Scoring Matrices, Multiple Sequence Alignment

Manu Madhavan

Lecture 7

Recap

- Global and local alignment
- Smith-Waterman algorithm
- BLAST

Outline

- Scoring Matrices
- Multiple sequence alignment methods

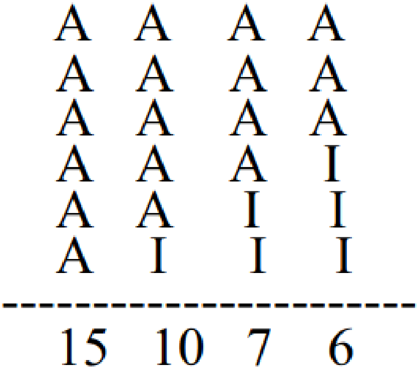
Multiple Sequence Alignment

- Simultaneously align a number of sequences
- $S_1, S_2, ..., S_k$ a set of sequences over the same alphabet. As for the pair-wise alignment, the goal is to find alignment that maximizes some scoring function
- How to score alignment?

MSA: Scoring

Sum of pairs (SP) Score

- Consider all pairs of letters in each column and add the scores
- Let's take match=1 and mismatch=0

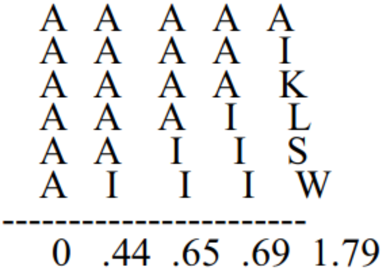


```
 A   A   A   A
 A   A   A   A
 A   A   A   A
 A   A   A   I
 A   A   I   I
 A   I   I   I
-----------------
15  10   7   6
```

MSA: Scoring

Entropy Based scoring

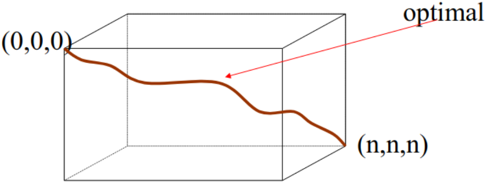
- $Entropy = -\sum_j (\frac{c_j}{C} ln \frac{c_j}{C})$
- c_j is the occurrence of nucleotide j in the column
- C is the number of symbols in the column



```
A  A   A   A   A
A  A   A   A   I
A  A   A   A   K
A  A   A   I   L
A  A   I   I   S
A  I   I   I   W
---------------------
 0  .44 .65 .69 1.79
```

MSA: Multidimensional Dynamic Programming

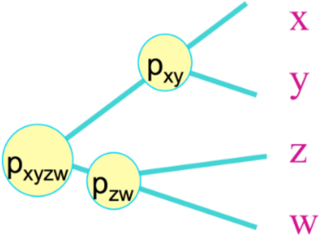
$$\text{Align}(S^1{}_i, S^2{}_j, S^3{}_k) = \max \begin{cases} \text{Align}(S^1{}_{i-1}, S^2{}_{j-1}, S^3{}_{k-1}) + s(a_i, a_j, a_k) \\ \text{Align}(S^1{}_{i-1}, S^2{}_j, S^3{}_{k-1}) + s(a_i, -, a_k) \\ \text{Align}(S^1{}_i, S^2{}_{j-1}, S^3{}_{k-1}) + s(-, a_j, a_k) \\ \text{Align}(S^1{}_{i-1}, S^2{}_{j-1}, S^3{}_k) + s(a_i, a_j, -) \\ \text{Align}(S^1{}_i, S^2{}_j, S^3{}_{k-1}) + s(a_i, -, -) \\ \text{Align}(S^1{}_i, S^2{}_{j-1}, S^3{}_k) + s(-, a_j, -) \\ \text{Align}(S^1{}_{i-1}, S^2{}_j, S^3{}_k) + s(-, -, a_k) \end{cases}$$



- Complexity will be $O(2^k N^k)$, N is the length of the sequence and k is the number of sequences
- Some heuristics should be applied!

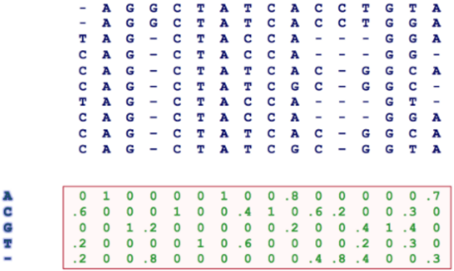
Progressive Alignment

- First align pair(s) of most closely related sequences
- It assumes knowledge of the evolutionary tree
- Then interactively align the alignments to obtain an alignment for larger number of sequences
- A **profile** is when you take pairwise multiple alignment and convert it into a vector of probabilities
- Example: for (A,C,G,T,-):
  $P_x = (0.8, 0.2, 0, 0, 0)$ and $P_y = (0.6, 0, 0, 0, 4)$



Progressive Alignment

- A profile representation of a multiple alignment contains the probabilities of each letter at a given position

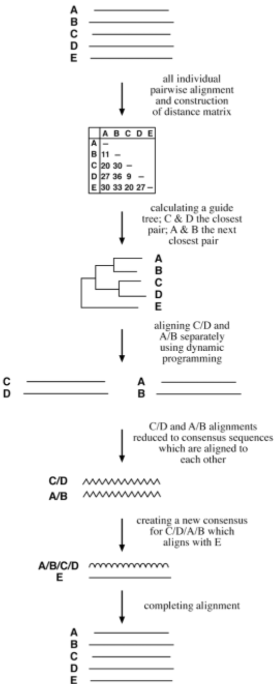


-	A	G	G	C	T	A	T	C	A	C	C	T	G	T	A
-	A	G	G	C	T	A	T	C	A	C	C	T	G	G	A
T	A	G	-	C	T	A	C	C	A	-	-	-	G	G	A
C	A	G	-	C	T	A	C	C	A	-	-	-	G	G	-
C	A	G	-	C	T	A	T	C	A	C	-	G	G	C	A
C	A	G	-	C	T	A	T	C	G	C	-	G	G	C	-
T	A	G	-	C	T	A	C	C	A	-	-	-	G	T	-
C	A	G	-	C	T	A	C	C	A	-	-	-	G	G	A
C	A	G	-	C	T	A	T	C	A	C	-	G	G	C	A
C	A	G	-	C	T	A	T	C	G	C	-	G	G	T	A

A	0	1	0	0	0	0	1	0	0	.8	0	0	0	0	0	.7
C	.6	0	0	0	1	0	0	.4	1	0	.6	.2	0	0	.3	0
G	0	0	1	.2	0	0	0	0	0	.2	0	0	.4	1	.4	0
T	.2	0	0	0	0	1	0	.6	0	0	0	0	.2	0	.3	0
-	.2	0	0	.8	0	0	0	0	0	0	.4	.8	.4	0	0	.3

Progressive Alignment

- Example: for (A,C,G,T,-):
  $P_x = (0.8, 0.2, 0, 0, 0)$ and $P_y = (0.6, 0, 0, 0, 4)$
- For example, we could get the above profiles if P_x referred to the sequence $AAAAAAAACC$ and P_y referred to the sequence $AAAAAA----$
- Substitution scores of P_x and P_y is calculated based on sum of pairs score:
  $s(p_x, p_y) = 0.8 \times 0.6 \times s(A, A) + 0.2 \times 0.6 \times s(C, A) + 0.8 \times 0.4 \times s(A, -) + 0.2 \times 0.4 \times s(C, -)$
- This will result in a new profile P_{xy}



all individual pairwise alignment and construction of distance matrix

	A	B	C	D	E
A	–				
B	11	–			
C	20	30	–		
D	27	36	9	–	
E	30	33	20	27	–

calculating a guide tree; C & D the closest pair; A & B the next closest pair

aligning C/D and A/B separately using dynamic programming

C/D and A/B alignments reduced to consensus sequences which are aligned to each other

creating a new consensus for C/D/A/B which aligns with E

completing alignment

Scoring Matrices

- A simple scheme:
  - A positive value or high score is given for a match
  - a negative value or low score for a mismatch and gaps.
  - This assignment is based on the assumption that the frequencies of mutation are equal for all bases.
- **Transitions**: substitutions between purines[1] and purines or between pyrimidines[2] and pyrimidines
- **Transversions**: substitutions between purines and pyrimidines
- Transitions occurs more frequently than Transversions

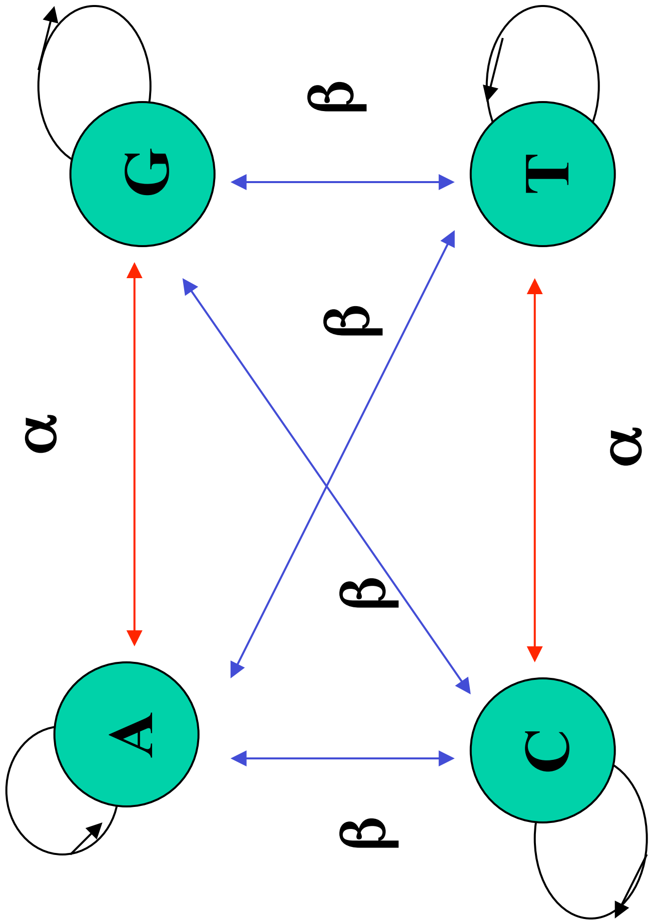
[1] A and G

[2] C and T

Scoring Matrices

- An amino-acid scoring matrix is a 20×20 table such that position indexed with amino-acids so that position X,Y in the table gives the score of aligning amino-acid X with amino-acid Y
- Identity matrix Exact matches receive one score and non-exact matches a different score (1 on the diagonal 0 everywhere else)
- Mutation data matrix a scoring matrix compiled based on observation of protein mutation rates: some mutations are observed more often then other (PAM, BLOSUM).
- Physical properties matrix amino acids with with similar biophysical properties receive high score.
- Genetic code matrix amino acids are scored based on similarities in the coding triple.

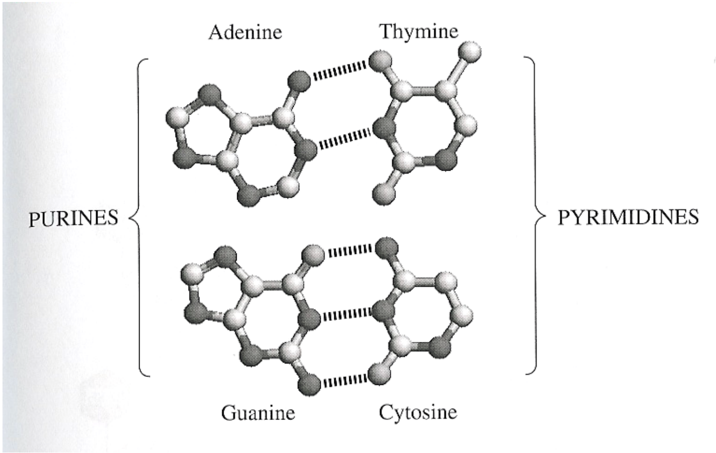
DNA evolution



⟷ transversion

⟷ transitions

β,α –probability of
transition/transversion
in “a unit of time”



PAM unit of time: time needed to acquire 1 mutation per 100 positions

•(Percent Accepted Mutation)

Example

Assuming equal probability for each mutation would be:

	A	T	G	C
A	.99	.0033	.0033	.0033
T	.0033	.99	.0033	.0033
G	.0033	.0033	.99	.0033
C	.0033	.0033	.0033	.99

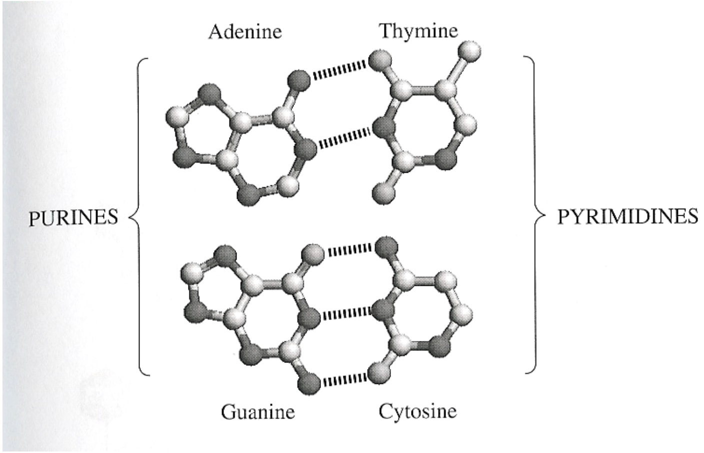
$\alpha = \beta = .0033$

Jukes-Cantor model

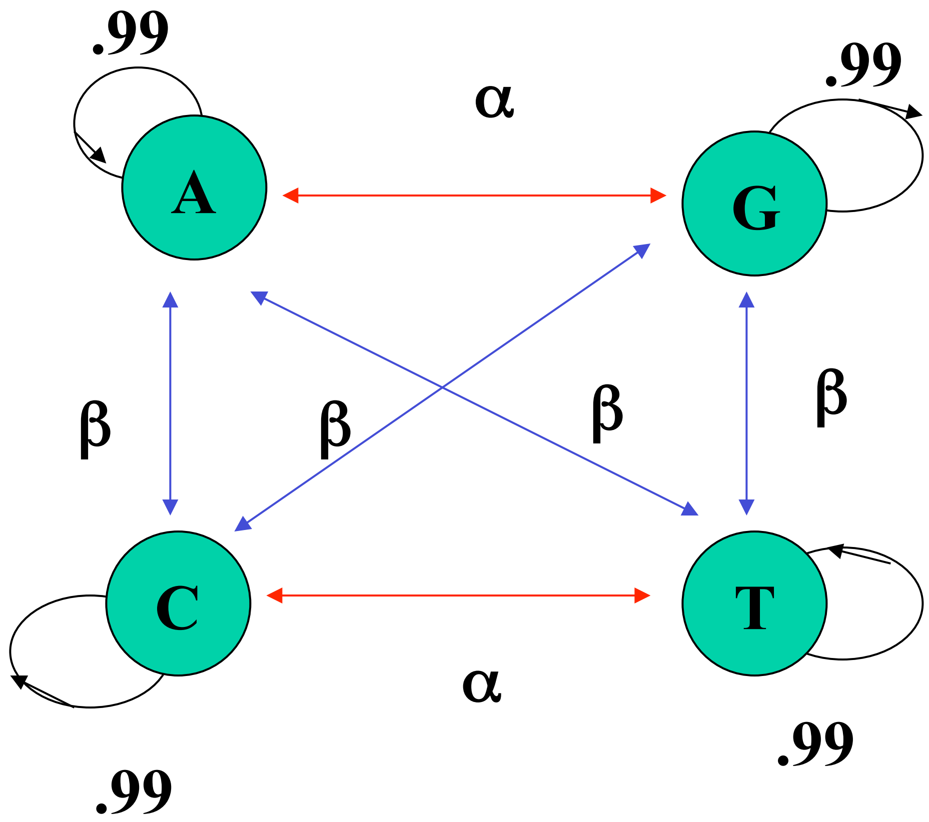
	A	T	G	C
A	.99	.0002	.0006	.0002
T	.0002	.99	.0002	.0006
G	.0006	.0002	.99	.0002
C	.0002	.0006	.0002	.99

$\alpha = .0002 \; \beta = .0006$

Kimura model



Exercise



What is the probability that A mutates to C in:

- One time step: **β**
- In exactly two time steps?

There are four ways of getting from A to C in two steps, sum up the probabilities of each such path.

1. A-A-C
2. A-G-C
3. A-T-C
4. A-C-C

.99*β + αβ +βα+β*.99

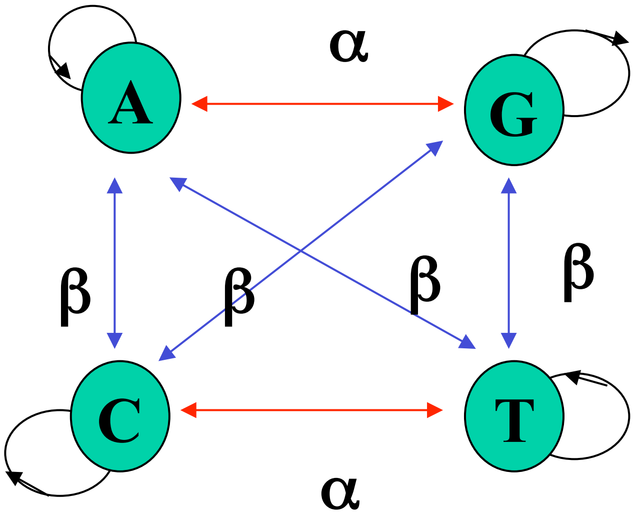
Transition probability in two steps

P^2 (a,b) (Matrix square)

P^2 (a,b) = probability of moving from a to b in exactly two time steps

To see why note that by definition $P^2 (a,c) = \Sigma_k P(a,k)(k,c)$

And recall our example with 4 possible ways of mutation from A to C:



- A-A-C
- A-G-C
- A-T-C
- A-C-C

$99^* \beta + \alpha\beta + \beta\alpha + \beta^*.99$

k

Transition probability in k time steps

P^k (a,b) (Matrix kth power)

This gives us probability of mutation from a to b in k time steps

Log score: from mutation probabilities to scoring matrix

p_b = probability of observing amino acid b

$M(a,b) / p_a\, p_b$ = the odds of seeing substitution as a result of mutation versus seeing it by chance

The odds of seeing the whole alignment by chance versus as a result of mutation will be the product of the above

SCORING FUNCTION:

$$\mathbf{score(a,b) = \log_{10} (M(a,b) / p_a p_b)}$$

Note that score is not necessarily symmetric

Comments on log score

- The idea used in many scoring function
- We take log of the fraction:

$$\frac{\text{frequency of observation}}{\text{probability of the event by chance}}$$

- If the fraction is greater than one (the log is positive) then the observation is more frequent than expected by chance.
- If the observations are independent, the odds are multiplied (and the logs are summed up)

PAM units

PAM – Point Accepted Mutation /Percent Accepted Mutation

Two sequences S and T are defined to be 1 PAM unit diverged if a series of accepted point mutation (and no insertion/deletion) can convert S to T with an average of one mutation per 100 res.

Point accepted mutation – mutation of one residuum accepted by evolution.

Is possible for two sequences to be more than 100 PAM apart?

Yes: One position can mutate multiple times.

How to estimate PAM distances?

Problem 1: given two sequences you cannot tell their PAM distance in the strict sense of the above definition since one residuum could mutate more than once

Problem 2 : A change could happen by deletion followed by insertion and this would look as point mutation

Solution: If we take sequences that are closely related (where mutation are very rear the above problems are unlikely to occur) and then scale the resulting matrix to correspond to 1 PAM unit

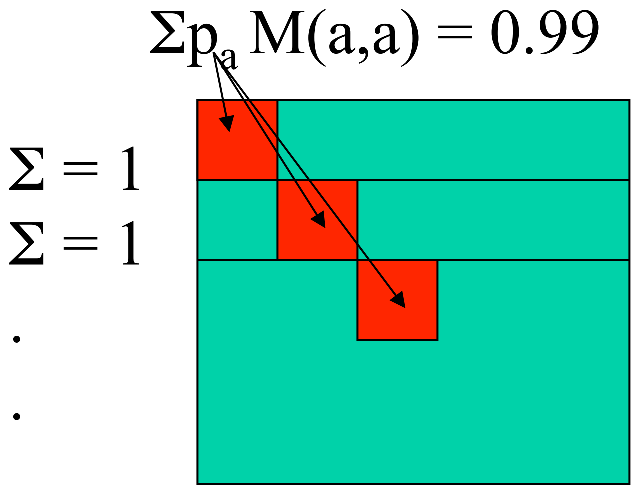
Deriving PAM 1 matrix (Margaret Dayhoff)

- Take a set of highly similar sequences (approximated to be few PAM units apart)
- Align them pair-wise and obtain a list of accepted mutations for the set.

Let **p_a – probability of amino-acid a**
f_{ab} – frequency of substitution (aligning) between a and b

(we assume that mutations are undirected $f_{ab} = f_{ba}$)

First we construct matrix M’ based on the sequences we have and then scale it so that probability of NOT mutating is .99



$\Sigma p_a \, M(a,a) = 0.99$

$\Sigma = 1$

$\Sigma = 1$

.

.

Deriving M' matrix

Let $\mathbf{p_a}$ – **probability of amino-acid a**

$\mathbf{f_{ab}}$ – **frequency of substitution (aligning) between a and b**

(we assume that mutations are undirected $f_{ab} = f_{ba}$)

# of mutation involving a is

$$f_a = \sum_{a \neq b} f_{ab}$$

# total mutations

$$f = \Sigma_a f_a$$

$$\mathbf{M'(a,b) = Pr(a \rightarrow b)}$$

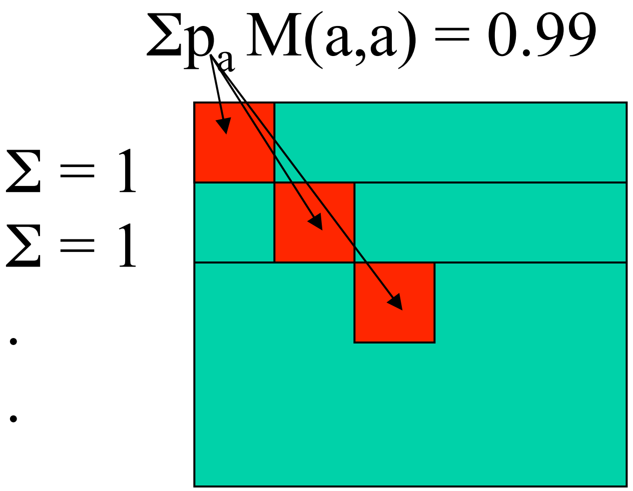
$$\mathbf{= f_{ab}/f}$$

M' (a,b) = probability of mutation between a and b in the set.

We need to scale to estimate how many of them would be per 100 mutations

Scaling M' to obtain M

We need to scale M' to make it PAM 1 that is to ensure $\sum_{a \neq b} p_a M(a,b)=0.01$



Let $m_a=1/100\ f_a/(f\ p_a)$
Set $M(a,b) = M'(a,b)m_a$
$M(a,a) = 1-m_a$

This ensures that

$$\Sigma_b M(a,b) =1$$
$$\Sigma_a p_a M(a,a)=0.99$$

More details in the notes on the class webpage

Next

- Phylogentic Analysis