Pattern Discovery in Biosequences

ISMB 2002 tutorial (Appendix)

Stefano Lonardi

University of California, Riverside

Index

- Periodicity of Strings
- DNA micro-arrays
- Sequence alignment
- Expectation Maximization
- Megaprior heuristics for MEME

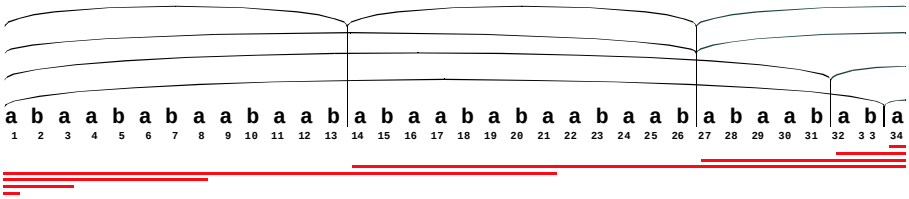
Periodicity of Strings

Periods of a string

- Definition: a string y has *period* w, if y is a non-empty prefix of w^k for some integer $k=1$

- Definition: The period y of y is called *trivial* period
- Definition: the set of period lengths of y is
  $P(y)=\{ |w|: w \text{ is a period of } y, w \neq y\}$

Example



P(y) = {13,26,31,33}

Borders of a string

- Definition: a *border w* of *y* is any nonempty string which is both a prefix and a suffix of *y*
- Definition: the border *y* of *y* is called the *trivial* border
- Fact: a string *y* has a period of length $d<m$ iff it has a non-trivial border of length $m-d$

Finding Borders/Periods

- Borders can be found using the *failure function* of the string as done, e.g., in the preprocessing step of the classical linear time string search algorithms (Knuth, Morris, Pratt)
- Borders can be computed in $O(|y|)$, and so do periods

DNA micro-arrays

Gene expression

- Gene expression does depend on “space location” and “time location”
  - Cells from different tissues produce different proteins
  - Certain genes are expressed only during development or in response to changes to environment, while others are always active (*housekeeping* genes)
  - …

Comparative hybridization

- Comparative hybridization can reveal genes which are preferentially expressed
  - in specific tissues
  - during specific phases of cell cycle (e.g., mitosis, sporulation, death)
  - during specific changes in the environment (e.g., cold/heat shock, nutrient availability, …)
  - in the context of heterogeneous diseases (e.g., certain types of cancer, diabetes, …)

DNA microarrays

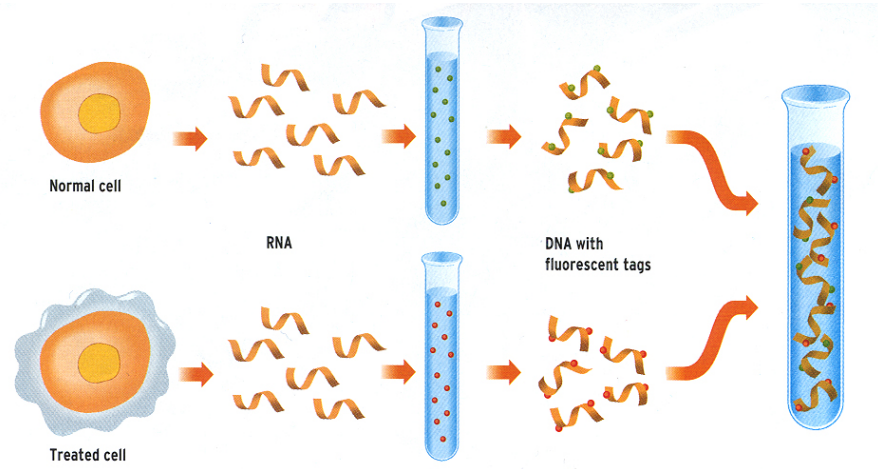
- Monitor the activity of several thousand genes *simultaneously*
- They exploit, in a clever way, the property of DNA to hybridize
- DNA “chips” with probes in the order of 10,000-100,000 are common nowadays
- Perlegen, a spin-off of Affymetrix, is building chips with 60 millions probes to discover SNPs in human genome

DNA microarrays

- They “measure” the amount of mRNA in the cell
- However, we cannot measure directly the mRNA because it is quickly degraded by RNA-digesting enzymes
- We use reverse transcription to get cDNA out of the mRNA
- The assumption is that amount of cDNA will be proportional to the mRNA

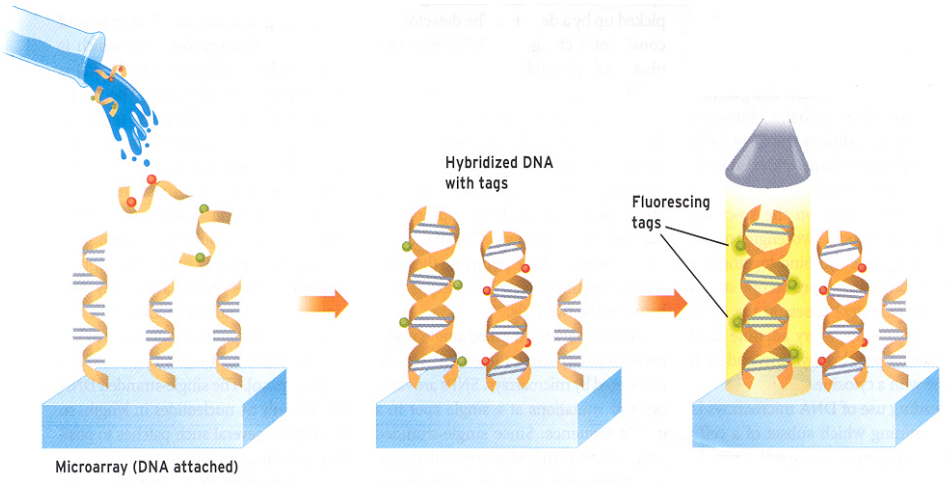
DNA microarrays

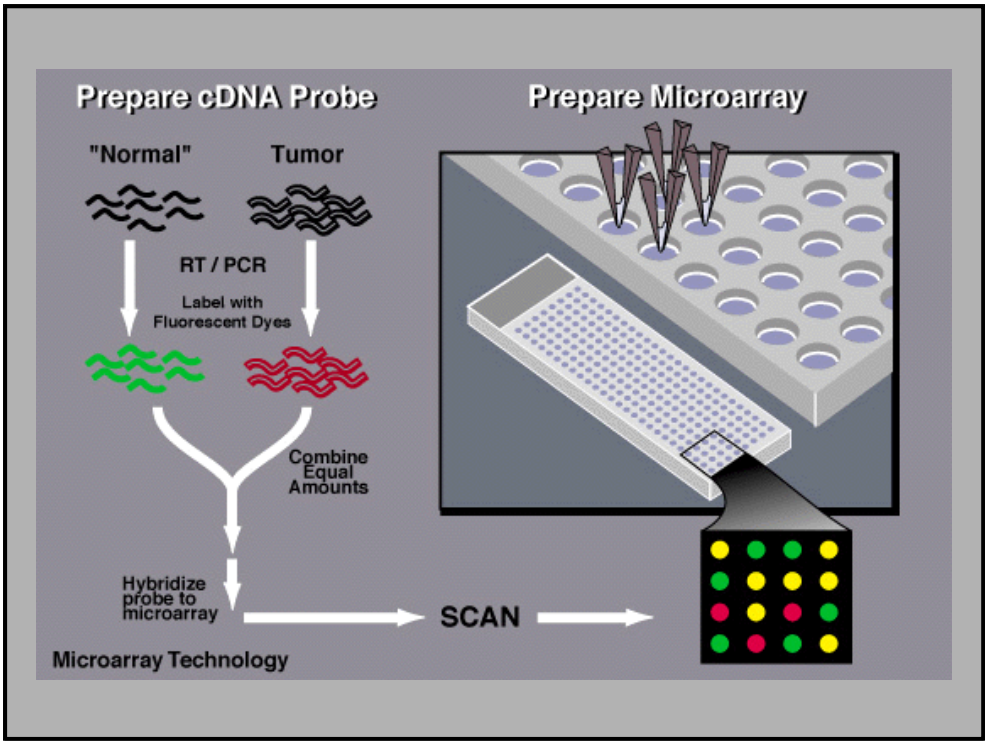
- cDNA is labeled using fluorescent dyes
- The fluorescent dyes can be detected only if stimulated by a specific frequency of light by a laser
- The number of fluorescent dyes molecules which label each cDNA depends on the cDNA length and its composition

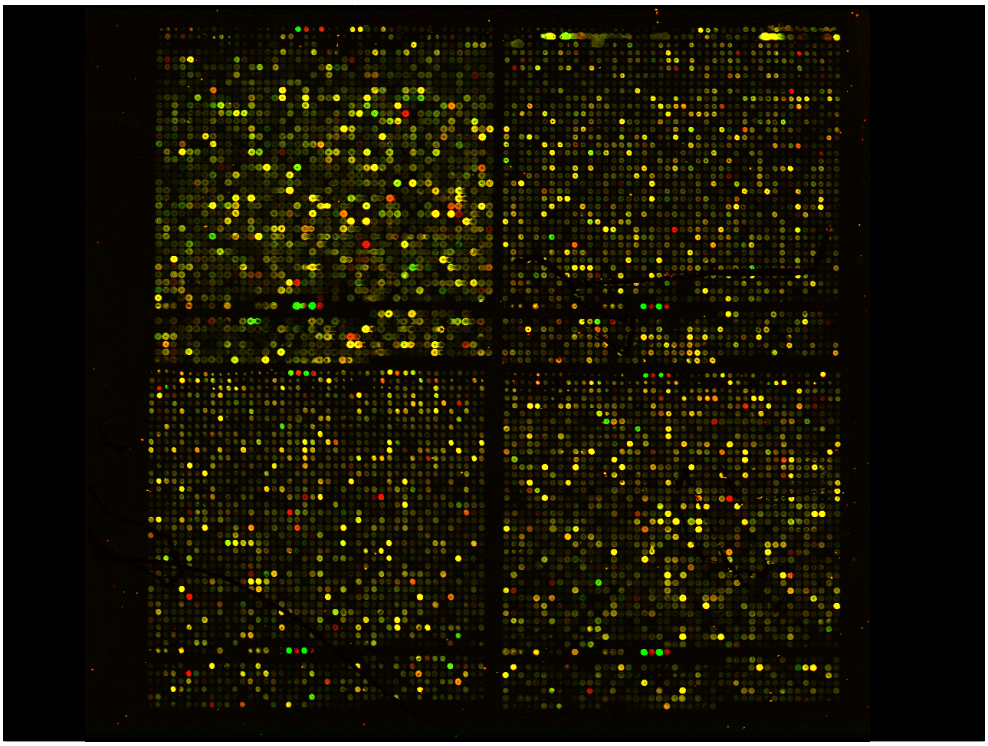


DNA microarrays

- The array holds thousands of spots each containing a different DNA sequence
- If the cDNA happens to be complementary of the DNA of a given spot, that cDNA will hybridize, and will be detected by its fluorescence




Prepare cDNA Probe
Prepare Microarray
"Normal"
Tumor
RT / PCR
Label with
Fluorescent Dyes
Combine
Equal
Amounts
Hybridize
probe to
microarray
SCAN
Microarray Technology



Sequence alignment

Global alignment

- Clearly there are many other possible alignments
- Which one is *better?*
- We assign a *score* to each
  - *match (e.g., 2)*
  - *insertion/deletion (e.g., -1)*
  - *substitution (e.g., -2)*
- Both previous alignments scored
  *4*2+3*(-1)+1*(-2)=3 4*2+1*(-1)+2*(-2)=3*

Scoring function

- Given two symbols *a*, *b* in $S \cup \{-\}$ we define $\sigma(a,b)$ the score of aligning *a* and *b*, where *a* and *b* can not be both "**-**"

- In the previous example
  $\sigma(a,b) = +2$ *if* $a = b$
  $\sigma(a,b) = -2$ *if* $a \neq b$
  $\sigma(-,a) = -1$
  $\sigma(a,-) = -1$

Global alignment

- Definition: Given two strings *w* and *y*, an *alignment* is a mapping of *w*,*y* into strings *w'*,*y'* that may contain spaces, where $|w'| = |y'|$ and the removal of the spaces from *w'* and *y'* returns *w* and *y*
- Definition: The *value* of an alignment *(w',y')* is

$$\sum_{i=1}^{|w'|} s\left(w'_{[i]}, y'_{[i]}\right)$$

Global alignment

- Definition: A *optimal global alignment* of w and y is one that achieves maximum score
- How to find it?
- How about checking all possible alignments?

Checking all alignments

$|w| = |y| = m$

for all i, $0 \le i \le m$ do

  for all subsequences A of w with $|A| = i$ do

    for all subsequences B of y with $|B| = i$ do

      form an alignment that matches $A_{[j]}$ with $B_{[j]}$

      $\forall\ 1 \le j \le i$, and matches all others with spaces

Example

- Given w=**ATCTG**

  y=**CATGA** *(m=5)*
- Suppose *i=2*
- Suppose we choose A=**CG** B=**CT**
- We are considering the score of the following alignment (length is *2m-i=8*)

  ATCT-G--

  --C-ATGA

Time complexity

A string of length *m* has $\binom{m}{i}$ subsequences of length *i*.

Thus, there are $\binom{m}{i}^2$ pairs of subsequences, each of length *i*. The length of each alignment is $2m$-1.

The total number of operations is at least

$$\sum_{i=0}^{m}\binom{m}{i}^2(2m-1) \geq m\sum_{i=0}^{m}\binom{m}{i}^2 = m\binom{2m}{i} > 2^{2m}$$

Checking all alignments

- The previous algorithms runs in $O(2^{2m})$ time
- How bad is it?
- Choose $m=1,000$ and try to wait your computer to run $2^{2,000}$ operations!

Needleman & Wunsch, 70

- The first algorithm to solve efficiently the alignment of two sequences
- Based on *dynamic programming*
- Runs in $O(m^2)$ time
- Uses $O(m^2)$ space

Alignment by dyn. programming

- Let *w* and *y* be two strings, $|w|=n$, $|y|=m$
- Define *V(i,j)* as the value of the alignment of the strings $w_{[1..i]}$ with $y_{[1..j]}$
- The idea is to compute *V(i,j)* for all values of $0 \leqslant i \leqslant n$ and $0 \leqslant j \leqslant m$
- In order to do that, we establish a recurrence relation between *V(i,j)* and *V(i-1,j), V(i,j-1),V(i-1,j-1)*

Alignment by dyn. programming

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + s\,(w_{[i]}, y_{[j]}) \\ V(i-1,j) + s\,(w_{[i]}, "-") \\ V(i,j-1) + s\,("-", y_{[j]}) \end{cases}$$

$$V(0,0) = 0$$

$$V(i,0) = V(i-1,0) + s\,(w_{[i]}, "-")$$

$$V(0,j) = V(0,j-1) + s\,("-", y_{[j]})$$

Example

$s(a,a) = +1$ [match]

$s(a,b) = -1,\ \text{if } a \neq b$ [substitution]

$s(a,"-") = -2$ [deletion]

$s("-",a) = -2$ [insertion]

$$V(i,j) = \max \begin{cases} V(i-1,j-1) + s(w_{[i]}, y_{[j]}) \\ V(i-1,j) + s(w_{[i]},"-") \\ V(i,j-1) + s("-", y_{[j]}) \end{cases}$$

$V(0,0) = 0$

$V(i,0) = V(i-1,0) + s(w_{[i]},"-")$

$V(0,j) = V(0,j-1) + s("-", y_{[j]})$

		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

Example

AG-C
AAAC

		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

Example

A-GC
AAAC

		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

Example

-AGC
AAAC

		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

Variations

- Local alignment [Smith, Waterman 81]
- Multiple sequence alignment (local or global)
- Theorem [Wang, Jiang 94]: the optimal sum-of-pairs alignment problem is *NP*-complete

Expectation Maximization

Expectation maximization

The goal of EM is to find the model that maximizes the (log) likelihood

$$L(q) = \log P(x \mid q) = \log \sum_y P(x, y \mid q).$$

Suppose our current estimated of the parameters is q^t.
We want to know what happens to L when we move to q.

$$L(q) - L(q^t) = \log \frac{\sum_y P(x, y \mid q)}{\sum_y P(x, y \mid q^t)} = \log \frac{\sum_y P(x \mid y, q) P(y \mid q)}{\sum_y P(x \mid y, q^t) P(y \mid q^t)}$$

Expectation maximization

After some (complex) algebraic manipulations one finally gets

$$L(q) - L(q^t) = Q(q \mid q^t) - Q(q^t \mid q^t) + \sum_y P(y \mid x, q^t) \log \frac{P(y \mid x, q^t)}{P(y \mid x, q)}$$

where $Q(q \mid q^t) \equiv \sum_y P(y \mid x, q^t) \log P(x, y \mid q).$

Convergence

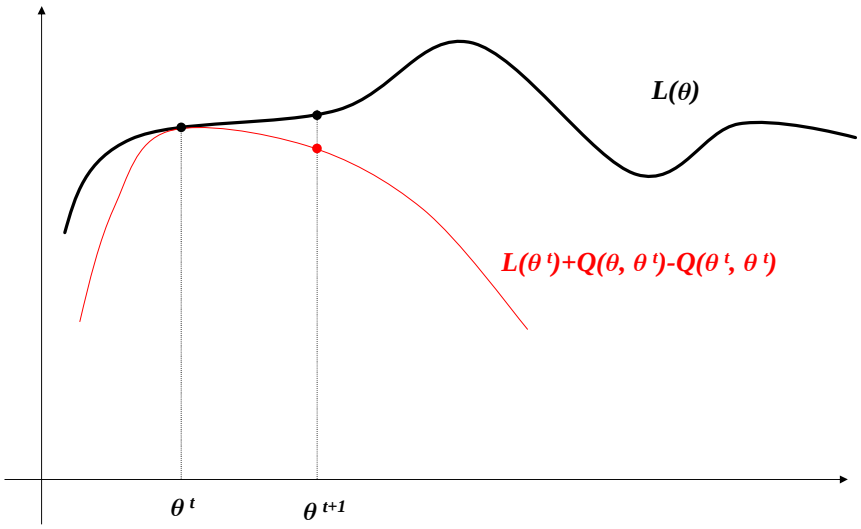
The last term is $H\left(P(y \mid x,q^{t}) \,\|\, P(y \mid x,q)\right)$ which is always non-negative, and therefore

$$L(q) - L(q^{t}) \geq Q(q \mid q^{t}) - Q(q^{t} \mid q^{t})$$

with equality iff $P(y \mid x,q^{t}) = P(y \mid x,q^{t+1})$.

Choosing $q^{t+1} = \arg\max_{q} Q(q \mid q^{t})$ will always make the difference positive and thus the likelihood of the new model q^{t+1} larger than the likelihood of q^{t}.

A step of EM



Expectation maximization

EM iterates 1), 2) until convergence

1) E-Step: compute the $Q(\theta \mid \theta^{t})$ function with respect to the current parameters θ^{t}

2) M-Step: choose $\theta^{t+1}=\text{argmax}_{\theta} Q(\theta \mid \theta^{t})$

Expectation maximization

- The likelihood increases at each step, so the procedure will always reach a maximum asymptotically
- It has been proved that the number of iterations to convergence is *linear* in the input size
- Each step, however, require quadratic time in the size of the input

Expectation maximization

- More importantly, EM can get stuck (easily) in local maxima
- Standard techniques in combinatorial optimization can be used to alleviate this problem

EM for pattern discovery

- The first attempt to use EM for pattern discovery has been proposed by Lawrence and Reilly [Proteins, 1990]
- Input: multisequence $\{x_1,x_2,\ldots,x_k\}$
  pattern length m
- Output: a matrix profile $q_{i,b}$, $b \in \Sigma$, $1 \leqslant i \leqslant m$, and positions s_j, $1 \leqslant j \leqslant k$, of the profile

EM for pattern discovery

- Assumption: there is exactly one occurrence of the profile in each sequence
- The missing information in this case are the positions s_j of the motif in $\{x_1,x_2,\ldots,x_k\}$ (in fact, *if* we knew the positions, the problem of finding the profile would be trivial)

Lawrence-Reilly EM

The objective is to maximize the following log likelihood

$$L(q) = k\sum_{i=1}^{m}\sum_{b\in\Sigma} f^{i}(b)\log(q_{b,i}) + k(n-m)\sum_{b\in\Sigma} f^{0}(b)\log(q_{b,0})$$

where $q_{b,0}$ is the unknown distribution outside the site,

$q_{b,i}$ is the unknown distribution inside the site (profile),

$f^{0}(b)$ is the observed count of b outside the site,

$f^{i}(b)$ is the observed count of b in the site at position i

Lawrence-Reilly EM

The value of q that maximizes the log likelihood L is

$$q_{i,b} = f^{i}(b)/k$$

$$q_{0,b} = f^{0}(b)/(k(n-m))$$

which corresponds to idea of computing the profile by counting the symbols column-by-column

Lawrence-Reilly EM

- E-step: use the current parameters $q^{(t)}$ to compute

  P(observing x_i|profile starts at position s in x_i)

  for all $1 \le i \le k,\ 1 \le s \le |x_i| - m + 1,$ and then

  $r_{i,s} = P$(profile starts at position s in x_i) using Bayes

  for all $1 \le i \le k,\ 1 \le s \le |x_i| - m + 1.$

  Align the profile at each position (i, s) and for each column $1 \le j \le m,$ accumulate in the $\hat{q}_{x_{i,[s+j-1]},j}$ the contributions of $r_{i,s+j-1}$. At the end, $\hat{q}$ contains the expected count of each symbol in each position of the profile.

Lawrence-Reilly EM

- M-step: use the expected count $\hat{q}$ of each symbol in each position to compute the ML (re)estimate of the parameters

$$q_{b,i}^{(t+1)} = \frac{\hat{q}_{b,i}}{k}, \quad b \in \Sigma,\ 1 \le i \le m$$

$$q_{b,0}^{(t+1)} = \frac{\hat{q}_{b,0}}{k(n-m)}, \quad b \in \Sigma$$

- Termination: when $\left\| q^{(t+1)} - q^{(t)} \right\| \le e$ or max iterations reached

Lawrence-Reilly EM

- Constrains in the structure of the profile can be easily incorporated (e.g., being palindrome)
- Variable length gaps within the profile can be handled by adding new variables to the model (that increase the complexity of the model, however)

Megaprior heuristics for MEME

Convex combination problem

- Bailey and Gribskov [ISMB, 1996] describe a problem common to all statistical methods (HMMs, Gibbs, MEME) which discover profiles in protein sequences
- These algorithms are prone to produce profiles that are incorrect because two or more distinct patterns can be incorrectly combined

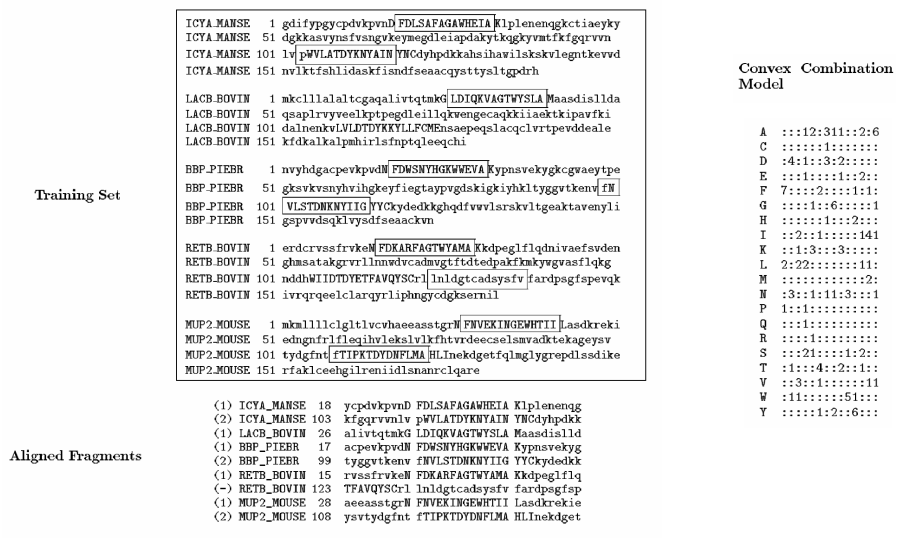
Convex combination problem

- MEME is likely to produce these profile if the estimated number of occurrences is inaccurate or missing
- MEME tends to select a profile that is a combination of two or more patterns because the convex combination can maximize the objective function by explaining more of the data using fewer free parameters

Convex combination problem

- The authors call this profile *convex combination*, because the parameters of the profile that erroneously combines distinct patterns are a weighted average of the parameters of the correct profiles, where the weights are positive and sum up to one – i.e., a convex combination

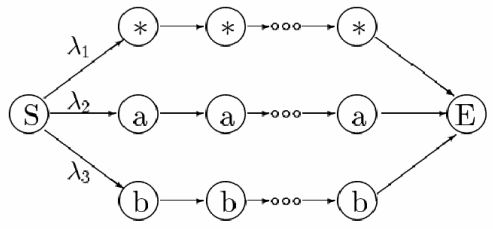
Example of convex combination



From Bailey and Gribskov [ISMB, 1996]

Example

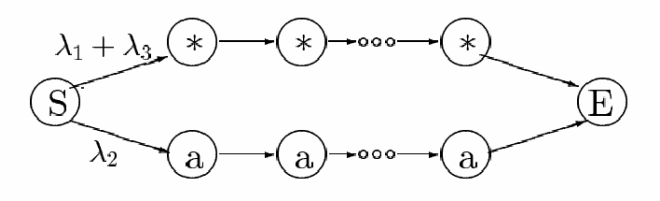
- Suppose we generate a random sequence, with a symmetric Bernoulli source and we inject two substrings of size *m*, **aa…aaa**, and **bb…bbb**
- The following HMM would explain the sequence



From Bailey and Gribskov [ISMB, 1996]

Example

- One would expect that MEME finds

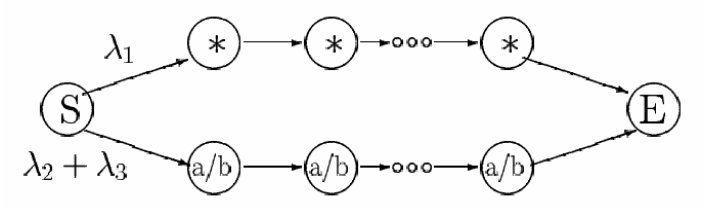


or the one modeling the all "**b**" component

From Bailey and Gribskov [ISMB, 1996]

Example

- Unfortunately, the following convex combination has sometimes higher likelihood



From Bailey and Gribskov [ISMB, 1996]

Convex combination problem

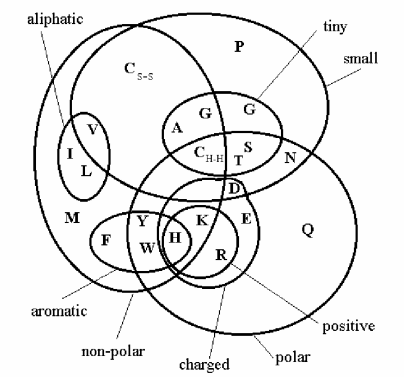
- Convex combinations are undesirable because the make unrelated sequence region to appear to be related
- The problem becomes worse and worse as the size of the alphabet, the length of the profile, or the size of the dataset increases
- In fact, convex combinations are less of a problem with DNA sequences

Convex combination problem

- Bailey and Gribskov propose a heuristic solution based on the use of prior distributions, called *megaprior heuristic*
- Megaprior heuristic is now part of MEME

Megaprior heuristic

- The idea is to use our prior knowledge about the similarities about the amino acids



Megaprior heuristic

- The heuristic is based on the biological knowledge about what constitute a “reasonable” column in a profile
- The prior distribution favor amino-acids in the same class to be in the same column
- Although it does not forbid two amino acid, say one hydrophobic and hydrophilic, to be in the same column, it makes it less likely to happen