EECS730: Introduction to Bioinformatics

Lecture 05: Index-based alignment algorithms

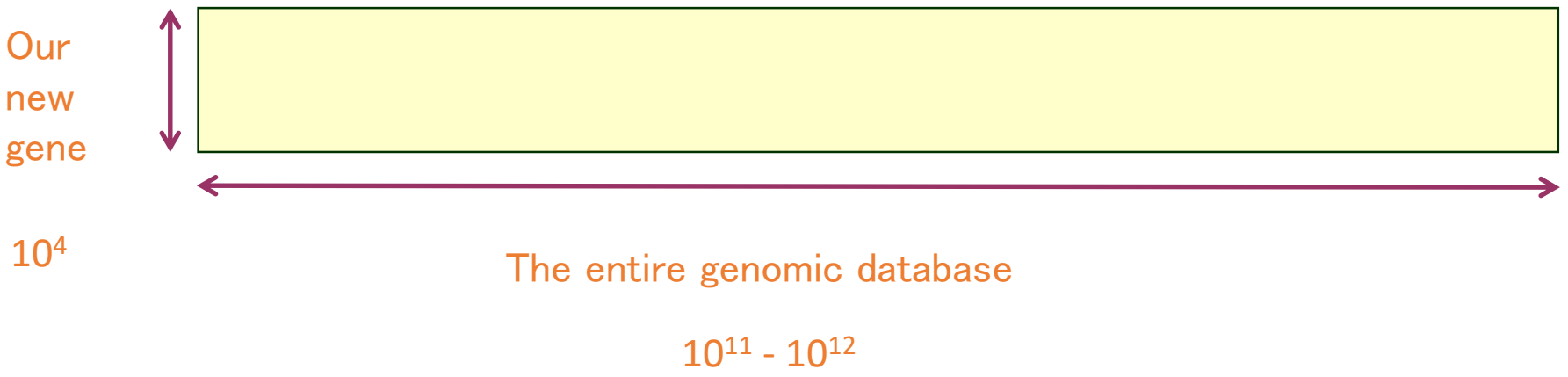
Slides adapted from Dr. Shaojie Zhang (University of Central Florida)

Real applications of alignment

- Database search
- Assume we have a gene g and a genome G, and we want to find the homolog of g in G
- Smith-Waterman algorithm (local alignment) would take O(g*G) time.
- Even more ambitious, if you want to search g against all homologs from a collection of genomes…
- As of 2014, there are 157,943,793,171nt (~160 billion) being registered in the database NCBI NT (non-redundant nucleotide).

Naïve Smith-Waterman

- Given a newly discovered gene,
  - Does it occur in other species?
  - How fast does it evolve?



Let’s try a shorter one…

>gi|57013850|sp|P69905.2|HBA_HUMAN RecName: Full=Hemoglobin subunit alpha; AltName: Full=Alpha-globin; AltName: Full=Hemoglobin alpha chain
MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKK
VADALTNA
VAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASV
STVLTSK YR

Different flavors of BLAST

- BLASTN: nucleotide to nucleotide
- BLASTP: protein to protein
- BLASTX: nucleotide to protein; finding protein that is encoded by the query
- TBLASTX: nucleotide to nucleotide; finding nucleotide sequences that code for the same/similar protein
- TBLASTN: protein to nucleotide; finding nucleotides that code for the query

Index-based searches

(BLAST- **B**asic **L**ocal **A**lignment **S**earch **T**ool)

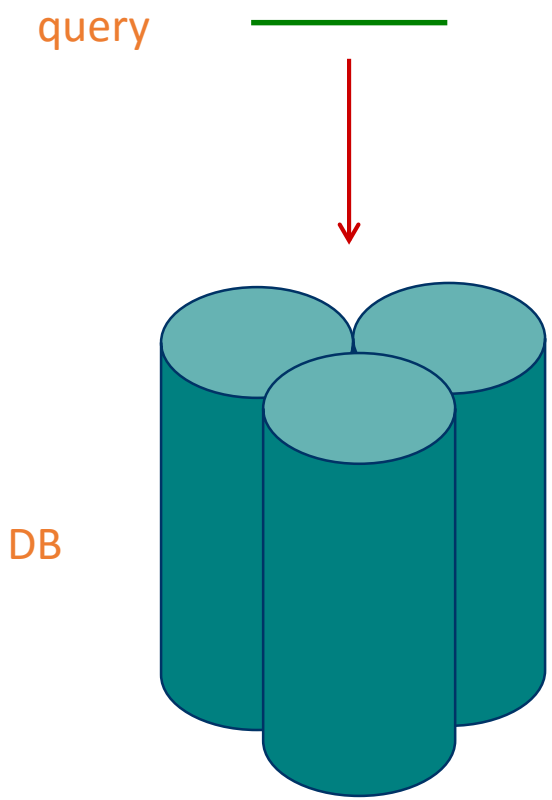
Main idea:

1. Construct a dictionary of all the **words** in the query

2. Initiate a local alignment for each word match between query and DB

Running Time: O(MN)

However, orders of magnitude faster than Smith-Waterman



Words

- A *k*-long sequence fragment
- It is also called *k*-mer
- Intuition: if we require a *k*-mer to initialize the alignment, we can expect to speedup the alignment by a^k times (a is the size of the alphabet, 4 for DNA and 20 for protein), given that the distribution of different *k*-mers are uniform

The indexing scheme

Dictionary:

All words of length *k* (11-13, tunable)

Alignment initiated between *k*-mer matches

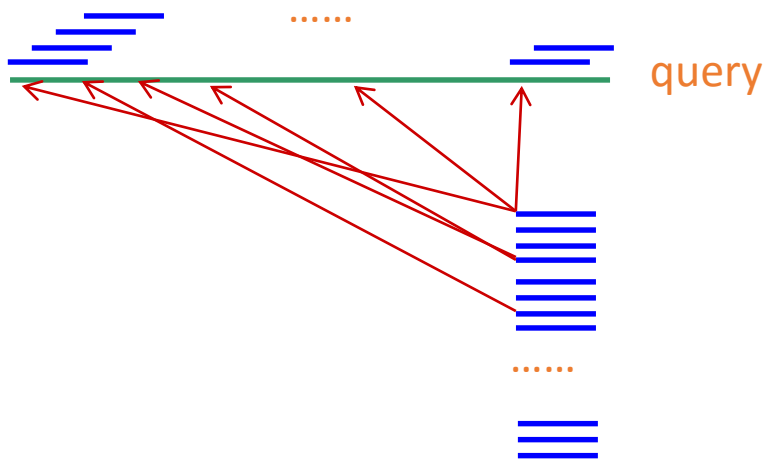
Alignment:

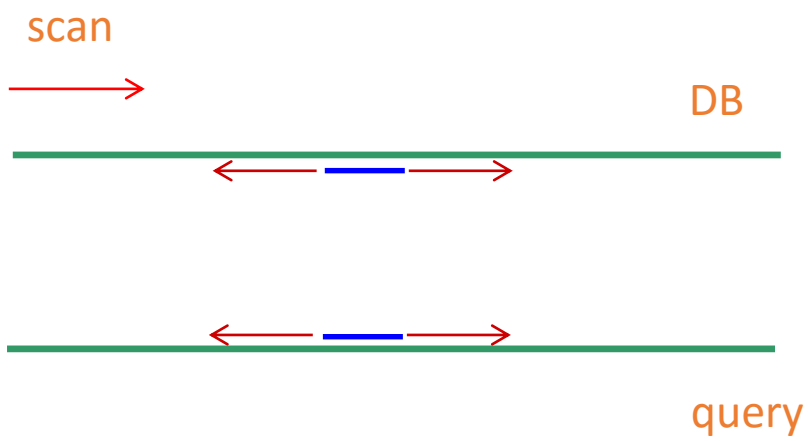
Ungapped extensions until score below statistical threshold

Gapped extension until score below statistical threshold

Output:

All local alignments with score > statistical threshold





Example:

k = 4

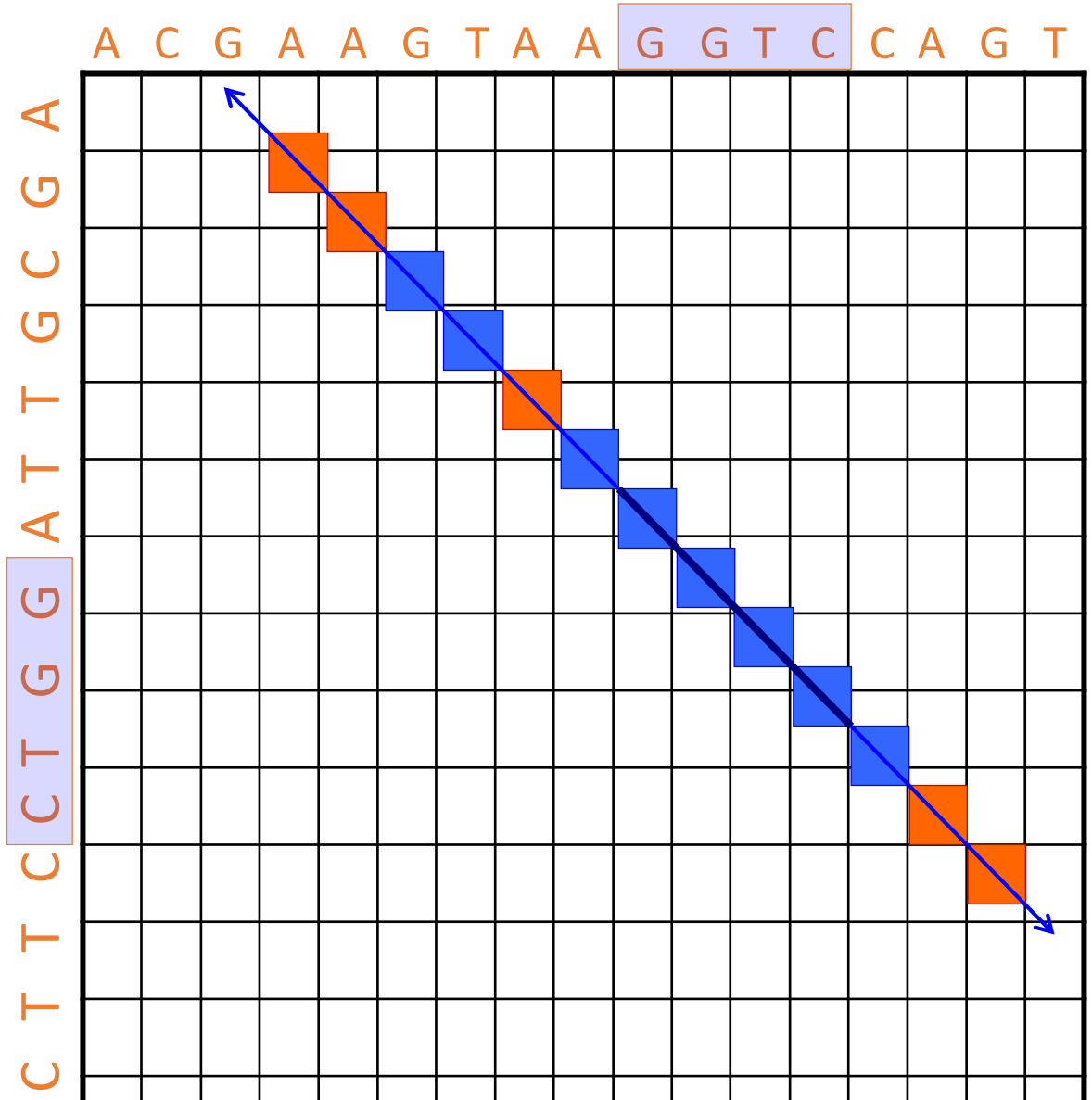
The matching word GGTC
initiates an alignment

Extension to the left and right with
no gaps until alignment falls <
T below best so far

Output:

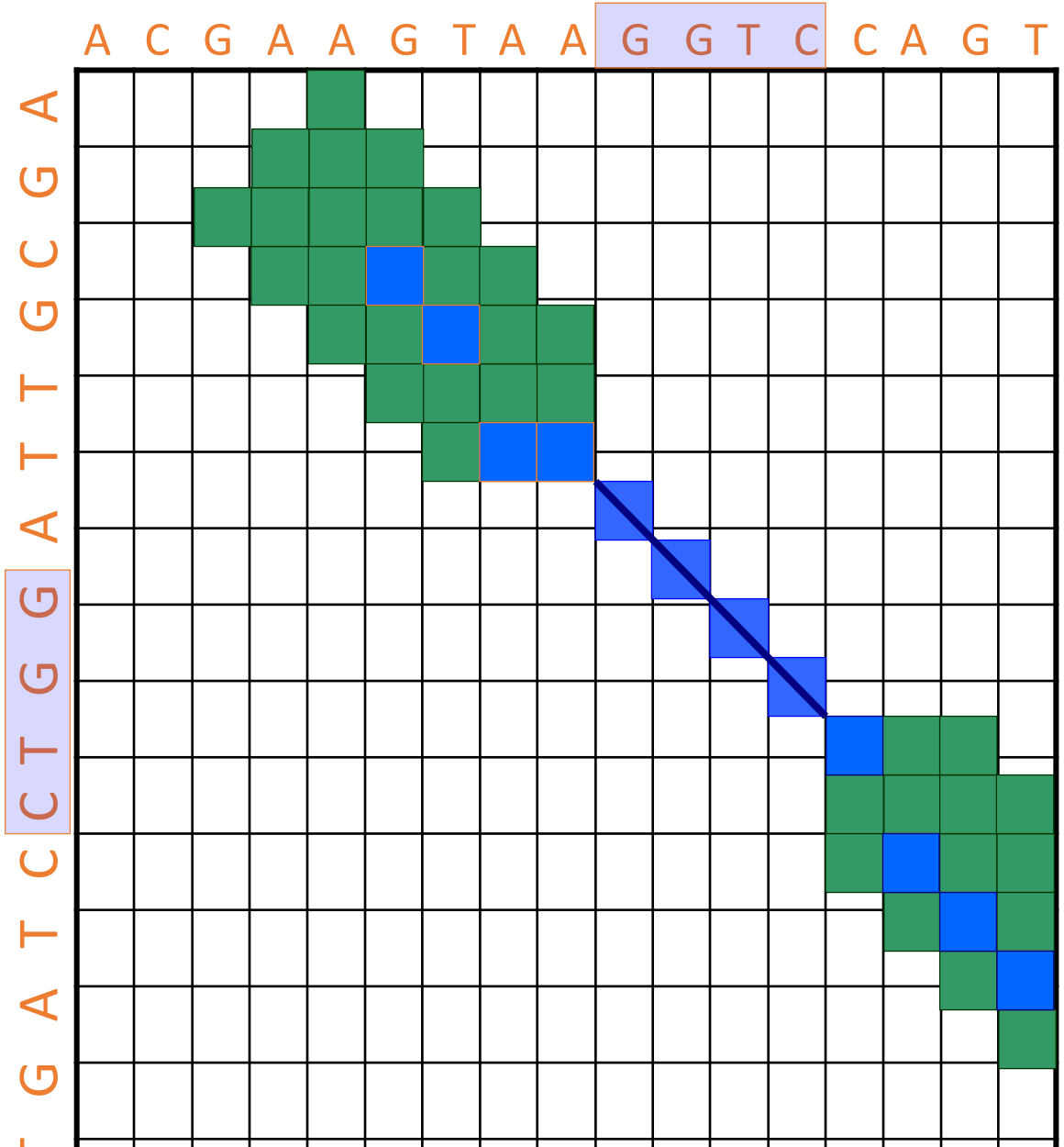
GTAAGGTCC

GTTAGGTCC



Gapped extensions

- Extensions with gaps in a band around anchor
- Terminates after significant score drop-off



Output:

GTAAGGTCC–AGT

GTTAGGTCCTAGT

Sensitivity/Speed tradeoff

	long words (k = 15)	short words (k = 7)
Sensitivity		✓
Speed	✓	

Table 3. Sensitivity and Specificity of Single Perfect Nucleotide K-mer Matches as a Search Criterion

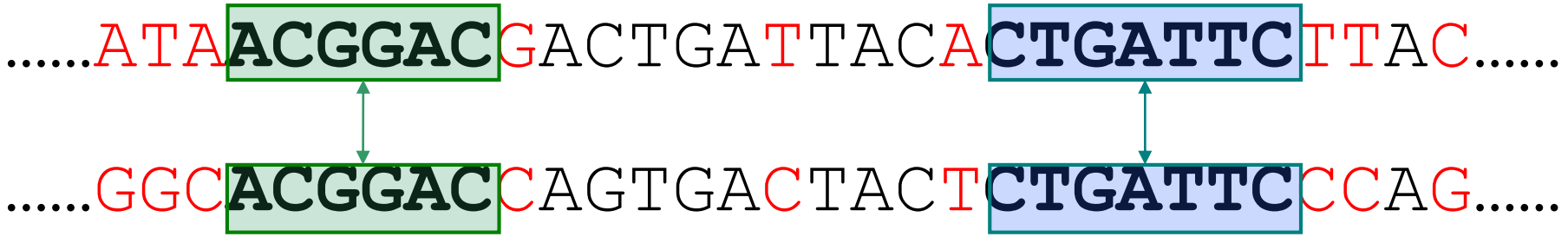
	7	8	9	10	11	12	13	14
A. 81%	0.974	0.915	0.833	0.726	0.607	0.486	0.373	0.314
83%	0.988	0.953	0.897	0.815	0.711	0.595	0.478	0.415
85%	0.996	0.978	0.945	0.888	0.808	0.707	0.594	0.532
87%	0.999	0.992	0.975	0.942	0.888	0.811	0.714	0.659
89%	1.000	0.998	0.991	0.976	0.946	0.897	0.824	0.782
91%	1.000	1.000	0.998	0.993	0.981	0.956	0.912	0.886
93%	1.000	1.000	1.000	0.999	0.995	0.987	0.968	0.957
95%	1.000	1.000	1.000	1.000	0.999	0.998	0.994	0.991
97%	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999
B. K	7	8	9	10	11	12	13	14
F	1.3e+07	2.9e+06	635783	143051	32512	7451	1719	399

(*A*) Columns are for K sizes of 7–14. Rows represent various percentage identities between the homologous sequences. The table entries show the fraction of homologies detected as calculated from equation 3 assuming a homologous region of 100 bases. The larger the value of K, the fewer homologies are detected.
(*B*) K represents the size of the perfect match. F shows how many perfect matches of this size expected to occur by chance according to equation 4 in a genome of 3 billion bases using a query of 500 bases.

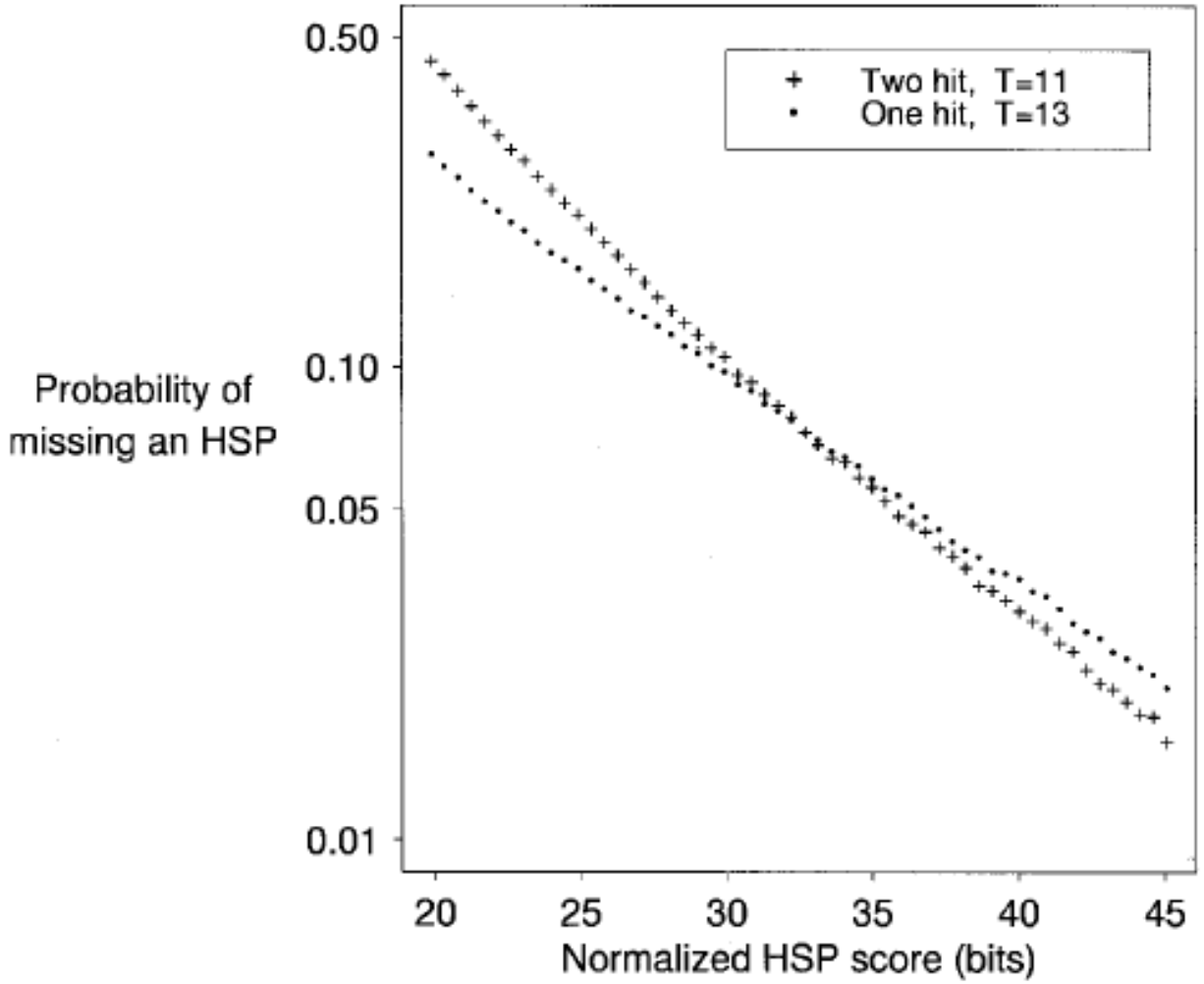
Kent WJ, Genome Research 2002

Using gapped seeds

- To allow variations in between



The BLAST configuration



Gapped seeds

Table 7. Sensitivity and Specificity of Multiple (2 and 3) Perfect Nucleotide K-mer Matches as a Search Criterion

	2,8	2,9	2,10	2,11	2,12	3,8	3,9	3,10	3,11	3,12
A. 81%	0.681	0.508	0.348	0.220	0.129	0.389	0.221	0.112	0.051	0.021
83%	0.790	0.638	0.475	0.326	0.208	0.529	0.339	0.193	0.099	0.045
85%	0.879	0.762	0.615	0.460	0.318	0.676	0.487	0.313	0.180	0.093
87%	0.942	0.866	0.752	0.611	0.461	0.809	0.649	0.470	0.305	0.177
89%	0.978	0.940	0.868	0.761	0.625	0.910	0.801	0.648	0.476	0.314
91%	0.994	0.980	0.947	0.884	0.787	0.969	0.914	0.815	0.673	0.505
93%	0.999	0.996	0.986	0.962	0.912	0.993	0.976	0.933	0.851	0.722
95%	1.000	1.000	0.998	0.993	0.979	0.999	0.997	0.987	0.961	0.902
97%	1.000	1.000	1.000	1.000	0.999	1.000	1.000	0.999	0.997	0.987

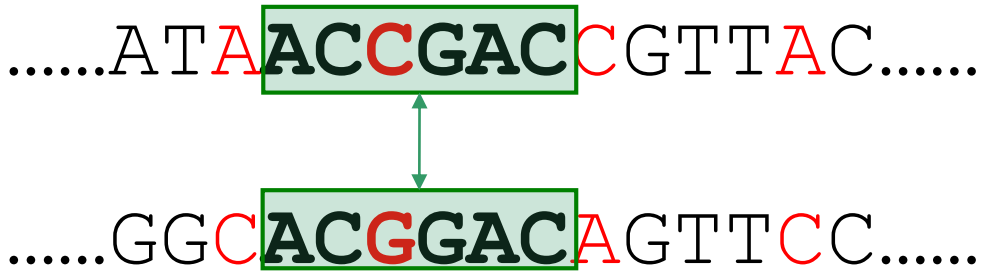
B. N,K	2,8	2,9	2,10	2,11	2,12	3,8	3,9	3,10	3,11	3,12
F	524	27	1.4	0.1	0.0	0.1	0.0	0.0	0.0	0.0

(*A*) Columns are for N sizes of 2 and 3 and K sizes of 8–12. Rows represent various percentage identities between the homologous sequences. The table entries show the fraction of homologies detected as calculated by equation 10. (*B*) N and K represent the number and size of the near-perfect matches, respectively. F shows how many perfect clustered matches expected to occur by chance according to equation 14 in a translated genome of 3 billion bases using a query of 167 amino acids.

Kent WJ, Genome Research 2002

Inexact matches

- Pre-building high-scoring *k*-mer pairs allowing mismatches
- Rescue homologs without long stretches of perfect matches
- More being used in protein alignment



Inexact matches

Table 5. Sensitivity and Specificity of Single Near-Perfect (One Mismatch Allowed) Nucleotide K-mer Matches as a Search Criterion

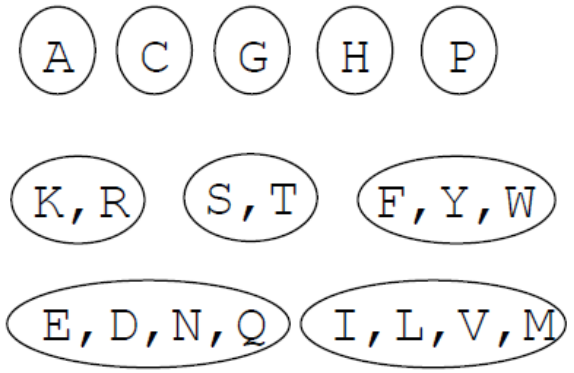
	12	13	14	15	16	17	18	19	20	21	22
A. 81%	0.945	0.880	0.831	0.721	0.657	0.526	0.465	0.408	0.356	0.255	0.218
83%	0.975	0.936	0.904	0.820	0.770	0.649	0.591	0.535	0.480	0.361	0.318
85%	0.991	0.971	0.954	0.900	0.865	0.767	0.719	0.669	0.619	0.490	0.445
87%	0.997	0.990	0.983	0.954	0.935	0.867	0.833	0.796	0.757	0.634	0.591
89%	1.000	0.997	0.995	0.984	0.976	0.939	0.920	0.897	0.872	0.775	0.741
91%	1.000	1.000	0.999	0.996	0.994	0.979	0.971	0.962	0.950	0.890	0.869
93%	1.000	1.000	1.000	0.999	0.999	0.996	0.994	0.991	0.988	0.963	0.954
95%	1.000	1.000	1.000	1.000	1.000	1.000	0.999	0.999	0.999	0.994	0.992
97%	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

B. K	12	13	14	15	16	17	18	19	20	21	22
F	275671	68775	17163	4284	1070	267	67	17	4.2	1.0	0.3

(*A*) Columns are for K sizes of 12–22. Rows represent various percentage identities between the homologous sequences. The table entries show the fraction of homologies detected as calculated by equation 6 assuming a homologous region of 100 bases. (*B*) K represents the size of the near-perfect match. F shows how many perfect matches of this size expected to occur by chance according to equation 7 in a genome of 3 billion bases using a query of 500 bases.

Kent WJ, Genome Research 2002

Reduced alphabet



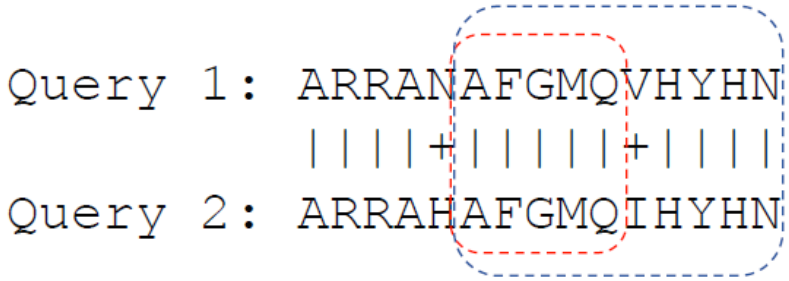


Table 1 A list of reduced amino acid alphabets

Alphabet	Size of the alphabet	Amino acid groups
all.20	20	P G E K R Q D S N T H C I V W Y F A L M
dssp.5	5	[AEHKQR] [FILMVWY] [CST] [DN] [GP]
dssp.10	10	[EKQR] [IV] [LY] F [AM] W [HT] C [DNS] [GP]
gbmr.4	4	G [ADEKNQRST] [CFHILMVWY] P
gbmr.10	10	G D N [AEFIKLMQRVW] Y H C T S P
hsdm.5	5	[LIVFMY] W C [DNTSKEQRAGP] H
sdm.6	6	[YFLIVM] C W [DNTSQKERAG] H P
murphy.5	5	[LVIMC] [ASGTP] [FYW] [EDNQ] [KRH]
murphy.10	10	A [KR] [EDNQ] C G H [ILVM] [FYW] P [ST]
td.5	5	[PG] [EKRQ] [DSNTHC] [IVWYF] [ALM]
td.10	10	P G [EKRQ] [DSN] T [HC] [IV] [WYF] A [LM]

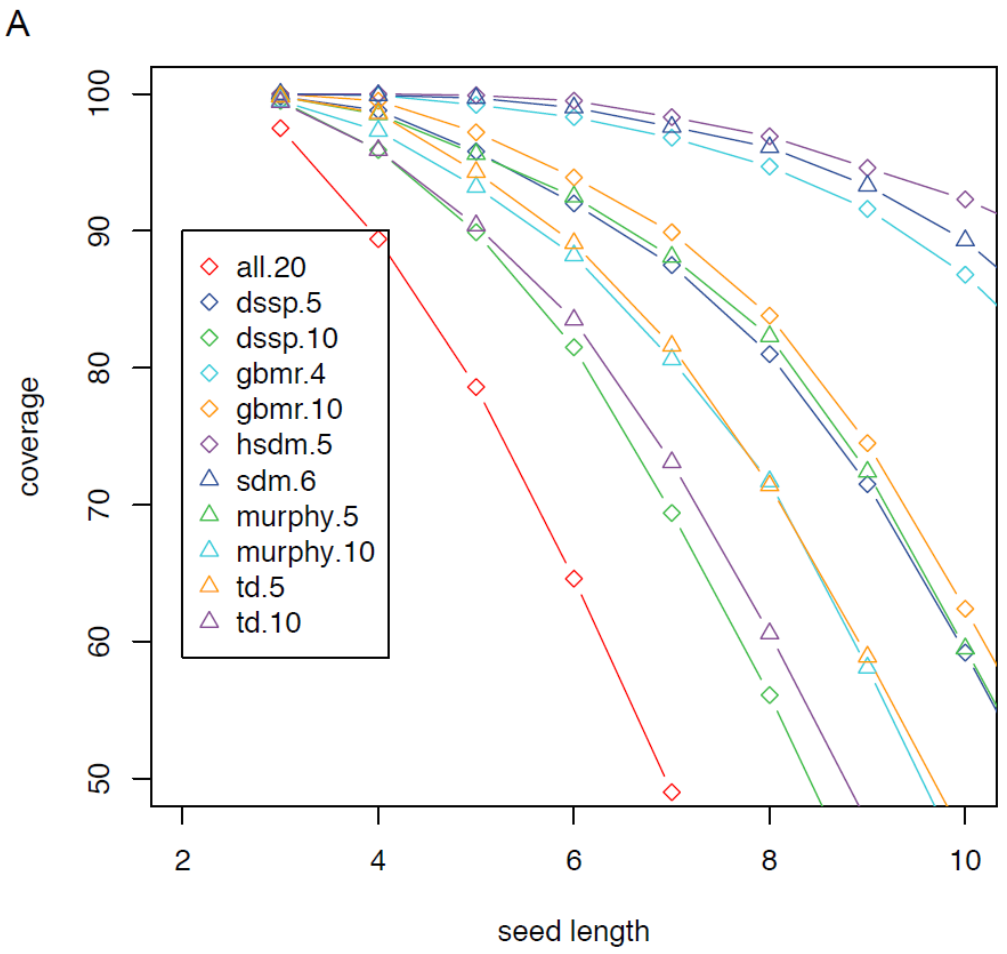
The alphabets were downloaded from http://www.rpgroup.caltech.edu/publications/supplements/alphabets/HP/Welcome.html.

Longest exact match (using 20 amino acids) = 5

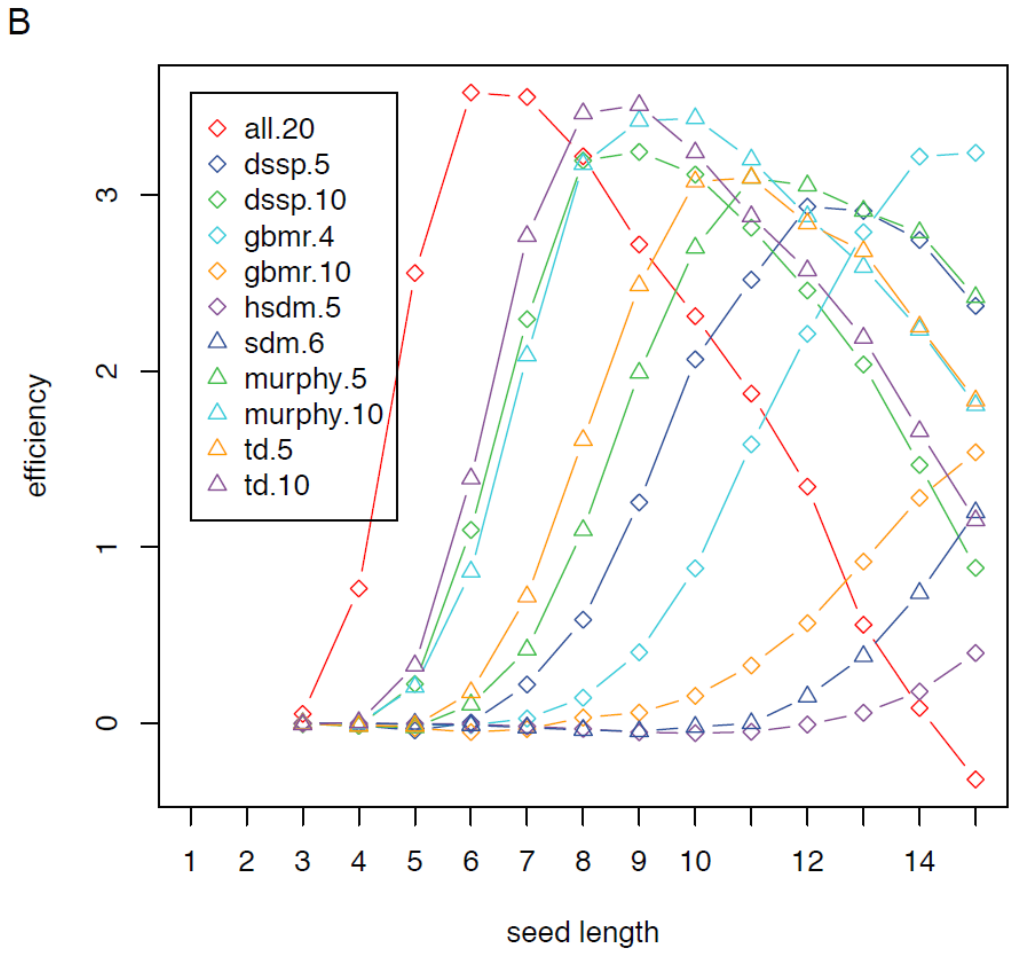
Longest "exact" match (using the reduced alphabet) = 10

Ye et al. *BMC Bioinformatics*, 2011

Choice of reduced alphabet in seeding



Percentage of related alignment sharing the seed

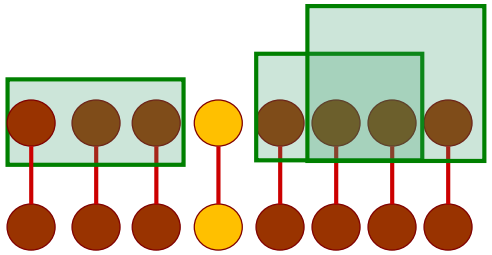


$$\log \frac{\text{Percentage of related alignment sharing the seed}}{\text{Percentage of unrelated alignment sharing the seed}}$$

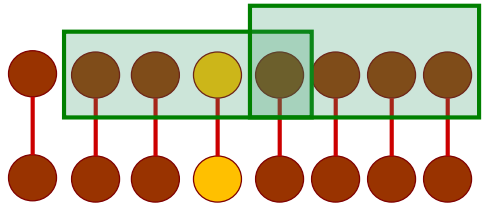
Ye et al. *BMC Bioinformatics*, 2011

Patterns

- Non-consecutive words
- Increase the probability of at least one hit while reduce the number of hits



3 seeds



2 seeds

pattern

No 5-mer perfect matches

On a 100-long 70% conserved region:

	Consecutive	Non-consecutive
Expected # hits:	1.07	0.97
Prob[at least one hit]:	0.30	0.47

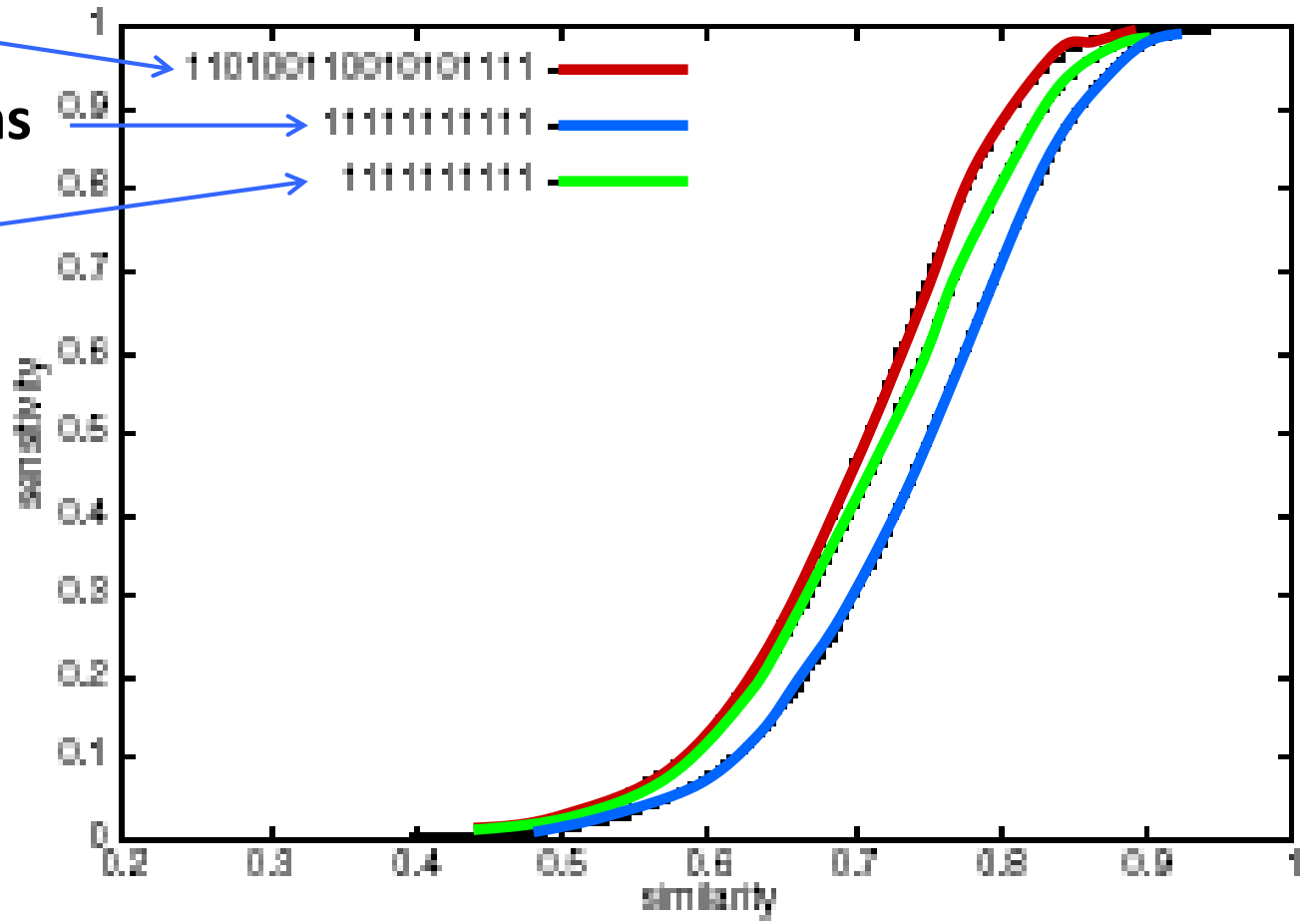
Patterns

Note that using patterns less seeds will be generated, so it is also faster!!!

11 positions

11 positions

10 positions



Variants of BLAST

- NCBI BLAST: search the universe http://www.ncbi.nlm.nih.gov/BLAST/
- MEGABLAST:
  - Optimized to align very similar sequences
    - Works best when $k = 4i \geq 16$
    - Linear gap penalty
- WU-BLAST: (Wash U BLAST)
  - Very good optimizations
  - Good set of features & command line arguments
- BLAT
  - Faster, less sensitive than BLAST
  - Good for aligning huge numbers of queries
- CHAOS
  - Uses inexact k-mers, sensitive
- PatternHunter
  - Uses patterns instead of k-mers
- BlastZ
  - Uses patterns, good for finding genes

BLAST statistics

- Score matrices used to seek local alignments of variable length should have a **negative expected score**. Otherwise the alignment will span over the entire sequence (good alignment score even for random sequences).

- $\sum p_{i*} p_{j*} S_{i,j} < 0$; p_i is the frequency of character i; p_j is the frequency of character j; and $S_{i,j}$ is the score for substituting character i with j.

Log odds score

- Let $S_{i,j}$ be the scaled (to integers to facilitate score computation) log-odds likelihood for substituting *i* with *j*.

- $S_{i,j} = \ln(q_{i,j} / p_{i*}p_j) / \lambda$; $q_{i,j}$ is the frequency of substituting *i* with *j*; λ is a scaling parameter that converts the score back to the "probability" space.

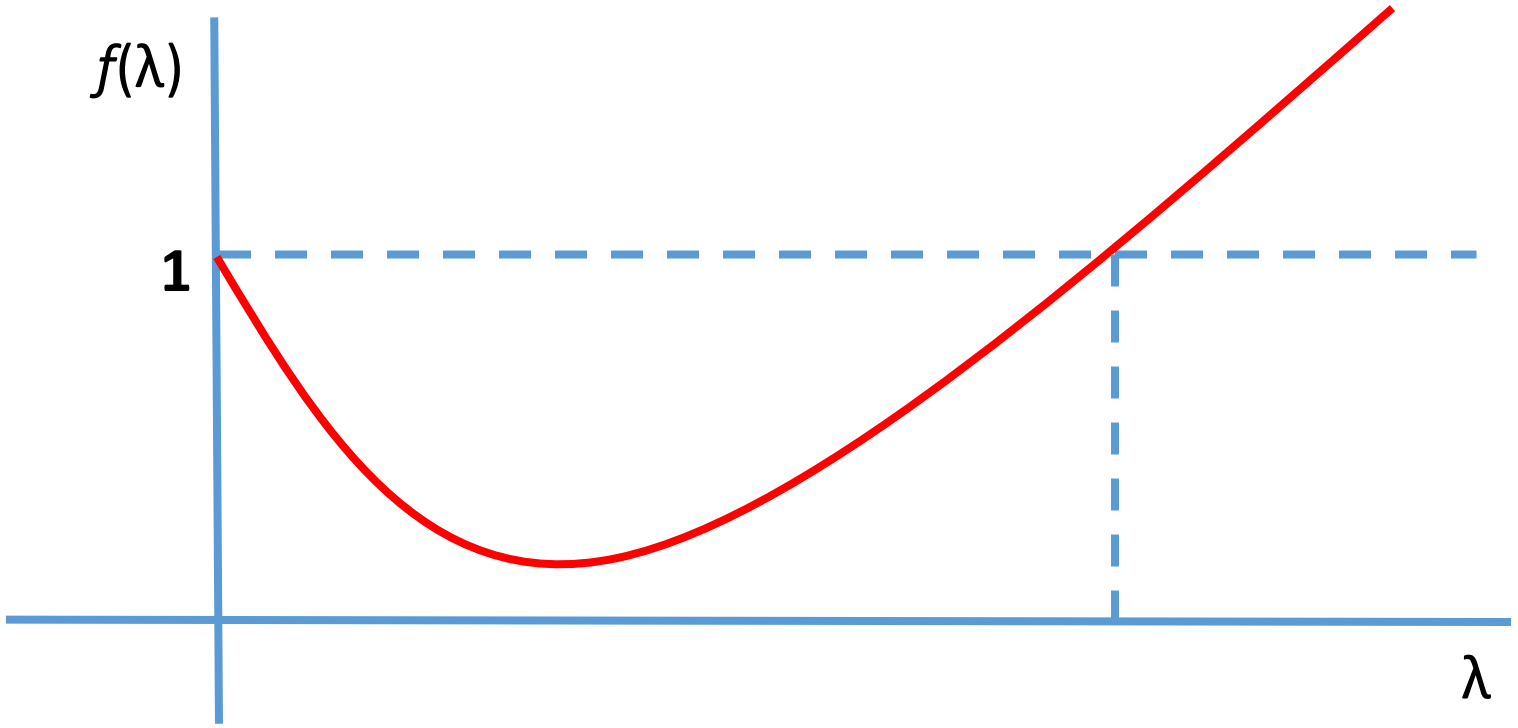
- Find the appropriate value for λ

Finding λ

- We know that the sum of all $q_{i,j}$ should be 1.
- Recall that $S_{i,j} = \ln(q_{i,j} / p_{i*}p_j) / \lambda$
- $q_{i,j} = p_{i*}p_{j*}e^{\wedge}(\lambda * S_{i,j})$; and $f(\lambda) = \sum p_{i*}p_{j*}e^{\wedge}(\lambda * S_{i,j}) = 1$.

- We know that λ is always a solution, i.e. $f(0) = 1$
- $f'(0) = \sum p_{i*}p_{j*} S_{i,j} < 0$ (the negative expected score assumption)
- $f''(\lambda) = \sum p_{i*}p_{j*} (S_{i,j})^{\wedge}2 * e^{\wedge}(\lambda * S_{i,j}) > 0$

Sketching the function



The *E*-value

- Given a particular scoring system, how many distinct local alignments with score ≥ *S* can one expect to find by chance from the comparison of two random sequence of lengths *m* and *n*? The answer, *E*(*S,m,n*), should depend upon *S*, and the lengths of the sequences compared.

- If we double the size of m, we will get twice more local alignments; if we double the size of n, we will also get twice more local alignments.

- *E*(*S,m,n*) is proportional to *m***n*

Frequency of observing a stretch of alignment

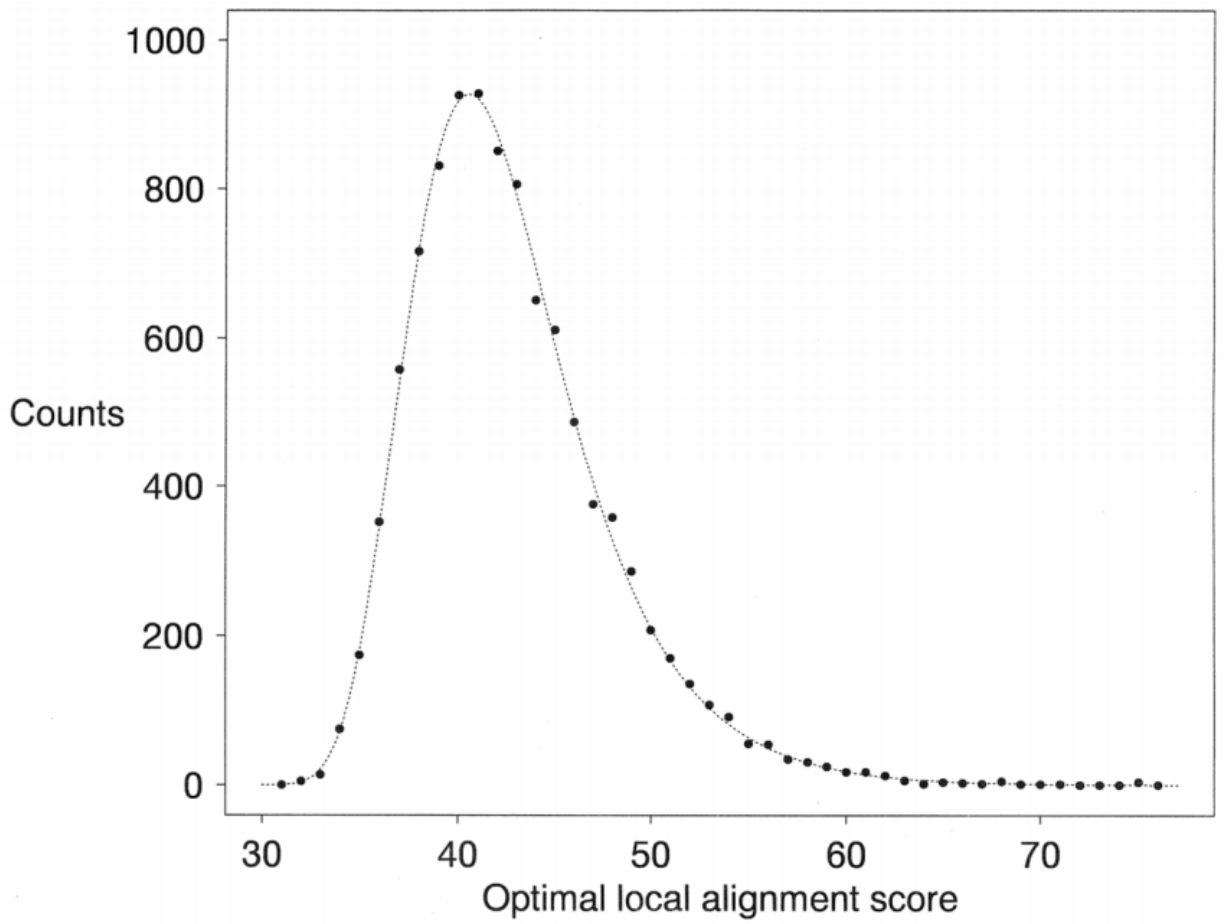
- $1/\Pi\ (q_{i,j} / p_{i*}p_j) = e\text{^}(-\log(\Pi\ (q_{i,j} / p_{i*}p_j)\ )) = e\text{^}(-\sum\log(q_{i,j} / p_{i*}p_j))$
- $e\text{^}(-\sum\log(q_{i,j} / p_{i*}p_j)) = e\text{^}(-\sum\ \lambda\ *S_{i,j}) = e\text{^}(-\lambda\ *\ \sum S_{i,j}) = e\text{^}(-\lambda\ *\ S)$
- $E(S,m,n)$ is proportional to $e\text{^}(-\lambda\ *\ S)$

The *E*-value

- $E(S,m,n)$ is proportional to $m*n$
- $E(S,m,n)$ is proportional to $e^{(-\lambda * S)}$
- $E(S,m,n) = K * m * n * e^{(-\lambda * S)}$
- K can be determined through simulation

With gaps

- The theory is provably valid only for local alignments without gaps.
- However, although no formal proof is available, random simulation suggests the theory remains valid when gaps are allowed, with sufficiently large gap costs.
- In this case, no analytic formulas for the statistical parameters λ and K are available, but these parameters may be estimated by random simulation.



https://wwwold.cs.umd.edu/class/fall2011/cmsc858s/Local_Alignment_Statistics.pdf