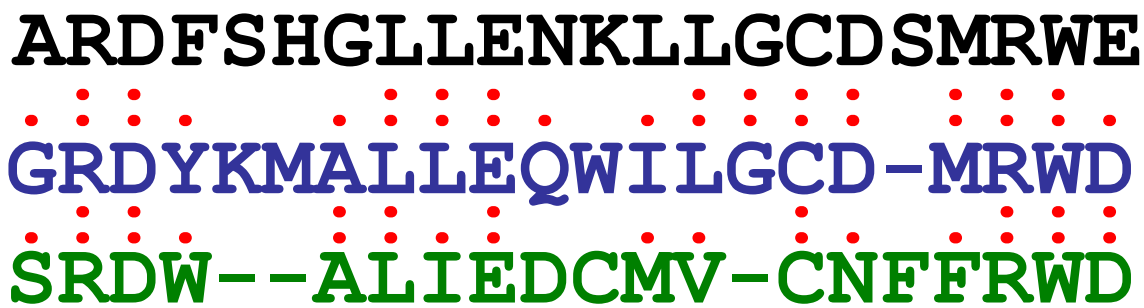
7.93 – Lecture #4

More Pairwise Sequence Comparisons

- and –

Multiple Sequence Alignment



Outline

- Definitions of PAM matrices &PAM matrix calculations
- Other amino acid similarity matrices-BLOSUM, Gonnet
- Gaps – linear vs affine
- Alignment statistics (brief!)
- What you need to know to do an alignment

Outline (cont)

- Multiple sequence alignments: MSA, Clustal
- Block analysis
- Position-Specific Scoring Matrices (PSSM)
- Information content, Shannon entropy
- Sequence logos
- Other approaches: Genetic algorithms, expectation maximization, MEME,Gibbs sampler
- Hidden Markov Models

Amino Acid Substitution Matrices

Margaret Dayhoff, 1978, PAM Matrices

****Evolutionary model** ****based on a small data set.**

Assumes symmetry: A→B = B→A

Assumes amino acid substitutions observed over short periods of time can be extrapolated to long periods of time

71 groups of protein sequences, 85% similar
1572 amino acid changes.
Functional proteins ↔"Accepted" mutations by natural selection

PAM1 matrix means probablilty of each amino acid changing into another is ~ 1% and probability of not changing is ~99%

Construction of a Dayhoff Matrix: PAM1

Step 1: **Measure pair exchange frequencies for each amino acid within families of related proteins**

```
….GDSFHYFVSHG….
….GDSFHYYVSFG….
….GDSYHYFVSFG….
….GDSFHYFVSFG….
….GDSFHFFVSFG….
```

900 Phe (F)….+ another 100 probable Phe but…

100 Phe (F) → 80 Tyr (Y), 3 Trp (W), 2 His (H)….

Gives A_{ij}, i.e. $A_{FY}=80/1000=0.08$

$A_{FW}=3/1000=0.003$

….by evolution!

Do this for all 20 amino acids

	C	D	E	F	G	H	I	.....
C	A_{CC}	A_{CD}	A_{CE}					
D	A_{DC}							
E	A_{EC}							
F								
G								

Gives A_{ij} = pair exchange frequency

Step 2: Measure frequency of occurrence of each amino acid in the whole collection of protein sequences

$$f_i = \frac{\text{Observations of amino acid i}}{\text{Observations of all amino acids}}$$

$$f_i = \frac{\text{Observations of amino acid i}}{\sum_{i=1}^{20} aa_i}$$

$$\text{i.e. } f_{Phe} = \frac{1000}{40{,}000} = 0.04$$

Step 3: Calculate relative mutabilities of each a.a.

Chance on average that a given a.a. will mutate

$$m_i = f_i \times \text{(number of times amino acid i is observed to mutate)}$$

example

$$m_{Phe} = f_{Phe} \times \text{(number of times Phe is observed to mutate)}$$

$$m_{Phe} = 0.04 \times (100) = 4$$

Step 4: Calculate mutation probability matrix for each amino acid being replaced by every other amino acid

Chance that a given a.a. j will replace a.a. i

$$M_{ij} = m_i \times \frac{A_{ij}}{\sum_{i=1}^{20} A_{ij}}$$

example

$$M_{FY} = m_{Phe} \times \frac{\text{(number of times Phe} \rightarrow \text{Tyr)}}{\text{(number of times Phe} \rightarrow \text{mutates)}}$$

$$M_{FY} = 4 \times \frac{(0.08)}{(0.10)} = 3.2$$

Step 5: Calculate evolutionary distance scale so that only 1/100 amino acids change

M_{ii} reflects amino acid conservation

$$M_{ii} \propto 1 - \sum_{i=1}^{20} M_{ij}$$

example

$$M_{FF} \propto 1 - \text{(frequency of Phe mutations)}$$

***Use a scale factor λ so that M_{ii} is ~ 0.99*
i.e. chance of it mutating is ~ 1%

i.e. *this defines a PAM1 matrix….*

$$M_{ii} = 1 - \lambda \sum_{i=1}^{20} M_{ij} = \sim 0.99$$

λ is our evolutionary scale factor

… and for any particular mutation probability, λM_{ij} reflects the normalized measure of how likely amino acid j will replace amino acid i over 1 PAM

Real PAM1 values

Amino Acid Change	*PAM 1 Probability Score*
F→A	0.0002
F→R	0.0001
F→N	0.0001
F→D	0.0000
F→C	0.0000
F→Q	0.0000
F→E	0.0000
F→G	0.0001
F→H	0.0002
F→I	0.0007
F→L	0.0013
F→K	0.0000
F→M	0.0001
F→F	0.9946
F→P	0.0001
F→S	0.0003
F→T	0.0001
F→W	0.0001
F→Y	0.0021
F→V	0.0001

SUM = 1.0

Note – this is really just 1 column in a much bigger probability matrix

	….. E	F	G ……
A		0.0002	
C		0.0000	
D		0.0000	
E		0.0000	
F		0.9946	
G		0.0001	

Next, assume that mutations at each site are independent of previous mutations. Therefore, calculate changes predicted for more distantly related proteins that have undergone N mutations/100 amino acids by *multiplying the PAM1 matrix against itself N times.*

Example: PAM2 matrix:

	aa1	aa2	aa3
aa1	a	b	c
aa2	d	e	f
aa3	g	h	i

X

	aa1	aa2	aa3
aa1	a	b	c
aa2	d	e	f
aa3	g	h	i

	aa1	aa2	aa3
aa1	A	B	C
aa2	D	E	F
aa3	G	H	I

$$A=a^2+bd+cg+\ldots$$

$$B=ab+be+ch+\ldots$$

$$C=ac+bf+ci+\ldots$$

$$D=da+ed+fg+\ldots$$

PAM 250 matrix

- Multiply PAM1 matrix by itself 250 times!

(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)
(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)(PAM1)

Amino Acid Change	PAM 1 Score	PAM 250 Score
F→A	0.0002	0.04
F→R	0.0001	0.01
F→N	0.0001	0.02
F→D	0.0000	0.01
F→C	0.0000	0.01
F→Q	0.0000	0.01
F→E	0.0000	0.01
F→G	0.0001	0.03
F→H	0.0002	0.02
F→I	0.0007	0.05
F→L	0.0013	0.13
F→K	0.0000	0.02
F→M	0.0001	0.02
F→F	0.9946	0.32
F→P	0.0001	0.02
F→S	0.0003	0.03
F→T	0.0001	0.03
F→W	0.0001	0.01
F→Y	0.0021	0.15
F→V	0.0001	0.05

These are the M_{ij} values! i.e. the chance that one amino acid will replace another at 250 PAMs in two proteins that are evolutionarily related to each other!

SUM = 1.0

PAM 250 matrix – 250% expected change

Sequences still ~ 15-30 % similar, i.e. Phe will match Phe ~ 32% of the time
Ala will match Ala ~ 13% of the time

Expected % similarity

Other PAM matrices:

PAM 120 – 40%
PAM 80 – 50%
PAM 60 – 60%

} **Use for similar sequences**

PAM250 – 15-30% similarity.

Where do the numbers in the PAM250 Matrix table come from?

Step 6: Calculate relatedness odds

Chance that two amino acids in a sequence alignment come from related proteins via evolution versus the chance that they are from two unrelated proteins aligned by chance.

M_{ij} = prob. that j replaces i in related proteins

-vs –

P_i^{ran} = prob. that j replaces i because the proteins are completely unrelated…i.e. i was there by chance

Now, $P_i^{ran} = f_i$, the frequency of occurrence of amino acid i

Where do the numbers in the PAM250 Matrix table come from?

Step 6: Calculate relatedness odds

Relative odds of evolution rather than chance:

$$R_{ij} = \frac{M_{ij}}{f_i}$$

Where do the numbers in the PAM250 Matrix table come from?

Step 7: Calculate log (relatedness odds) and multiply by 10 to clear fractional values

Example: Phe →Tyr (which must = Tyr →Phe)

$$R_{ij} = \frac{M_{ij}}{f_i}$$

$M_{FY} = 0.15$
$f_{Phe}=0.04$
So $R_{FY}=0.15/0.04 = 3.75$
$Log_{10}R_{FY}=Log_{10}(3.75)=0.57$
$10 \times 0.57 = 5.7$

Likewise
$M_{YF} = 0.20$
$f_{Tyr}=0.03$
So $R_{YF}= 6.7$
$Log_{10}(6.7)=0.83$
$10 \times 0.57 = 8.3$

So average = (5.7+8.3)/2 = 7….the number in the PAM250 table!

Remember…

Saw last time how to use these numbers + dynamic programming in order to "score" an alignment…

But we have to use the right matrix!!!

PAM 250 matrix – 250% expected change

Sequences still ~ 15-30 % similar, i.e. Phe will match Phe ~ 32% of the time
Ala will match Ala ~ 13% of the time

Expected % similarity

Other PAM matrices: **PAM 120 – 40%**
PAM 80 – 50%
PAM 60 – 60%

} **Use for similar sequences**

PAM250 – 15-30% similarity.

Use the correct PAM matrix for alignments based on how similar the sequences to be algned are! But wait…..how do we know that in the first place? Usually don't!!!!.
So…… try PAM200, PAM120, PAM60, PAM80, and PAM30 matrix and use the one that gives the highest ungapped aligment score

Alternative amino acid matrices

- Gonnett, Cohen & Benner
  -All against All database matching using DARWIN
  1,700,000 matches
  Compile mutation matrices at different PAMs DIRECTLY

- BLOSUM = Blocks Amino Acid Substitution Matrices
  -based on a much larger dataset from Prosite families identified by Bairoch using conserved amino acid blocks that define each family.
  Typically used for multiple sequence alignment.
  AA substitutions noted, log odds ratios derived.

  for example…Block patterns 60% identical give rise to Blosum60 matrix, etc.…i.e. conservation of functional blocks.

  Not based on explicit evolutionary model

GAPS

```
AKHFRGCVS
AKKF--CVG
```

- **Linear Gap Penalty**

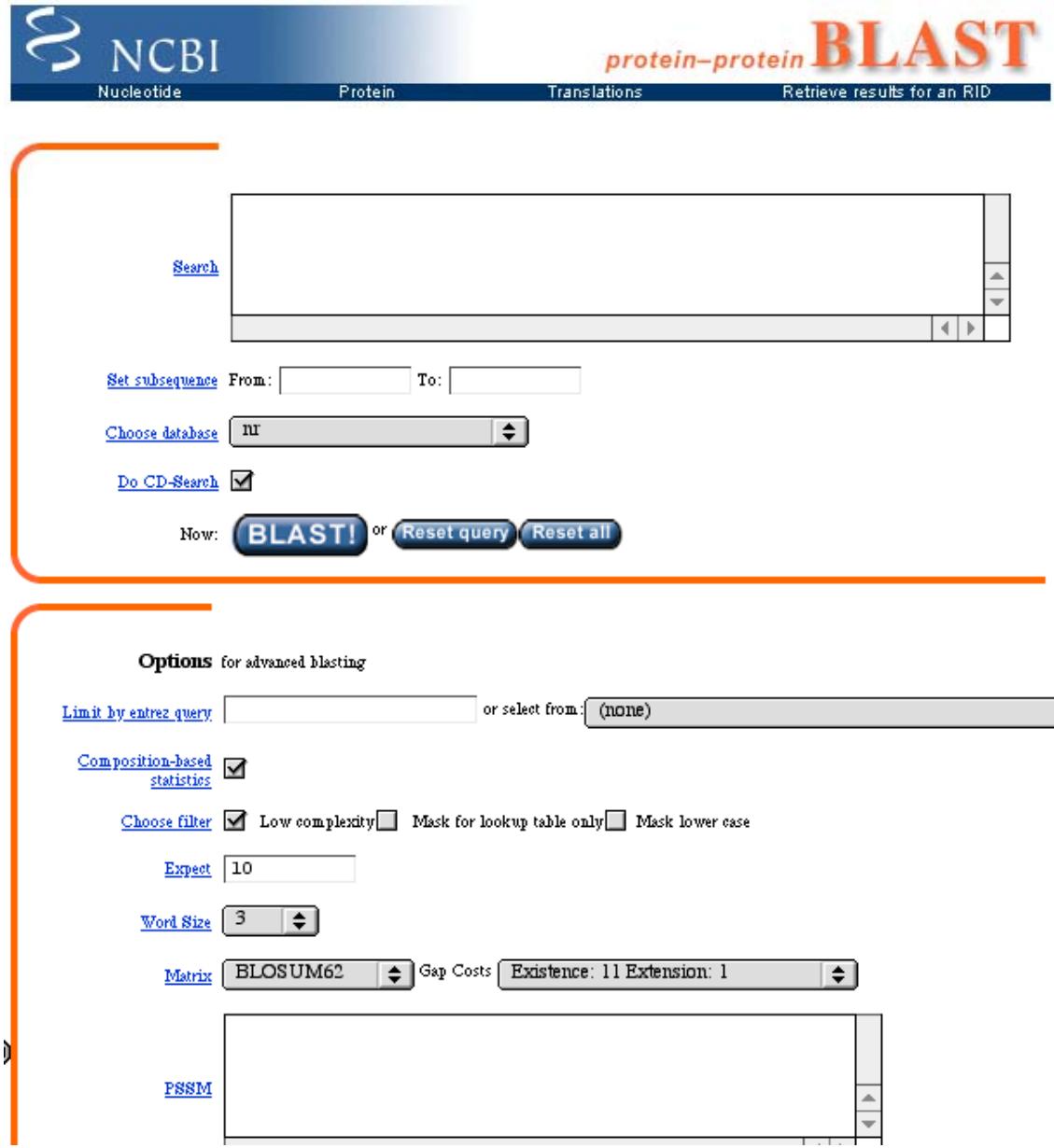
$$W_n = n\gamma,$$

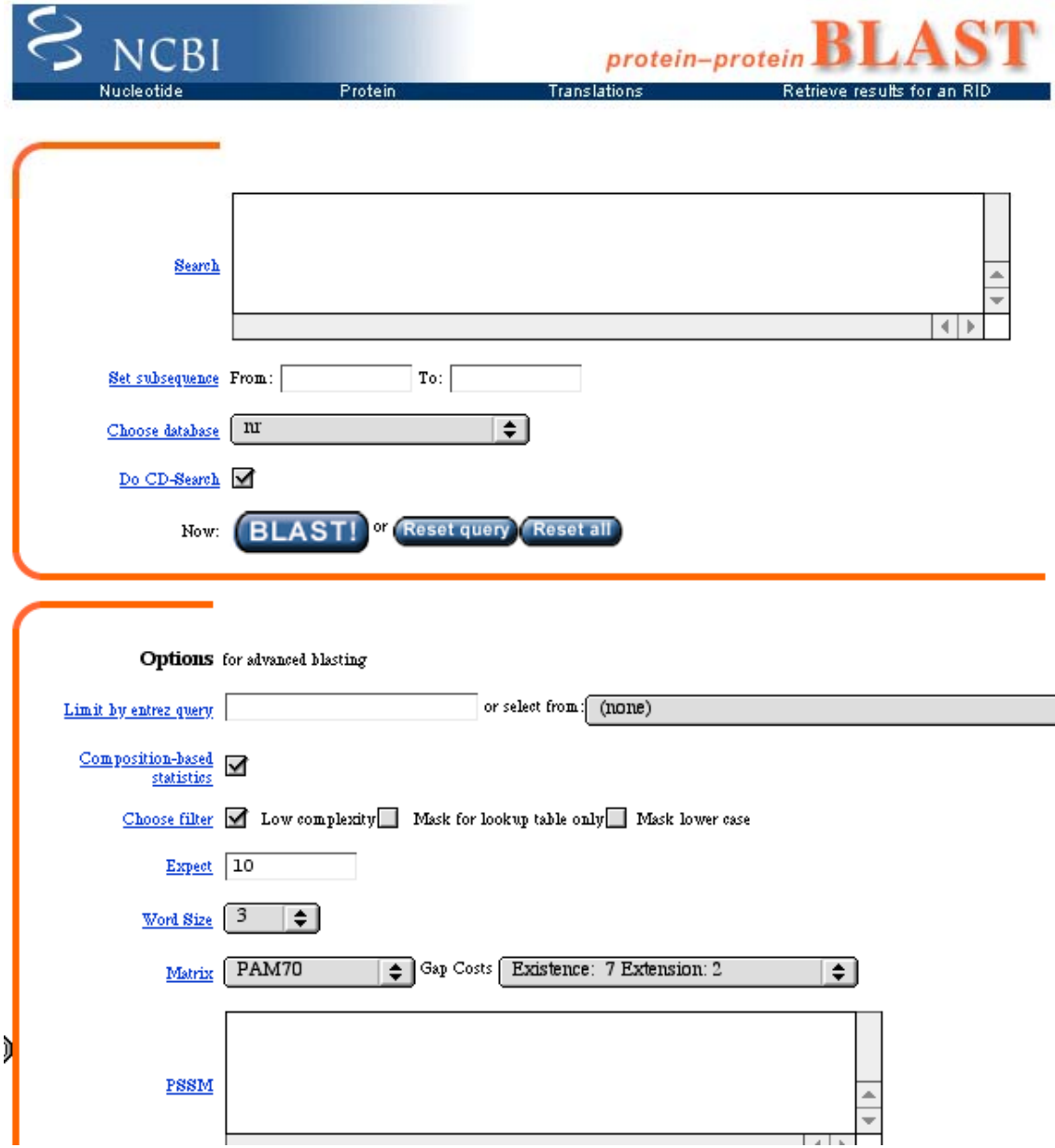
n= # of gaps, γ = gap penalty

- **Affine gap penalty**

$$W_n = g + n\gamma,$$

=# of gaps, γ = gap extension penalty,
and g = gap opening penalty


NCBI
protein–protein BLAST
Nucleotide
Protein
Translations
Retrieve results for an RID
Search
Set subsequence
From:
To:
Choose database
nr
Do CD-Search
Now:
BLAST!
or
Reset query
Reset all
Options for advanced blasting
Limit by entrez query
or select from:
(none)
Composition-based statistics
Choose filter
Low complexity
Mask for lookup table only
Mask lower case
Expect
10
Word Size
3
Matrix
BLOSUM62
Gap Costs
Existence: 11 Extension: 1
PSSM


NCBI
protein–protein BLAST
Nucleotide
Protein
Translations
Retrieve results for an RID
Search
Set subsequence
From:
To:
Choose database
nr
Do CD-Search
Now:
BLAST!
or
Reset query
Reset all
Options for advanced blasting
Limit by entrez query
or select from:
(none)
Composition-based statistics
Choose filter
Low complexity
Mask for lookup table only
Mask lower case
Expect
10
Word Size
3
Matrix
PAM70
Gap Costs
Existence: 7 Extension: 2
PSSM

Simplified Alignment Statistics

- How can we tell how good an alignment is based on its score? What are the chances that two random sequences would give a similar score when they were aligned?

- Consider an easier problem – what is the longest run of heads I will get in a random series of coin tosses?

  Fair coin p=0.5, Erdös and Rényi – longest run = $\log_{1/p}(n)$

  here this is $\log_2(n)$. If n=100, longest run is 6.65

- For two sequences of length n and m, we're doing nm comparisons, so the longest length of the predicted match would be $\log_{1/p}(mn)$

- More precisely, the expectation value, or the mean of the longest match turns out to be $E(M) \sim = \log_{1/p}(Kmn)$ where *K* is a constant that depends on amino acid composition.

  ….OK, this is really only true for ungapped local alignments and I'm neglecting edge effects and mismatches

A few notes…

- **$E(M) \sim = \log_{1/p}(Kmn)$ means that the match length gets bigger as the log of the product of the sequence lengths. Using the amino acid substitution matrices, we can turn these match lengths into alignment scores, S.**
- **More commonly see two parameters used: $\lambda = \ln(1/p)$ and the parameter *K* we already talked about.**
- **We want to know the number of High-Scoring Pairs, HSP, (i.e. high scoring runs of amino acids).**
- **This number of HSPs, E, that exceed some score S is given by $E = Kmne^{-\lambda S}$**
- **So we can evaluate how good a sequence scores, (i.e. its S) by looking at how many HSPs (i.e. E value) we would expect for that score.**

Notes (cont).

- **Where do we get that distribution function that tells us how E and are related? Need to look at the scores in some model of aligned random sequences….**

Notes (cont)

- The random sequence alignment scores would give rise to an "extreme value" distribution – like a skewed gaussian.

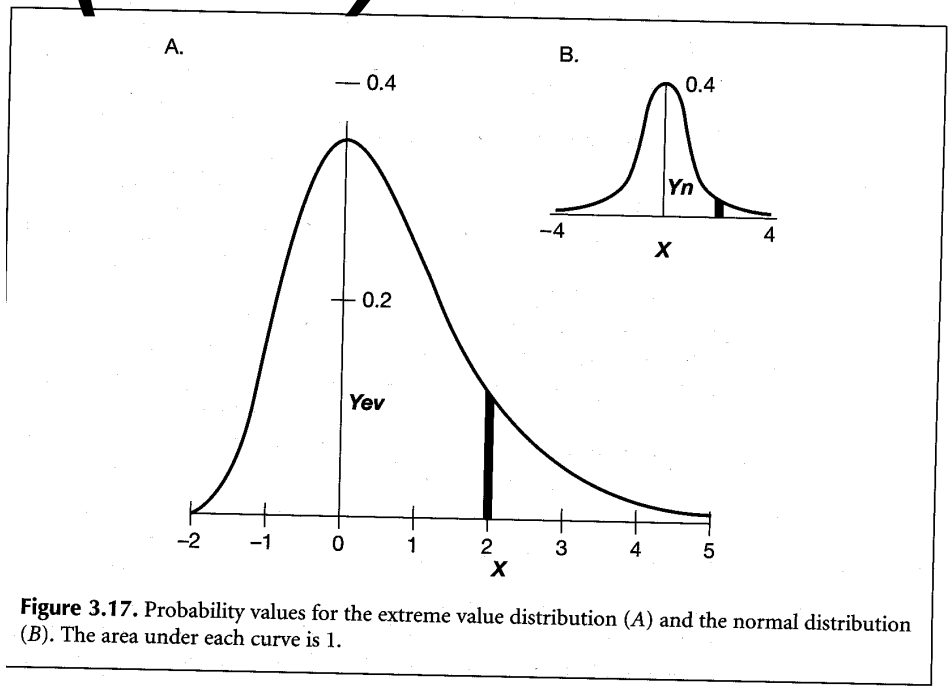


Figure 3.17. Probability values for the extreme value distribution (*A*) and the normal distribution (*B*). The area under each curve is 1.

For a normal distribution with a mean m and a variance σ, the height of the curve is described by $Y=1/(\sigma\sqrt{2\pi})\ \exp[-(x-m)^2/2\sigma^2]$

For an extreme value distribution, the height of the curve is described by $Y=\exp[-x-e^{-x}]$ …and $P(S\geq x) = 1-\exp[-e^{-\lambda(x-u)}]$ where $u=(\ln K mn)/\lambda$

Can show that mean extreme score is ~ $\log_2(nm)$, and the probability of getting a score that exceeds some number of "standard deviations" x is: $P(S\geq x)\sim K mn e^{-\lambda x}$. *$K$ and λ are tabulated for different matrices ******

The End of Statistics (for now)

- **Two ways to get the parameters:**

1- For many amino acid substitution matrices, Altschul and Gish have tabulated their score distribution for 10,000 random amino acid sequences using various gap penalties

2- Even better! Calculate the distribution for the two sequences you are aligning by keeping one of them fixed and scrambling the other one – this preserves BOTH sequence length and amino acid composition!

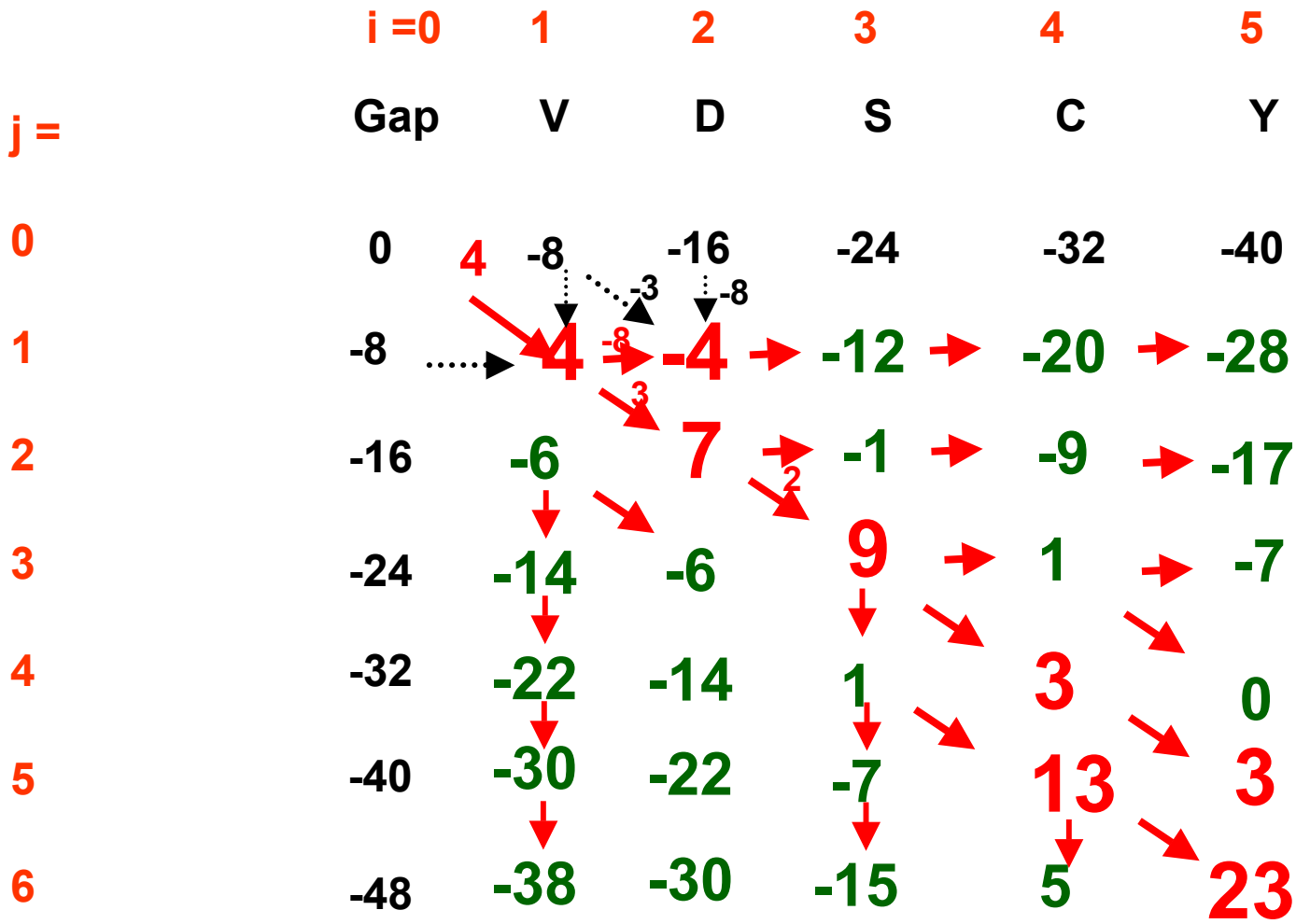
Multiple Sequence Alignments

- Sequences are aligned so as to bring the greatest number of single characters into register.
- If we include gaps, mismatches, then even dynamic programming becomes limited to ~ 3 sequences unless they are very short….need an alternative approach…

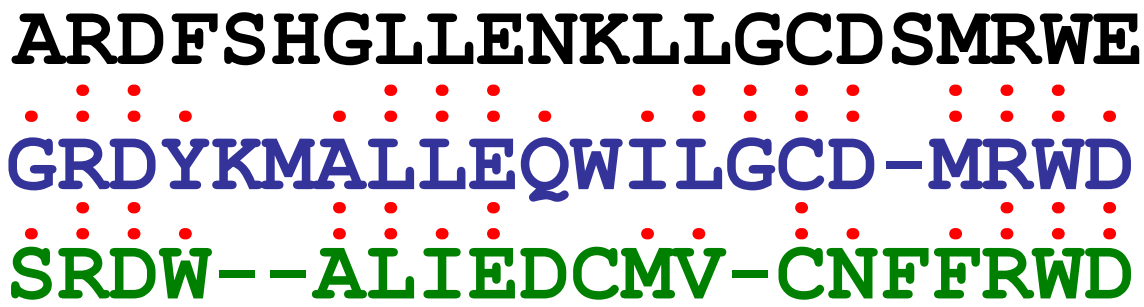
Why?

Consider the 2 sequence comparison

…..an O(mn) problem – order n^2



For 3 sequences….



An O(mnj) problem !

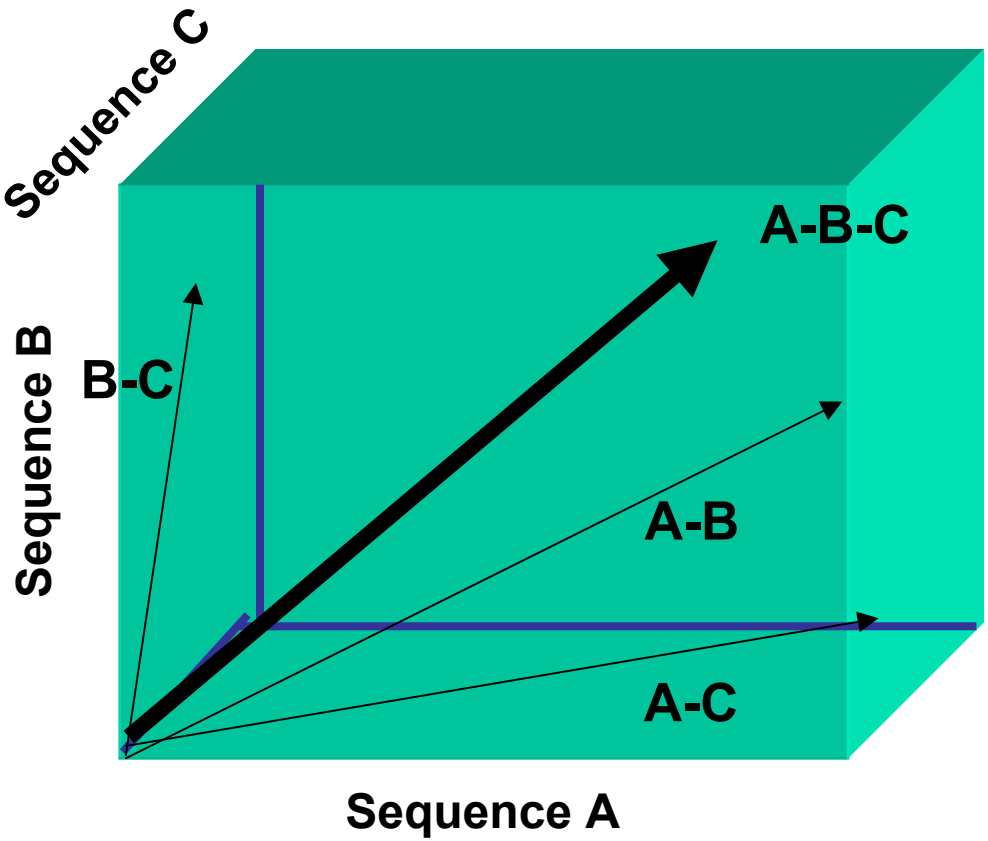
Consider sequences each 300 amino acids

2 sequences – $(300)^2$
3 sequences – $(300)^3$
but for v sequences – $(300)^v$

Uh Oh !!!
Our polynomail problem
Just became exponential!

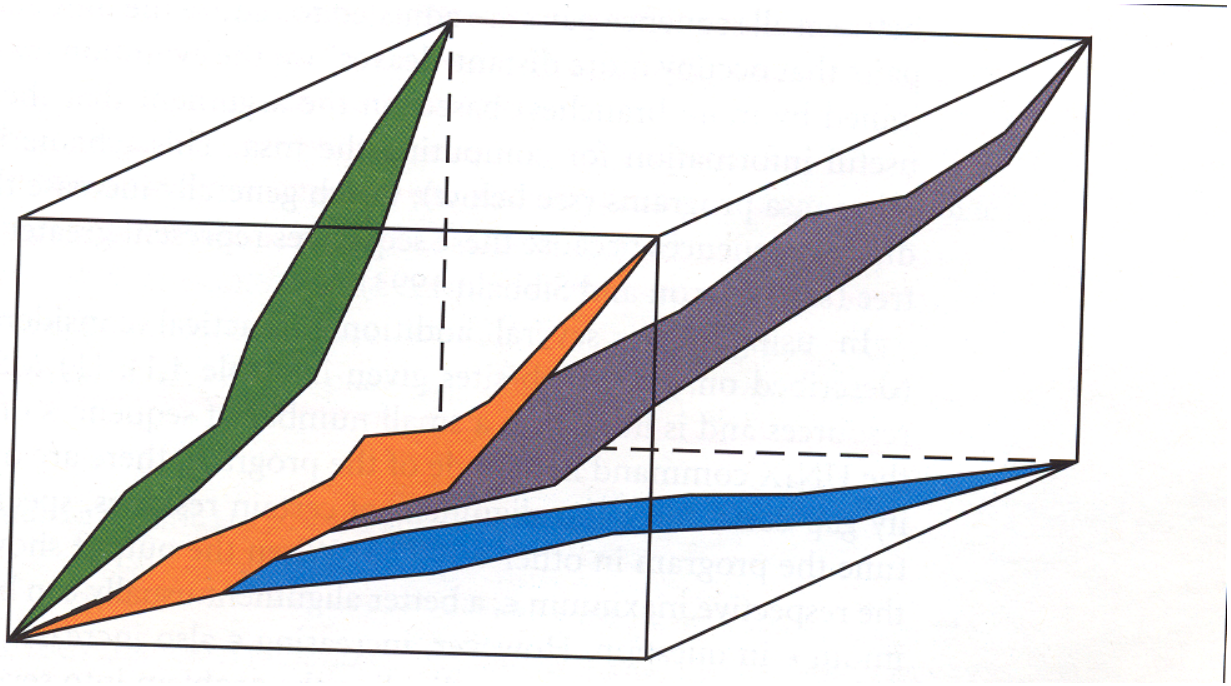
Consider pairwise alignments between 3 sequences

Carillo and Lipman – Sum of Pairs method



Do we need to Score each node?

Get the multiple alignment score within the cubic lattice by Adding together the scores of the pairwise alignments…

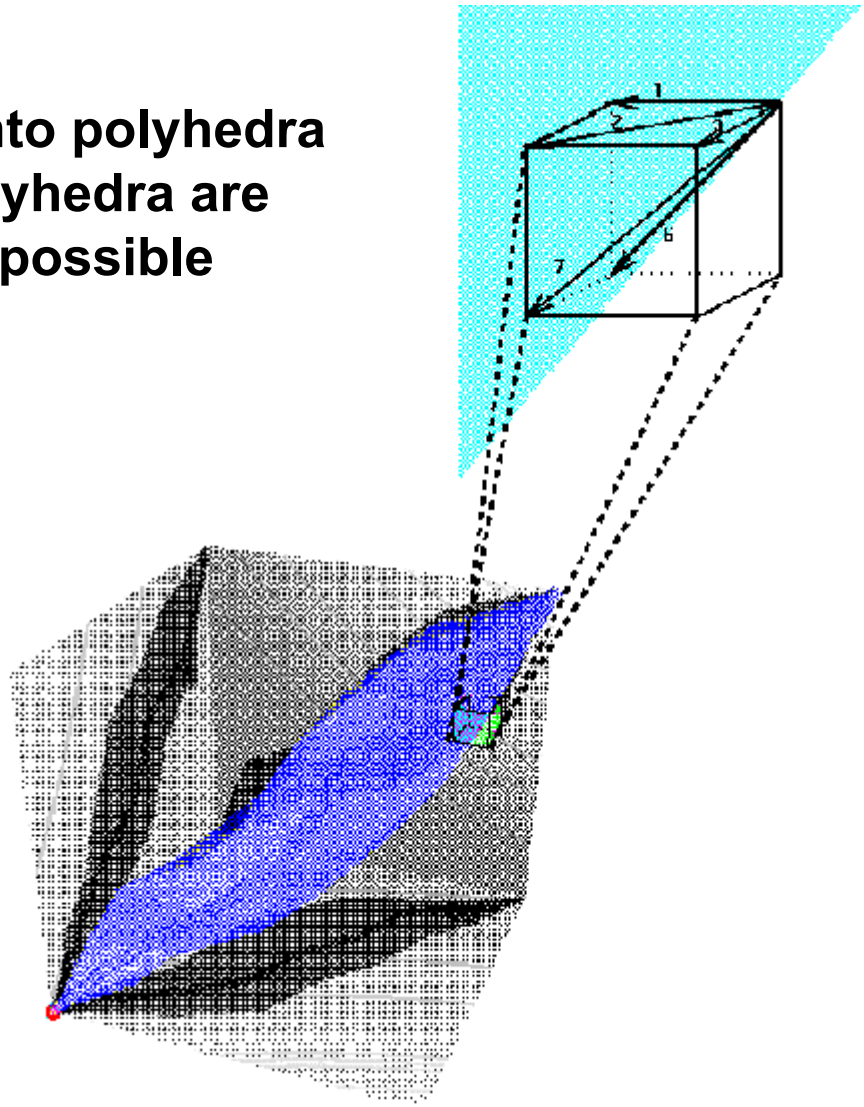


In practice, doesn't give optimal alignment…But we're close!

Seems reasonable that the optimal alignment won't be far from the diagonal we were on…so we just set bounds on the location of the msa within the cube based on each pairwise-alignment.

Then just do dynamic programing within the volume defined by the pre-imposed bounds

….the volume is broken into polyhedra and the borders of the polyhedra are defined by paths through possible alignments



Still takes too long for more than three sequences…need a better way!

- **Progressive Methods of Multiple Sequence Alignment**

Concept – simple:

1-Use DP to build pairwise alignments of most closely related sequences

2- Then progressively add less related sequences or groups of sequences…

ClustalW

Higgins and Sharp 1988

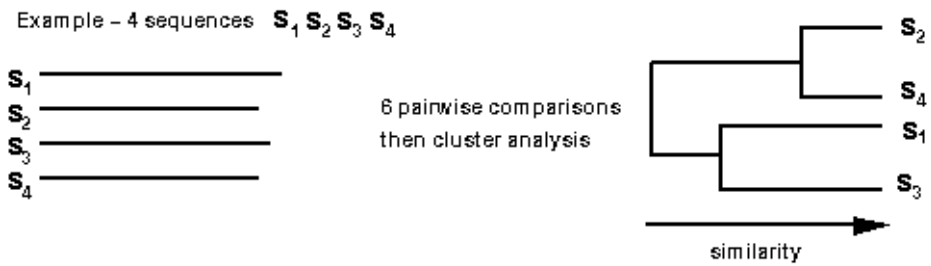
- 1- **Do pairwise analysis of all the sequences (you choose similarity matrix).**
- **2- Use the alignment scores to make a phylogenetic tree.**
- **3- Align the sequences to each other guided by the phylogenetic relationships in the tree.**

New features: Clustal →ClustalW (allows weights)→ClustalX (GUI-based)

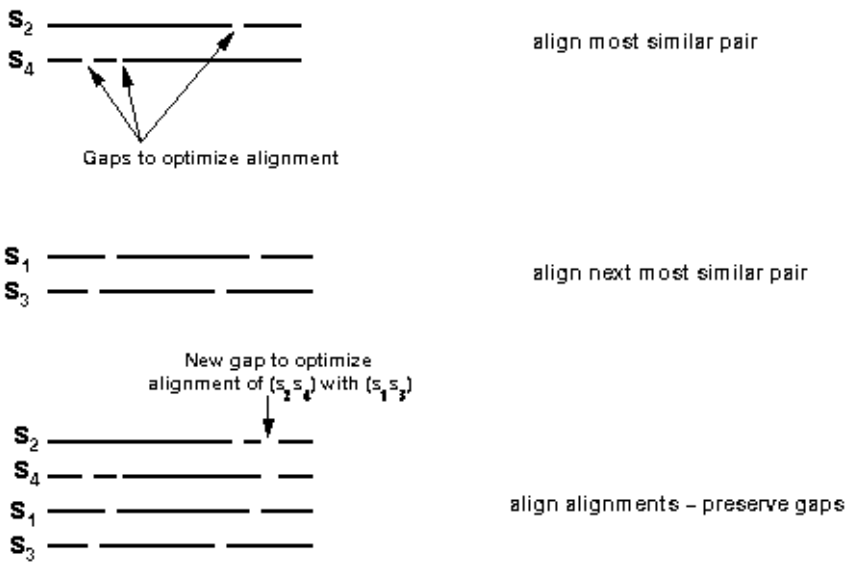
Weighting is important to avoid biasing an alignment by many sequence Members that are closely related to each other evolutionarily!

Steps in Multiple Alignment

(A) Pairwise Alignment



(B) Multiple alignment following the tree from A



Progressive Alignments

Note that the final msa is EXTREMELY DEPENDENT on the initial pairwise sequence alignments!

If the sequences are close in evolution, and you can see the alignment – GREAT!

If the sequences are NOT close in evolution, and you CANNOT See the alignment – errors will be propogated to the final msa

Has led to other approaches to do msa's that aren't so Dependent on initial states….i.e. genetic algorithm

Finding patterns (i.e. motifs and domains) in Multiple Sequence Analysis

---

Block Analysis, Position Specific Scoring Matrices (PSSM)

BUILD an msa from groups of related proteins

BLOCKS represent a conserved region in that msa that is LACKING IN GAPS – i.e. no insertions/deletions

The BLOCKS are typically anwhere from 3-60 amino acids long, based on exact amino acid matches – i.e. alignment will tolerate mismatches, but doesn't use any kind of PAM or BLOSUM matrix…in fact they generate the BLOSUM matrix!

A single proteins contain numerous such BLOCKS separated by stretches of intervening sequences that can differ in length and composition.

These blocks may be whole domains, short sequence motifs, key parts of enzyme active sites etc, etc.
BLOCKS database….so far exploration limited. Lots of stuff to probe!

Can use these conserved BLOCKS to derive a PSSM

- **The dirty secret behind prosite! Scansite! And in a twised way Psi-BLAST!**

```
   12345……..11
….GDSFHYFVSHG….
….GDAFHYYISFG….
….GDSYHYFLSFG….
….SDSFHYFMSFG….
….GDSFHFFASFG….
```

Now build a matrix with 20 amino acids as the columns, and 11 rows For the positions in the BLOCK

Each matrix entry is the log(frequency of the amino acid occurance) at that position in the BLOCK.

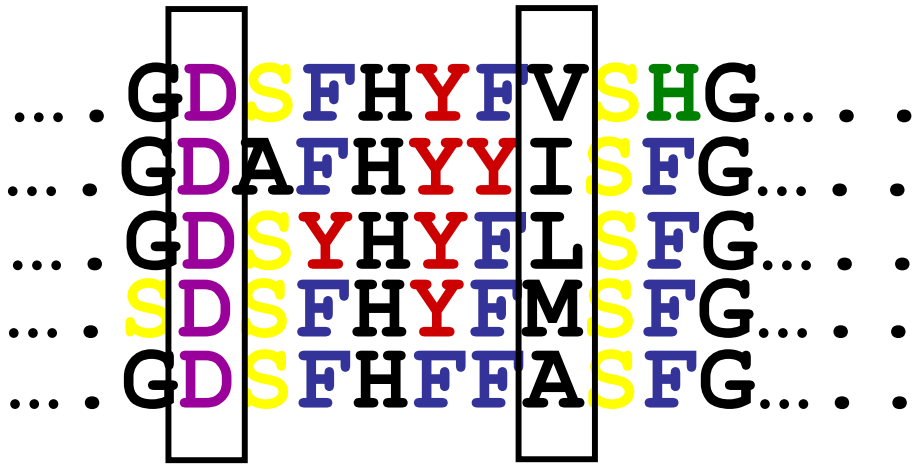
```
....GDSFHYFVSHG....
....GDAFHYYISFG....
....GDSYHYFLSFG....
....SDSFHYFMSFG....
....GDSFHFFASFG....
```

	A	C	D	E	F	G	H	I	K....
1						Log(4)			
2			Log(5)						
3									
4									
5									

We can now use the PSSM to search a database for other proteins that have the BLOCK (or motif).

Problem 1 – We need to think about what kind of information is Contained within the PSSM.

→Leads to concepts of Information Content & Entropy (next time)



Problem 2 –The PSSM must accurately represent the expected BLOCK Or motif….and we have only limited amounts of data! Is it a good statistical Sampling of the BLOCK/motif? Is it too narrow because of small dataset? Should we broaden it by adding extra amino acids that we choose using Some type of randomization scheme (called adding pseudocounts). If so, How many should we add?