SGN-6156, Lecture 3
Biological sequence analysis

Harri Lähdesmäki, harri.lahdesmaki@tut.fi

(part of the material by Juha Kesseli)

Department of Signal Processing,

Tampere University of Technology

08.04.2008

Alignment with the affine gap penalty

- A standard assumption is to use the affine gap score

$$\gamma(g) = -d - e(g-1)$$

- An $O(nm)$ implementation exists (but with slightly increased memory requirements)
- Three matrices
  - $M(i,j)$: assuming x_i is aligned to y_j
  - $I_x(i,j)$: assuming x_i is aligned to a gap
  - $I_y(i,j)$: assuming y_j is aligned to a gap

- Recursions for global alignment:

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) \\ I_x(i-1,j-1) + s(x_i, y_j) \\ I_y(i-1,j-1) + s(x_i, y_j) \end{cases}$$

$$I_x(i,j) = \max \begin{cases} M(i-1,j) - d \\ I_x(i-1,j) - e \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i,j-1) - d \\ I_y(i,j-1) - e \end{cases}$$

- The above recursions can be expressed in an intuitive way using finite state machines, see Figure 2.9 in (Durbin et al., 1998)
- State variable updated according to the maximum of transition scores

- See Figure 2.10 in (Durbin et al., 1998)

Estimating the substitution matrices

- Previously, we assumed that the model parameters (substitution matrices) are known.
- An intuitive approach would count the frequencies of aligned residue pairs and gaps and sets the parameters to normalized counts. This corresponds to the maximum likelihood method, as shown previously.
- Problems:
  - constructing a representative (random) sample of "true" alignments.
  - evolutionarily distance between aligned sequence pairs can be very different.

PAM matrix

- Point Accepted Mutation (PAM) matrix (Dayhoff matrix) is an older substitution matrix that does not perform as well as BLOSUM for more distantly related sequences.
- The matrix was originally obtained by studying very similar proteins and then extrapolating the results to more divergent sequences.
- One PAM unit refers to the evolutionary time during which one point mutation is accepted for every 100 residues.
- PAM matrices for longer time periods can be obtained by multiplying the original matrix with itself, once for a period twice as long.
- In other words, PAM1 model (say matrix S) can be viewed as a Markov chain corresponding to transition/substitutions probabilities during one time unit.

- Transition/Substitutions probabilities for n time steps are S^n, e.g.

$$P(a|b, t = 2) = \sum_c P(c|b, t = 1)P(a|c, t = 1)$$

- PAM matrices have been updated based on more current information and are still in use.
- Commonly used PAM250 matrix (there are several versions) corresponds to evolutionary time resulting in 20% sequence similarity. Note that unlike with BLOSUM, a PAM matrix with a lower number should be used for closer matches.
- Note that the PAM units are not the same for different families of proteins, since the speed of evolution varies.

Estimating the substitution matrix (continued)

- The Blocks database contains multiply aligned ungapped segments that correspond to the most highly conserved regions of proteins.
- The database is generated automatically based on proteins in InterPro database.

BLOSUM

- BLOSUM (BLOcks SUbstitution Matrix) matrices are based on Blocks.
- E.g. BLOSUM62 uses entries from Blocks that are clustered so that sequences with at least 62% pairwise identity with one of the previous sequences in a cluster are added to it.
  - In estimating the frequencies, pairs of individual sequences are replaced with pairs of clusters, each member in the cluster getting an equal weight in the estimation.
- If we want to find more distant relationships, we should use a scoring matrix constructed with lower sequence identity requirement. The lower the percentage identity required the more distant the sequences to be compared should be for the scoring to be succesful.

BLOSUM in practice

- The BLOSUM matrix A can be obtained as follows:
    1. From all the sequence data in the blocks database, estimate the frequency of each amino acid f_i, $i = 1, \ldots, 20$.
    2. Taking all pairs of sequences in each block, estimate the frequency of different pairs of amino acids, $f_{i,j}$, $i, j = 1, \ldots, 20$.
    3. Compute (and round to nearest integer) the following score:

$$s_{i,j} = \log_2 \frac{f_{i,j}}{f_i f_j}$$

- The scores are normalized by the background frequencies of different amino acids. Thus, scoring with this matrix means comparing the likelihood of the sequences resulting from substitutions with the likelihood of observing the sequences by chance.

Markov chains

- The following is mostly based on Section 3 of (Durbin et al., 1998)
- Discrete time and discrete state Markov chains (and their extensions hidden Markov models (HMM)) are widely used in biological sequence analysis
- Markov chain can be used to model sequences in which the probability of an element depends on the previous element(s) (the context)
- The first order Markov model is defined by transition probabilities

$$a_{st} = P(x_i = t | x_{i-1} = s), \quad s, t \in \mathcal{A}$$

  and these probabilities remain unchanged for all i (homogeneous)
- See figure on page 48 in (Durbin et al., 1998)

- E.g. a sequence $x = (x_1, x_2, \ldots)$ of DNA can be modeled by giving probabilities $p(x_i = a|x_{i-1} = a)$, $p(x_i = a|x_{i-1} = c)$, . . . , independent of i
- Note that $a_{aa}+a_{ac}+a_{ag}+a_{at} = 1$ and sums for different conditioning terms are independent (multinomials)
- Generally, the probability of a sequence $x = (x_1, \ldots, x_L)$ factorizes as

$$P(x) = P(x_1) \prod_{i=2}^{L} P(x_i|x_{i-1}) = P(x_1) \prod_{i=2}^{L} a_{x_{i-1}x_i}$$

- This corresponds to so called evaluation of x
- Given a nucleotide sequence $acgttcg$ we can compute its probability given a first-order Markov model by

$$p(acgttcg) = p(a)a_{ac}a_{cg}a_{gt}a_{tt}a_{tc}a_{cg}$$

- In a zeroth-order Markov chain the elements are independent. This is sometimes used as the simplest “background model.”
- The sum of xs over all sequences of length L equals 1
- Higher-order models take a longer preceding sequence into account
- The nth order Markov model is defined by transition probabilities

$$P(x_i|x_{i-1},\ldots,x_{i-n})$$

- E.g. in a second-order model we would model the sequence $acgttcg$ as

$$p(acgttcg) = p(ac)p(g|ac)p(t|cg)p(t|gt)p(c|tt)p(g|tc).$$

- Markov chains can also be inhomogeneous

Estimating model parameters

- Parameters can be estimated from a training data that is assumed to represent a feature/phenomenon of interest.
- The conditional probabilities of a Markov chain can be estimated simply based on counts from the data, i.e.

$$a_{st} = \frac{c_{st}}{\sum_{t'} c_{st'}},$$

  where c_{st} is the number of times s is followed by t

- As discussed before, these are the maximum-likelihood (ML) parameters (well-known benefits of ML estimation)
- Note that if higher-order models are used the number of probabilities to be estimated increases significantly.

- In addition to taking more time, more data is required as well — estimating probabilities from low counts will not produce a reliable model.
- In small sample setting Bayesian approach can be more appropriate than ML.

A simple Markov chain method

- At its simplest we can try to find, e.g., the protein coding regions in a prokaryote genome as follows:
  1. Learn the probabilities for the Markov model θ_1 from data containing known protein-coding regions (e.g. ML method)
  2. Learn the probabilities for the Markov model θ_2 from known non-coding regions (e.g. ML method)
  3. For each sequence stretch x of interest:
     - Compute the probability of x given model θ_1, $p_1 = P(x|\theta_1)$
     - Compute the probability of x given model θ_2, $p_2 = P(x|\theta_2)$
     - If $p_1/p_2 > t$, a threshold selected e.g. by finding an optimal value using training data, consider the given sequence a potential coding region
     - Alternatively, a Bayesian model comparison: $P(\theta_1|x)$ vs. $P(\theta_2|x)$

4. Resolve overlaps, if any

A Markov chain example

- CpG islands, CG-rich regions in gene promoters, are clues to the location of genes
- Example from (Durbin et al., 1998): classify a set of sequences into CpG-islands and “others”
- Learn model parameters for the two first-order Markov models
- See table on page 50 in (Durbin et al., 1998)
- Compute the likelihood ratios as above, see Figure 3.2 in (Durbin et al., 1998)

Another Markov chain example

- Finding prokaryotic genes
- Use the first and “higher-order” models in the same way as above
- See Figure 3.11 and 3.12 in (Durbin et al., 1998)

Beginning and end of a Markov chain

- Additional 'begin' and 'end' states, $\mathcal{B}$ and $\mathcal{E}$:
  - The probability of starting the chain with symbol s is $P(x_1 = s) = a_{\mathcal{B}s}$
  - The probability of ending the chain with symbol t is $P(x_L = t) = a_{t\mathcal{E}}$
- See Figure 3.1 in (Durbin et al., 1998)
- End state allows modeling sequences of different lengths

Testing for significance of the model

- Often the correct order of the Markov model is not known.
- For example, to check if a Markov chain of order 1 should be used instead of order 0 we test the null hypothesis
  - H_0: The probability of a nucleotide in position $i+1$ is independent of the nucleotide in position i.

  against the alternative H_1
- We can compile a table from the counts we have observed based on sequence data, e.g.:

		Nucleotide in position $i+1$				
		a	g	c	t	Total
Nucleotide in position i	a	70	61	79	60	270
	g	48	60	60	55	223
	c	79	45	51	68	243
	t	72	57	53	81	263
	Total	269	223	243	264	

- Denote the elements of the above matrix as x_{ij} and the row and column sums as $n_{xj} = \sum_i x_{ij}$ and $n_{ix} = \sum_j x_{ij}$.
- If the null hypothesis were true and rows and columns were independent, based on the marginals we would expect to have a table with values $E_{ij} = n_{ix} n_{xj} / n$.
- In that case, the square error is asymptotically distributed as

$$Y = \sum_{i,j} \frac{(x_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2(9).$$

- For the above table we find $Y \approx 19.585$ and the corresponding value from the cumulative distribution function of $\chi^2(9)$ as $0.9793 > 0.95$, so that with p-value 0.02 the dependence is significant.
- For sequence alignment, a zeroth-order model is often used as a background model, so that independent nucleotides actually make alignment work better.

Hidden Markov models (HMM)

- Consider the previous Markov model examples. Given long sequences, how does one locate CpG islands (or protein coding parts) in them without classifying each entire sequence as cpG or non-CpG?
- For example, combine CpG and non-CpG Markov models and allow a small transition probability between them (see Figure 3.3 in (Durbin et al., 1998))
- Distinguish sequence of states π (path) and sequence of symbols x
- Unobserved path π follows a Markov chain with transition probabilities

$$a_{kl} = P(\pi_i = l | \pi_{i-1} = k)$$

- Symbols are generated by emission probabilities: the probability of emitting b given that the state is k

$$e_k(b) = P(x_i = b|\pi_i = k)$$

- The above transition and emission probabilities do not depend on i (homogeneous)
- A generative model (simulation):
  - Choose the first state π_1 according to the probabilities $a_{\mathcal{B}i}$
  - Emit the first observation x_1 according to e_{π_1}
  - Choose the next state π_2 according to a_{1i}
  - Emit the second observation x_2 according to e_{π_2}
  - etc.
- See example on page 54 in (Durbin et al., 1998)

- The probability of a path and symbols

$$P(\pi, x) = a_{\mathcal{B}\pi_1} e_{\pi_1}(x_1) \left(\prod_{i=2}^{L} a_{\pi_{i-1}\pi_i} e_{\pi_i}(x_i) \right) a_{\pi_L \mathcal{E}}$$

Path estimation in HMM

- Often the unknown path π is of interest
- Thus, a path needs to be estimated (decoding)
- Find a path π that has the highest probability

$$\pi^* = \arg\max_{\pi} P(\pi, x)$$

- Viterbi is a dynamic programming algorithm for finding π^*
- Let $v_k(i)$ denote the probability of the most probable path ending in state k with observation i
- Probability $v_l(i+1)$ for all l can be found as

$$v_l(i+1) = \left[\max_k (v_k(i) a_{kl})\right] e_l(x_{i+1})$$

- The actual path π^* can be obtained by backtracking

Viterbi algorithm for HMM decoding

- Initialization: $i = 0$, $v_0(0) = 1$, $v_k(0) = 0$ for $k > 0$
- Recursion: $i = 1, \ldots, L$, for all l

$$\begin{aligned} v_l(i) &= \left[\max_k (v_k(i-1)a_{kl})\right] e_l(i) \\ \mathrm{ptr}_i(l) &= \arg\max_k (v_k(i-1)a_{kl}) \end{aligned}$$

- Termination:

$$\begin{aligned} P(\pi^*, x) &= \max_k (v_k(L)a_{l0}) \\ \pi_L^* &= \arg\max_k (v_k(L)a_{k0}) \end{aligned}$$

- Backtracking: $i = L, \ldots, 1$

$$\pi_{i-1}^* = \mathrm{ptr}_i(\pi_i^*)$$

- See Figure 3.5 in (Durbin et al., 1998)

References

- R. Durbin, S. R. Eddy, A. Krogh and G. Mitchison (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press.