CS 466 Introduction to Bioinformatics Lecture 12

Mohammed El-Kebir

October 17, 2018



Course Announcements

10/17/2018	12	MSA 4	HW3 MSA
10/22/2018	14	Phylogeny I	
10/24/2018	15	Phylogeny II	
10/29/2018	16	Phylogeny III	
10/31/2018	17	Phylogeny IV	HW4 Phylogeny
11/5/2018	18	Assembly I	
11/7/2018	19	Assembly II	
11/12/2018	20	нммт	
11/14/2018	21	НММ II	HW5 HMM, implementation Project proposal deadline
11/19/2018		Thanksgiving	
11/21/2018		Thanksgiving	
11/26/2018	22	Pattern Matching I	
11/28/2018	23	Pattern Matching II	
12/3/2018	24	Review	
12/5/2018	25	Final	
12/10/2018	26		
12/14/2018	27	Project deadline	

Course Project

Project

There are three kinds of projects.

- 1. Implement an algorithm discussed in class, and make it available on Github.
- 2. Benchmark algorithms discussed in class that solve the same problem on simulated or real data. Write a report about your findings.
- 3. Write a small survey paper, summarizing state-of-the-art algorithms for a specific computational biology problem.

Project proposal due on Nov. 14

(Motivation, Datasets/papers, Planned method/experiments, Timeline)

Project report due on Dec. 14

Outline

- Scoring matrices
- Tree/star alignment
- Progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book "Algorithms on Strings, Trees and Sequences" by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes

Substitution Matrices

- Given a pair (v, w) of aligned sequences, we want to assign a score that measure the relative likelihood that the sequences are related as opposed to being unrelated
- We need two models:
 - Random model R: each letter $a \in \Sigma$ occurs independently with probability q_a
 - Match model *M*: aligned pair $(a, b) \in \Sigma \times \Sigma$ occur with joint probability $p_{a,b}$

$$\Pr(\mathbf{v}, \mathbf{w} | R) = \prod_{i} q_{v_i} \cdot \prod_{i} q_{w_i}$$

$$\Pr(\mathbf{v}, \mathbf{w} | M) = \prod_{i} p_{v_i, w_i}$$

$$\log \frac{\Pr(\mathbf{v}, \mathbf{w} | M)}{\Pr(\mathbf{v}, \mathbf{w} | R)} = \sum_{i} s(v_i, w_i) \text{ where } s(a, b) = \log \frac{p_{a,b}}{q_a q_b}$$

BLOSUM (Blocks Substitution Matrices)

- Henikoff and Henikoff, 1992
- Computed using ungapped alignments of protein segments (blocks) from BLOCKS database
- Thousands of such blocks go into computing a single BLOSUM matrix
- Example of a one such block (right):
 - 31 positions (columns)
 - 61 sequences (rows)
- Given threshold *L*, block is pruned down to largest set *C* of sequences that have at least *L*% sequence identity to another sequence in *C*
 - How to compute *C*?

SHLLRHORTHDKTA	PKPLMEGPVAGOGEDVE
CHILDHOD THD UD I	E IDENE CONF CUNENTE
SHELKHURIHDKDI	FVFEWESRVESHWENIE
SHLLRHQRIHDKNV	Q C P E W K S R M E S Q L E N V E
SHLLPHOPTHDENN	OF PENKSEMEGOLENVE
SHEEKIQKINDKAV	Q . T L W K S K H L O Q B L N Y L
SHLLRHQRIHDKNV	QETEWKSRMESQLENVE
SHLLRHORIHDKNV	OFPEWKSRMESOLENVE
CHLIDHODTHDUNG	OF DEHVEDTESOLENUE
SHEEKHQKINDKW	QCPE WKSKIESQLENVE
SHLLRHQRIHDKNV	QEPEWKSRTESQLENVE
SHLLRHORTHDKSV	OF PENEGRIESOMONVE
	o provide the second se
SHLERHURIHDKSV	QEPEWEGRIESQWQNVE
SHLLRHQRIHDKNA	PNPEWESQMEIQERNVE
SHILDHODTHDVSN	OV DE MECOVECOMENVE
SHEEKHQKINDKSK	QAFEWECKVEGQWENVE
SHLLRHQRIHDKNA	PEPGWECRVEGQWENVE
SHLLRHORVHDKKI	OESEWGCRTESOWENVO
SHLIDHODVHDVVI	OF SEMCCOTESOMENVO
SHEEKHQKVHDKKI	QLSEWGCKIESQWENVQ
SHLLRHRRIHDKNV	Q D P E W E Y R G E G Q W E N N E
SHLLRHRRTHDKNY	OD PEMEYRGEGOMENNE
CHLIDHODTHDDH	ON DEVE OD TE COVENUE
SHLLRHURIHDRNA	UDPEWESRTESUWENVD
SHLLRHORIHDRNA	Q PEWESRTESQWENVD
SHLLDHODTHDVNV	ODSEMESDMESOMENVE
SHEEKHQKINDKNY	Q SEWESKIESQWENVE
SHLLRHQRIHDKNV	QNPEWESRTESQWENTE
SHLLRHORIHDKNV	ONPEWESRTESOWENTE
SHLIDHODTHDUNG	OIDFUESDTESOUFNTE
SUPPERUATION	UNPEWESKIESUWENIE
SHLLRHQRIHDKNV	QNPEWESRTESQWENTE
SHLLRHORTHDKNY	O JPE MERRTESOMENTE
SHLERHURIHDKNY	ANALEMEKKIE SAMENIE
SHLLRHQRIHDKNF	QNPEWEGRTESQWENVE
SHLLRHORTHDKNE	O JPENEGRTESOMENVE
SHLLRHQRIHNKNV	ENPEWESRVESQWENVE
SHLLRHORIHNKNV	ENPEWESRVESOWENVE
SHLLDHODTHNKSV	O UDE MESOMESOMESVE
SHEEKHQKINKSV	QVFEWESKHESQWESVE
SHLLRHQRIHNKNV	QTLEWESRMESQWESVE
SHLLRHORIHNKNI	ONPOWESRKESOWENVE
SHIIDHODTHNUNI	OUDDWESDVESOWENVE
SHEEKHQKIHNKNI	QNFDWESKKESQWENVE
SHLLRHQRIHDKNV	QNPDWESRMESQWENVE
SHLLRHORIHDKNV	ONPOWESRMESOWENVE
SHLIDHODTHDVNU	ODFUESDUESDUENUE
SUPPERUATION	UPREWESRVESRWENVE
SHLLRHQRIHDKNV	Q D R E W E S R V E S R W E N V E
SHLLRHORTHDKNA	O JPKGOSBRESOMENFE
CHLIDHODTHDUN	OIDVCOCDDECOUENEE
SHLERHQRINDKNA	UNPROUSERE SUWENTE
SHLLRHQRIHEKSV	Q D L D W Q S R L E S Q W G D V E
SHLLRHORTHDNNN	O JPD MESRMESOEGHTE
CULIDUODTUDUUU	ODDUESDNESOFCUTE
SHLERHURIHDKNV	QUIDWESKHESUEGHIE
SHLLRHQRIHDKSV	QNPKWECRKGGQEENAE
SHLLRHORIHDKSV	ON PRIMECREGGOEENAE
CHLIDHODTHDUCK	O IDD UE CD VE C CUE VAE
SUPPERUNKIUNKPA	UNFDWESKHESSWENAL
SHLLRHQRIHDKSV	Q N P D WE SRMESSWENAE
SHLLEHERVHDKDV	ODPEMEDRVERSEGSVE
	O DEVEDDUED CECCUE
SHLLRHRRVHDKDV	UPPEWEDRVERSEGSVE
SHLLRHQRIHDKNN	QD SEWESRMENQWENAE
SHLLRHORTHDENN	OD SEMESBMENOMENAK
CHLIDHODUND	E CENENDUE NOUEVAR
SHLIKHURVHDKNI	LPSEWENRVENUWERTE
SHLLRHORVHDKNI	EDSEWENRVENQWEDTE
SHLLRHOPTHAPHY	REPOWEGRLEGOWENTE
and D BRIDERINARIA	DEPENDENTE COMENTE
SHLERHQRIHAKNV	REPDWEGRMESQWENTE
SHLLRHORIHERNI	Q C P D W E G R M E S O W E N V G
SHLIPHOPTHEPNT	OF PDMF GPMFSONFNVC
SHEEKNUKINEKWI	C. LOWLORNE SQUENVG
SHLLRHQRIHNRCH	HPAVFESETETQUGNLE
SHLLRHORIHNRCH	HDAVFESETETONGNLE
SHLIPHOPTHNPFF	HOPECECEVETONENIE
SHEERIQKINKFF	IN TECEOR VETQUENEE
SHLLRHQRIHNRFF	HPFECEGEVETQUENLE

BLOSUM (Blocks Substitution Matrices)

$$\log \frac{\Pr(\mathbf{v}, \mathbf{w} | R)}{\Pr(\mathbf{v}, \mathbf{w} | M)} = \sum_{i} s(v_i, w_i) \text{ where } s(a, b) = \frac{1}{\lambda} \log \frac{p_{a,b}}{q_a q_b}$$

- Null model frequencies $q_a q_b$ of letters a and b:
 - Count the number of occurrences of *a* (*b*) in all blocks
 - Divide by sum of lengths of each block (sequences * positions)
- Match model frequency $p_{a,b}$:
 - Count the number of pairs (*a*, *b*) in all columns of all blocks
 - Divide by the total number of pairs of columns:
 - $\sum_{C} n(C) \binom{m(C)}{2}$
 - m(C) is the number of sequences in block C
 - n(C) is the number of positions in block C

SHLLRHORTHDKTA	PKPLMEGPVAGOGEDVE
SHILDHODTHDVDE	FUDENESDVESHMENTE
CULIDUODTUDUU	OF DE WUGDWEGOLENVE
SHLERHQRIHDKNV	QL FEWKSKME SQLENVE
SHLLRHURIHDKNV	Q PEWKSRMEGULENVE
SHLLRHQRIHDKNV	QETEWKSRMESQLENVE
SHLLRHQRIHDKNV	QEPEWKSRMESQLENVE
SHLLRHORIHDKNV	Q E P E W K S R T E S Q L E N V E
SHLLRHORIHDKNY	O E PENKSRTESOLENVE
SHLLRHORTHDKST	OF PENEGRIESOMONVE
SHIIDHODTHDUST	OF DENE COTESONONYE
SHEEKHQKINDKS	Q.FEWEGRIESQWQNVE
SHLLRHURIHDKNA	PNPEWESQMEIQERNVE
SHLLRHQRIHDKSN	QKPEWECRVEGQWENVE
SHLLRHQRIHDKNA	PEPGWECRVEGQWENVE
SHLLRHQRVHDKKI	QESEWGCRTESQWENVQ
SHLLRHQRVHDKKI	QESEWGCRTESQWENVQ
SHLLRHRRIHDKNV	O D P E W E Y R G E G O W E N N E
SHLLRHRRIHDKNY	OD PENEYRGEGOMENNE
SHLIDHODTHDDNA	OD DE MESDITESOMENVD
SHLIDHODTHDDN	O DENESDIESQUENVD
SHEEKHQKINDKNA	Q FEWESKIESQWENVD
SHLLRHURIHDKNV	UPSEWESRMESUWENVE
SHLLRHQRIHDKNV	QNPEWESRTESQWENTE
SHLLRHORIHDKNY	ONPEWERRTESOWENIE
SHLLBHORTHDKNY	O UPENERRTESOMENTE
SHLLPHORTHDENE	OUPENEGRTESONENVE
SHLLPHOPTHDENE	OUPFMECRTESOMENVE
SHILDWODTHNEN	FUPFUFSDUFSOUFNUF
SHEEKHQKIHNKN	ENFEWESRVESQWENVE
SHLLKHURIHNKNV	ENPEWESKVESUWENVE
SHLLRHQRIHNKSV	QNPEWESRMESQWESVE
SHLLRHQRIHNKNV	QTLEWESRMESQWESVE
SHLLRHQRIHNKNI	QNPDWESRKESQWENVE
SHLLRHQRIHNKNI	QNPDWESRKESQWENVE
SHLLRHQRIHDKNV	QNPDWESRMESQWENVE
SHLLRHQRIHDKNV	Q JPD WE SRME SQ WENVE
SHLLRHQRIHDKNV	Q D R E W E S R V E S R W E N V E
SHLLRHORIHDKNV	Q D RE WE SRVE SR WE N VE
SHLLRHORIHDKNA	O JPKGOSRRESOMENFE
SHLLRHORTHDKNA	O JPKGOSBRESOMENFE
SHLLPHOPTHEKST	ODLDWOSPLESONGDVE
SHILDHODTHDNNS	OUDDWESDWESOFCHIE
CHLIDHODTHDVNV	ON DD WEST MESS OF CHIE
SHEEKHQKINDKN	Q FDWESKNESQEGHIE
SHLLKHURIHDKSV	UNPRWECKKGGUEENAE
SHLLRHURIHDKSV	UNPRWEERKGGUEENAE
SHLLRHQRIHDKSV	QNPDWESRMESSWENAE
SHLLRHQRIHDKSV	QNPDWESRMESSWENAE
SHLLRHRRVHDKDV	Q D PEWEDRVERSEGSVE
SHLLRHRRVHDKDV	Q D PEWEDRVERSEGSVE
SHLLRHQRIHDKNM	Q D SE WE SRMENQ WENAE
SHLLRHORIHDKNN	Q D SE WE SRMENQWENAK
SHLLRHORVHDKNI	EDSEWENRVENOWEKTE
SHLLRHORVHDKNI	EDSEWENRVENOWEDTE
SHLLPHOPTHAPHY	REPOWEGRLEGOWENTE
SHIDDHODTHAVNE	DE DOME COME COMENTE
GUIIDUODTUEDUA	OF PD WE COME SQUENTE
GULLDUODTUD	CEPDWEGKHESUWENVG
SHLLRHURIHERNI	Q. PDWEGRMESUWENVG
SHLLRHQRIHNRCH	HPAVFESETETQWGNLE
SHLLRHQRIHNRCH	HDAVFESETETQWGNLE
SHLLRHQRIHNRFH	HPPECEGEVETQWENLE
SHLLRHQRIHNRFF	HPPECEGEVETQWENLE

BLOSUM (Blocks Substitution Matrices)

$$\log \frac{\Pr(\mathbf{v}, \mathbf{w} | R)}{\Pr(\mathbf{v}, \mathbf{w} | M)} = \sum_{i} s(v_i, w_i) \text{ where } s(a, b) = \frac{1}{\lambda} \log \frac{p_{a,b}}{q_a q_b}$$

- Null model frequencies $q_a q_b$ of letters a and b:
 - Count the number of occurrences of *a* (*b*) in all blocks
 - Divide by sum of lengths of each block (sequences * positions)
- Match model frequency $p_{a,b}$:
 - Count the number of pairs (*a*, *b*) in all columns of all blocks
 - Divide by the total number of pairs of columns:
 - $\sum_{C} n(C) \binom{m(C)}{2}$
 - m(C) is the number of sequences in block C
 - n(C) is the number of positions in block C

Example:
$$(\lambda = 0.5)$$
 $q_A = \frac{7}{15}$ $A A T$ $A A T$ $G A T$ $q_T = \frac{3}{15}$ $T A L$ $p_{A,T} = \frac{4}{30}$ $T A V$ $A A L$ $s(A,T) = 2 \cdot \log \frac{\frac{4}{30}}{\frac{7}{15} \cdot \frac{3}{15}} \approx 0.3$

BLOSUM62

Ala 4

$\log \frac{\Pr(\mathbf{v}, \mathbf{w} R)}{\Pr(\mathbf{v}, \mathbf{w} M)} = \sum_{i} s(v_i, w_i) \text{ where } s(a, b) = \frac{1}{\lambda} \log \frac{p_{a,b}}{q_a q_b}$

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	lle	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val		9
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-		
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5			<u>http</u>	<u>s://d</u>	oi.or	g/10.	1038	/nbt08	04-1035
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	and	1 +10.5	for W/	W, wh	ich we	re rou	nded t	to +4 a	ind +11		
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	nur	nbers ((with B	LOSUN	M62's o	origina	$1 \lambda = 0$).347) :	and yo	ou get +3.	8 for L/L
lle	-1	-3	-3	-3	-1	-3	-3	-4	-3	4		try	ptopha	in is a n	nuch r	arer ai	nino a	cid (f_{L}	= 0.09	$99, f_{W}$	= 0.013).	Run those
His	-2	0	1	-1	-3	0	0	-2	8			try	ptopha	n/tryp	otopha	n (W/V	W) pai	rs (p_{LL}	= 0.03	$\mathbf{B71}, p_{\mathrm{W}}$	$_{\rm W} = 0.006$	65), but
Gly	0	-2	0	-1	-3	-2	-2	6				leu	cine/le	eucine	(L/L) I	oairs w	vere in	fact n	nore c	ommo	on than	
Glu	-1	0	0	2	-4	2	5					the	homo	logous	alignm	nent da	ata tha	t BLOS	SUM6	2 was	trained o	on,
Gln	-1	1	0	0	-3	5						sur	prising	g it wou	ıld be t	to see	two of	them	align	togeth	ner by ch	ance. In
Cys	0	-3	-3	-3	9							ide	ntitites	s get th	e sam	e score	e? The	rarer	the an	nino a	cid is, th	e more
Asp	-2	-2	1	6								SCO	ore +11,	while l	eucine	: (L/L)	pairs o	only so	core +	4; why	[,] shouldn	't all
Asn	-2	0	6									cou	ınterin	tuitive	at firs	t glanc	e. For	instan	nce, tr	yptop	han (W/V	N) pairs
Arg	-1	5										Thi	s expla	ins sor	ne det	ails in	BLOSU	J M62 1	that n	nay see	em	

Outline

Scoring matrices

- Tree/star alignment
- Current progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book "Algorithms on Strings, Trees and Sequences" by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes

Example – Tree Alignment



Figure 14.6: a. A tree with its nodes labeled by a (multi)set of strings, b. A multiple alignment of those strings that is consistent with the tree. The pairwise scoring scheme scores a zero for each match and a one for each mismatch or space opposite a character. The reader can verify that each of the four induced alignments specified by an edge of the tree has a score equal to its respective optimal distance. However, the induced alignment of two strings which do not label adjacent nodes may have a score greater than their optimal pairwise distance.

Outline

- Multiple sequence alignment
- Scoring matrices
- Tree/star alignment
- Current progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book "Algorithms on Strings, Trees and Sequences" by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes

Progressive Alignment – Feng and Doolittle (1987)

- 1. Compute pairwise sequence alignments of *n* sequences
- 2. Generate complete graph G = (V, E) with edge weights $w : E \to \mathbb{R}$
- 3. Compute a (rooted) minimum spanning tree *T* of *G*
- 4. Perform sequence-sequence, sequence-profile and profileprofile alignment to construct MSA according to guide tree *T*



Minimum spanning tree is a tree T spanning all vertices of G with minimum total weight

'Once a gap, always a gap'

Progressive Alignment – ClustalW (1994)

- Widely used alignment method by Thompson, Higgins and Gibson (1994)
- W stands for weighted:
 - Input sequences are weighted to compensate for biased representation
 - Different substitution matrices depending on expected similarity in guide tree (BLOSUM80 for closely related sequences, and BLOSUM50 for distant sequences)
 - Position-specific gap-open and gap-extend penalties depending on context (hydrophobic vs. hydrophilic)

Three steps:

- 1. Construct pairwise alignments
- 2. Build guide tree T using neighbor joining*
- 3. Progressive alignment guided by T

ClustalW – Step 2: Guide Tree

Create Guide Tree using the similarity matrix

("cluster" distances. Details to come...)



ClustalW uses the neighbor-joining method Guide tree roughly reflects evolutionary relationships Calculate:

V _{1,3}	= alignment (v ₁ , v ₃)
V _{1,3,4}	<pre>= alignment((v_{1,3}),v₄)</pre>
V _{1,2,3,4}	= alignment($(v_{1,3,4}), v_2$)

ClustalW – 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 FOS_MOUSE FOS_CHICK FOSB_MOUSE FOSB_HUMAN 

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

MUSCLE (Edgar, 2004)

<u>Multiple Sequence Comparison by Log-Expectation</u>

Three phases:

- 1. Draft progressive alignment: fast heuristic
- 2. Improved progressive: use tree derived in phase 1
- 3. Refinement of MSA
 - Remove sequence from MSA and realign to profile of remaining sequences
 - Repeat until convergence



Progressive MSA



Summary

- 1. Optimal pairwise alignment by dynamic programming in $O(n^2)$ time
- 2. Optimal multiple alignment with SP-score by dynamic programming in $O(k^2 2^k n^k)$ time
- 3. Multiple alignment with SP-score is NP-hard (Jiang and Wang, 1994)
- 4. Carrillo-Lipman enables us to decide whether alignment passes through a vertex (i_1, i_2, i_3) for k = 3 sequences (generalizes to k > 3)
- 5. Star alignment gives 2-approximation algorithm
- 6. Progressive alignment methods are widely used, but come with no theoretical bounds on alignment quality

History

- 1975 Sankoff Formulated MSA problem and gave dynamic programming solution
- 1988 Carrillo-Lipman Branch and Bound approach for MSA
- 1990 Feng-Doolittle Progressive alignment
- 1993 Gusfield Star alignment: 2-approximation algorithm
- 1994 Jiang and Wang MSA with SP-score is NP-hard
- 1994 Thompson-Higgins-Gibson: ClustalW Most popular multiple alignment program
- 2000 Notredam-Higgins-Heringa: T-coffee Use library of pairwise alignments
- 2004 Edgar: MUSCLE Refinement

Outline

- Scoring matrices
- Tree/star alignment
- Progressive alignment methods

Reading:

- Material based on Chapter 14.6 in book "Algorithms on Strings, Trees and Sequences" by Dan Gusfield
- Chapter 6.7 in Jones and Pevzner
- Lecture notes