CS342: Bioinformatics
Multiple Alignments

Section 6.10

Multiple Alignment versus Pairwise Alignment

- Up until now we have only tried to align two sequences.
- What about more than two? And what for?
- A faint similarity between two sequences becomes significant if present in many
- Multiple alignments can reveal subtle similarities that pairwise alignments do not reveal

Generalizing Pairwise Alignment

- Alignment of 2 sequences is represented as a 2-row matrix
- In a similar way, we represent alignment of 3 sequences as a 3-row matrix

```
A T _ G C G _
A _ C G T _ A
A T C A C _ A
```

- Score: more conserved columns, better alignment

Alignment Paths

- Align 3 sequences: ATGC, AATC,ATGC

0	1	1	2	3	4
	A	--	T	G	C
0	1	2	3	3	4
	A	A	T	--	C
0	0	1	2	3	4
	--	A	T	G	C

x coordinate

y coordinate

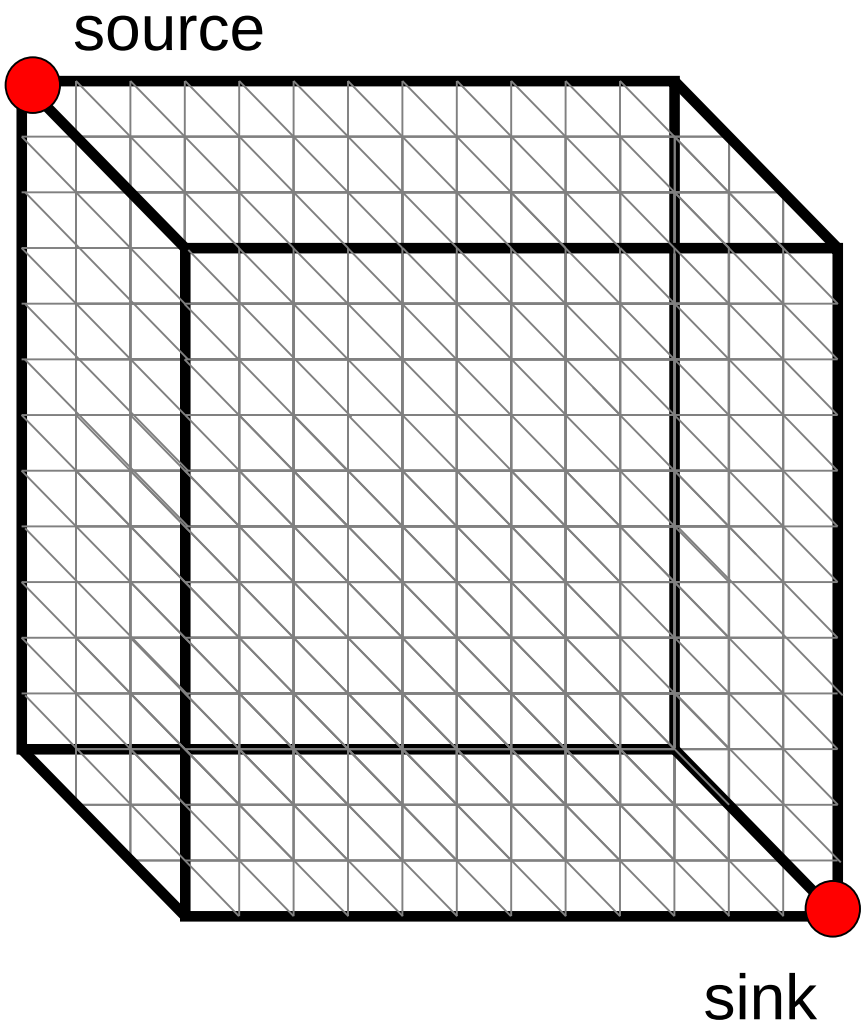
z coordinate

- Resulting path in *(x,y,z)* space:

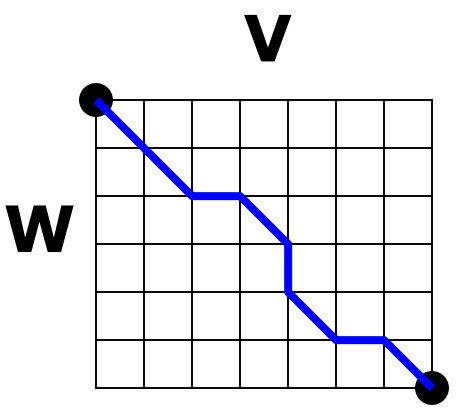
$$(0,0,0)\rightarrow(1,1,0)\rightarrow(1,2,1)\rightarrow(2,3,2)\rightarrow(3,3,3)\rightarrow(4,4,4)$$

Aligning Three Sequences

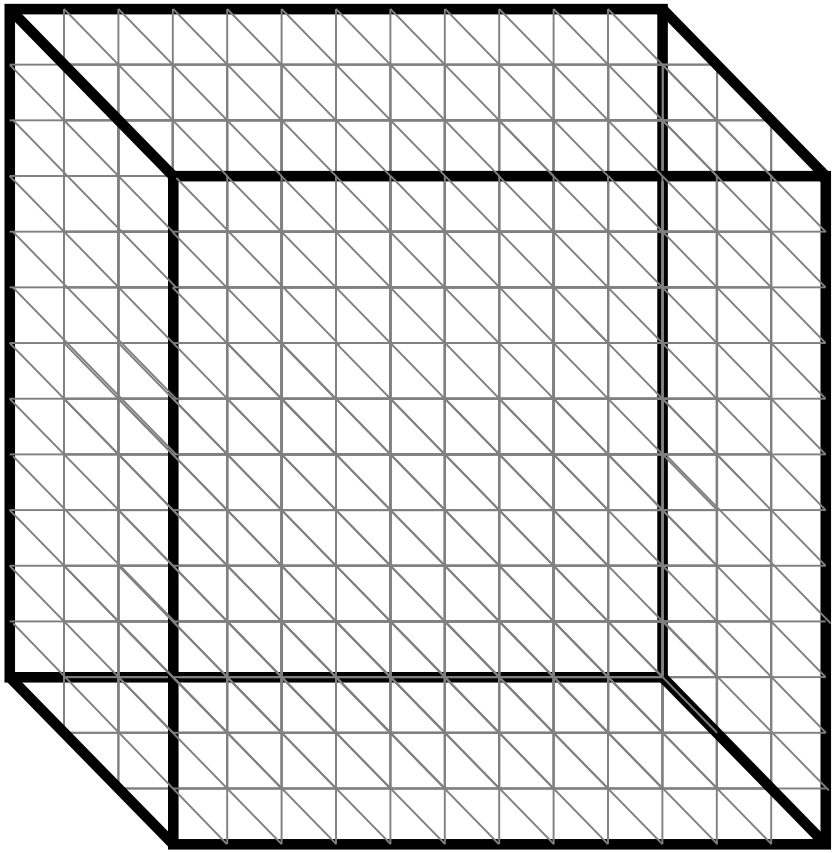
- Same strategy as aligning two sequences
- Use a 3-D "Manhattan Cube", with each axis representing a sequence to align
- For global alignments, go from source to sink



2-D vs 3-D Alignment Grid

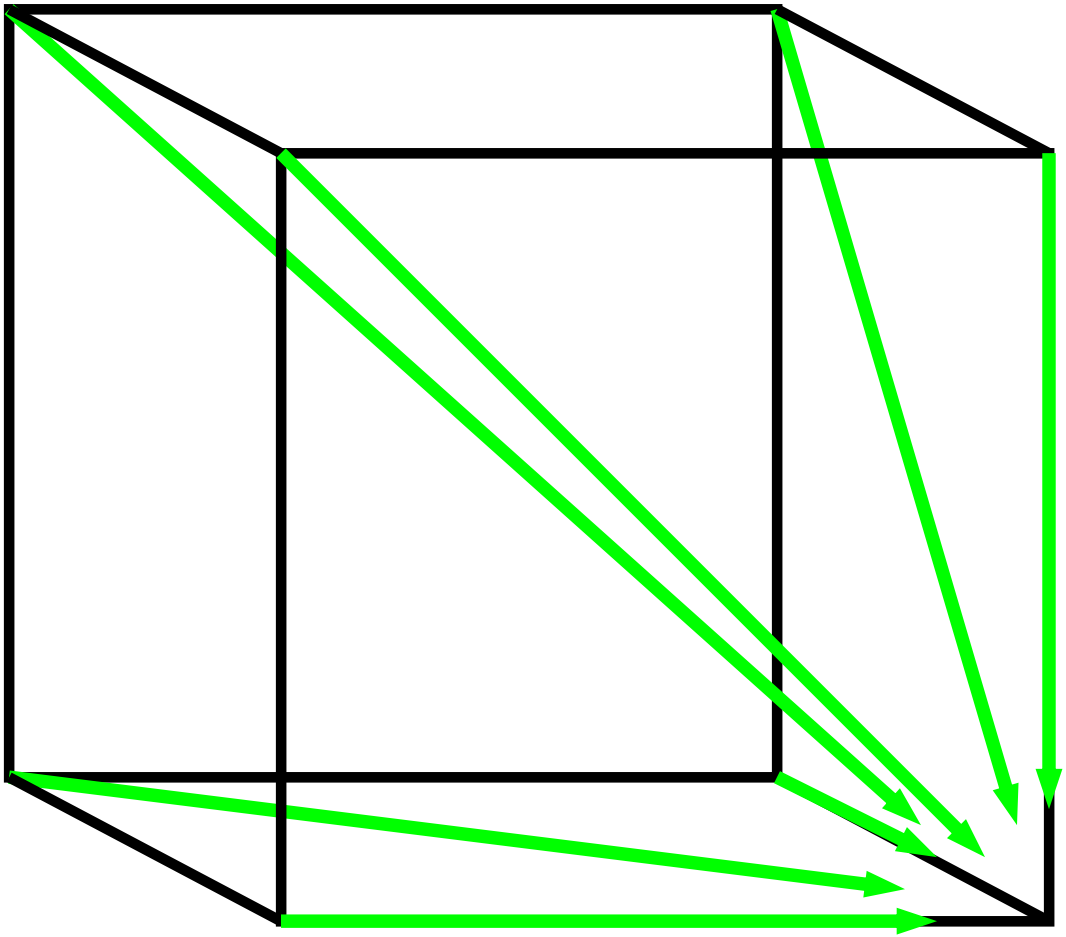


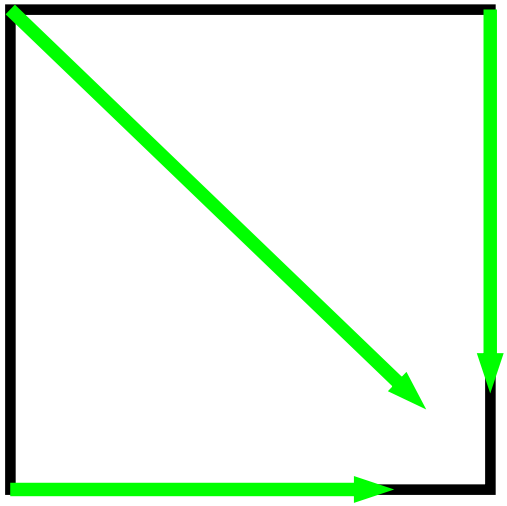
2-D edit graph



3-D edit graph

2-D cell versus 3-D Alignment Cell





In **2-D**, 3 edges lead to each interior vertex

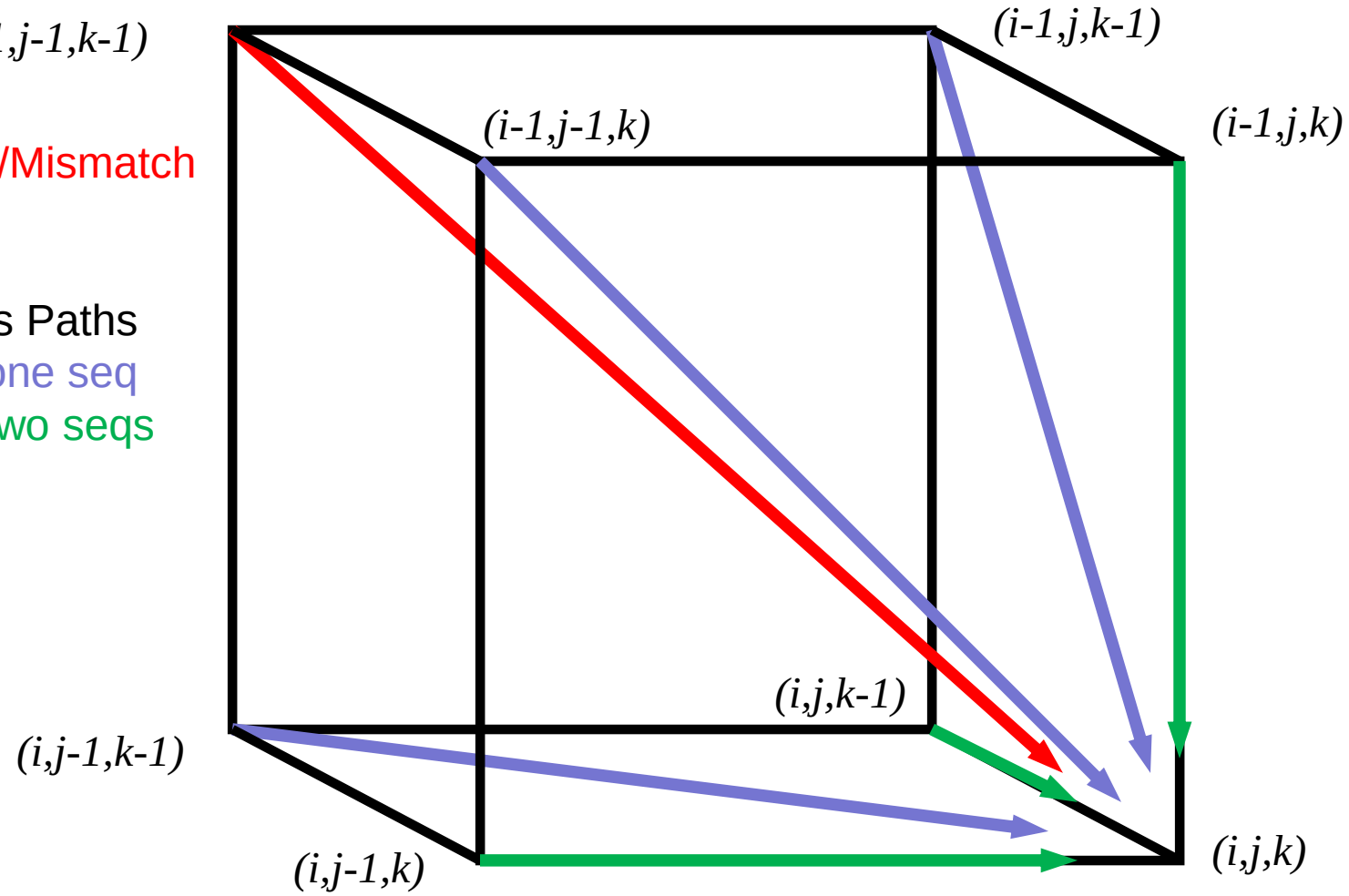
In **3-D**, 7 edges lead to each interior vertex

- 2-D *[(i-1,j-1), (i-1,j), (i,j-1)] → (i,j)*
- 3-D *[(i-1,j-1,k-1), (i-1,j,k), (i,j-1,k), (i,j,k-1), (i,j-1,k-1), (i-1,j,k-1), (i-1,j-1,k),] → (i,j,k)*

Architecture of 3-D Alignment Cell

1- Match/Mismatch Path

6 – indels Paths
- 3 in one seq
- 3 in two seqs



Multiple Alignment: Dynamic Programming

- $s_{i,j,k} = \max$ of:

$s_{i-1,j-1,k-1} + \delta(v_i, w_j, u_k)$	cube diagonal: no indels
$s_{i-1,j-1,k} + \delta(v_i, w_j, \_)$	face diagonal: one indel
$s_{i-1,j,k-1} + \delta(v_i, \_, u_k)$	
$s_{i,j-1,k-1} + \delta(\_, w_j, u_k)$	
$s_{i-1,j,k} + \delta(v_i, \_, \_)$	Lattice edge: two indels
$s_{i,j-1,k} + \delta(\_, w_j, \_)$	
$s_{i,j,k-1} + \delta(\_, \_, u_k)$	

- $\delta(x, y, z)$ is an entry in the 3-D scoring matrix

Multiple Alignment: Running Time

- For 3 sequences of length n, the run time is $7n^3$; $O(n^3)$
- For k sequences, build a k-dimensional table, with run time $(2^k\text{-}1)(n^k)$; $O(2^k n^k)$
- Conclusion: dynamic programming approach for alignment between two sequences is easily extended to k sequences but it is impractical due to exponential running time

Multiple Alignment Induces Pairwise Alignments

Every multiple alignment induces pairwise alignments

```
x:  AC-GCGG-C
y:  AC-GC-GAG
z:  GCCGC-GAG
```

Induces:

```
x: ACGCGG-C;   x: AC-GCGG-C;   y: AC-GCGAG
y: ACGC-GAC;   z: GCCGC-GAG;   z: GCCGCGAG
```

Inverse Problem: Do Pairwise Alignments imply a Multiple Alignment?

Given 3 arbitrary pairwise alignments:

```
x: ACGCTGG-C;   x: AC-GCTGG-C;   y: AC-GC-GAG
y: ACGC--GAC;   z: GCCGCA-GAG;   z: GCCGCAGAG
```

Can we construct a multiple alignment that induces them?

NOT ALWAYS

Why? Because pairwise alignments may be arbitraily inconsistent

Combining Optimal Pairwise Alignments into Multiple Alignment

Can combine pairwise alignments into multiple alignment

Can ***not*** combine pairwise alignments into multiple alignment

Inferring Multiple Alignment from Pairwise Alignments

- From an optimal multiple alignment, we can infer pairwise alignments between all pairs of sequences, but they are not necessarily optimal
- It is difficult to infer a “good” multiple alignment from optimal pairwise alignments between all sequences
- Are we stuck, or is there some other trick?

Multiple Alignment using Profile Scores

	-	A	G	G	C	T	A	T	C	A	C	C	T	G
	T	A	G	-	C	T	A	C	C	A	-	-	-	G
	C	A	G	-	C	T	A	C	C	A	-	-	-	G
	C	A	G	-	C	T	A	T	C	A	C	-	G	G
	C	A	G	-	C	T	A	T	C	G	C	-	G	G
A	0	5	0	0	0	0	5	0	0	4	0	0	0	0
C	3	0	0	0	5	0	0	2	5	0	3	1	0	0
G	0	0	5	1	0	0	0	0	0	1	0	0	2	5
T	1	0	0	0	0	5	0	3	0	0	0	0	1	0
-	1	0	0	4	0	0	0	0	0	0	2	4	2	0

- Thus far we have aligned a **sequence against other sequences**
- Can we align a **sequence against a profile?**
- Can we align a **profile against a profile?**

Aligning alignments

- Given two alignments, can we align them?

```
x GGGCACTGCAT
y GGTTACGTC--     Alignment 1
z GGGAACTGCAG

w GGACGTACC--     Alignment 2
v GGACCT-----
```

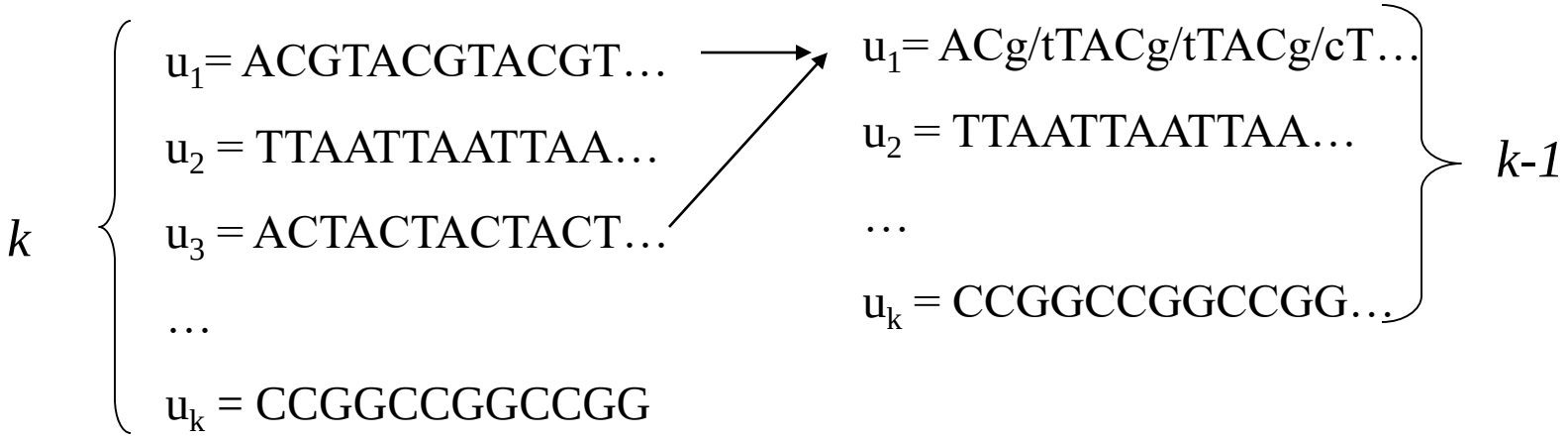
Aligning alignments

- Given two alignments, can we align them?
- Hint: don't use the sequences…
  align their profiles

```
x GGGCAC=TGCAT
y GGTTAC=GTC--
z GGGAAC=TGCAG
  ||  || | |   Combined Alignment
w GG==ACGTACC--
v GG==ACCT-----
```

Multiple Alignment: Greedy Approach

- Choose most similar pair of strings and combine into a profile, thereby reducing alignment of *k* sequences to an alignment of of *k-1* sequences/profiles. **Repeat**
- This is a heuristic *greedy* method



Greedy Approach: Example

- Consider these 4 sequences

```
S1: GATTCA
S2: GTCTGA
S3: GATATT
S4: GTCAGC
```

Scoring Matrix:
Match = 1
Mismatch = -1
Indel = -1

Greedy Approach: Example

- There are $\binom{4}{2}$ = 6 possible alignments

```
s2  GTCTGA
s4  GTCAGC (score = 2)
```

```
s1  GATTCA--
s4  G-T-CAGC (score = 0)
```

```
s1  GAT-TCA
s2  G-TCTGA (score = 1)
```

```
s2  G-TCTGA
s3  GATAT-T (score = -1)
```

```
s1  GAT-TCA
s3  GATAT-T (score  = 1)
```

```
s3  GAT-ATT
s4  G-TCAGC (score = -1)
```

Greedy Approach: Example

s_2 *and* s_4 are closest; combine:

s2 GTCTGA
s4 GTCAGC

$\rangle$ $s_{2,4}$ (profile) GTCt/aGa/c

new set of 3 sequences:

s_1	GATTCA
s_3	GATATT
$s_{2,4}$	GTCt/aGa/c

Repeat

Greedy Approach: Example

Repeat for $\binom{3}{2}$ = 3 possible alignments

s_1 : GAT-TCA
s_3 : GATAT-T
(score = 1 + 1 + 1 - 1 + 1 - 1 - 1 = 1)

s_1 : GAT-TCA
$s_{2,4}$: G-TCtGa
(score = 2 - 2 + 2 - 2 + 1 - 1 + 1 = 1)

s_3 : GATAT-T
$s_{2,4}$: G-TCtGa
(score = 2 - 2 + 2 - 2 + 1 - 1 - 1 = -1)

Progressive Alignment

- *Progressive alignment* is a variation of greedy algorithm with a somewhat more intelligent strategy for choosing the order of alignments.
- Progressive alignment works well for close sequences, but deteriorates for distant sequences
  - Gaps in consensus string are permanent
  - Use profiles to compare sequences

- CLUSTAL

ClustalW (Clustal Omega)

- Popular multiple alignment tool commonly used today
- ‘W’ stands for ‘weighted’ (different parts of alignment are weighted differently).
- Three-step process

  1.) Construct pairwise alignments

  2.) Build Guide Tree

  3.) Progressive Alignment guided by the tree

Step 1: Pairwise Alignment

- Aligns each sequence against each other giving a similarity matrix
- Similarity = exact matches / sequence length (percent identity)

	v_1	v_2	v_3	v_4
v_1	–			
v_2	.17	–		
v_3	.87	.28	–	
v_4	.59	.33	.62	–

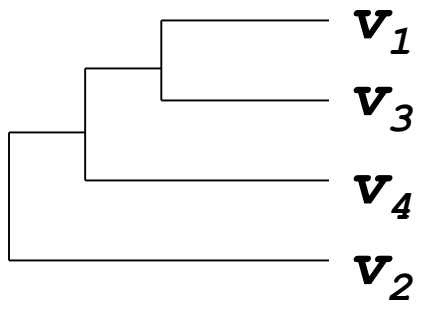
(.17 means 17 % identical)

Step 2: Guide Tree

- Create Guide Tree using the similarity matrix
  - ClustalW uses the neighbor-joining method (we will discuss this later in the course, in the section on clustering)
  - Guide tree roughly reflects evolutionary relations

Step 2: Guide Tree (cont'd)

	v_1	v_2	v_3	v_4
v_1	–			
v_2	.17	–		
v_3	.87	.28	–	
v_4	.59	.33	.62	–



Calculate:

$v_{1,3}$ = alignment (v_1, v_3)

$v_{1,3,4}$ = alignment$((v_{1,3}), v_4)$

$v_{1,2,3,4}$ = alignment$((v_{1,3,4}), v_2)$

Step 3: Progressive Alignment

- Start by aligning the two most similar sequences
- Following the guide tree, add in the next sequences, aligning to the existing alignment
- Insert gaps as necessary

```
FOS_RAT          PEEMSVTS-LDLTGGLPEATTPESEEAFTLPLLNDPEPK-PSLEPVKNISNMELKAEPFD
FOS_MOUSE        PEEMSVAS-LDLTGGLPEASTPESEEAFTLPLLNDPEPK-PSLEPVKSISNVELKAEPFD
FOS_CHICK        SEELAAATALDLG----APSPAAAEEAFALPLMTEAPPAVPPKEPSG--SGLELKAEPFD
FOSB_MOUSE       PGPGPLAEVRDLPG-----STSAKEDGFGWLLPPPPPPP-----------------LPFQ
FOSB_HUMAN       PGPGPLAEVRDLPG-----SAPAKEDGFSWLLPPPPPPP-----------------LPFQ
                 .   . :   ** .     :..  *:.*   *   . *                  **:
```

Dots and stars show how well-conserved a column is.