SITE IDENTIFICATION

Problems

1. How to identify functional regions from sequence data?

2. How to find genes?

3. Sites?

   (a) Instances of known sites?

   (b) Instances of unknown sites?

Site A short sequence that contains some signal recognized by some enzyme

1. Origins of replication

2. Transcription start and stop sites

3. Promoters, or transcription factor binding sites

4. Introns splice sites

5. . . .

Challenge Instances of a single site will generally not be identical, but will instead vary slightly

We'll start with the problem of *finding instances of known sites* and finish with the problem of *finding instances of unknown sites*

Representation of Known Sites

Suppose we have a large sample $\mathcal{A}$ of length n sites, and a large sample $\mathcal{B}$ of length n non-sites. Given a new sequence $s = s_1 s_2 \ldots s_n$. Is s more likely to be a site or a non-site? If we have an efficient method to do this, then we can screen an entire genome, testing every length n substring, and generate a complete list of candidates sites.

Example of site: the *Cyclic AMP receptor protein* (or CRP, a transcription factor in E. Coli.) binds to sites of length 22. Position 3–9 from 23 CRP binding sites are shown below

```
TTGTGGC
TTTTGAT
AAGTGTC
ATTTGCA
CTGTGAG
ATGCAAA
GTGTTAA
ATTTGAA
TTGTGAT
ATTTATT
ACGTGAT
ATGTGAG
TTGTGAG
CTGTAAC
CTGTGAA
TTGTGAC
GCCTGAC
TTGTGAT
TTGTGAT
GTGTGAA
CTGTGAC
ATGAGAC
TTGTGAG
```

Profile for CRP Binding Sites

A	.35	.043	0	.043	.13	.83	.26
C	.17	.087	.043	.043	0	.043	.3
G	.13	0	.78	0	.83	.043	.17
T	.35	.87	.17	.91	.043	.087	.26

The *signal* is not easy to detect at first glance. Notice, though, that T predominates in the 2nd and 4th column, and G in the 3rd and 5th column, for instance. The goal is to capture the most relevant information from these 23 sites in a concise form.

Identifying Sites By Probabilities

Profile A of $\mathcal{A}$ is a $c \times n$ matrix where $A_{r,j}$ is the fraction of sequences in $\mathcal{A}$ that have residue r in position j, where c is the number of distinct residues.

In terms of probabilities: let $t = t_1 t_2 \cdots t_n$ be a random uniform sequence from $\mathcal{A}$, then
$A_{r,j} = \mathtt{pr}(t_j = r \mid t \in \mathcal{A})$.
Ex: $A_{T,2} = \mathtt{pr}(t_2 = T \mid t \in \mathcal{A})$

Independence assumption: Two probabilistic events E and F are said to be *independent* if the probability that they both occur is the product of their individual probabilities, that is $\mathtt{pr}(E \bigcap F) = \mathtt{pr}(E) \cdot \mathtt{pr}(F)$.
Residues at any two positions are uncorrelated

Under the independence assumption, the probability that a randomly chosen site has a specified sequence $r_1 r_2 \cdots r_n$ is: $\mathtt{pr}(t = t_1 t_2 \cdots t_n \mid t \text{ is a site}) = \prod_{j=1}^{n} A_{r_j,j}$

Example: prob. that a random CRP site is `TTGTGAC` is $\mathtt{pr}(t = \mathtt{TTGTGAC} \mid t \text{ is a site}) = (.35)(.87)(.78)(.91)(.83)(.83)(.3) = 0.045$

If we form a profile B from the sample $\mathcal{B}$ of non-sites we can then test whether a given sequence s is *more likely* to be a site or a non-site

Identifying Sites By Probabilities

Likelihood ratio: Given sequence $s = s_1 s_2 \cdots s_n$, the *likelihood ratio*, denoted by $LR(A, B, s)$ is defined to be

$$\frac{\mathtt{pr}(t = s \mid t \text{ is a site})}{\mathtt{pr}(t = s \mid t \text{ is a non-site})} = \frac{\prod_{j=1}^{n} A_{s_j,j}}{\prod_{j=1}^{n} B_{s_j,j}} = \prod_{j=1}^{n} \frac{A_{s_j,j}}{B_{s_j,j}}$$

Example: Let $\mathcal{B} = \{A, C, G, T\}^7$, the set of all length seven sequences. The corresponding profile B has $B_{r,j} = 0.25$ for all r and j. Then for $s = \mathtt{TTGTGAC}$,

$$LR(A, B, s) = \frac{\prod_{j=1}^{n} A_{s_j,j}}{\prod_{j=1}^{n} B_{s_j,j}} = \frac{0.045}{(0.25)^7} = 732$$

Testing a sequence: s is *more likely* a

1. site: if $LR(A, B, s) \geq L$
2. non-site: if $LR(A, B, s) < L$

Where L is a pre-specified constant cutoff

Log likelihood ratio:

$$LLR(A, B, s) = \log_2 \prod_{j=1}^{n} \frac{A_{s_j,j}}{B_{s_j,j}} = \sum_{j=1}^{n} \log_2 \frac{A_{s_j,j}}{B_{s_j,j}}$$

s is more likely a site if $LLR(A, B, s) \geq \log_2 L$

Weight Matrix

Weight matrix: a $c \times n$ matrix W that assigns a score to each $s = s_1 s_2 \cdots s_n$ according to formula $\sum_{j=1}^{n} W_{s_j,j}$

In a log likelihood weight matrix, we have $W_{i,j} = \log_2 \frac{A_{r,j}}{B_{r,j}}$.

In order to compute $LLR(A, B, s)$, we only need to add the corresponding scores from W : $LLR(A, B, s) = \sum_{j=1}^{n} W_{s_j,j}$. Example

A	.48	-2.5	$-\infty$	-2.5	$-.94$	1.7	.061
C	$-.52$	-1.5	-2.5	-2.5	$-\infty$	-2.5	.28
G	$-.94$	$-\infty$	1.6	$-\infty$	1.7	-2.5	$-.52$
T	.48	1.8	$-.52$	1.9	-2.5	-1.5	.061

Log likelihood weight matrix for CRP binding sites

We often take $B_{r,j}$to be the *background distribution* of residue r: $B_{r,j}$ is the frequency of r within the entire genome

Example: given 8 start sites `ATG`,`ATG`,`ATG`,`ATG`,`ATG`,`GTG`,`GTG`, and `TTG`, we assume a uniform background distribution $B_{r,j} = 0.25$ then

A	.625	0	0
C	0	0	0
G	.25	0	1
T	.125	1	0

Profile

A	1.32	$-\infty$	$-\infty$
C	$-\infty$	$-\infty$	$-\infty$
G	0	$-\infty$	2
T	-1	2	$-\infty$

LLR weight matrix

	.701	2	2

Positional relative entropy

Relative Entropy

How informative is the LLR test for distinguishing between sites and non-sites? For equal distributions of $\mathcal{A}$ and $\mathcal{B}$ every entry in matrix is 0, thus uninformative

Definitions

1. *Sample space* S: set of all possible values of a random variables

2. *Probability distribution* P for a sample space S assigns a probability $P(s)$ to every $s \in S$ satisfying

   (a) $0 \leq P(s) \leq 1$

   (b) $\sum_{s \in S} P(s) = 1$

3. *Relative entropy* (or information content) of P with respect to $Q$is

$$D_b(P\|Q) = \sum_{s \in S} P(s) \log_b \frac{P(s)}{Q(s)}$$

   where P, Q are probability distributions on S

4. By convention:

$$P(s) \log_b \frac{P(s)}{Q(s)} = 0$$

   whenever $P(s) = 0$

5. *Expected value* of function $f(s)$ with respect to probability distribution P on S is

$$E(f(s)) = \sum_{s \in S} P(s) f(s)$$

Relative Entropy

Relative entropy measures how different the distributions of P, Q are. For instance, when they have same distributions then $D_b(P\|Q) = 0$

To distinguish between sites and non-sites: $D_b(P\|Q)$ must be large

With independence assumption:

$$D_b(P\|Q) = \sum_{j=1}^{n} D_b(P_j\|Q_j)$$

where P_j and Q_j are the distributions imposed by P, Q at column j

When $b = 2$ the relative entropy is measured in bits

Example: Previous tables shows the relative entropies $D_2(P_j\|Q_j)$ for each residue position j separately

- At position 2, residues A, C, G do not contribute to the relative entropy (see first table). Residue T contributes $1 \cdot W_{T,2} = 2$ (see first 2 tables). Hence $D_2(P_2\|Q_2) = 2$. This means that there are 2 bits of information in position 2. If residues are coded as A=00, C=01, G=10 and T=11, then only 2 bits (11) are necessary to encode the fact that this residue is always T
- Position 3 has the same relative entropy of 2
- For position 1, the relative entropy is 0.7 so there are 0.7 bits of information, indicating that column 1 (of first table) is more similar to the background distribution than columns 2 and 3 are
- The total relative of all 3 positions is 4.7

Effect of Non-Uniform Background Distribution

Consider the same 8 sites but change the background distribution to $B_{A,j} = B_{T,j} = 0.375$, $B_{C,j} = B_{G,j} = 0.125$. The new weight matrix and relative entropy is given below

A	0.737	$-\infty$	$-\infty$
C	$-\infty$	$-\infty$	$-\infty$
G	1	$-\infty$	3
T	-1.58	1.42	$-\infty$

LLR weight matrix

	.512	1.42	3

Positional relative entropy

.12	1.3	1.1	1.5	1.2	1.1		.027	

Positional relative entropy for CRP binding sites

The relative entropy of each column has changed: last 2 columns have different entropy

The site distribution in position 2 is now more similar to the background distribution than the site distribution in position 3 is, since G is rarer in the background distribution. Thus the relative entropy of position 3 is greater than that of position 2

An interpretation of $D_2(P_3\|Q_3) = 3$ is that the residue G is $2^3 = 8$ times more likely to occur in the third position of a site than a non-site.

The total relative entropy is 4.93

Non-negativity of relative entropy: For any probability distributions P and Q over a sample space S, $D_b(P\|Q) \geq 0$, with equality if and only if P and Q are identical

Finding Instances of Unknown Sites

We are not given a sample $\mathcal{A}$ of known sites, but we want to find sequences that are significantly similar to each other,without *a priori* knowledge of what those sequences look like

Problem: Given a set of sequences, find instances of a short site that occur more often than you would expect by chance, with no *a priori* knowledge about the site

Given a collection of k such instances, this induces a profile A. We can also compute a profile B from the background distribution. From A and B we can compute $D_2(A\|B)$ and use that as a measure of how good the collection is. The goal is to find the collection that maximizes $D_2(A\|B)$.

Relative entropy site selection problem (RESSP): Take as input k sequences and an integer n, and output one length n substring from each input sequence, such that the resulting relative entropy is maximized.

The RESSP is NP-complete

Finding Instances of Unknown Sites

As example of finding instances of unknown sites, consider the genes involved in digestion in yeast. It is likely that many of these genes have some transcription factors in common and therefore similarities in their promoters regions. Applying the RESSP to 1000bp DNA sequences upstream of known digestion genes may well yield some of these promoters.

As defined, RESSP limits its solution to contain exactly one site per input sequence, which may not be realistic in all applications. In some applications, there may be zero or many sites in some of the input sequences.

Effects on relative entropy of increasing the number of sites or length of each site

- Increasing the number of site will not increase the relative entropy, which is a function only of the fraction $P(s)$ of sites containing each residue s, and not the absolute number of such sites. The relative entropy measures the degree of conservation

- Increasing the length n of each site *does* increase the relative entropy, as it is additive and always non-negative. If comparing relative entropies of different length sites is important, one may normalize by dividing by the length n of the site or, alternatively, subtracting the expected relative entropy from each position.

Greedy Algorithm

The algorithm picks the locally best choice at each step without concern for the impact on future choices. In most applications, the greedy method will result in solutions that are far from optimal, for some input instances. However, it does work efficiently, and may produce good solutions on many inputs instances

The user specifies a maximum number d of profiles to retain at each step. profiles with lower relative entropy scores than the top d will be discarded; this is precisely the greedy aspect of the algorithm

Algorithm {Assumes single-site per sequence}
Input: sequences $s_1, s_2, \ldots, s_k$, and n, d, and the background distribution

1. Create a singleton set for each possible length n substring of each of the k input sequences

2. For each set S retained so far, add each possible length n substring from an input sequence s_i not yet represented in S. Compute the profile and relative entropy with respect to the background for each new set. Retain the d sets with the highest relative entropy

3. Repeat step 2 until each set has k members

Pruning the number of sets to d is crucial, in order to avoid the exponentially many possible sets. The greedy nature of this pruning biases the selection from the remaining input sequences. High scoring profiles chosen from the first few sequences may not be well represented in the remaining sequences, whereas medium scoring profiles may be well represented in most of the k sequences, and thus would have yielded superior scores

Gibbs Sampler

Idea: Start with a complete set of k substrings, from which we iteratively remove at random, and then add a new one at random with probability proportional to its score, hopefully resulting in an improved score.

Algorithm {Assumes single-site per sequence}

Input: sequences $s_1, s_2, \ldots, s_k$, n, and the background distribution

Initialize set T to contain substrings $t_1, t_2, \ldots, t_k$, where t_i is a substring of s_i chosen randomly and uniformly. Then perform a series of iterations, each of which consists of the following steps:

1. Choose i randomly and uniformly from $\{1, 2, \ldots, k\}$ and remove t_i from T

2. For every j in $\{1, 2, \ldots, |s_i| - n + 1\}$:

   (a) Let t_{ij} be the length n substring of s_i that starts at position j.

   (b) Compute D_j, the relative entropy of $T \bigcup \{t_{ij}\}$ with respect to the background

   (c) Let $P_j = \frac{D_j}{\sum_h D_h}$

3. Randomly choose t_i to be t_{ij} with probability P_j, and add t_i to T

We iterate until a stopping condition is met, either a fixed number of iterations or relative stability of the scores, and return the best solution set T seen in all iterations