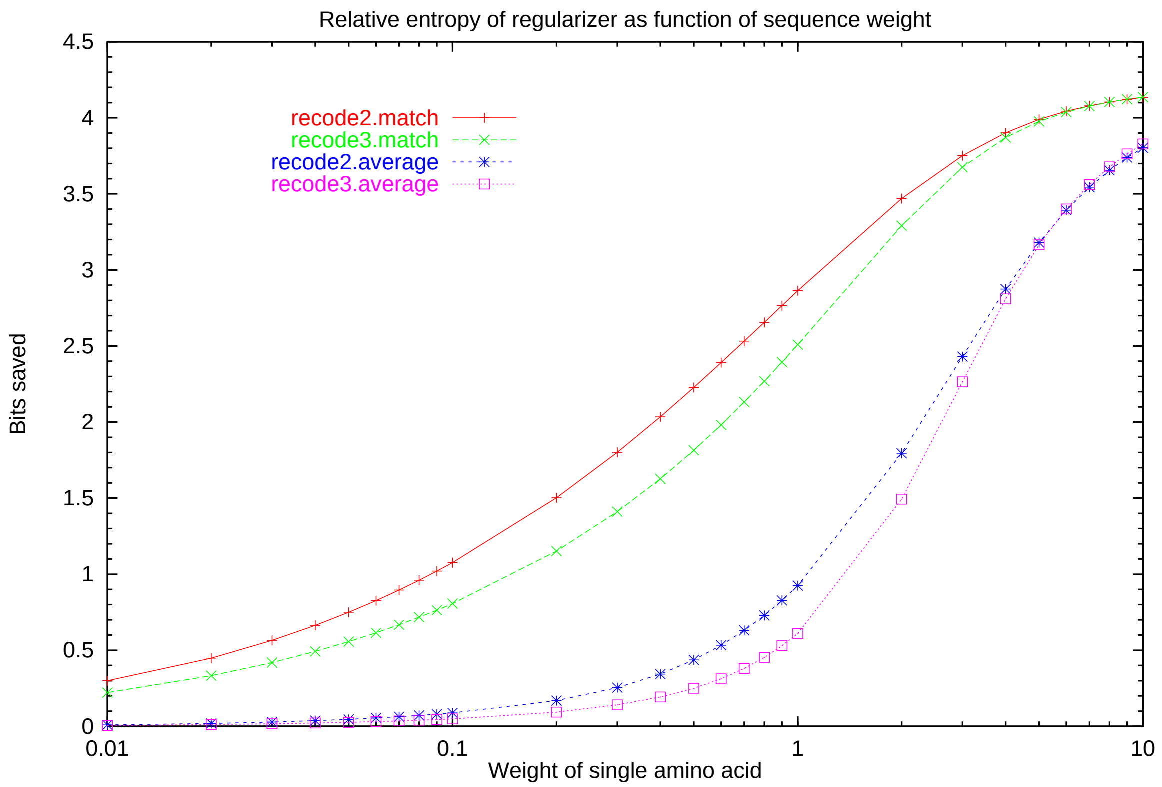


Peak savings versus weight

The previous slide showed the *average* bit savings as a function of weight, but an exact match to the single character saves more than average.

Matching column entropy is computed as

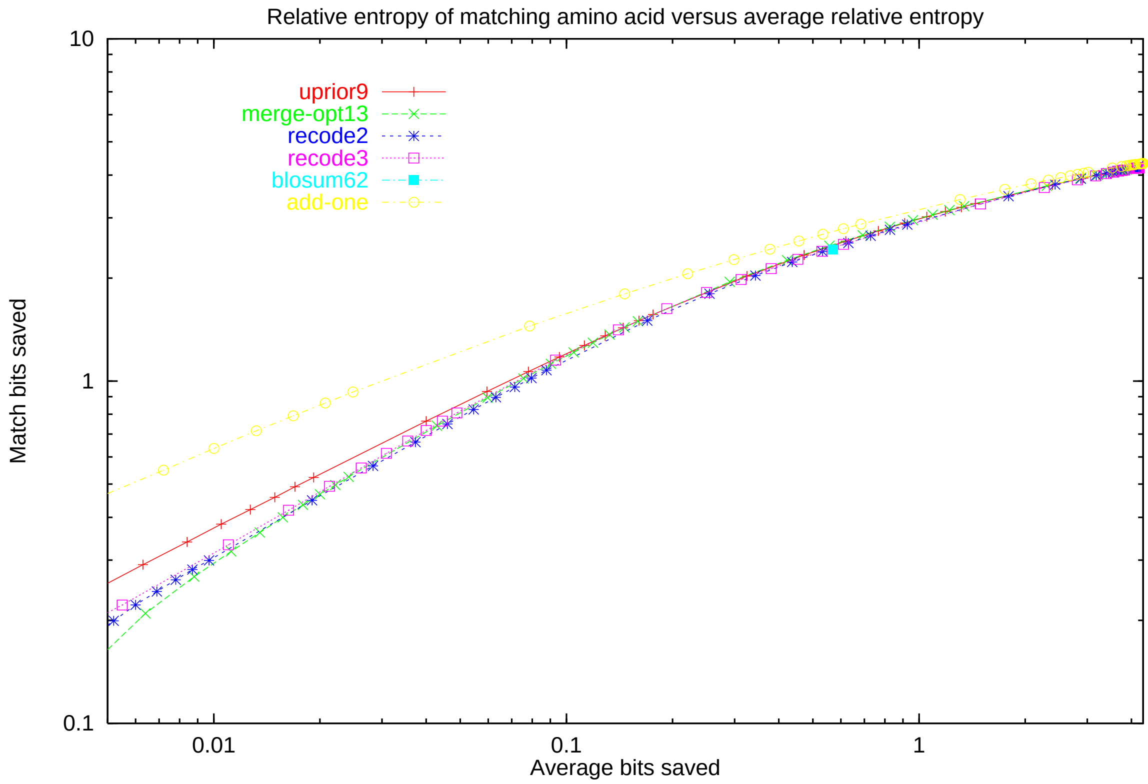
$$H_{c,\mathrm{match}} = \sum_{\text{amino acids a}} P(a) \log \frac{\hat{P}(a|a)}{P(a)}$$



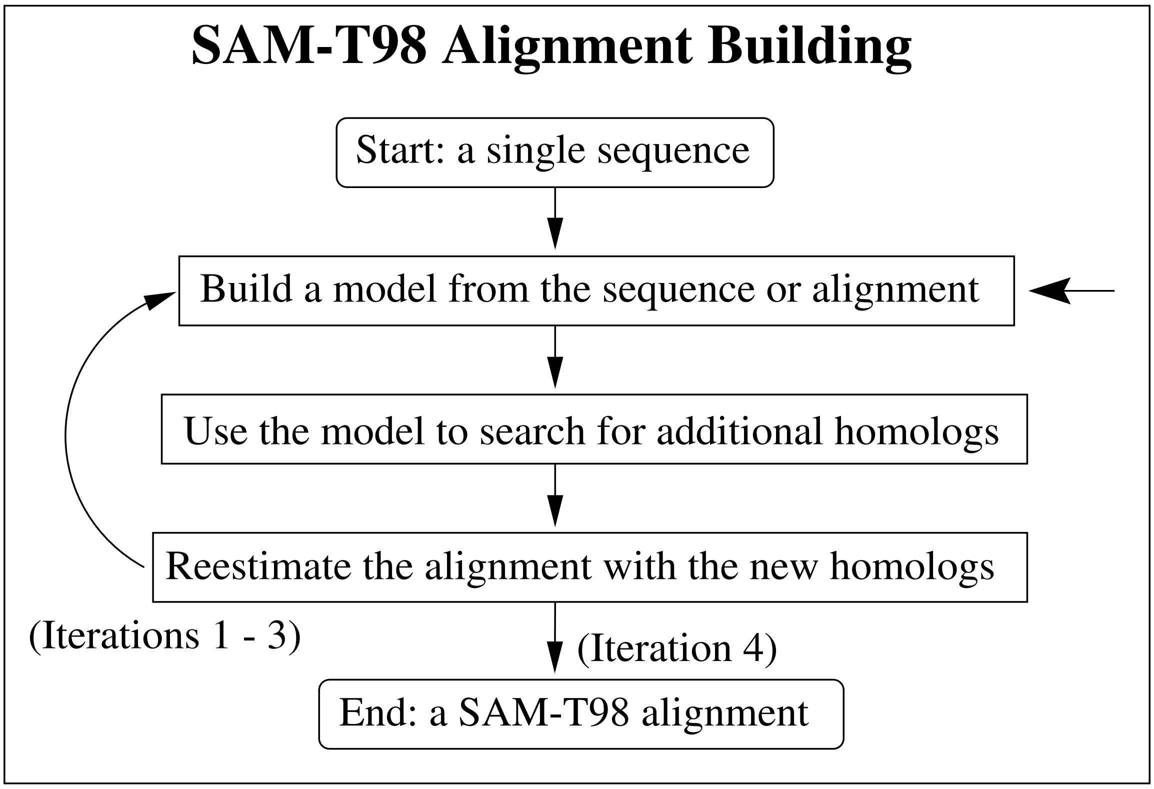
Peak savings versus Average savings

We can plot the savings for the matching character versus the average savings—all the regularizers trained on protein data behave similarly.

At 0.5 bits/character on average, we get about 2.2 bits of savings for a matching character.



Model building, Continued



1. Sequence Weighting
2. Column Regularizers

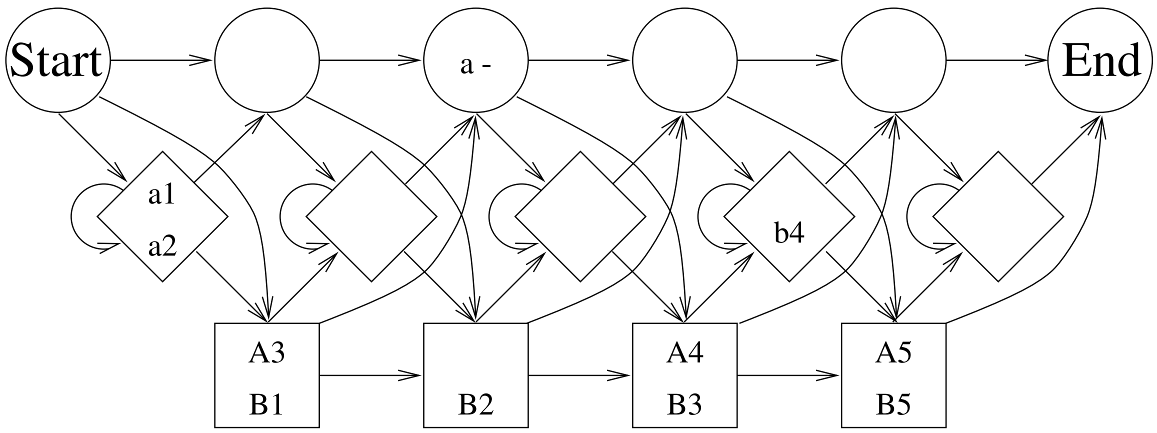
3. Transition Regularizers

Model Transitions

There are nine types of transitions that can be made:

- From the match state: match-match, match-delete, and match-insert.
- From the delete state: delete-match, delete-delete, and delete-insert.
- From the insert state: insert-match, insert-delete, and insert-insert.

The probability of entering a delete or insert state from a match state is analogous to the gap opening cost in other methods, and the delete-delete and insert-insert probabilities are analogous to the gap extension cost.




```
a1 a2 A3    -    A4  .  A5
 .  . B1   B2    B3 b4  B5
```

Transition posteriors in SAM-T98

- SAM-T98 applies a system of *transition regularizers* to estimate transition costs.
- These transition regularizers contain small pseudocounts representing prior odds for each transition.
- Because we just use pseudocounts, transition parameter estimation is crude compare to the use of Dirichlet mixture priors to estimate match state probabilities.
- For any adjacent states in the model (or columns in the training alignment), SAM-T98 computes the transition costs based on two factors:
  1. observed transitions in the (weighted) training alignment
  2. the pseudocounts in the transition regularizer

Transition posteriors in SAM-T98, contd.

- Each alignment column is modeled with distinct transition costs.
- All transitions have finite cost (i.e. $0.0 < \text{probability} \leq 1.0$).
- When the sequence weights very small, the transition costs reflect the regularizer.
- When the sequence weights are large, the transition costs reflect the observations. Gaps are most likely where the alignment already contains gaps.
- The transition regularizers we use are stiff, so observations do not change transition probabilities much from the background.

Selecting a transition regularizer

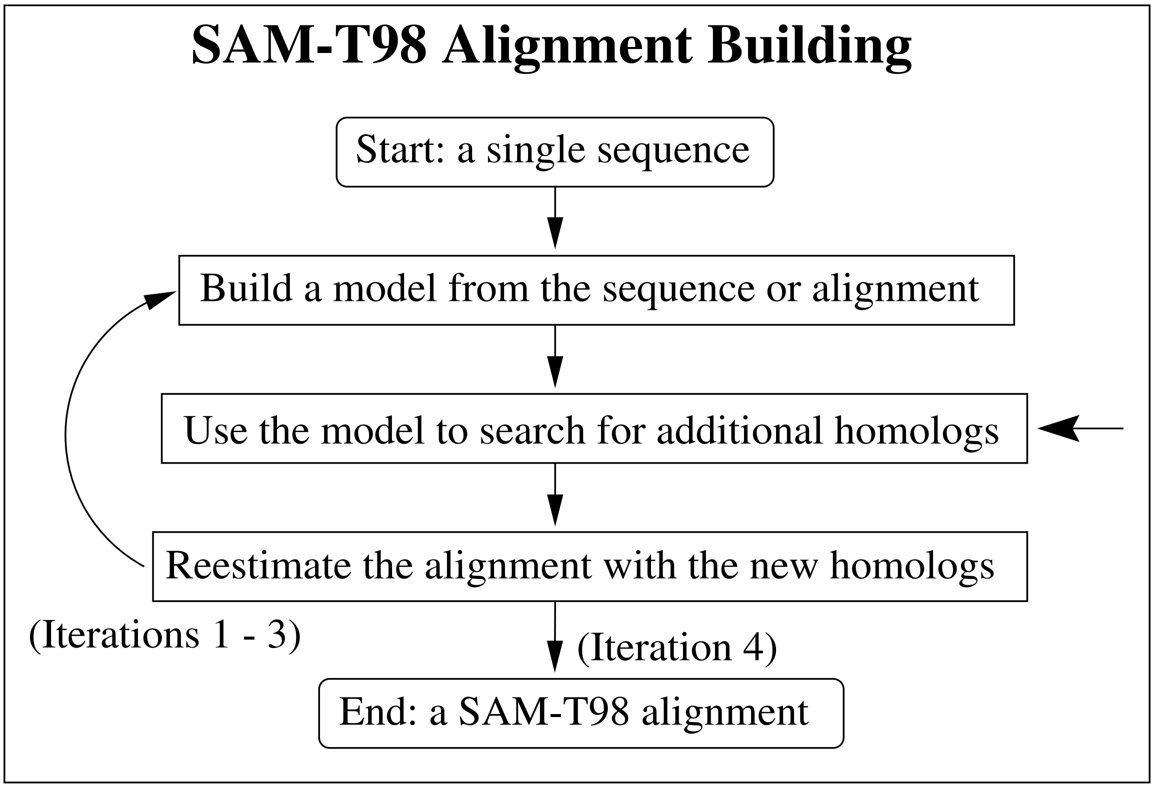
- We have never found a transition regularizer that works well in all contexts.
- SAM-T98 uses three different transition regularizers:
  - In the first iteration, only very close homologs are admitted into the training alignment, and the transition regularizer used allows numerous gaps.
  - The transition regularizer used in the next two iterations encourages long stretches of match states.
  - The final iteration uses a transition regularizer trained on FSSP structural alignments [HS97].
    - Based on structural alignments of distantly related proteins.
    - Not as stiff and is hopefully a better approximation to the “true” transition parameter values.

Secondary-Structure-dependent gap opening and gap extension penalties

Advantages Gap costs are appropriate for the secondary structure.

Disadvantages Gaps are not uniformly likely across a secondary structure element. For example, deletion of two residues at the start of a helix is less surprising than deletion of two residues in the middle.

Searching with the model



Searching for additional homologs means applying the model to a database containing potential homologs, scoring each sequence in turn, and selecting those that score well. Issues include:

- Execution time
  - Reducing the number of sequences to score
  - Reducing the time necessary to score each sequence
- Significance: when does a score imply homology?
  - Selecting a reasonable null model
  - Selecting a reasonable score threshold given the null model

Reducing the Number of Sequences to Score

Problem:
- As of April 1999, the non-redundant database contains close to 400,000 sequences.
- Performing a full search with an HMM takes on the order of fifteen hours.
- SAM-T98 requires four rounds of database searching.

One Solution: Use BLAST [AGM+90] to select potential remote homologs from the non-redundant database. Search against the BLAST hits rather than the full database.

Reducing the Number of Sequences to Score, contd.

SAM-T98 uses BLAST to identify two pools of potential homologs:

1. A small pool of very strong matches ($\ln(\text{P-value}) \leq -10.5$), used for the first iteration only.
2. A larger pool of potential homologs ($E \leq 500$), used for subsequent iterations.

The "BLAST trick" yields a potential homolog set of 10,000 sequences in the worst case, far fewer in a normal case.

Reducing the Time Necessary to Score Each Sequence

- Kestrel [HHKS96], a specialized parallel processor under development at UCSC, reduces the search times on a database containing 10^7 amino acids as shown:

	Kestrel (20 MHz)	On a 433 MHz DEC Alpha				
Query Size	≤ 512	32	64	128	256	512
Execution Time (seconds)	41	80	136	232	442	862

- Runtime comparison scoring a database of 100 sequences (all 256 amino acid in length) against a database of 10^7 amino acids:
  - Kestrel $\longrightarrow$ 1hr 8min
  - 433 MHz DEC Alpha $\longrightarrow$ 12hr 17min
- At this time, SAM-T98 does not use Kestrel. However, if database sizes continue to grow exponentially, specialized processors such as Kestrel might become the best solution for limiting database search times.
- We expect Kestrel to obviate the need for the "BLAST trick".

- The *model* M is a computable function that assigns a probability $\mathrm{Prob}\,(A \mid M)$ to each string A.
- When given a string A, we want to know how likely the model is. That is, we want to compute something like $\mathrm{Prob}\,(M \mid A)$.
- Bayes Rule:

$$\mathrm{Prob}\,(M \mid A) = \mathrm{Prob}\,(A \mid M)\,\frac{\mathrm{Prob}(M)}{\mathrm{Prob}(A)}\ .$$

- Problem: $\mathrm{Prob}(A)$ and $\mathrm{Prob}(M)$ are inherently unknowable.

Null models

- Standard solution: ask how much more likely M is than some *null hypothesis* (represented by a *null model*).

$$\frac{\text{Prob}\,(M \mid A)}{\text{Prob}\,(N \mid A)} = \frac{\text{Prob}\,(A \mid M)}{\text{Prob}\,(A \mid N)}\,\frac{\text{Prob}(M)}{\text{Prob}(N)}\ .$$

- $\frac{\text{Prob}(M)}{\text{Prob}(N)}$ is the *prior odds ratio*, and represents our belief in the likelihood of the model before seeing any data.
- $\frac{\text{Prob}(M|A)}{\text{Prob}(N|A)}$ is the *posterior odds ratio*, and represents our belief in the likelihood of the model after seeing the data.
- We can generalize to a forced choice among many models $(M_1, \ldots, M_n)$

$$\frac{\text{Prob}\,(M_i \mid A)}{\Sigma_j\,\text{Prob}\,(M_j \mid A)} = \frac{\text{Prob}\,(A \mid M_i)\,\text{Prob}(M_i)}{\Sigma_j\,\text{Prob}\,(A \mid M_j)\,\text{Prob}(M_j)}\ .$$

The $\text{Prob}(M_j)$ values can be scaled arbitrarily without affecting the ratio.

Standard Null Model

- Null model is an i.i.d (independent, identically distributed) model, that is, each letter is treated as being independently drawn from the background distribution.

- $$\text{Prob}\,(A \mid N, \text{len}\,(A)) = \prod_{i=1}^{\text{len}(A)} \text{Prob}(A_i)\ .$$

- $$\text{Prob}\,(A \mid N) = \text{Prob}(\text{string of length len}\,(A)) \prod_{i=1}^{\text{len}(A)} \text{Prob}(A_i)\ .$$

- The length modeling is often omitted, but one must be careful then to normalize the probabilities correctly.

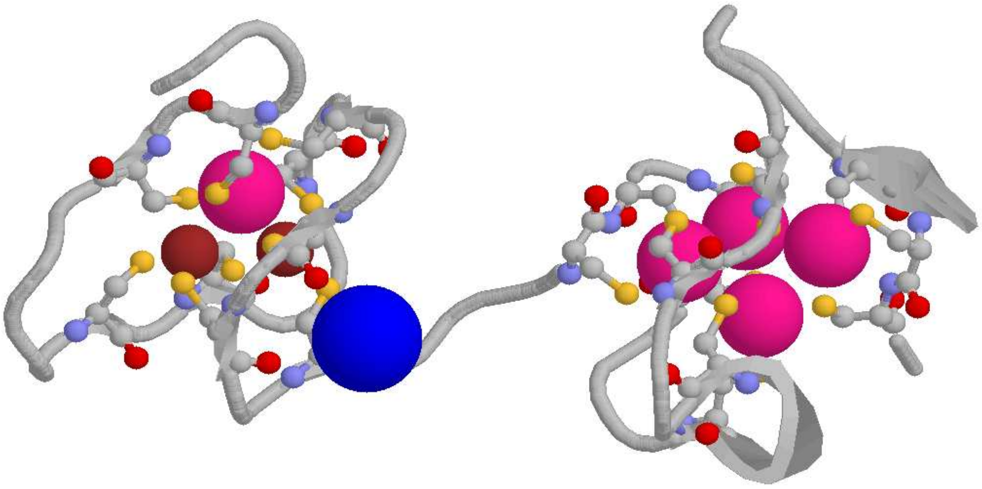
Reversed model for null

- When using the standard null model, certain sequences and HMMs have anomalous behavior. Many of the problems are due to unusual composition—a large number of some usually rare amino acid.
- For example, metallothionein, with 24 cysteines in only 61 total amino acids, scores well on any model with multiple highly conserved cysteines.
- We avoid this (and several other problems) by using a reversed model M^r as the null model.
- The probability of a sequence in M^r is exactly the same as the probability of the reversal of the sequence given M.
- If we assume that M and M^r are equally likely, then
$$\frac{\text{Prob}\,(M \mid S)}{\text{Prob}\,(M^r \mid S)} = \frac{\text{Prob}\,(S \mid M)}{\text{Prob}\,(S \mid M^r)} .$$
- This method corrects for composition biases, length biases, and several subtler biases.

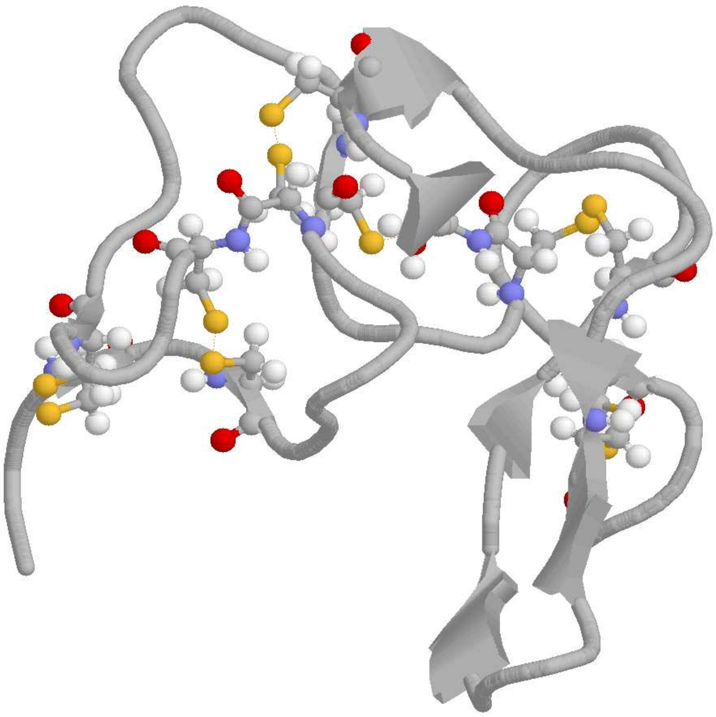
A cysteine-rich protein, such as metallothionein, can match any HMM that has several highly-conserved cysteines, even if they have quite different structures:

		cost in nats	
HMM	sequence	model − standard null	model − reversed-model
1kst	4mt2	-21.15	0.01
1kst	1tabI	-15.04	-0.93
4mt2	1kst	-15.14	-0.10
4mt2	1tabI	-21.44	-1.44
1tabI	1kst	-17.79	-7.72
1tabI	4mt2	-19.63	-1.79

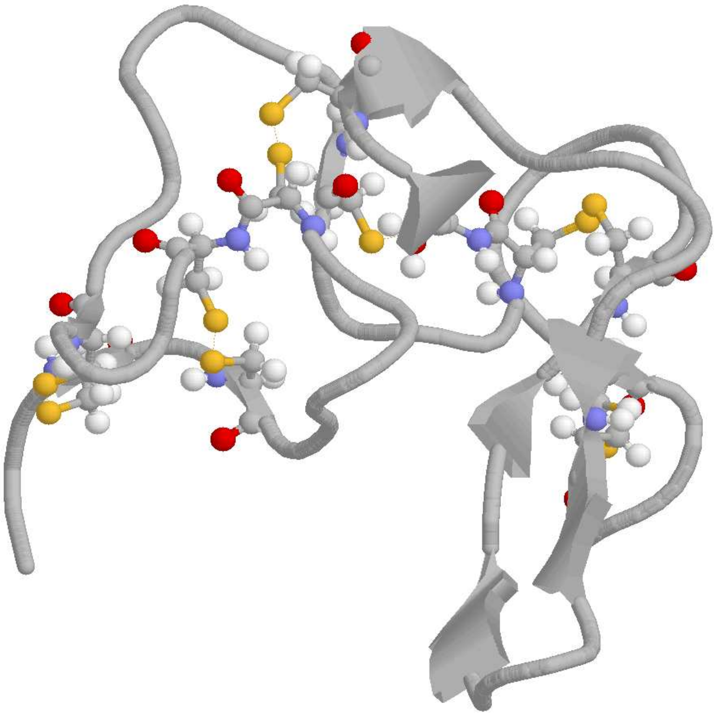
Metallothionein Isoform II (4mt2)



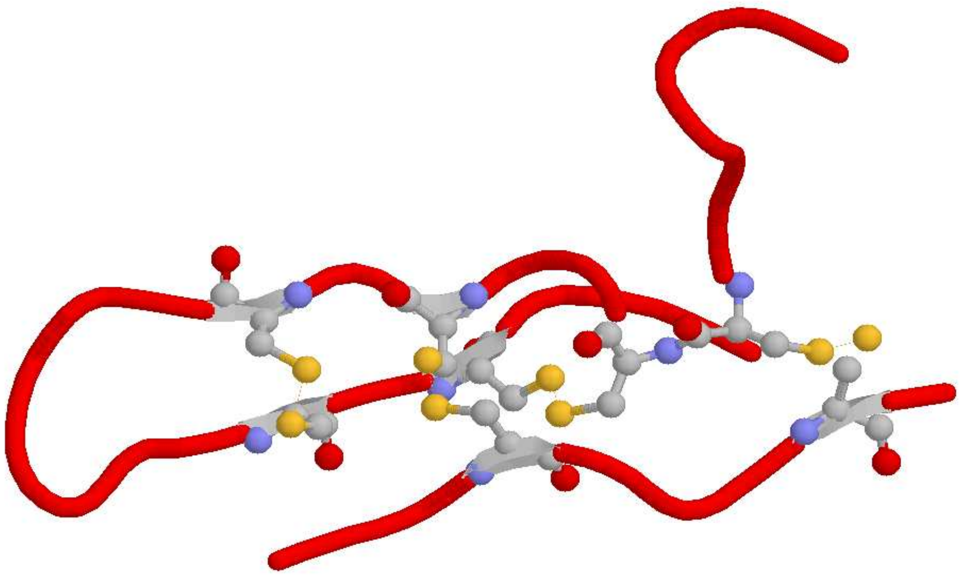
Kistrin (1kst)



Kistrin (1kst)



Trypsin-binding domain of Bowman-Birk Inhibitor (1tabI)



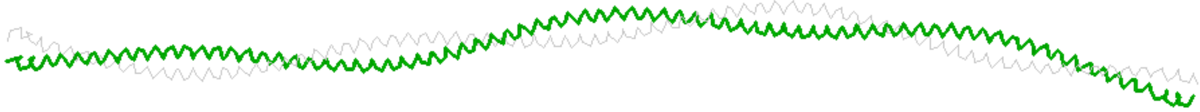
Long helices as source of error

Long helices can provide strong similarity signals from the periodic hydrophobicity, even when the overall folds are quite different:

		cost in nats, normalized using	
HMM	sequence	Null model	reversed-model
1av1A	2tmaA	-22.06	2.13
1av1A	1aep	-21.25	1.03
1av1A	1cii	-13.67	-1.75
1av1A	1vsgA	-7.89	-0.51
2tmaA	1cii	-20.62	0.46
2tmaA	1av1A	-17.96	1.01
2tmaA	1aep	-12.01	0.78
2tmaA	1vsgA	-8.25	0.08
1vsgA	2tmaA	-14.82	-1.20
1vsgA	1av1A	-13.04	-2.68
1vsgA	1aep	-13.02	-3.52
1vsgA	1cii	-11.12	0.28
1aep	1av1A	-11.30	1.79
1aep	2tmaA	-10.73	1.06
1aep	1cii	-8.35	1.38
1aep	1vsgA	-6.87	0.53
1cii	2tmaA	-23.24	-1.48
1cii	1av1A	-19.49	-5.62
1cii	1aep	-12.85	-1.77
1cii	1vsgA	-10.20	-1.57

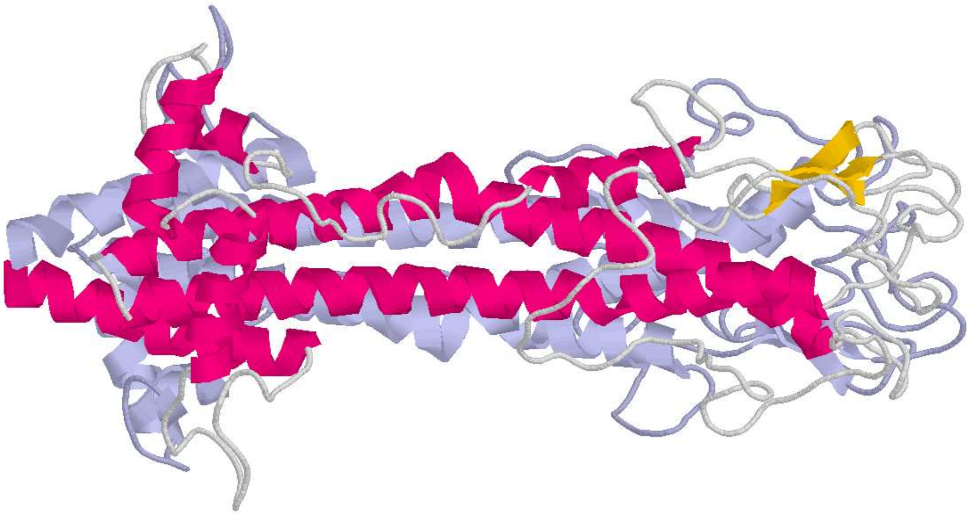
Helix examples

Tropomyosin (2tmaA)



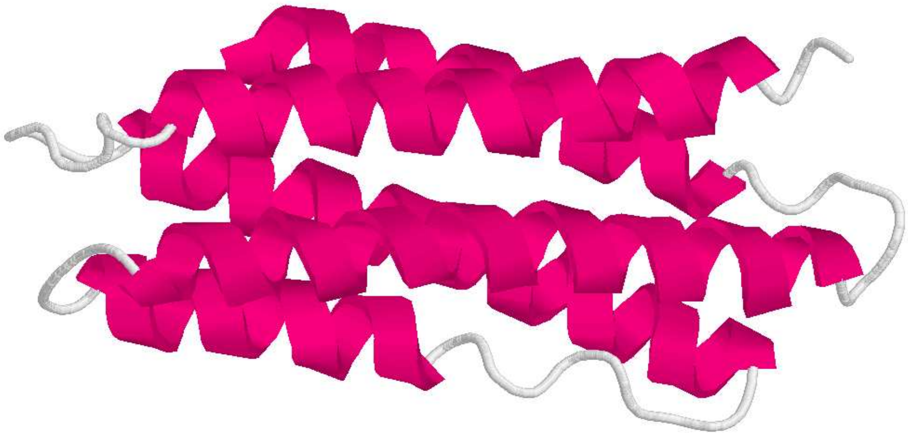
Colicin Ia (1cii)

Flavodoxin mutant (1vsgA)

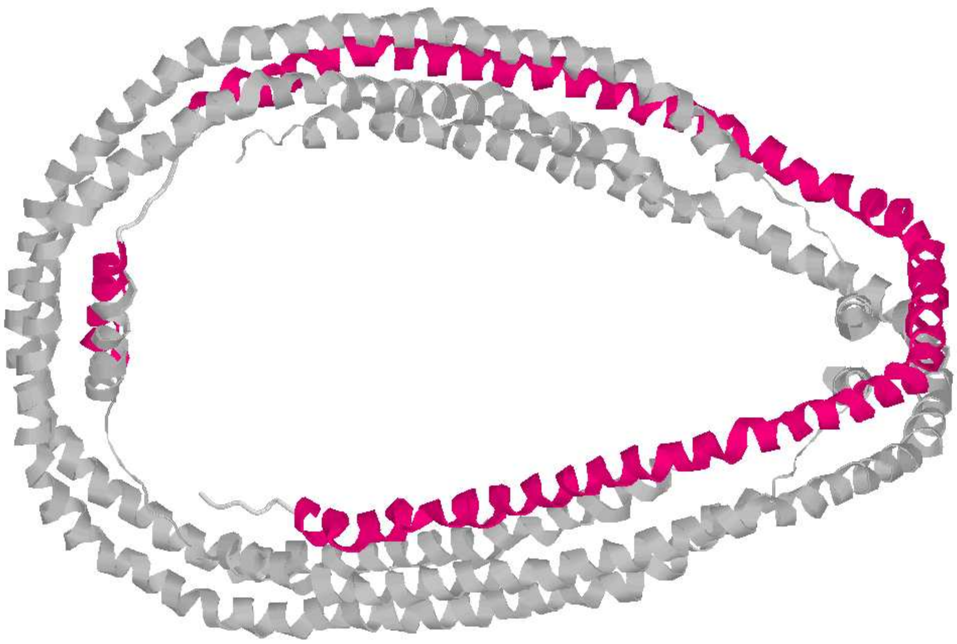


Helix examples

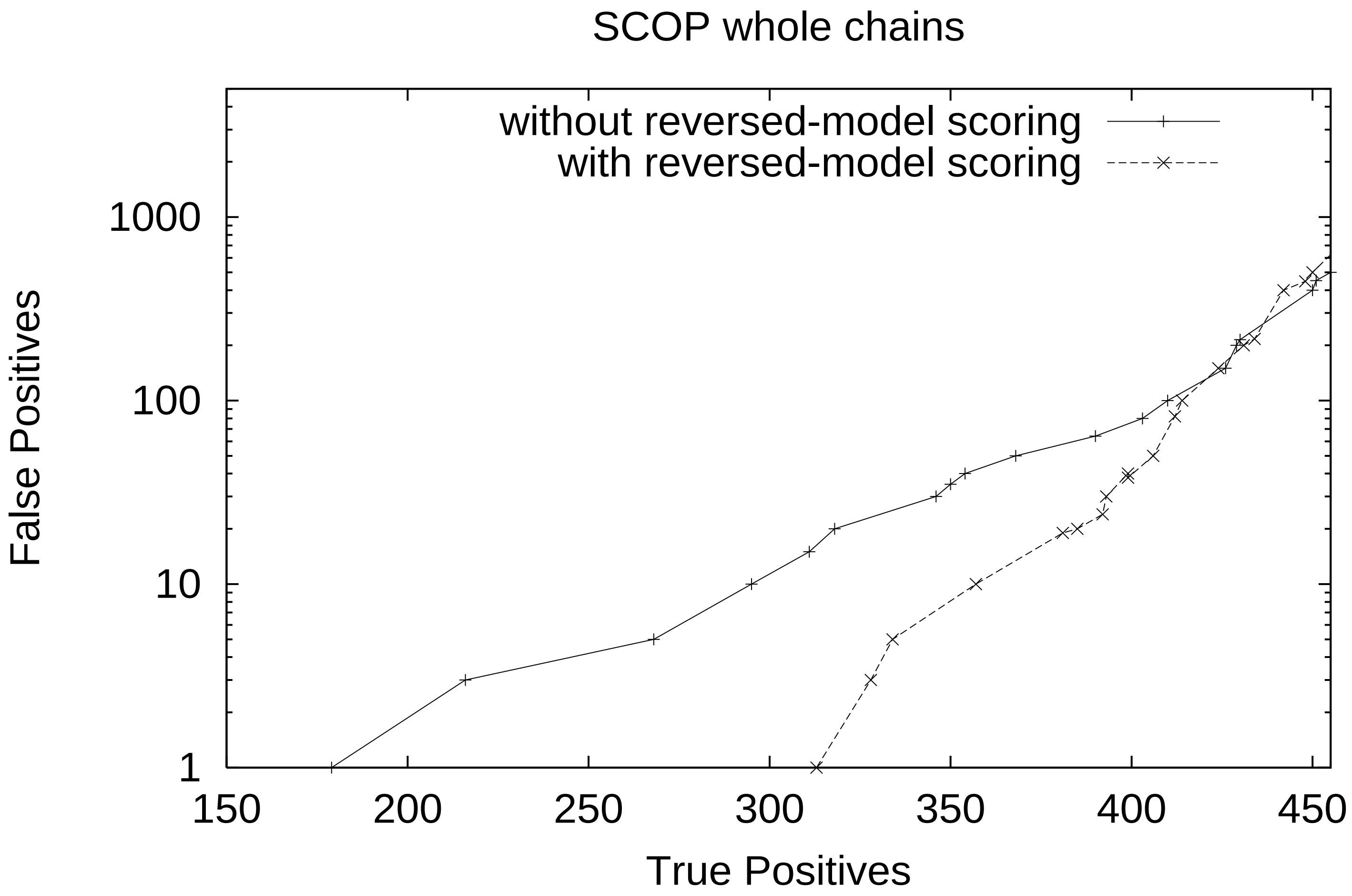
Apolipophorin III (1aep)



Apolipoprotein A-I (1av1A)

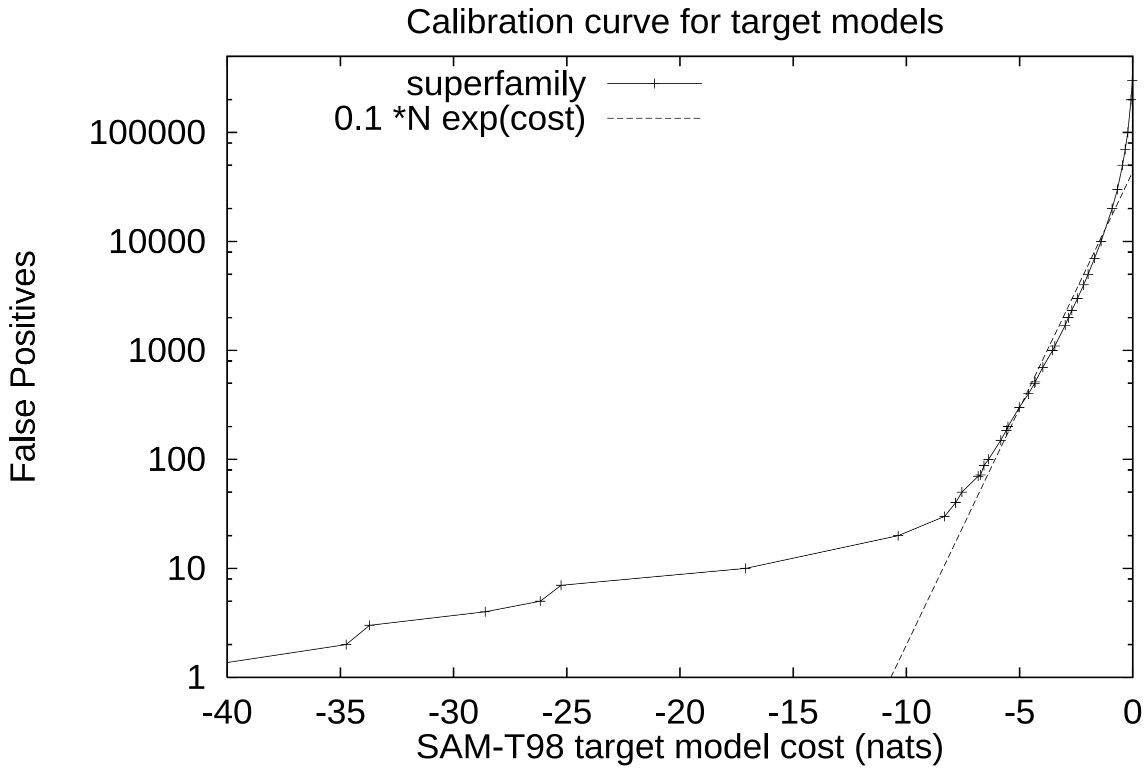


Discrimination Performance as a Function of Null Model

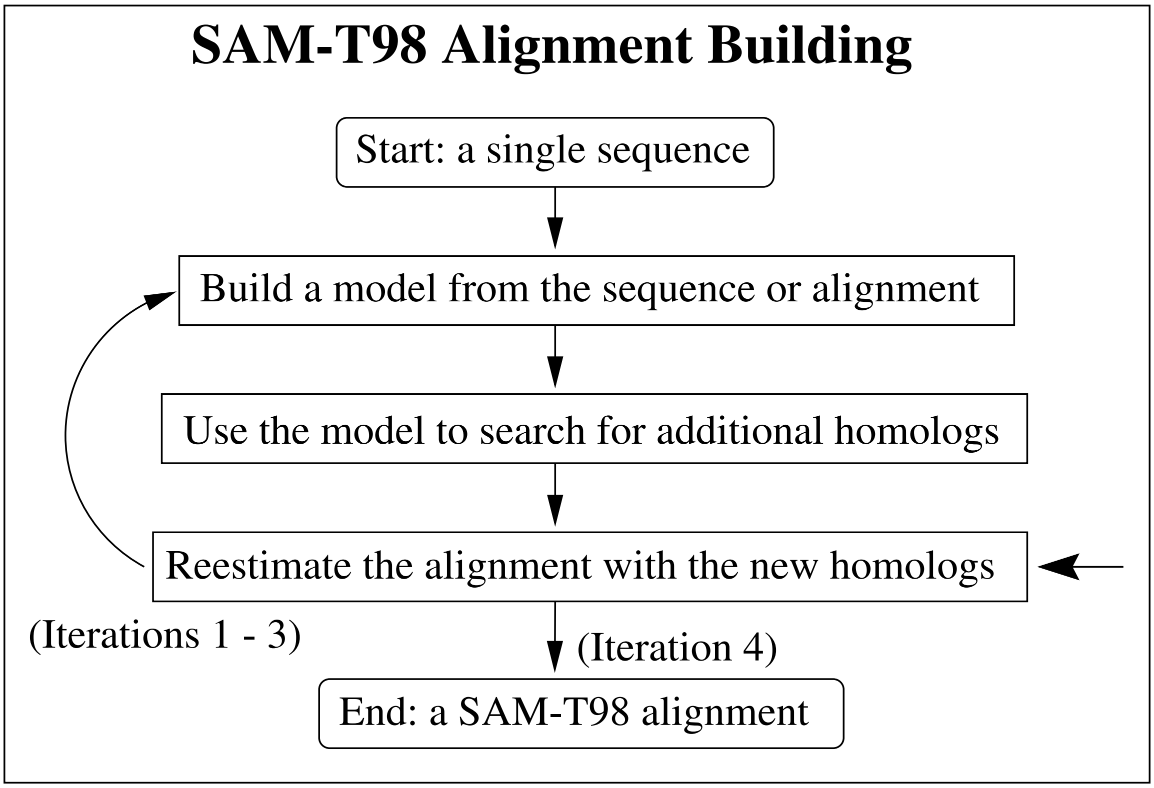


Null Model Scores: Significance and Threshold Selection

After selecting a null model, the next step is to test the performance and select threshold values using a test appropriate to the final application.



On the first iteration, SAM-T98 uses a strict threshold (-40 nats) to admit only strong matches. In later iterations, the threshold is progressively loosened (-30, -24, and -16 nats).



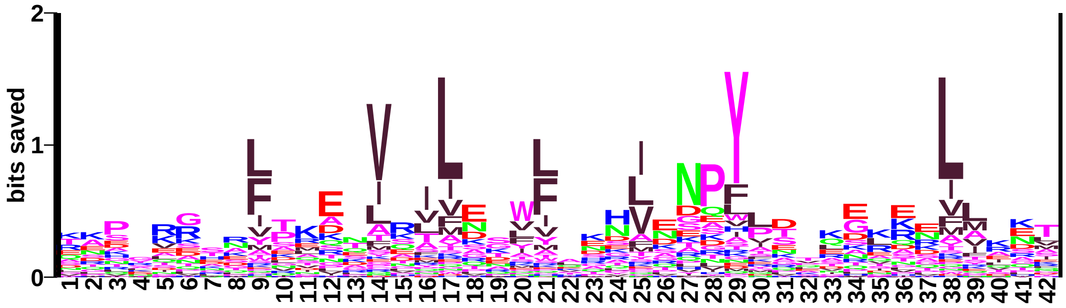
Alignment reestimation adds the new homologs to the alignment and attempts to find new signal made available by their addition. In SAM-T98, reestimation is performed by `buildmodel`, SAM's implementation of the *EM Algorithm*.

Issues in alignment reestimation:

- Keeping the seed sequence from drifting into a gap
- Preserving signal from a trusted seed alignment

Effects of Reestimation

The sequence logos shown were generated by the alignment produced for 1yrnB in the third iteration of SAM-T98. The upper logo reflects the alignment before reestimation, and the lower logo after reestimation.



- Asparagine 27 increases substantially.
- Leucines 17 and 38 and Tyrosine 29 all increase slightly.
- Proline 3 decreases substantially. However, Proline 3 does not figure prominently in either the FSSP [HS97] or HSSP [SS96] alignment.

Effects of Reestimation, Continued

The excerpts below show sequences from the alignment before reestimation (left) and after reestimation (right). The increase in the signal of Asparagine 27 is due to the realignment of the second Histidine in Sequence 11.

```
    20         30                20         30
    |          |                 |          |
 1 WFA.KNI.ENPYLD             1 WFA.KNI.ENPYLD
 2 WFA.KNIaENPYLD             2 WFA.KNIaENPYLD
 3 WFA.KNI.ENPYLD             3 WFA.KNI.ENPYLD
 4 WFAkKNI.ENPYLD             4 WFAkKNI.ENPYLD
 5 WFA.KNI.ENPYLD             5 WFA.KNI.ENPYLD
 6 WFA.KNI.ENPYLD             6 WFA.KNI.ENPYLD
 7 WLL.NHL.NNPYPT             7 WLL.NHL.NNPYPT
 8 WFA.KNI.ENPYLD             8 WFA.KNI.ENPYLD
 9 WLH.EHV.NNPYPT             9 WLH.EHV.NNPYPT
10 WFA.KNI.ENPYLD            10 WFA.KNI.ENPYLD
11 VFQ.H--.-HQYLS            11 VFQ.HH-.--QYLS
```

A small change of alignment can yield a pronounced change in signal.

Pitfalls in Alignment Reestimation

Issue: The alignment of certain sequences should not be reestimated.

- If a seed alignment is used, alignment of the seed sequences should not change.
- If some insert state is used by a large number of homologs, its costs will drop and the template sequence might choose the insert over some match state(s).

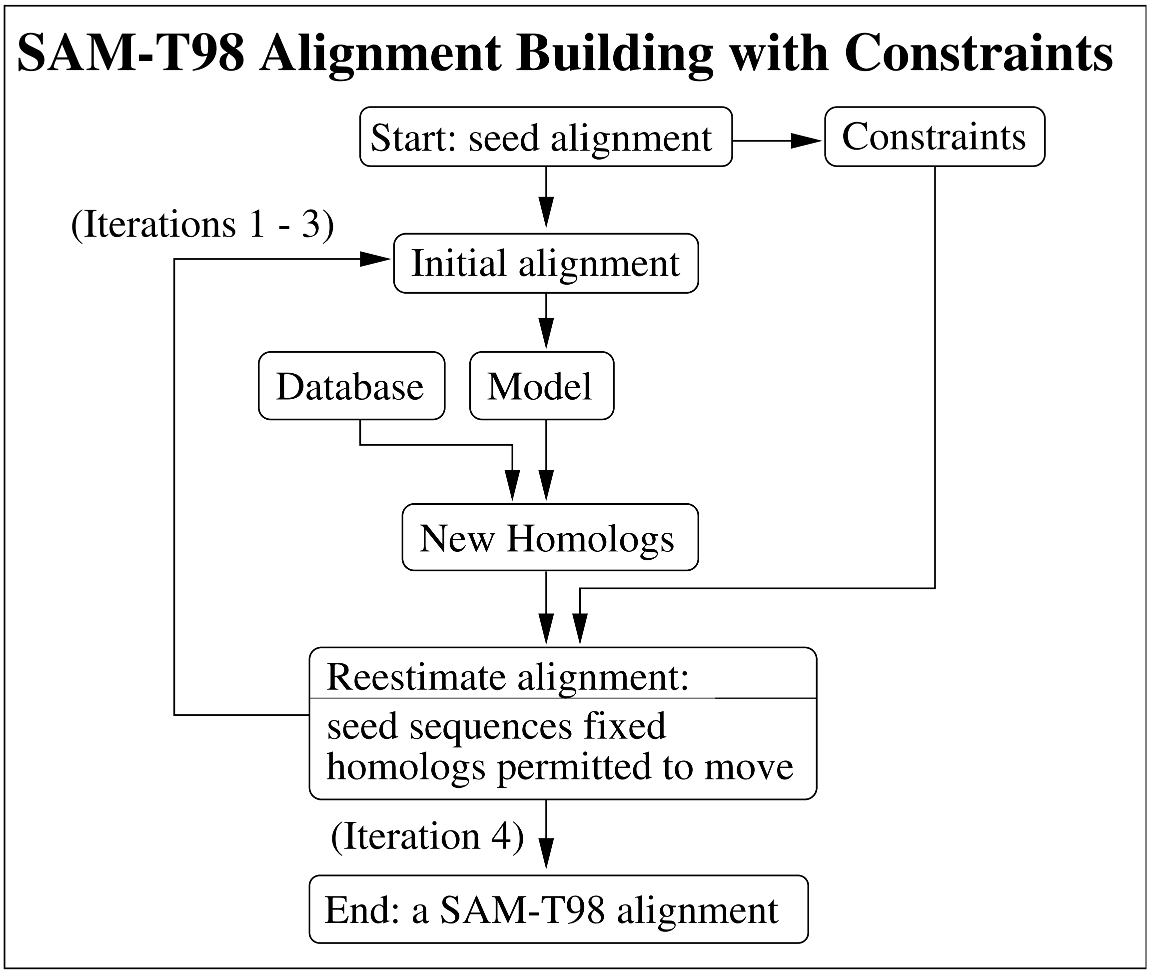
To illustrate, here is a portion of the alignment for 1yrnB from the fourth iteration of SAM-T98:

```
                          10
                          |
1yrnB   t....KPYRGhr---FTKENVRIL
1hom    m....-RKRG..RQTYTRYQTLEL
1b72B   a....--RRK..RRNFNKQATEIL
1mnmC   inkstKPYRGhr---FTKENVRIL
1enh    .....---RP..RTAFSSEQLARL
1fjlA   ialkrKQRRS..RTTFSASQLDEL
1aplC   t....KPYRGhr---FTKENVRIL
```

Note the loss of potential alignment to 1yrnB by 1aplC and 1mnmC.

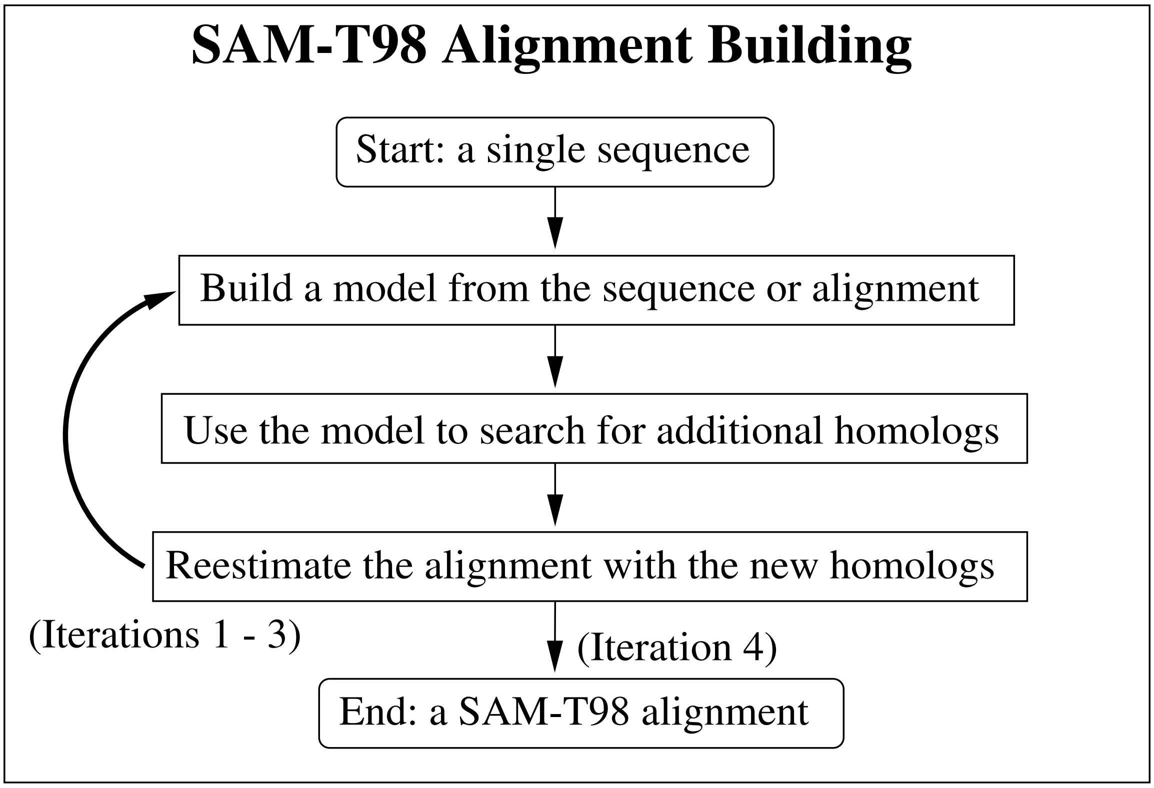
Constraining Seed Sequences

To keep seed sequences from shifting during alignment reestimation, *constraints* fix the residues of the seed sequences to certain states in the model (or columns in the alignment).



Advantages:

- The seed signal is not lost to unintended reestimation.
- The resulting alignments tend to be more accurate.



In a progressive method such as SAM-T98, care must be taken to avoid mistakes early in the process. Later, after the signal is established, less care is required.

Avoiding early mistakes

Database searched:

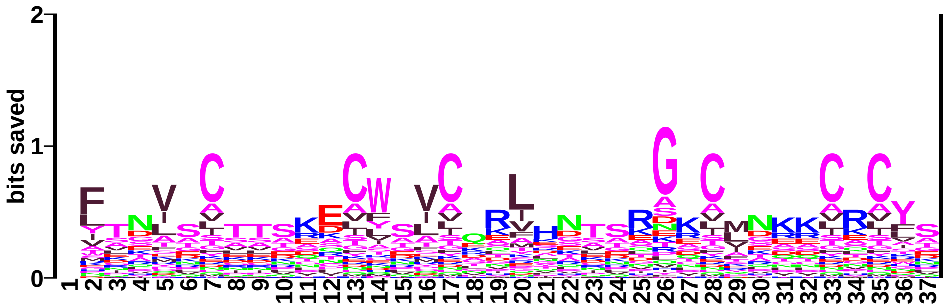
- During the first iteration, SAM-T98 searches against strong BLAST hits to admit only strong homologs.
- In subsequent iterations, the database searched is a large pool of potential but weaker homologs.

Threshold score:

- During the first iteration, the threshold score for admitting new homologs is very conservative (-40 nats).
- This threshold score is loosened with each iteration (-30, -24, and -16 nats).
- These thresholds were selected for superfamily recognition.

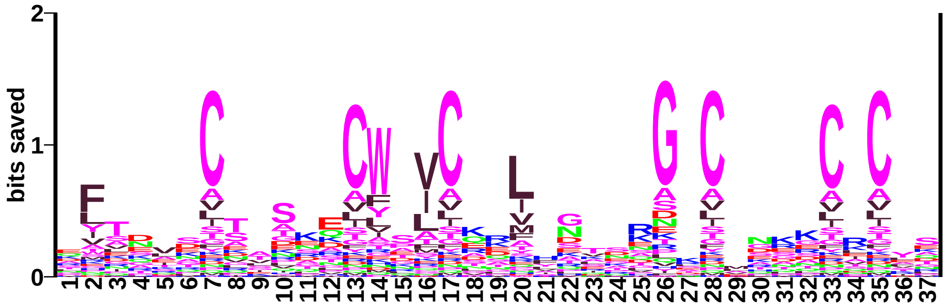
2crd, starting out

- 1 sequence aligned, total sequence weight = 0.91.
- To achieve the desired variability, the total sequence weight was set at less than one sequence.
- Initially, the glycine and the cysteines were more important, because the regularizer thinks these residues are more commonly conserved.



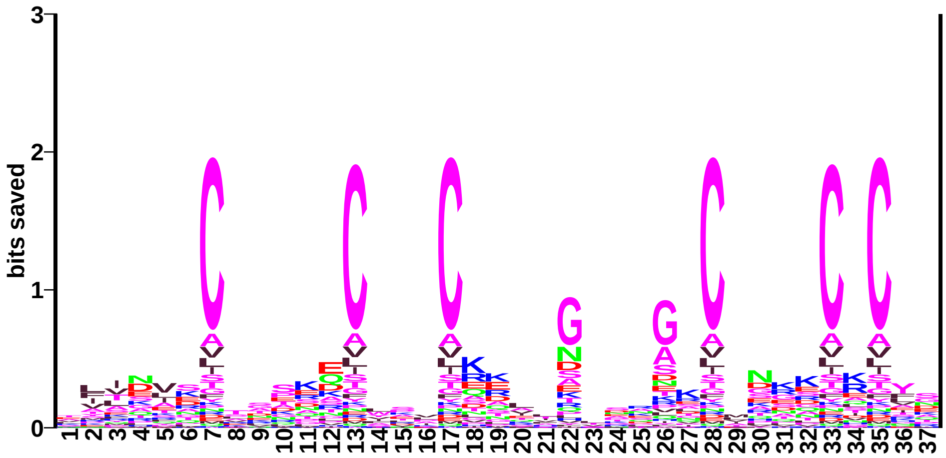
2crd, Iteration 1

- 14 sequences aligned, total sequence weight = 1.17.
- The alignment increased by 13 sequences, but the total sequence weight increased by only one fourth of one sequence. This indicates that the fourteen sequences were very close homologs.
- Tryptophan 14, Glycine 26, and all the cysteine signals increased substantially.
- An additional glycine signal started to appear at column 22.



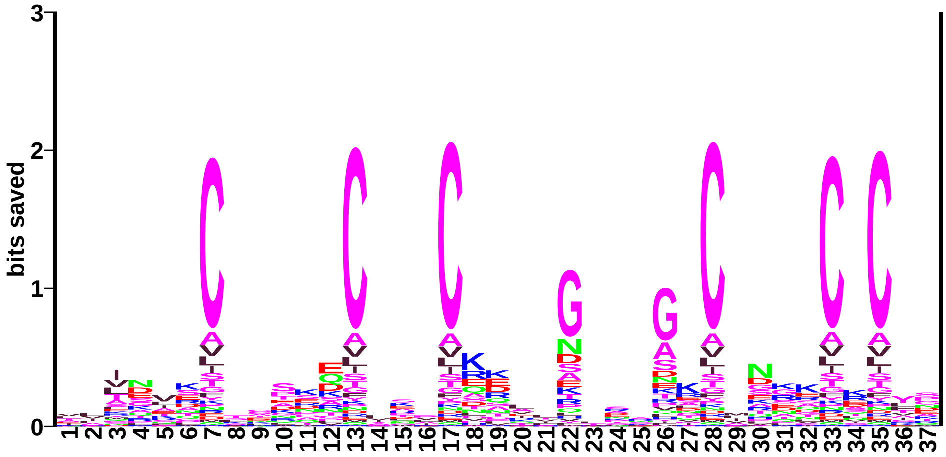
2crd, Iteration 2

- 28 sequences aligned, total sequence weight = 1.48.
- The cysteine signals increased.
- Tryptophan 14 disappeared.
- Glycine 22 increases, and Glycine 26 decreased.
- As the variability and the total sequence weight increased, the signal from the priors began to decrease.



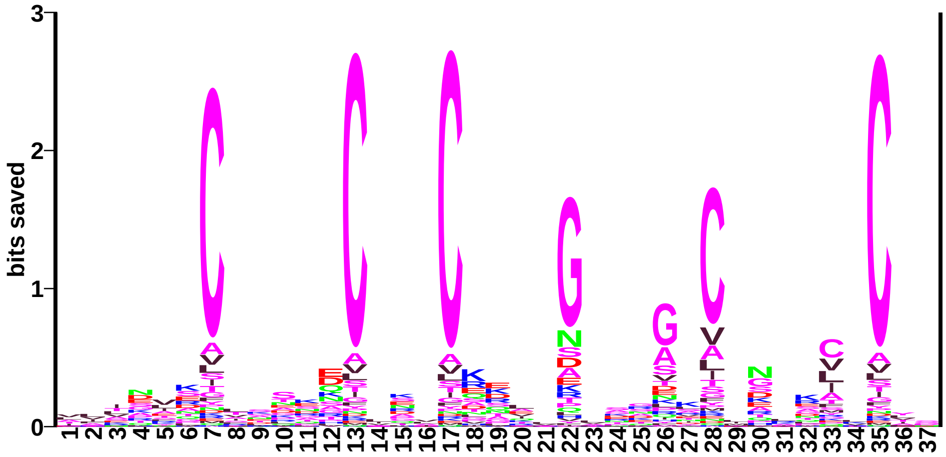
2crd, Iteration 3

- 41 sequences aligned, total sequence weight = 1.51.
- The variability within the alignment increased very slightly.
- Glycine 26 increased slightly.



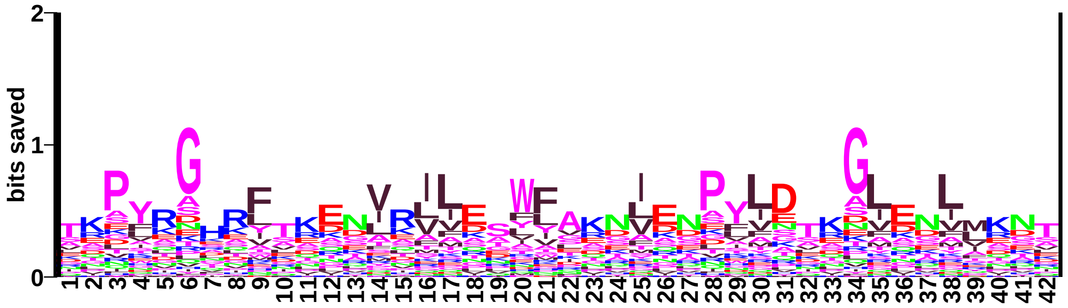
2crd, Iteration 4

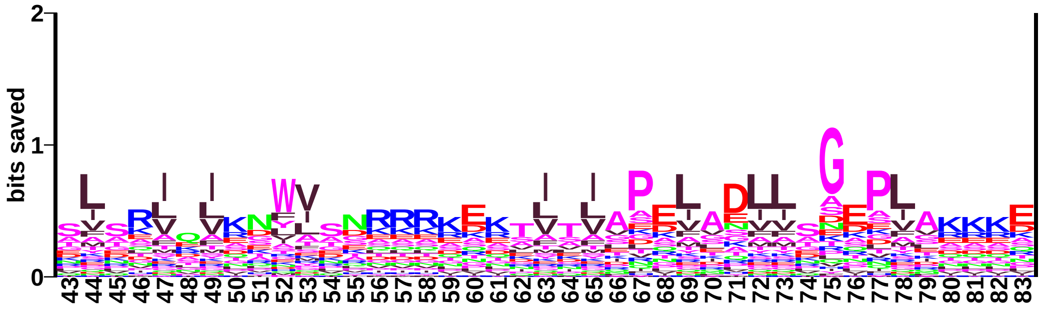
- 60 sequences aligned, total sequence weight = 1.99.
- The total sequence weight increased by a third, reflecting the variability of the new sequences in the alignment.
- Columns savings decreased toward the end of the alignment as the diverse new homologs were aligned to previously conserved columns rather than opening gaps. The effect was most vivid at Cysteine 33.
- The other cysteine signals increased substantially.
- Glycine 22 increased substantially.



1yrnB, Starting out

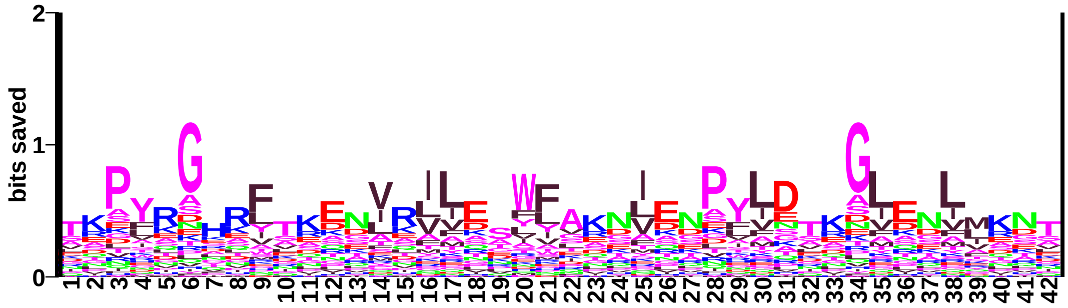
- 1 sequence aligned, total sequence weight = 0.91.
- To achieve the desired variability, the training alignment was weighted at slightly less than one sequence.
- The glycines at columns 6, 34, and 75 received special importance at the start.

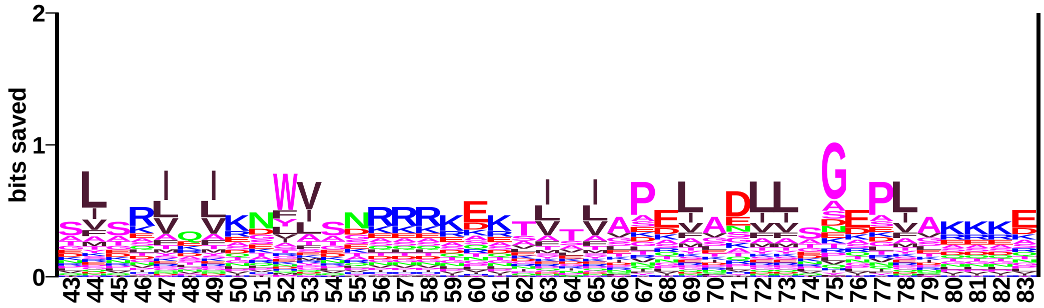




1yrnB, Iteration 1

- 9 sequences aligned, total sequence weight = 0.94.
- Eight very close homologs to 1yrnB were added to the alignment. Their scores were -100 and stronger.
- The homogeneity of the alignment was reflected by the very small increase in total sequence weight from 0.91.
- The glycines were further emphasized.

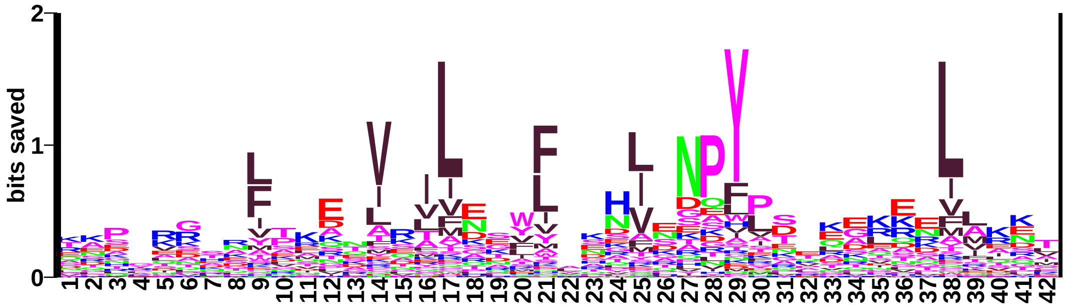


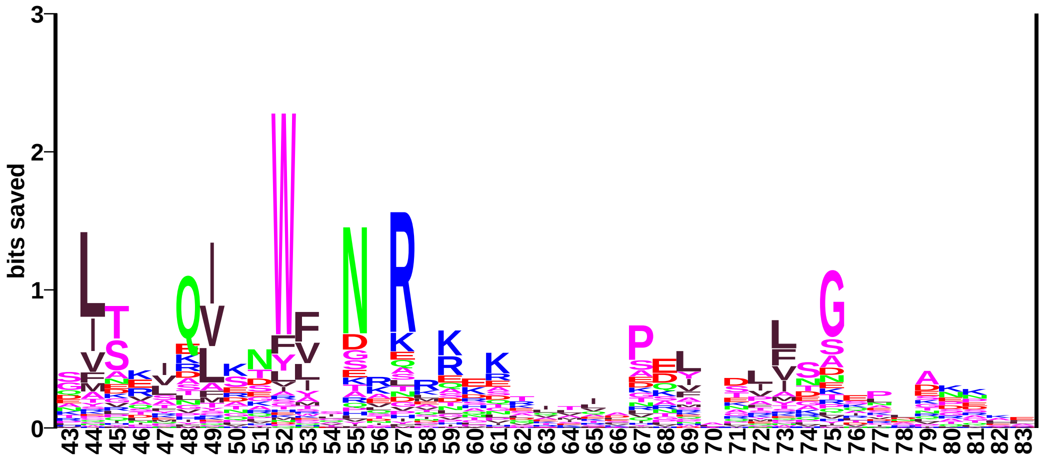


The alignment was unchanged in Iteration 2. No sequences met the Iteration 2 threshold but not the Iteration 1 threshold.

1yrnB, Iteration 3

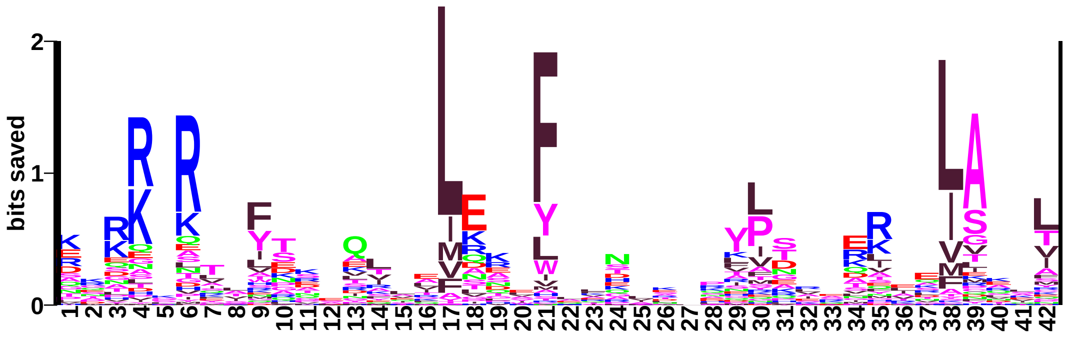
- 15 sequences aligned, total sequence weight = 2.05.
- The variability of the alignment increased substantially, and the total sequence weight more than doubled.
- Tryptophan 52 was emphasized.
- Leucines 17 and 38, Tyrosine 29, Asparagine 55, and Arginine 57 began to stand out. Less specific hydrophobic signals emerged at Columns 21, 44, and 49.

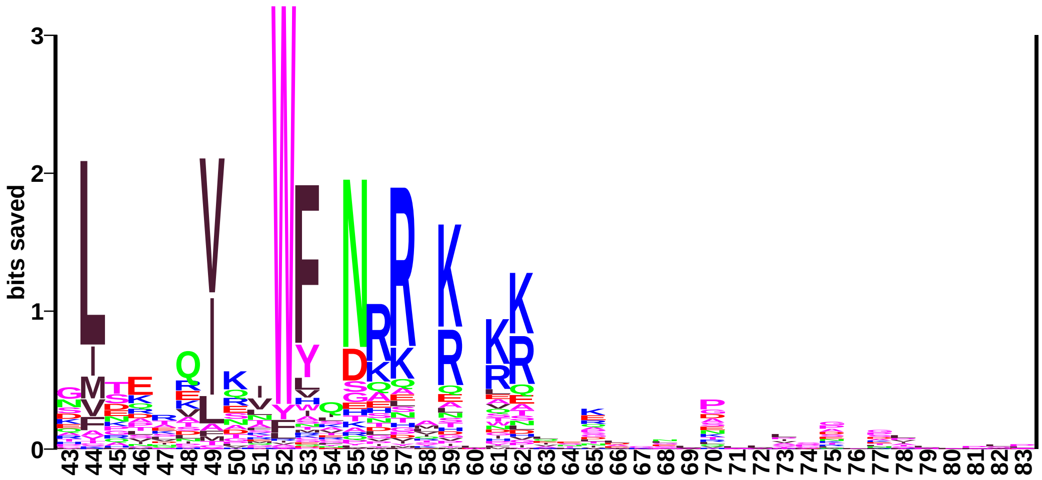


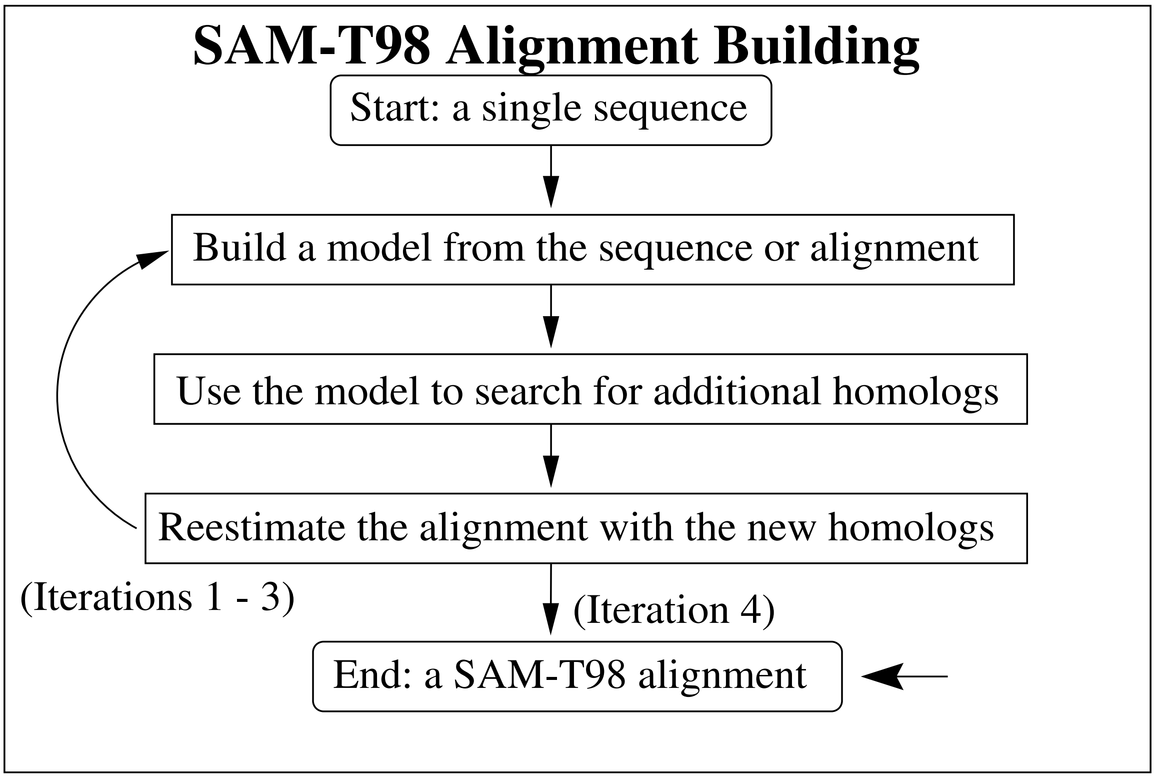


1yrnB, Iteration 4

- Number of homologs increased dramatically to 1999. Total sequence weight increased dramatically to 5.34.
- The most prominent signals from Iteration 3 were emphasized further, especially Leucine 17 and Tryptophan 52.
- Phenylalanine 21, Leucine 44, and Alanine 39 appeared.





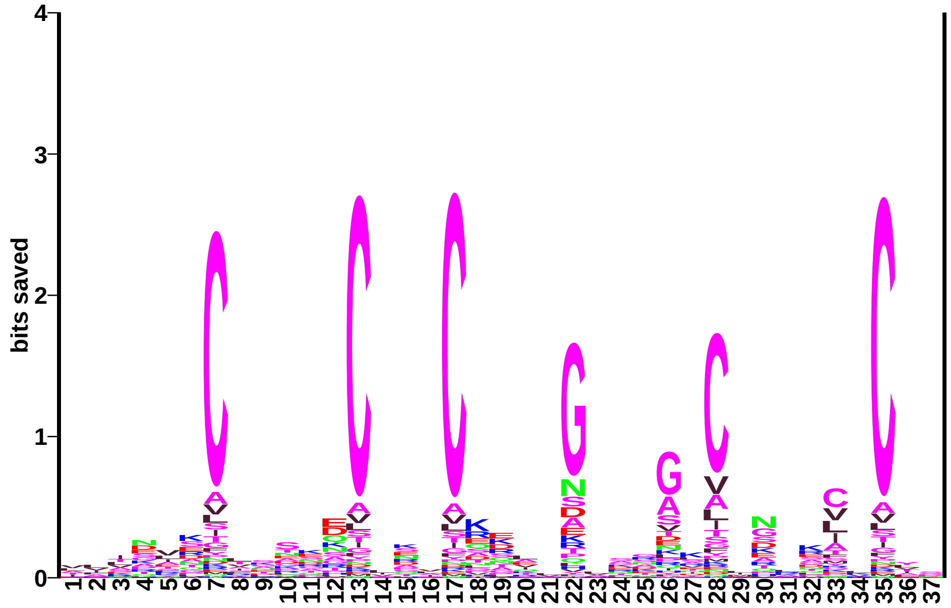


- An alignment can be valuable in its own right.
- If there are homologs with known structure, the final alignment might be sufficient for structure prediction.

Final SAM-T98 Alignment for 2crd

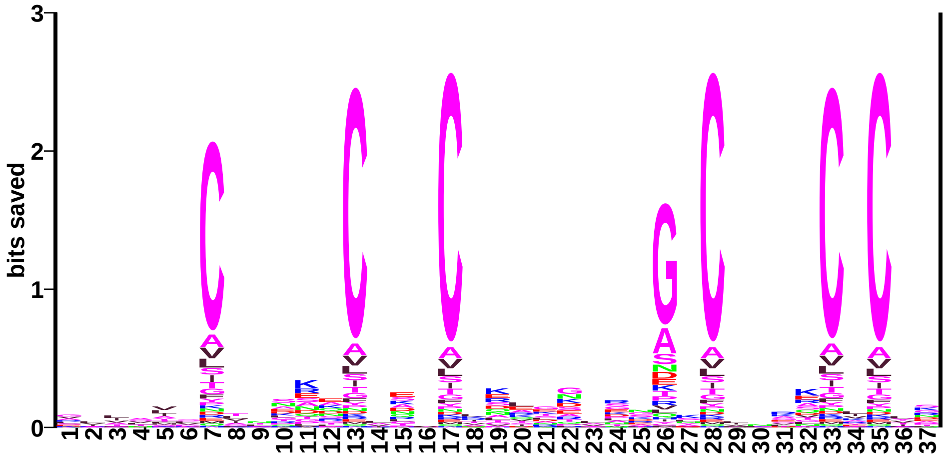
Shown below is an excerpt from the final SAM-T98 alignment for 2crd, showing remote homologs of known structure. The logo shown reflects the full alignment.

```
2crd XFTNVSCTTSKECWSVCQRLHNTSRG.KCMNKKCRCYS...
1bah XFTNVSCTTSKEXWSVCQRLHNTSRG.KCMNKKXRCYS...
1cmr ------CTTSKECWSVCQRLHNTSKG.WCDHRGCICES...
1lir XFTQESCTASNQCWSICKRLHNTNRG.KCMNKKCRCYS...
1sxm TIINVKCTSPKQCSKPCKELYGSSAGaKCMNGKCKCYNnx.
1txm ----VSCTGSKDCYAPCRKQTGCPNA.KCINKSCKCYGcx.
1big XFTDVKCTGSKQCWPVCKQMFGKPNG.KCMNGKCRCYS...
1bkt VGINVKCKHSGQCLKPCKDA-GMRFG.KCINGKCDCTPkx.
2bmt XFTNVSCSASSQCWPVCKKLFGTYRG.KCMNSKCRCYS...
```



The logo below represents the FSSP [HS97] structural alignment for 2crd. The FSSP logo and the SAM-T98 logo will differ due to homologs of unknown structure in SAM-T98 and additional distant homologs in FSSP.

- Cysteine 33 and Glycine 26 are emphasized less in the SAM-T98 alignment while Glycine 22 is emphasized more.
- Both alignments emphasize the cysteines.



SAM-T98 alignment for 1yrnB

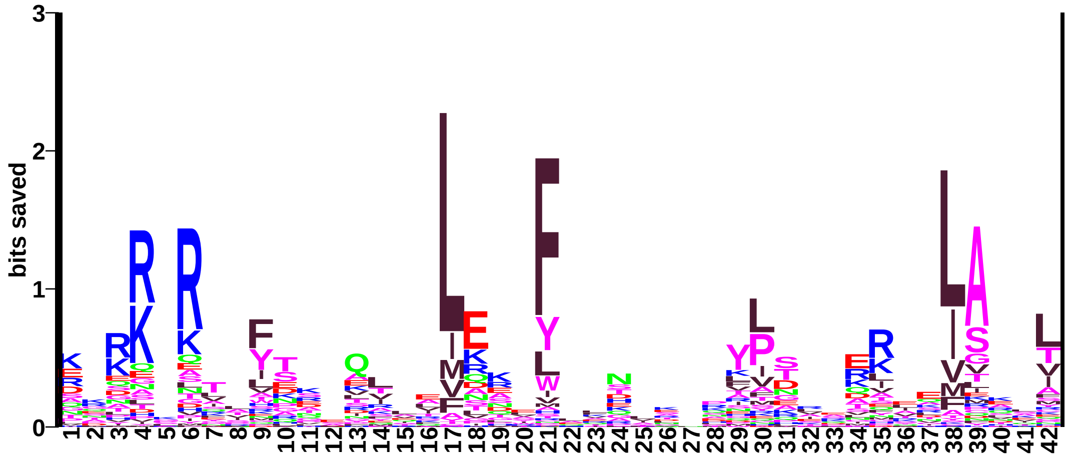
```
         10        20        30        40        50        60        70
          |         |         |         |         |         |         |
1yrnB GHRFTKENVRILESWFAKNIENPYLDTKGLENLMKNTSLSRIQIKNWVSNRRRKEKTITIAPELADLL
1hom  RQTYTRYQTLELEKEFHFN---RYLTRRRRIEIAHALCLTERQIKIWFQNRRMKWKKENKTKGEPG--
1hddC RTAFSSEQLARLKREFNEN---RYLTERRRQQLSSELGLNEAQIKIWFQNKRAKIKK----------
1fft  RVLFSQAQVYELERRFKQQ---KYLSAPEREHLASMIHLTPTQVKIWFQNHRYKMKRQAKDK------
1yrnA --SISPQARAFLEQVFRRK---QSLNSKEKEEVAKKCGITPLQVRVWFINKRMRSK-----------
1ocp  RTSIENRVRWSLETMFLKC---PKPSLQQITHIANQLGLEKDVVRVWFCNRRQKGKR----------
```

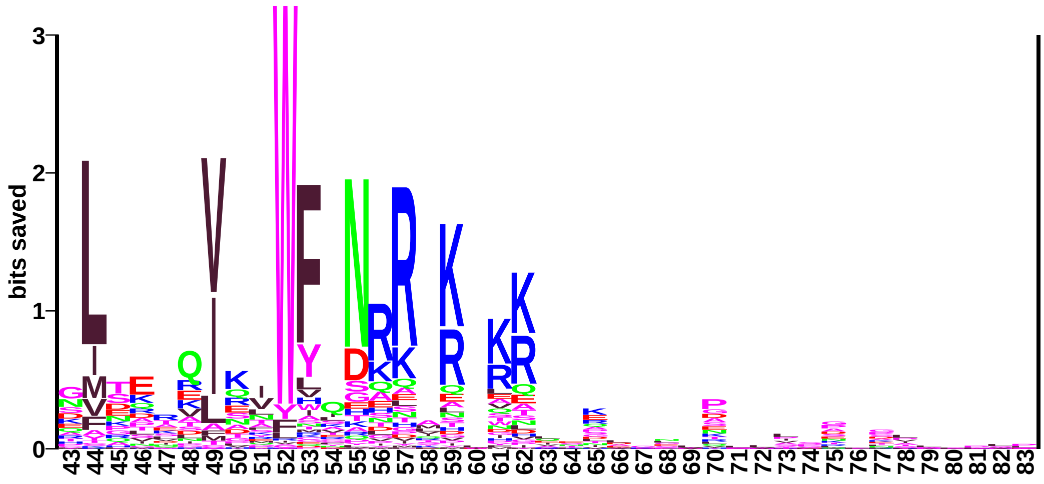
Shown is a portion of the final SAM-T98 alignment, showing selected sequences of known structure. For brevity, other sequences of known structure were omitted when their alignment was similar to some sequence shown. The sequences shown represent other sequences of known structure, as follows:

Representative	Other Structures Represented
1yrnB	1mnmC and 1aplC
1hom	1san, 1ahdP, 9antA, 1ftz, and 1b72A
1hddC	2hddA, 3hddA, 1enh, and 1fjlA
1fft	1vnd
1yrnA	1akhA and 1b72B
1ocp	1octC, 1pog, 1hdp, and 1au7A

Final SAM-T98 alignment for 1yrnB, continued

The logo shown represents the final SAM-T98 alignment for 1yrnB, including homologs of unknown structure.

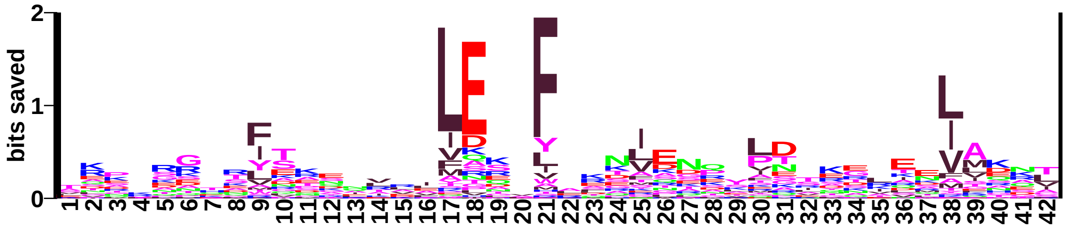


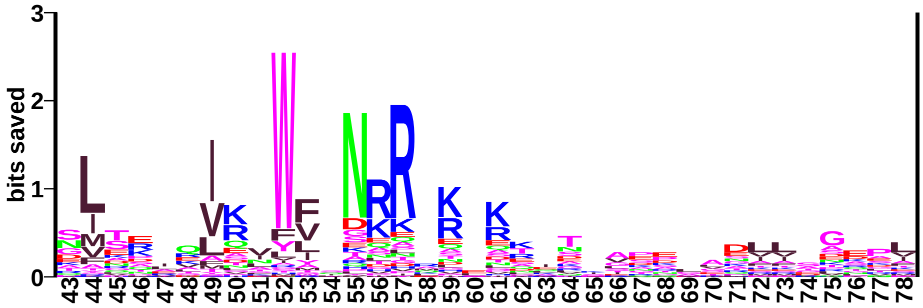


Comparison with the 1yrnB structural alignment

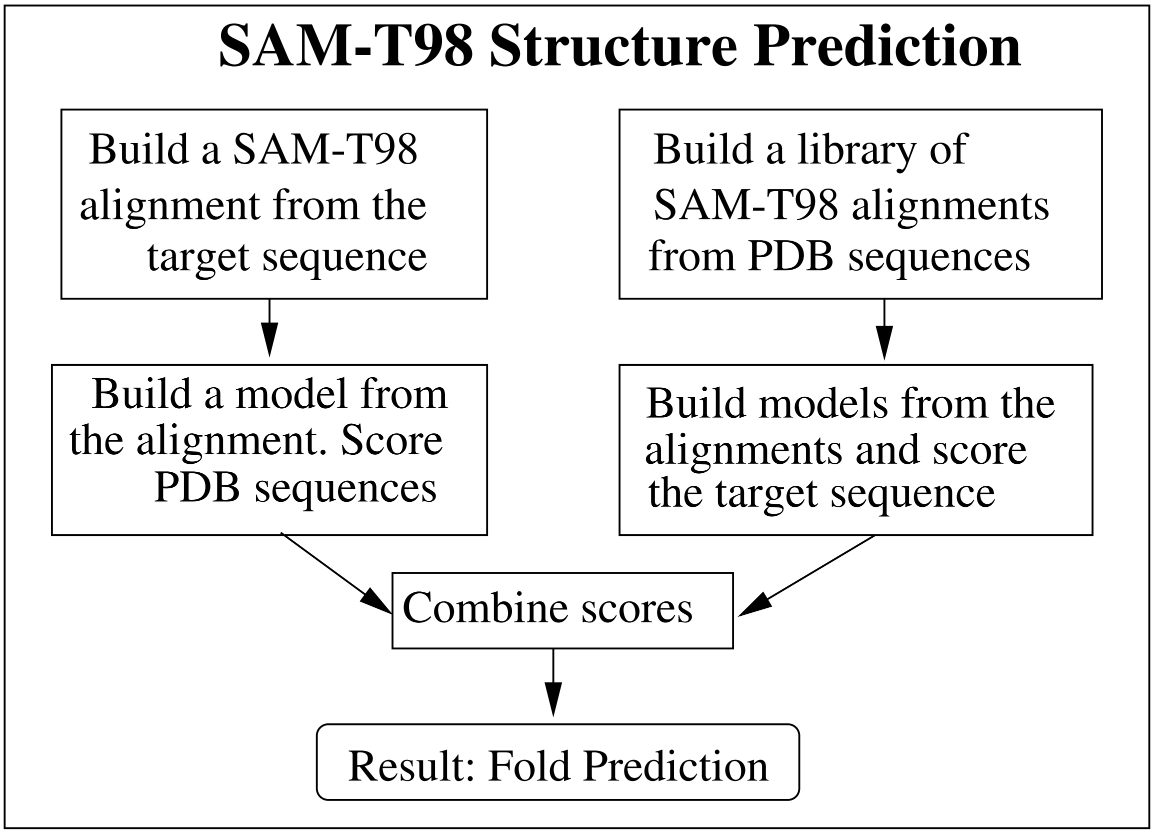
The logo below represents the 1yrnB structural family according to FSSP. The SAM-T98 logo will differ because of different sets of homologs, and slight differences in the 1yrnB sequence resulting from positions not fully resolved.

- FSSP places more emphasis on Glutamic Acid 18. SAM-T98 places more emphasis on Alanine 39 and others.
- The two logos reflect similar alignments.



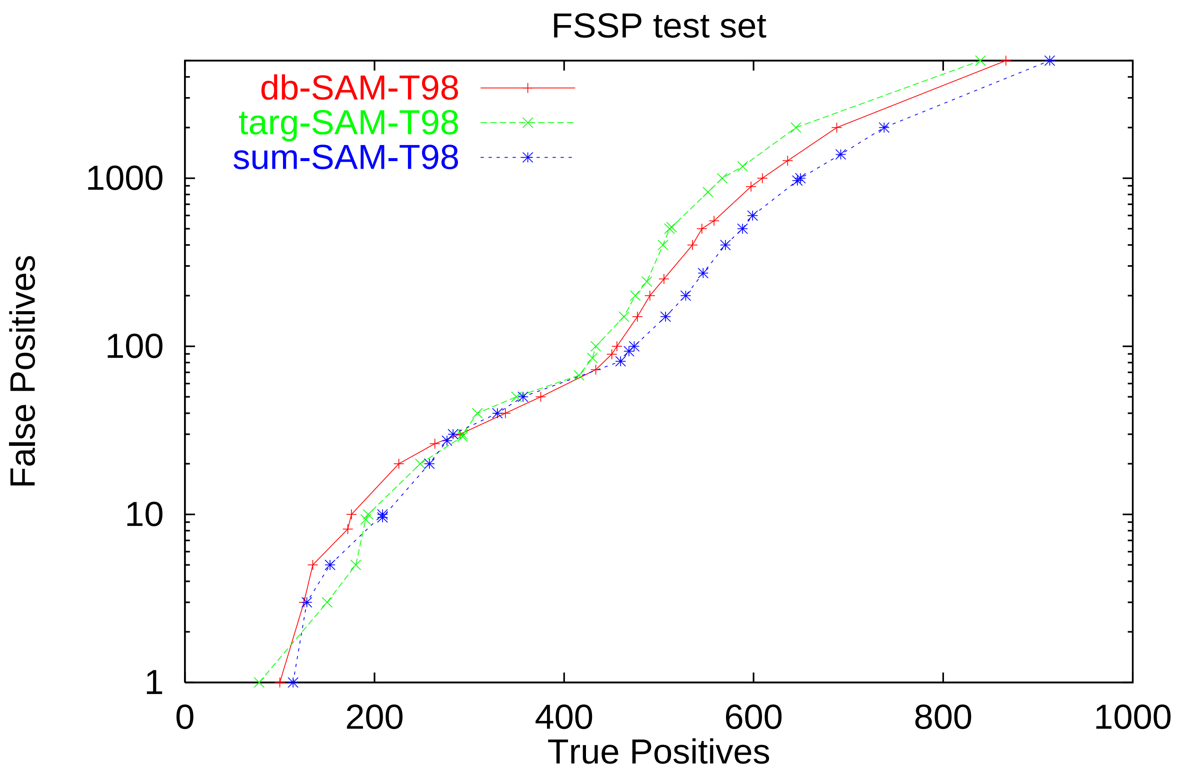


SAM-T98 structure prediction



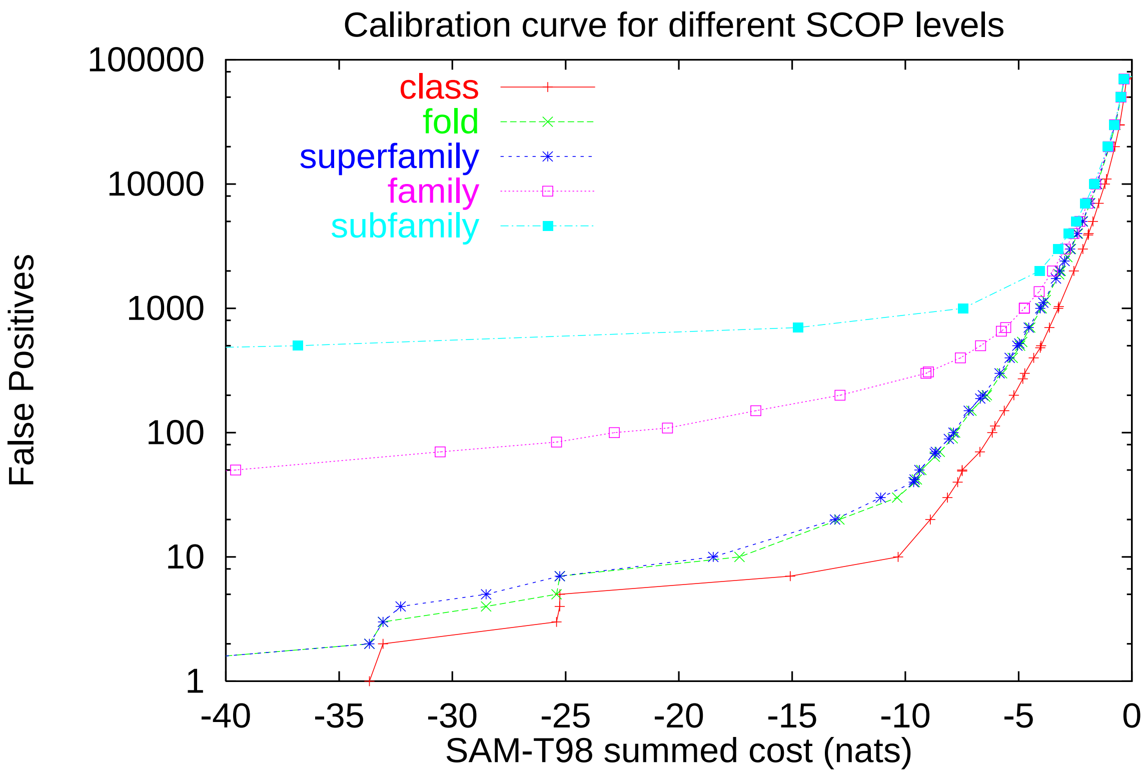
- Building the target models or the structure models... or both.
- Effects of bidirectional scoring.
- Issues in library building:
  - When is library building worthwhile?
  - Library size vs. structure space coverage.

Bidirectional scoring



- Bidirectional scoring yields a modest performance gain.
- The work necessary for bidirectional scoring is substantial compared to the work necessary to build one SAM-T98 alignment for the target sequence (targ-SAM-T98).

- 935 SCOP domains with known structural relatedness
- All-on-all scoring
- Rank by
  score and count number of false positives as a function of score.



- Allows us to assess significance of a score for different levels of protein relatedness.

BiDirectional Scoring Results for 2crd in rank order

The SCOP relatedness to 2crd is as follows: S = same subfamily, F = same family, P = same superfamily, U = unclassified, O = other.

	ID	SCOP		ID	SCOP
Score $\leq$ -50	**2crd**	S	-40 $<$ Score $\leq$ -30	1tsk	F
	1big	U		1cmr	S
	2bmt	U	-30 $<$ Score $\leq$ -20	1gps	P
	1lir	F		1gpt	P
	1agt	F		1acw	P
	1mtx	F		1cn2	U
	2ktx	F	-20 $<$ Score $\leq$ -10	1myn	P
	1ktx	F		1b3cA	U
	1sxm	F		2b3c	U
	1bah	S		1vnb	P
-50 $<$ Score $\leq$ -40	1sco	F		1vna	P
	1bkt	U		2sn3	P
	2pta	F			
	1txm	F			

BiDirectional Scoring Results for 2crd in rank order, contd.

- No known false positives were found.
- The more distant superfamily relatives appear with weak scores, but so also does one subfamily relative (1cmr).
- Since the method was tuned for distant homology detection, the signal for 1cmr was muted by the other family and superfamily inclusions.
- The alignment describes the family better than the subfamily.

BiDirectional Scoring Results for 1yrnB in rank order

The SCOP relatedness to 1yrnB is as follows: S = same subfamily, F = same family, U = unclassified, O = other.

	ID	SCOP		ID	SCOP
Score $\leq$ -60	2hoa	F	-50 $<$ Score $\leq$ -40	**1yrnB**	S
	1ahdP	F		1aplC	S
	1hom	F		1akhB	S
	9antA/B	U		1aplD	S
	1b8iA	U		1mnmC/D	S
	1san	U	-40 $<$ Score $\leq$ -30	1octC	F
	1b72A	U		1hdp	F
	3hddA/B	U		1yrnA	S
	1hddC	F		1akhA	S
	1fjlA/B/C	F		1pog	F
	1ftz	F		1au7A/B	F
	2hddA/B	F		1ocp	F
-60 $<$ Score $\leq$ -50	1nk2P	U	-30 $<$ Score $\leq$ -20	none	
	1nk3P	U	-20 $<$ Score $\leq$ -10	1lfb	F
	1vnd	F		2lfb	F
	1ftt	F			
	1enh	F			
	1b72B	U			
	1b8iB	U			

BiDirectional Scoring Results for 1yrnB in rank order, contd.

- No known false positives were found.
- The strongest hit was not the best structural match.
- The alignment provides a better description of the 1yrnB family than the 1yrnB subfamily because the method is tuned for superfamily modeling.

Should you build an HMM structure library?

- If you want to make structure predictions for only a few proteins, build HMMs for these proteins and score the protein structure sequence database.
- If you want to make structure predictions for 10,000 sequences (for example), it would be more efficient to build HMMs for protein structures.
- Anywhere in between these two extremes, the decision to build an HMM structure library depends on your time and computing power resources.

Building Alignments for a Structure Library

To achieve the best coverage with a limited number of alignments, use an established protein family hierarchy to choose a set of structures that covers “structure space” to a desired granularity.

FSSP: Superfamily level structure hierarchy [HS97]

CATH: Family level domain hierarchy [OMJ+97].

Pfam: Family level domain hierarchy [SED97].

SCOP: Multi-level hierarchy [HMBC97]. Includes some associations that are beyond most contemporary modeling methods.

All structure hierarchies are updated periodically, and the alignment library should also be updated accordingly.

Future directions in HMMs for sequence analysis

- HMM-based recognition and assembly of I-sites, motifs, and other small substructures [BB97, GBEB97].
- Fisher Kernel methods [JDH99]
- Using similarity between HMMs to predict homology (Lyngso ISMB99).
- Specialized hardware [HHKS96].
- Neural networks for improved parameter selection.

HMMs on the Web

- HMM software packages

  SAM http://www.cse.ucsc.edu/research/compbio/

  HMMer http://hmmer.wustl.edu/

- Applications of HMM software

  SAM-T98 www.cse.ucsc.edu/research/compbio/

  Pfam http://pfam.wustl.edu/

  Prosite http://www.expasy.ch/prosite/

  Meta-MEME http://metameme.sdsc.edu/

- Commercial vendors of HMM systems

  Neomorphic http://www.neomorphic.com/

  TimeLogic http://www.timelogic.com/

  Net-ID http://www.netid.com/

References

[AGM+90] Stephen F. Altshul, Warren Gish, Webb Miller, Myers Eugene W., and Lipman David J. Basic local alignment search tool. *JMB*, 215:403–410, 1990.

[BB97] C. Bystroff and D. Baker. Blind predictions of local protein structure in casp2 targets using the i-sites library. *Proteins: Structure, Function and Genetics, Suppl.*, 1:167–171, 1997.

[BHK97] Christian Barrett, Richard Hughey, and Kevin Karplus. Scoring hidden Markov models. *CABIOS*, 13(2):191–199, 1997.

[Edd98] S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–63, 1998.

[GBEB97] W. N. Grundy, W. Bailey, T. Elkan, and C. Baker. Meta-MEME: Motif-based hidden Markov models of protein families. *CABIOS*, 13(4):397–406, 1997.

[GME87] Michael Gribskov, Andrew D. McLachlan, and David Eisenberg. Profile analysis: Detection of distantly related proteins. *PNAS*, 84:4355–4358, July 1987.

[HH92] Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *PNAS*, 89:10915–10919, November 1992.

[HH94] Steven Henikoff and Jorja G. Henikoff. Position-based sequence weights. *JMB*, 243(4):574–578, November 1994.

[HHKS96] Jeffrey D. Hirschberg, Richard Hughey, Kevin Karplus, and Don Speck. Kestrel: A programmable array for sequence analysis. In *Application-Specific Array Processors*, pages 25–34, Los Alamitos, CA, July 1996. IEEE Computer Society.

[HK96] Richard Hughey and Anders Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996. Information on obtaining SAM is available at `http://www.cse.ucsc.edu/research/compbio/sam.html`.

[HMBC97] T. Hubbard, A. Murzin, S. Brenner, and C. Chothia. SCOP: a structural classification of proteins database. *NAR*, 25(1):236–9, January 1997.

[HS97] Liisa Holm and Chris Sander. Dali/FSSP classification of three-dimensional protein folds. *NAR*, 25:231–234, 1 Jan 1997.

[JDH99] T. Jaakkola, M. Diekhans, and D. Haussler. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, Aug 1999.

[Kar95] Kevin Karplus. Regularizers for estimating distributions of amino acids from small samples. In *ISMB-95*, Menlo Park, CA, July 1995. AAAI/MIT Press.

[KBC+99] Kevin Karplus, Christian Barrett, Melissa Cline, Mark Diekhans, Leslie Grate, and Richard Hughey. Predicting protein structure using only sequence information. *Proteins: Structure, Function, and Genetics*, to appear, 1999.

[KBH98] Kevin Karplus, Christian Barrett, and Richard Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[KBM+94] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *JMB*, 235:1501–1531, February 1994.

[KKB+97] Kevin Karplus, Kimmen Sjölander, Christian Barrett, Melissa Cline, David Haussler, Richard Hughey, Liisa Holm, and Chris Sander. Predicting protein structure using hidden Markov models. *Proteins: Structure, Function, and Genetics*, Suppl. 1:134–139, 1997.

[OMJ+97] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. Cath- a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–108, August 1997.

[PKB+98] J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. Sequence comparisons using multiple sequences detect twice as many remote homologues as pairwise methods. *JMB*, 284(4):1201–1210,

1998. Paper available at http://www.mrc-lmb.cam.ac.uk/genomes/jong/assess_paper/assess_paperNov.html.

[SED97] E.L.L Sonnhammer, S.R. Eddy, and R. Durbin. Pfam: A comprehensive database of protein families based on seed alignments. *Proteins*, 28:405–420, 1997.

[SKB+96] K. Sjölander, K. Karplus, M. P. Brown, R. Hughey, A. Krogh, I. S. Mian, and D. Haussler. Dirichlet mixtures: A method for improving detection of weak but significant protein sequence homology. *CABIOS*, 12(4):327–345, August 1996.

[SS90] T.D. Schneider and R.M Stephens. Sequence logos: a new way to display consensus sequences. *NAR*, 8(10):6097–100, 1990.

[SS96] Reinhard Schneider and Chris Sander. The HSSP database of protein structure-sequence alignments. *NAR*, 24(1):201–205, 1 Jan 1996.

[WU-] WU-BLAST WWW archives.
http://blast.wustl.edu/.