A Brief Overview of Gibbs Sampling

Eric C. Rouchka
Washington University
Institute for Biomedical Computing
Statistics Study Group
May 20, 1997

Gibbs Sampler

The purpose of this overview is to present the ideas of Gibbs sampling in terms of the data, parameters, model, and procedure both in a general sense and through an application of Gibbs sampling for multiple sequence alignment.

The first requirement for the Gibbs sampler is the observable data. The observed data will be denoted Y. In the general case of the Gibbs sampler, the observed data remains constant throughout.

Gibbs sampling requires a vector of parameters of interest that are initially unknown. These parameters will be denoted by the vector Φ. Nuisance parameters, Θ, are also initially unknown. The goal ofGibbs sampling is to find estimates for the parameters of interest in order to determine how well the observable data fits the model of interest, and also whether or not data independent of the observed data fits the model described by the observed data.

Gibbs sampling requires an initial starting point for the parameters. Then, one at a time, a value for each parameter of interest is sampled given values for the other parameters and data. Once all of the parameters of interest have been sampled, the nuisance parameters are sampled given the parameters of interest and the observed data. At this point, the process is started over. The power of Gibbs sampling is that the joint distribution of the parameters will converge to the joint probability of the parameters given the observed data.

Explanation in Mathematical Terms

The Gibbs sampler requires a random starting point of parameters of interest, Φ, and nuisance parameters, Θ, with observed data Y, from which a converging distribution can be found. For the sampler, there is an initial starting point $(\Theta_1^{(0)}, \Theta_2^{(0)}, \ldots, \Theta_D^{(0)}, \Phi^{(0)})$. Steps a-d are then repeatedly run.

a) Sample $\Theta_1^{(i+1)}$ from $p(\Theta_1 | \Theta_2^{(i)},...,\Theta_D^{(i)},\Phi^{(i)},Y)$.

b) Sample $\Theta_2^{(i+1)}$ from $p(\Theta_2 | \Theta_1^{(i+1)},\Theta_3^{(i)},...,\Theta_D^{(i)},\Phi^{(i)},Y)$.

c) Sample $\Theta_D^{(i+1)}$ from $p(\Theta_D | \Theta_1^{(i+1)},...,\Theta_{(D-1)}^{(i+1)},\Phi^{(i)},Y)$.

d) Sample $\Phi^{(i+1)}$ from $p(\Phi | \Theta_1^{(i+1)},...,\Theta_D^{(i+1)},Y)$.

The vectors $\Theta^{(0)},\Theta^{(1)},...,\Theta^{(t)}$ represent the realization of a Markov chain, where the transition probability from Θ' to Θ is defined as:

$$K(\Theta',\Theta) = p(\Theta_1 | \Theta_2^{'},...,\Theta_D^{'},\Phi^{'},Y) * p(\Theta_2 | \Theta_1,\Theta_3^{'},...\Theta_D^{'},\Phi^{'},Y)...*$$

$$p(\Theta_D | \Theta_1,...,\Theta_{(D-1)},\Phi^{'},Y)$$

The joint distribution of $(\Theta_1^{(i)},...,\Theta_D^{(i)},\Phi^{(i)})$ converges geometrically to $p(\Theta_1,...,\Theta_D,\Phi | Y)$ as $i \rightarrow \infty$.

The Gibbs sampler differs from the Metropolis algorithm because in each step only one parameter, Θ_d, is allowed to change.

Multiple Alignment Using Gibbs Sampling

One application of Gibbs sampling useful in computational molecular biology is the detection and alignment of locally conserved regions (motifs) in sequences of amino acids or nucleic acids assuming no prior information in the patterns or motifs. Gibbs sampling strategies claim to be fast and sensitive, avoiding the problem that EM algorithms fall into as far as getting trapped by local optima. As an example, a set of 29 DNA sequences have been provided. These sequences contain sequences necessary for recognition byerythroid transcription factors, most notably a six nucleotide GATA binding site.

Basic Algorithm

First the basic multiple alignment strategy is examined where a single motif is desired. The most basic implementation, known as a site sampler, assumes that there is exactly one motif element located within each sequence.

Notation

- N : number of sequences
- $S_1 \ldots S_N$: set of sequences
- W : width of motif to be found in the sequences
- J : the number of residues in the alphabet. $J = 4$ for nucleic acid sequences and 20 for amino acid sequences.
- $c_{i,j,k}$: Observed counts of residue j in position i of motif k. j ranges from $1 \ldots J$. i ranges from $0..W$ where $c_{0,j}$ contains the counts of residue j in the background. If it is assumed that only a single motif is searched for, the k term can drop out.
- $q_{i,j}$: frequency of residue j occurring in position i of the motif. i ranges from $0..W$ as above. Note that in the literature, $q_{0,j}$ (the vector of background residue frequencies) is sometimes denoted as p_j. This is the parameter of interest, Φ.
- a_k : vector of starting positions of the motifs within the sequences. k ranges from $1..N$. This is the nuisance parameter, Θ.
- b_j : pseudocounts for each residue – needed according to Bayesian statistical rules to eliminate problems with zero counts.
- B : The total number of pseudocounts. $B = \sum_j b_j$.

Initialization

Once the sequences are known, the counts for each residue can calculated. Initially, $c_{0,j}$ will contain the total counts of residue j within all of the sequences and $c_{i,j}$ is initialized to 0 for all other values of i . This is a summary observed data. The site sampler is then initialized by randomly selecting a position for the motif within each sequence and recording these positions in a_k . The counts are updated according to this initial alignment. After the observed counts are set, $q_{i,j}$ can be calculated.

$$q_{i,j} = \frac{c_{i,j} + b_j}{N - 1 + B}$$

Equation 1: Motif Residue Frequencies

$$q_{0,j} = \frac{c_{0,j} + b_j}{\sum_{k=1}^{j} c_{0,k} + B}$$

Equation 2 : Background Residue Frequencies

Predictive Update Step

The first step, known as the predictive update step, selects one of the sequences and places the motif within that sequence in the background and updates the residue counts. One of the N sequences, z, is chosen. The motif in sequence z is taken from the model and placed in the background. The observed counts $c_{i,j}$ are updated as are the frequencies $q_{i,j}$. The selection of z can be random or in a specified order.

Sampling Step

In the sampling step, a new motif position for the selected sequence is determined by sampling according to a weight distribution. All of the possible segments of width W within sequence z are considered. For each of these segments x, a weight A_x is calculated according to the ratio $A_x = \frac{Q_x}{P_x}$ where $Q_x = \prod_{i=1}^{W} q_{i,r_i}$ is the model residue frequency according to equation 1 if segment x is in the motif model, and $P_x = \prod_{i=1}^{W} q_{0,r_i}$ is the background residue frequency according to equation 2. r_i refers to the residue located at position i of segment x. Once A_x is calculated for every possible x, a new position a_z is chosen by randomly sampling over the set of weights A_x. Thus, possible starting positions with higher weights will be more likely to be chosen as the new motif position than those position with lower weights. Since this is a stochastic process, the starting position with the highest weight is not guaranteed to be chosen.

Once the iterative predictive update and sampling steps have been performed for all of the sequences, a probable alignment is present. For this alignment, a maximum a posteriori (MAP) estimate can be calculated using equation 3:

$$F = \sum_{i=1}^{W} \sum_{j=1}^{J} c_{i,j} \log \frac{q_{i,j}}{q_{0,j}}$$

Equation 1: Alignment conditional log-likelihood

The goal is to maximize F. This is accomplished in the following manner:

```
globalMaxAlignmentProb = 0
For Iteration = 1 to 10:
     Initialize Random alignment
     localMaxAlignmentProb = 0;
     while (not in local maximum and innerloop < MAXLOOP) do
           for each sequence do{
               Predictive Update
               Sample
           }
           calculate AlignmentProb
           if (AlignmentProb > localMaxAlignmentProb)
           {
              localMaxAlignmentProb = AlignmentProb;
              not in local maximum = true;
           }
           innerloop++;
    }
    if (localMaxAlignmentProb == globalMaxAlignmentProb)
        exit → max found twice
    else if (localMaxAlignmentProb > globalMaxAlignmentProb)
         globalMaxAlignmentProb = localMaxAlignmentProb
}
```

Explanation

The idea is that the more accurate the predictive update step is, the more accurate the sampling step will be since the background will be more distinguished from the motif description. Given random positions a_k in the sampling step, the pattern description $q_{i,j}$ will not favor any particular segment. Once some correct a_k have been selected by chance, the $q_{i,j}$ begins to favor a particular motif.

Details

There are a couple of problems that need to be addressed. First, it is possible that the correct pattern has not been chosen, but rather a shift of it has. This can be taken care of by shifting the alignment to the left and right by a specified number of columns and sampling from the values of F .

Another problem is that the pattern width W must also be specified. In order to decide what the width should be, the incomplete-data log-probability ratio as shown in Equation 4 can be implemented.

$$G = F - \sum_{i=1}^{N} (\log L_i' + \sum_{j=1}^{L_i'} Y_{i,j} \log Y_{i,j})$$

Equation 2 : Incomplete-data log-probability ratio

Where L_i' is the number of the possible positions for the pattern within sequence i and $Y_{i,j}$ is the normalized weight of position j . Dividing G by the number of free parameters needed to specify the pattern ($19 * W$ for protein sequences, $3 * W$ for nucleotide sequences) results in an information per parameter quantity. It is then desired to maximize the information per parameter to determine the value of W .

Improving the Algorithm

The method of determining motifs as described above requires multiple runs on the same data set with varying widths to find the correct pattern size. The *Protein Science* paper discusses a method to determine the width of the motif in a single run of the program, while at the same time determining gaps within the motif. The Gibbs sampler described thus far also requires the existence of exactly one motif in each sequence. Another improvement made is to allow multiple motifs within sequences, and allow the possibility that a sequence does not have any motifs. The improvements made within the *Protein Science* paper describe a technique known as a motif sampler.

Allowing a Variable Number of Motif Sites

Assume that there are m different motif patterns that we are searching for in the sequences. Let n_k represent the number of sites matching motif k in the sequence. Initially, it is not known how many motif sites there are. To overcome this, we make a prior expectation e_k for each n_k . The new algorithm allows the prior expectations to become posterior expectations as it learns the number of sites for each motif. For the initialization step, e_k random starting points are selected for motif pattern k instead of selecting one starting point randomly within each of the N sequences. Now we can go through all of the possible motif starting locations in each sequence and decide if it is a motif starting site by using equation 5.

$$\frac{p_j}{1 - p_j} A_x$$

Equation 3: Current motif site probability

Where p_j is the posterior probability that any site belongs to the model (see the appendix of the *Protein Science* paper for the prior and posterior calculations of P_j), and A_x is the same as in the site sampler.

Width Optimization by Column Sampling

In order to help introduce gaps and include only the most informative positions of the motif, column sampling is introduced where only$C$ columns out of a specified number of contiguous columns $w_{\max} \geq C$ are used for the residue frequency model. This is accomplished in a two step process. First, turn off one column either randomly or by selecting it proportional to how little information it provides. Then sample one of the columns that are turned off proportional to how information rich it is and turn it on. The column move operations need to be weighted in order to assure that there is not a bias to longer motif widths. A discussion is provided in the appendix of the*Protein Science* paper.

Properties of Gibbs Sampling

- Requires relatively large sets (15 or more sequences) for weakly conservedpatterms to reach statistical significance.

- It is a heuristic and not an exhaustive search, so you are not guaranteed to reach an optimal value, but you will not get stuck in local maximums the way EM algorithms do.

- Have to have some idea of how wide the motif is and how many motifs there are for the algorithm to work best.

- Gibbs sampling allows you to viewsuboptimal results.

- Fast and sensitive – generally finds an optimized local alignment model for N-sequences in N-linear time.

Site Sampler Example

The site sampler is tested using a set of erythroid sequences. The set is tested for the presence of a GATA box, which should have a sequence (T/A)GATA(A/G), which in the reverse complement is (C/T)TATC(A/T). Since the width of the GATA box is shown, it is known that for this example $W = 6$. The process of determining the best alignment using the site sampler is described.

Initialization

The first step in the site sampler is to randomly assign an alignment to the set of sequences. Figure 1 indicates one such random alignment.

TCAGAACCAGTTATAA**ATTTAT**CATTTCCTTCTCCACTCCT
CCCACGCA**GCCGCC**CTCCTCCCCGGTCACTGACTGGTCCTG
TCGACCCTCTGAACCTATCAGGGACCA**CAGTCA**GCCAGGCAAG
AAAACACTTGAG**GGAGCA**GATAACTGGGCCAACCATGACTC
GGGTGAATGGTACTGCT**GATTAC**AACCTCTGGTGCTGC
AGCCTAGAGT**GATGAC**TCCTATCTGGGTCCCCAGCAGGA
GCCTCAGGATCCAGCACACAT**TATCAC**AAACTTAGTGTCCA
CATTATCAC**AAACTT**AGTGTCCATCCATCACTGCTGACCCT
TCGGAACAAGGCAAA**GGCTAT**AAAAAAAATTAAGCAGC
GCCCCTTCCCCA**CACTAT**CTCAATGCAAATATCTGTCTGAAACGGTTCC
CATGCCCTCAAGTGTGCAGATTGGT**CACAGC**ATTTCAAGG
GATTGGTCACAGCAT**TTCAAG**GGAGAGACCTCATTGTAAG
TCCCCAACTCCCAACTGACCTTAT**CTGTGG**GGGAGGCTTTTGA
CCTTATCTGT**GGGGGA**GGCTTTTGAAAAGTAATTAGGTTTAGC
ATTATTTTCCTTATCAGAAGC**AGAGAG**ACAAGCCATTTCTCTTTCCTCCCGGT
AGG**CTATAA**AAAAAATTAAGCAGCAGTATCCTCTTGGGGGCCCCTTC
CCAGCACACACACTTATC**CAGTGG**TAAATACACATCAT
TCAAATAGGTACGGATAAG**TAGATA**TTGAAGTAAGGAT
ACTTGGGGTTCCAGTTTGATAAGAAAAGACTT**CCTGTG**GA
TGGCCGC**AGGAAG**GTGGGCCTGGAAGATAACAGCTAGTAGGCTAAGGCCAG
CAACCACAACCTCTGTATCCGGTAGTGGCAGATGGAAA
CTGTATCCGGTAG**TGGCAG**ATGGAAAGAGAAACGGTTAGAA
GAAAAAAAATAAATGAAGTCTGCC**TATCTC**CGGGCCAGAGCCCCT
TGCCTTGTCTGTTGTAGATAATGAATCTATCCTCCA**GTGACT**
GGCCAGGCTGAT**GGGCCT**TATCTCTTTACCCACCTGGCTGT
CAACAGCAGGTCCTACTATCGCCTCCCTCT**AGTCTC**TG
CCAACCG**TTAATG**CTAGAGTTATCACTTTCTGTTATCAAGTGGCTTCAGCTATGCA
GGGAGGGTGGGGCCCCTATCTCTCCTA**GACTCT**GTG
CTTTGTC**ACTGGA**TCTGATAAGAAACACCACCCCTGC

Figure 1: Initial Motif Sites for Site Sampler

The number of A's in all of the sequences combined is 327, the number of C's is 317, the number of G's is 272, and the number of T's is 304. If we assume that thepseudocounts

needed for Bayesian statistics are equal to 10% of the observed nucleotide frequencies, we know the following information before any of the initial motif sites are set:

$$c_{0,1} = 327; \quad c_{0,2} = 317; \quad c_{0,3} = 272; \quad c_{0,4} = 304; \quad \sum_{i=1}^{4} c_{0,i} = 1220$$

$$b_1 = 32.7; \quad b_2 = 31.7; \quad b_3 = 27.2; \quad b_4 = 30.4; \quad B = 122$$

If we assume we have the initial random alignment as described in figure 1, we can recalculate the counts and calculate the residue frequencies. Table 1 gives the results of these calculations.

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	279	6	12	6	6	11	7
C	280	8	3	5	7	7	7
G	225	9	8	10	7	5	8
T	262	6	6	8	9	6	7

Table 1: Caclulation of observed counts for inital alignment of figure 1

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	.267	.256	.296	.256	.256	.289	.263
C	.267	.263	.230	.243	.256	.256	.256
G	.216	.240	.233	.246	.226	.213	.233
T	.250	.241	.241	.254	.261	.241	.248

Table 2: Calculation of residue frequencies for initial alignment of figure 1

Predictive Update Step

Now that we have the initial random alignment for the site sampler, we begin the predictive update step by choosing one of the sequences to update. For simplicity, let's just choose the first sequence. In the predictive update stage, the motif for the selected sequence is placed in the background and the counts and frequencies are updated. Since the motif in the first sequence is ATTTAT, tables 1 and 2 can be recalculated.

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	281	5	12	6	6	10	7
C	280	8	3	5	7	7	7
G	225	9	8	10	7	5	8
T	266	6	5	5	5	6	6

Table 3: Recalculated observed counts

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	.267	.251	.298	.258	.258	.285	.265
C	.265	.264	.231	.245	.258	.258	.258
G	.215	.241	.235	.248	.228	.215	.235
T	.252	.243	.236	.236	.236	.243	.243

Table 4: Recalculated residue frequencies

Sampling Step

Once the counts and frequencies have been updated in the predictive update step, we can begin the sampling step. In this step, we look at all of the possible motif starting positions within the sequence selected in the predictive update step. Since the length of the sequence is 41 and the width of the motif is 6, there are 41 - 6 + 1 = 36 possible starting sites. We calculate the probability of each of these sites being in the model and then sample from their weights. Table 5 shows the weights for sequence 1 given table 3 and 4.

Using the information in table 5, one of the segments will be sampled in according to the normalized value of A_x . The predictive update and sampling steps are repeated for each of the sequences. Once each of the sequences have been sampled, an alignment is present and the alignment probability is tested. This procedure is repeated until a plateau is reached. Then another initial random alignment is tested and the process begins again. For the example used thus far, the final alignment is as shown in figure 2. This alignment yields the counts and frequencies described in tables 6 and 7.

Sequence	**X**	**A_x**	**Normalized A_x**
TCAGAA	1	0.906	.030
CAGAAC	2	1.283	.042
AGAACC	3	0.905	.030
GAACCA	4	1.131	.037
AACCAG	5	1.093	.036
ACCAGT	6	0.700	.023
CCAGTT	7	0.822	.027
CAGTTA	8	1.142	.037
AGTTAT	9	0.919	.030
GTTATA	10	0.902	.029
TTATAA	11	0.856	.028
TATAAA	12	1.021	.033
ATAAAT	13	0.839	.027
TAAATT	14	0.923	.030
AAATTT	15	0.875	.029
AATTTA	16	0.873	.028
ATTTAT	17	0.787	.026
TTTATC	18	0.757	.025
TTATCA	19	0.781	.025
TATCAT	20	0.997	.033
ATCATT	21	0.723	.024
TCATTT	22	0.698	.023
CATTTC	23	0.907	.030
ATTTCC	24	0.725	.024
TTTCCT	25	0.762	.025
TTCCTT	26	0.743	.024
TCCTTC	27	0.674	.022
CCTTCT	28	0.709	.023
CTTCTC	29	0.791	.026
TTCTCC	30	0.731	.024
TCTCCA	31	0.732	.024
CTCCAC	32	0.864	.028
TCCACT	33	0.696	.023
CCACTC	34	0.761	.025
CACTCC	35	0.904	.030
ACTCCT	36	0.695	.023

Table 5 :Weights for segments within sequence 1

TCAGAACCAGTTATAAAT**TTATCA**TTTCCTTCTCCACTCCT

CCCACGCAGCCGCCCTCCTCCC**CGGTCA**CTGACTGGTCCTG

TCGACCCTCTGGAAC**CTATCA**GGGACCACAGTCAGCCAGGCAAG

AAAACACTTGAGGGAGC**AGATAA**CTGGGCCAACCATGACTC

GGGTGAA**TGGTAC**TGCTGATTACAACCTCTGGTGCTGC

AGCCTAGAGTGATGACTC**CTATCT**GGGTCCCCAGCAGGA

GCCTCAGGATCCAGCACACA**TTATCA**CAAACTTAGTGTCCA

CA**TTATCA**CAAACTTAGTGTCCATCCATCACTGCTGACCCT

TCGGAACAAGGCAAAGG**CTATAA**AAAAAATTAAGCAGC

GCCCCTTCCCCACA**CTATCT**CAATGCAAATATCTGTCTGAAACGGTTCC

CATGCCCTCAAGTGTGCAGAT**TGGTCA**CAGCATTTCAAGG

GATTGGTCACAGCATTTCAAGGGAGAGACCTCA**TTGTAA**G

TCCCCAACTCCCAACTGACC**TTATCT**GTGGGGGAGGCTTTTGA

CC**TTATCT**GTGGGGGAGGCTTTTGAAAAGTAATTAGGTTTAGC

ATTATTTTCC**TTATCA**GAAGCAGAGAGACAAGCCATTTCTCTTTCCTCCCGGT

AGG**CTATAA**AAAAAATTAAGCAGCAGTATCCTCTTGGGGGCCCCTTC

CCAGCACACACAC**TTATCC**AGTGGTAAATACACATCAT

TCAAATAGGTACGGATAAGT**AGATAT**TGAAGTAAGGAT

ACTTGGGGTTCCAGTT**TGATAA**GAAAAGACTTCCTGTGGA

TGGCCGCAGGAAGGTGGGCCTGGA**AGATAA**CAGCTAGTAGGCTAAGGCCAG

CAACCACAACCT**CTGTAT**CCGGTAGTGGCAGATGGAAA

CTGTATCCGGTAGTGGCAGATGGAAAGAGAAACGGTTAGAA

GAAAAAAAATAAATGAAGTCTGC**CTATCT**CCGGGCCAGAGCCCCT

TGCC**TTGTCT**GTTGTAGATAATGAATCTATCCTCCAGTGACT

GGCCAGGCTGATGGGCC**TTATCT**CTTTACCCACCTGGCTGT

CAACAGCAGGTCCTACTATCGCCTCCCTC**TAGTCT**CTG

CCAACCGTTAATGCTAGAGTTATCACTTTCTG**TTATCA**AGTGGCTTCAGCTATGCA

GGGAGGGTGGGGCCC**CTATCT**CTCCTAGACTCTGTG

CTTTGTCACTGGATC**TGATAA**GAAACACCACCCCTGC

Figure 2: Final Alignment for Site Sampler

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	276	3	1	21	0	11	15
C	287	10	0	0	0	18	2
G	256	0	8	8	0	0	0
T	227	16	20	0	29	0	12

Table 6: Final site sampler residue counts

	Motif Position (0 = Background)						
Nucleotide	**0**	**1**	**2**	**3**	**4**	**5**	**6**
A	.264	.236	.223	.356	.217	.289	.315
C	.273	.276	.210	.210	.210	.329	.223
G	.242	.180	.233	.233	.180	.180	.180
T	.220	.307	.334	.201	.393	.201	.281

Table 7: Final site sampler residue frequencies

If we were to go back to the predictive update/sampling stages with these results, we would sample from the results gathered in table 8.

Sequence	X	A_x	Normalized A_x
TCAGAA	1	1.369	.023
CAGAAC	2	0.631	.011
AGAACC	3	0.941	.016
GAACCA	4	0.922	.016
AACCAG	5	0.380	.006
ACCAGT	6	0.427	.007
CCAGTT	7	0.921	.016
CAGTTA	8	1.591	.027
AGTTAT	9	1.968	.034
GTTATA	10	0.903	.015
TTATAA	11	6.114	.104
TATAAA	12	1.129	.019
ATAAAT	13	2.048	.035
TAAATT	14	1.475	.025
AAATTT	15	2.071	.035
AATTTA	16	1.316	.022
ATTTAT	17	3.004	.051
TTTATC	18	1.150	.020
TTATCA	19	6.622	.113
TATCAT	20	1.152	.020
ATCATT	21	0.994	.017
TCATTT	22	2.873	.049
CATTTC	23	1.048	.018
ATTTCC	24	2.106	.036
TTTCCT	25	2.183	.037
TTCCTT	26	1.421	.024
TCCTTC	27	1.087	.019
CCTTCT	28	1.931	.033
CTTCTC	29	0.810	.014
TTCTCC	30	2.715	.046
TCTCCA	31	1.053	.018
CTCCAC	32	0.829	.014
TCCACT	33	1.029	.018
CCACTC	34	0.615	.010
CACTCC	35	1.161	.020
ACTCCT	36	0.751	.013

Table 8: Weights for sampling step after a near maximal alignment has been found

Comparison of site sampler with the motif sampler

The site sampler and motif sampler follow the same basic Gibbs techniques. The difference is that the motif sampler will allow for the detection of zero or more motif locations in each sequence, whereas the site sampler detects exactly one. Thus, for the initialization step with the motif sampler, a random alignment is made according to a guestimate as to how many motif sites exist in total. An example of an initial alignment is given in figure 3.

T**CAGAAC**CAGTTATAA**ATTTAT**CATTTCCTTCTCCACTCCT
CCCACGCA**GCCGCC**CTCCTCCCCGGTCACTGACTGGTCCTG
TCGACCCTCTGAACCTATCAGGGACCA**CAGTCA**GCCAGGCAAG
AAAACACTTGAGGGAGCAGATAACTGGGCCAACCATGACTC
GGGTGAATGGTACTGCT**GATTAC**AACCTCTGGTGCTGC
AGCCTAGAGTGATGACTCCTATCT**GGGTCC**CCAGCAGGA
GCCTCAGGATCCAGCACACATTATCACAAACTTAGTGTCCA
CATTATCACAAACTTAGTGTCCATCCATCACTGCTGACCCT
TCGGAACAAGGCAAAGGCTATAAAAAAAAT**TAAGCA**GC
GCCCCT**TCCCCA**CACTATCTCAATGCAAATATCTGTCTGAAACGGTTCC
CATGCCCTCAAGTGT**GCAGAT**TGGTCACAGCATTTCAAGG
GATTGGTCACAGCATTTCAAGGGAGAGACCTCATTGTAAG
TCCCCAACTCCC**AACTGA**CCTT**ATCTGT**GGGGGAGGCTTTTGA
CCTTATCTGTGGGGGAGGCTTTTGAAA**AGTAAT**TAGGTTTAGC
A**TTATTT**TCCTTATCA**GAAGCA**GAGAGACAAGCCA**TTTCTC**TTTCCTCCCGGT
AGGCTATAAAAAAAATTAAG**CAGCAG**TATCCTCTTGGGGGCCCCTTC
CCAGCACACACACTTATCCAGTGGTAAATAC**ACATCA**T
TCAAA**TAGGTA**CGGATAAGTAGATATTGAAGTAAGGAT
ACTTGGGGTTCCAGTTTGATAAGAAAAGACTTCCTGTGGA
TGGCCGCAGGAAGGTGGGCC**TGGAAG**ATAACAGCTAGTAGGCTAAGGCCAG
CAACCACAACCTCTGTATCCGGTAGTGGCAGATGGAAA
CTGTATCCGGTAGTGGCAGATG**GAAAGA**GAAACGGTTAGAA
GAAAAAAAATAAATGAAGTCTGCCTATCTC**CGGGCC**AGAGCCCCT
TGCCTTGTCT**GTTGTA**GATAATGAATCTATCCTCCAGTGACT
GGCCAGGCTGATGGGCCT**TATCTC**TTTACCCACCTGGCTGT
CA**ACAGCA**GGTCCTACTATCGCCTCCC**TCTAGT**CTCTG
CCAACCG**TTAATG**CTAGAGTTATCACTTTCTGTTATCA**AGTGGC**TTCAGCTATGCA
GGGAGGGTGGGGCCCCTATCTCTC**CTAGAC**TCTGTG
CTTTGTCACTGGATCT**GATAAG**AAACACCACCCCTGC

Figure 3: Initial Alignment for Motif Sampler (e = 30)

Note that the guestimate does not need to be the exact number of motif positions to be found. This is just a starting number that will evolve within the motif sampler. Using the same data that is used with the site sampler, the maximal alignment using the motif sampler is given in figure 4. With the motif sampler, any given location is sampled into the model based on the ratio of the site being in the model to it being in the background.

TCAGAACCAGTTATAAAT**TTATCA**TTTCCTTCTCCACTCCT
CCCACGCAGCCGCCCTCCTCCCCGGTCACTGACTGGTCCTG
TCGACCCTCTGGAAC**CTATCA**GGGACCACAGTCAGCCAGGCAAG
AAAACACTTGAGGGAGCAGATAACTGGGCCAACCATGACTC
GGGTGAATGGTACTGCTGATTACAACCTCTGGTGCTGC
AGCCTAGAGTGATGACTC**CTATCT**GGGTCCCCAGCAGGA
GCCTCAGGATCCAGCACACA**TTATCA**CAAACTTAGTGTCCA
CA**TTATCA**CAAACTTAGTGTCCATCCATCACTGCTGACCCT
TCGGAACAAGGCAAAGGCTATAAAAAAAATTAAGCAGC
GCCCCTTCCCCACA**CTATCT**CAATGCAAATATCTGTCTGAAACGGTTCC
CATGCCCTCAAGTGTGCAGATTGGTCACAGCATTTCAAGG
GATTGGTCACAGCATTTCAAGGGAGAGACCTCATTGTAAG
TCCCCAACTCCCAACTGACC**TTATCT**GTGGGGGAGGCTTTTGA
CC**TTATCT**GTGGGGGAGGCTTTTGAAAAGTAATTAGGTTTAGC
ATTATTTTCC**TTATCA**GAAGCAGAGAGACAAGCCATTTCTCTTTCCTCCCGGT
AGGCTATAAAAAAAATTAAGCAGCAGTATCCTCTTGGGGGCCCCTTC
CCAGCACACACAC**TTATCC**AGTGGTAAATACACATCAT
TCAAATAGGTACGGATAAGTAGATATTGAAGTAAGGAT
ACTTGGGGTTCCAGTTTGATAAGAAAAGACTTCCTGTGGA
TGGCCGCAGGAAGGTGGGCCTGGAAGATAACAGCTAGTAGGCTAAGGCCAG
CAACCACAACCTCTGTATCCGGTAGTGGCAGATGGAAA
CTGTATCCGGTAGTGGCAGATGGAAAGAGAAACGGTTAGAA
GAAAAAAAATAAATGAAGTCTGC**CTATCT**CCGGGCCAGAGCCCCT
TGCC**TTGTCT**GTTGTAGATAATGAAT**CTATCC**TCCAGTGACT
GGCCAGGCTGATGGGCC**TTATCT**CTTTACCCACCTGGCTGT
CAACAGCAGGTCCTACTATCGCCTCCCTCTAGTCTCTG
CCAACCGTTAATGCTAGAG**TTATCA**CTTTCTG**TTATCA**AGTGGCTTCAGCTATGCA
GGGAGGGTGGGGCCC**CTATCT**CTCCTAGACTCTGTG
CT**TTGTCA**CTGGATCTGATAAGAAACACCACCCCTGC

Figure 2: Final alignment for motif sampler

References

Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC. 1993. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* **262**:208-214.

This paper describes a Gibbs sampling strategy where there is assumed to be a single occurrence of the motif within each sequence. No gaps are allowed within the alignment. The implementation is known in the literature as a site sampler.

Neuwald AF, Liu JS, Lawrence CE. 1995. Gibbs motif sampling: detection of outer membrane repeats. *Protein Science* **4**:1618-1632.

This paper describes a Gibbs sampling strategy where the number of motifs is not known. This is the motif sampler. Examples are presented in the location of theimmunoglobulin fold and hth motifs.

Liu JS, Neuwald AF, Lawrence CE. 1995. Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American Statistical Association* 90, 432:1156-1171.

This paper contains more of the derivations for implementing aBayesian model in the Gibbs sampler. I have not covered any of the details of this paper here, but if further research into the derivations of the various formulas sounds interesting, this should be a good place to start.

Tanner, MA. 1993. Tools for Statistical Inference. Springer-Verlag.