Exercises

1) Describe these alignments as global/local and exhaustive/approximate: BLAST (appro, local), Needleman-Wunsch (exhaustive, global), Smith-Waterman (exhaustive, local), Bowtie (appr, local)

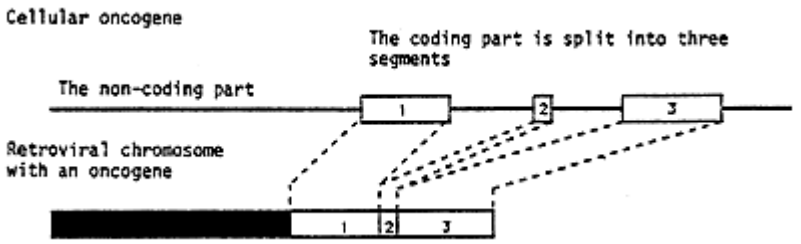
2) Which alignment method would you use: BLAST, Needleman-Wunsch, Smith-Waterman?

a) Generate an alignment between a cDNA and the human genome (e.g. v-src to human) BLAST

b) Find homolog of human cDNA in chicken (e.g. human and chicken src) BLAST

c) Find best alignment of human and chicken homologs (e.g. human and chicken src) Needleman-Wunsch

d) Align v-src to c-src Smith-Waterman



Exercises

1) Fill in the scoring matrix using Needleman-Wunsch with match = 1, mismatch = -1 and gap = -2

	-	G	A	C	C	C	C	T	A	T	T	G
-												
A												
T												
G												
G												
C												
A												
T												
T												

2) How many optimal alignments are there? 1

3) When using the affine gap penalty, do you get more gaps with B = -1 or B = -2, affine gap penalty = A+B*L? -2

4) Write down the NW and SW alignments given the following matrices.

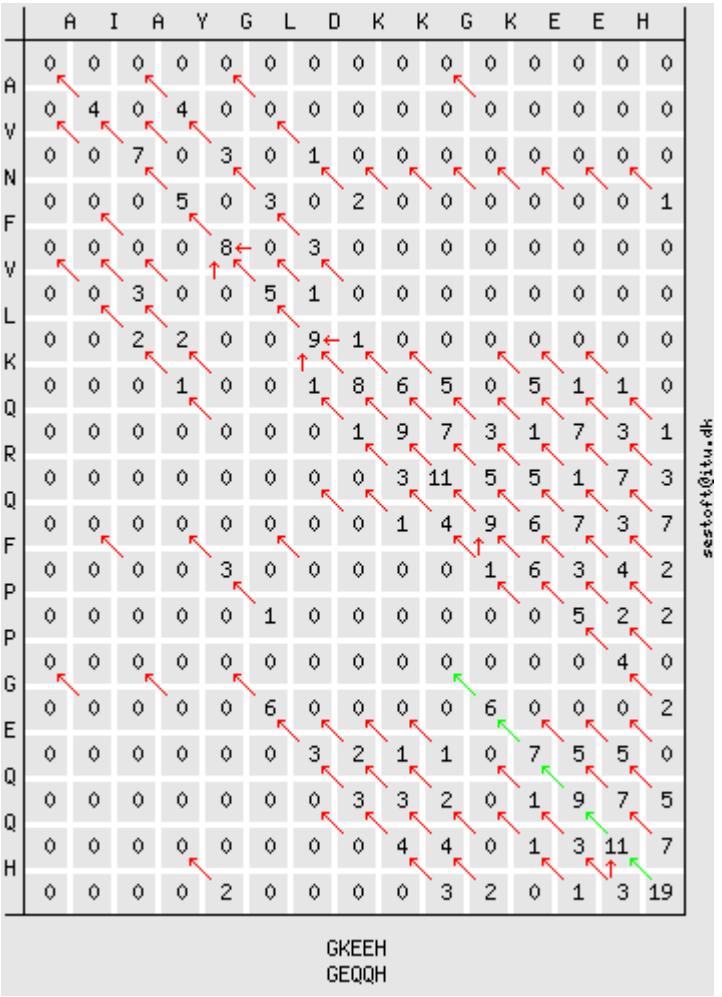
Lower extension penalty = larger & fewer gaps

match = 1 mismatch = –1 gap = –2

		G	A	C	C	C	C	T	A	T	T	G
	0	–2	–4	–6	–8	–10	–12	–14	–16	–18	–20	–22
A	–2	–1	–1	–3	–5	–7	–9	–11	–13	–15	–17	–19
T	–4	–3	–2	–2	–4	–6	–8	–8	–10	–12	–14	–16
G	–6	–3	–4	–3	–3	–5	–7	–9	–9	–11	–13	–13
G	–8	–5	–4	–5	–4	–4	–6	–8	–10	–10	–12	–12
C	–10	–7	–6	–3	–4	–3	–3	–5	–7	–9	–11	–13
A	–12	–9	–6	–5	–4	–5	–4	–4	–4	–6	–8	–10
T	–14	–11	–8	–7	–6	–5	–6	–3	–5	–3	–5	–7
T	–16	–13	–10	–9	–8	–7	–6	–5	–4	–4	–2	–4

```
GACCCCTATTG
|    |  |||
-ATGGC-ATT-
```


A I A Y G L D K K G K E E H
A V N F V L K Q R Q F P P G E Q Q H
sestoft@itu.dk
AIAYGL-DKK---GKEEH
AVNFVLKQRQFPPGEQQH


A I A Y G L D K K G K E E H
A V N F V L K Q R Q F P P G E Q Q H
sestoft@itu.dk
GKEEH
GEQQH

5) What is the complexity of Needleman-Wunsch algorithm and Smith-Waterman? O(nm) time and space

6) Align these two sequences using Smith-Waterman with a gap penalty of -8 and Blosum62 scoring.

	-	H	E	A	G	A	W	G	H	E	E
-	0	0	0	0	0	0	0	0	0	0	0
P	0										
A	0										
W	0										
H	0										
E	0										
A	0										
E	0										
A	0										

BLOSUM62 Substitution matrix (e.g., used in sequence alignment scoring)

Table shows bonus or penalty score for substituting one amino acid for another

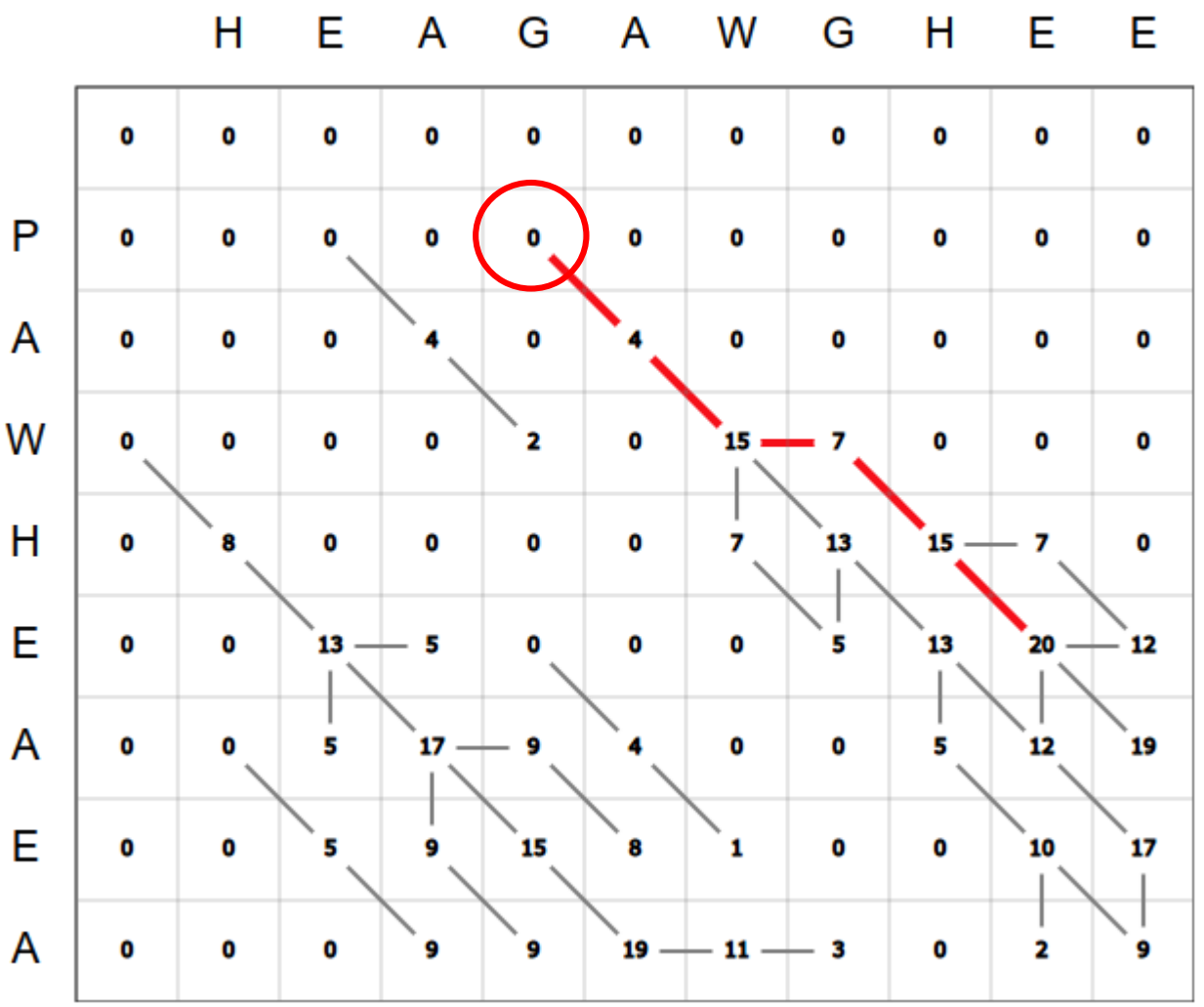
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W

7) Local or global alignments:

a) query can only be represented once (global)

b) handles rearrangements (local)

6) Align these two sequences using Smith-Waterman with a gap penalty of -8 and Blosum62 scoring.



AWGHE

AW-HE

BLOSUM62 Substitution matrix (e.g., used in sequence alignment scoring)

Table shows bonus or penalty score for substituting one amino acid for another

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W

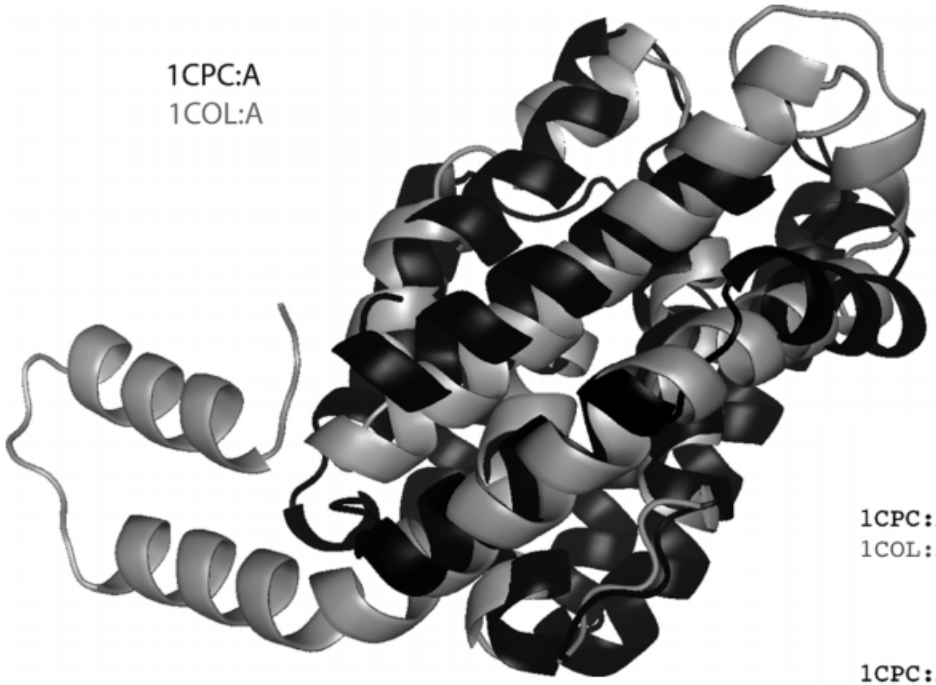
Today's objectives

- Homology search problem
- How does BLAST work
- Pairwise vs multiple sequence alignment

Homology Search Problem

- Optimal alignment (Needleman-Wunsch) of two genes is fast enough, but searching a database (NCBI) of proteins using a single query can be slow
- Large databases can yield spurious hits, i.e. slight sequence similarity by chance rather than homology
- Finding distant homologs is important, protein sequence can diverge to the point where there is no significant similarity even though protein structure is conserved

Structure compared to sequence homology



Structure alignment for c-phycocyanin (1CPC:A) (black) and colicin A (1COL:A) (gray). The sequence identity is 11.9%.

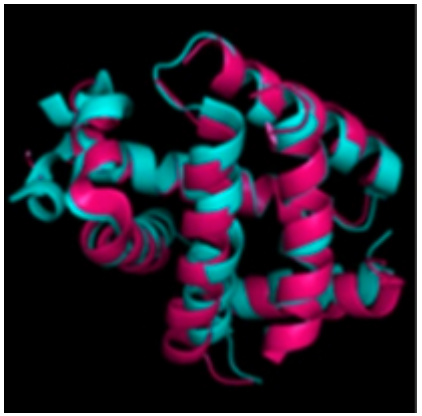
- Speed
- Distant homologs
- Significance

```
                10        20        30        40        50        60
1CPC:A    ------------------------------------------------------------MKTPLT
1COL:A    AKDERELLEKTSELIAGMGDKIGEHLGDKYKAIAKDIADNIKNFQGKTIRSFDDAMASLNKITANP--
                                                                      *

          70        80        90        100       110       120       130
1CPC:A    EAVAAADSQGRFLSSTEIQTAFGRFRQASASLAAAKALTEKASSL-ASGAANAVYSKFPYTTSQNGPN
1COL:A    -------------------------AMKINKADRDALVNAWKHVDAQDMANKLGNLSKAFK-----V
                                    *   **        *   **

           140       150       160       170       180       190       200
1CPC:A    FASTQTGKDKCVRDIGYYLRMVTYCLVVGGTGPLDDYLIGGIAEINRTFD---LSPSWYVEALKYIKA
1COL:A    A-----DVVMKVEKVREKSIEGYET-------GNWGPLMLEVESWVLSGIASSVALGIFSATLGAYAL
                  *                         *                         *

             210       220       230       240
1CPC:A    NHGLSGDPAVEANSYIDYAINALS--------------
1COL:A    SLGVPAIAVGIAGILLAAVVGALIDDKFADALNNEIIR
            *         *         **
```



Lumbricus terrestris hemoglobin and
Paramphistomum epiclitum hemoglobin, Identity=12.1%

What is expected identity of random protein alignment?
P(match|amino acid) ~ 1/20

BLAST algorithm

BLAST - **B**asic **L**ocal **A**lignment **S**earch **T**ool (Altschul et al. 1990).

- rapidly compares a query sequence to a database (target) to find all sequences and their alignments (pairwise) above some cutoff score
- uses a seed and extend heuristic to improve speed
- seeding is accomplished through preprocessing the query (dictionary lookup)
- for biological sequences of length n, there are 4^n and 20^n different strings, for DNA and proteins, respectively

Seed and Extend

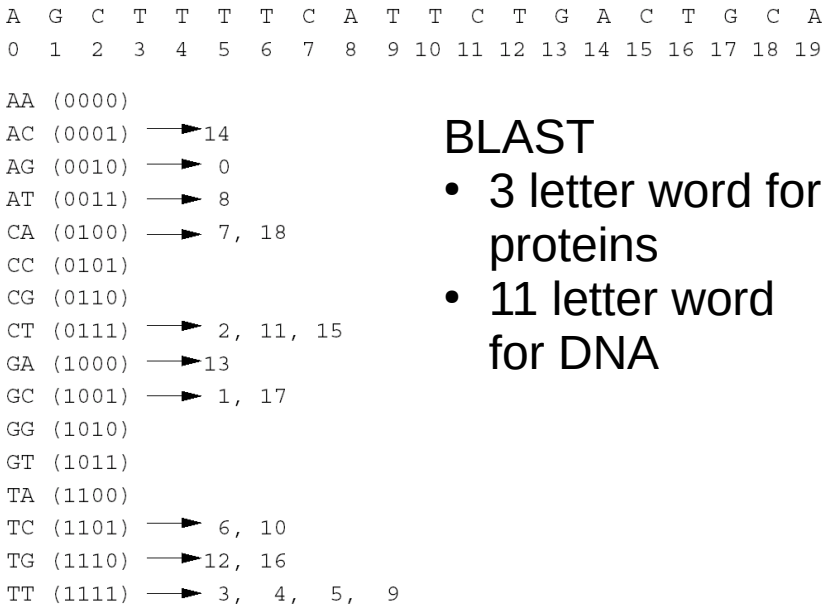
```
FAKDFLAGGVAAAISKTAVAPIERVKLLLQVQHASKQITADKQYKGIIDCVVRIPKEQGV
F  D  +GG AAA+SKTAVAPIERVKLLLQVQ ASK I  DK+YKGI+D ++R+PKEQGV
FLIDLASGGTAAAVSKTAVAPIERVKLLLQVQDASKAIAVDKRYKGIMDVLIRVPKEQGV
```

- Homologous sequences are likely to contain a short high scoring word pair, i.e. a seed.
- BLAST then tries to extend high scoring word pairs to compute maximal high scoring segment pairs (HSPs)

Hash tables for the seeds

- A hash table is a data structure which implements an associative array abstract data type, a structure that can map keys to values.
- Uses a hash function to compute an index into an array of buckets or slots
- Ideally, the hash function will assign each key to a unique bucket, but most employ an imperfect hash function
- hash tables turn out to be more efficient than search trees or any other table lookup structure

- k-tuple: substring of length k
- For alphabet size of A, # entries = A^k
- Memory is ~O(n)
- Lookup time is constant ~ O(1)!

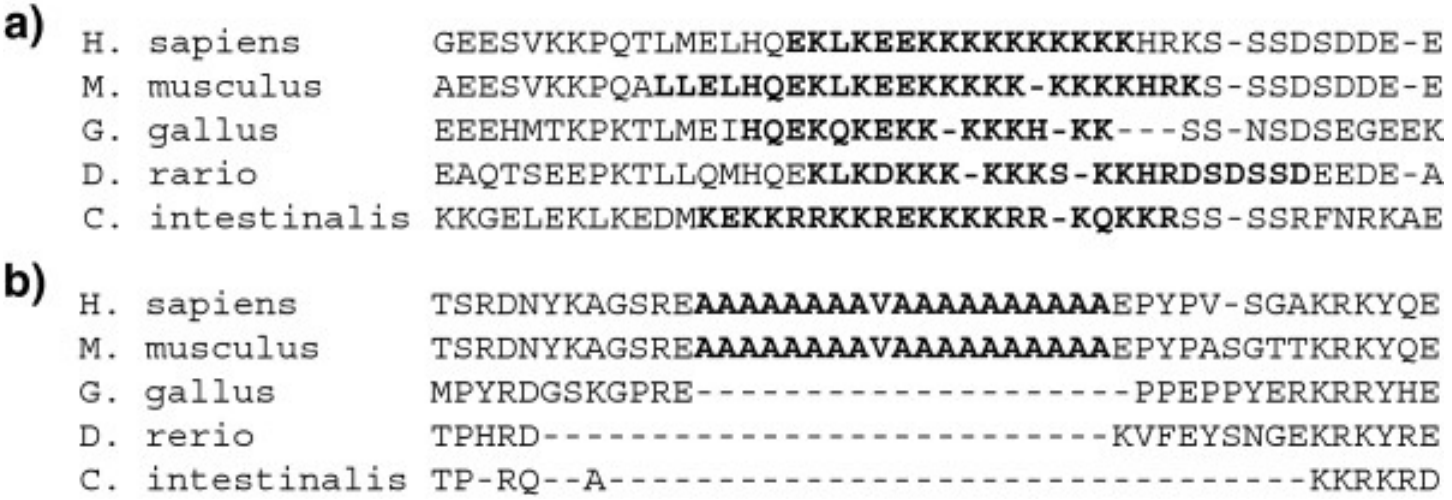


BLAST
- 3 letter word for proteins
- 11 letter word for DNA

BLAST algorithm

e.g. heuristic

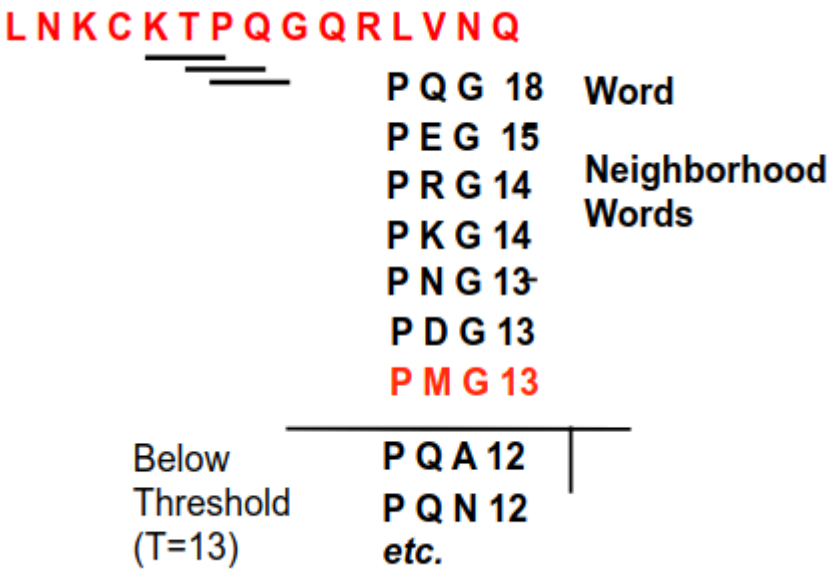
- Remove low complexity regions and repeats from query
- Make word list of query (protein k = 3, DNA k = 11)
- Identify high scoring words, above a threshold T, for word list



Low complexity-regions (bold) increase chances of spurious hits. Masking them improves signal-to-noise ratio.

BLAST algorithm

- Remove low complexity regions and repeats from query
- Make word list of query (protein k = 3, DNA k = 11)
- Identify high scoring words, above a threshold T, for word list



BLOSUM62 matrix

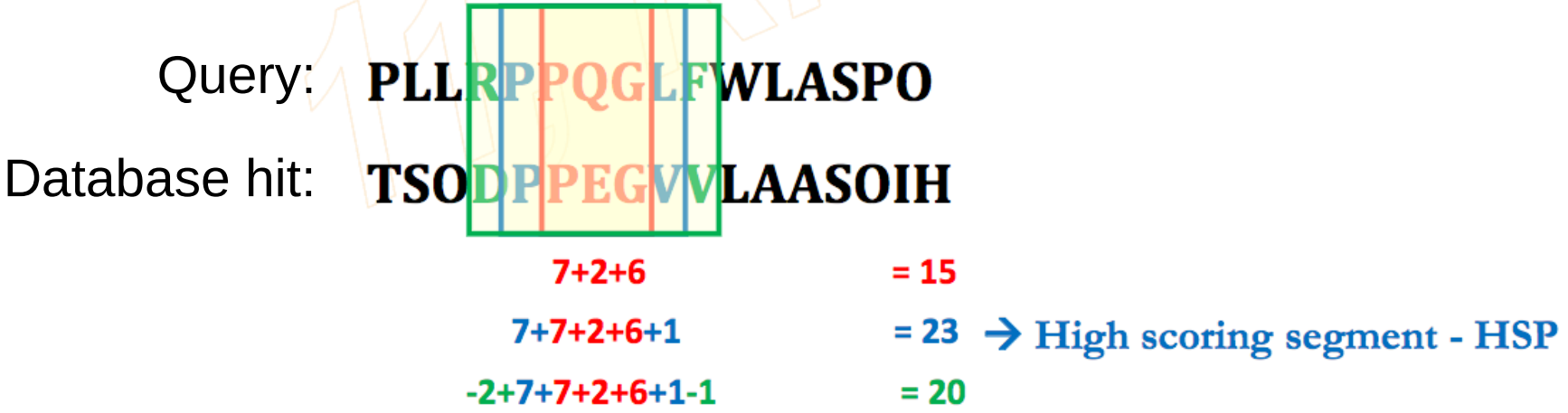
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Ala A	4																			
Arg R	-1	5																		
Asn N	-2	0	6																	
Asp D	-2	-2	1	6																
Cys C	0	-3	-3	-3	9															
Gln Q	-1	1	0	0	-3	5														
Glu E	-1	0	0	2	-4	2	5													
Gly G	0	-2	0	-1	-3	-2	-2	6												
His H	-2	0	1	-1	-3	0	0	-2	8											
Ile I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

7(P) +5(Q) +6(G)=18
7(P) +2(Q-E) +6(G)=15

Word score = sum of BLOSUM62 scores
BLOSUM62 are alignments with <62% identity

BLAST algorithm

- Find high scoring words in database (query is tree of words)
- Extend the matches (seeds) to high-scoring segment pairs (HSPs) in both directions until alignment score drops more than X below best alignment score



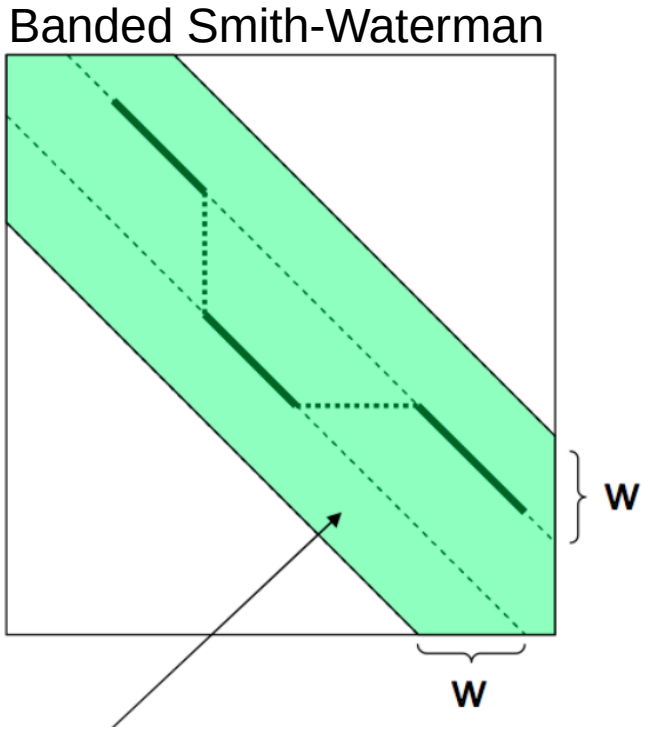
Extension is done separately for the left and right side!
Extension is 90% of computing BLAST result

BLAST2

BLAST2 – allows for gaps, adopts a lower neighborhood word score threshold, exact matched regions within ungapped distance A from each other are joined and then extended. Regions with gaps are joined with banded Smith-Waterman

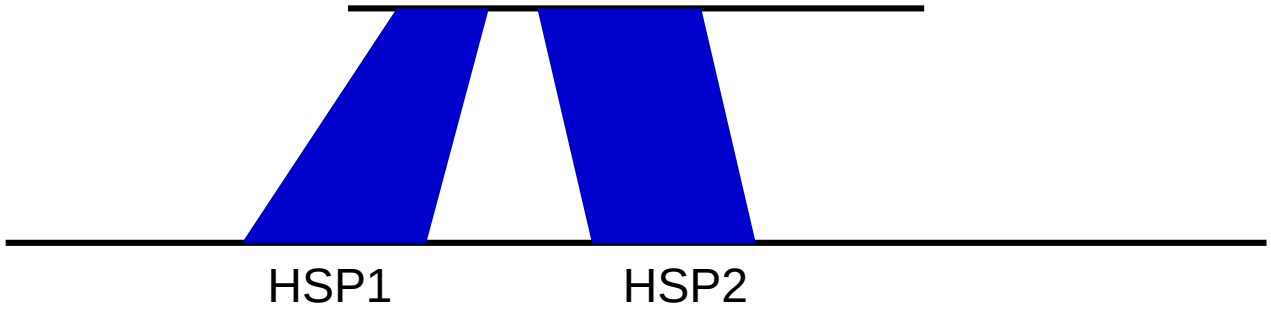
Diagonal: no gaps needed

Off diagonal: gaps needed



BLAST algorithm

- List and evaluate significance of HSPs
- Join HSP into longer alignment
- Show the gapped Smith-Waterman local alignments of the query and each of the matched database sequences.



BLAST 'hit'

Question: Is the alignment score obtained significantly higher than one would expect from two unrelated sequences?

Answer: We need to know the specific distribution that **random** alignment scores follow to be able to compute this!

Ungapped optimal local alignment scores follow a Gumbel extreme value distribution (EVD).

BLAST significance

Question: Is the alignment score obtained significantly higher than one would expect from two unrelated sequences?

Using the Gumbel distribution, the probability of a score greater than x is:

$$P(S \geq x) = 1 - \exp\left(-K m n e^{-\lambda x}\right)$$

where K and λ are parameters that are fit to the distribution of ungapped alignment scores of randomized database, m and n are the lengths of the query and target

https://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html

BLAST significant hit

$$P(S \geq x) = 1 - \exp\left(-K m n e^{-\lambda x}\right)$$

The expected value (E) is the number of alignments with scores greater than or equal to S, that are expected to occur by chance in a database

$$E = K m n e^{-\lambda S}$$

$$S' = \frac{\lambda S - \ln(k)}{\ln(2)}$$

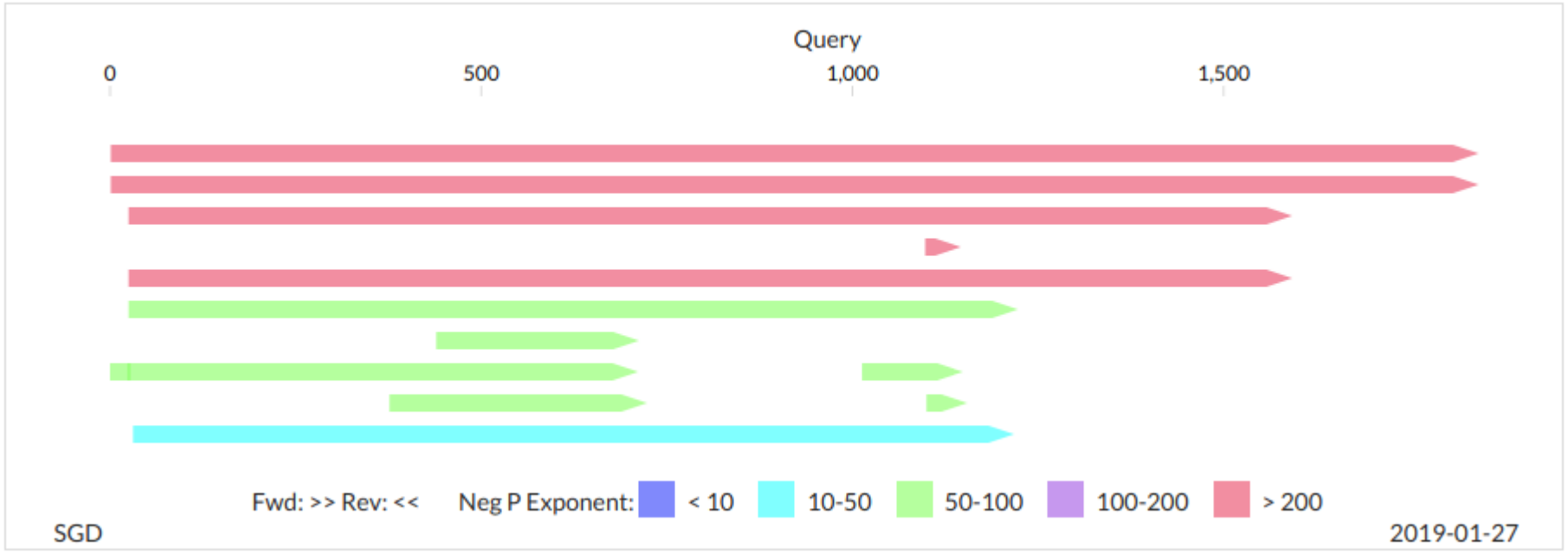
Two kinds of scores:
Raw scores (S): calculated from substitution matrix
Bit scores (S'): Comparable between searches, normalized for use of different scoring matrices and different database sizes

Parameters

Options: For descriptions of BLAST options and parameters, refer to the BLAST documentation at NCBI.

Output format:	gapped alignments	
Comparison Matrix:	BLOSUM62	
Cutoff Score (E value):	0.01	
Word Length (W value):	default	Default = 11 for BLASTN, 3 for all others
Expect threshold (E threshold):	default	
Number of best alignments to show:	50	
Filter options:	◉ On ○ Off	DUST file for BLASTN, SEG filter for all others

Graphic of HSPs



Summary text

Zheng Zhang, Scott Schwartz, Lukas Wagner, and WebbMiller (2000), "A greedy algorithm for aligning DNA sequences", JComput Biol 2000; 7(1-2):203-14. **Query=** UserInputSequence (1,842 letters)

Database: Sc_nuclear_chr.fsa; Sc_mito_chr.fsa;2-micron_chr.fsa

18 sequences; 12,163,423 total letters

Sequences producing significant alignments:	Score (bits)	E value
ref|NC_001136| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	3402	0
ref|NC_001146| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	3147	0
ref|NC_001144| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	701	0
ref|NC_001133| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	651	0
ref|NC_001137| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	209	2e-53
ref|NC_001142| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	204	7e-52
ref|NC_001134| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=ge...	195	4e-49

Hit 5

>ref|NC_001137| [org=Saccharomyces cerevisiae] [strain=S288C] [moltype=genomic] [chromosome=V]

Length = 576,874

Score = 209 bits (113), Expect = 2e-53 [Retrieve Sequence / Genome Browser]

Identities = 867/1228 (70%), Gaps = 63/1228 (5%), Frame = +1 / +1

```
Query: 25     GCTATCGGTATCGATTTAGGTACAACCTACTCTTGTGTTGCTACTTACG-AATC-CT-CC 81
              ||| | ||||| |||||||||||||||||| || |||||||||  ||  | || |  |
Sbjct: 364598 GCTGTTGGTATTGATTTAGGTACAACCTATTCATGTGTTGCTCATTTTGCAAACGATAGG 364657

Query: 82     GTTGAAATTATTGCCAACGAACAAGGTAACAGAGTCACCCCATCTTTCGTTGCTTTCACT 141
              |||||||||||| || ||||| |||||||| |||   || || ||||  || ||||| |||
Sbjct: 364658 GTTGAAATTATCGCTAACGATCAAGGTAATAGAACGACGCCTTCTTATGTGGCTTTTACT 364717
```

.....

Score = 54.7 bits (29), Expect = 8e-07 [Retrieve Sequence / Genome Browser]

Identities = 195/275 (70%), Gaps = 11/275 (4%), Frame = +1 / +1

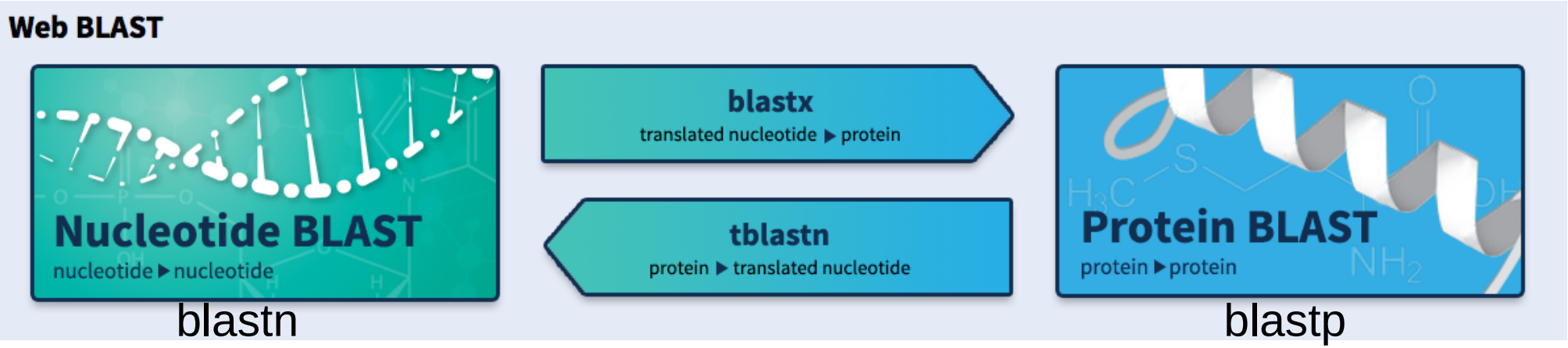
```
Query: 439   ACTGTCCCAGCTTACTTTAACGACGCTCAAAGACAAGCTACCAAGGATGCCGGTGCCATT 498
              || || || |||||||| || || || ||||||||||||| || || || ||    |||
Sbjct: 95139 ACCGTTCCTGCTTACTTCAATGATGCCCAAAGACAAGCTACTAAAGACGCAGGACAAATT 95198

Query: 499   TCTGGTTTGAACGTTTTGCGTATCATCAACGAACCTACTGCCGCTGCTATTGCTTACGGT 558
                |||  | || || || ||| |  |||||||||| || || |||||  | |||||||||
Sbjct: 95199 ATTGGGCTTAATGTATTACGTGTTGTCAACGAACCAACAGCTGCTGCCCTAGCTTACGGT 95258

Query: 559   CTAGGTGCTGGTAAGTCCGAAAAGGAAAGACATGTTTTGATTTTCGATTTGGGTGGTGGT 618
              |||| |      || || ||     |||| ||| |  || ||||||| ||||| ||||||
Sbjct: 95259 CTAGATA-----AA-TCAGAGCCA-AAAGTCAT-TGCTG-TTTTCGACTTGGGCGGTGGT 95309
```

.....

BLAST flavors



blastn blastp

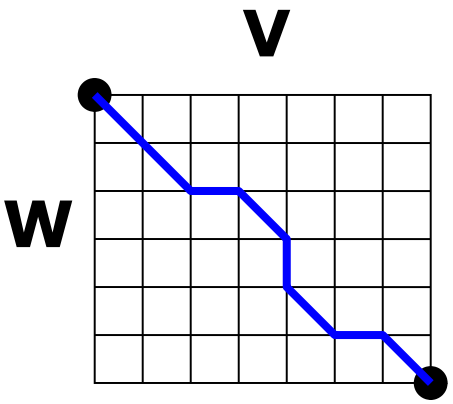
Similar algorithms

- BLAT (Blast Like Alignment Tool), 50-500x faster than BLAST, BLAT indexes database rather than query
- FASTA (predecessor to BLAST without removal of low complexity and no threshold for word scores), faster but less sensitive than BLAST
- PSI-BLAST (position specific iterative), closely related proteins are used to generate a "profile" sequence, which is then queried against a protein database, and the process is repeated.

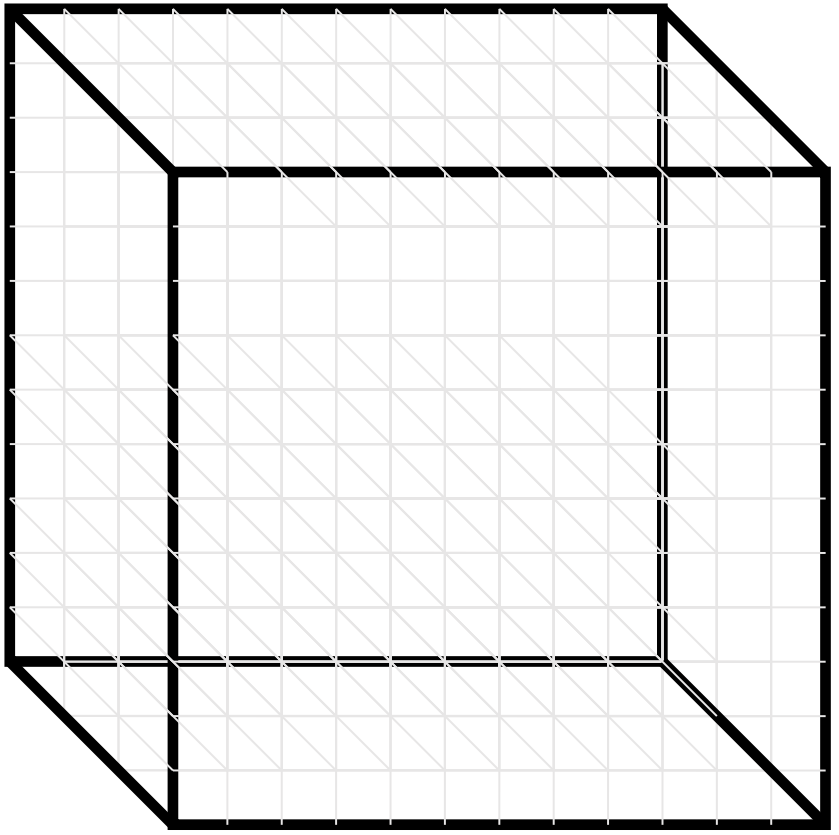
Multiple Sequence Alignment (MSA)

Generalized dynamic programming is not practical:

2D vs 3D alignment grid



2D table



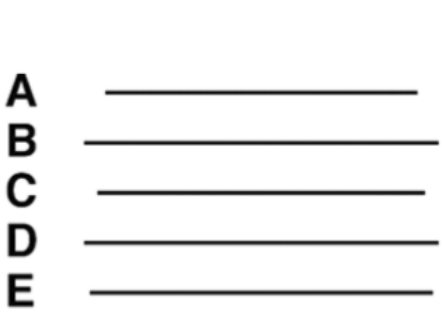
3D graph

Multiple sequence alignments use heuristic approaches

Types of MSA algorithms

- Progressive: ClustalW
- Iterative: Muscle
- Consistency Based: Coffee and Probcons
- Hidden Markov models: HMMER

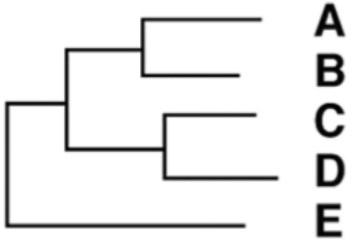
Clustalw – progressive alignment

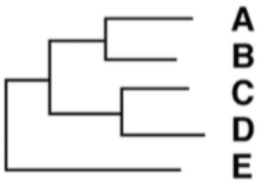


all individual pairwise alignment and construction of distance matrix

	A	B	C	D	E
A	–				
B	11	–			
C	20	30	–		
D	27	36	9	–	
E	30	33	20	27	–

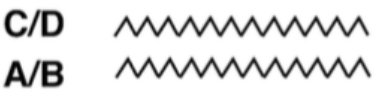
calculating a guide tree; C & D the closest pair; A & B the next closest pair





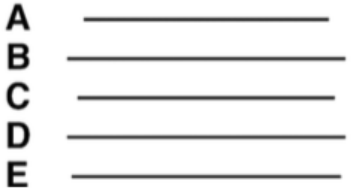
aligning C/D and A/B separately using dynamic programming

C/D and A/B alignments reduced to consensus sequences which are aligned to each other

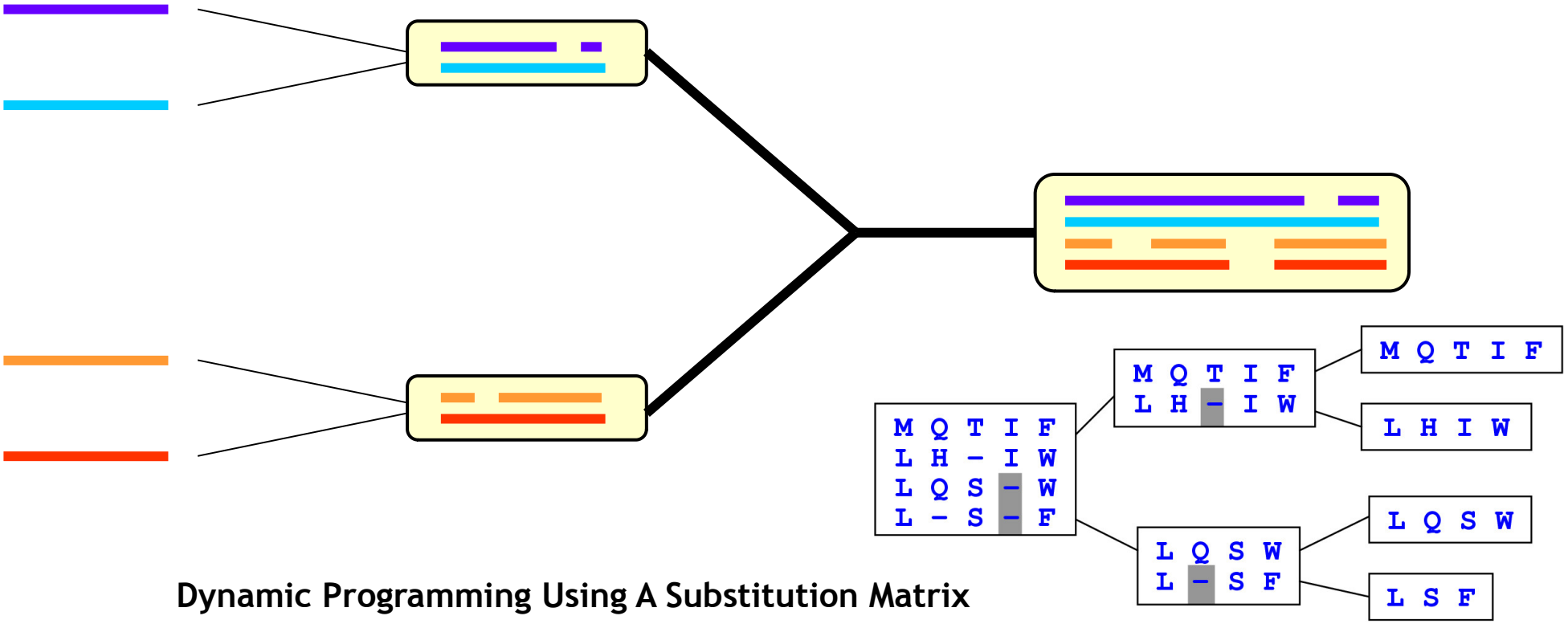


creating a new consensus for C/D/A/B which aligns with E

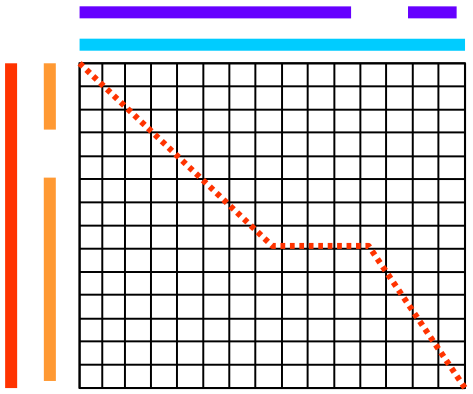
completing alignment



Progressive alignment



Dynamic Programming Using A Substitution Matrix



Problems with progressive alignments

- A major limitation is the “greedy” nature of the algorithm: it depends on initial pairwise alignment.
- Once gaps introduced in the early steps of alignment, they are fixed. Any errors made in these steps cannot be corrected. This problem of “once an error, always an error” can propagate throughout the entire alignment.
- The final alignment result is also influenced by the order of sequence addition
- The final alignment could be far from optimal.

Iterative and consistency alignment

- Progressive: ClustalW
- Iterative: similar to progressive methods but repeatedly realign the initial sequences as well as adding new sequences to the growing MSA
- Consistency Based: attempt to find the optimal multiple sequence alignment given multiple different alignments of the same set of sequences
- Hidden Markov models: can assign likelihoods to all possible combinations of gaps, matches, and mismatches to determine the most likely MSA or set of possible MSAs

Benchmarking pairwise alignments

STEP 1: simulate sequence divergence – we know the correct alignment

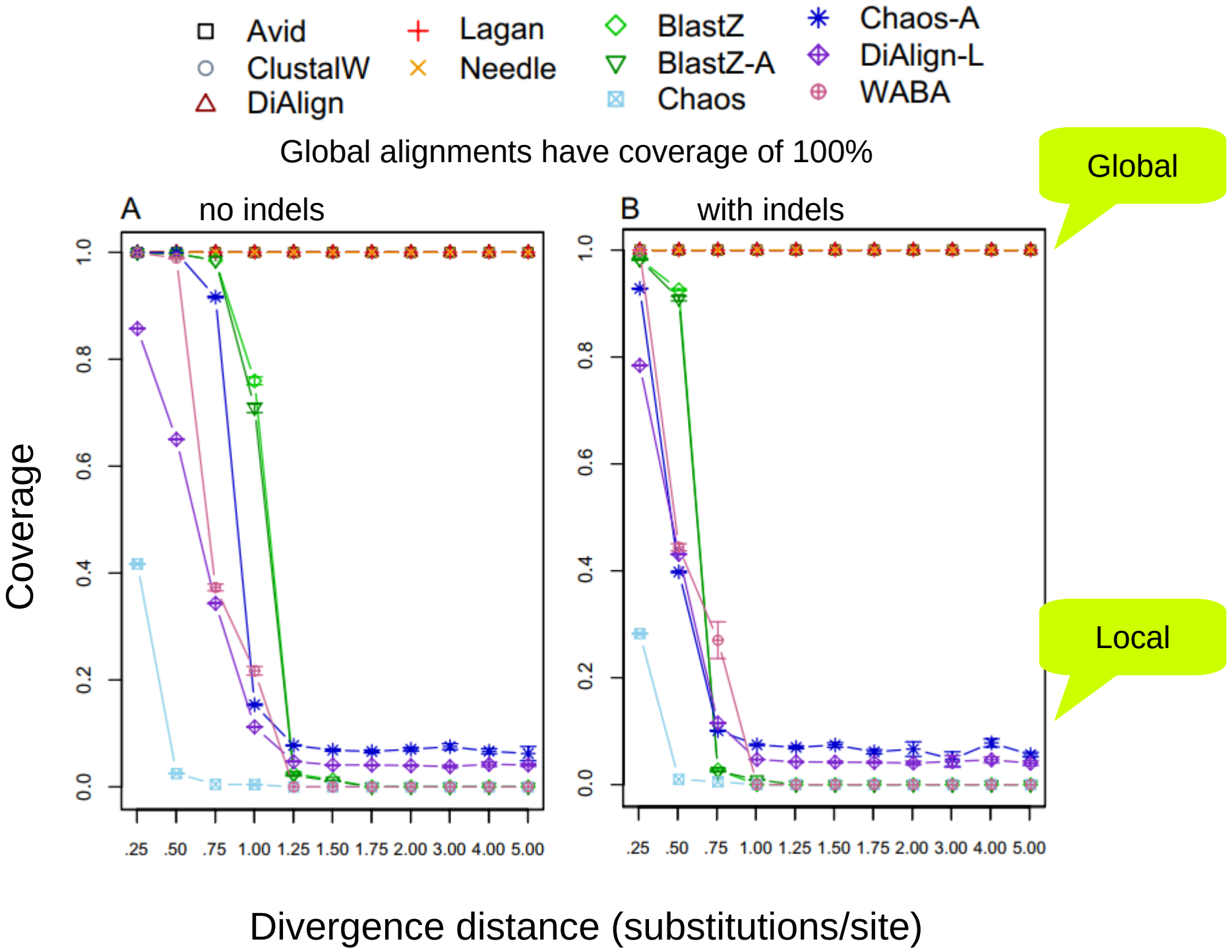
Table 1: Summary of parameters used in simulations of noncoding sequence evolution.

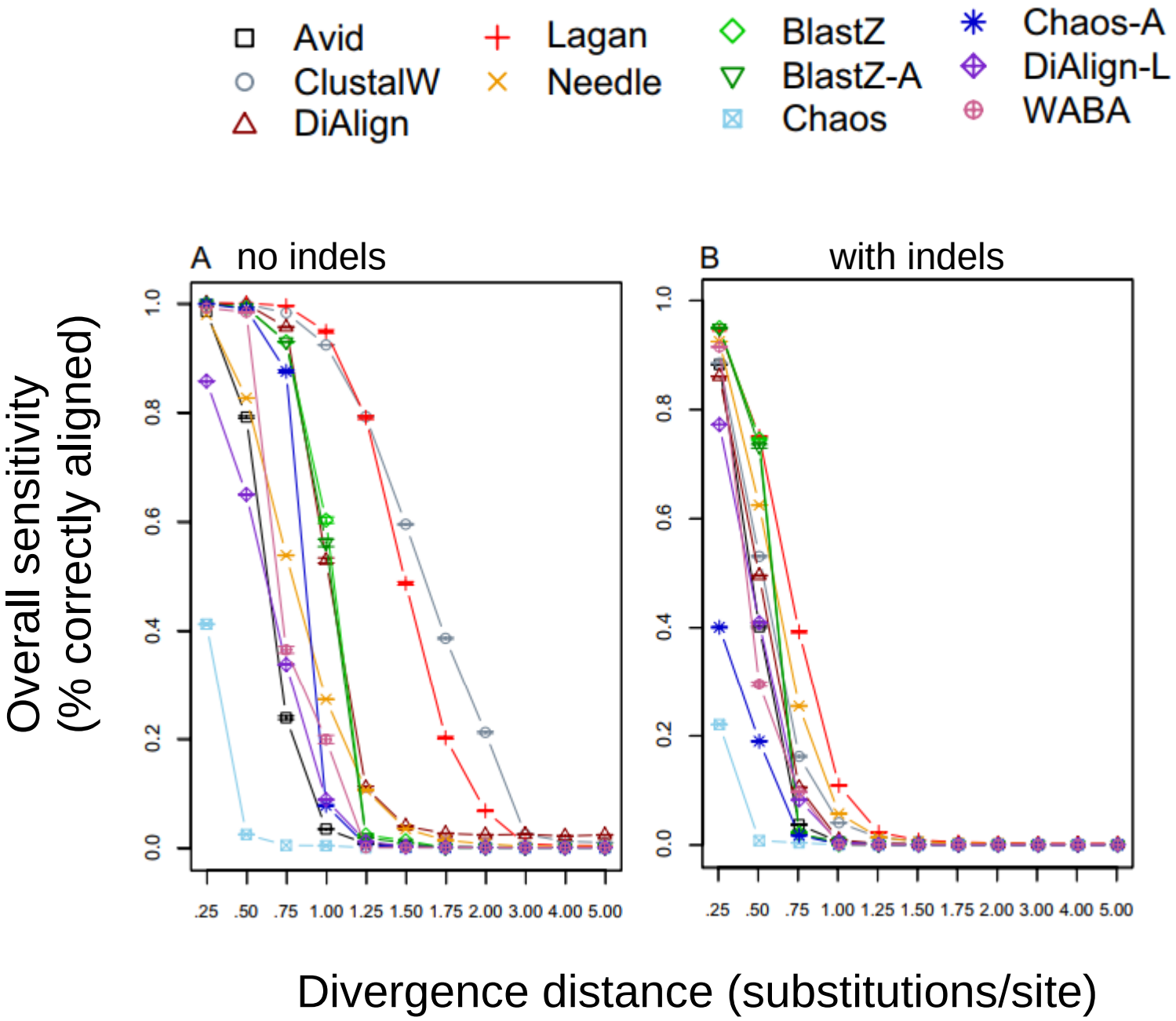
Parameter	Value	Source
Sequence length	10 Kb	D. mel
AT : GC	60 : 40	*Drosophila spp.*
Transition / Transversion Bias	2	*Drosophila spp.*
Substitution model	HKY85	-
Point substitutions : Indels	10 : 1	*Drosophila spp.*
Indel spectrum	-	*D. mel*
Median constrained block length	18 bp	*D. mel* vs. *D. vir*
Mean density of constrained blocks	0.2	*D. mel* vs. *D. vir*

STEP 2: Align simulated sequences with different alignment program

STEP 3: Compare their performance

Pollard, D.A., Bergman, C.M., Stoye, J., Celniker, S.E. and Eisen, M.B., 2004. Benchmarking tools for the alignment of functional noncoding DNA. BMC bioinformatics, 5(1), p.6.


Avid
ClustalW
DiAlign
Lagan
Needle
BlastZ
BlastZ-A
Chaos
Chaos-A
DiAlign-L
WABA
Global alignments have coverage of 100%
A no indels
B with indels
Coverage
0.0
0.2
0.4
0.6
0.8
1.0
.25 .50 .75 1.00 1.25 1.50 1.75 2.00 3.00 4.00 5.00
Divergence distance (substitutions/site)
Global
Local


Avid
ClustalW
DiAlign
Lagan
Needle
BlastZ
BlastZ-A
Chaos
Chaos-A
DiAlign-L
WABA
A no indels
B with indels
Overall sensitivity
(% correctly aligned)
0.0
0.2
0.4
0.6
0.8
1.0
.25 .50 .75 1.00 1.25 1.50 1.75 2.00 3.00 4.00 5.00
Divergence distance (substitutions/site)

Exercises

1) How is the word length (k) expected to affect the sensitivity and speed of BLAST?
2) How many hash keys are needed for blastn with k=11 and for blastp with k=3
3) How would you change BLAST parameters if you were trying to find very distantly related homologs?

   Expect threshold (E-value)

   Comparison matrix (BLOSSUM62/BLOSSUM45)

   Word length (k)

   Filtering low complexity (on/off)
4) Whats the probability of observing a BLAST score greater than one observed with an E-value of 5?

Exercises

5) Given the BLOSUM62 scoring matrix, extend the seed to find an ungapped alignment until a drop in score. Write the alignment and score.

```
DKSQVDVIVLVGGSTKVQKLVTDY
seed        GGS
NNLWRNGWRLAGGSSIVQWSRHYA
```

BLOSUM 62 scoring matrix

(positive values are shaded)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**4**																			
R	-1	**5**																		
N	-2	0	**6**																	
D	-2	-2	**1**	**6**																
C	0	-3	-3	-3	**9**															
Q	-1	**1**	0	0	-3	**5**														
E	-1	0	0	**2**	-4	**2**	**5**													
G	0	-2	0	-1	-3	-2	-2	**6**												
H	-2	0	**1**	-1	-3	0	0	-2	**8**											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	**4**										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	**2**	**4**									
K	-1	**2**	0	-1	-3	**1**	**1**	-2	-1	-3	-2	**5**								
M	-1	-1	-2	-3	-1	0	-2	-3	-2	**1**	**2**	-1	**5**							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	**6**						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	**7**					
S	**1**	-1	**1**	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	**4**				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	**1**	**5**			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	**1**	-4	-3	-2	**11**		
Y	-2	-2	-2	-3	-2	-1	-2	-3	**2**	-1	-1	-2	-1	**3**	-3	-2	-2	**2**	**7**	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	**3**	**1**	-2	**1**	-1	-2	-2	0	-3	-1	**4**