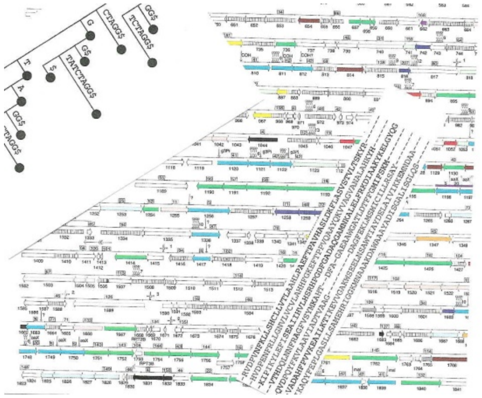
Lecture 3
Scoring Matrices
Position Specific Scoring Matrices
Motifs.

Principles of Computational Biology

Teresa Przytycka, PhD

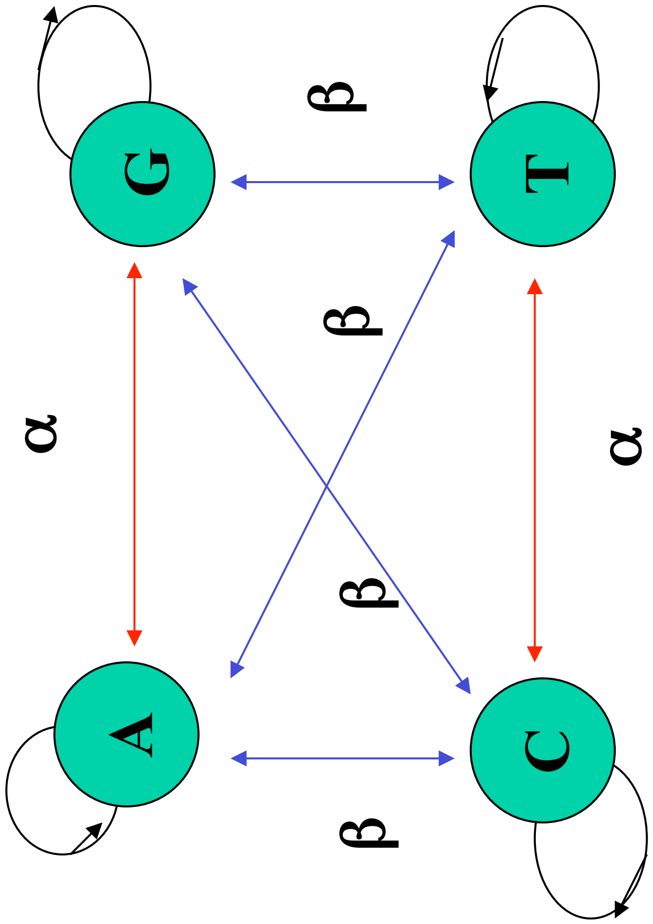


Scoring Matrices

An amino-acid scoring matrix is a 20x20 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.

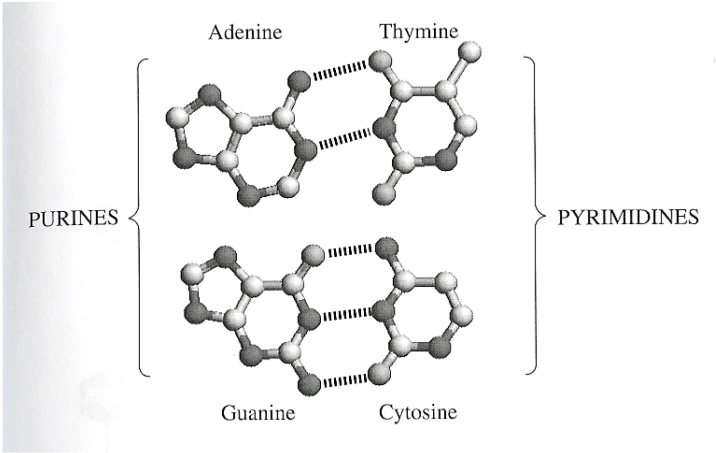
DNA evolution



↔ 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

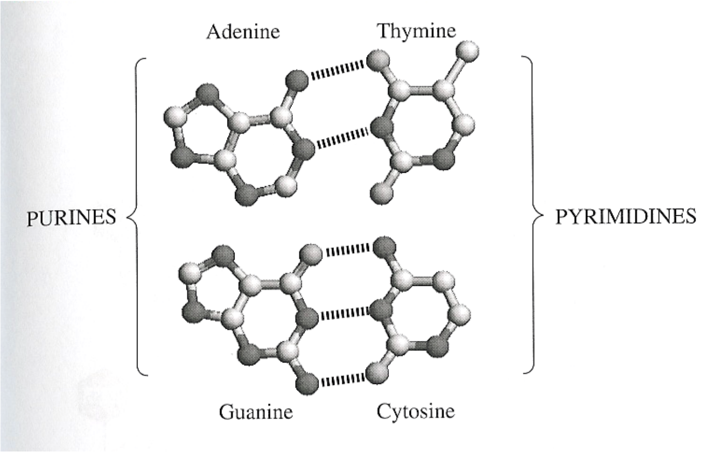
$\alpha = \beta = .0033$

Jukes-Cantor model

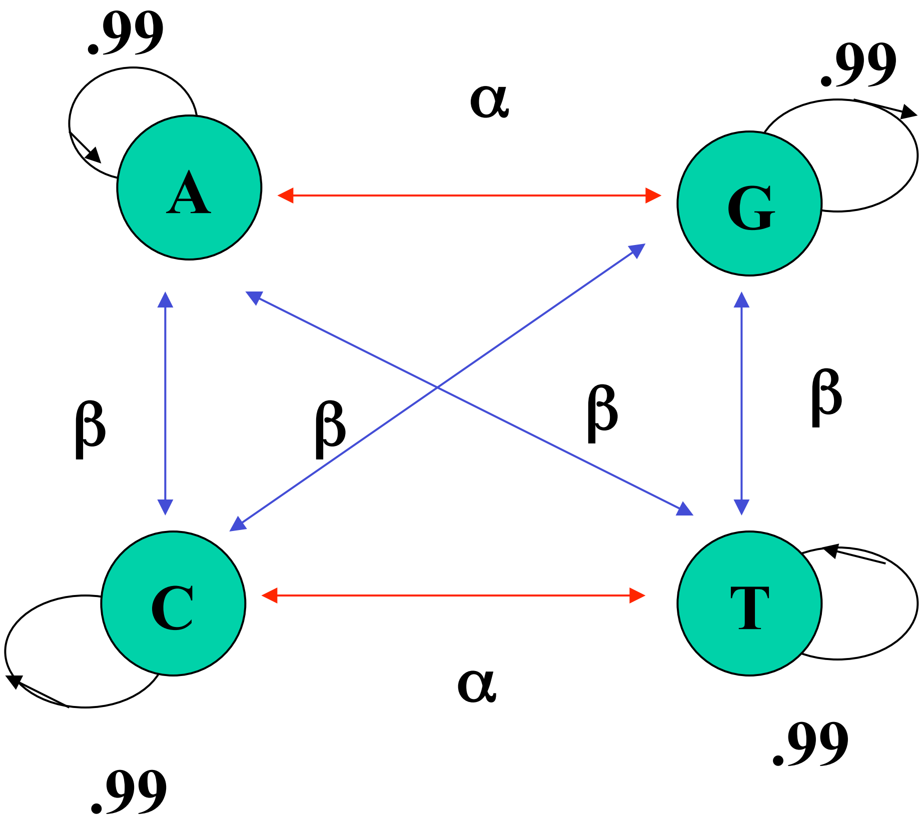
	A	T	G	C
A	.99	.0002	.0006	.0002
T	.0002	.99	.0002	.0006
G	.0006	.0002	.99	.0002
C	.0002	.0006	.0002	.99

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

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

Random walks in graphs

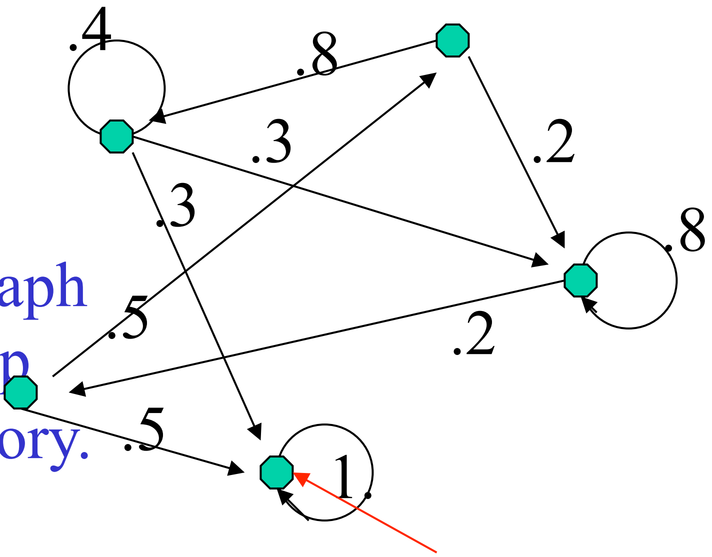
Given is a weighted directed graph such that for each vertex the sum of the weights over all outgoing edges is equal to one.

A random walk is a stochastic process which progresses in steps. Starting from some vertex, each step consists in moving to another vertex with probability equal to the weight of the edge leading to this vertex.

Question: starting at a given vertex what is the probability of reaching another vertex in n steps?

Markov process – a random walk in such graph

Markov property: the probability of next step depends on current stage and not on the history.



Absorbing state

Matrix representation

- P(i,j) = probability of moving form i to j in one step.
- This is a stochastic matrix which means that for each row the sum of the entries equals 1.

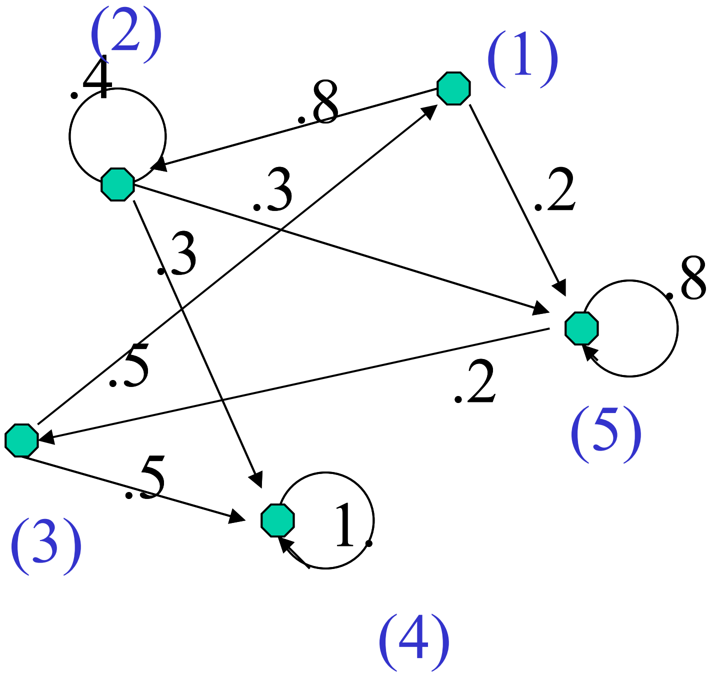
vertices

(states)

$$
\mathbf{P:}\quad
\begin{array}{c|ccccc}
 & (1) & (2) & (3) & (4) & (5) \\
\hline
(1) & 0 & .8 & 0 & 0 & .2 \\
(2) & 0 & .4 & 0 & .3 & .3 \\
(3) & .5 & 0 & 0 & .5 & 0 \\
(4) & 0 & 0 & 0 & 1 & 0 \\
(5) & 0 & 0 & .2 & 0 & .8
\end{array}
$$

edge weights

(transition probabilities)



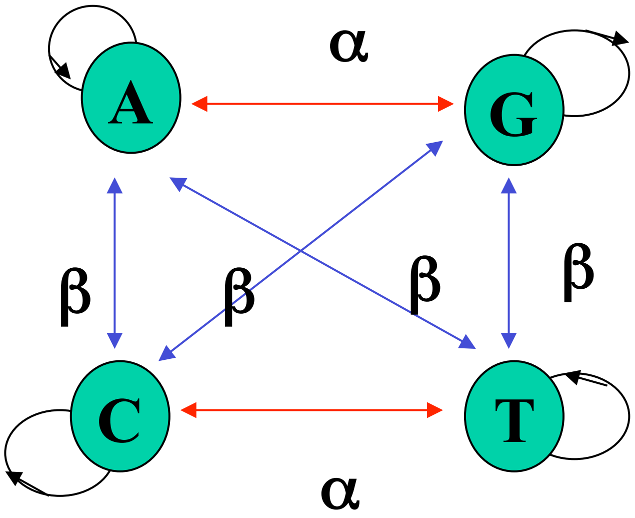
Transition probability in two steps

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 k^{th} 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)

So far:

- If we know the probability of mutation in one time step we can use it to compute probability of mutation in k steps
- If we know probability of mutation and frequency of occurrence of amino acid in we know how to compute the scoring function
- Missing:
  - The probability of mutation between amino-acids in one time unit.

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.

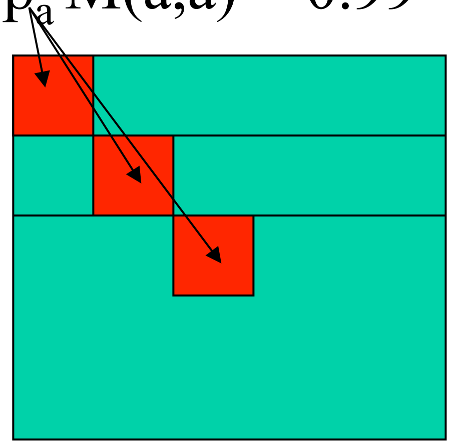
Let **p_a – probability of amino-acid a**

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

$$f = \Sigma_a f_a$$

M' (a,b) = Pr(a→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

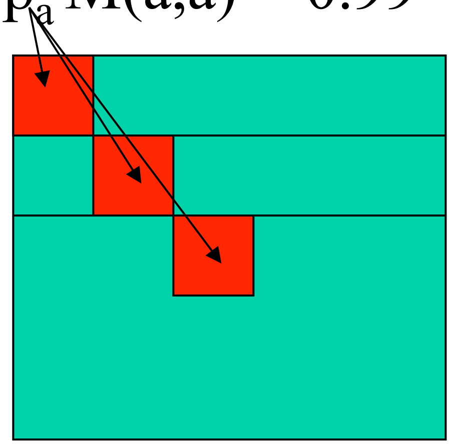
We need to scale M' to make it PAM 1 that is to ensure $\sum_{a \neq b} p_a M(a,b)=0.01$

Let $m_a=1/100\ f_a/(f\ p_a)$

Set $M(a,b) = M'(a,b)m_a$

$M(a,a) = 1-m_a$

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



$\Sigma = 1$

$\Sigma = 1$

.

.

This ensures that

$$\Sigma_b M(a,b) =1$$

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

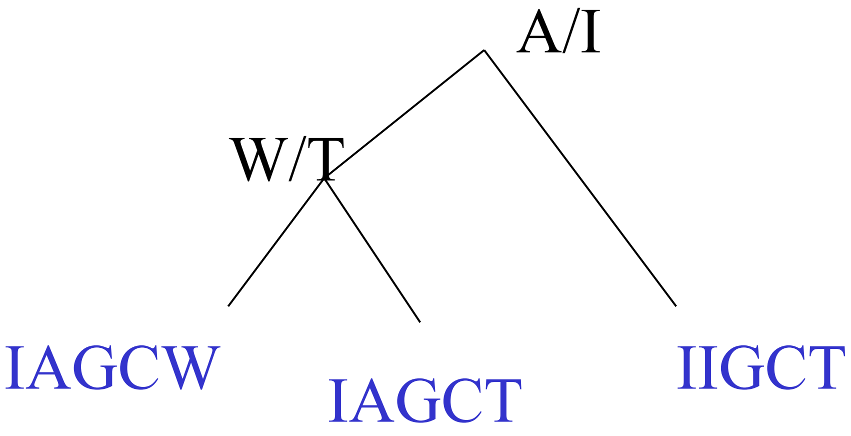
More details in the notes on the class webpage

From Markov model to PAMn

- There is a sequence of PAM matrices PAM1, PAM2,….
- PAM 1 is obtained from M by transforming it to log score
- To obtain PAMn
  - Step 1 : Compute $(M)^n$
  - Step 2: change to log scores
- Dayhoff multiplies the scores by 10 and round up to integer (for faster calculations)

Important detail – how to count substitutions?

Example: In an alignment:
IAGCW
IAGCT
I IGCT
Substitution A – I should not be counted twice. Dayhoff constructs phylogenetic tree and uses the tree and counts substitutions in the tree (to be discussed later with more details)



A tree that minimizes number of changes

Possible problems with Dayhoff's approach

- Some mutation may be rare and underrepresented in PAM1 (which is based on closely related proteins only).
- The mutation rate depends on the position of an amino-acid in the structure.
- Require construction phylogenic tree while algorithms for phylogenetic trees, need scoring matrices for proper construction (remains a problem for many other methods)

BLOSUM

Block Substitution Matrix (Henikoff, Henikoff 1992)
Block – a short contiguous interval of multiple aligned sequences

BLOCKS – data base of of highly conserved aligned sequences representing hundredths of protein groups.
http://blocks.fhcrc.org/blocks/ (http://blocks.fhcrc.org/blocks/)

Step 1: Take BLOCKS alignments and divide them into groups.
Why?
Recall that it was reasonable to use different PAM matrices depending on PAM distances between two sequences.
How?
Choose a percent threshold say 80%

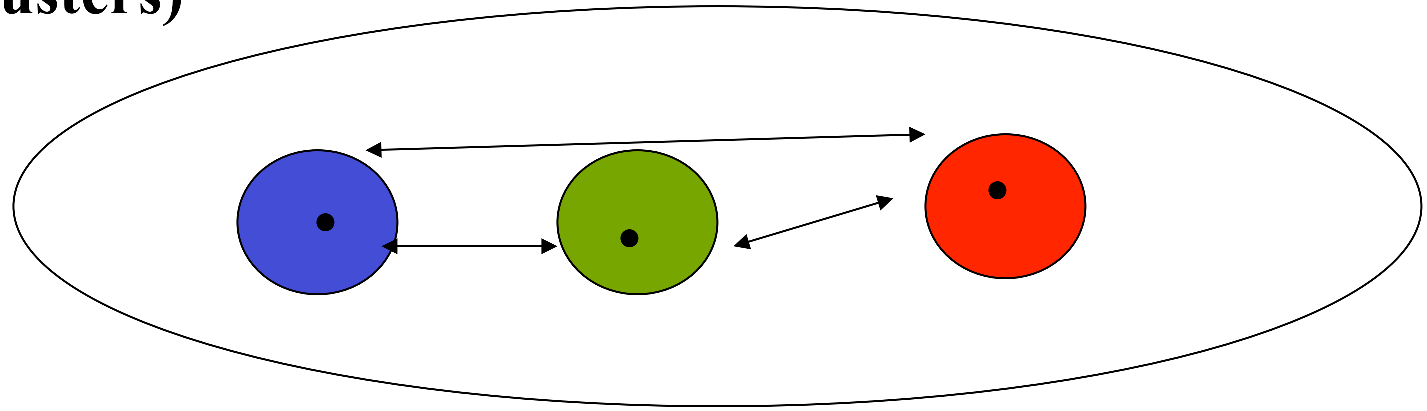
BLOSUM62 – most similar to PAM250 (believed to be better)

BlOSUM80 illustration

•Cluster sequences so that there is 80% or more identity between sequences in each cluster

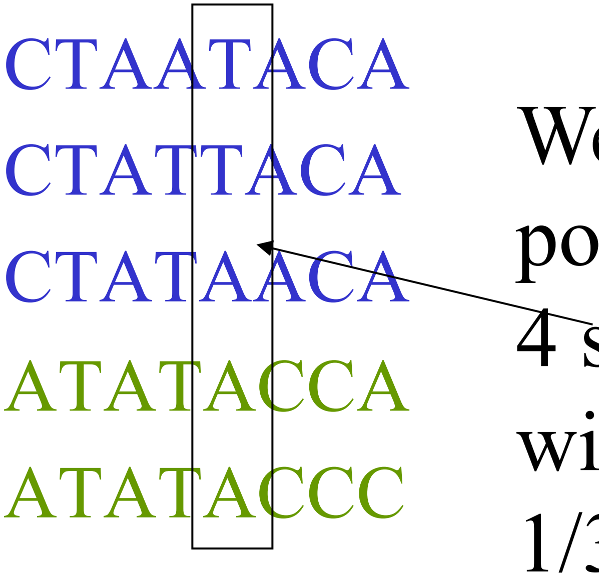
• The percent of similarity between any two sequences in two different clusters should be more than 80%

•In we want to estimate mutations between clusters (not within clusters)



- Difference between sequences in two different clusters is more than 0.80 percent identity
- The clusters sub-clusters are of a large cluster of evolutionary related proteins that can be alignment without gaps

To consider between cluster distances we either use one representative per subcluster or use weighting schema:



CTAATACA
CTATTACA
CTATAACA
ATATACCA
ATATACCC

Weighted frequency of A at two positions:
4 substitutions between A and T with weight
1/3*1/2 = 4/6

Towards Position Specific Scoring Matrices (PSSM)

Scoring matrices discussed so far are used in pairwise sequence alignment (previous class) .

Can be used to;

- estimate the evolutionary distance between a pair of proteins.
- In a sequence data base, find proteins similar to a given proteins (we will discuss methods to do so in the next class).

A different scenario: we have a set of related proteins and ask the question if a given sequence is a member of this set.

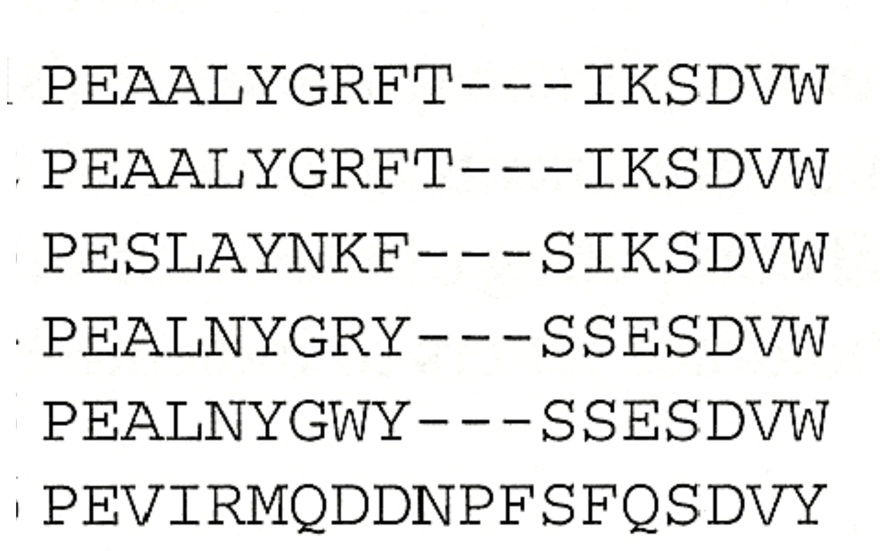
Towards Position Specific Scoring Matrices (PSSM)

Naïve solutions:

- Use one protein from the set and compare with a query proteins
- Use all proteins in the set and compare each with a query proteins
- Compare to a consensus sequence

Consensus sequence

- A sequence where each position is defined by majority vote based on multiple sequence alignment.



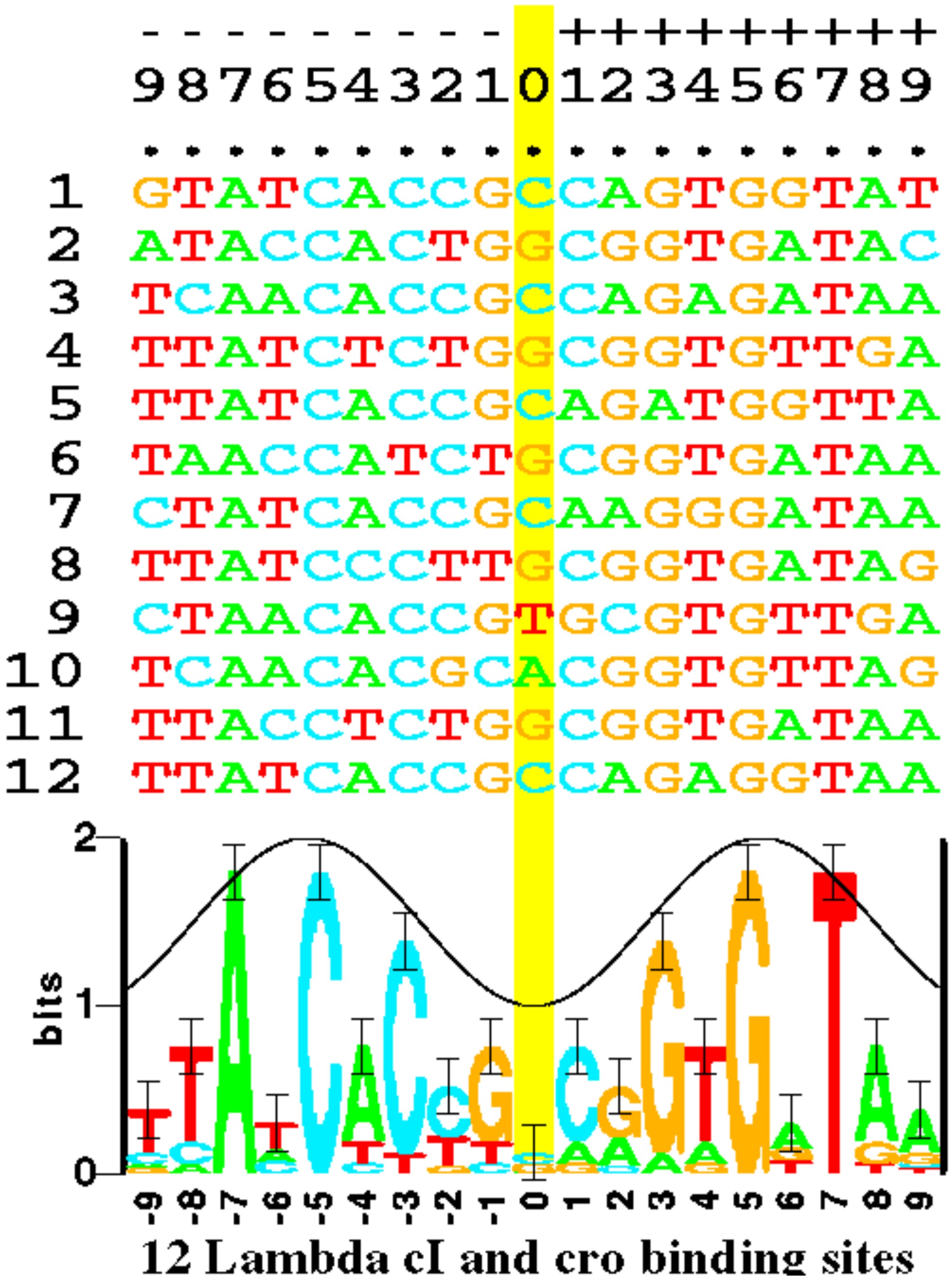
```
PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLAYNKF---SIKSDVW
PEALNYGRY---SSESDVW
PEALNYGWY---SSESDVW
PEVIRMQDDNPFSFQSDVY
```

PEALNYGRFTPFS I KSDVW

http://www.lecb.ncifcrf.gov/~toms/sequencelogo.html

Explore this site!

Sequence Logos



```
    ---------0+++++++++
    9876543210123456789
    ...................
 1  GTATCACCGCCAGTGGTAT
 2  ATACCACTGGCGGTGATAC
 3  TCAACACCGCCAGAGATAA
 4  TTATCTCTGGCGGTGTTGA
 5  TTATCACCGCAGATGGTTA
 6  TAACCATCTGCGGTGATAA
 7  CTATCACCGCAAGGGATAA
 8  TTATCCCTTGCGGTGATAG
 9  CTAACACCGTGCGTGTTGA
10  TCAACACGCACGGTGTTAG
11  TTACCTCTGGCGGTGATAA
12  TTATCACCGCCAGAGGTAA
```

12 Lambda cI and cro binding sites

A sequence logo displays the frequencies of bases at each position, as the relative heights of letters. Total height is measured equals information content for this position and illustrates conservation of the position.

Examples:

- Recall that:

$I(P) = H(R)- H(P) = \Sigma_{i=1..n} r_i \log (r_i) - \Sigma_{i=1..n} p_i \log (p_i)$

column 7:

$I(\text{column } 7) = - 4 *(1/4 \log (1/4)) + 1 \log 1 = 2$

- column 6:

$I(\text{column } 6) = - 4 *(1/4 \log (1/4)) + 3/12 \log(3/12)+3/12 \log(3/12) + (6/12) \log(6/12)$

$= 2 - (1/4 \log ¼+ 1/4 \log ¼+1/2 \log(1/2))$

$= 2 -½- ½ -1/2 = -1.5 + 2 = .5$

Searching data base for members of a protein family

Goal: Given a family search for other proteins/motives that match the family.

Methods:

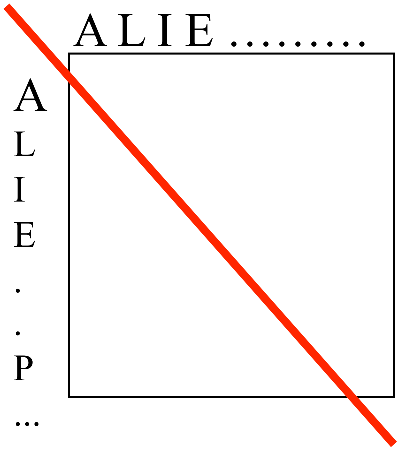
- Alignment to consensus sequence
- Alignment to a family profile
- Searching against family “fingerprint”
- HMMs and other probabilistic family models (a later lecture)

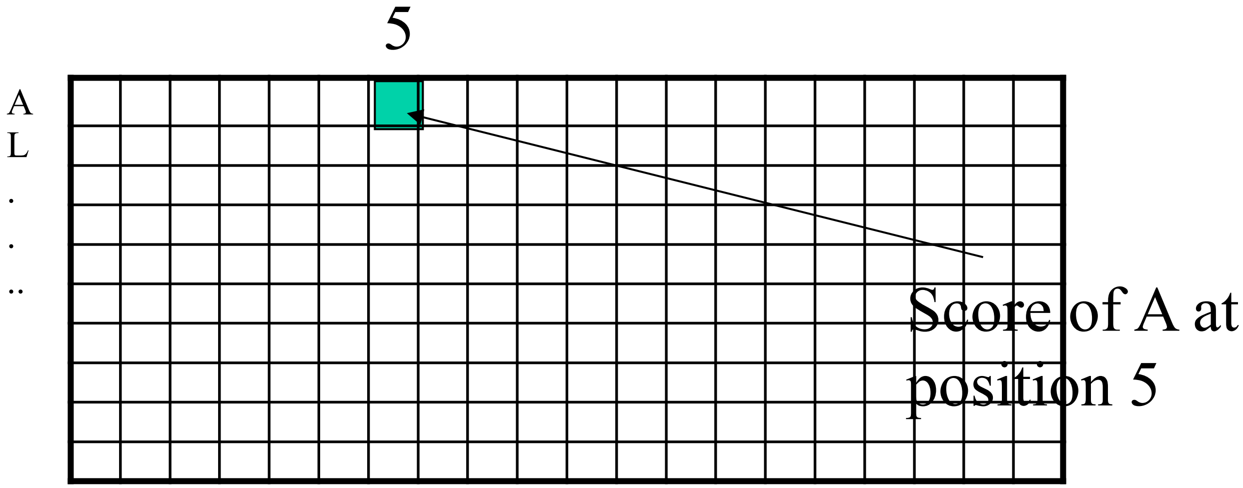
Profile alignment

- Given: Sample sequences of a protein family.
- Goal: provide a stochastic model of the family.
- Method:
  - Multiple align sequences in the family.
  - Construct position specific score matrix: the score $s_i(a)$ of amino acid a on the position i and depends on frequency of observing a in this position in multiple alignment.
    - Usual considerations – should sequences in the family be weighted equally?
    - Pseudo-counter – create small frequencies for amino-acids not represented in the profile.

Position Specific Scoring Matrices

Scoring matrix where the alignment score depends on position.
“Alignment” = alignment to a profile





Computing scores for the profile:

- log odds score (next slide)

Family profile

Multiple sequence alignment

```
PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLAYNKF---SIKSDVW
PEALNYGRY---SSESDVW
PEALNYGWY---SSESDVW
PEVIRMQDDNPFSFQSDVY
```

Collect frequencies of each amino-acid (and gaps for gapped alignment) at each position*

A	0
L	0
I	0
E	0
.	.
.	.
	.
	.
P	6
…	…

Compute log odds score for observing given amino-acid at given position.

*) Usually pseudocounts are added to avoid zero probabilities – before collecting the frequency data all position are not initialized to zero but to a pseudocount value

Example

```
PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLAYNKF---SIKSDVW
PEALNYGRY---SSESDVW
PEALNYGWY---SSESDVW
PEVIRMQDDNPFSFQSDVY
```

	1	2	3	4	5	6	7	8	9
P									
E									
…									
-									

P_i = prob. of proline at pos i.

P = prob. of proline in the data base

Score for P in pos i is:

log(Pi/P)

Estimating P_i:

$$\frac{\text{\#of P at col. i + pseudo-counter for P}}{\text{\#rows + sum of pseudo-counters}}$$

Choosing Pseudocounts values:background a.a. distribution; or equal value to each a.a.

Let pseudo-counter = 1/21 for each a.a and gap

$P_1 = (6+ 1/21) /(6 + 21/21)$

$E_1 = (0+ 1/21)/ (6+21/21)$

This is under assumption that sequences are equally similar.

If the sequences form clusters of highly similar sequences weighting sequences in necessary

CTAATACA
CTATTACA
CTATAACA

Weight each sequence 1/3

ATATACCA
ATATACCC

Weight each sequence = 1/2

…or remove redundant sequences if the dataset is large