Outline

- Multiple sequence alignment (MSA)
- Introduction to MSA
- Methods of MSA
  - Progressive global alignment
  - Iterative methods
  - Alignments based on locally conserved patterns

Motivation

- Similar genes can be conserved across species that perform similar or identical functions.
- Many genes are represented in highly conserved forms across organisms.
- By performing a simultaneous alignment of multiple sequences having similar or identical functions
  - we can gain information about
    - which regions have been subject to mutations over evolutionary time
    - and which are evolutionarily conserved.
  - Such knowledge tells
    - which regions or domains of a gene are critical to its functionality.
- Sometimes genes that are similar in sequence can be mutated or rearranged to perform an altered function.
  - By looking at multiple alignments of such sequences,
    - we can tell which changes in the sequence have caused a change in the functionality.

Multiple sequence alignment

- a collection of three or more protein (or nucleic acid) sequences that are partially or completely aligned.
- Homologous residues are aligned in columns across the length of the sequences.
  - These aligned residues are homologous in an evolutionary sense:
    - they are presumably derived from a common ancestor.
  - The residues in each column are also presumed to be homologous in a structural sense:
    - aligned residues tend to occupy corresponding positions in the three-dimensional structure of each aligned protein.

Multiple sequence alignment

- Multiple sequence alignment yields
  - information concerning the structure and function of proteins,
  - can help lead to the discovery of important sequence domains or motifs with biological significance
    - while at the same time uncovering evolutionary relationships among genes.
- In multiple sequence alignment, the idea is
  - to take three or more sequences, and align them
  - so that the greatest number of similar characters are aligned in the same column of the alignment.

Multiple sequence alignment

- The difficulty with multiple sequence alignment is that
  - now there are a number of different combinations of
    - matches,
    - insertions,
    - and deletions
  - that must be considered when looking at several different sequences.
- Methods to guarantee the highest scoring alignment are not feasible.
- Therefore, approximation methods are put to use in multiple sequence alignment.

When and why are multiple sequence alignments used?

- If a protein (or gene) you are studying is related to a larger group of proteins, this group membership can often provide insight into the likely function, structure, and evolution of that protein.
- Most protein families have distantly related members.
  - Multiple sequence alignment is a far more sensitive method than pairwise alignment to detect homologs
- When the output of any database search (such as a BLAST search) is examined, a multiple sequence alignment format can be extremely useful to reveal conserved residues or motifs in the output.

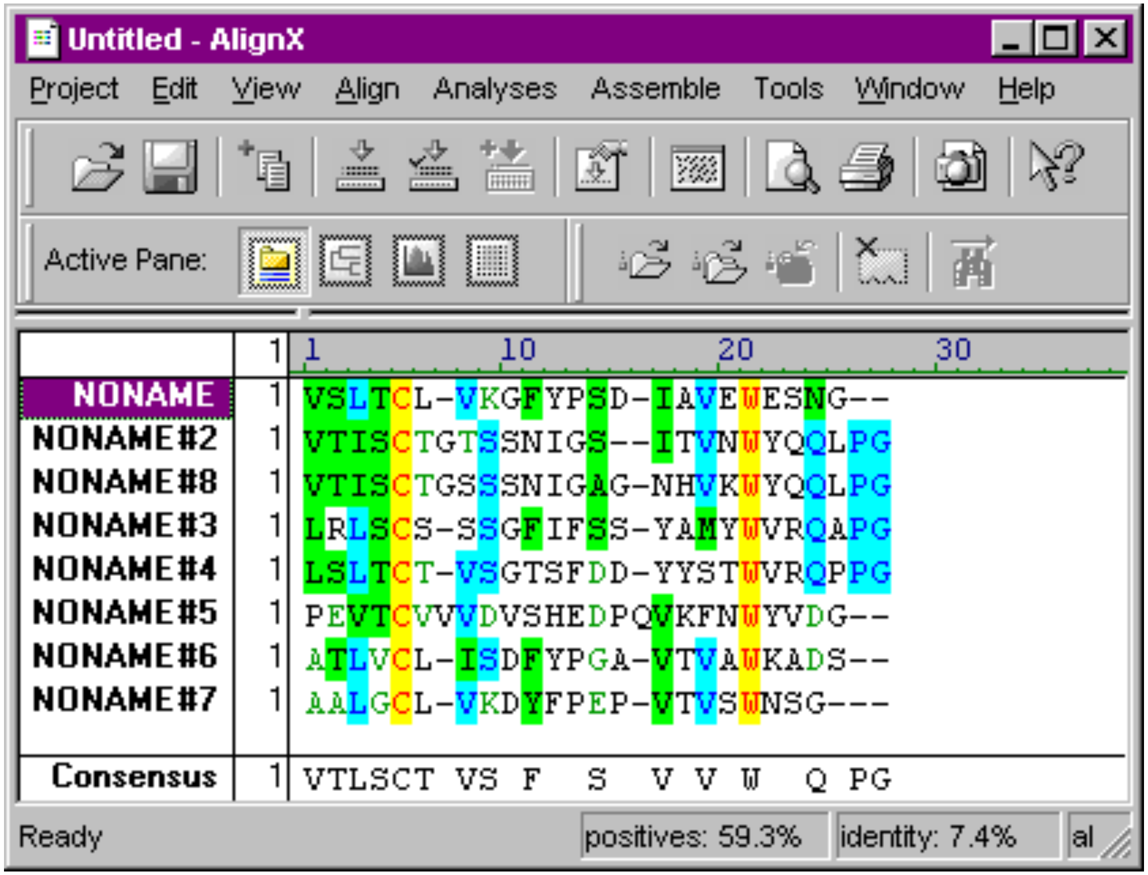
When and why are multiple sequence alignments used?

- Each human genome harbors ~11,000 nonsynonymous single-nucleotide variants (causing an amino acid substitution) of which ~300 are predicted to be deleterious.
  - Algorithms that predict whether variants are harmful often rely on DNA and/or protein multiple sequence alignments to assess cross-species conservation.
    - Deleterious variants tend to occur at more conserved positions.
- Analysis of population data can provide insight into many biological questions involving evolution, structure, and function.

When and why are multiple sequence alignments used?

- When the complete genome of any organism is sequenced, a major portion of the analysis consists of defining the protein families to which all the gene products belong.
- Database searches effectively perform multiple sequence alignments, allowing comparisons of each novel protein (or gene) to the families of all other known genes.
- The most critical part of making a phylogenetic tree is to produce an optimal multiple sequence alignment.
- The regulatory regions of many genes contain consensus sequences for transcription factor-binding sites and other conserved elements.
  - Many such regions are identified based on conserved noncoding sequences that are detected using multiple sequence alignment.

Example Multiple Alignment



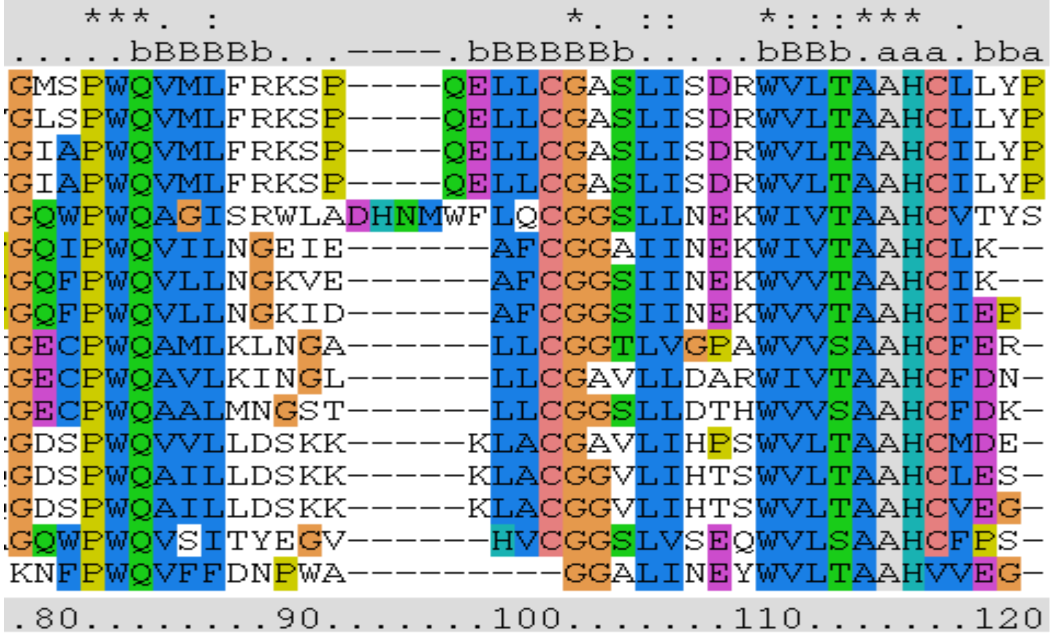
```
             1  1         10        20        30
NONAME       1  VSLTCL-VKGFYPSD-IAVEWESNG--
NONAME#2     1  VTISCTGTSSNIGS--ITVNWYQQLPG
NONAME#8     1  VTISCTGSSSNIGAG-NHVKWYQQLPG
NONAME#3     1  LRLSCS-SSGFIFSS-YAMYWVRQAPG
NONAME#4     1  LSLTCT-VSGTSFDD-YYSTWVRQPPG
NONAME#5     1  PEVTCVVVDVSHEDPQVKFNWYVDG--
NONAME#6     1  ATLVCL-ISDFYPGA-VTVAWKADS--
NONAME#7     1  AALGCL-VKDYFPEP-VTVSWNSG---

Consensus    1  VTLSCT VS F  S  V V W  Q PG

Ready                    positives: 59.3%   identity: 7.4%   al
```

- Example multiple alignment of 8 immunoglobulin sequences.

Example MSA



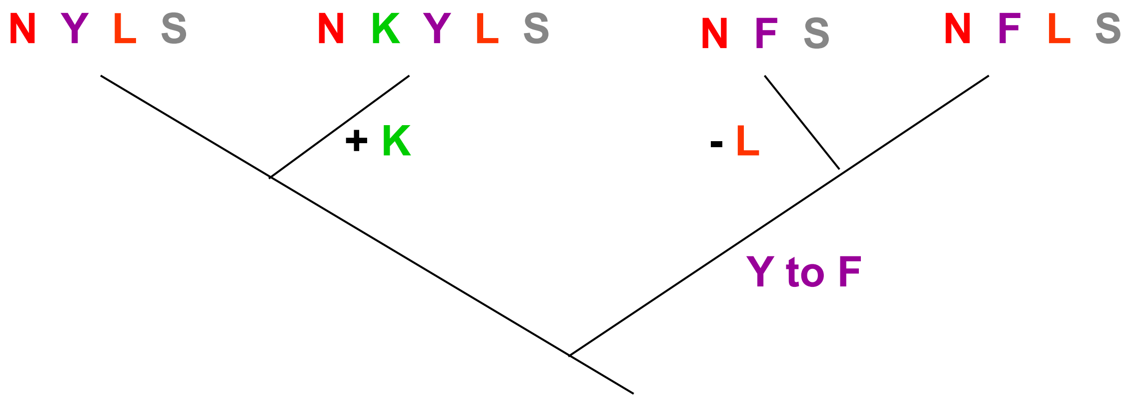
```
   ***. :              *. ::   *:::*** .
.....bBBBBb...----.bBBBBBb.....bBBb.aaa.bba
GMSPWQVMLFRKSP----QELLCGASLISDRWVLTAAHCLLYP
GLSPWQVMLFRKSP----QELLCGASLISDRWVLTAAHCLLYP
GIAPWQVMLFRKSP----QELLCGASLISDRWVLTAAHCILYP
GIAPWQVMLFRKSP----QELLCGASLISDRWVLTAAHCILYP
GQWPWQAGISRWLADHNMWFLQCGGSLLNEKWIVTAAHCVTYS
GQIPWQVILNGEIE------AFCGGAIINEKWIVTAAHCLK--
GQFPWQVLLNGKVE------AFCGGSIINEKWVVTAAHCIK--
GQFPWQVLLNGKID------AFCGGSIINEKWVVTAAHCIEP-
GECPWQAMLKLNGA------LLCGGTLVGPAWVVSAAHCFER-
GECPWQAVLKINGL------LLCGAVLLDARWIVTAAHCFDN-
GECPWQAALMNGST------LLCGGSLLDTHWVVSAAHCFDK-
GDSPWQVVLLDSKK-----KLACGAVLIHPSWVLTAAHCMDE-
GDSPWQAILLDSKK-----KLACGGVLIHTSWVLTAAHCLES-
GDSPWQAILLDSKK-----KLACGGVLIHTSWVLTAAHCVEG-
GQWPWQVSITYEGV------HVCGGSLVSEQWVLSAAHCFPS-
KNFPWQVFFDNPWA---------GGALINEYWVLTAAHVVEG-
.80.........90........100.......110.......120
```

- Each row is a different protein sequence
- Each column is a different aligned position

Relationship of MSA to Phylogenetic analysis

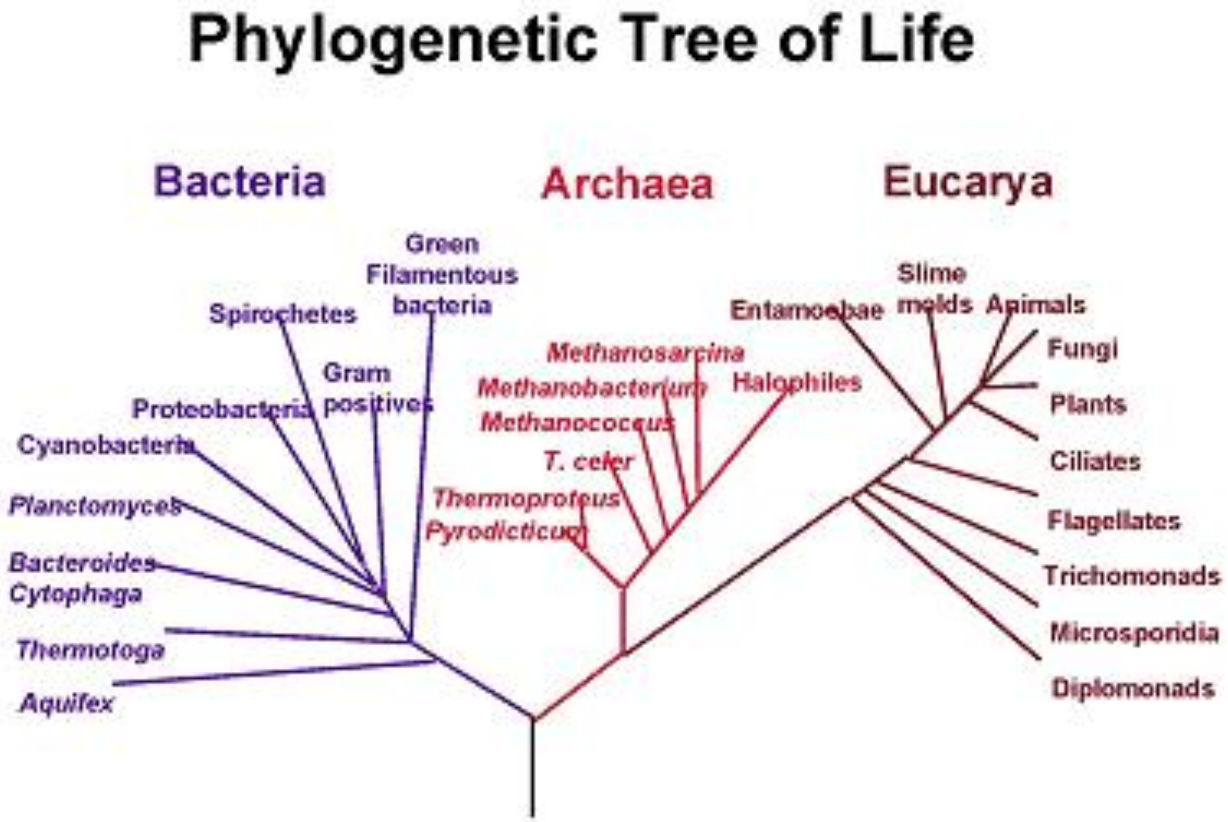
once the msa has been found, the number or types of changes in the aligned sequences may be used for a phylogenetic analysis

```
seqA        N  -  F  L  S
seqB        N  -  F  -  S
seqC        N  K  Y  L  S
seqD        N  -  Y  L  S
```

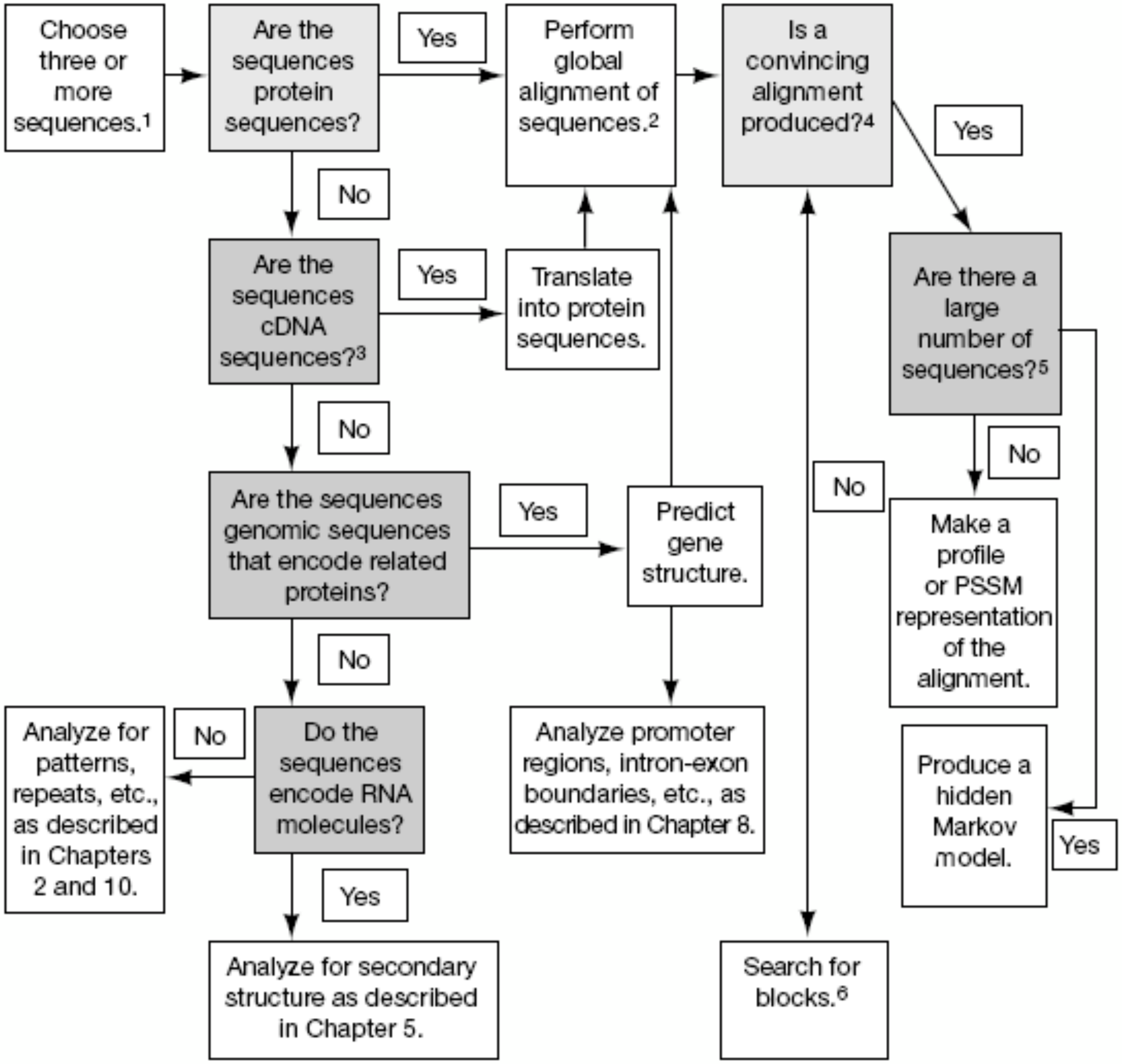


hypothetical evolutionary tree that could have generated three sequence changes

Phylogenetic analysis



MSA method


Choose three or more sequences.[1]
Are the sequences protein sequences?
Yes
Perform global alignment of sequences.[2]
Is a convincing alignment produced?[4]
Yes
No
Are the sequences cDNA sequences?[3]
Yes
Translate into protein sequences.
Are there a large number of sequences?[5]
No
No
No
Are the sequences genomic sequences that encode related proteins?
Yes
Predict gene structure.
Make a profile or PSSM representation of the alignment.
No
Analyze for patterns, repeats, etc., as described in Chapters 2 and 10.
No
Do the sequences encode RNA molecules?
Analyze promoter regions, intron-exon boundaries, etc., as described in Chapter 8.
Produce a hidden Markov model.
Yes
Yes
Analyze for secondary structure as described in Chapter 5.
Search for blocks.[6]

Approaches to Multiple Alignment

- There are many approaches to multiple sequence alignment;
  - in the past decade many dozens of programs have been introduced
    - https://doi.org/10.1093/bib%2F6.1.6
- We will consider four approaches to multiple sequence alignment:
  - Exact methods (Dynamic Programming)
  - Progressive Alignment
  - Iterative Alignment
  - Statistical Modeling

Dynamic Programming Approach

- Exact methods of multiple alignment use dynamic programming
  - guaranteed to find optimal solutions.
- Dynamic programming with two sequences
  - Relatively easy to code
  - Guaranteed to obtain optimal alignment
- Can this be extended to multiple sequences?

Dynamic Programming With 3 Sequences

- Consider the following amino acid sequences to align

  VSNS, SNA, AS

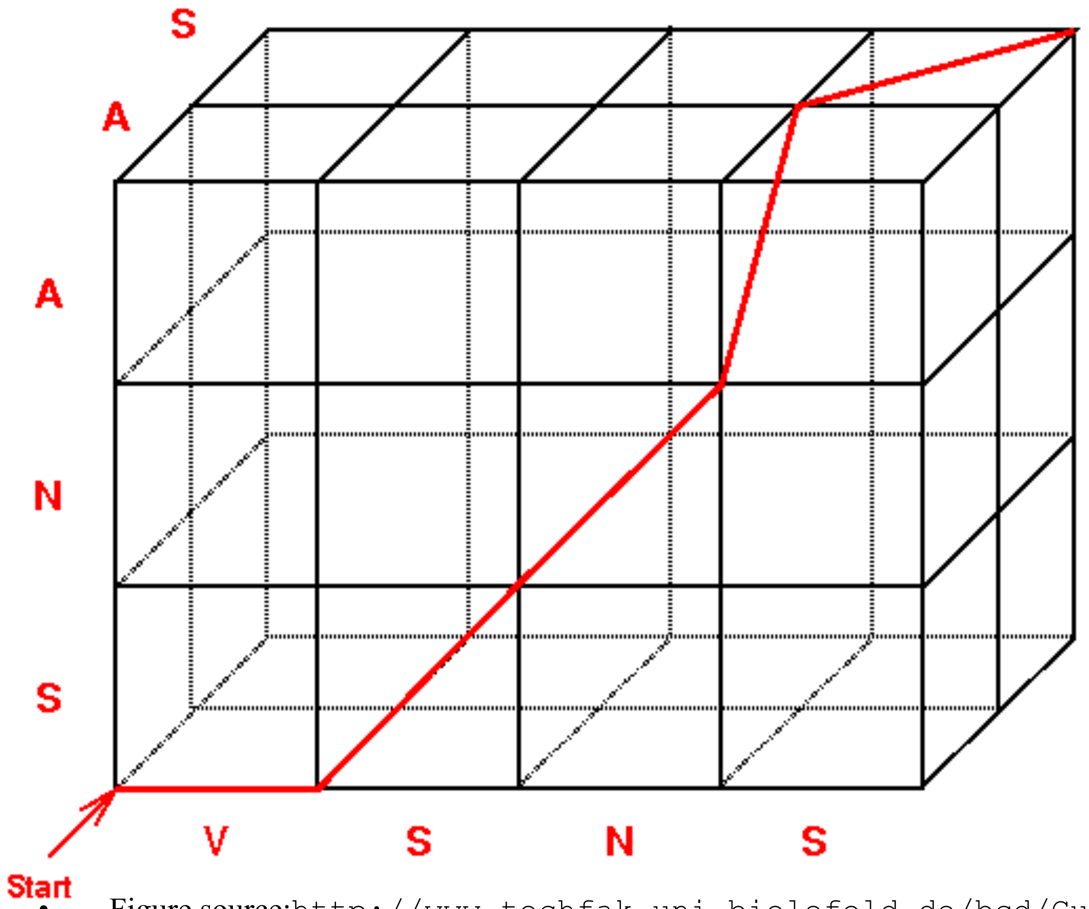
- Instead of filling a two dimensional matrix as we did with two sequences,
  - we now fill a three dimensional space.
- Put one sequence per axis (x, y, z)
- Three dimensional structure results

Dynamic Programming With 3 Sequences

Possibilities:

- All three match;
- A & B match with gap in C
- A & C match with gap in B
- B & C match with gap in A
- A with gap in B & C
- B with gap in A & C
- C with gap in A & B

Dynamic Programming With 3 Sequences



V	S	N	_	S
_	S	N	A	_
_	_	_	A	S

- Figure source:http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/node2.html#SECTION00020000000000000000

Dynamic Programming

- Suppose the length of each sequence is n residues.
- If there are two such sequences,
  - then the number of comparisons needed to fill in the scoring matrix
    - is n^2,
    - since it is a two-dimensional matrix.
- The number of comparisons needed to fill in the scoring cube
  - when three sequences are aligned
    - is n^3,
  - and when four sequences are aligned,
    - is n^4.

- Thus, as the number of sequences increases,
  - the number of comparisons needed increases exponentially, i.e. n^N
    - where n is the length of the sequences,
    - and N is the number of sequences.
- Thus, without any changes to the dynamic programming approach, this becomes impractical for even a small number of short sequences rather quickly.

Example

2 protein sequences length = 300, excluding gaps

number of comparisons by dynamic programming

$$300^2 = 9 * 10^4$$

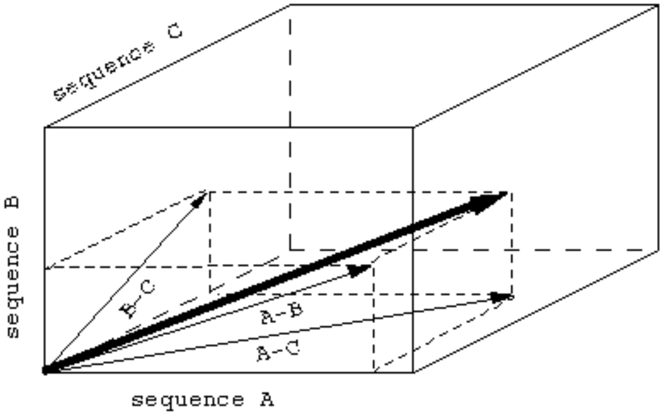
3 protein sequences length = 300, excluding gaps

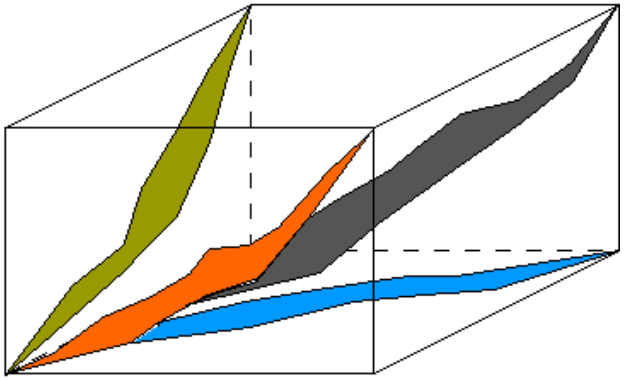
number of comparisons by dynamic programming

$$300^3 = 2.7 * 10^7$$

Reduction of space and time

- Carrillo and Lipman:
  - multiple sequence alignment space bounded by pairwise alignments
  - Projections of these alignments lead to a bounded alignments
    - Following figures show how the two dimensional search spaces can be projected into a three dimensional volume that can be searched





Reduction of space and time

- Idea for reduction of memory and computations:
  - Multiple sequence alignment imposes an alignment on each of the pairs of sequences.
- Alignments found for each of the pairs of sequences can impose bounds on the location of the MSA within the cube (three sequences) or N-dimensional space (N sequences).
  - Step 1
    - Find pairwise alignment for sequences.
  - Step 2
    - Trial msa produced by predicting a phylogenetic tree for the sequences
  - Step 3
    - Sequences multiply aligned in the order of their relationship on the tree

Reduction of space and time

- This is a heuristic alignment

- Therefore the alignment is not guaranteed to be optimal

- Alignment provides a limit to the volume within which optimal alignments are likely to be found

MSA

- MSA: Developed by Lipman, 1989
- Incorporates extended dynamic programming
- MSA calculates the multiple alignment score within the lattice by adding the scores of the corresponding pairwise alignments in the multiple sequence alignment.
- This measure is known as the sum of pairs (SP) measure.
- The optimal alignment is based on the best SP score.

Scoring of msa’s

- MSA uses Sum of Pairs (SP)
  - Scores of pair-wise alignments in each column added together
  - Columns can be weighted to reduce influence of closely related sequences
  - Weight is determined by distance in phylogenetic tree

Sum of Pairs Method

- The sum of pairs method scores all possible combinations of pairs of residues in a column of a multiple sequence alignment.
- For instance, consider the alignment

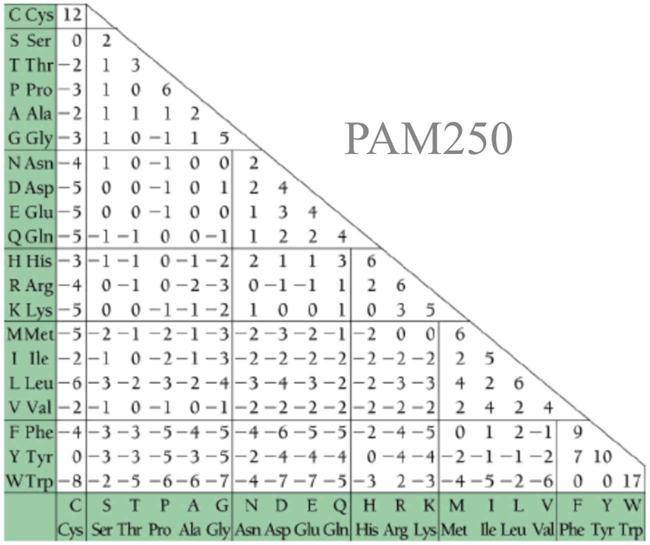
E	C	S	Q	(1)
S	N	S	G	(2)
S	W	K	N	(3)
S	C	S	N	(4)

- Since there are four sequences,
  - there will be six different alignments to consider for each column.
- The alignments, listed by the sequence number are listed as follows:

  (1)-(2); (1)-(3); (1)-(4); (2)-(3); (2)-(4); (3)-(4)

Sum of Pairs Method

E	C	S	Q	(1)
S	N	S	G	(2)
S	W	K	N	(3)
S	C	S	N	(4)



PAM250

C Cys	12																			
S Ser	0	2																		
T Thr	−2	1	3																	
P Pro	−3	1	0	6																
A Ala	−2	1	1	1	2															
G Gly	−3	1	0	−1	1	5														
N Asn	−4	1	0	−1	0	0	2													
D Asp	−5	0	0	−1	0	1	2	4												
E Glu	−5	0	0	−1	0	0	1	3	4											
Q Gln	−5	−1	−1	0	0	−1	1	2	2	4										
H His	−3	−1	−1	0	−1	−2	2	1	1	3	6									
R Arg	−4	0	−1	0	−2	−3	0	−1	−1	1	2	6								
K Lys	−5	0	0	−1	−1	−2	1	0	0	1	0	3	5							
M Met	−5	−2	−1	−2	−1	−3	−2	−3	−2	−1	−2	0	0	6						
I Ile	−2	−1	0	−2	−1	−3	−2	−2	−2	−2	−2	−2	−2	2	5					
L Leu	−6	−3	−2	−3	−2	−4	−3	−4	−3	−2	−2	−3	−3	4	2	6				
V Val	−2	−1	0	−1	0	−1	−2	−2	−2	−2	−2	−2	−2	2	4	2	4			
F Phe	−4	−3	−3	−5	−4	−5	−4	−6	−5	−5	−2	−4	−5	0	1	2	−1	9		
Y Tyr	0	−3	−3	−5	−3	−5	−2	−4	−4	−4	0	−4	−4	−2	−1	−1	−2	7	10	
W Trp	−8	−2	−5	−6	−6	−7	−4	−7	−7	−5	−3	2	−3	−4	−5	−2	−6	0	0	17
	C Cys	S Ser	T Thr	P Pro	A Ala	G Gly	N Asn	D Asp	E Glu	Q Gln	H His	R Arg	K Lys	M Met	I Ile	L Leu	V Val	F Phe	Y Tyr	W Trp

	Residues	Score	Residues	Score	Residues	Score	Residues	Score
1-2	E-S	0	C-N	-4	S-S	2	Q-G	-1
1-3	E-S	0	C-W	-8	S-K	0	Q-N	1
1-4	E-S	0	C-C	12	S-S	2	Q-N	1
2-3	S-S	2	N-W	-4	S-K	0	G-N	0
2-4	S-S	2	N-C	-4	S-S	2	G-N	0
3-4	S-S	2	W-C	-8	K-S	0	N-N	2
		6		-16		6		3

- Problem with this approach:
  - more closely related sequences will have a higher weight
- The MSA program gets around this by calculating weights to associate to each sequence alignment pair.
- The weights are assigned based on the predicted tree of the aligned sequences.

Summary of MSA

1. Calculate all pairwise alignment scores
2. Use the scores to predict tree
3. Calculate pair weights based on the tree
4. Produce a heuristic msa based on the tree
5. Calculate the maximum weight for each sequence pair
6. Determine the spatial positions that must be calculated to obtain the optimal alignment
7. Perform the optimal alignment
8. Report the weight found compared to the maximum weight previously found

Progressive Alignments

- MSA program is limited in size

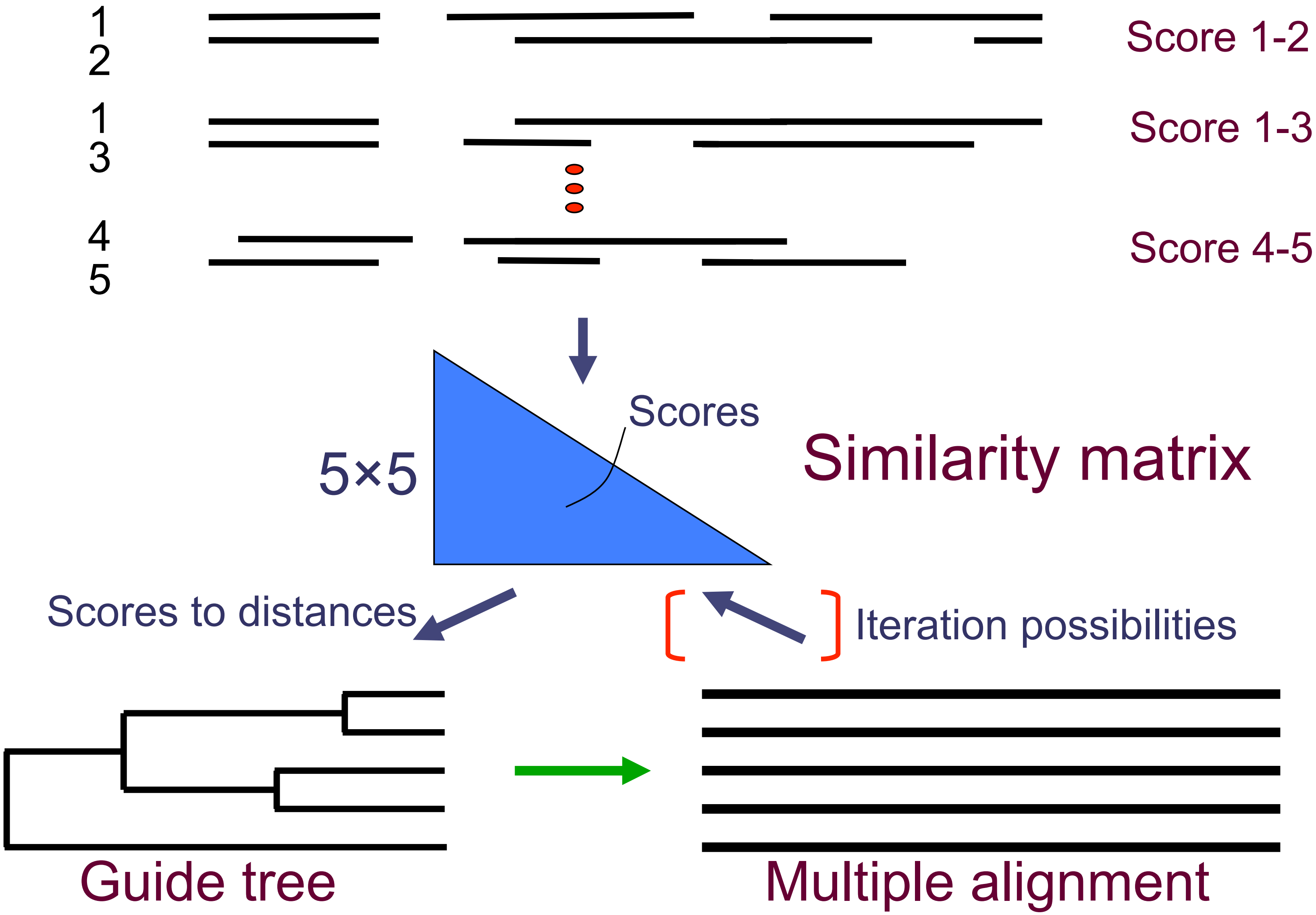
- The approach of progressive alignment is
  - to begin with an alignment of the most alike sequences,
  - and then build upon the alignment using other sequences.

- Progressive alignments work by
  - first aligning the most alike sequences using dynamic programming,
  - and then progressively adding less related sequences to the initial alignment.

Progressive multiple sequence alignment

- alignment on each of the pairs of sequences
- next, trail msa is produced by first predicting a phylogenetic tree for the sequences
- sequences are then multiply aligned in order of their relationship on the tree
  - starting with the most related sequences
  - then progressively adding less related sequences to the initial alignment
- used by PILEUP and CLUSTALW
- not guaranteed to be optimal

Progressive msa - general principles



1
2

Score 1-2

1
3

Score 1-3

4
5

Score 4-5

5×5

Scores

Similarity matrix

Scores to distances

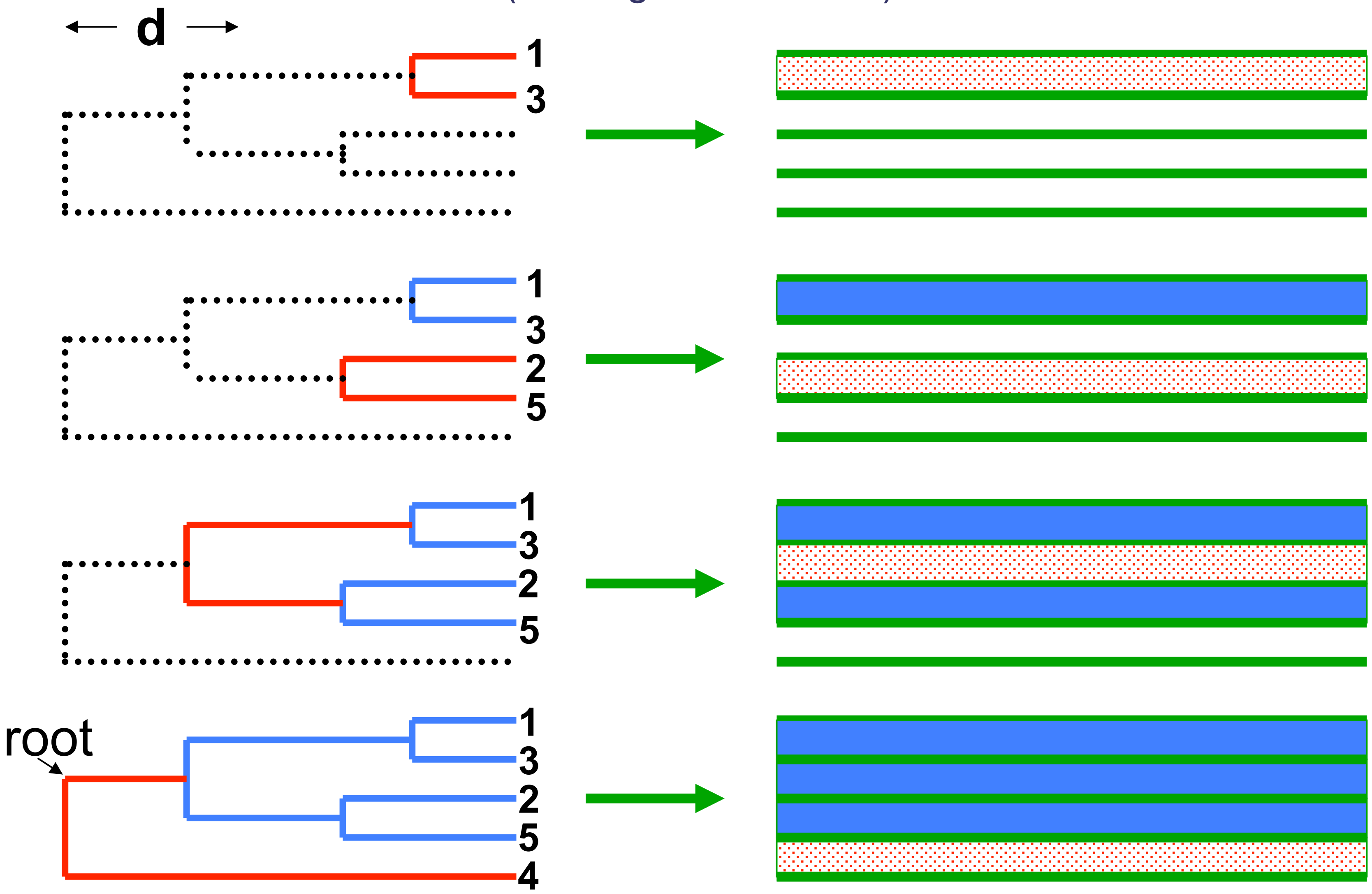
Iteration possibilities

Guide tree

Multiple alignment

General progressive msa technique

(follow generated tree)



CLUSTALW and CLUSTALX

- CLUSTALW and CLUSTALX are progressive alignment programs that follow the following steps:
  - Perform pairwise alignments of all of the sequences
  - Use the alignment scores to produces a phylogenetic tree using neighbor-joining methods
  - Align the sequences sequentially, guided by the phylogenetic relationships indicated by the tree

CLUSTALW

- The initial pairwise alignments are calculated using an enhanced dynamic programming algorithm, and

- the genetic distances used to create the phylogenetic tree are calculated by dividing the total number of mismatched positions by the total number of matched positions.

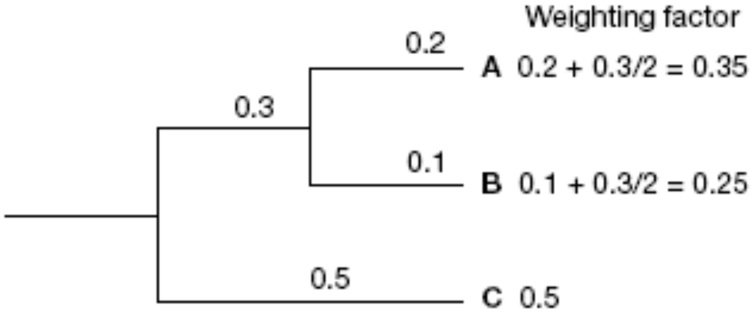
CLUSTALW

- Alignments are associated a weight based on their distance from the root node (next slide)

- Gaps are added to an existing profile in progressive methods

- CLUSTALW incorporates a statistical model in order to place gaps where they are most likely to occur

A. Calculation of sequence weights



B. Use of sequence weights

Column in alignment 1

```
Sequence A (weight a)     .........K.........
Sequence B (weight b)     .........I.........
```

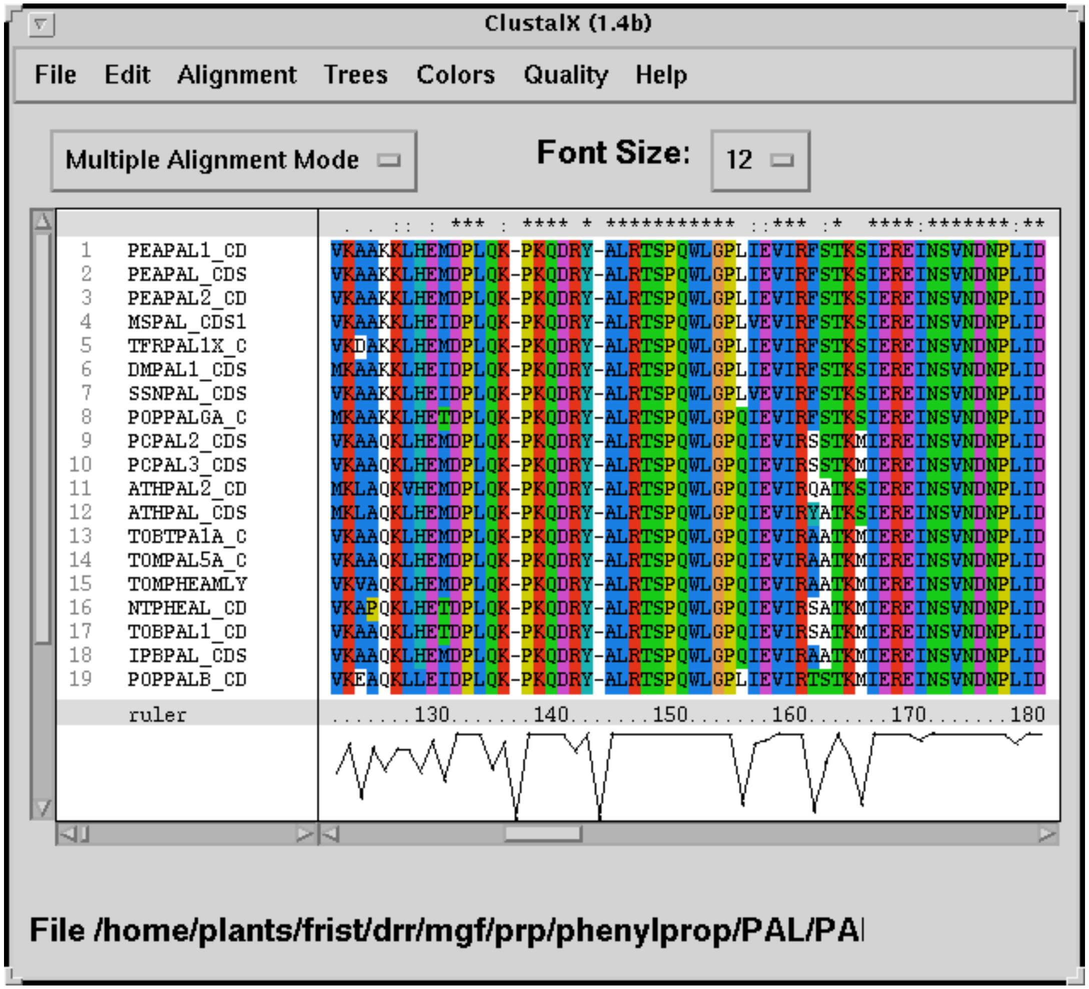
Column in alignment 2

```
Sequence C (weight c)     .........L.........
Sequence D (weight d)     .........V.........
```

Score for matching these two column in an msa =

[a x c x score (K,L) +
a x d x score (K,V) +
b x c x score (I,L) +
b x d x score (I,V)] / 4

CLUSTALW / CLUSTALX



```
                 . . :: : *** : **** * *********** ::*** :*  ****:*******:**
 1  PEAPAL1_CD   VKAAKKLHEMDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRFSTKSIEREINSVNDNPLID
 2  PEAPAL_CDS   VKAAKKLHEMDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRFSTKSIEREINSVNDNPLID
 3  PEAPAL2_CD   VKAAKKLHEMDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRFSTKSIEREINSVNDNPLID
 4  MSPAL_CDS1   VKAAKKLHEIDPLQK-PKQDRY-ALRTSPQWLGPLVEVIRFSTKSIEREINSVNDNPLID
 5  TFRPAL1X_C   VKDAKKLHEMDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRFSTKSIEREINSVNDNPLID
 6  DMPAL1_CDS   MKAAKKLHEIDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRFSTKSIEREINSVNDNPLID
 7  SSNPAL_CDS   VKAAKKLHEIDPLQK-PKQDRY-ALRTSPQWLGPLVEVIRFSTKSIEREINSVNDNPLID
 8  POPPALGA_C   MKAAKKLHETDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRFSTKSIEREINSVNDNPLID
 9  PCPAL2_CDS   VKAAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRSSTKMIEREINSVNDNPLID
10  PCPAL3_CDS   VKAAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRSSTKMIEREINSVNDNPLID
11  ATHPAL2_CD   MKLAQKVHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRQATKSIEREINSVNDNPLID
12  ATHPAL_CDS   MKLAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRYATKSIEREINSVNDNPLID
13  TOBTPA1A_C   VKAAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRAATKMIEREINSVNDNPLID
14  TOMPAL5A_C   VKAAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRAATKMIEREINSVNDNPLID
15  TOMPHEAMLY   VKVAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRAATKMIEREINSVNDNPLID
16  NTPHEAL_CD   VKAPQKLHETDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRSATKMIEREINSVNDNPLID
17  TOBPAL1_CD   VKAAQKLHETDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRSATKMIEREINSVNDNPLID
18  IPBPAL_CDS   VKAAQKLHEMDPLQK-PKQDRY-ALRTSPQWLGPQIEVIRAATKMIEREINSVNDNPLID
19  POPPALB_CD   VKEAQKLLEIDPLQK-PKQDRY-ALRTSPQWLGPLIEVIRTSTKMIEREINSVNDNPLID
    ruler        .......130.......140.......150.......160.......170.......180
```

File /home/plants/frist/drr/mgf/prp/phenylprop/PAL/PA|

- 'W' stands for "weighting"
  - ability to provide weights to sequence and program parameters
- CLUSTALX – with graphical interface
- provides global msa
- Not constructed to perform local alignments.
- Similarity in small regions is a problem.
- Problems with large insertions.
- Problems with repetitive elements, such as domains.
- ClustalW does not guarantee an optimal solution

CLUSTALW

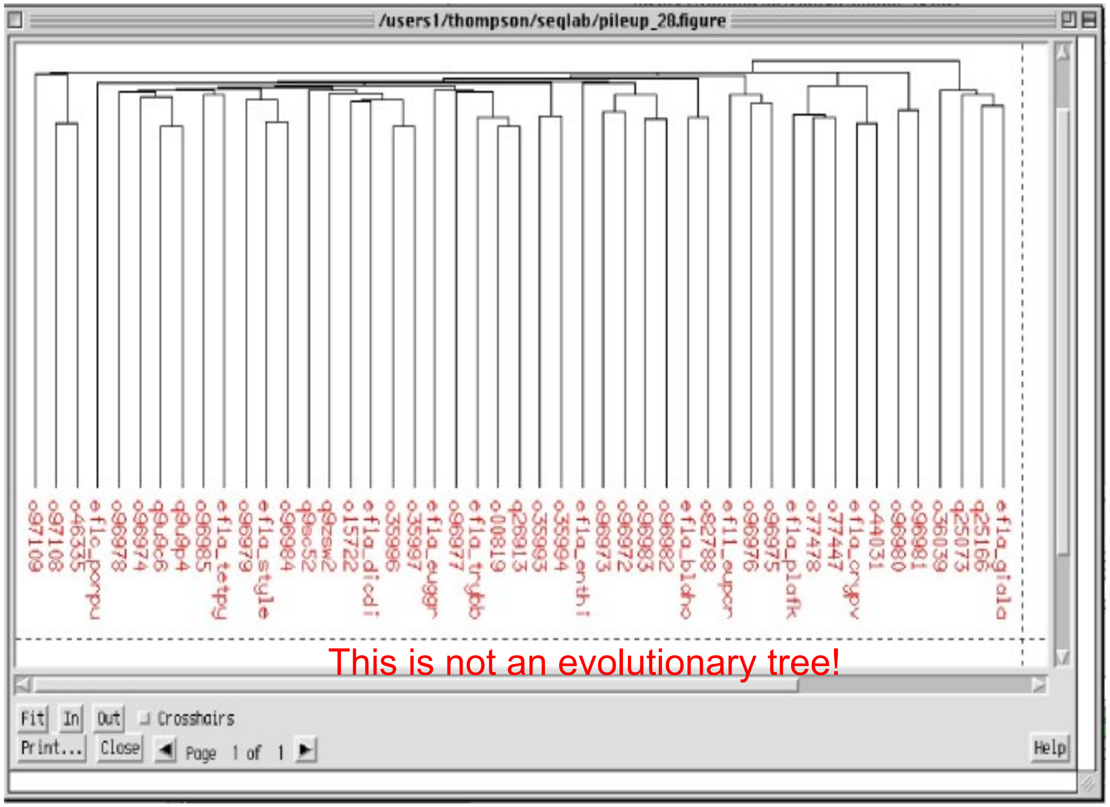
- http://www.ebi.ac.uk/clustalw/

PILEUP

- PILEUP is the multiple sequence alignment program that is part of the Genetics Computer Group (GCG) package developed at the University of Wisconsin.
- Sequences initially aligned in a pair-wise fashion using Needleman-Wunsch algorithm.
- Scores used to produce tree using unweighted pair group method using arithmetic averages (UPGMA)
- The resulting tree is then used to guide the alignment of the most closely related sequences and groups of sequences

PILEUP

- very similar to CLUSTALW
- part of the genetic computer group (GCG)
- does not guarantee optimal alignment
- plots a cluster dendogram of similarities between sequences



This is not an evolutionary tree!

Shortcoming of Progressive Approach

- Dependence upon initial pair-wise sequence alignments
  - Ok if sequences are similar
  - Errors in alignment propagated if not similar

- Suitable scoring matrices and gap penalties must be chosen to apply to the sequences as a set

Iterative Methods

- Iterative alignment methods begin by making an initial alignment of the sequences.
- These alignments are then revised to give a more reasonable result.
- The objective of this approach is to improve the overall alignment score
- Alignment is repeatedly refined
- Selection of groups is based on the phylogenetic tree
- Programs using iterative methods:
  - MultAling
  - PRRP
  - DIALIGN

MultAlign

- Pairwise scores recalculated during progressive alignment

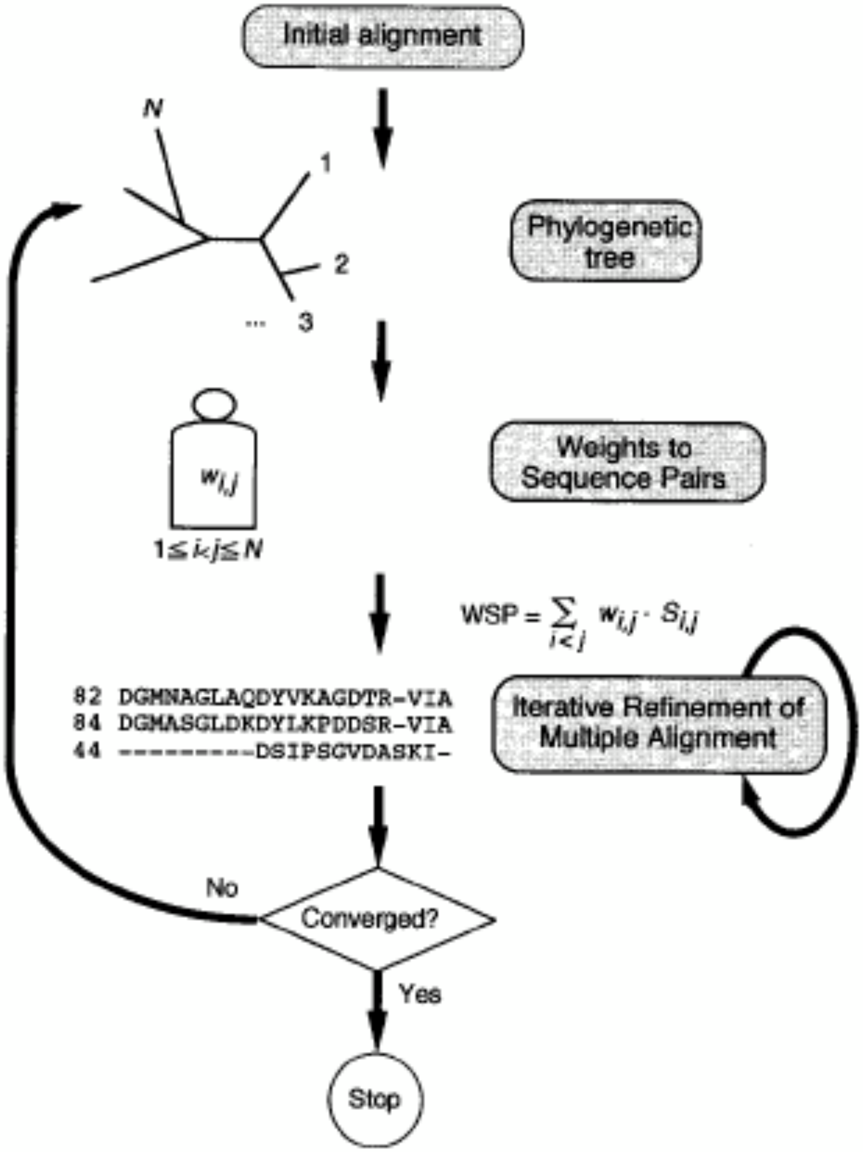
- Tree is recalculated

- Alignment is refined

PRRP

- Initial pairwise alignment predicts tree
- Tree produces weights
- Locally aligned regions considered to produce new alignment and tree
- Continue until alignments converge

Iterative procedure used by PRRP to compute MSA


Initial alignment
N
1
2
3
...
Phylogenetic tree
$w_{i,j}$
$1 \le i,j \le N$
Weights to Sequence Pairs
WSP = $\sum_{i<j} w_{i,j} \cdot S_{i,j}$
Iterative Refinement of Multiple Alignment
No
Converged?
Yes
Stop

```
82 DGMNAGLAQDYVKAGDTR-VIA
84 DGMASGLDKDYLKPDDSR-VIA
44 ---------DSIPSGVDASKI-
```

DIALIGN

- Pairs of sequences aligned to locate ungapped aligned regions

- Diagonals of various lengths identified

- Collection of weighted diagonals provide alignment

Genetic Algorithms

- The goal of genetic algorithms used in sequence alignment is to generate as many different multiple sequence alignments by rearrangements that simulate gaps and genetic recombination events.
- SAGA (Serial Alignment by Genetic Algorithm) is one such approach that yields very promising results, but becomes slow when more than 20 sequences are used.

Genetic Algorithm Approach

1) Sequences (up to 20) written in row, allowing for overlaps of random length – ends padded with gaps (100 or so alignments)

```
XXXXXXXXXX-----
---------XXXXXXXX
--XXXXXXXXX-----
```

Genetic Algorithm Approach

2) The initial alignments are scored by the sum of pairs method.

   - Standard amino acid scoring matrices and gap open, gap extension penalties are used

3) Initial alignments are replaced to give another generation of multiple sequence alignments

   - One half of the multiple sequence alignments are chosen to proceed to the next generation unchanged (natural selection).
     - This half is chosen by assigning probabilities to each sequence based on an inverse proportion of their SP scores (the best alignments, since the SP scores are weighted according to their distance from the parent).
   - The other half of the alignments are sent to the next generation, but are first subject to mutation.

Genetic Algorithm Approach

4) Mutation

- In the mutation process, gaps are inserted into the sequences subject to mutation and rearranged in an attempt to create a better scoring alignment.
- In this step
  - the sequences subject to mutation split into two sets based on estimated phylogenetic tree
  - gaps of random lengths inserted into random positions in the alignment

Genetic Algorithm Approach

- Mutations:

- XXXXXXXX XXX---XXX—XX
- XXXXXXXX XXX---XXX—XX
- XXXXXXXX X—XXX---XXXX
- XXXXXXXX X—XXX---XXXX
- XXXXXXXX X—XXX---XXXX

Genetic Algorithm Approach

5) Recombination of two parents to produce next generation alignment is accomplished

6) The next generation is evaluated going back to step 2, and steps 2-5 are repeated a number (100-1000) times.

   – The best scoring multiple sequence alignment is then obtained (note that it may not be the optimal scoring alignment).

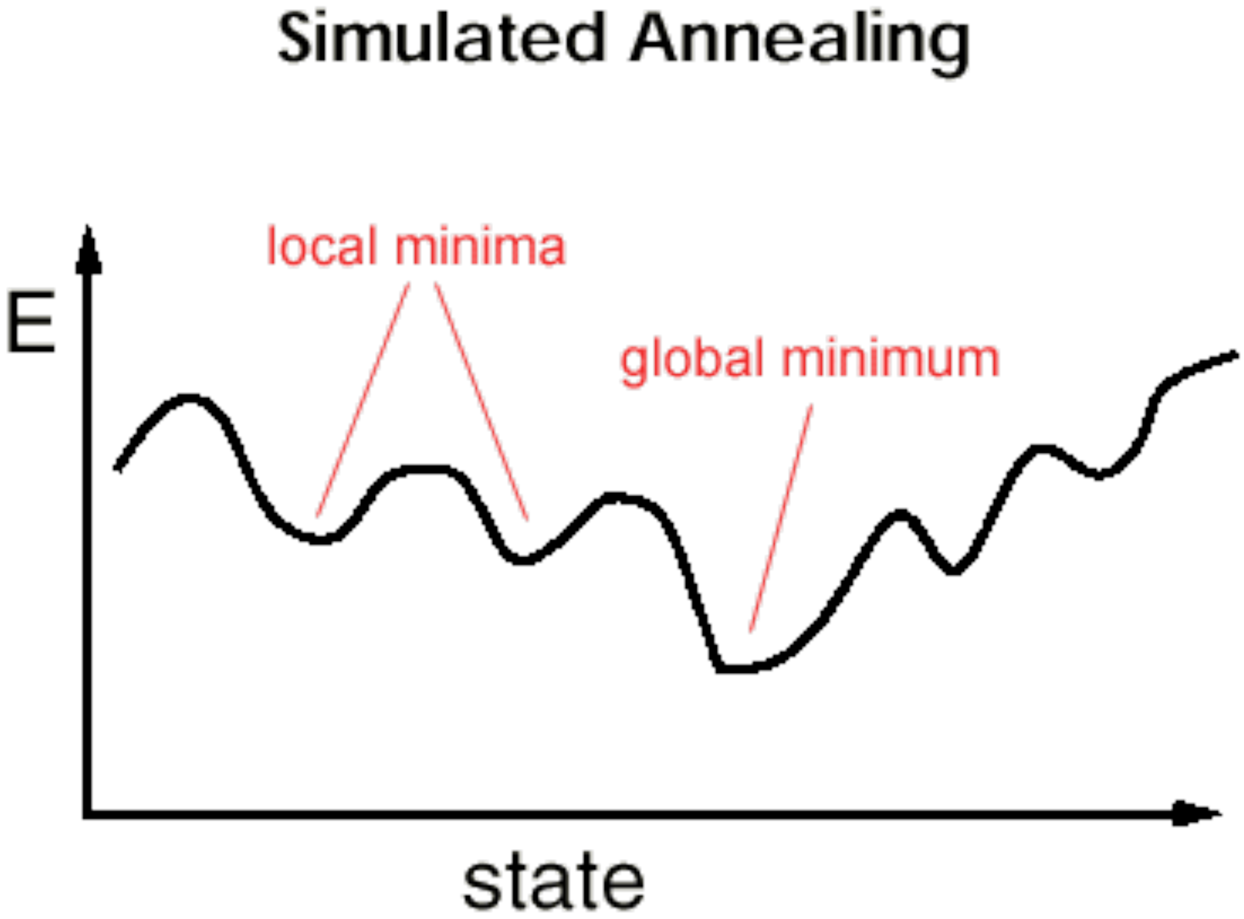
7) The entire process is repeated several times, starting from a different initial alignment each time.

   – The best scoring multiple sequence alignment is then chosen and reported to the user.

Simulated Annealing

- Another approach to sequence alignment that works in a manner similar to genetic algorithms is simulated annealing.
- In these approaches, you begin with a heuristically determined multiple sequence alignment that is then changed using probabilistic models that identifies changes in the alignment that increase the alignment score.
- The drawback of simulated annealing approaches is that you can get stuck finding only the locally optimal alignment rather than the alignment score that is globally optimal
- Rearranges current alignment using probabilistic approach to identify changes that increase alignment score

Simulated Annealing



http://www.cs.berkeley.edu/~amd/CS294S97/notes/day15/day15.html

Group Approach

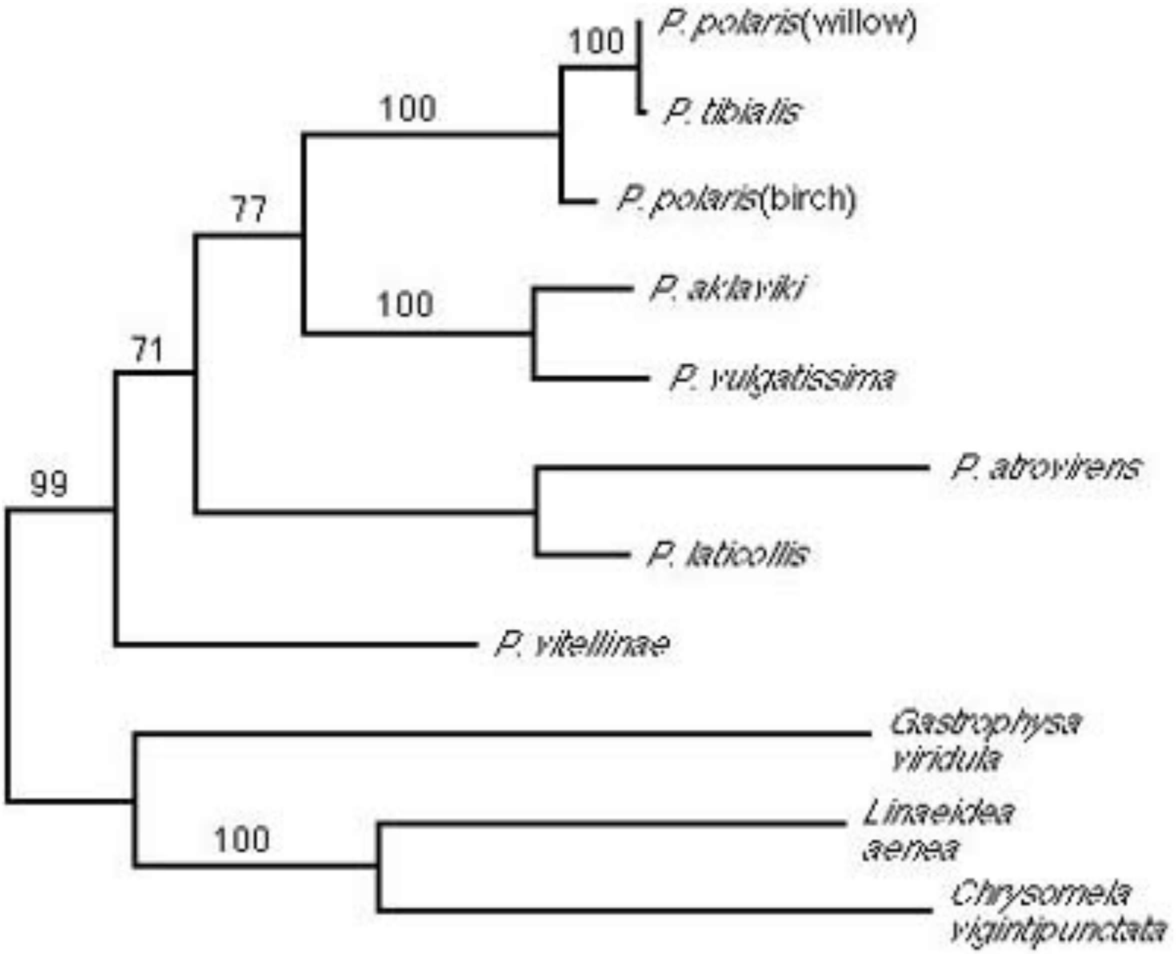
- Sequences aligned into similar groups
- Consensus of group is created
- Alignments between groups is formed

- EXAMPLES: PIMA, MULTAL

Tree Approach

- Tree created
- Two closest sequences aligned
- Consensus aligned with next best sequence or group of sequences
- Proceed until all sequences are aligned

Tree Approach to msa



- www.sonoma.edu/users/r/rank/ research/evolhost3.html

Tree Approach to msa

- PILEUP, CLUSTALW and ALIGN

- TREEALIGN rearranges the tree as sequences are added, to produce a maximum parsimony tree (fewest evolutionary changes)

Profile Analysis

- Create multiple sequence alignment
- Select conserved regions
- Create a matrix to store information about alignment
  - One row for each position in alignment
  - one column for each residue; gap open; gap extend

Profile Analysis

- Profile can be used to search target sequence on database for occurrence

- Drawback: profile is skewed towards training data

Local Multiple Sequence Alignment

Sequence File Formats

Localized Alignments

- Just like with pairwise alignments, we may not be interested in the global alignment of multiple sequences, but rather only specific regions that are conserved.
- Local Alignment of msas are important:
  - Given regions of genomic DNA occurring upstream or before a certain gene, there might be sequences where transcription factors bind to the DNA so that the gene can be transcribed.
  - Thus, if we are interested in determining if there is any signal in the regions upstream of a certain family of genes across several different organisms, it would be important to only find the conserved region, and not try to align all of the genomic DNA
  - Localized alignments of protein sequences can yield information about conserved domains found in otherwise unrelated proteins.

Approaches to Local Alignment

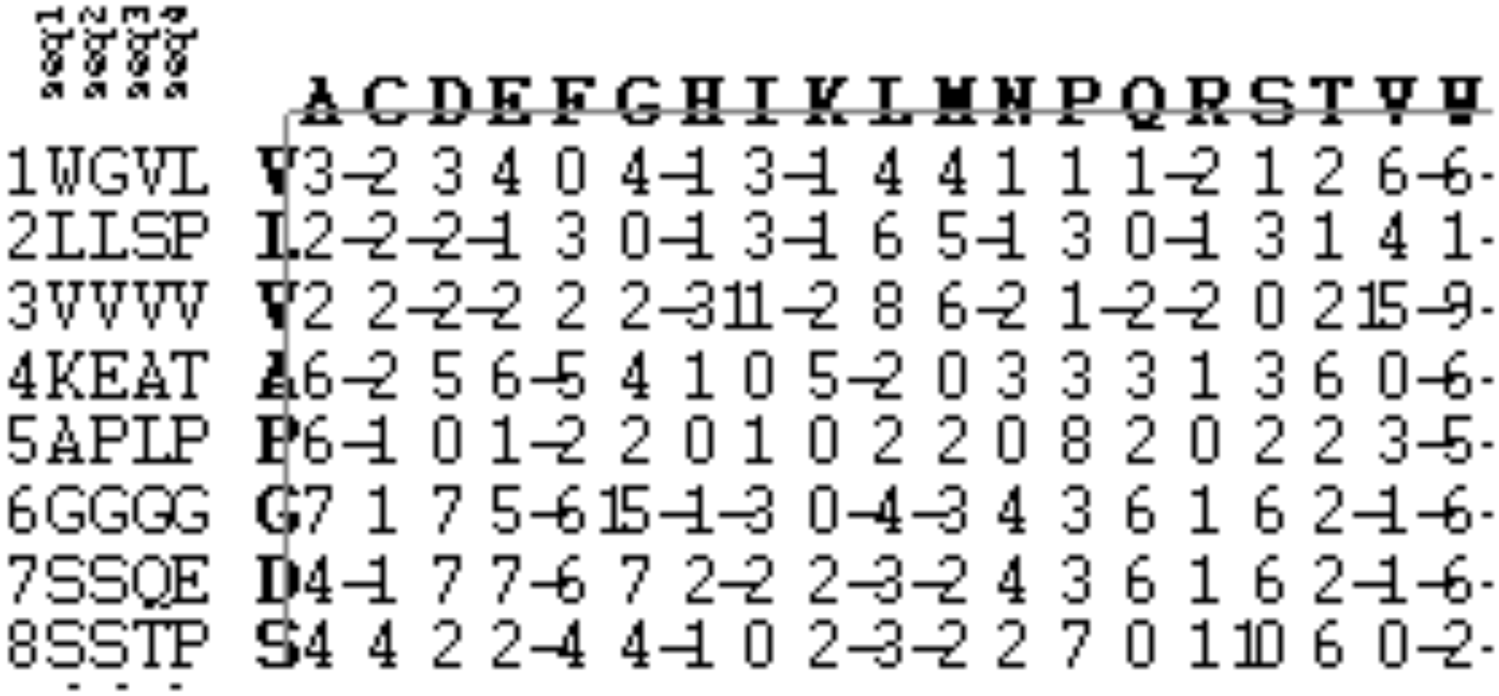
- Profile Analysis
- Block Analysis
- Pattern-searching or statistical methods

Profile Analysis

- Profiles are found by first multiply aligning the sequences, determining which regions are the most highly conserved, and
- then creating a scoring matrix for the alignment of the highly conserved region.
- Profile is composed of:
  - Columns: one for each residue;
    - columns for insertions and deletions as well
  - Rows: one for each position in the conserved region or motif

Profile Analysis

- Profiles describe a msa by a scoring matrix:



seq1 seq2 seq3 seq4		A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	
1WGVL	**V**	3	-2	3	4	0	4	-1	3	-1	4	4	1	1	1	-2	1	2	6	-6	-
2LLSP	**L**	2	-2	-2	-1	3	0	-1	3	-1	6	5	-1	3	0	-1	3	1	4	1	-
3VVVV	**V**	2	2	-2	-2	2	2	-3	11	-2	8	6	-2	1	-2	-2	0	2	15	-9	-
4KEAT	**A**	6	-2	5	6	-5	4	1	0	5	-2	0	3	3	3	1	3	6	0	-6	-
5APLP	**P**	6	-1	0	1	-2	2	0	1	0	2	2	0	8	2	0	2	2	3	-5	-
6GGGG	**G**	7	1	7	5	-6	15	-1	-3	0	-4	-3	4	3	6	1	6	2	-1	-6	-
7SSQE	**D**	4	-1	7	7	-6	7	2	-2	2	-3	-2	4	3	6	1	6	2	-1	-6	-
8SSTP	**S**	4	4	2	2	-4	4	-1	0	2	-3	-2	2	7	0	1	10	6	0	-2	-
. . .																					

Profile Searches

Once a profile is created, it can be used to search a target sequence or database for possible matches to the profile using the profiles scores to evaluate the likelihood at each position.

Profile scores evaluate likelihood of a match at each position

Drawback to Profiles

- Profiles only as representative as the variation in the training sets.

- Thus, there is a bias in the profile towards the training data.

- Training sets can be erroneous if not carefully constructed

Calculating Profiles

- Each cell is the log-odds score
  - The value of an individual cell is calculated as the log odds score of finding a particular residue in a particular location in an alignment divided by the probability of aligning the two amino acids by random chance using a particular scoring scheme (such as PAM250, BLOSUM80, …).
    - PAM (Percent Accepted Mutation)
    - BLOSUM (Blocks Substitution Matrix)
  - Additional penalties must be calculated for gap opening and gap extension in the profile as well.
- Some methods take in sequence weights as well

Shannon Entropy

- One method to calculate the observed column variation given the expected variation in the evolutionary model is to use an information measure known as entropy.
  - Entropy is the amount of information of the observed column variation if expected variation in the evolutionary model is known

- The smaller the entropy, the more conserved a column is.

Entropy

- The entropy (***H***) for a single column is calculated by the following formula:

$$H = -\sum_{residues(a)} f_a \log(p_a)$$

- ***a***: is a residue (amino acid),
- f_a: frequency of residue ***a*** in a column,
- p_a : probability (expected frequency) of residue ***a*** in that column

Entropy

- *H* is calculated for each 20 ancestor amino acids and for a large number of evolutionary distances (PAM1, PAM2, PAM4, ...).
- The distance that gives the minimum value for *H* for each column-possible ancestor combination is the best estimate of the distance that generates the column diversity from that ancestor.
- This analysis provides 20 possible models (M_a for a = 1,2,3....20) as to show how the amino acid frequencies in a column (*F*) may have originated.

Entropy

- The next step in the evolutionary profile construction determines the extent to which each M_a predicts F by the Bayes conditional probability analysis.

$$P(M_a|F) = P(M_a) \times P(F|M_a) / \sum_{\text{all } a\text{©s}} P(M_a) \times P(F|M_a)$$

where the prior distribution $P(M_a)$ is given by the background amino acid frequencies and

$$P(F|M_a) = P_{aa1}^{faa1} \times P_{aa2}^{faa2} \times P_{aa3}^{faa3} \ldots\ldots P_{aa20}^{faa20}$$

i.e., the product of the expected amino acid frequencies in M_a raised to the power of the fraction observed for each amino acid in the msa column.

Entropy

- From $P(M_a|F)$, the weights for each of the 20 possible distributions that give rise to the msa column diversity are calculated as follows:

$$W_a = P(M_a|F) - P(M_{random}|F)$$

where W_a is the weight given to M_a and $P(M_{random}|F)$ is calculated as above using amino acid distribution.

Log-odds score

- Another measure of creating a profile is by using log-odds score.
- In this method, the $\log_2$ of the ratio of observed/ background frequencies is calculated for each position.
- What results is the amount of information available in an alignment given in bits.
- A new sequence can then be searched to see if it possibly contains the motif.
- Profiles can also indicate log-odds score:
  - Log_2(observed:expected)
- Result is a bit score

Log-odds score

- The log odds scores for the profile (Profile_{ij}) are given by

$$\text{Profile}_{ij} = \log\left[\sum_{\text{all } a\text{©s}} (W_{ai} \times P_{aij}) / P_{\text{random}j}\right]$$

where W_{ai} is the weight of an ancestral amino acid a at row i in the profile, P_{aij} is the frequency of amino acid distribution that best matches at row i, and $P_{\text{random}j}$ is the background frequency of amino acid j.

BLOCKS

- Blocks are similar to profiles in the sense that
  - they represent locally conserved regions within a multiple sequence alignment.
- However, the difference is that ...
  - blocks lack insert and delete (indels) positions in the sequences.
  - Instead, every column includes only matches and mismatches
- Blocks can be determined either
  - by performing a multiple sequence alignment, or
  - by searching a database for similar sequences of the same length.

BLOCKS

- Generally determined by performing multiple alignment first

- Ungapped regions are then separated into blocks

- Algorithms have been developed for searching for blocks

BLOCKS

- Statistical approaches to finding the most alike sequences have been proposed, such as
  - the Expectation-Maximization algorithms and
  - the Gibbs sampler.

- In any case, once a set of blocks has been determined, the information contained within the block alignment can be displayed as a sequence profile.

BLOCKS Programs

- A global sequence alignment will usually contain ungapped regions that are aligned between multiple sequences.
- These regions can be extracted to produce blocks.
- Two widely used programs:
  - BLOCKS
  - eMOTIF

http://www.blocks.fhcrc.org/blocks/process_blocks.html

http://dna.stanford.edu/emotif/

- Example
  - 10 Truncated Kinase proteins
  - Approximately 75 residues in length

>D28   CD28 S. CEREVISIAE CELL CYCLE CONTROL PROTEIN KINASE

ANYKRLEKVGEGTYGVVYKALDLRPGQGQR**VVALKKIRLESEDEGVPSTAIREISLLKEL**

>SKH   SKH HELA MYSTERY PUTATIVE PROTEIN KINASE

AKYDIKALIGRGSFSRVVRVEHRATRQPYAIKMIETKYREGREVCESELRVLRRVRHANI

>APK   CAPK BOVINE CARDIAC MUSCLE CYCLIC AMP-DEPENDENT (ALPHA)

DQFERIKTLGTGSFGRVMLVKHMETGNHYAMKILDKQKVVKLKQIEHTLNEKRILQAVNF

>EE1   WEE1 S. POMBE MITOTIC INHIBITOR

TRFRNVTLLGSGEFSEVFQVEDPVEKTLKYAVKKLKVKFSGPKERNRLLQEVSIQRALKG

>GFR   EGFR HUMAN EPIDERMAL GROWTH FACTOR RECEPTOR

TEFKKIKVLGSGAFGTVYKGLWIPEGEKVKIPVAIKELREATSPKANKEILDEAYVMASV

>DGM PDGF RECEPTOR, MOUSE KINASE REGION

DQLVLGRTLGSGAFGQVVEATAHGLSHSQATMKVAVKMLKSTARSSEKQALMSELYGDLV

>FES THIS IS VFES TYROSINE KINASE

VLNRAVPKDKWVLNHEDLVLGEQIGRGNFGEVFSGRLRADNTLVAVKSCRETLPPDIKAK

>AF1   RAF1 HUMAN C-RAF-1 ONCOGENE

SEVMLSTRIGSGSFGTVYKGKWHGDVAVKI LKVVDPTPEQFQAFRNEVAVLRKTRHVNIL

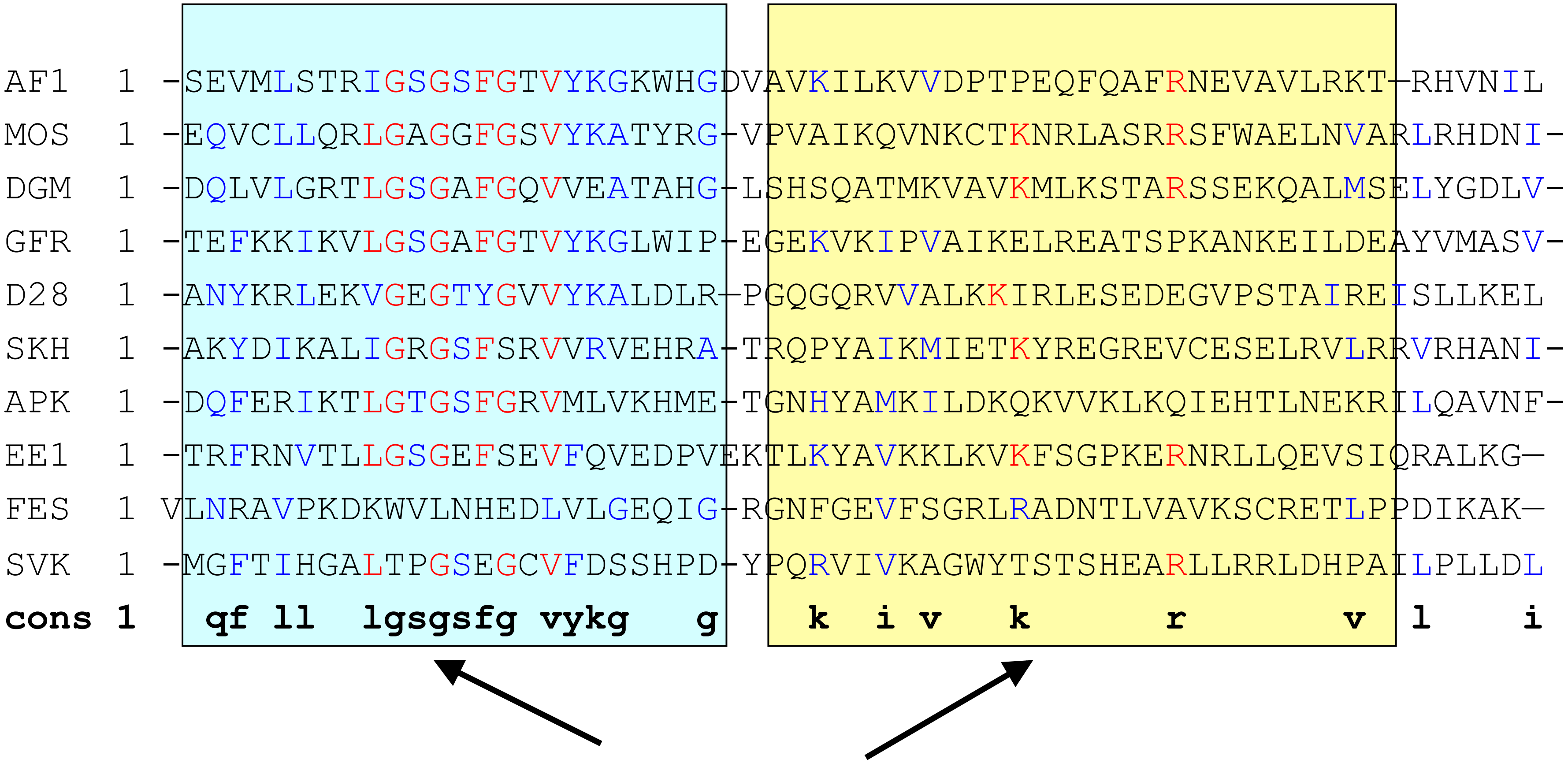
>MOS   CMOS HUMAN C-MOS ONCOGENE

EQVCLLQRLGAGGFGSVYKATYRGVPVAIKQVNKCTKNRLASRRSFWAELNVARLRHDNI

>SVK   HSVK HERPES SIMPLEX VIRUS PUTATIVE PROTEIN KINASE

MGFTIHGALTPGSEGCVFDSSHPDYPQRVIVKAGWYTSTSHEARLLRRLDHPAILPLLDL

Multiple Alignment created using ClustalW; Colors Added using BoxShade



```
AF1  1 -SEVMLSTRIGSGSFGTVYKGKWHGDVAVKILKVVDPTPEQFQAFRNEVAVLRKT-RHVNIL
MOS  1 -EQVCLLQRLGAGGFGSVYKATYRG-VPVAIKQVNKCTKNRLASRRSFWAELNVARLRHDNI-
DGM  1 -DQLVLGRTLGSGAFGQVVEATAHG-LSHSQATMKVAVKMLKSTARSSEKQALMSELYGDLV-
GFR  1 -TEFKKIKVLGSGAFGTVYKGLWIP-EGEKVKIPVAIKELREATSPKANKEILDEAYVMASV-
D28  1 -ANYKRLEKVGEGTYGVVYKALDLR-PGQGQRVVALKKIRLESEDEGVPSTAIREISLLKEL
SKH  1 -AKYDIKALIGRGSFSRVVRVEHRA-TRQPYAIKMIETKYREGREVCESELRVLRRVRHANI-
APK  1 -DQFERIKTLGTGSFGRVMLVKHME-TGNHYAMKILDKQKVVKLKQIEHTLNEKRILQAVNF-
EE1  1 -TRFRNVTLLGSGEFSEVFQVEDPVEKTLKYAVKKLKVKFSGPKERNRLLQEVSIQRALKG-
FES  1 VLNRAVPKDKWVLNHEDLVLGEQIG-RGNFGEVFSGRLRADNTLVAVKSCRETLPPDIKAK-
SVK  1 -MGFTIHGALTPGSEGCVFDSSHPD-YPQRVIVKAGWYTSTSHEARLLRRLDHPAILPLLDL
cons 1   qf ll  lgsgsfg vykg   g    k  i v   k      r       v  l    i
```

BLOCKS Server located blocks

Taking this alignment, blocks can be generated using the BLOCKS server:

```
ID   x6676xbli; BLOCK
AC   x6676xbliA; distance from previous blocks=(1,1)
DE   ../tmp/6676.blin
BL   UNK motif;  width=24; seqs=10; 99.5%=0; strength=0
AF1              (   1) SEVMLSTRIGSGSFGTVYKGKWHG  41
MOS              (   1) EQVCLLQRLGAGGFGSVYKATYRG  48
DGM              (   1) DQLVLGRTLGSGAFGQVVEATAHG  49
GFR              (   1) TEFKKIKVLGSGAFGTVYKGLWIP  41
D28              (   1) ANYKRLEKVGEGTYGVVYKALDLR  61
SKH              (   1) AKYDIKALIGRGSFSRVVRVEHRA  54
APK              (   1) DQFERIKTLGTGSFGRVMLVKHME  46
EE1              (   1) TRFRNVTLLGSGEFSEVFQVEDPV  55
FES              (   1) LNRAVPKDKWVLNHEDLVLGEQIG 100
SVK              (   1) MGFTIHGALTPGSEGCVFDSSHPD  73
//
```

Taking this alignment, blocks can be generated using the BLOCKS server:

```
ID   x6676xbli; BLOCK
AC   x6676xbliB; distance from previous blocks=(2,2)
DE   ../tmp/6676.blin
BL   UNK motif;  width=28; seqs=10; 99.5%=0; strength=0
AF1             (  27) AVKILKVVDPTPEQFQAFRNEVAVLRKT  87
MOS             (  27) PVAIKQVNKCTKNRLASRRSFWAELNVA  75
DGM             (  27) SHSQATMKVAVKMLKSTARSSEKQALMS  92
GFR             (  27) GEKVKIPVAIKELREATSPKANKEILDE  83
D28             (  27) PGQGQRVVALKKIRLESEDEGVPSTAIR  83
SKH             (  27) RQPYAIKMIETKYREGREVCESELRVLR  74
APK             (  27) GNHYAMKILDKQKVVKLKQIEHTLNEKR  85
EE1             (  27) TLKYAVKKLKVKFSGPKERNRLLQEVSI  77
FES             (  27) GNFGEVFSGRLRADNTLVAVKSCRETLP 100
SVK             (  27) PQRVIVKAGWYTSTSHEARLLRRLDHPA  92
//
```

Statistical Methods

- Commonly used methods for locating motifs:
  - Expectation-Maximization (EM)
  - Gibbs Sampling

Expectation-Maximization

- In the expectation-maximization algorithms,
  - the starting point is a set of sequences expected to have a common sequence pattern that may not be easily detectible.
  - An initial guess is made as to the location and size of the site of interest in each of the sequences.
  - These initial sites are then aligned.
  - Approximate length of signal must be given

- Randomly assign locations of this motif in each sequence

Expectation-Maximization

- Two steps:
  - Expectation Step
  - Maximization Step

Expectation-Maximization

- Expectation step
  - In the expectation step, background residue frequencies are calculated based on those residues that are not in the initially aligned sites.
  - Column specific residues are calculated for each position in the initial motif alignment.
  - Using this information, the probability of finding the site at any position in the sequences can then be calculated.
  - Residues not in a motif are background
- Frequencies used to determine probability of finding site at any position in a sequence to fit motif model

Maximization Step

- Maximization step
  - In the maximization step, the counts of residues for each position in the site as found in the expectation step are used to calculate the location within each sequence that maximally aligns to the motif pattern calculated in the expectation step.
  - This is done for each of the sequences.
  - Once a new motif location has been calculated, the expectation step is repeated.
  - This cycle continues until the solution converges.

Example of EM: begin with an initial, Random alignment:

TCAGAACCAGTTATAA**ATTTAT**CATTTCCTTCTCCACTCCT

CCCACGCA**GCCGCC**CTCCTCCCCGGTCACTGACTGGTCCTG

TCGACCCTCTGAACCTATCAGGGACCA**CAGTCA**GCCAGGCAAG

AAAACACTTGAG**GGAGCA**GATAACTGGGCCAACCATGACTC

GGGTGAATGGTACTGCT**GATTAC**AACCTCTGGTGCTGC

AGCCTAGAGT**GATGAC**TCCTATCTGGGTCCCCAGCAGGA

GCCTCAGGATCCAGCACACAT**TATCAC**AAACTTAGTGTCCA

CATTATCAC**AAACTT**AGTGTCCATCCATCACTGCTGACCCT

TCGGAACAAGGCAAA**GGCTAT**AAAAAAAATTAAGCAGC

GCCCCTTCCCCA**CACTAT**CTCAATGCAAATATCTGTCTGAAACGGTTCC

CATGCCCTCAAGTGTGCAGATTGGT**CACAGC**ATTTCAAGG

GATTGGTCACAGCAT**TTCAAG**GGAGAGACCTCATTGTAAG

TCCCCAACTCCCAACTGACCTTAT**CTGTGG**GGGAGGCTTTTGA

CCTTATCTGT**GGGGGA**GGCTTTTGAAAAGTAATTAGGTTTAGC

ATTATTTTCCTTATCAGAAGC**AGAGAG**ACAAGCCATTTCTCTTTCCTCCCGGT

AGG**CTATAA**AAAAAATTAAGCAGCAGTATCCTCTTGGGGGCCCCTTC

CCAGCACACACACTTATC**CAGTGG**TAAATACACATCAT

TCAAATAGGTACGGATAAG**TAGATA**TTGAAGTAAGGAT

ACTTGGGGTTCCAGTTTGATAAGAAAAGACTT**CCTGTG**GA

TGGCCGC**AGGAAG**GTGGGCCTGGAAGATAACAGCTAGTAGGCTAAGGCCAG

CAACCACAACCTCTGTATCCGGTAGTGGCAGATGGAAA

CTGTATCCGGTAG**TGGCAG**ATGGAAAGAGAAACGGTTAGAA

GAAAAAAAATAAATGAAGTCTGCC**TATCTC**CGGGCCAGAGCCCCT

TGCCTTGTCTGTTGTAGATAATGAATCTATCCTCCA**GTGACT**

GGCCAGGCTGAT**GGGCCT**TATCTCTTTACCCACCTGGCTGT

CAACAGCAGGTCCTACTATCGCCTCCCTCT**AGTCTC**TG

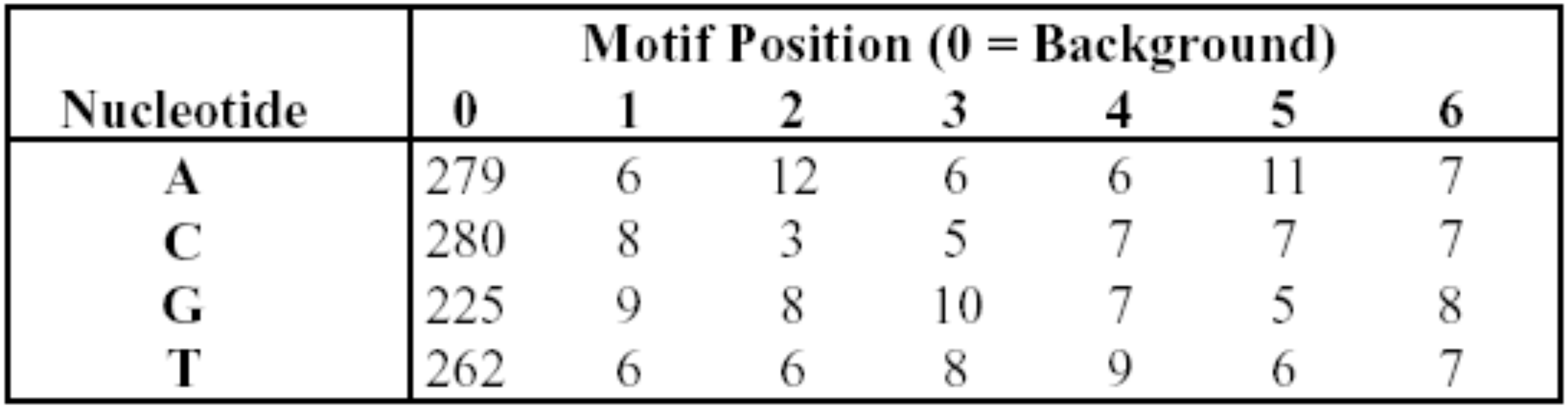
CCAACCG**TTAATG**CTAGAGTTATCACTTTCTGTTATCAAGTGGCTTCAGCTATGCA

GGGAGGGTGGGGCCCCTATCTCTCCTA**GACTCT**GTG

CTTTGTC**ACTGGA**TCTGATAAGAAACACCACCCCTGC

Residue Counts

- From this alignment, the frequency of each base occurring is calculated.
- In this case, the motif we are searching for is six bases wide.
- Therefore, we need to calculate seven different sets of frequencies:
  - One for the background, and
  - one for each of the columns in the motif.
- Calculating the total counts, we get:

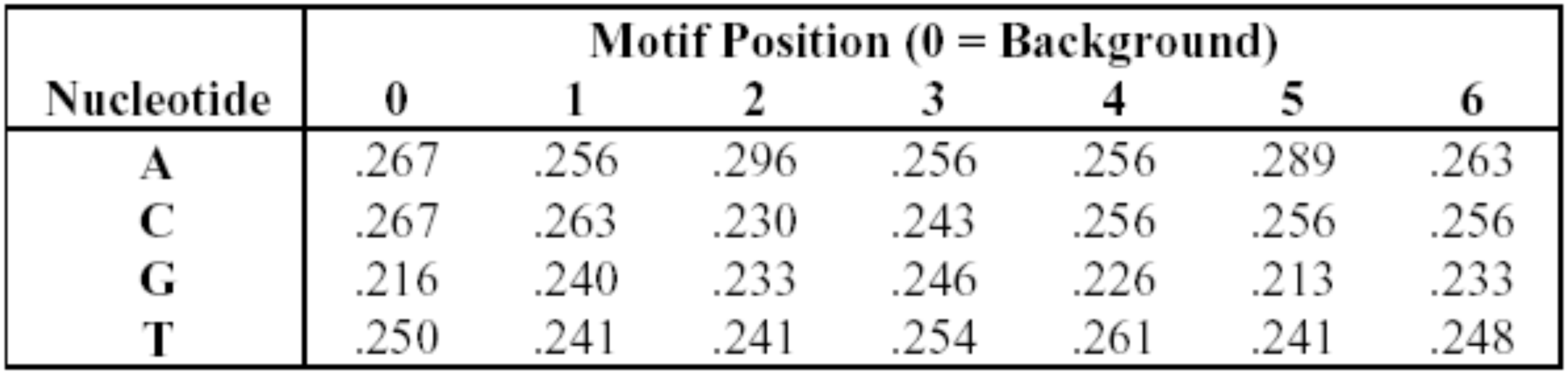


	Motif Position (0 = Background)						
Nucleotide	0	1	2	3	4	5	6
A	279	6	12	6	6	11	7
C	280	8	3	5	7	7	7
G	225	9	8	10	7	5	8
T	262	6	6	8	9	6	7

Table 1: Caclulation of observed counts for inital alignment of figure 1

Residue Frequencies

- After calculating the observed counts for each of the positions, we can convert these to observed frequencies:



Nucleotide	Motif Position (0 = Background) 0	1	2	3	4	5	6
A	.267	.256	.296	.256	.256	.289	.263
C	.267	.263	.230	.243	.256	.256	.256
G	.216	.240	.233	.246	.226	.213	.233
T	.250	.241	.241	.254	.261	.241	.248

Table 2: Calculation of residue frequencies for initial alignment of figure 1

Example Maximization Step

- In the expectation step, the residue frequencies for the motif are used to estimate the composition of the motif site.
- The expectation step attempts to maximally discriminate between sequence within and not within the site.
- For each possible sequence, each possible motif location is considered in order to find the most probable location given the current motif.
- Consider the first sequence:

- `TCAGAACCAGTTATAA`**`ATTTAT`**`CATTTCCTTCTCCACTCCT`

- There are 41 residues; 41-6+1 = 36 sites to consider

	1	2	3	4	5	6	1*2*3*4*5*6	RANDOM	ODDS
TCAGAA	.241	.230	.256	.226	.289	.263	0.000244	0.000274	0.89
CAGAAC	.263	.296	.246	.256	.289	.256	0.000363	0.000362	1.00
AGAACC	.256	.233	.256	.256	.256	.256	0.000256	0.000362	0.71
GAACCA	.240	.296	.256	.256	.256	.263	0.000313	0.000362	0.87
AACCAG	.256	.296	.243	.256	.289	.233	0.000317	0.000362	0.88
ACCAGT	.256	.230	.243	.256	.213	.248	0.000193	0.000274	0.71
CCAGTT	.263	.230	.256	.226	.241	.248	0.000209	0.000257	0.81
CAGTTA	**.263**	**.296**	**.246**	**.261**	**.241**	**.263**	**0.000317**	**0.000257**	**1.23**
AGTTAT	.256	.233	.254	.261	.289	.248	0.000283	0.000241	1.18
GTTATA	.240	.241	.254	.256	.241	.263	0.000238	0.000241	0.99
TTATAA	.241	.241	.256	.261	.289	.263	0.000295	0.000297	0.99
TATAAA	.241	.296	.254	.256	.289	.263	0.000353	0.000297	1.19
ATAAAT	.256	.241	.256	.256	.289	.248	0.000290	0.000318	0.91
TAAATT	.241	.296	.256	.256	.241	.248	0.000279	0.000297	0.94
AAATTT	.256	.296	.256	.261	.241	.248	0.000303	0.000297	1.02
AATTTA	.256	.296	.254	.261	.241	.263	0.000318	0.000297	1.07
ATTTAT	.256	.241	.254	.261	.289	.248	0.000293	0.000278	1.05
TTTATC	.241	.241	.254	.256	.241	.256	0.000233	0.000278	0.84

TTATCA	.241	.241	.256	.261	.256	.263	0.000261	0.000297	0.88
TATCAT	.241	.296	.254	.256	.289	.248	0.000332	0.000297	1.12
ATCATT	.256	.241	.243	.256	.241	.248	0.000229	0.000297	0.77
TCATTT	.241	.230	.256	.261	.241	.248	0.000221	0.000278	0.80
CATTTC	.263	.296	.254	.261	.241	.256	0.000318	0.000297	1.07
ATTTCC	.256	.241	.254	.261	.256	.256	0.000268	0.000297	0.90
TTTCCT	.241	.241	.254	.256	.256	.248	0.000240	0.000278	0.86
TTCCTT	.241	.241	.243	.256	.241	.248	0.000216	0.000278	0.78
TCCTTC	.241	.230	.243	.261	.241	.256	0.000217	0.000297	0.73
CCTTCT	.263	.230	.254	.261	.256	.248	0.000255	0.000297	0.86
CTTCTC	.263	.241	.254	.256	.241	.256	0.000254	0.000297	0.86
TTCTCC	.241	.241	.243	.261	.256	.256	0.000241	0.000297	0.81
TCTCCA	.241	.230	.254	.256	.256	.263	0.000243	0.000318	0.76
CTCCAC	.263	.241	.243	.256	.289	.256	0.000292	0.000339	0.86
TCCACT	.241	.230	.243	.256	.256	.248	0.000219	0.000318	0.69
CCACTC	.263	.230	.256	.256	.241	.256	0.000245	0.000339	0.72
CACTCC	.263	.296	.243	.261	.256	.256	0.000324	0.000339	0.95
ACTCCT	.256	.230	.254	.256	.256	.248	0.000243	0.000318	0.76

- The six base site CAGTTA beginning at base 8 is calculated to have the highest odds probability.
- Therefore, it is chosen as the new site in sequence 1.
- This is repeated for each of the sequences.
- In the maximization step, the newly chosen sites for each of the sequences are used to recalculate the frequency table.
- The expectation/maximization cycle is then repeated, until the results converge on a set of motifs.

Maximization Step

- Before: Random Alignment
- TCAGAACCAGTTATAA**ATTTAT**CATTTCCTTCTCCACTCCT
- After: Maximal location (given random motif alignment) (first round)
- TCAGAAC**CAGTTA**TAAATTTATCATTTCCTTCTCCACTCCT

Available E-M Programs

- MEME – Uses E-M algorithms as explained
- **Multiple EM for Motif Elicitation (MEME)** is a program developed that uses the expectation-maximization methods as described previously.
- ParaMEME searches for blocks using the EM algorithm, while MetaMEME searches for profiles using Hidden Markov Models.
- MEME locates one or more ungapped patterns in a single DNA or protein sequence, or in a series of sequences.
- A search is conducted on a variety of motif widths in order to determine the most likely width for the profile.
- This likelihood is based on the log likelihood score calculated after the EM algorithm.

MEME Software

- One of three types of motif models can be chosen:
  - OOPS: One expected occurrence per sequence
  - ZOOPS: Zero or one expected occurrence per sequence
  - TCM: Any number of occurrences of the motif

MEME Software

- Various prior knowledge can be added to MEME, including the expected number of motifs, the expected length of the motif, and whether or not the motif is palindromic (only applicable for DNA sequences).
  - Palindromic sequences (DNA)
  - Expected number of motifs
  - Expected length of motifs