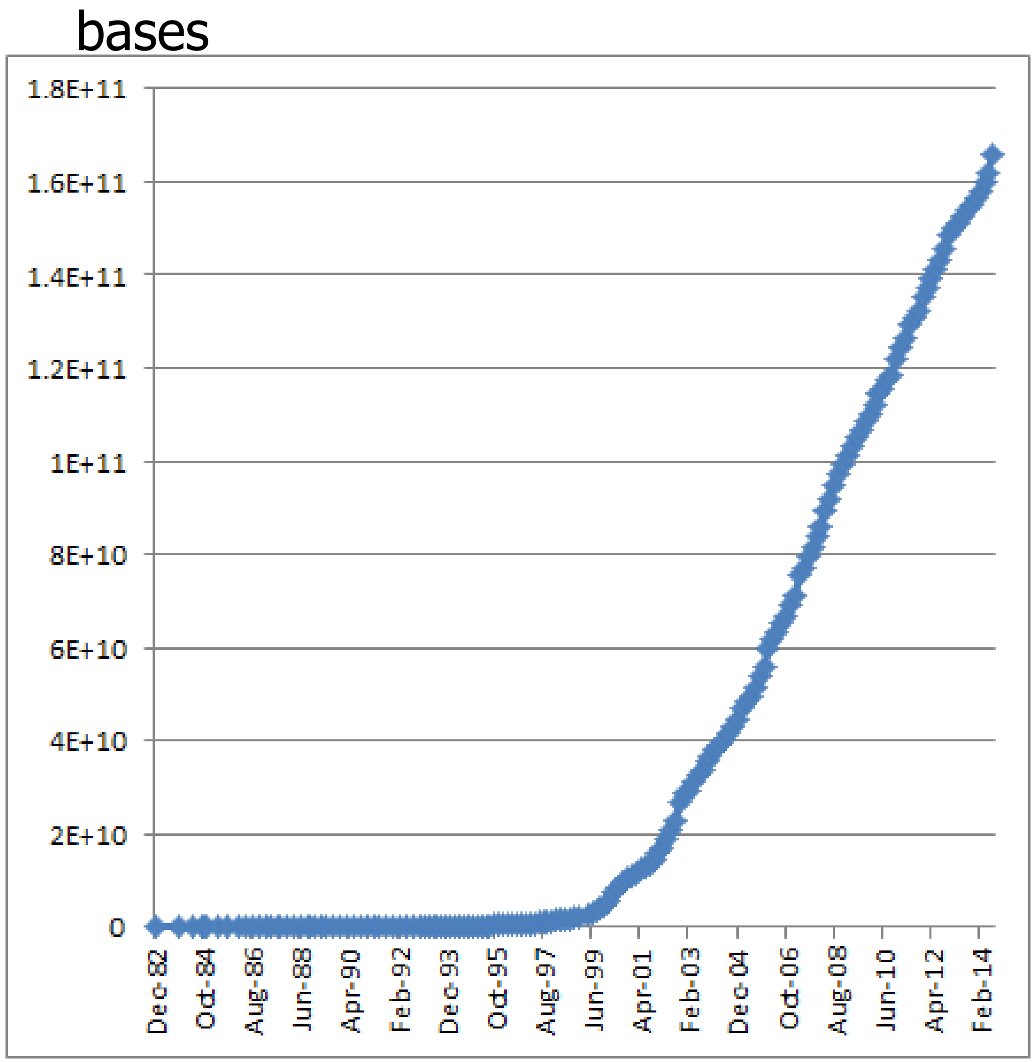
Combinatorial Methods in Bioinformatics

Searching biological database

Wing-Kin Sung, Ken 宋永健
ksung@comp.nus.edu.sg

Biological database

- Biological database stores known sequences of proteins, RNAs and DNAs.
- There are many biological database. The most common database is genbank.
- The size of biological database increases exponentially.
- In Dec 2019, genbank stores 215 millions annotated genes and 388 billions base pairs.



Why searching biological database?

- Example:
  - In November 2005, an unknown virus from a child in an Amish community was suspected to be an intestinal virus.
  - Through database searching, the unknown sequence was found to be a polio virus.
  - This is the first polio case in the U.S. since 1999 and it demonstrates the power of database search.
- In molecular biology, it is common to search query sequences on various databases since sequence similarity often leads to functional similarity.
- For instance, in 2002, GenBank alone served 100,000 search queries daily which had risen to over 140,000 queries daily by 2004.

Problem definition

- Consider a database D of genomic sequences (or protein sequences)
- Given a query string Q,
    - we look for a string S in D which is the closest match to the query string Q using some score function.
    - There are two choices of the score function:
        - **Semi-global alignment score**: the best possible alignment score between a substring A of S and Q
        - **Local alignment score**: the best possible alignment score between a substring A of S and a substring B of Q.

Example

- Database D:
  - S_1=ACTCGGATGGT
  - S_2=CACCAGTACGCA
  - S_3=TCAGGTGGTA
- Query Q=GGTTGG
- Assume match scores +1, mismatch/indel scores -1

- Solution: There are two hits with the highest score 4 (same solutions for both local or semi-global alignment)
  - S_1=ACTC$\underline{\text{GGATGG}}$T
  - S_2=CACCAGTACGCA
  - S_3=TCA$\underline{\text{GGTGG}}$TA

```
GGTTGG
GGATGG
```

```
GGTTGG
GGT-GG
```

Measurement of the goodness of a search algorithm

- Efficiency
  - Measure the running time of an algorithm.
- Sensitivity
  - Sensitivity is the ratio of the number of true positives found by the algorithm over the actual number of true positives.
  - An algorithm with 100% sensitivity finds all true positives.
- A good search algorithm should be both sensitive and specific

Different approaches

- Exhaustive search algorithms
    - E.g. Smith-Waterman Algorithm and BWT-SW
- Heuristic algorithms
    - E.g. FastA, BLAST, BLAT and PatternHunter
- Filter-based algorithms
    - E.g. QUASAR and LSH

Exhaustive search algorithm

- Input:
    - the database D (total length: n) and
    - the query Q (length: m)
- Output: all closest matches (based on local alignment or semi-global alignment)

Algorithm

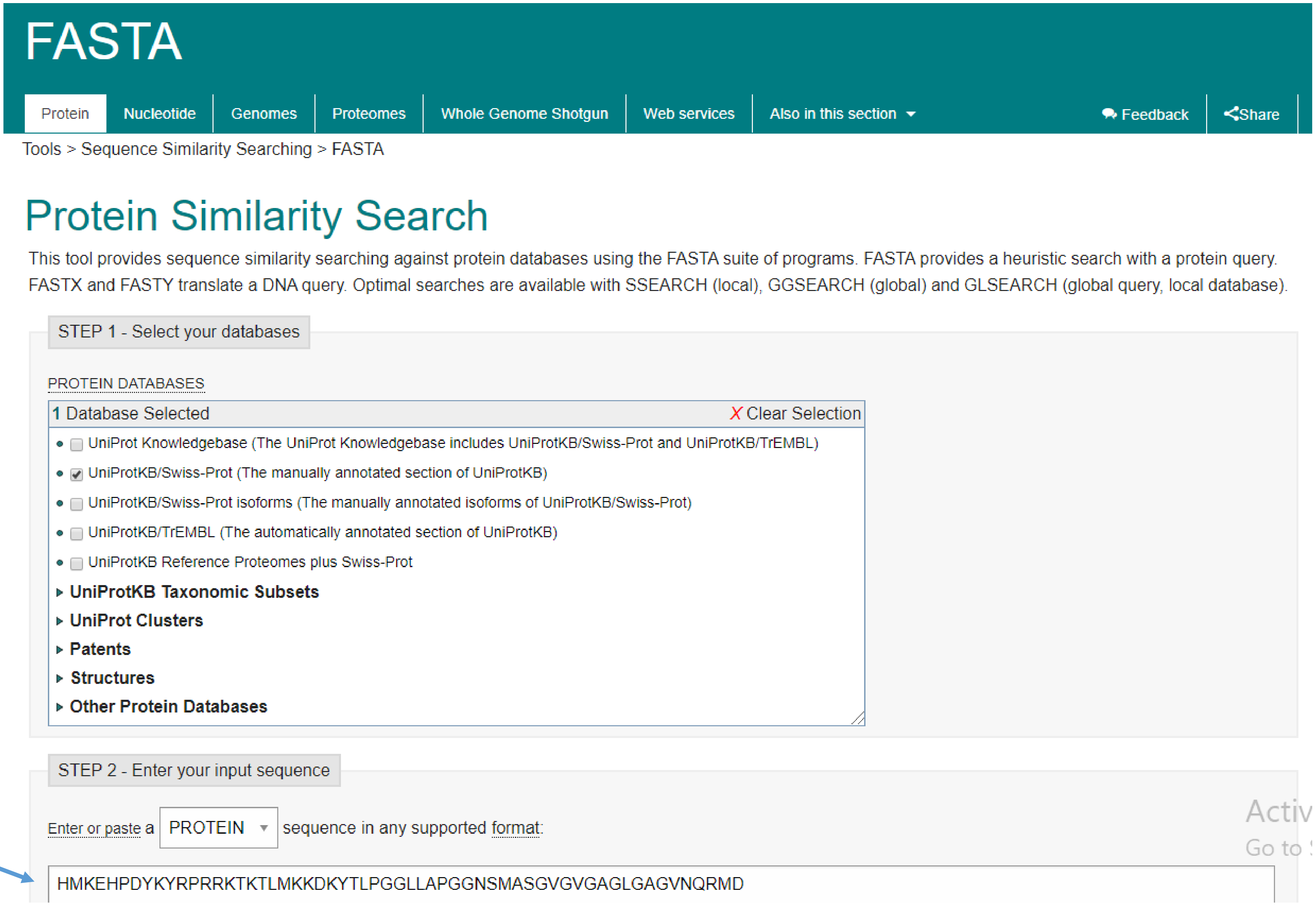
- For every sequences S in the database,
    - Compute the best alignment between S and Q by dynamic programming
- Return all alignments with the best score

- Time: $\sum_{S \in D} m|S| = O(nm)$.
- This is a brute force algorithm. So, it is the most sensitive algorithm.

What is FastA?

- Given a database D and a query Q,
    - FastA does local alignment with all database sequences and returns the database sequences with the highest local alignment score
    - Its assumption is that good local alignment should have some exact match subsequences.
- History of FastA
    - 1983: Wilbur-Lipman algorithm
    - 1985: FastP
    - 1988: FastA

Example



Tools > Sequence Similarity Searching > FASTA

Protein Similarity Search

This tool provides sequence similarity searching against protein databases using the FASTA suite of programs. FASTA provides a heuristic search with a protein query. FASTX and FASTY translate a DNA query. Optimal searches are available with SSEARCH (local), GGSEARCH (global) and GLSEARCH (global query, local database).

STEP 1 - Select your databases

PROTEIN DATABASES

1 Database Selected — X Clear Selection

- ☐ UniProt Knowledgebase (The UniProt Knowledgebase includes UniProtKB/Swiss-Prot and UniProtKB/TrEMBL)
- ☑ UniProtKB/Swiss-Prot (The manually annotated section of UniProtKB)
- ☐ UniProtKB/Swiss-Prot isoforms (The manually annotated isoforms of UniProtKB/Swiss-Prot)
- ☐ UniProtKB/TrEMBL (The automatically annotated section of UniProtKB)
- ☐ UniProtKB Reference Proteomes plus Swiss-Prot

UniProtKB Taxonomic Subsets
UniProt Clusters
Patents
Structures
Other Protein Databases

STEP 2 - Enter your input sequence

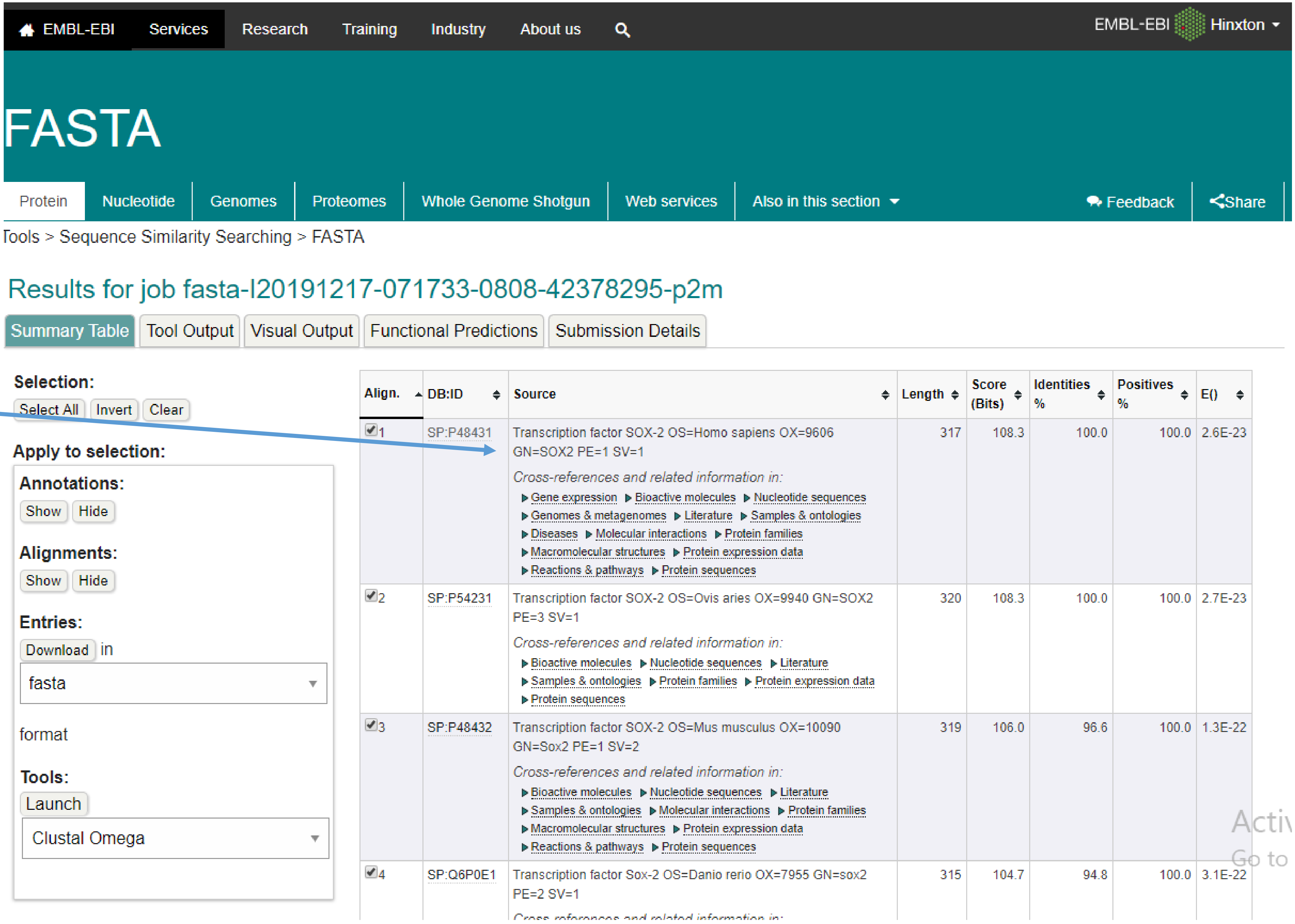
Enter or paste a PROTEIN sequence in any supported format:

```
HMKEHPDYKYRPRRKTKTLMKKDKYTLPGGLLAPGGNSMASGVGVGAGLGAGVNQRMD
```

The input is a substring of human Sox2 protein sequence

Example

It can find the correct alignment!



Tools > Sequence Similarity Searching > FASTA

Results for job fasta-I20191217-071733-0808-42378295-p2m

Align.	DB:ID	Source	Length	Score (Bits)	Identities %	Positives %	E()
✓1	SP:P48431	Transcription factor SOX-2 OS=Homo sapiens OX=9606 GN=SOX2 PE=1 SV=1 *Cross-references and related information in:* Gene expression · Bioactive molecules · Nucleotide sequences · Genomes & metagenomes · Literature · Samples & ontologies · Diseases · Molecular interactions · Protein families · Macromolecular structures · Protein expression data · Reactions & pathways · Protein sequences	317	108.3	100.0	100.0	2.6E-23
✓2	SP:P54231	Transcription factor SOX-2 OS=Ovis aries OX=9940 GN=SOX2 PE=3 SV=1 *Cross-references and related information in:* Bioactive molecules · Nucleotide sequences · Literature · Samples & ontologies · Protein families · Protein expression data · Protein sequences	320	108.3	100.0	100.0	2.7E-23
✓3	SP:P48432	Transcription factor SOX-2 OS=Mus musculus OX=10090 GN=Sox2 PE=1 SV=2 *Cross-references and related information in:* Bioactive molecules · Nucleotide sequences · Literature · Samples & ontologies · Molecular interactions · Protein families · Macromolecular structures · Protein expression data · Reactions & pathways · Protein sequences	319	106.0	96.6	100.0	1.3E-22
✓4	SP:Q6P0E1	Transcription factor Sox-2 OS=Danio rerio OX=7955 GN=sox2 PE=2 SV=1	315	104.7	94.8	100.0	3.1E-22

FastP

- Lipman and Pearson observed that the occurrences of indels are far less than that of replacements.

- Hence, FastP focused to identify high scoring alignments without indels.
- For each sequence S in the database D, the highest scoring ungapped alignment between S and Q is identified in three steps:
    1. Identification of hotspots
    2. Locating diagonal runs
    3. Rescoring the best diagonal runs

Step 1 of FastP: Identification of hotspots

- Every k-tuple (length-k substring) of a database sequence is called a hotspot if the k-tuple appears in the query sequence.
- The larger the value of k, the algorithm is faster but less sensitivity
    - Usually, k= 4-6 for DNA sequence and
    - k= 1-2 for protein sequence.

```
Query: CAACTTGCCT


Seq1: TCGGTTGCGTAGGTCCG

Seq2: TTGCGTAGGTACAACATGCCTCGT

Seq3: TCGAAGTAGCCGTCGTC

Seq4: TAGTAACTGGCCTAGTCGTAGTC
```

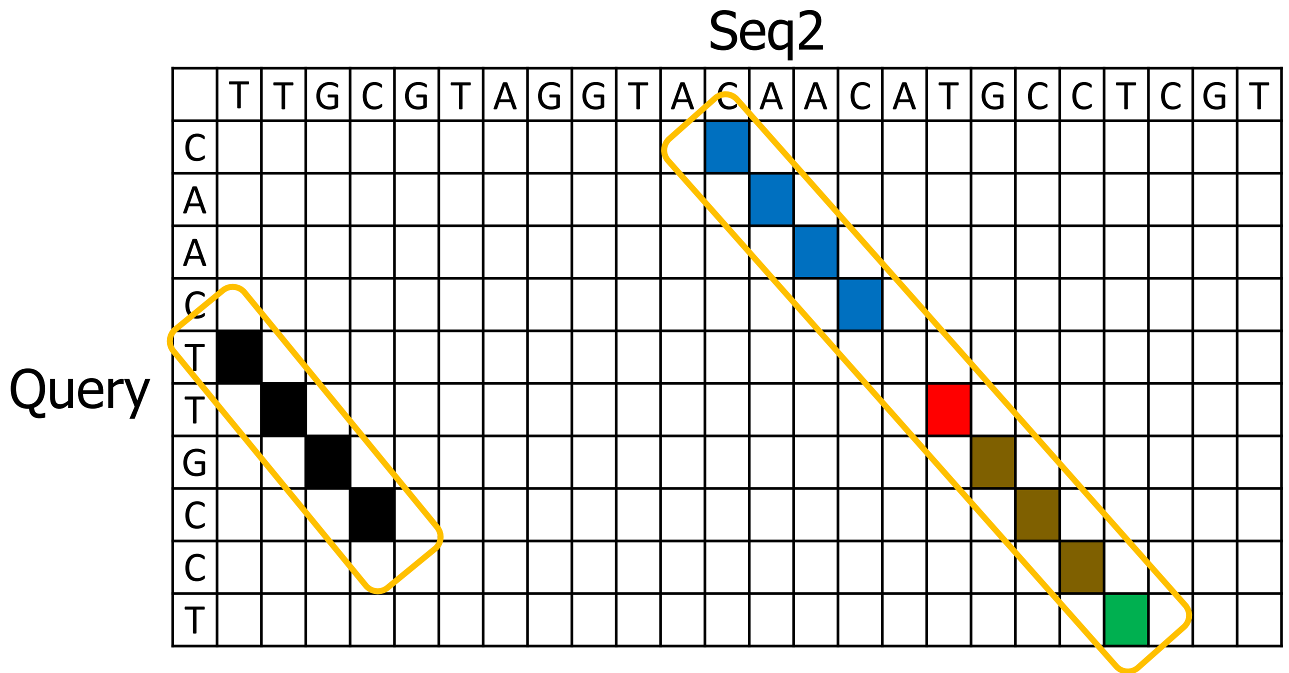
Step 1 of FastP: Identification of hotspots

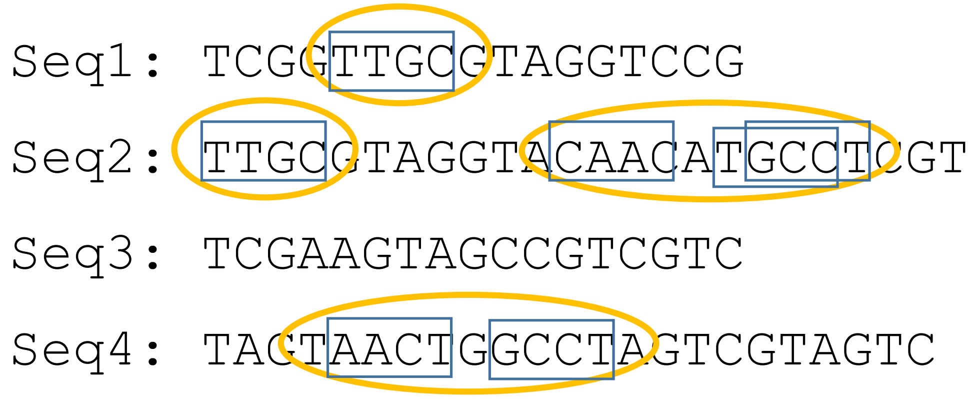
- Hotspots can be found in O(n+m) time using table lookup as follows.

1. Build a lookup table that contains all k-tuples.
2. For each k-tuple in query, flag it in the lookup table
3. For each k-tuple in database, if it is flagged in the lookup table, a hotspot is identified.

Step 2 of FastP: Find the 10 best diagonal runs

- If two nearby hotspots have the same distance in both the query sequence and the database sequence, they can be merged to form a diagonal run.
- Each hot spot is assigned a positive score. Interspot space is given a negative score that decrease with length.
- The score of a diagonal run is the sum of scores for hot spots and interspot spaces
- This steps identifies the 10 highest scoring diagonal runs for each database sequence





```
Query: CAACTTGCCT

Seq1: TCGGTTGCGTAGGTCCG
Seq2: TTGCGTAGGTACAACATGCCTCGT
Seq3: TCGAAGTAGCCGTCGTC
Seq4: TACTAACTGGCCTAGTCGTAGTC
```

Step 3 of FastP: Rescore the 10 best diagonal runs

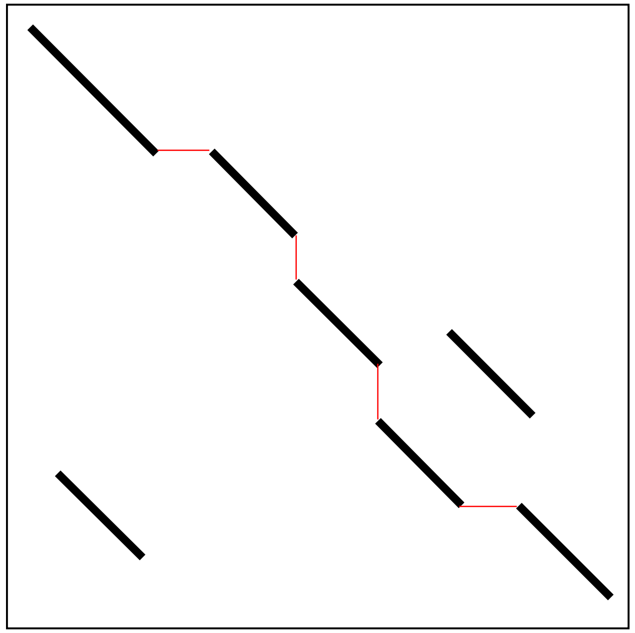
- Step 3: Rescore the 10 best diagonal runs for each database sequence
  - Using the substitution matrix for amino acid (or nucleotide), the diagonal runs are rescored.
  - Sub-region of each diagonal run whose similarity score is maximum is determined.
  - For each database sequence, the score of the best of the 10 sub-regions is called the init1 score.
- In other words, the init1 score is expected to be the best ungapped local alignment score between the database sequence and the query.
- After these 3 steps, we rank and report the database sequences according to their init1 scores.

FastA (I)

- FastA using the same first 3 steps of FastP.
- Then, it executes two additional steps, namely, Steps 4 and 5.

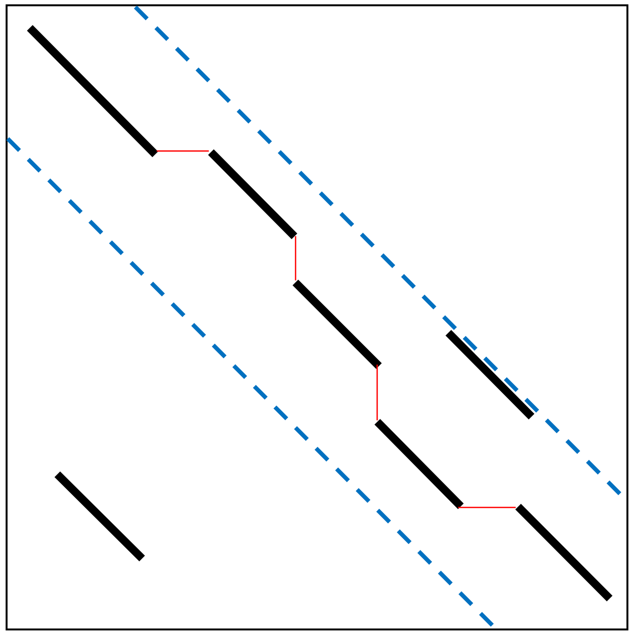
Step 4 of FastA: Attempts to join the sub-regions by allowing indels

- For the 10 sub-regions in Step 3, discard those with scores smaller than a given threshold
- For the remaining sub-regions, attempts to join them by allowing indels
- The score of the combined region is the sum of the scores of the sub-regions minus the penalty for gaps
- The best score can be computed using dynamic programming and it is called initn score.



Step 5 of FastA: Banded Smith-Waterman DP

- Sequences with initn smaller than a threshold are discarded
- For the remaining sequences, apply banded Smith-Waterman dynamic programming to complete the score opt.
- Rank the sequences based on their opt scores.

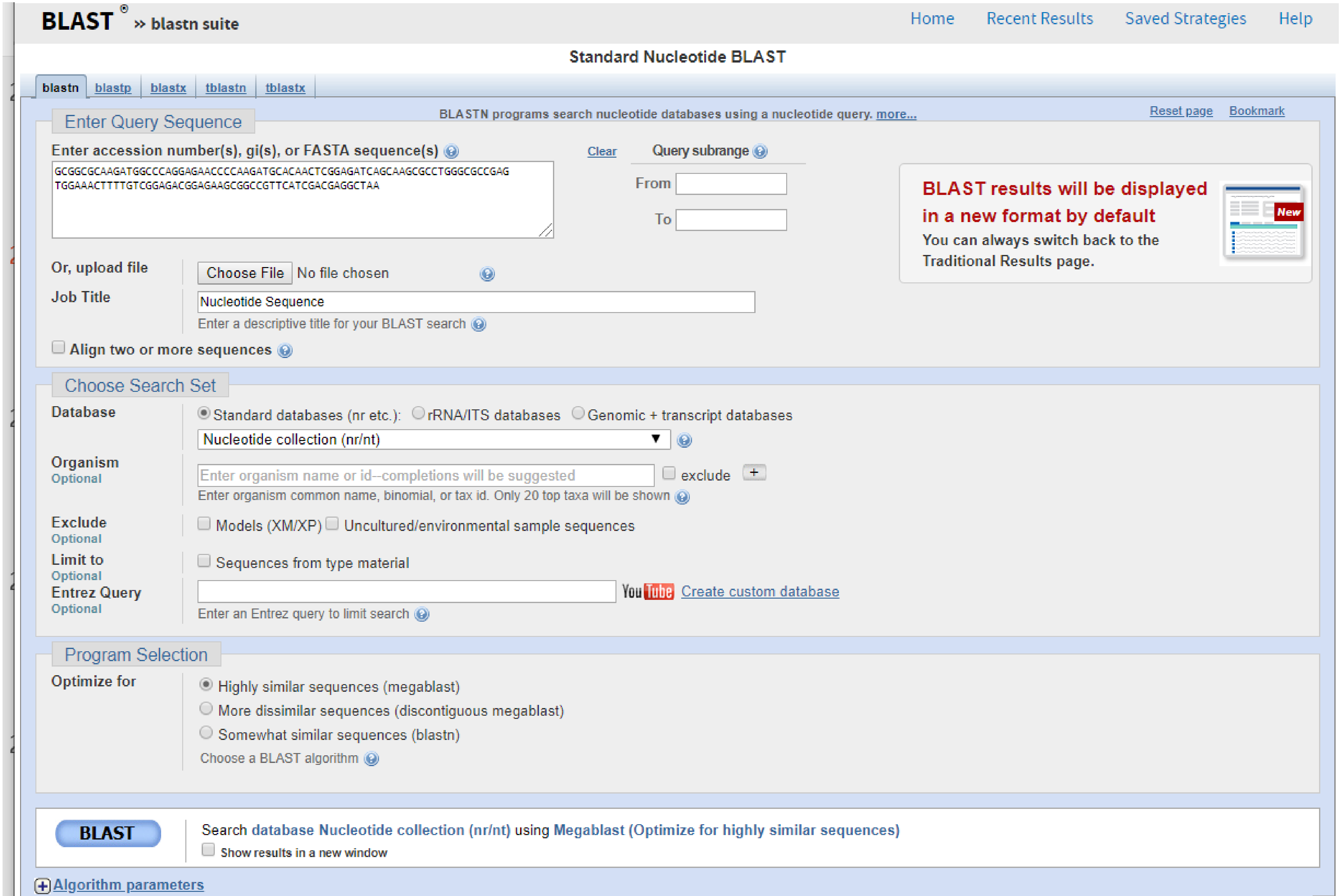


Conclusion for FlastA

- FastA is much faster than Smith-Waterman.
- It is less sensitive than
  Smith-Waterman Algorithm.

What is BLAST?

- BLAST = Basic Local Alignment Search Tool
- Input:
  - A database D of sequences
  - A sequence s
- Aim of BLAST:
  - Compare s against all sequences in D as fast as possible using heuristics.
- Disadvantage of BLAST:
  - To be fast, it scarifies the accuracy. Thus, less sensitive.



BLAST® » blastn suite

Home Recent Results Saved Strategies Help

Standard Nucleotide BLAST

blastn blastp blastx tblastn tblastx

BLASTN programs search nucleotide databases using a nucleotide query. more...

Reset page Bookmark

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s)

```
GCGGCGCAAGATGGCCCAGGAGAACCCCAAGATGCACAACTCGGAGATCAGCAAGCGCCTGGGCGCCGAG
TGGAAACTTTTGTCGGAGACGGAGAAGCGGCCGTTCATCGACGAGGCTAA
```

Clear

Query subrange

From

To

BLAST results will be displayed in a new format by default

You can always switch back to the Traditional Results page.

Or, upload file: Choose File No file chosen

Job Title: Nucleotide Sequence

Enter a descriptive title for your BLAST search

☐ Align two or more sequences

Choose Search Set

Database: [● Standard databases (nr etc.):] ○ rRNA/ITS databases ○ Genomic + transcript databases

Nucleotide collection (nr/nt)

Organism
Optional

Enter organism name or id--completions will be suggested ☐ exclude +

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown

Exclude
Optional

☐ Models (XM/XP) ☐ Uncultured/environmental sample sequences

Limit to
Optional

☐ Sequences from type material

Entrez Query
Optional

YouTube Create custom database

Enter an Entrez query to limit search

Program Selection

Optimize for

[● Highly similar sequences (megablast)]
○ More dissimilar sequences (discontiguous megablast)
○ Somewhat similar sequences (blastn)

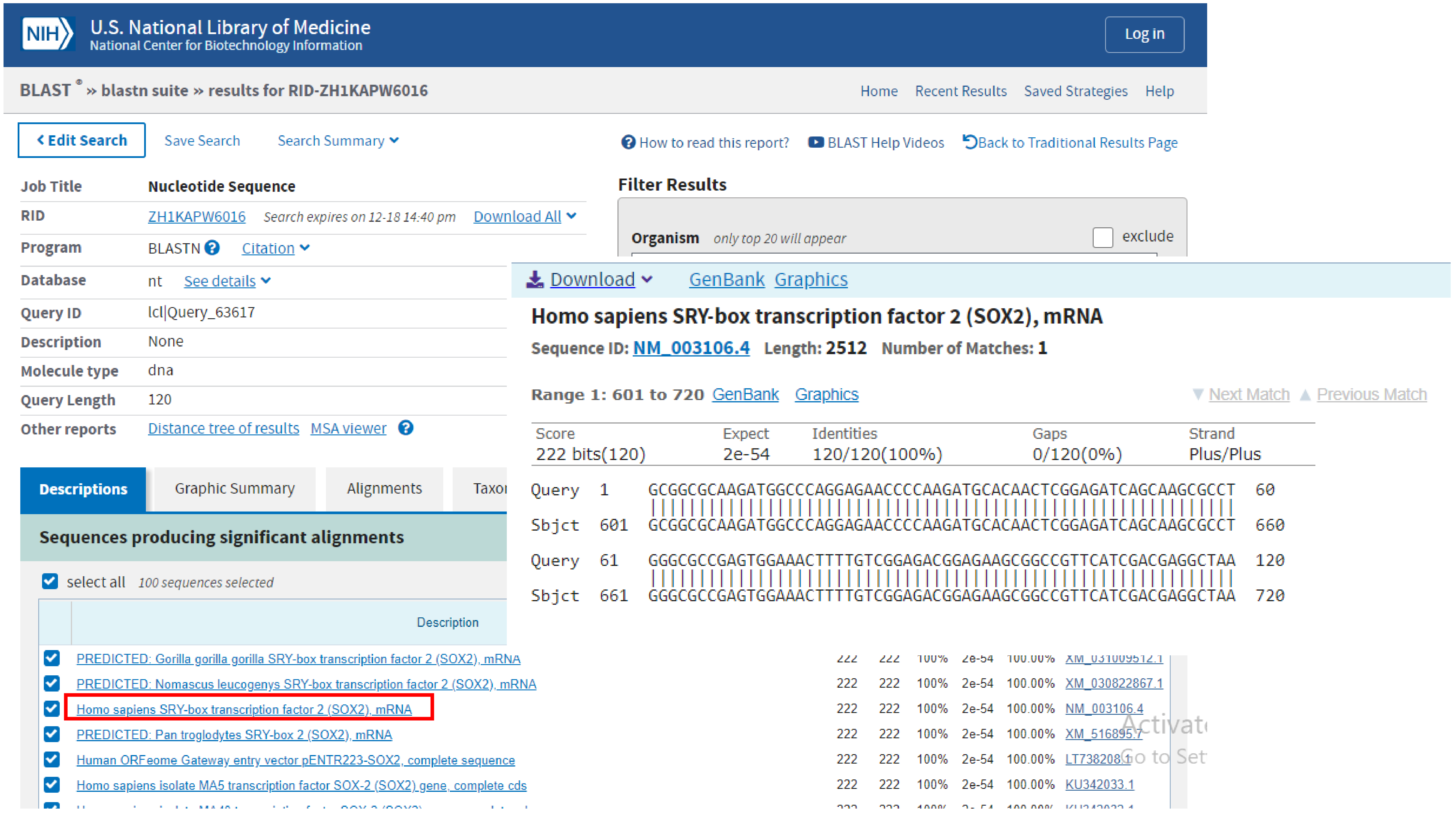
Choose a BLAST algorithm

BLAST

Search database Nucleotide collection (nr/nt) using Megablast (Optimize for highly similar sequences)

☐ Show results in a new window

Algorithm parameters


U.S. National Library of Medicine
National Center for Biotechnology Information
Log in
BLAST ® » blastn suite » results for RID-ZH1KAPW6016
Home Recent Results Saved Strategies Help
Edit Search Save Search Search Summary
How to read this report? BLAST Help Videos Back to Traditional Results Page

Job Title	Nucleotide Sequence
RID	ZH1KAPW6016 *Search expires on 12-18 14:40 pm* Download All
Program	BLASTN Citation
Database	nt See details
Query ID	lcl|Query_63617
Description	None
Molecule type	dna
Query Length	120
Other reports	Distance tree of results MSA viewer

Filter Results

Organism *only top 20 will appear* exclude

Download GenBank Graphics

Homo sapiens SRY-box transcription factor 2 (SOX2), mRNA

Sequence ID: NM_003106.4 Length: 2512 Number of Matches: 1

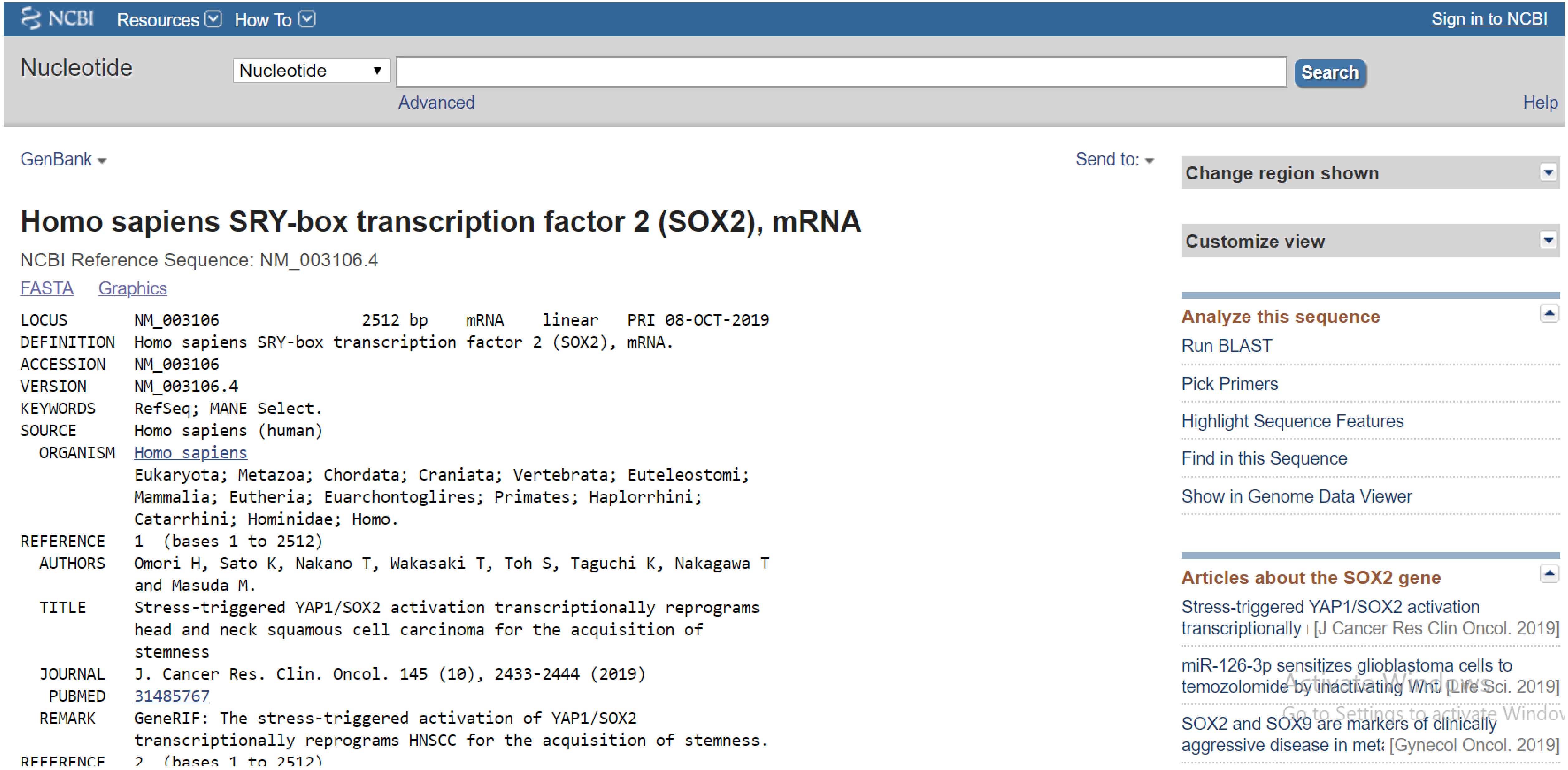
Range 1: 601 to 720 GenBank Graphics

```
Score            Expect   Identities      Gaps        Strand
222 bits(120)    2e-54    120/120(100%)   0/120(0%)   Plus/Plus

Query  1    GCGGCGCAAGATGGCCCAGGAGAACCCCAAGATGCACAACTCGGAGATCAGCAAGCGCCT  60
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  601  GCGGCGCAAGATGGCCCAGGAGAACCCCAAGATGCACAACTCGGAGATCAGCAAGCGCCT  660

Query  61   GGGCGCCGAGTGGAAACTTTTGTCGGAGACGGAGAAGCGGCCGTTCATCGACGAGGCTAA  120
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  661  GGGCGCCGAGTGGAAACTTTTGTCGGAGACGGAGAAGCGGCCGTTCATCGACGAGGCTAA  720
```

Sequences producing significant alignments

select all *100 sequences selected*

Description						
PREDICTED: Gorilla gorilla gorilla SRY-box transcription factor 2 (SOX2), mRNA	222	222	100%	2e-54	100.00%	XM_031009512.1
PREDICTED: Nomascus leucogenys SRY-box transcription factor 2 (SOX2), mRNA	222	222	100%	2e-54	100.00%	XM_030822867.1
Homo sapiens SRY-box transcription factor 2 (SOX2), mRNA	222	222	100%	2e-54	100.00%	NM_003106.4
PREDICTED: Pan troglodytes SRY-box 2 (SOX2), mRNA	222	222	100%	2e-54	100.00%	XM_516895.7
Human ORFeome Gateway entry vector pENTR223-SOX2, complete sequence	222	222	100%	2e-54	100.00%	LT738208.1
Homo sapiens isolate MA5 transcription factor SOX-2 (SOX2) gene, complete cds	222	222	100%	2e-54	100.00%	KU342033.1



Homo sapiens SRY-box transcription factor 2 (SOX2), mRNA

NCBI Reference Sequence: NM_003106.4

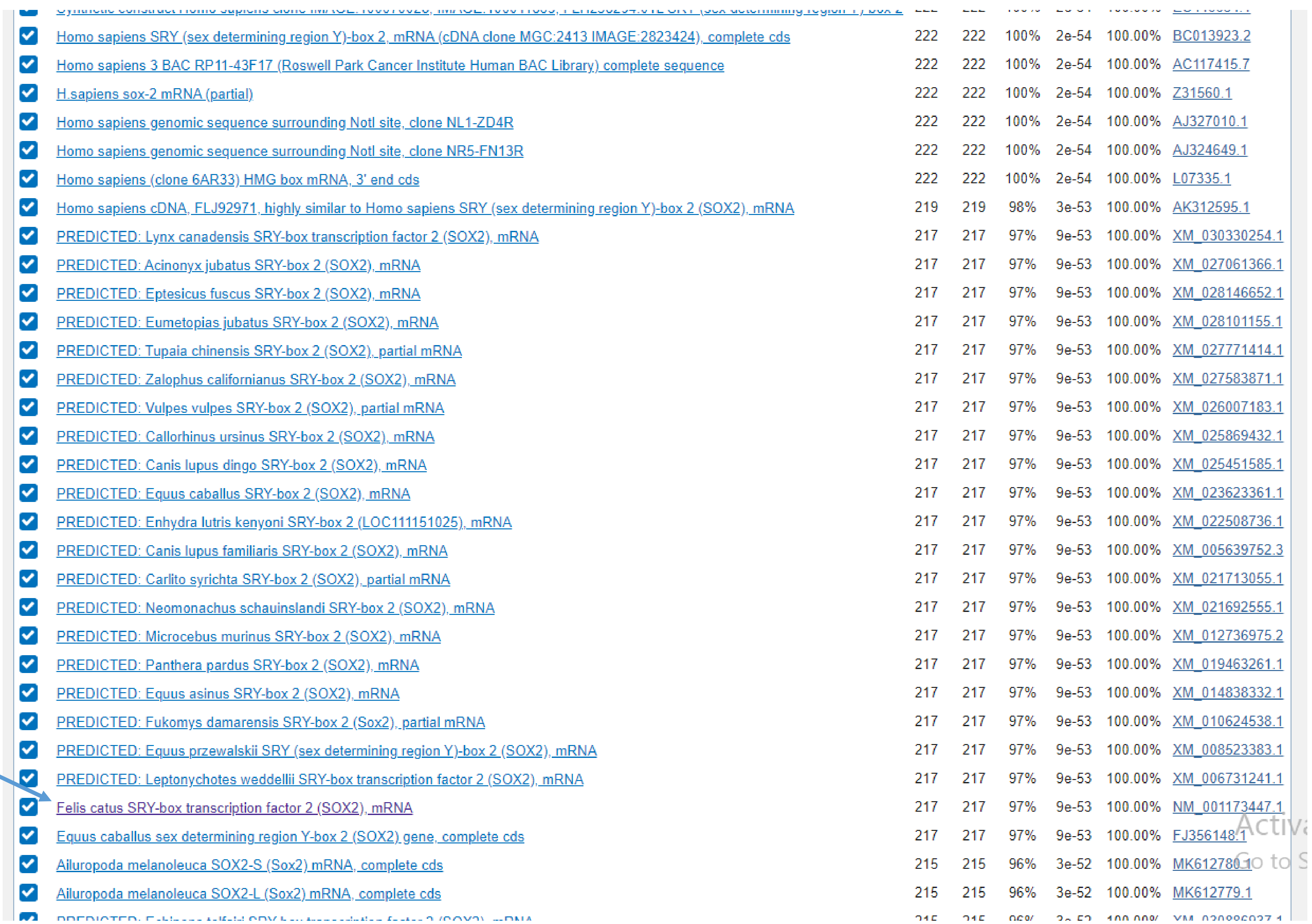
FASTA Graphics

```
LOCUS       NM_003106               2512 bp    mRNA    linear   PRI 08-OCT-2019
DEFINITION  Homo sapiens SRY-box transcription factor 2 (SOX2), mRNA.
ACCESSION   NM_003106
VERSION     NM_003106.4
KEYWORDS    RefSeq; MANE Select.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 2512)
  AUTHORS   Omori H, Sato K, Nakano T, Wakasaki T, Toh S, Taguchi K, Nakagawa T
            and Masuda M.
  TITLE     Stress-triggered YAP1/SOX2 activation transcriptionally reprograms
            head and neck squamous cell carcinoma for the acquisition of
            stemness
  JOURNAL   J. Cancer Res. Clin. Oncol. 145 (10), 2433-2444 (2019)
   PUBMED   31485767
  REMARK    GeneRIF: The stress-triggered activation of YAP1/SOX2
            transcriptionally reprograms HNSCC for the acquisition of stemness.
REFERENCE   2  (bases 1 to 2512)
```

Analyze this sequence
- Run BLAST
- Pick Primers
- Highlight Sequence Features
- Find in this Sequence
- Show in Genome Data Viewer

Articles about the SOX2 gene
- Stress-triggered YAP1/SOX2 activation transcriptionally [J Cancer Res Clin Oncol. 2019]
- miR-126-3p sensitizes glioblastoma cells to temozolomide by inactivating Wnt [Life Sci. 2019]
- SOX2 and SOX9 are markers of clinically aggressive disease in met [Gynecol Oncol. 2019]

Scroll down the screen.
You can find SOX2 from domestic cat.



✓	Synthetic construct Homo sapiens clone IMAGE:100010020, IMAGE:100011603, FLH256294.01L SRY (sex determining region Y)-box 2	222	222	100%	2e-54	100.00%	[illegible]
✓	Homo sapiens SRY (sex determining region Y)-box 2, mRNA (cDNA clone MGC:2413 IMAGE:2823424), complete cds	222	222	100%	2e-54	100.00%	BC013923.2
✓	Homo sapiens 3 BAC RP11-43F17 (Roswell Park Cancer Institute Human BAC Library) complete sequence	222	222	100%	2e-54	100.00%	AC117415.7
✓	H.sapiens sox-2 mRNA (partial)	222	222	100%	2e-54	100.00%	Z31560.1
✓	Homo sapiens genomic sequence surrounding NotI site, clone NL1-ZD4R	222	222	100%	2e-54	100.00%	AJ327010.1
✓	Homo sapiens genomic sequence surrounding NotI site, clone NR5-FN13R	222	222	100%	2e-54	100.00%	AJ324649.1
✓	Homo sapiens (clone 6AR33) HMG box mRNA, 3' end cds	222	222	100%	2e-54	100.00%	L07335.1
✓	Homo sapiens cDNA, FLJ92971, highly similar to Homo sapiens SRY (sex determining region Y)-box 2 (SOX2), mRNA	219	219	98%	3e-53	100.00%	AK312595.1
✓	PREDICTED: Lynx canadensis SRY-box transcription factor 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_030330254.1
✓	PREDICTED: Acinonyx jubatus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_027061366.1
✓	PREDICTED: Eptesicus fuscus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_028146652.1
✓	PREDICTED: Eumetopias jubatus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_028101155.1
✓	PREDICTED: Tupaia chinensis SRY-box 2 (SOX2), partial mRNA	217	217	97%	9e-53	100.00%	XM_027771414.1
✓	PREDICTED: Zalophus californianus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_027583871.1
✓	PREDICTED: Vulpes vulpes SRY-box 2 (SOX2), partial mRNA	217	217	97%	9e-53	100.00%	XM_026007183.1
✓	PREDICTED: Callorhinus ursinus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_025869432.1
✓	PREDICTED: Canis lupus dingo SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_025451585.1
✓	PREDICTED: Equus caballus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_023623361.1
✓	PREDICTED: Enhydra lutris kenyoni SRY-box 2 (LOC111151025), mRNA	217	217	97%	9e-53	100.00%	XM_022508736.1
✓	PREDICTED: Canis lupus familiaris SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_005639752.3
✓	PREDICTED: Carlito syrichta SRY-box 2 (SOX2), partial mRNA	217	217	97%	9e-53	100.00%	XM_021713055.1
✓	PREDICTED: Neomonachus schauinslandi SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_021692555.1
✓	PREDICTED: Microcebus murinus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_012736975.2
✓	PREDICTED: Panthera pardus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_019463261.1
✓	PREDICTED: Equus asinus SRY-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_014838332.1
✓	PREDICTED: Fukomys damarensis SRY-box 2 (Sox2), partial mRNA	217	217	97%	9e-53	100.00%	XM_010624538.1
✓	PREDICTED: Equus przewalskii SRY (sex determining region Y)-box 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_008523383.1
✓	PREDICTED: Leptonychotes weddellii SRY-box transcription factor 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	XM_006731241.1
✓	Felis catus SRY-box transcription factor 2 (SOX2), mRNA	217	217	97%	9e-53	100.00%	NM_001173447.1
✓	Equus caballus sex determining region Y-box 2 (SOX2) gene, complete cds	217	217	97%	9e-53	100.00%	FJ356148.1
✓	Ailuropoda melanoleuca SOX2-S (Sox2) mRNA, complete cds	215	215	96%	3e-52	100.00%	MK612780.1
✓	Ailuropoda melanoleuca SOX2-L (Sox2) mRNA, complete cds	215	215	96%	3e-52	100.00%	MK612779.1
✓	PREDICTED: Echinops telfairi SRY-box transcription factor 2 (SOX2), mRNA	215	215	96%	3e-52	100.00%	XM_030886937.1

History of BLAST

- 1990: Birth of BLAST1
  - It is very fast and dedicate to the search of local similarities **without** gaps
  - Altschul et al, Basic local alignment search tool. J. Mol. Biol., 215(3):403-410, 1990.
  - The most highly cited paper in 1990 and the third most highly cited paper in 1983-2002.
- 1996-1997: Birth of BLAST2
  - BLAST2 allows insertion of gaps
  - BLAST2 have two versions. Developed by two groups of authors independently
    - 1997: NCBI-BLAST2 (National Center for Biotechnology Information)
    - 1996: WU-BLAST2 (Washington University)

BLAST1

- A heuristic method which searches for local similarity without gap
- Basic idea:
  - Find hits between query and database sequences
  - Extend the hits to get the ungapped local alignments
- BLAST1 can be divided into three steps:
  - Step 1: Query preprocessing
  - Step 2: Generation of hits by database scanning
  - Step 3: Extension of hits

Step 1: Query preprocessing (DNA)

- Step 1 finds w-tuples that are similar to some w-tuples in the query Q.
- These w-tuples are called neighbor
- For DNA, w=11 by default. A w-tuple X is a neighbor of a w-tuple Y if X=Y.
- Query preprocessing builds a table of neighbors of w-tuples in the query sequence.

Q=GTCATCATG

w-tuple	positions
ATCA	4
CATC	3
CATG	6
GTCA	1
TCAT	2, 5

Step 1: Query preprocessing (protein)

- For protein, w-tuple X is a neightbor of another w-tuple Y if $\sum_{i=1}^{w} \delta(X[i], Y[i]) \geq T$. By default, w=3 and T=13.
- For example, the (3,13)-neighbor of NII is { NII, NIV, NVI } while the (3,13)-neighbor of IIN is { IIN, IVN, VIN }.

Q=`NIIN`

w-tuple	positions
IIN	2
IVN	2
NII	1
NIV	1
NVI	1
VIN	2

Step 2: Generation of hits by database scanning

- For every database sequence S in the database D,
  - Scan every position q in the sequence S, if there is an exact match between the w-tuple at position q in S and a w-tuple in the query, a hit is found.
- A hit is characterized by the positions in both query and DB sequences.

Q=GTCATCATG

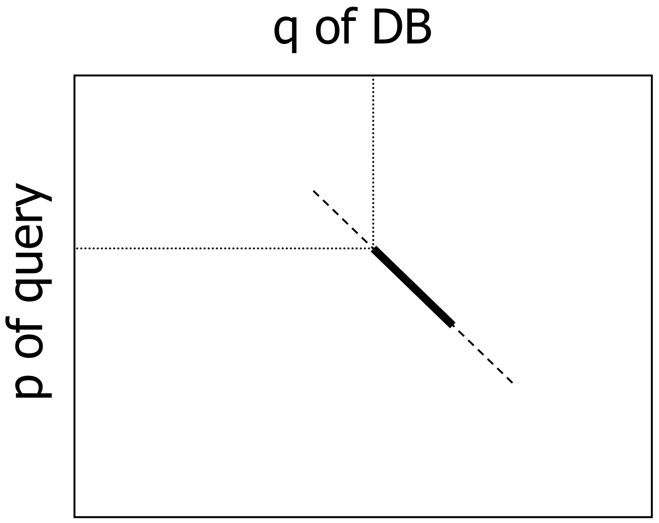
>seq1
CCGCTCATGATGATCA

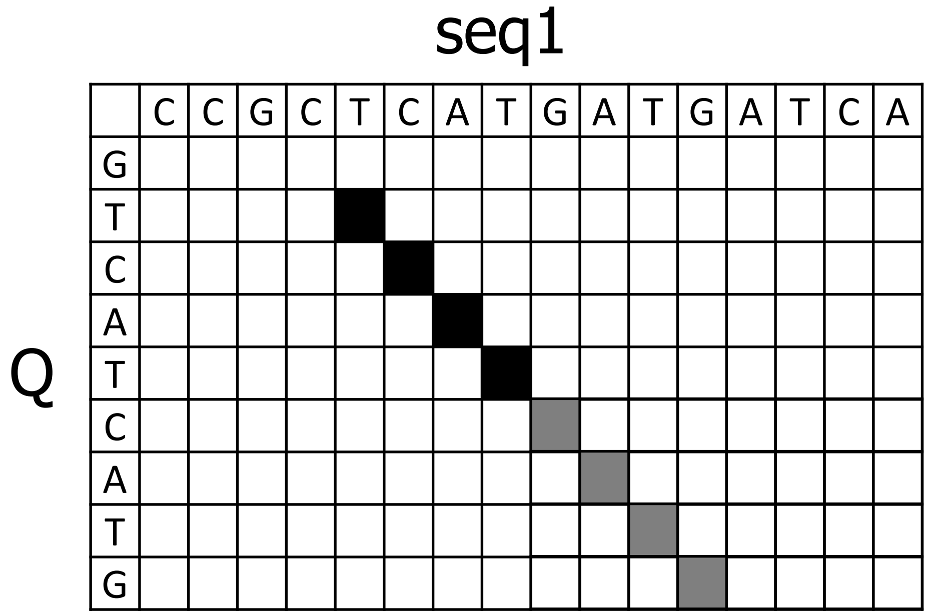
w-tuple	positions
ATCA	4
CATC	3
CATG	6
GTCA	1
TCAT	2, 5

The list of hits:
- (5 of seq1, 2 of Q)
- (5 of seq1, 5 of Q)
- (13 of seq1, 4 of Q)

Step 3: Extension of hits (I)

- For every hit, extend it in both directions, without gap.
- Equivalently, in the dotplot, the hit is extended along the diagonal.





Q= G**TCATCATG**
Seq1=CCGC**TCATGATG**ATCA

Step 3: Extension of hits (II)

- The extension is extended greedy. The extension is stopped as soon as the score decreases by more than X(parameter of the program) from the highest value reached so far.
- Example: Assume the match score is +5, the mismatch score is -4, and the extension truncation threshold is X=13. Suppose the hit between S and T is GTCG.

```
S = GTCTCACCGTCGCACGACATCTC
T = GCGGAACTGTCGAACAATCCTCT
```

```
      -10 -6  -2 +2+6 +1 -4              -4  +1 +6+2+7 +3 -1 -5  -9
       -4  -4  -4 -4 +5 +5 -4             -4  +5 +5-4 +5 -4  -4 -4  -4
```

From 6 to -10, drop more than X. Truncate.

From 7 to -9, drop more than X. Truncate.

Step 3: Extension of hits (III)

From 6 to -10, drop more than X. Truncate.

From 7 to -9, drop more than X. Truncate.

```
       -10 -6  -2 +2 +6 +1 -4               -4  +1 +6 +2 +7 +3 -1 -5  -9
        -4 -4  -4 -4 +5 +5 -4               -4  +5 +5 -4 +5 -4 -4 -4  -4
S  =  GTCTCACCGTCGCACGACATCTC
T  =  GCGGAACTGTCGAACAATCCTCT
```

- To obtain the highest ungapped alignment score , we trim the first 4 aligned positions and the last 4 aligned positions. Then, the resultant ungapped alignment is

```
ACCGTCGCACGA
ACTGTCGAACAA
+5 +5 -4 +5 +5 +5 +5 -4 +5 +5 -4 +5
```

- This is known as the high scoring segment pair (HSP), its score is 28.

Step 3: Extension of hits (IV)

- If the HSP score is better than or equal to α (parameter of the program), it will be reported.

- For every database sequence, it may have multiple HSPs. The best scoring HSP is called the maximal segment pair (MSP).

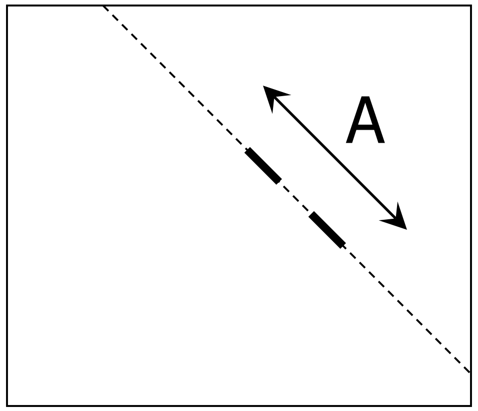
NCBI-Blast2

- Report gapped local alignments.

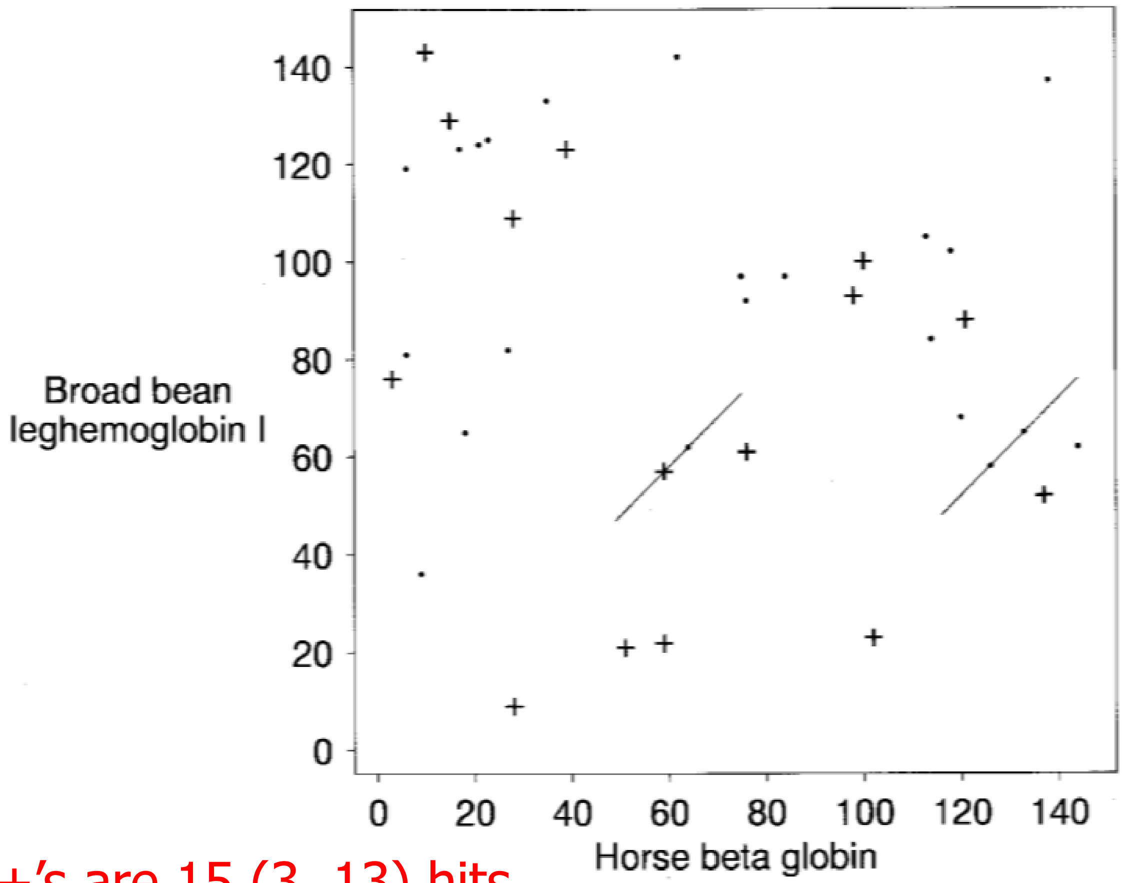
- The first 2 steps are the same as BLAST1.
- Two major differences:
  - Two-hits requirement (implemented for protein)
  - Gapped extension

Step 3: Two-hits requirement

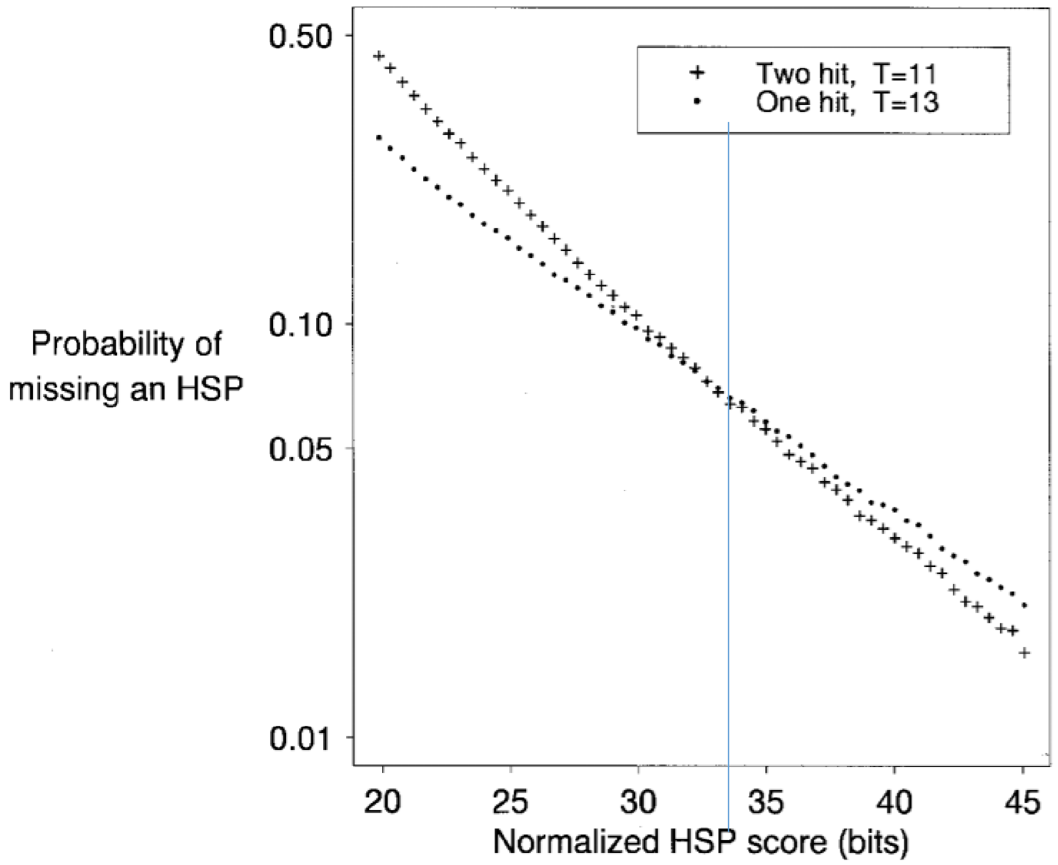
- To extend a hit, we require that there is another hit on the same diagonal within a distance smaller than A
- By default, A=40
- Note: Two-hits requirement is implemented for protein sequences (not DNA).



Comparing One hit (3,13)-neighbor and Two hit (3, 11)-neighbor



+'s are 15 (3, 13) hits.
.'s are 22 (3, 11) hits.
The two lines are pairs of (3,11) hits.



The two-hit method (A=40) is more sensitive for HSPs with score > 33 bits.

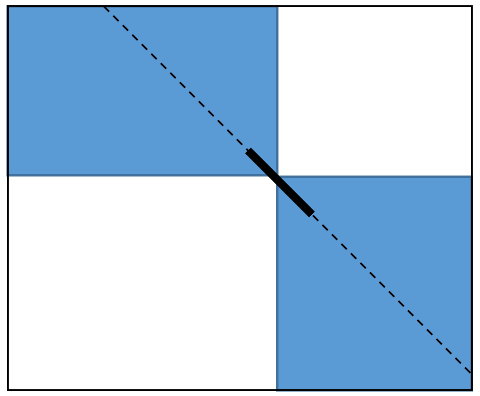
Step 4: Gapped extension (I)

- Obtain a set of HSPs by performing ungapped extension similar to Step 3 of BLAST1
- Among the generated HSP, we perform gapped extension for those with score > some threshold

- Note: The ungapped extension is a filter. It reduces the number of gapped extensions.

Step 4: Gapped extension (II)

- For each HSP, the length-11 segment with the highest alignment score is identified. (If the HSP is shorter than 11, the whole segment is taken.)
- Gapped extension is started from the middle of the segment.



Illustration

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

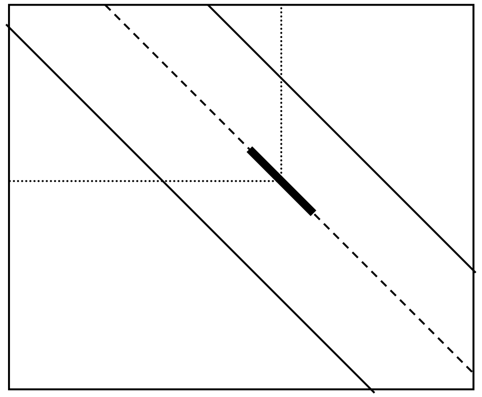
```
S = TAAC-AGTTGA-CAAT
T = T-ACGAGTTGAGCA-T
```

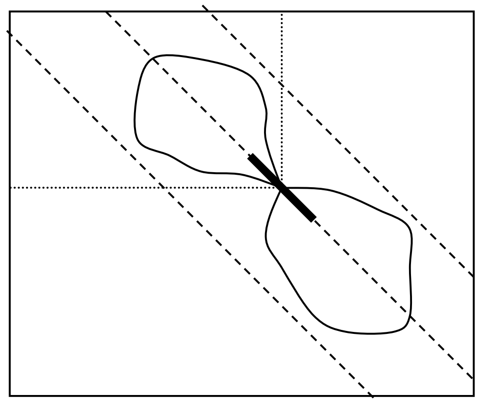
		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
G	1	-1	↖ 2	1	0	-1	-2	-3	-4	-5	-6	-7
T	2	-2	1	↖ 4	3	2	1	0	-1	-2	-3	-4
A	3	-3	0	3	↖ 6	← 5	4	3	2	1	0	-1
C	4	-4	-1	2	5	5	↖ 7	6	5	4	3	-1
A	5	-5	-2	1	4	4	6	↖ 9	8	7	6	5
A	6	-6	-3	0	3	3	6	↑ 8	7	6	5	5
T	7	-7	-4	-1	2	2	4	7	↖ 10	9	8	7
C	8	-8	-5	-2	1	1	4	6	9	9	8	7
C	9	-9	-6	-3	0	0	3	5	8	8	8	7
C	10	-10	-7	-4	-1	-1	2	4	7	7	7	7

Assume match scores +2
Mismatch/indel score -1.

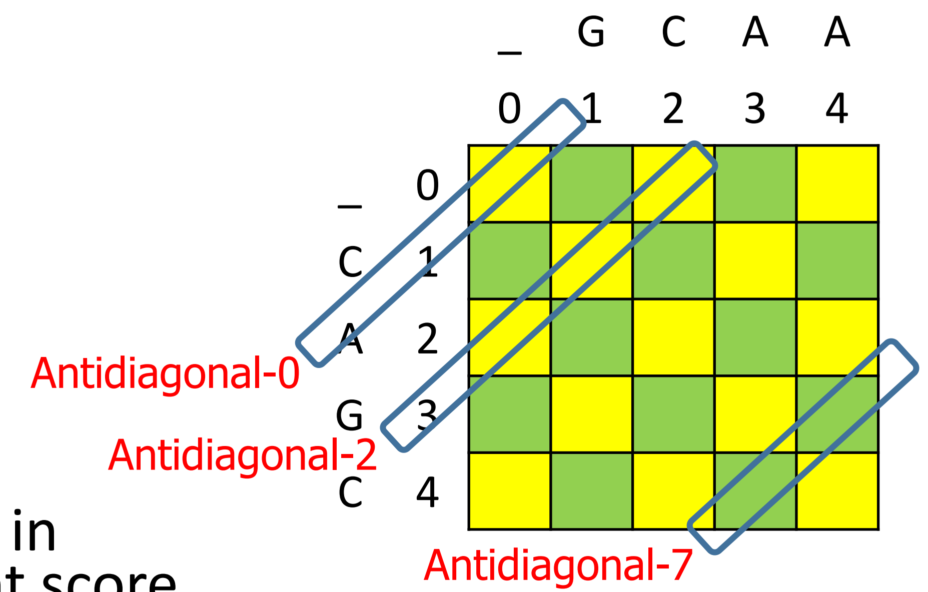
Step 4: Gapped extension (II)

- Running DP starting from the middle of the segment is slow.
- To reduce time, banded dynamic programming can be used.
- To further speedup, BLAST proposed to use X-drop algorithm for gapped extension
- X-drop is a modified semi-global alignment algorithm
  - It fill in only entries that are no more than X below the best alignment score yet found.


Banded DP


X-drop algorithm

X-drop algorithm



- Let antidiagonal-d be the entries V[i,j] in the table where i+j=d.
- The X-drop algorithm fills in entries in antidiagonal-d in increasing order of d and maintain the best alignment score seen so far in the variable τ.

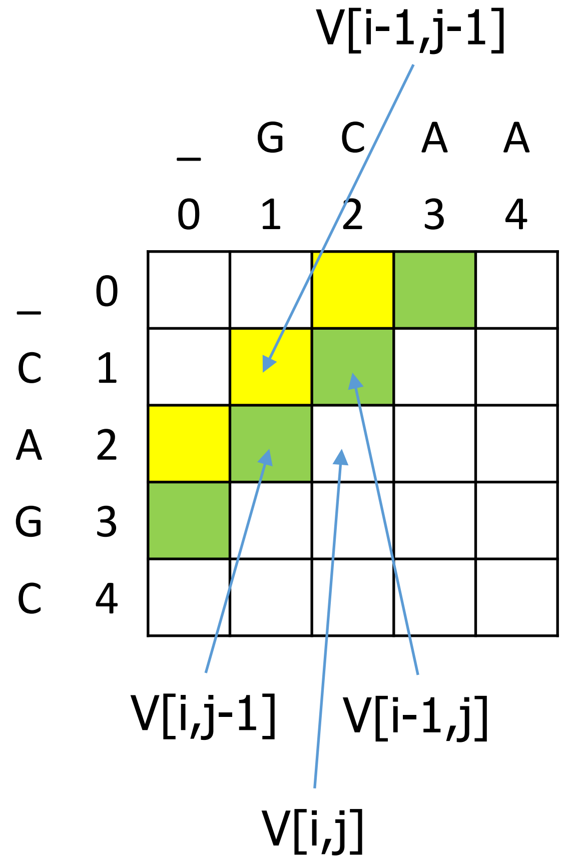
- For each antidiagonal, X-drop algorithm keeps entries that are consistent.
- An entry V[i,j] is consistent if
  - Either V[i-1,j-1], V[i-1,j] or V[i,j-1] is consistent; and
  - V[i,j] is no more than X below the best alignment score seen so far (i.e. $V[i,d-i] \geq \tau - X$).
- The X-drop algorithm will stop when the algorithm cannot generate consistent entries.

Consistent entries

- X-drop algorithm computes V[i,j] on antidiagonal-d if:
  - V[i-1,j-1] on antidiagonal-(d-2) is consistent; or
  - V[i, j-1] or V[i-1,j] on antidiagonal-(d-1) is consistent.

- $V[i,j] = \max\begin{cases} V[i-1,j-1] + \delta(S[i],T[d-i]) \\ V[i-1,j] + \delta(S[i],\_) \\ V[i,j-1] + \delta(\_,T[d-i]) \end{cases}$

- After V[i, j] is computed, V[i,j] is consistent if V[i,j] $\geq \tau$-X.



X-drop algorithm

- 1. Set V[0,0]=0 as a consistent entry, τ=0.
- 2. For d = 1, 2, …
    - 2.1. If both antidiagonal-(d-1) and antidiagonal-(d-2) do not have consistent entries, then stop;
    - 2.2. For every V[i,j] on antidiagonal-d such that V[i-1,j-1], V[i-1,j] or V[i,j-1] is a consistent entry,
        - Compute V[i,j]
        - If V[i,j] $\geq \tau$-X, set V[i,j] as a consistent entry
    - 2.3 Set τ = max {τ, max{V[i,j] | V[i,j] is a consistent entry on antidiagonal-d } }

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTTGACAATCCC
T = TGGTACGAGTTTGAGCATGGT
```

τ=0
d=0
Black: Consistent
Green: Not consistent

		_ 0	G 1	T 2	A 3	G 4	C 5	A 6	T 7	G 8	G 9	T 10
_	0	0										
G	1											
T	2											
A	3											
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

$\tau=0$
$d=1$
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1									
G	1	-1										
T	2											
A	3											
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTTGACAATCCC
T = TGGTACGAGTTTGAGCATGGT
```

τ=2
d=2
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2									
T	2	-2										
A	3											
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=2
d=3
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1								
T	2	-2	1									
A	3											
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=4
d=4
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4								
A	3		0									
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT

τ=9
d=12
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8			
A	6						5	8				
T	7											
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=9
d=13
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7			
T	7							7				
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=10
d=14
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10			
C	8											
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=10
d=15
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10	9		
C	8								9			
C	9											
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=10
d=16
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10	9	8	
C	8								9	9		
C	9								8			
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTTGACAATCCC
T = TGGTACGAGTTTGAGCATGGT
```

τ=10
d=17
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10	9	8	
C	8								9	9	8	
C	9								8	8		
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

τ=10
d=18
Black: Consistent
Green: Not consistent

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10	9	8	
C	8								9	9	8	
C	9								8	8	8	
C	10											

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Illustration (X-drop approach)

```
S = CCCTAACAGTTGACAATCCC
T = TGGTACGAGTTGAGCATGGT
```

```
S = TAAC-AGTTGA-CAAT
T = T-ACGAGTTGAGCA-T
```

Assume match scores +2
Mismatch/indel score -1.
Assume X=1

Perform semi-global alignment starting from the middle of the hit.
Since this example is symmetric, we only show the DP on the right segment.

		_	G	T	A	G	C	A	T	G	G	T
		0	1	2	3	4	5	6	7	8	9	10
_	0	0	-1	-2								
G	1	-1	2	1	0							
T	2	-2	1	4	3	2						
A	3		0	3	6	5	4					
C	4			2	5	5	7	6	5			
A	5				4	4	6	9	8	7		
A	6						5	8	7	7		
T	7							7	10	9	8	
C	8								9	9	8	
C	9								8	8	8	
C	10											

X-drop algorithm allowing affine gap penalty

- The previous algorithm is for linear gap.
- We can generalize the X-drop algorithm to handle affine gap penalty.
- Exercise!

BLAST2 algorithm

1. Query preprocessing
2. Generation of hits by database scanning
3. Filter hits by 2-hit requirement (for protein only)
4. Generate HSP by ungapped extension. For each HSP with score > some threshold, perform gapped extension using X-drop algorithm.

BLAST1 vs. NCBI-BLAST2

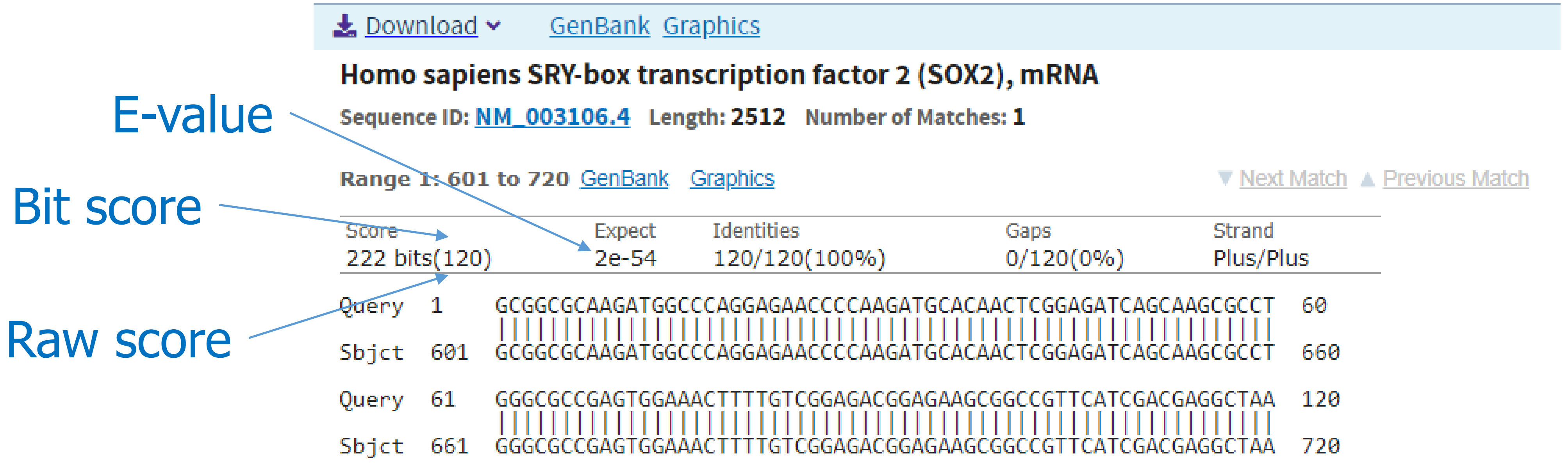
- BLAST1 spends 90% of its time on extension
- For NCBI-BLAST2, due to the two-hits requirement, the number of extensions is reduced.
  - NCBI-BLAST2 is about 3 times faster than BLAST1.

BLAST program options

Program	Query	Database	Alignment type
blastn	DNA	DNA	Search DNA query sequence in DNA database
blastp	Protein	Protein	Search protein query sequence in protein database
blastx	DNA	Protein	Convert DNA query sequence into protein sequences in all 6 reading frames. Search these translated proteins in protein database
tblastn	Protein	DNA	Search protein query sequence again protein sequences generated from the 6 reading frames of the DNA sequences in the DNA database
tblastx	DNA	DNA	Convert DNA query sequence into protein sequences in all 6 reading frames. Search these translated protein query sequence again protein sequences generated from the 6 reading frames of the DNA sequences in the DNA database

Statistics for local alignment

- An ungapped local alignment consists of a pair of equal length segments.
- BLAST finds the local alignments whose score cannot be improved by extension. Such local alignments are called high-scoring segment pairs or HSPs.
- To determine the significant of each HSP, BLAST gives E-value and bit score. Below, we give a brief discussion on them.



Homo sapiens SRY-box transcription factor 2 (SOX2), mRNA

Sequence ID: NM_003106.4 Length: 2512 Number of Matches: 1

Range 1: 601 to 720 GenBank Graphics

Score	Expect	Identities	Gaps	Strand
222 bits(120)	2e-54	120/120(100%)	0/120(0%)	Plus/Plus

```
Query  1    GCGGCGCAAGATGGCCCAGGAGAACCCCAAGATGCACAACTCGGAGATCAGCAAGCGCCT  60
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  601  GCGGCGCAAGATGGCCCAGGAGAACCCCAAGATGCACAACTCGGAGATCAGCAAGCGCCT  660

Query  61   GGGCGCCGAGTGGAAACTTTTGTCGGAGACGGAGAAGCGGCCGTTCATCGACGAGGCTAA  120
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  661  GGGCGCCGAGTGGAAACTTTTGTCGGAGACGGAGAAGCGGCCGTTCATCGACGAGGCTAA  720
```

Assumption

- We required the expected score for aligning a random pair of residues/bases to be negative.
- Otherwise, the longer the alignment, the higher is the score independent of whether the segments aligned are related or not.

	A	C	G	T
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
T	-1	-1	-1	2

Good!

Expected score =
$$\frac{1}{16}\big(4*2+12*(-1)\big)=-\frac{1}{4}$$

	A	C	G	T
A	5	-1	-1	-1
C	-1	5	-1	-1
G	-1	-1	5	-1
T	-1	-1	-1	5

Not Good!

Expected score =
$$\frac{1}{16}\big(4*5+12*(-1)\big)=\frac{1}{2}$$

Raw Score for an ungapped alignment in BLAST

- There are 12 matches and 3 mismatches
- Raw score = 12*2 – 3*3 = 15.

```
GAACGTACTCCTACG
GCACTTACTGCTACG
```

	A	C	G	T
A	2	-3	-3	-3
C	-3	2	-3	-3
G	-3	-3	2	-3
T	-3	-3	-3	2

BLAST score matrix

Expected score = $\frac{1}{16}\left(4 * 2 + 12 * (-3)\right) = -1.75$

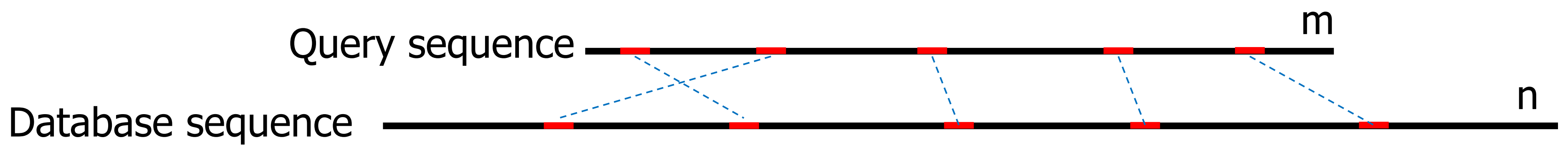
E-value

- Let S be the raw score of a HSP between a length-m query sequence and a length-n database sequence.
- E-value of the HSP is defined as the expected number of random alignments whose score is bigger than the raw score S.
- Denote E be such E-value.
- When E is small, the HSP is unlikely to be a random alignment.
- By default, BLAST reports the HSP if $E \leq 10$.

E-value (II)

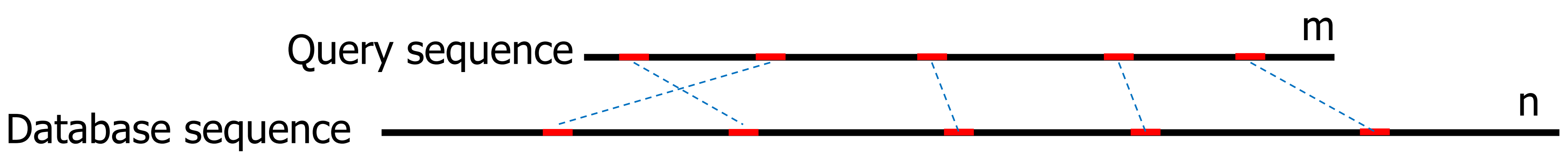
- When both m (length of query) and n (length of database sequence) are sufficiently long,
  - E follows the extreme distribution (Gumbel distribution). We have:
    - $E = Kmne^{-\lambda S}$

      for some parameters K and λ which depends on the similarity matrix δ and the expected frequencies of the residues/bases.
    - For more information on estimating K and λ, please read
      - http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-3.html
      - http://oreilly.com/catalog/blast/chapter/ch04.pdf



E-value (III)

- We can justify the equation $E = Kmne^{-\lambda S}$ intuitively:
  - Double the length of either sequence will double the expected number of HSPs. (i.e. $E \propto nm$)
  - Double the score S will exponentially reduce the expected number of HSPs. (i.e. $\ln E \propto S$ or $E \propto e^{-\lambda S}$)
- Hence, $E \propto mne^{-\lambda S}$.



Example

	A	C	G	T
A	2	-3	-3	-3
C	-3	2	-3	-3
G	-3	-3	2	-3
T	-3	-3	-2	2

BLAST matrix δ

- The raw score for the following HSP is S=15.

```
GAACGTACTCCTACG
GCACTTACTGCTACG
```

- For matrix δ, ln K ≈ 0.42 and λ ≈ 0.626
- Suppose m (query length) = 20, n (database sequence length) = 70.

- The E-value is $E = Kmne^{-\lambda S} = 0.178035469$

Bit score

- The raw score S of an alignment depends on the scoring system.
- Without knowing the scoring system, the raw score is meaningless.

- The bit score is a normalized raw score, which is independent of the scoring system. It is defined as:

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

- By definition, $2^{-S'} = Ke^{-\lambda S}$. Hence, $E = Kmne^{-\lambda S} = mn2^{-S'}$.
- E-value can be computed without parameters depending on the scoring system.

Example

	A	C	G	T
A	2	-3	-3	-3
C	-3	2	-3	-3
G	-3	-3	2	-3
T	-3	-3	-2	2

Matrix δ

- The raw score for the following HSP is S=15.

```
GAACGTACTCCTACG
GCACTTACTGCTACG
```

- For matrix δ, ln K ≈ 0.42 and λ ≈ 0.626
- Suppose m (query length) = 20, n (database sequence length) = 70.

- The bit score is $S' = \frac{\lambda S - \ln K}{\ln 2} = 12.94097452$
- Note that $E = Kmne^{-\lambda S} = 0.178035469$ and $mn2^{-S'} = 0.178035469$

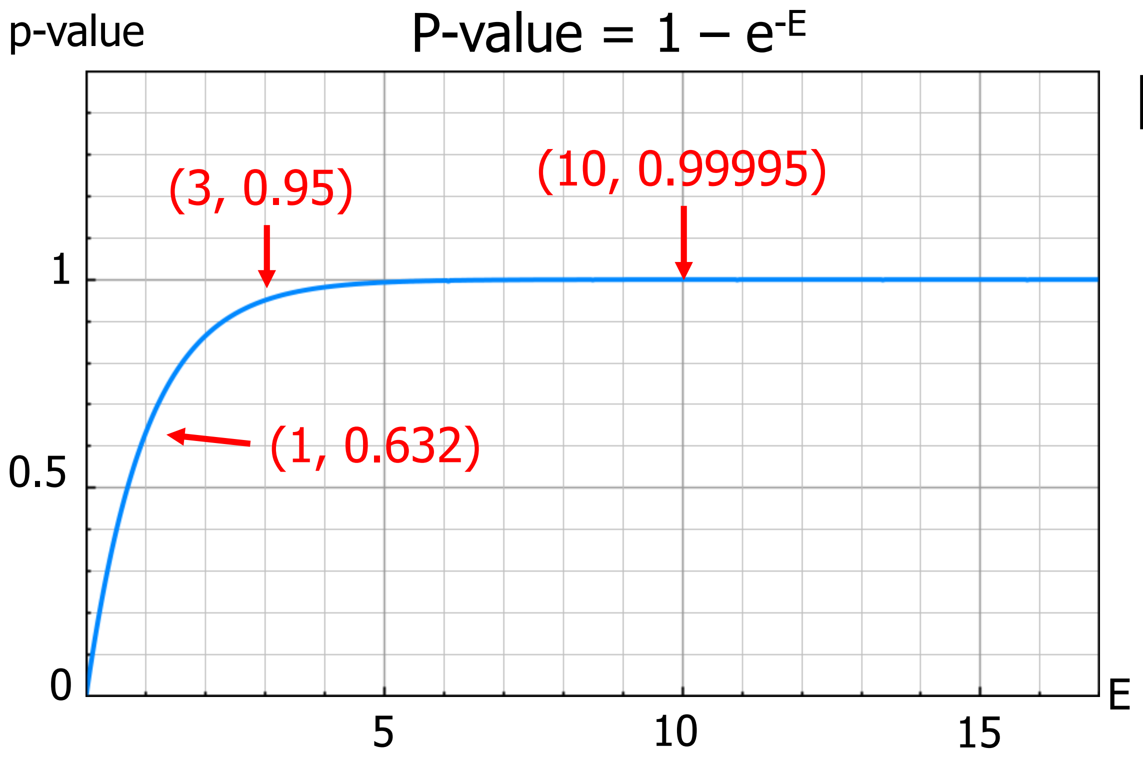
P-value

- Let X be the number of random alignments with score $\geq$ S.
- X ~ Poisson(E) where $E = Kmne^{-\lambda S}$ is the E-score.
- Pr(exactly x random alignments with score $\geq$ S) $= \frac{e^{-E}E^{x}}{x!}$
  - where $E = Kmne^{-\lambda S}$ is the E-score

- Definition: P-value is the probability of getting at least 1 random alignment with score $\geq$ S.
- Hence, p-value = Pr(X $\geq$ 1) $= 1 - \Pr(X = 0) = 1 - e^{-E}$.

Why there is no p-value in BLAST?

- In BLAST, p-value is not shown since we expect p-value and E-value are approximately the same when E<0.01 while p-value is almost 1 when E>10.



Note:

When E=1, p-value is $1-e^{-1} = 0.632$.
When E=3, p-value is $1-e^{-3} = 0.95$.
When E=10, p-value is $1-e^{-10} = 0.99995$
When E increases, p-value is approaching 1.
When E>10, p-value≈1.

When E=0.1, p-value is $1 – e^{-0.1} = 0.095$.
When E=0.01, p-value is $1 – e^{-0.01} = 0.00995$.
When E<0.01, $1-e^{-E} \approx E$.

Local alignment with gaps

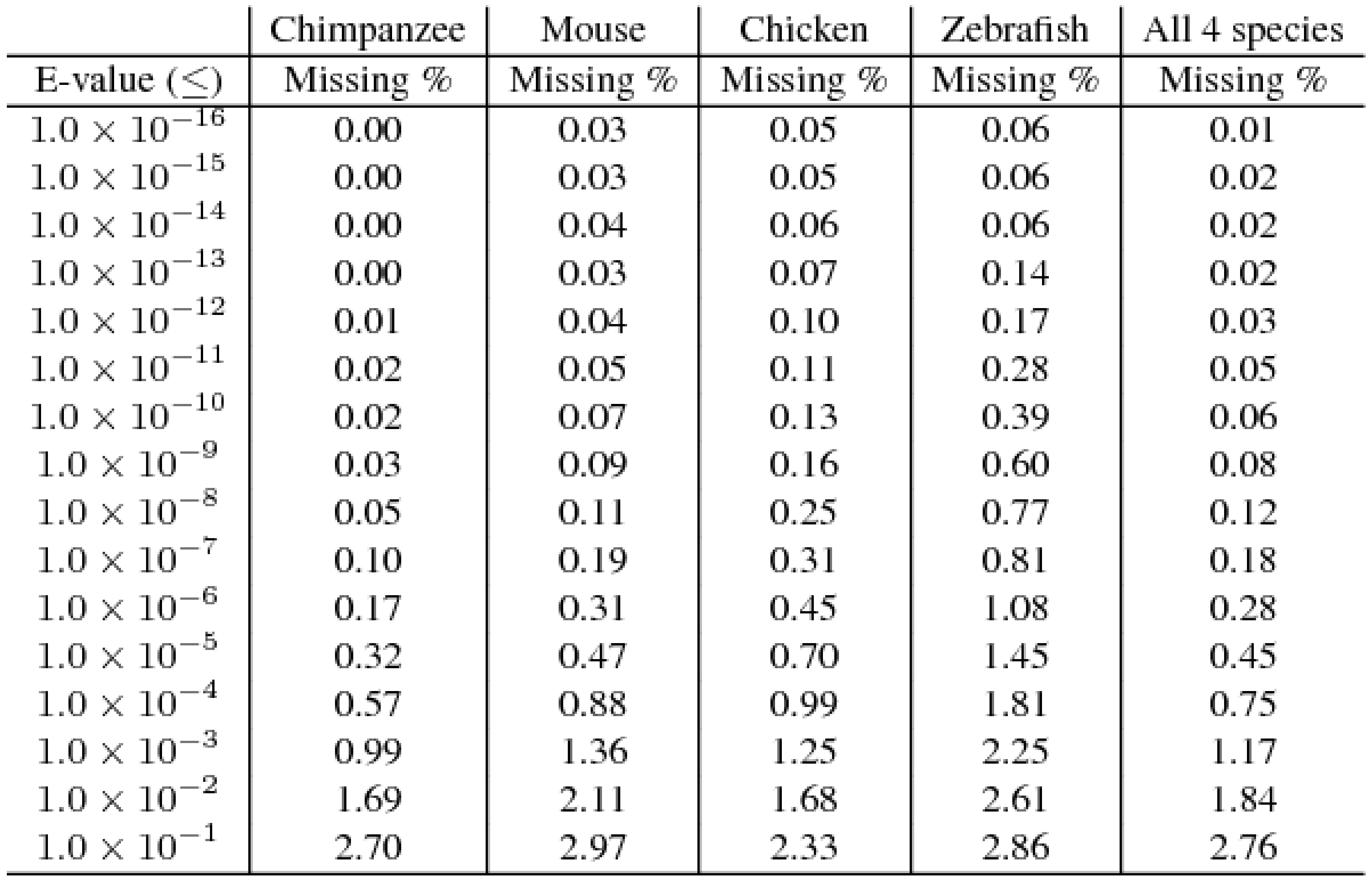
- There is no solid theoretical foundation for local alignment with gaps.
- Moreover, experimental results suggested that the theory for ungapped local alignment can be applied to the gapped local alignment as well.

Completeness of BLAST (I)

- BLAST is the most popular solution for finding local alignments. It is well-known that BLAST is heuristics and it will miss solution.
- We would like to check how many good alignments are missed by BLAST.

- We extracted 2000 mRNA sequences from each of the 4 different species. We aligned them on human genome. Then, we checked how many significant alignments are missed by BLAST.

Completeness of BLAST (II)



	Chimpanzee	Mouse	Chicken	Zebrafish	All 4 species
E-value ($\leq$)	Missing %	Missing %	Missing %	Missing %	Missing %
1.0×10^{-16}	0.00	0.03	0.05	0.06	0.01
1.0×10^{-15}	0.00	0.03	0.05	0.06	0.02
1.0×10^{-14}	0.00	0.04	0.06	0.06	0.02
1.0×10^{-13}	0.00	0.03	0.07	0.14	0.02
1.0×10^{-12}	0.01	0.04	0.10	0.17	0.03
1.0×10^{-11}	0.02	0.05	0.11	0.28	0.05
1.0×10^{-10}	0.02	0.07	0.13	0.39	0.06
1.0×10^{-9}	0.03	0.09	0.16	0.60	0.08
1.0×10^{-8}	0.05	0.11	0.25	0.77	0.12
1.0×10^{-7}	0.10	0.19	0.31	0.81	0.18
1.0×10^{-6}	0.17	0.31	0.45	1.08	0.28
1.0×10^{-5}	0.32	0.47	0.70	1.45	0.45
1.0×10^{-4}	0.57	0.88	0.99	1.81	0.75
1.0×10^{-3}	0.99	1.36	1.25	2.25	1.17
1.0×10^{-2}	1.69	2.11	1.68	2.61	1.84
1.0×10^{-1}	2.70	2.97	2.33	2.86	2.76

- 2000 queries for each species.
- BLAST only missed 0.06% of those 8000 queries (with E-value smaller than $1.0\text{x}10^{-10}$).
- In conclusion, BLAST is accurate enough in most cases, yet the few alignments missed could be critical for biological research.

Variation of BLAST

- MegaBLAST
- BLAT
- PatternHunter
- PSI-BLAST

MegaBLAST

- BLAST targets to find alignments with similarity $\approx$ 89%.
- For many database queries, the query sequences are highly similar with the database sequences.
- MegaBLAST targets to find alignments with similarity > 95%. (For example, compare homologous sequences between human and chimpanzee.)
- Up to 10 times faster than the traditional BLAST.
- Now, it is the default database search method in NCBI.

MegaBLAST uses longer w-tuples

- MegaBLAST targets to find high similarity alignment.
- Long exact match hits are expected to occur between query and database sequences.
- Hence, MegaBLAST uses longer w-tuple to find hits.
- Unlike BLAST, which sets w=11. MegaBLAST sets w=28.
- This modification reduces the number of hits. However, it also reduces the sensitivity.

Penalty for gaps

- Since MegaBLAST targets high homology alignment, we expect it has less long gaps.
- MegaBLAST uses linear gap penalty instead of affine gap penalty.

Why $r_{ind} = r_{mis} – r_{mat}/2$?

- For efficiency purpose, MegaBLAST computes differences between two sequences instead of computing their similarity.
- Below lemma shows that the similarity score can be computed by the difference score when $r_{ind} = r_{mis} – r_{mat}/2$.

- **Lemma**: Suppose $r_{ind} = r_{mis} – r_{mat}/2$. An alignment of S[1..n] and T[1..m] with edit distance d has score $(n+m)r_{mat}/2 – d\,(r_{mat} – r_{mis})$.
- Proof: Exercise!

Correctness of the lemma

- $r_{ind} = r_{mis} – r_{mat}/2 \Leftrightarrow 2\ r_{ind} + r_{mat} = 2\ r_{mis}$.
- For any two mismatches in an alignment, we can replace it by two indels and one match. Then, both the alignment score and the edit distance are the same.

```
……A……G……        ……A-……G……
……C……T……        ……-C……G……
```

Score = $2\ r_{mis}$
Edit dist = 2

Score = $2\ r_{ind} + r_{mat}$
Edit dist = 2

MegaBLAST score matrix

	A	C	G	T
A	1	-2	-2	-2
C	-2	1	-2	-2
G	-2	-2	1	-2
T	-2	-2	-2	1

MegaBLAST score matrix

- MegaBLAST targets 95% similarity (i.e. τ=0.95).
- Match/Mismatch ratio = $\frac{-\log(4*\tau)}{\log\left(4*\frac{(1-\tau)}{3}\right)} = \frac{-\lg(4*0.95)}{\lg(4*0.05/3)} \approx 0.5.$
- MegaBLAST sets the default match score is r_{mat}=1 and mismatch score is r_{mis}=-2.
- MegaBLAST sets $r_{ind} = r_{mis} – r_{mat}/2 = -2 – 1/2 = -2.5$.

Example

$$V[i,j] = \max\begin{cases} V[i-1,j-1] + \delta(S[i],T[j]) \\ V[i-1,j] - 2.5 \\ V[i,j-1] - 2.5 \end{cases}$$

$\delta(x,y) = 1$ if x=y; otherwise, $\delta(x,y) = -2$.

Match score r_{mat} = +1
Mismatch score r_{mis} = -2.
Indel score r_{ind} = -2.5

S = GTACAATCCC
T = GTAGCAATGT

S = GTA-CAAT
T = GTAGCAAT

	V	_ 0	G 1	T 2	A 3	G 4	C 5	A 6	A 7	T 8	G 9	T 10
_	0	0	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20	-22.5	-25
G	1	-2.5	1	-1.5	-4	-6.5	-9	-11.5	-14	-16.5	-19	-21.5
T	2	-5	-1.5	2	-0.5	-3	-5.5	-8	-10.5	-13	-15.5	-18
A	3	-7.5	-4	-0.5	3	0.5	-2	-4.5	-7	-9.5	-12	-14.5
C	4	-10	-6.5	-3	0.5	1	1.5	-1	-3.5	-6	-8.5	-11
A	5	-12.5	-9	-5.5	-2	-1.5	-1	2.5	0	-2.5	-5	-7.5
A	6	-15	-11.5	-8	-4.5	-4	-3.5	0	3.5	1	-1.5	-4
T	7	-17.5	-14	-10.5	-7	-6.5	-6	-2.5	1	4.5	2	-0.5
C	8	-20	-16.5	-13	-9.5	-9	-8.5	-5	-1.5	2	2.5	0
C	9	-22.5	-19	-15.5	-12	-11.5	-11	-7.5	-4	-0.5	0	0.5
C	10	-25	-21.5	-18	-14.5	-14	-13.5	-10	-6.5	-3	-2.5	-2

Example

$$D[i,j] = \min \begin{cases} D[i-1,j-1] + \delta(S[i],T[j]) \\ D[i-1,j] + 1 \\ D[i,j-1] + 1 \end{cases}$$

$\delta(x,y) = 0$ if x=y; otherwise, $\delta(x,y) = 1$.

Let D[i,j] be the minimum number of differences between the alignment of S[1..i] and T[1..j].

```
S = GTACAATCCC
T = GTAGCAATGT
```

```
S = GTA-CAAT
T = GTAGCAAT
```

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3	4	5	6	7	8	9	10
G	1	1	0	1	2	3	4	5	6	7	8	9
T	2	2	1	0	1	2	3	4	5	6	7	8
A	3	3	2	1	0	1	2	3	4	5	6	7
C	4	4	3	2	1	1	1	2	3	4	5	6
A	5	5	4	3	2	2	2	1	2	3	4	5
A	6	6	5	4	3	3	3	2	1	2	3	4
T	7	7	6	5	4	4	4	3	2	1	2	3
C	8	8	7	6	5	5	5	4	3	2	2	3
C	9	9	8	7	6	6	6	5	4	3	3	3
C	10	10	9	8	7	7	7	6	5	4	4	4

Example

$$V[i,j] = \frac{(i+j)}{2} - 3D[i,j]$$

Match score r_{mat} = +1
Mismatch score r_{mis} = -2.
Indel score r_{ind} = -2.5

		_	G	T	A	G	C	A	A	T	G	T
	V	0	1	2	3	4	5	6	7	8	9	10
_	0	0	-2.5	-5	-7.5	-10	-12.5	-15	-17.5	-20	-22.5	-25
G	1	-2.5	1	-1.5	-4	-6.5	-9	-11.5	-14	-16.5	-19	-21.5
T	2	-5	-1.5	2	-0.5	-3	-5.5	-8	-10.5	-13	-15.5	-18
A	3	-7.5	-4	-0.5	3	0.5	-2	-4.5	-7	-9.5	-12	-14.5
C	4	-10	-6.5	-3	0.5	1	1.5	-1	-3.5	-6	-8.5	-11
A	5	-12.5	-9	-5.5	-2	-1.5	-1	2.5	0	-2.5	-5	-7.5
A	6	-15	-11.5	-8	-4.5	-4	-3.5	0	3.5	1	-1.5	-4
T	7	-17.5	-14	-10.5	-7	-6.5	-6	-2.5	1	4.5	2	-0.5
C	8	-20	-16.5	-13	-9.5	-9	-8.5	-5	-1.5	2	2.5	0
C	9	-22.5	-19	-15.5	-12	-11.5	-11	-7.5	-4	-0.5	0	0.5
C	10	-25	-21.5	-18	-14.5	-14	-13.5	-10	-6.5	-3	-2.5	-2

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3	4	5	6	7	8	9	10
G	1	1	0	1	2	3	4	5	6	7	8	9
T	2	2	1	0	1	2	3	4	5	6	7	8
A	3	3	2	1	0	1	2	3	4	5	6	7
C	4	4	3	2	1	1	1	2	3	4	5	6
A	5	5	4	3	2	2	2	1	2	3	4	5
A	6	6	5	4	3	3	3	2	1	2	3	4
T	7	7	6	5	4	4	4	3	2	1	2	3
C	8	8	7	6	5	5	5	4	3	2	2	3
C	9	9	8	7	6	6	6	5	4	3	3	3
C	10	10	9	8	7	7	7	6	5	4	4	4

Property of D table

- D[i-1,j-1] ≤ D[i,j] ≤ D[i-1,j-1]+1

- Diagonal k consists of entries D[i,j] such that i – j = k.
- Property: The values are monotonic increasing for every diagonal.

Diagonal 0

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3	4	5	6	7	8	9	10
G	1	1	0	1	2	3	4	5	6	7	8	9
T	2	2	1	0	1	2	3	4	5	6	7	8
A	3	3	2	1	0	1	2	3	4	5	6	7
C	4	4	3	2	1	1	1	2	3	4	5	6
A	5	5	4	3	2	2	2	1	2	3	4	5
A	6	6	5	4	3	3	3	2	1	2	3	4
T	7	7	6	5	4	4	4	3	2	1	2	3
C	8	8	7	6	5	5	5	4	3	2	2	3
C	9	9	8	7	6	6	6	5	4	3	3	3
C	10	10	9	8	7	7	7	6	5	4	4	4

Diagonal 5

Proof: $D[i-1,j-1] \leq D[i,j] \leq D[i-1,j-1]+1$

- **Lemma**: $D[i,j] \leq D[i-1,j-1]+1$, $D[i-1,j]+1$, $D[i,j-1]+1$.
- Proof: $D[i,j]= \min\{D[i-1,j-1]+\delta(S[i],T[j]), D[i-1,j]+1, D[i,j-1]+1\}$

- **Lemma**: $D[i,j]\leq D[i+1,j+1]$.
- Proof: By induction. Suppose $D[x,y] \leq D[x+1,y+1]$ for $x+y<i+j$.
- By contrary, suppose $D[i,j]>D[i+1,j+1]$.
- Since $D[i+1,j+1]=\min\{D[i,j]+\delta(S[i+1],T[j+1]), D[i+1,j]+1, D[i,j+1]+1\}$, we have either (1) $D[i+1,j+1]=D[i+1,j]+1$ or (2) $D[i+1,j+1]=D[i,j+1]+1$.
    - Case 1: By induction, $D[i+1,j]\geq D[i,j-1]$.
    - Since $D[i,j-1]+1 \geq D[i,j]$, we have $D[i+1,j+1]=D[i+1,j]+1\geq D[i,j-1]+1 \geq D[i,j]$. Contradiction.
    - Case 2: By induction, $D[i,j+1]\geq D[i-1,j]$.
    - Since $D[i-1,j]+1 \geq D[i,j]$, we have $D[i+1,j+1]=D[i,j+1]+1\geq D[i-1,j]+1 \geq D[i,j]$. Contradiction.
- The lemma is true.

Diagonal and R(d,k)

- Diagonal k consists of entries D[i,j] such that i – j = k.
- Define R(d,k) = max{ i | D[i,i-k]=d}.

- E.g.
  - Diagonal 0: R(0,0)=3, R(1,0)=4, R(2,0)=8, R(3,0)=9 and R(4,0)=10.
  - Diagonal 5: R(0,5)=…=R(4,5)=invalid, R(5,5)=8, R(6,5)=9 and R(7,5)=10.

Diagonal 0

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3	4	5	6	7	8	9	10
G	1	1	0	1	2	3	4	5	6	7	8	9
T	2	2	1	0	1	2	3	4	5	6	7	8
A	3	3	2	1	0	1	2	3	4	5	6	7
C	4	4	3	2	1	1	1	2	3	4	5	6
A	5	5	4	3	2	2	2	1	2	3	4	5
A	6	6	5	4	3	3	3	2	1	2	3	4
T	7	7	6	5	4	4	4	3	2	1	2	3
C	8	8	7	6	5	5	5	4	3	2	2	3
C	9	9	8	7	6	6	6	5	4	3	3	3
C	10	10	9	8	7	7	7	6	5	4	4	4

Diagonal 5

Computing R(d,k)

- The computation of R(d,k) has two steps.
- 1. Find the biggest index i such that the alignment between S[1..i] and T[1..i-k] has d differences and the last aligned pair is not match.
- 2. Find R(d,k).

- Step 1 is computed based on the following lemma.

Lemma: Let i be the biggest index such that the alignment between S[1..i] and T[1..i-k] has d differences and the last aligned pair is not match. We have:

$$i = \max \left\{ \begin{array}{c} R(d-1,k)+1 \\ R(d-1,k-1)+1 \\ R(d-1,k+1) \end{array} \right\}.$$

Proof of the lemma

Lemma: Let i be the biggest index such that the alignment between S[1..i] and T[1..i-k] has d differences and the last aligned pair is not match. We have:

$$i = \max \left\{ \begin{matrix} R(d-1, k) + 1 \\ R(d-1, k-1) + 1 \\ R(d-1, k+1) \end{matrix} \right\}.$$

- Proof: Three cases: Last aligned pair is (1) deletion, (2) mismatch and (3) insertion.
- The largest index i such that D[i,i-k]=d and the last aligned pair is (S[i],-), that is, deletion.
  - This means that i-1 is the biggest index such that D[i-1, i-k]=d-1. This means that i-1 = R(d-1, k-1).
- The largest index i such that D[i,i-k]=d and the last aligned pair is (S[i],T[i-k]) where S[i]$\neq$T[i-k].
  - This means that i-1 is the biggest index such that D[i-1,i-1-k]=d-1. This means that i-1=R(d-1,k).
- The largest index i such that D[i,i-k]=d and the last aligned pair is (-,T[i-k]).
  - This means that i-1 is the biggest index such that D[i-1, i-k]=d-1. This means that i-1 = R(d-1, k-1).
- We choose the maximum index i among the three cases.

Computing R(d,k)

- After step 1, i is the biggest index such that the alignment between S[1..i] and T[1..i-k] has d differences and the last aligned pair is not match.
- Suppose R(d,k)=j. This means that the alignment between S[i..j] and T[1..j-k] has d differences.
- If j>i, the alignment between S[i+1..j] and T[i-k+1..j-k] must be matches. Hence, S[i+1..j]=T[i-k+1..j-k].
- j can be found by Extend(i,k)
- Hence, R(d,k)= Extend(i, k).

```
Extend(i,k)
  while (S[i+1]=T[i-k+1]) {
    i++;
  }
  return i;
```

Algorithm

```
• Set R(0,0)=extend(0, 0);
• d=0; $L_0$=$U_0$=0.
• Repeat
    • d=d+1
    • For k=$L_{d-1}$-1 to $U_{d-1}$+1
        • $i = \max\left\{\begin{matrix} R(d-1,k)+1 \\ R(d-1,k-1)+1 \\ R(d-1,k+1) \end{matrix}\right\}$
        • R(d,k)=Extend(i, k);
    • Set ($L_d$,$U_d$) = be a subrange of ($L_{d-1}$-1,$U_{d-1}$+1) such that R(d,$L_d$) and R(d,$U_d$) are valid;
• Until ($L_d$-1,$U_d$+1) is an invalid range
```

Example

- First, we handle d=0 (no mismatch).
- We set R(0,0) = Extend(0,0).
- We set $L_0=0, U_0=0$.

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0										
G	1		0									
T	2			0								
A	3				0							
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Example

- Next, we handle d=1 (i.e. 1 difference)
- We need to find R(1,k) for $-1=L_0-1 \leq k \leq U_0+1=1$.
- By $i = \max \begin{Bmatrix} R(d-1,k-1)+1 \\ R(d-1,k)+1 \\ R(d-1,k+1) \end{Bmatrix}$ and R(d,k)=Extend(i,k), we have:
- k=-1: i=R(0,0)=3, R(1,-1) = Extend(3, -1) = 7
- k=0: i=R(0,0)+1=4, R(1,0)=Extend(4, 0)=4
- k=1: i=R(0,0)+1=4, R(1,1)=Extend(4, 1)=4
- So, $L_1=-1$ and $U_1=1$

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1									
G	1	1	0	1								
T	2		1	0	1							
A	3			1	0	1						
C	4				1	1	1					
A	5							1				
A	6								1			
T	7									1		
C	8											
C	9											
C	10											

Example

- Next, we handle d=2 (i.e. 2 difference)
- We need to find R(2,k) for $-2=L_1-1 \leq k \leq U_1+1=2$.
- By $i = \max\begin{Bmatrix} R(d-1,k-1)+1 \\ R(d-1,k)+1 \\ R(d-1,k+1) \end{Bmatrix}$ and R(d,k)=Extend(i,k), we have:
- k=-2: i=R(1,-1)=7, R(2,-2) = Extend(7, -2) = 7
- k=-1: i=max{R(1,-1)+1, R(1,0)}=8, R(2,-1)=Extend(8, -1)=8
- k=0: i=max{R(1,-1)+1, R(1,0)+1, R(1,1)}=8, R(2,0)=Extend(8, 0)=0
- K=1: i=max{R(1,1)+1,R(1,0)+1}=5, R(2,1)=Extend(5, 1)=5
- k=2: i=R(1,1)+1=5, R(2,2)=Extend(5, 2)=5
- So, $L_2=-2$ and $U_2=2$

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2								
G	1	1	0	1	2							
T	2	2	1	0	1	2						
A	3		2	1	0	1	2					
C	4			2	1	1	1	2				
A	5				2	2	2	1	2			
A	6							2	1	2		
T	7								2	1	2	
C	8									2	2	
C	9											
C	10											

Example

- Next, we handle d=3 (i.e. 3 difference)
- We need to find R(3,k) for $-3=L_2-1 \le k \le U_2+1=3$.
- By $i = \max\left\{\begin{matrix} R(d-1,k-1)+1 \\ R(d-1,k)+1 \\ R(d-1,k+1) \end{matrix}\right\}$ and R(d,k)=Extend(i,k), we have:
- k=-3: i=R(2,-2)=7, R(3,-3) = Extend(7, -3) = 10
- k=-2: i=max{R(2,-2)+1, R(2,-1)}=8, R(3,-2)=Extend(8,-2)=8
- k=-1: i=max{R(2,-2)+1, R(2,-1)+1, R(2,0)}=9, R(3,-1)=Extend(9,-1)=9
- k=0: i=max{R(2,-1)+1, R(2,0)+1, R(2,1)}=9, R(3,0)=Extend(9,0)=9
- k=1: i=max{R(2,0)+1, R(2,1)+1, R(2,2)}=9, R(3,1)=Extend(9, 1)=9
- k=2: i=max{R(2,1)+1, R(2,2)+1}=6, R(3,2)=Extend(6,2)=6
- k=3: i=R(2,2)+1=6, R(3,3)=Extend(6,3)=6
- So, $L_3=-3$ and $U_3=3$

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3							
G	1	1	0	1	2	3						
T	2	2	1	0	1	2	3					
A	3	3	2	1	0	1	2	3				
C	4		3	2	1	1	1	2	3			
A	5			3	2	2	2	1	2	3		
A	6				3	3	3	2	1	2	3	
T	7							3	2	1	2	3
C	8								3	2	2	3
C	9									3	3	3
C	10											

Example

- When we handle d=10 (10 differences), we obtain the final table.

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0	1	2	3	4	5	6	7	8	9	10
G	1	1	0	1	2	3	4	5	6	7	8	9
T	2	2	1	0	1	2	3	4	5	6	7	8
A	3	3	2	1	0	1	2	3	4	5	6	7
C	4	4	3	2	1	1	1	2	3	4	5	6
A	5	5	4	3	2	2	2	1	2	3	4	5
A	6	6	5	4	3	3	3	2	1	2	3	4
T	7	7	6	5	4	4	4	3	2	1	2	3
C	8	8	7	6	5	5	5	4	3	2	2	3
C	9	9	8	7	6	6	6	5	4	3	3	3
C	10	10	9	8	7	7	7	6	5	4	4	4

X-drop criteria

- X-drop criteria sets V[i,j]=invalid if V[i,j]<τ-X, where
  τ = max{ max{V[p,q] | p+q<i+j }, max{ V[p,q] + δ(S[p+1],T[q+1])/2 | p+q=i+j-1} }.
- However, τ is time consuming to compute.
- Below lemma gives an alternative.

Lemma: Suppose $D[i,j] = d$. Let $d' = d - \left\lfloor \frac{X+\frac{r_{mat}}{2}}{r_{mat}-r_{mis}} \right\rfloor - 1$. Let $\tau[d'] = \max\{V[p,q] | D[p,q] \leq d'\}$. V[i,j]<$\tau$-X if and only if V[i,j]<τ[d']-X.

- ($\rightarrow$) Suppose $V[i,j]<\tau-X$. Then, there exist p,q such that $p+q<i+j$ and $V[i,j] < V[p,q] – X$.
- We claim that such (p,q) pair satisfies $D[p,q] \leq d'$.
- $D[p,q]=((p+q)r_{mat}/2 – V[p,q])/(r_{mat} – r_{mis})$
- $< ((p+q)r_{mat}/2 – V[i,j] – X)/(r_{mat} – r_{mis})$
- $< ((i+j)r_{mat}/2 – V[i,j] – X)/(r_{mat} – r_{mis})$
- $= D[i,j] – (X + r_{mat}/2)/(r_{mat} – r_{mis})$
- $= d – (X + r_{mat}/2)/(r_{mat} – r_{mis}) = d'$.

$V[i,j] = (n+m)r_{mat}/2 – D[i,j]\ (r_{mat} – r_{mis})$

- (🡐) Suppose $V[i,j] < \tau[d'] - X$.
- Note that $\tau[d'] = V[p,q]$ for some $D[p,q] \le d'$.
- If $p+q < i+j$, then $V[i,j] < V[p,q] - X \le \tau - X$.

$$d - d' = \left\lfloor \frac{X + \frac{r_{mat}}{2}}{r_{mat} - r_{mis}} \right\rfloor + 1 > \frac{X + \frac{r_{mat}}{2}}{r_{mat} - r_{mis}}$$

- If $(p+q) \ge i+j$, Select (p',q') on antidiagonal-$(i+j-1)$ such that $p-p' = q-q'$. Note that $D[p',q'] \le d'$.
- $V[p',q'] - V[i,j] = (p'+q') r_{mat}/2 - D[p',q'] (r_{mat} - r_{mis}) - (i+j) r_{mat}/2 - d (r_{mat} - r_{mis})$
- $\ge - r_{mat}/2 + (d - d') (r_{mat} - r_{mis})$
- $\ge -r_{mat}/2 + \left(\frac{X + \frac{r_{mat}}{2}}{r_{mat} - r_{mis}} \right) (r_{mat} - r_{mis}) > X$

- For antidiagonal-k, let i=B(d,k) and i'=R(d,k).
- This means that D[i,i-k]=D[i+1,i+1-k]=…=D[i',i'-k]=d.
- Let $d' = d - \left\lfloor \frac{X + \frac{r_{mat}}{2}}{r_{mat} - r_{mis}} \right\rfloor - 1$.
- Lemma: If V[i,i-k]$\geq\tau$[d']-X, then V[i+s,i+s-k] $\geq\tau$[d']-X for 0$\leq$s$\leq$i'-i.
- Proof: It follows from the fact that V[i+s,i+s-k] = V[i,i-k]+s * r_{mat}.

Algorithm

```
• Set i=extend(0, 0); R(0,0)=i; τ'=V[i, i]; τ[0]=V[i, i];
• d=0; $L_0=U_0=0$.
• Repeat
    • d=d+1; $d' = d - \left\lfloor \frac{X+\frac{r_{mat}}{2}}{r_{mat}-r_{mis}} \right\rfloor - 1$;
    • For k=$L_{d-1}$-1 to $U_{d-1}$+1
        • If V[i,i-k]<τ[d']
            • R(d,k)=invalid
        • else
            • $i = \max \left\{ \begin{matrix} R(d-1,k)+1 \\ R(d-1,k-1)+1 \\ R(d-1,k+1) \end{matrix} \right\}$
            • i'=Extend(i, k); R(d, k)=i'; τ' = max{τ', V[i', i'-k] };
    • τ[d]=τ'
    • Set ($L_d$,$U_d$) = be a subrange of ($L_{d-1}$-1,$U_{d-1}$+1) such that R(d,$L_d$) and R(d,$U_d$) are valid;
• Until ($L_d$-1,$U_d$+1) is an invalid range
```

$$V[i,j] = \frac{(i+j)}{2} - 3D[i,j]$$

Illustration of greedy algorithm with X-drop

- Suppose X=1.
- Then, $d' = d - \left\lfloor \frac{X+\frac{r_{mat}}{2}}{r_{mat}-r_{mis}} \right\rfloor - 1 = d - \left\lfloor \frac{1+\frac{1}{2}}{1-(-2)} \right\rfloor - 1 = d - 1$

- First, we handle d=0 (no mismatch).
- We set R(0,0) = Extend(0,0)=3.
- We set L_0=0,U_0=0. τ[0]=0. τ'=V[3,3]=3.

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0										
G	1		0									
T	2			0								
A	3				0							
C	4											
A	5											
A	6											
T	7											
C	8											
C	9											
C	10											

Diagonal and R(d,k)

$\tau[0]=0$. $\tau'=3$
$\tau[0]-X=-1$

- Next, we handle d=1 (i.e. 1 difference)
- $d' = d - 1 = 0$
- We need to find R(1,k) for $-1=L_0-1 \le k \le U_0+1=1$.
- By $i = \max \begin{Bmatrix} R(d-1,k-1)+1 \\ R(d-1,k)+1 \\ R(d-1,k+1) \end{Bmatrix}$ and R(d,k)=extend(i,i-k), we have:
- k=-1: i=R(0,0)=3, V[3,3+1]=0.5, R(1,-1) = Extend(3, -1) = 7, V[7,7+1]=4.5
- k=0: i=R(0,0)+1=4, V[4,4]=1, R(1,0)=Extend(4, 0)=4
- k=1: i=R(0,0)+1=4, V[4,3]=0.5, R(1,1)=Extend(4, 1)=4
- So, L_1=-1 and U_1=1
- τ'=max{0, 4.5, 1, 0.5}=4.5
- $\tau[1]$=4.5

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0										
G	1		0									
T	2			0								
A	3				0	1						
C	4				1	1	1					
A	5							1				
A	6								1			
T	7									1		
C	8											
C	9											
C	10											

Diagonal and R(d,k)

$\tau[1]=4.5.\ \tau'=4.5$
$\tau[1]\text{-}X=3.5$

- Next, we handle k=2 (i.e. 2 difference)
- We need to find R(2,k) for $-2=L_1-1 \leq k \leq U_1+1=2$.
- By $i = \max\begin{Bmatrix} R(d-1,k-1)+1 \\ R(d-1,k)+1 \\ R(d-1,k+1) \end{Bmatrix}$ and R(d,k)=extend(i,i-k), we have:
- k=-2: i=R(1,-1)=7, V[7,7+2]=2, R(2,-2) = invalid
- k=-1: i=max{R(1,-1)+1, R(1,0)}=8, V[8,8+1]=2.5, R(2,-1)= invalid
- k=0: i=max{R(1,-1)+1, R(1,0)+1, R(1,1)}=8, V[8,8]=2, R(2,0)= invalid
- k=1: i=max{R(1,1)+1,R(1,0)+1}=5, V[5,4]=-1.5<τ[1]-X, R(2,1)=invalid
- k=2: i=R(1,1)+1=5, V[5,3]=-2 <τ[1]-X, R(2,2)=invalid
- So, (L_2, U_2) is an invalid range.

		_	G	T	A	G	C	A	A	T	G	T
	D	0	1	2	3	4	5	6	7	8	9	10
_	0	0										
G	1		0									
T	2			0								
A	3				0	1						
C	4				1	1	1					
A	5				~~2~~	~~2~~	2	1				
A	6							2	1			
T	7								2	1	~~2~~	
C	8									~~2~~	~~2~~	
C	9											
C	10											

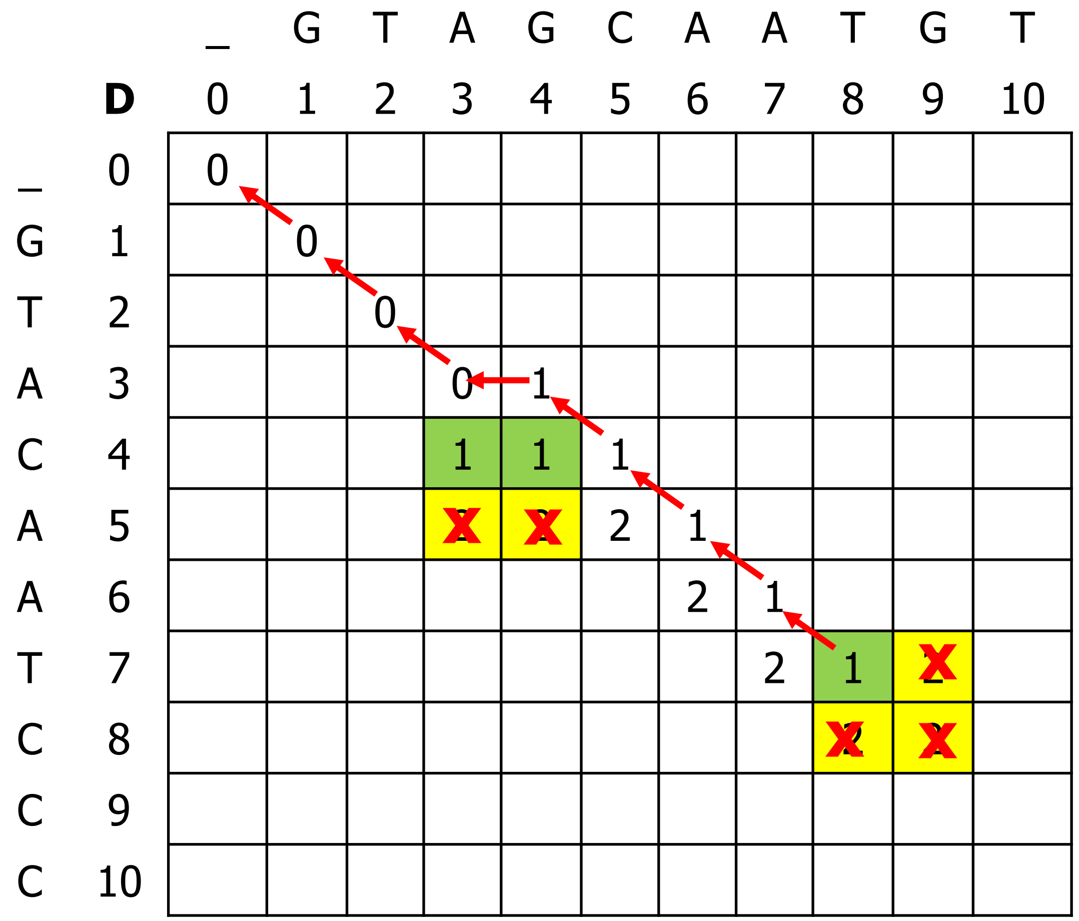
Back-tracking

$\tau[1]$=4.5. τ'=1
$\tau[1]$-X=3.5

- Find the entry with score $\tau[1]$=4.5, which is V[7,8].

- By back-tracking, we have

```
GTAGCAAT
GTA-CAAT
```



	D	0	1	2	3	4	5	6	7	8	9	10
		_	G	T	A	G	C	A	A	T	G	T
_	0	0										
G	1		0									
T	2			0								
A	3				0	1						
C	4				1	1	1					
A	5				~~2~~	~~2~~	2	1				
A	6							2	1			
T	7								2	1	~~2~~	
C	8									~~2~~	~~3~~	
C	9											
C	10											

BLAT

- BLAT aims to improve the efficiency of BLAST.
- Only for DNA.
- The main trick is to index the database and put the index in the main memory

- The algorithm is as follows.

1. For every w-tuple in the query, identify hits using the index of the database (w=11)
2. Accept hits satisfying the two-hit requirement.
3. Gapped extension.

- Note that BLAT is less sensitive than BLAST, but more sensitive than MegaBLAST.

Main trick of BLAT

- BLAST cannot build index of human genome since it it big.
- BLAT’s index stores the positions of non-overlapping w-tuples in memory.

Database = ACTTGTACTTGTACTTGTA

Index of all w-mers

w-mer	positions
ACTT	1, 7, 13
CTTG	2, 8, 14
GTAC	5, 11
TACT	6, 12
TGTA	4, 10
TTGT	3, 9, 15
TGTA	16

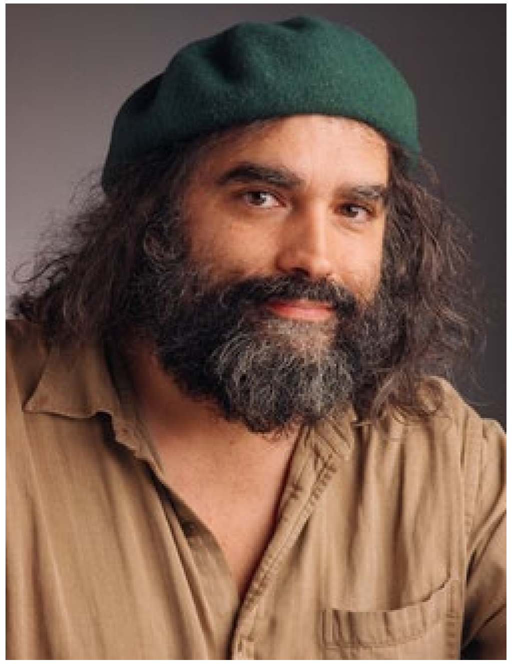
Store n integers

Index of w-mers at positions iw+1

w-mer	positions
ACTT	1, 13
GTAC	5
TTGT	9

Store n/w integers

About the inventor: Jim Kent



- Education: University of California, Santa Cruz
- Awards: Overton Prize, Benjamin Franklin Award

PatternHunter

- PatternHunter can only apply to DNA
- PatternHunter is similar to BLAST. Moreover, it uses gapped w-tuple.
  - For w=11, they use 111010010100110111
  - Example,

```
111010010100110111
ACTCCGATATGCGGTAAC
|||-|--|-|--||-|||
ACTTCACTGTGAGGCAAC
```

- They found that gapped w-tuple can increase the sensitivity while increase the efficiency.

Advantage of gapped w-tuple (I)

- Increase sensitivity
  - Gapped w-tuples are more independent.
  - Examples:
    - Two adjacent ungapped 11-tuples share 10 symbols

      -
        ```
        11111111111
         11111111111
        ```
        1/4 chances to have 2nd hit next to the 1st hit
    - Two adjacent gapped 11-tuples share 5 symbols

      -
        ```
        111010010100110111
         111010010100110111
        ```
        $1/4^6$ chances to have 2nd hit next to the 1st hit
  - If the w-tuples are more independent, the probability of having at least one hit in a homologous region is higher.

Advantage of gapped w-tuple (II)

- Reduce the number of hits.
  - For the same query length (says, 64),
    - It covers by 54 ungapped 11-tuples
    - It covers by 47 gapped 11-tuples
  - So, the number of hits is smaller.
- Thus, the efficiency is increased!

PatternHunter I

Ma et al., *Bioinformatics* 18:440-445, 2002

Proposition. The expected number of hits of a weight-W length-M model within a length-L region of similarity p is $(L - M + 1) * p^W$

Proof.

For any fixed position, the prob of a hit is p^W.

There are $L - M + 1$ candidate positions.

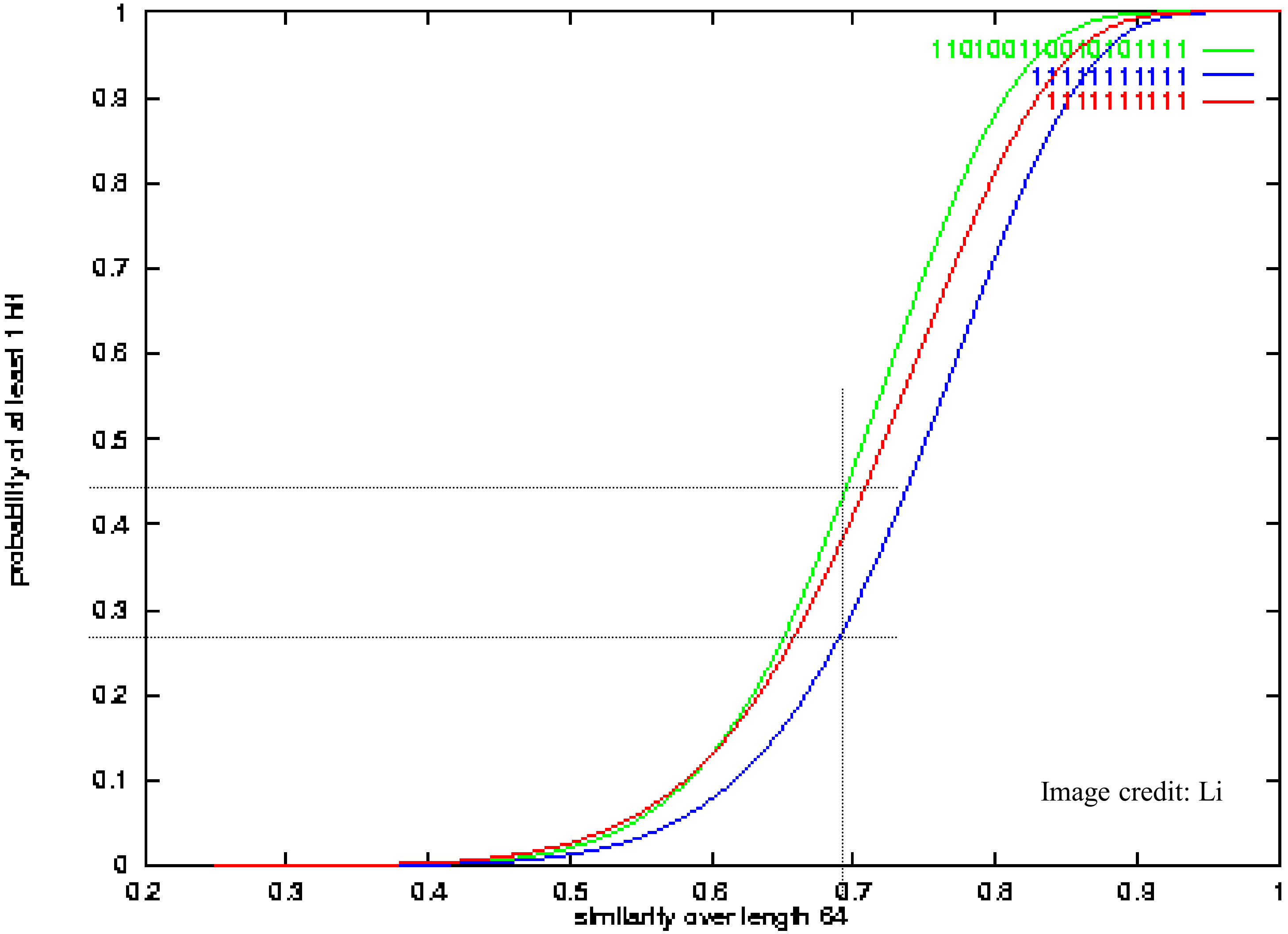
The proposition follows.

Implication

- For *$L = 1017$*
  - BLAST seed expects $(1017 - 11 + 1) * p^{11} = 1007 * p^{11}$ hits
  - But ~*1/4* of these overlap each other. So likely to have only ~$750 * p^{11}$ distinct hits
  - Our example spaced seed expects $(1017 - 18 + 1) * p^{11} = 1000 * p^{11}$ hits
  - But only $1/4^6$ of these overlap each other. So likely to have ~$1000 * p^{11}$ distinct hits

Spaced seeds likely to be more sensitive & more efficient

Sensitivity of PatternHunter I



More for PatternHunter

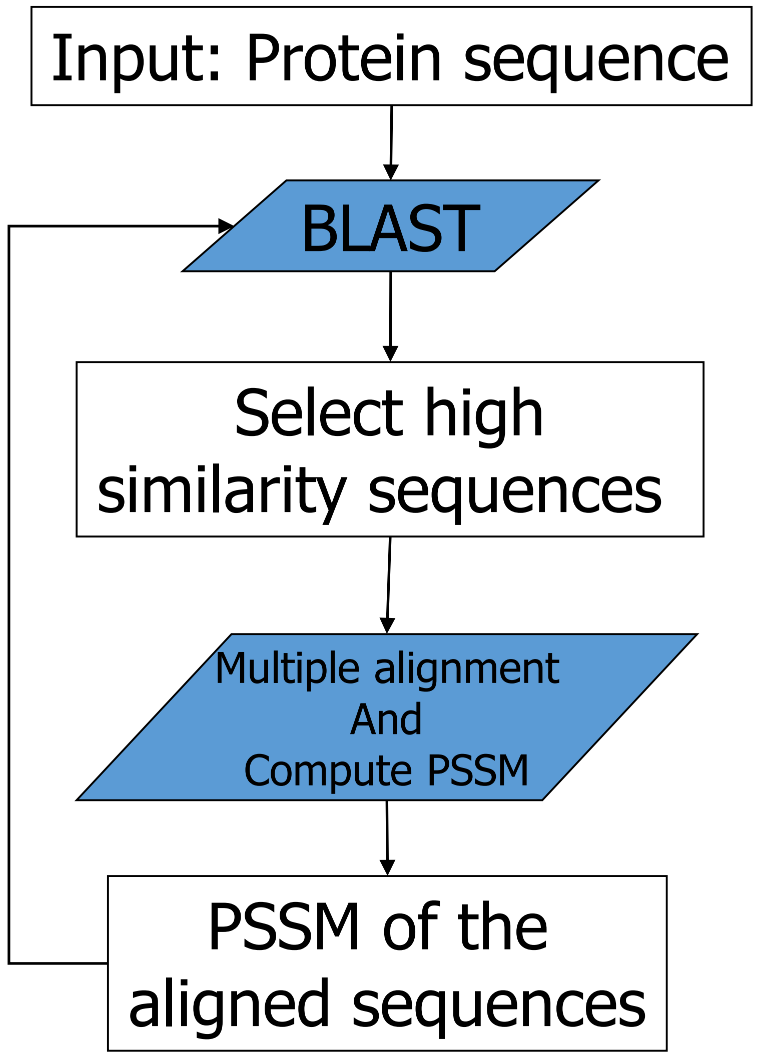
- To further improve the efficiency,
  - PatternHunter uses a variety of advanced data structures including priority queues, a variation of red-black tree, queues, hash tables.
  - PatternHunter also uses a new method of sequence alignment.
- To further improve the accuracy,
  - PatternHunter II suggested to use multiple gapped seeds.
  - They show that the accuracy can approach smith-waterman algorithm while the speed 3000 times faster than smith-waterman.
- PatternHunter II is both faster and sensitive than BLAST, MegaBLAST.

About the Inventor: Ming Li

- Ming Li
  - Canada Research Chair Professor of Bioinformatics, University Professor, Univ of Waterloo
  - Fellow, Royal Society of Canada. Fellow, ACM. Fellow, IEEE.

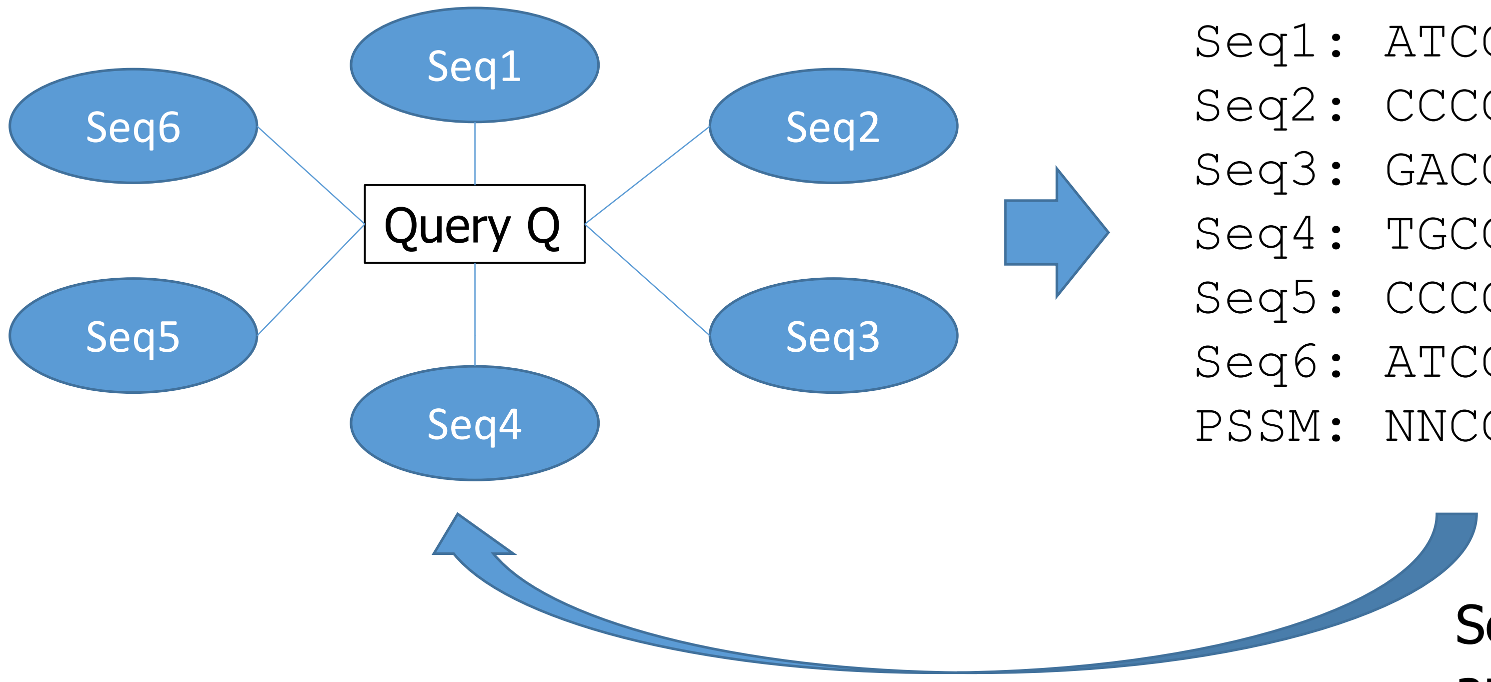
PSI-BLAST (Position Specific Iterated BLAST)

- PSI-BLAST is an implementation of BLAST for finding protein families. It allows us to detect distant homology.
- Input: a protein sequence
  - Using BLAST, we get a set of sequences that align with the query protein with E-score below a threshold, 0.01 (by default).
  - Align the selected sequences
  - Generate a PSSM profile from the multiple alignment
  - Iterate until no significant alignment found,
    - Using a modified BLAST, search the database with the PSSM profile.
    - Align the selected sequences
    - Generate a PSSM from the multiple alignment
- This version automatically combines statistically significant alignments produced by BLAST into a position-specific score matrix (PSSM).
- It is much more sensitive to weak but biologically relevant sequence similarities


Input: Protein sequence
BLAST
Select high similarity sequences
Multiple alignment And Compute PSSM
PSSM of the aligned sequences

Idea of PSI-BLAST (I)

- 1. Find a set of similar sequences of the query Q by BLAST
- 2. Use the similar sequences to generate a representative and set it to be Q.
- 3. Repeat 1 and 2.

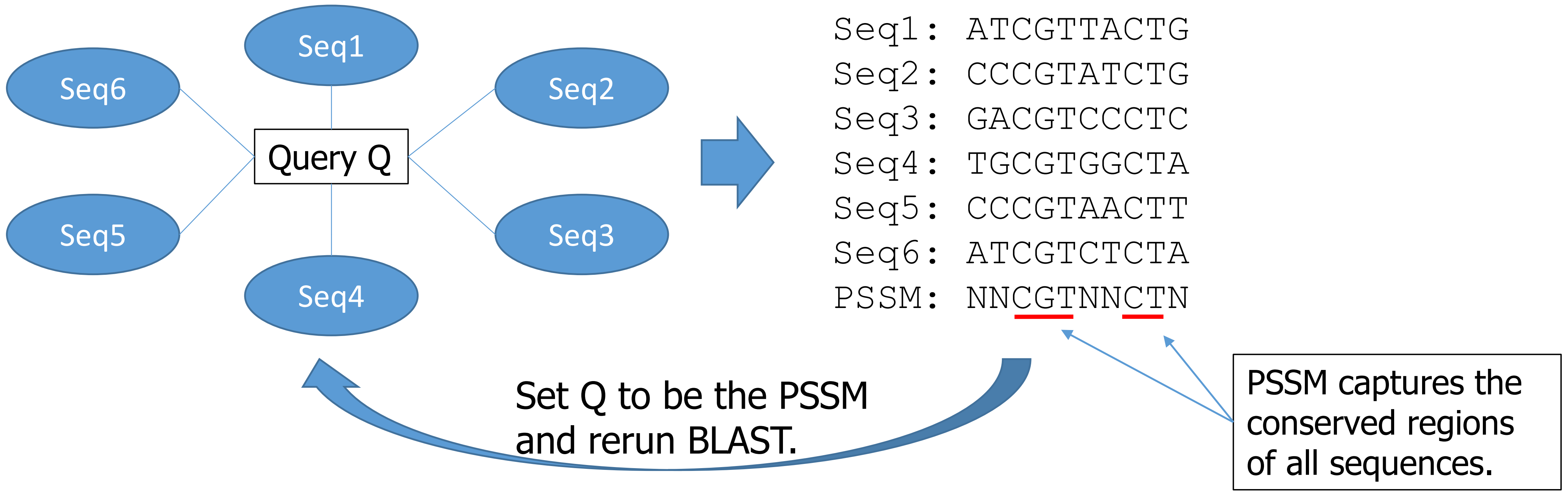


```
Seq1: ATCGTTACTG
Seq2: CCCGTATCTG
Seq3: GACGTCCCTC
Seq4: TGCGTGGCTA
Seq5: CCCGTAACTT
Seq6: ATCGTCTCTA
PSSM: NNCGTNNCTN
```

Set Q to be the PSSM and rerun BLAST.

Idea of PSSM (II)

- The PSSM will capture all the conserved amino acid residues (those residues form conserved regions like domains).
- When we run BLAST again, we can find the distant homologous sequences.


Seq1
Seq6
Seq2
Query Q
Seq5
Seq3
Seq4
Set Q to be the PSSM and rerun BLAST.
PSSM captures the conserved regions of all sequences.

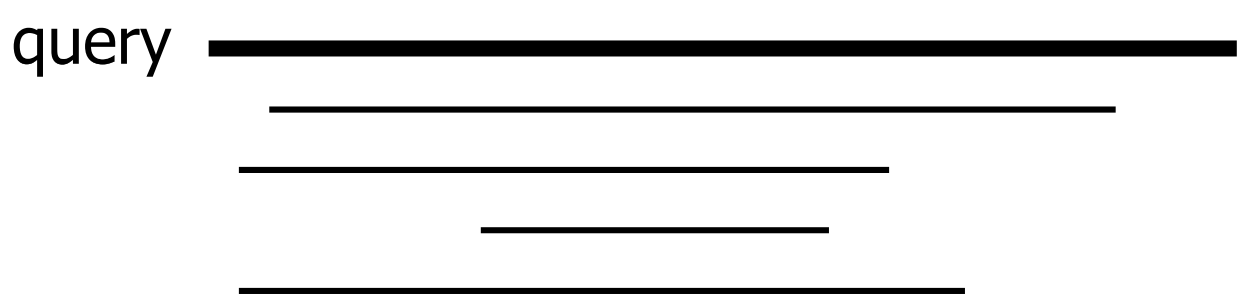
```
Seq1: ATCGTTACTG
Seq2: CCCGTATCTG
Seq3: GACGTCCCTC
Seq4: TGCGTGGCTA
Seq5: CCCGTAACTT
Seq6: ATCGTCTCTA
PSSM: NNCGTNNCTN
```

1. Find a set of sequences similar to the query

- Using BLAST 2.0, we get a set of sequences that align with the query protein with E-score below a threshold, 0.01 (by default).
- In the set of selected sequences, some have >98% identical. We keep one copy for those selected sequences which are >98% identical.

2. Multiple sequence alignment of the selected sequences

- Using the query sequence as the template, we aligned the selected sequences.
- All gap characters inserted into the query sequence are ignored.
- Note:
  - the length of the alignment is the same as the query sequence.
  - Some columns of the multiple sequence alignment may include nothing except the query.

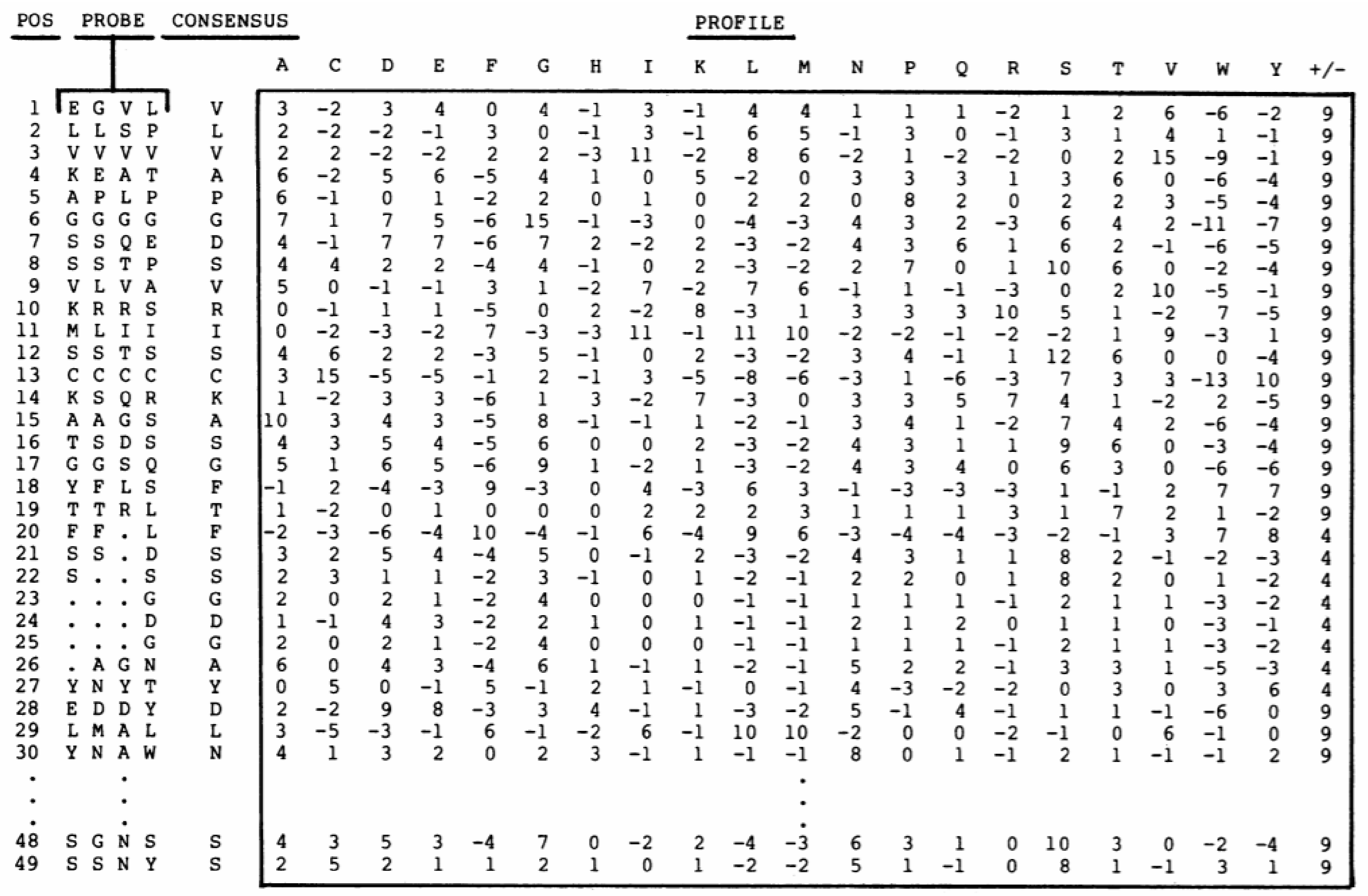


3. Generate a PSSM profile from the alignment

- Given the multiple alignment of length n,
  - We generate the position-specific score matrix (PSSM) profile, which is a 20xn matrix.
  - For each column and each residue a in the profile, we generate a log-odds score $\log(O_{ia}/P_a)$.
    - where O_{ia} is the observed frequency of residue a at position i and P_a is the expected frequency respectively of the residue a.
- Since number of sequences may be small, data-dependent pseudo frequency is added to O_{ia}.

Example

- This is an example PSSM M.
- For position p, amino acid a, $M(p,a) = \sum_{b=1}^{20} W(p,b)\delta(a,b)$, where
  - $\delta(a,b)$ is the substitution matrix,
  - $W(p,b) = \frac{n(p,b)}{n(p)}$,
  - n(p) is the number of probes in position p and
  - n(p,b) is the number of probes with amino acid b at position p.



POS	PROBE	CONSENSUS	PROFILE																				
			A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	+/-
1	E G V L	V	3	−2	3	4	0	4	−1	3	−1	4	4	1	1	1	−2	1	2	6	−6	−2	9
2	L L S P	L	2	−2	−2	−1	3	0	−1	3	−1	6	5	−1	3	0	−1	3	1	4	1	−1	9
3	V V V V	V	2	2	−2	−2	2	2	−3	11	−2	8	6	−2	1	−2	−2	0	2	15	−9	−1	9
4	K E A T	A	6	−2	5	6	−5	4	1	0	5	−2	0	3	3	3	1	3	6	0	−6	−4	9
5	A P L P	P	6	−1	0	1	−2	2	0	1	0	2	2	0	8	2	0	2	2	3	−5	−4	9
6	G G G G	G	7	1	7	5	−6	15	−1	−3	0	−4	−3	4	3	2	−3	6	4	2	−11	−7	9
7	S S Q E	D	4	−1	7	7	−6	7	2	−2	2	−3	−2	4	3	6	1	6	2	−1	−6	−5	9
8	S S T P	S	4	4	2	2	−4	4	−1	0	2	−3	−2	2	7	0	1	10	6	0	−2	−4	9
9	V L V A	V	5	0	−1	−1	3	1	−2	7	−2	7	6	−1	1	−1	−3	0	2	10	−5	−1	9
10	K R R S	R	0	−1	1	1	−5	0	2	−2	8	−3	1	3	3	3	10	5	1	−2	7	−5	9
11	M L I I	I	0	−2	−3	−2	7	−3	−3	11	−1	11	10	−2	−2	−1	−2	−2	1	9	−3	1	9
12	S S T S	S	4	6	2	2	−3	5	−1	0	2	−3	−2	3	4	−1	1	12	6	0	0	−4	9
13	C C C C	C	3	15	−5	−5	−1	2	−1	3	−5	−8	−6	−3	1	−6	−3	7	3	3	−13	10	9
14	K S Q R	K	1	−2	3	3	−6	1	3	−2	7	−3	0	3	3	5	7	4	1	−2	2	−5	9
15	A A G S	A	10	3	4	3	−5	8	−1	−1	1	−2	−1	3	4	1	−2	7	4	2	−6	−4	9
16	T S D S	S	4	3	5	4	−5	6	0	0	2	−3	−2	4	3	1	1	9	6	0	−3	−4	9
17	G G S Q	G	5	1	6	5	−6	9	1	−2	1	−3	−2	4	3	4	0	6	3	0	−6	−6	9
18	Y F L S	F	−1	2	−4	−3	9	−3	0	4	−3	6	3	−1	−3	−3	−3	1	−1	2	7	7	9
19	T T R L	T	1	−2	0	1	0	0	0	2	2	2	3	1	1	1	3	1	7	2	1	−2	9
20	F F . L	F	−2	−3	−6	−4	10	−4	−1	6	−4	9	6	−3	−4	−4	−3	−2	−1	3	7	8	4
21	S S . D	S	3	2	5	4	−4	5	0	−1	2	−3	−2	4	3	1	1	8	2	−1	−2	−3	4
22	S . . S	S	2	3	1	1	−2	3	−1	0	1	−2	−1	2	2	0	1	8	2	0	1	−2	4
23	. . . G	G	2	0	2	1	−2	4	0	0	0	−1	−1	1	1	1	−1	2	1	1	−3	−2	4
24	. . . D	D	1	−1	4	3	−2	2	1	0	1	−1	−1	2	1	2	0	1	1	0	−3	−1	4
25	. . . G	G	2	0	2	1	−2	4	0	0	0	−1	−1	1	1	1	−1	2	1	1	−3	−2	4
26	. A G N	A	6	0	4	3	−4	6	1	−1	1	−2	−1	5	2	2	−1	3	3	1	−5	−3	4
27	Y N Y T	Y	0	5	0	−1	5	−1	2	1	−1	0	−1	4	−3	−2	−2	0	3	0	3	6	4
28	E D D Y	D	2	−2	9	8	−3	3	4	−1	1	−3	−2	5	−1	4	−1	1	1	−1	−6	0	9
29	L M A L	L	3	−5	−3	−1	6	−1	−2	6	−1	10	10	−2	0	0	−2	−1	0	6	−1	0	9
30	Y N A W	N	4	1	3	2	0	2	3	−1	1	−1	−1	8	0	1	−1	2	1	−1	−1	2	9
.	.												.										
.	.												.										
.	.												.										
48	S G N S	S	4	3	5	3	−4	7	0	−2	2	−4	−3	6	3	1	0	10	3	0	−2	−4	9
49	S S N Y	S	2	5	2	1	1	2	1	0	1	−2	−2	5	1	−1	0	8	1	−1	3	1	9

FIG. 1. The concept of a profile. (*a*) A flow diagram of profile analysis. (*b*) A 49-residue sample profile for the immunoglobulin variable-region domain, generated from the four-probe sequences shown at the left (see Fig. 2*b* for details). The profile is shown in the box. The rightmost column of the profile gives the penalty for insertion/deletion (+/−). Positions 31–47 of the profile are omitted from the figure for clarity. Notice that where gaps appear in some of the probe sequences, the insertion/deletion penalty is lower than elsewhere.

Profile analysis: Detection of distantly related proteins. PNAS 1987.

4. Run BLAST again with the PSSM profile

- We apply a modified BLAST to the PSSM profile.
  - Basically, when we compare a position of the PSSM and a residue in the database, we use the corresponding log-odds score in that position.

- Repeat until we satisfy.

QUASAR

- QUASAR stands for Q-gram Alignment based on Suffix ARrays
- It is a good searching tool for identifying strong similar strings.
- Problem:
  - Input: a database D, a query S, k, w
  - Output: a set of (x, y) where
    - x and y are length-w substring in D and S, respectively
    - edit_dist(x, y) $\leq$ k

```
gcagactgctac
gccgacagccac
```

k=3
w=12
q=3
t=w+1-(k+1)q
 =12+1-(3+1)3
 =1

Observation

Lemma:

Given two length-w sequences X and Y, if they have edit distance $\leq$ k, then they share at least t common q-grams (length-q substrings) where t = w+1-(k+1)q.

Proof:

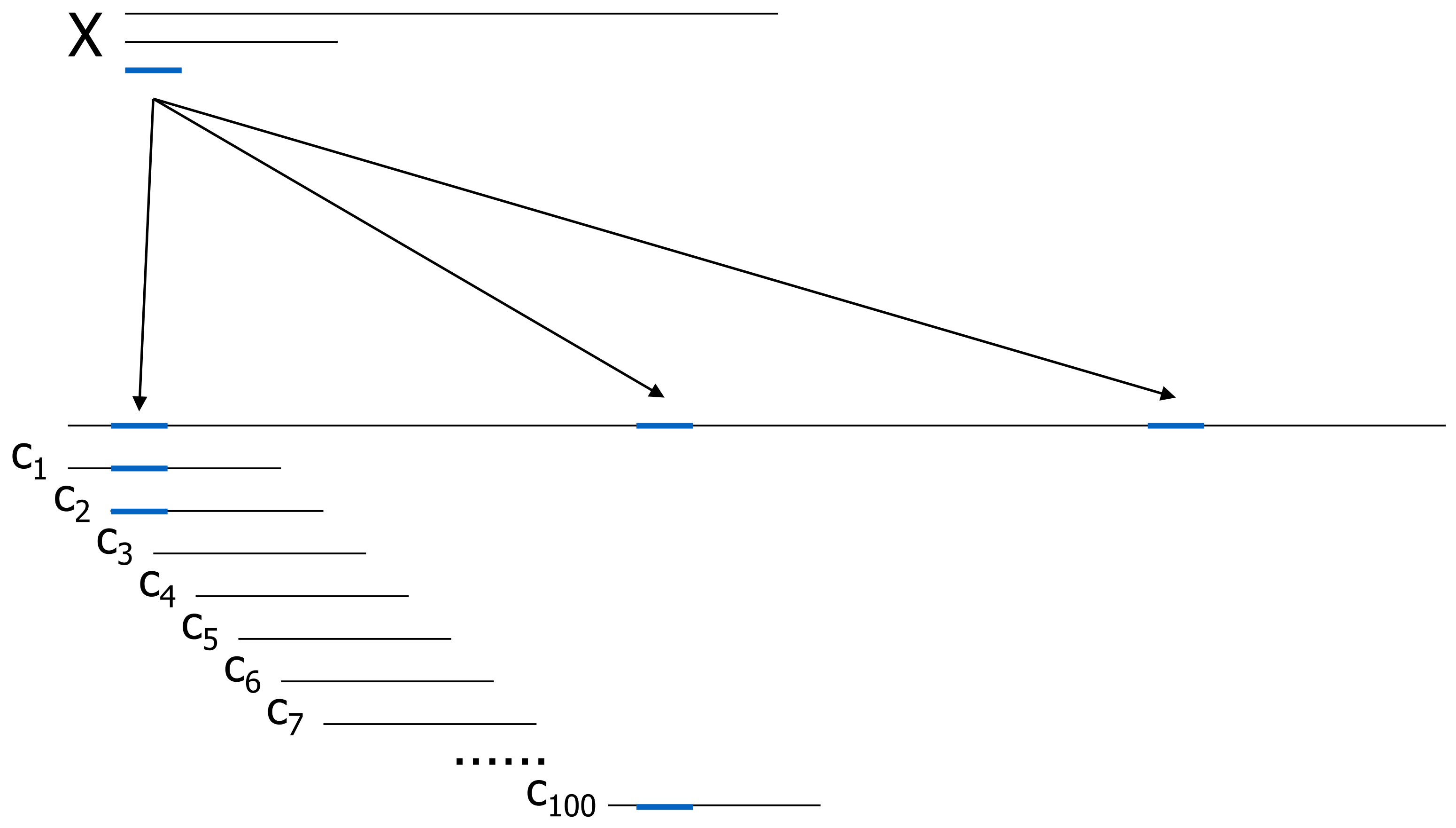
- Suppose X and Y has r differences.
- X has (w+1-q) q-grams
- Note that a q-gram in X overlaps with some difference iff X and Y does not share that q-gram
- For each difference, there are at most q q-grams overlap with the difference. In total, rq q-grams overlap with the r differences
- Thus, X and Y share (w+1-q – rq) q-grams, which is bigger than w+1-(k+1)q.

- We make use of this observation to do filtering!

Algorithm for finding potential approximate matches of S in D

- For every X = S[i..i+w-1] of the query where i=1, 2, …
  - For every length-w substring Y in D, associate a counter with it and initialize it to zero
  - For each q-gram Q in X,
    - Find the hitlist, that is, the list of positions in D so that Q occurs
    - Increment the counter for every length-w substring Y which contains Q
  - For every length-w substring Y in D with counter > t, X and Y are potential approximate match. We check it using an alignment algorithm!

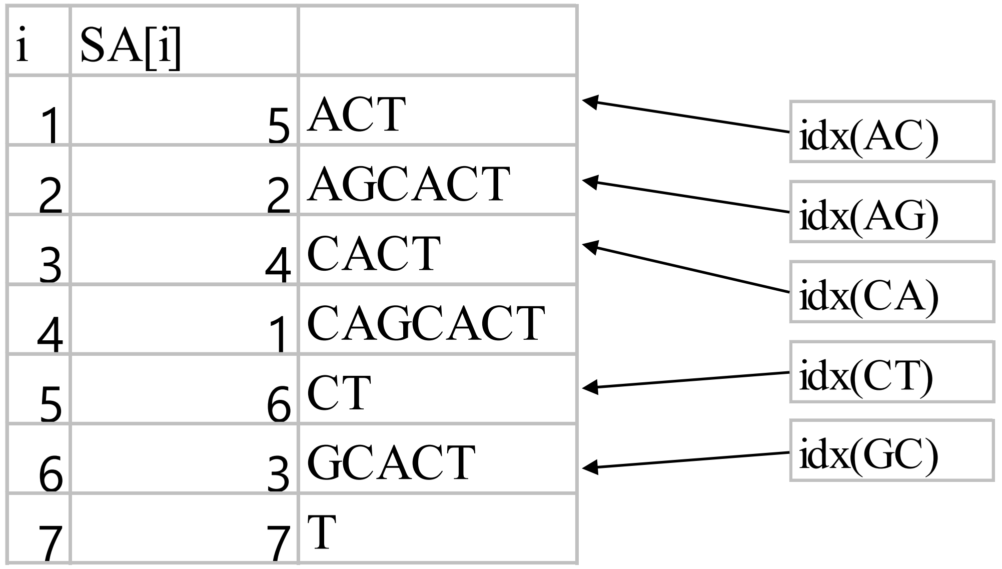
Illustration of the algorithm



How to get the hitlist?

- Based on the data-structures
  - A suffix array SA of the database D is the lexicographically ordered sequence of all suffixes in D.
  - An auxiliary array idx where for each q-gram Q, idx[Q] is the start of the hitlist for Q!

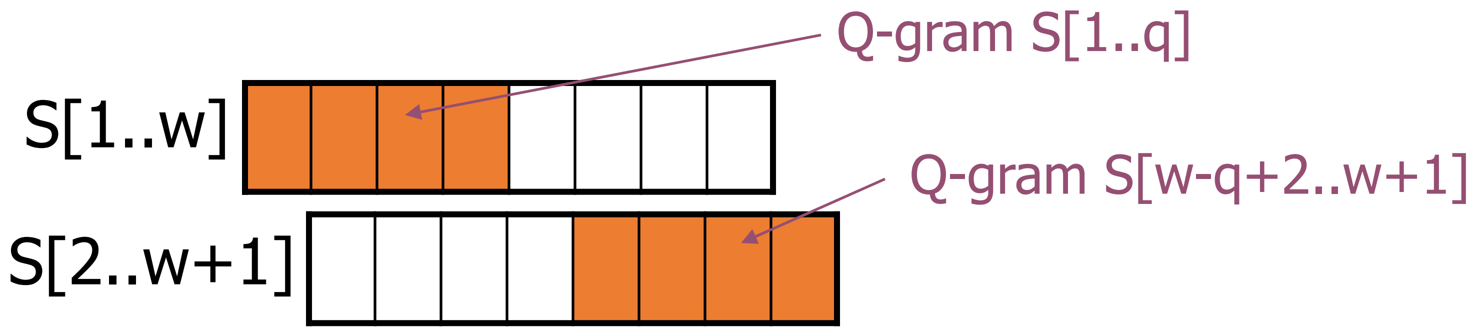
	1	2	3	4	5	6	7
Database D =	C	A	G	C	A	C	T



i	SA[i]	
1	5	ACT
2	2	AGCACT
3	4	CACT
4	1	CAGCACT
5	6	CT
6	3	GCACT
7	7	T

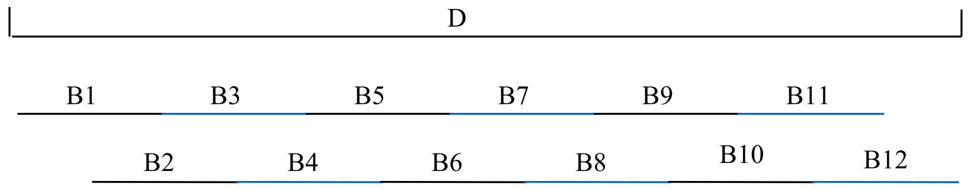
Speedup Feature 1: Window shifting

- In the previuos algorithm, building the counters list for S[i..w+i-1] is time consuming!
- Suppose the counters list for S[1..w] is given, can we determine the counters list for S[2..w+1] easily?
    - Idea: For every length-w string Y in D,
        - Decrement counter for Y if it contains q-gram S[1..q]
        - Increment counter for Y if it contains q-gram S[w-q+2..w+1]
- The window shifting idea reduce the time complexity.



Speedup Feature 2: Block addressing

- Another problem: too many counters
- Solution (Block addressing scheme):
  - Instead of associate a counter for every length-w substring Y in D
  - The database D is divided into blocks of size b (b ≥ 2w). Each block is associating a counter.
  - If a block contains more than t q-grams, this block has to be checked for approximate matches using an alignment algorithm



Weakness of QUASAR

- Extensive memory requirement
    - Construction phase:
        - Memory space = 9n (where n is DB size)
    - Query phase:
        - Memory space = 5n
- Not suitable for distant homologous sequences

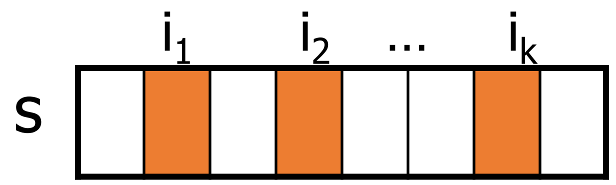
Locality-Sensitive Hashing (LSH)

LSH-ALL-PAIRS

- Input: biosequence database D
- Aim: find pairs of w-mers that differ by at most d substitutions (ungapped local alignment) in a collection of biosequences D.

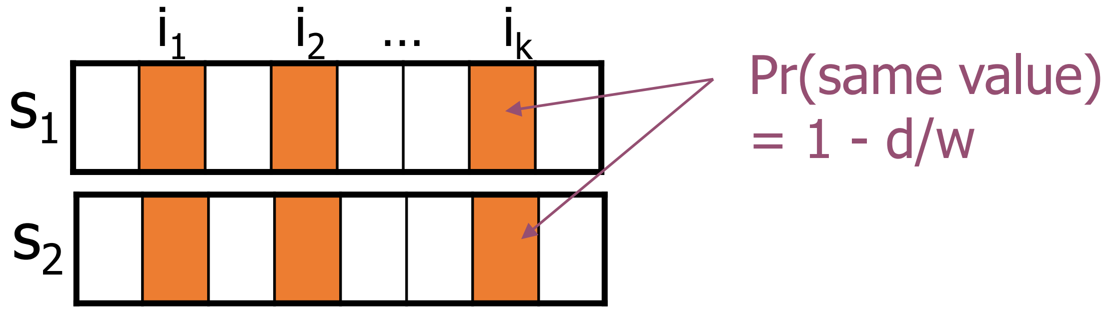
Locality-sensitive hash function

- Consider an w-mers s,
  - choose k indices $i_1, i_2, \ldots, i_k$ uniformly from the set {1, 2, …, w}
  - Define $\pi(s) = (s[i_1], s[i_2], \ldots, s[i_k])$. This function is called the locality-sensitive hash function



Property of locality-sensitive hash function (I)

- Consider two w-mers s_1 and s_2,
  - the more similar are they, the higher probability that $\pi(s_1) = \pi(s_2)$.
- More precisely, if the hamming distance of s_1 and s_2 = d,
  - $\Pr[\pi(s_1) = \pi(s_2)] \quad = \prod_{j=1,\ldots,k} \Pr[s_1[i_j]=s_2[i_j]]$
    $= (1 - d/w)^k$



Property of locality-sensitive hash function (II)

- Hence, s_1 and s_2 are similar if
  - $\pi(s_1) = \pi(s_2)$
- However, we may have false positive and false negative
  - False positive: s_1 and s_2 are dissimilar but $\pi(s_1) = \pi(s_2)$.
    - False positive can be distinguished from true positive by computing hamming distance between s_1 and s_2
  - False negative: s_1 and s_2 are similar but $\pi(s_1) \neq \pi(s_2)$.
    - We cannot detect false negative.
    - We can only reduce the number of false negative by repeating the test using different $\pi()$ functions

LSH-ALL-PAIRS

Algorithm:

1. Generate m random locality-sensitive hash functions $\pi_1(\ )$, $\pi_2(\ )$, …, $\pi_m(\ )$.
2. For every w-mer s in the database, compute $\pi_1(s)$, $\pi_2(s)$, …, $\pi_m(s)$.
3. For every pair of w-mers s and t such that $\pi_j(s) = \pi_j(t)$ for some j,
    - If hamming distance(s, t) < d, report (s, t)-pair.

Conclusion

- This lecture presents some database searching methods.
- In fact, there are many other methods. For examples:
  - CAFÉ, FLASH, RAMdb, FD, suffix tree, suffix array, compressed suffix array

More information

- Altschul, Madden, Schaffer, Zhang, Zhang, Miller and Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. NAR, 1997.
- Zhang, Schwartz, Wagner and Miller. A Greedy Algorithm for Aligning DNA sequences. Journal of Computational Biology, 2000.

- The list of database used by blast
  - ftp://ftp.ncbi.nlm.nih.gov/blast/db/