Basic Local Alignment Search Tool (BLAST)

Why use BLAST?

- BLAST **searches** for any entry in a selected **database** that is **similar** to your **query sequence** (protein or nucleotide)
- Identifying relatedness with BLAST is the **first step to identify possible function** of an unknown protein or gene

  - identifying orthologs and paralogs
  - discovering new genes or proteins
  - discovering variants of genes or proteins
  - investigating expressed sequence tags (ESTs)
  - exploring protein structure and function

- Searching for matches in a database with the **“needle” or “water” algorithm** is not feasible – it is **too slow**
- BLAST uses a **heuristic approach** – it is **not guaranteed** to be the **optimal answer**, but is close to it
- BLAST is available at https://blast.ncbi.nlm.nih.gov
- You can download and install BLAST+ on you personal computer: https://blast.ncbi.nlm.nih.gov/

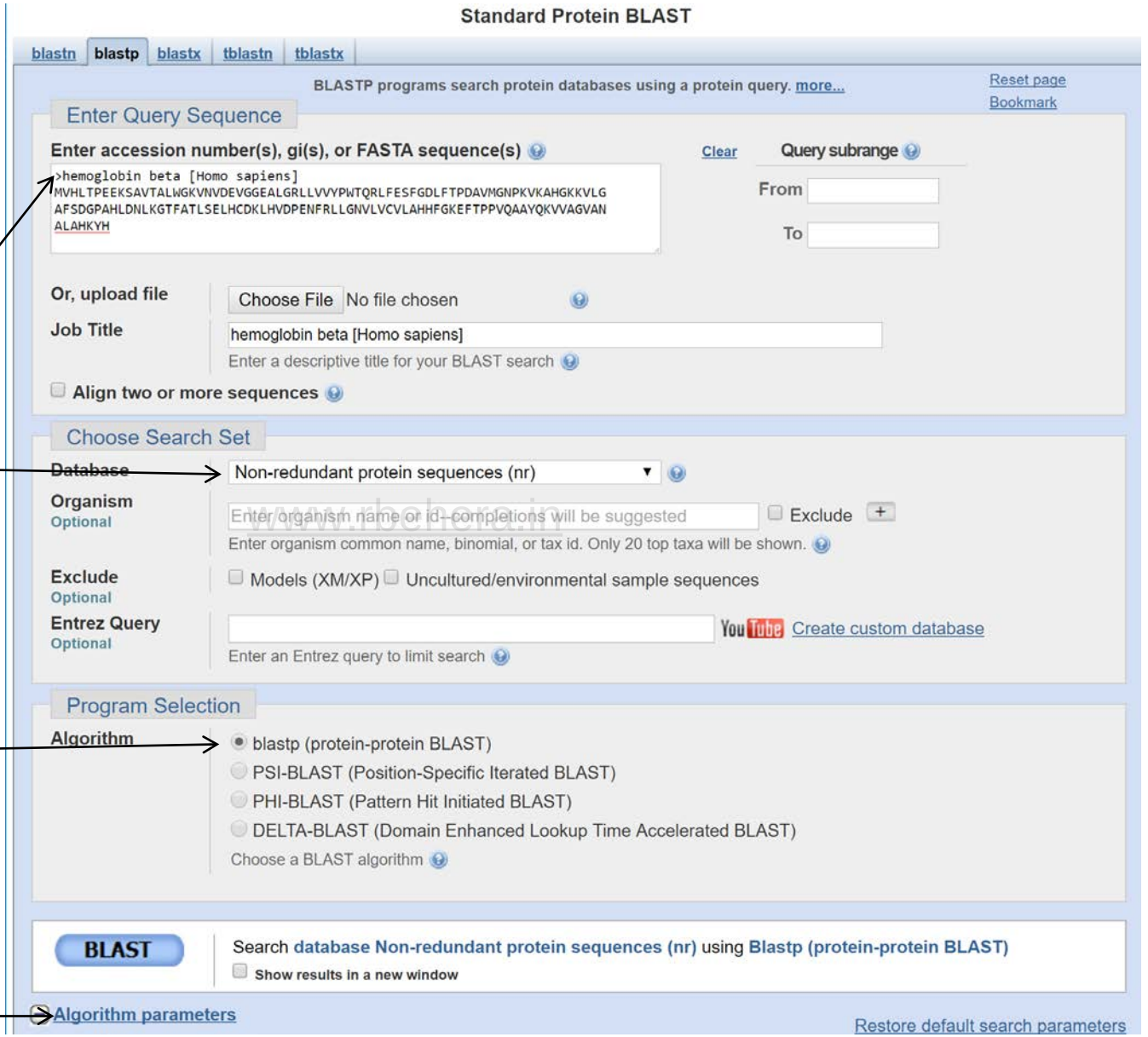
The BLAST webpage

Query sequence
FastA or accession number

Database

Algorithm

Parameters



BLAST protein databases

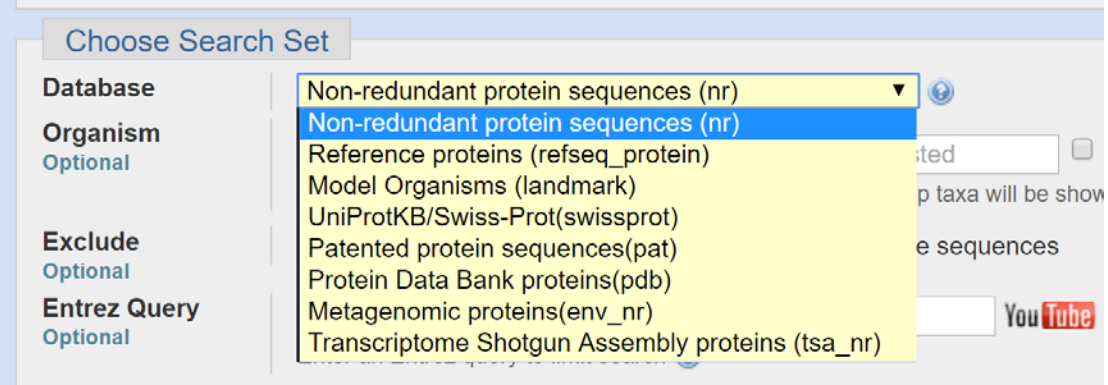


TABLE 4.1 Protein sequence databases that can be searched by BLAST searching at NCBI. PDB, Protein Data Bank. # indicates approximate number of sequences in database. Adapted from BLAST, NCBI, http://blast.ncbi.nlm.nih.gov/.

Database	Title	# sequences
nr	All nonredundant GenBank CDS translations + PDB + SwissProt + PIR + PRF excluding environmental samples from WGS projects	65 million
Reference proteins	NCBI protein reference sequences	50 million
UniProtKB/SwissProt	Nonredundant UniProtKB/SwissProt sequences	450,000
Patented protein sequences	Protein sequences derived from the Patent division of GenBank	1.3 million
Protein Data Bank	PDB protein database	77,000
Metagenomic proteins	Proteins from WGS metagenomic projects (env_nr)	6.5 million
Transcriptome	Transcriptome Shotgun Assembly (TSA) sequences	770,000

BLAST nucleotide databases

Database	Title	# sequences
Human Genomic + Transcript	Homo sapiens NCBI Annotation Release 104 RNAs; Homo sapiens all assemblies	55,000
Mouse Genomic + Transcript	Mus musculus NCBI Annotation RNAs; Mus musculus all assemblies	N/A
nr/nt	All GenBank+EMBL+DDBJ+PDB+RefSeq sequences, but excludes EST, STS, GSS, WGS, TSA, patent sequences as well as phase 0, 1, and 2 HTGS sequences	25 million
refseq_rna	NCBI transcript reference sequences	3.5 million
refseq_genomic	NCBI genomic reference sequences	2.7 million
NCBI Genomes	NCBI chromosome sequences	28,000
Expressed sequence tags (EST)	Database of GenBank+EMBL+DDBJ sequences from EST Divisions	75 million
Genomic survey sequences (gss)	Genome survey sequence, includes single-pass genomic data, exon-trapped sequences, and Alu PCR sequences	36 million
High-throughput genomic sequences (HTGS)	Unfinished high-throughput genomic sequences; sequences: phases 0,1 and 2	153,000
Patent sequences	Nucleotide sequences derived from the Patent division of GenBank	21 million
Protein Data Bank	PDB nucleotide database	8000
alu	Human *Alu* repeat elements	325
Sequence tagged sites (STS)	Database of GenBank+EMBL+DDBJ sequences from STS Divisions	1.3 million
Whole-genome shotgun (wgs)	Whole-genome-shotgun contigs	116 million
Transcriptome Shotgun Assembly (TSA)	Transcriptome shotgun assembly (TSA) sequences	15 million
16S ribosomal RNA sequences (Bacteria and Archaea)	16S ribosomal RNA sequences (bacteria and archaea)	7500

Different BLAST “flavours”

Program	Query	Number of database searches	Database
BLASTP	protein	1	protein

Use BLASTP to compare a protein query to a database of proteins.

BLASTN	DNA	1	DNA

Use BLASTN to compare both strands of a DNA query against a DNA database.

BLASTX	DNA	6	protein

BLASTX translates a DNA sequence into six protein sequences using all six possible reading frames, and then compares each of these proteins to a protein database.

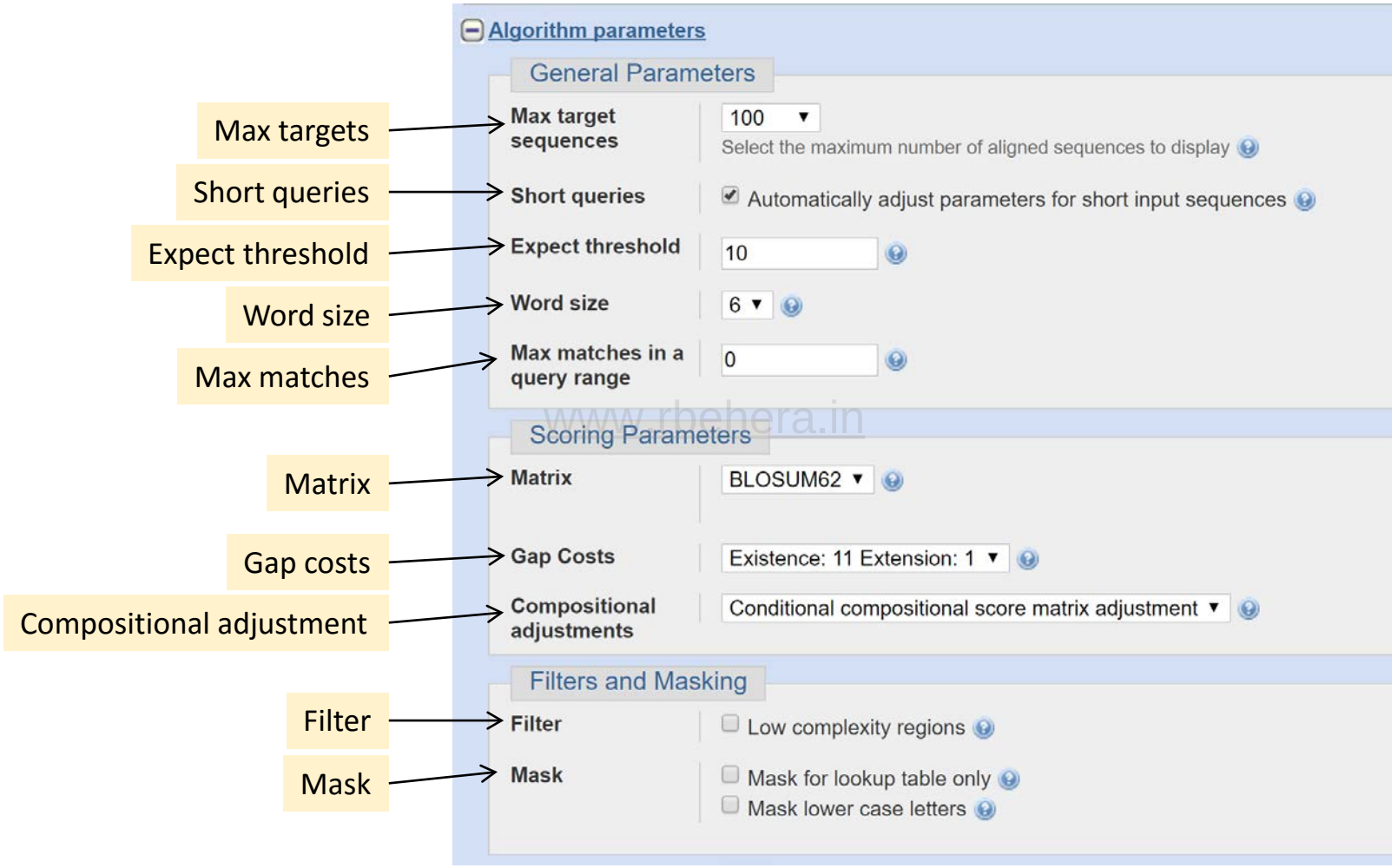
TBLASTN	protein	6	DNA

TBLASTN is used to translate every DNA sequence in a database into six potential proteins, and then to compare your protein query against each of those translated proteins.

TBLASTX	DNA	36	DNA

TBLASTX is the most computationally intensive BLAST algorithm. It translates DNA from both a query and a database into six potential proteins, then performs 36 protein-protein database searches.

Algorithm parameters

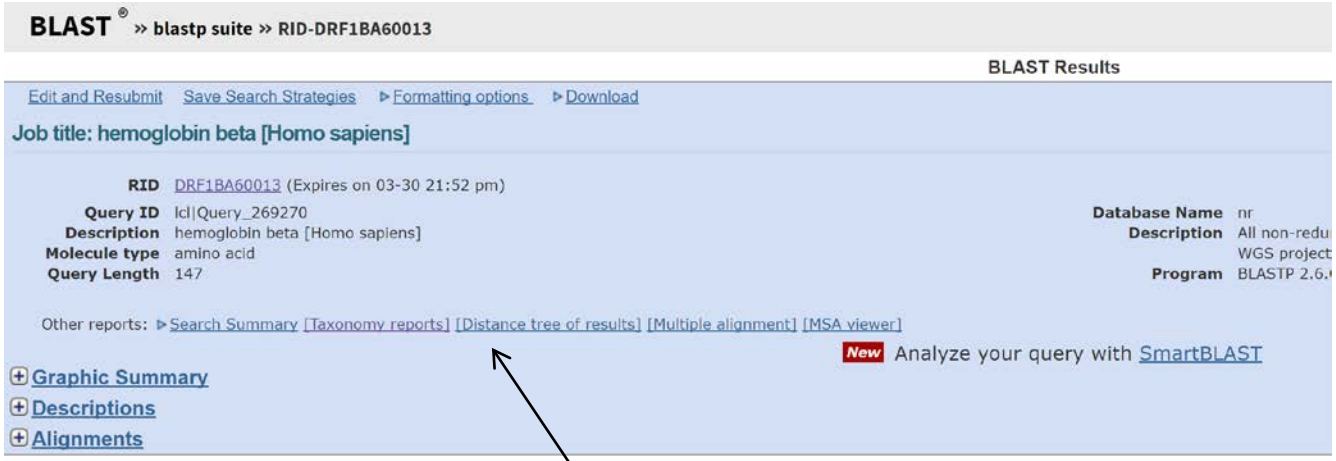


Algorithm parameters

Max targets – maximum number of sequence matches
Short queries – short sequences are more likely to be found, and word size can be adjusted
Expect threshold – the expected number of hits in a random model
Word size – the length of the seed that initiates the alignment
Max matches – adjust matches to different ranges in query sequence to avoid squelching
Matrix – choose scoring matrix

Gap cost – cost the create and extend a gap in the alignment
Compositional adjustment – the scoring matrix is adjusted to compensate for biases in the composition of the aligned sequences
Filter – mask regions of low complexity (simple repeats) that may cause spurious matches
Mask – mask the query when selecting seed sequences, or mask all lowercase letters in the FastA query sequence

BLAST output



Note, at the top of the result output page, links to display:

- Search summary
- Taxonomy report
- Distance Tree
- Multiple alignments
- Multiple Sequence Alignment (MSA) viewer

Search summary

- Data on the settings and result statistics of the search

Search Parameters	
Program	blastp
Word size	6
Expect value	10
Hitlist size	100
Gapcosts	11,1
Matrix	BLOSUM62
Filter string	F
Genetic Code	1
Window Size	40
Threshold	21
Composition-based stats	2

Database	
Posted date	Mar 24, 2017 4:20 PM
Number of letters	43,265,541,427
Number of sequences	118,106,513
Entrez query	none

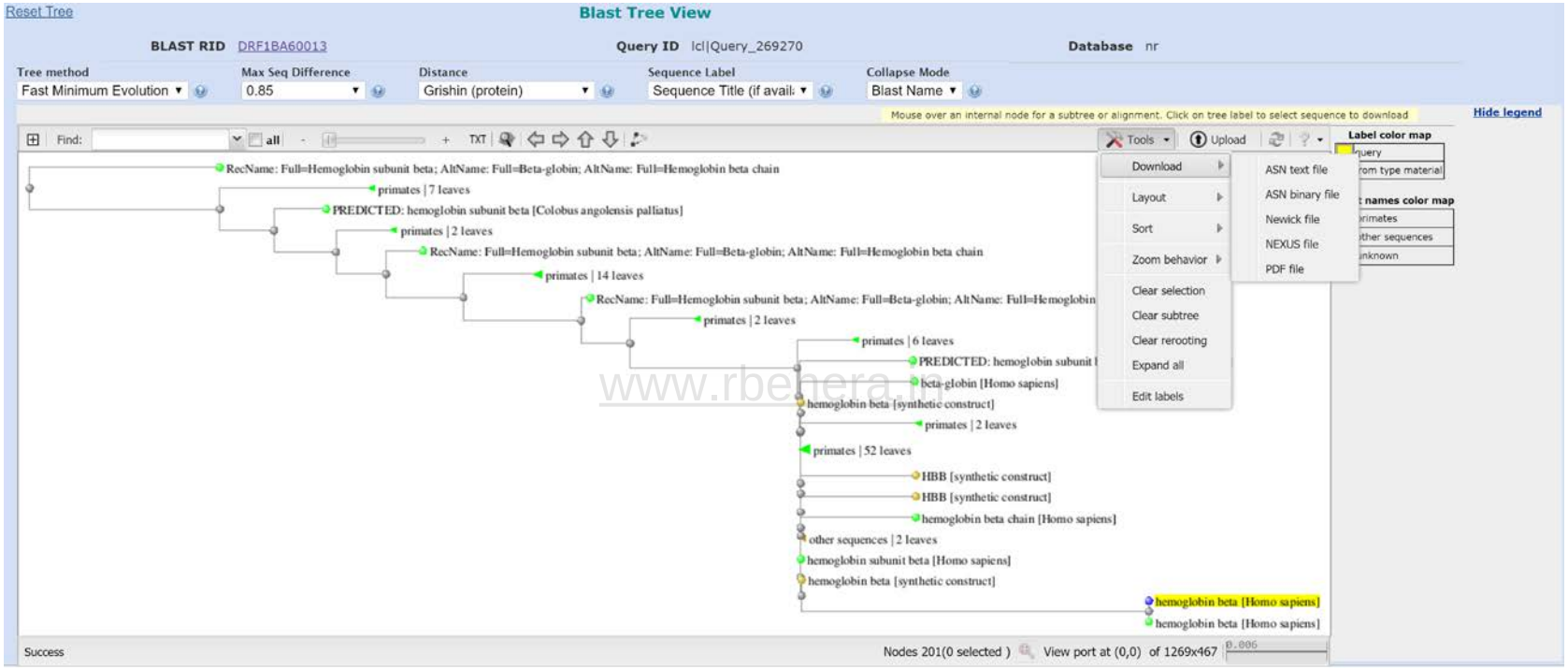
Karlin-Altschul statistics		
Lambda	0.320522	0.267
K	0.137501	0.041
H	0.427038	0.14
Alpha	0.7916	1.9
Alpha_v	4.96466	42.6028
Sigma		43.6362

Taxonomy report

- A tally on the number of phyla, families, species etc. that were matched

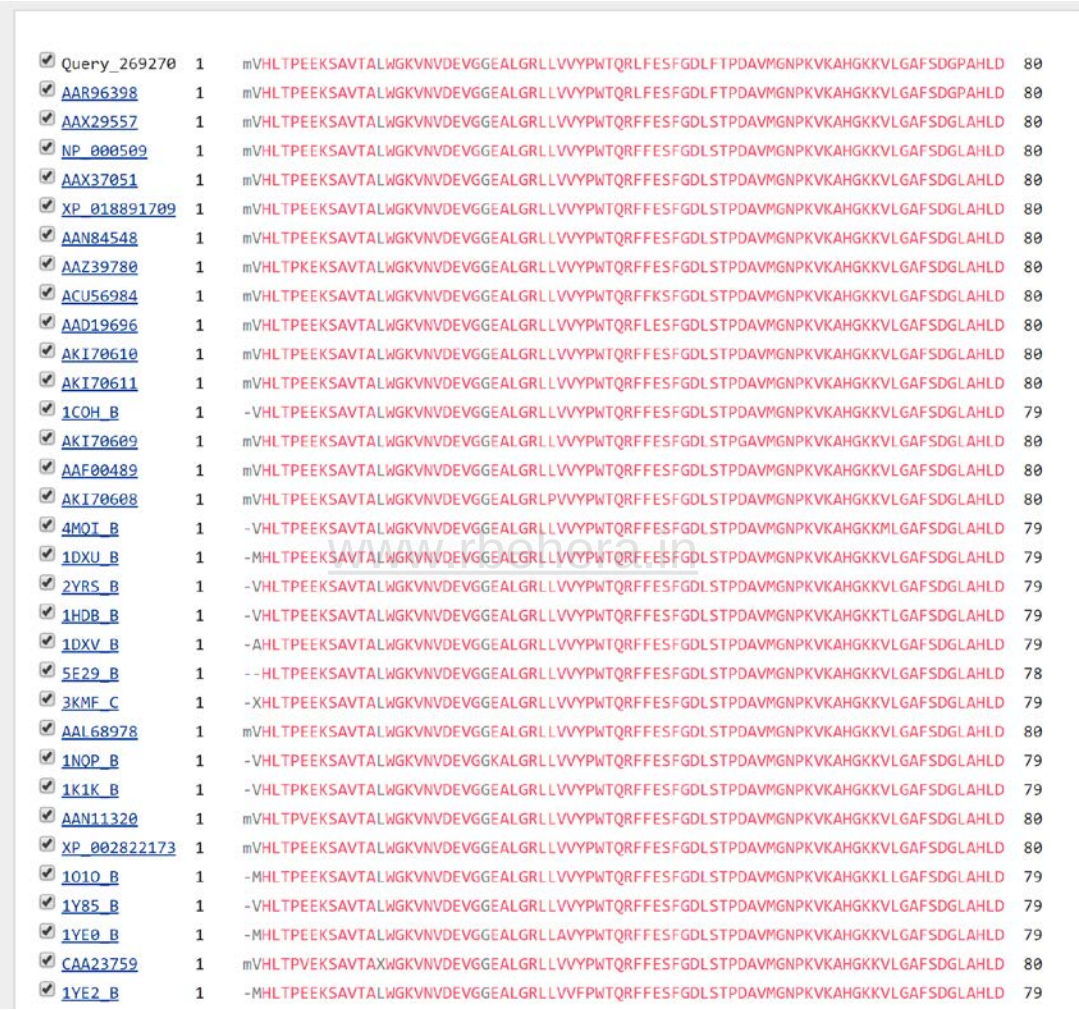
Organism	Blast Name	Score	Number of Hits	Description
root			669	
. Simiiformes	primates		655	
. . Catarrhini	primates		627	
. . . Hominoidea	primates		595	
. . . . Hominidae	primates		594	
. . . . . Homininae	primates		591	
. . . . . Homo sapiens	primates	301	584	Homo sapiens hits
. . . . . Pan troglodytes	primates	293	3	Pan troglodytes hits
. . . . . Pan paniscus	primates	293	2	Pan paniscus hits
. . . . . Gorilla gorilla gorilla	primates	291	2	Gorilla gorilla gorilla hits
. . . . . Pongo abelii	primates	288	2	Pongo abelii hits
. . . . . Pongo pygmaeus	primates	286	1	Pongo pygmaeus hits
. . . . Hylobates lar	primates	286	1	Hylobates lar hits
. . . Rhinopithecus bieti	primates	285	1	Rhinopithecus bieti hits
. . . Semnopithecus entellus	primates	284	1	Semnopithecus entellus hits
. . . Chlorocebus sabaeus	primates	284	1	Chlorocebus sabaeus hits
. . . Colobus angolensis palliatus	primates	283	1	Colobus angolensis palliatus hits
. . . Colobus polykomos	primates	283	1	Colobus polykomos hits
. . . Rhinopithecus roxellana	primates	283	1	Rhinopithecus roxellana hits
. . . Macaca fascicularis	primates	281	4	Macaca fascicularis hits
. . . Cercocebus atys	primates	281	2	Cercocebus atys hits
. . . Macaca nemestrina	primates	281	2	Macaca nemestrina hits
. . . Macaca fuscata fuscata	primates	281	1	Macaca fuscata fuscata hits
. . . Macaca speciosa	primates	281	1	Macaca speciosa hits
. . . Macaca mulatta	primates	281	4	Macaca mulatta hits
. . . Chlorocebus aethiops	primates	281	2	Chlorocebus aethiops hits
. . . Mandrillus leucophaeus	primates	280	2	Mandrillus leucophaeus hits
. . . Macaca arctoides	primates	278	1	Macaca arctoides hits
. . . Papio anubis	primates	278	3	Papio anubis hits
. . . Papio hamadryas	primates	278	1	Papio hamadryas hits
. . . Piliocolobus badius	primates	278	1	Piliocolobus badius hits
. . . Mandrillus sphinx	primates	278	2	Mandrillus sphinx hits

Distance Tree



- The phylogenetic tree of the multiple alignments are shown
- The data for the tree can also be downloaded in a selection of formats

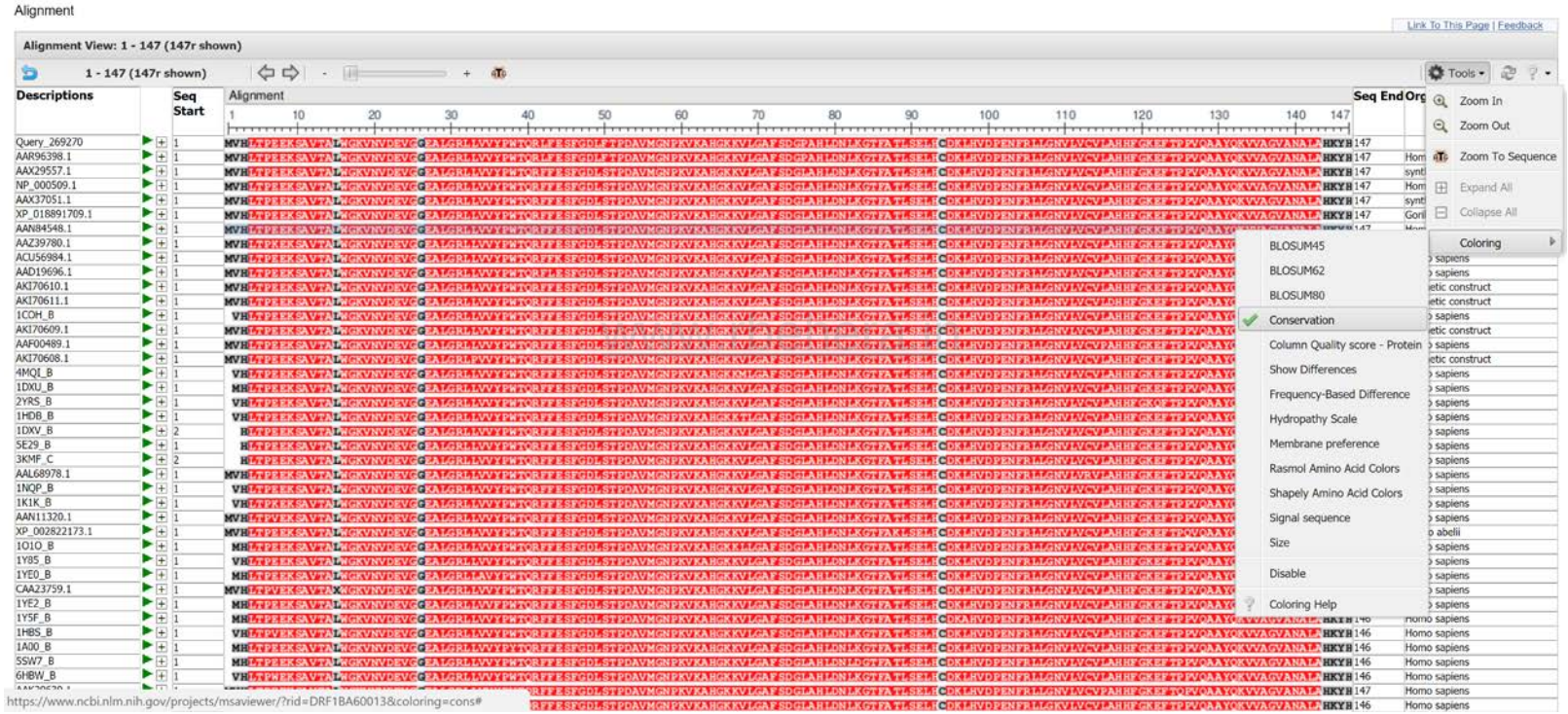
Multiple alignments



- This gives the multiple alignment of all the sequences returned for the query

MSA Viewer

- This allows viewing and some analysis of the multiply aligned sequences that were matched to the query



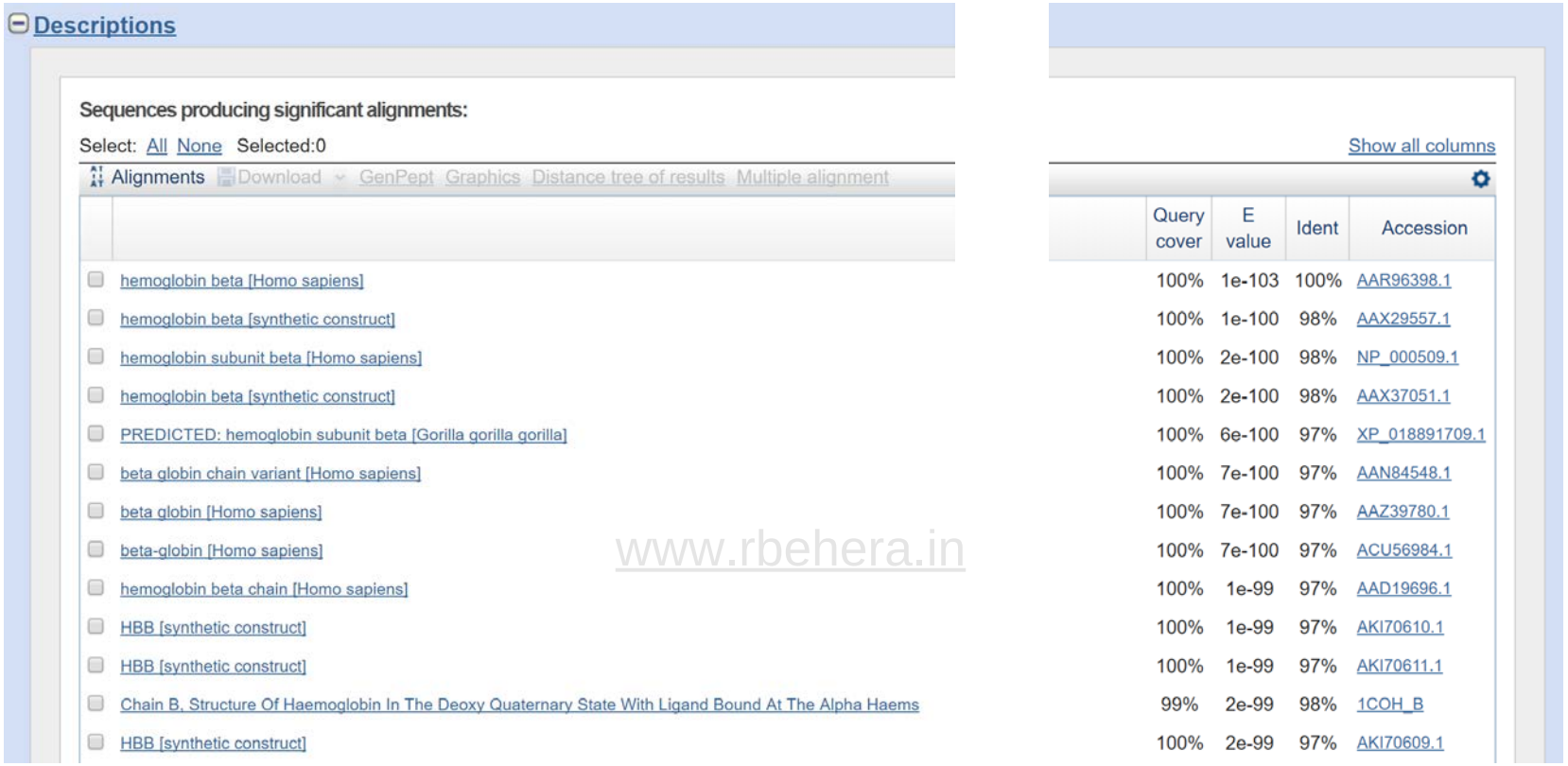
BLAST output

- The BLAST output contains several sections of information through which you can scroll

Identification of conserved domains and family classification (if any)

- Graphic representation of identified matches (mouse-over or click to see more)
- Colour indicates the score of the match

BLAST output



Descriptions

Sequences producing significant alignments:

Select: All None Selected:0

Show all columns

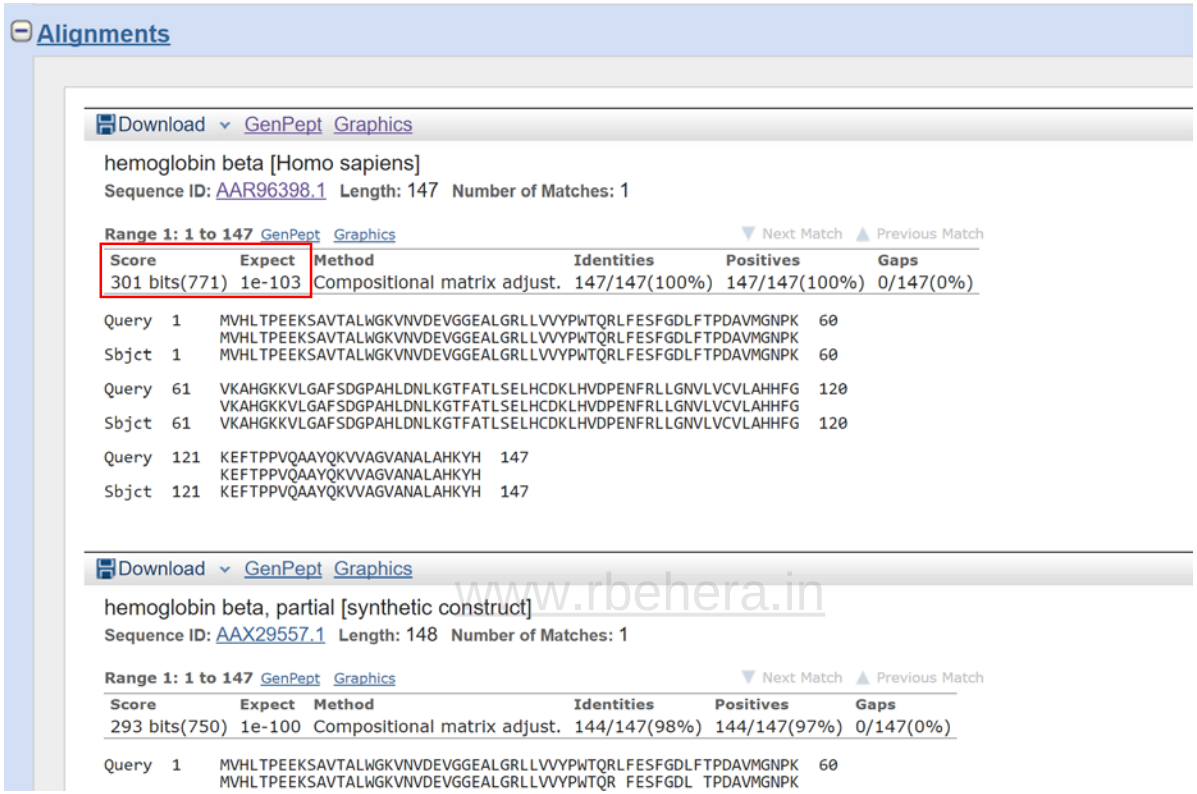
Alignments Download GenPept Graphics Distance tree of results Multiple alignment

	Query cover	E value	Ident	Accession
hemoglobin beta [Homo sapiens]	100%	1e-103	100%	AAR96398.1
hemoglobin beta [synthetic construct]	100%	1e-100	98%	AAX29557.1
hemoglobin subunit beta [Homo sapiens]	100%	2e-100	98%	NP_000509.1
hemoglobin beta [synthetic construct]	100%	2e-100	98%	AAX37051.1
PREDICTED: hemoglobin subunit beta [Gorilla gorilla gorilla]	100%	6e-100	97%	XP_018891709.1
beta globin chain variant [Homo sapiens]	100%	7e-100	97%	AAN84548.1
beta globin [Homo sapiens]	100%	7e-100	97%	AAZ39780.1
beta-globin [Homo sapiens]	100%	7e-100	97%	ACU56984.1
hemoglobin beta chain [Homo sapiens]	100%	1e-99	97%	AAD19696.1
HBB [synthetic construct]	100%	1e-99	97%	AKI70610.1
HBB [synthetic construct]	100%	1e-99	97%	AKI70611.1
Chain B, Structure Of Haemoglobin In The Deoxy Quaternary State With Ligand Bound At The Alpha Haems	99%	2e-99	98%	1COH_B
HBB [synthetic construct]	100%	2e-99	97%	AKI70609.1

The description section provides a listing of the matches showing

- Coverage of query (percentage of query aligned)
- The E-value of the match
- The percentage identity of the query-match
- The accession number of the match

BLAST output



Alignments

Download GenPept Graphics

hemoglobin beta [Homo sapiens]

Sequence ID: AAR96398.1 Length: 147 Number of Matches: 1

Range 1: 1 to 147 GenPept Graphics Next Match Previous Match

Score	Expect	Method	Identities	Positives	Gaps
301 bits(771)	1e-103	Compositional matrix adjust.	147/147(100%)	147/147(100%)	0/147(0%)

```
Query  1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRLFESFGDLFTPDAVMGNPK  60
            MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRLFESFGDLFTPDAVMGNPK
Sbjct  1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRLFESFGDLFTPDAVMGNPK  60

Query  61   VKAHGKKVLGAFSDGPAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG  120
            VKAHGKKVLGAFSDGPAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG
Sbjct  61   VKAHGKKVLGAFSDGPAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG  120

Query  121  KEFTPPVQAAYQKVVAGVANALAHKYH  147
            KEFTPPVQAAYQKVVAGVANALAHKYH
Sbjct  121  KEFTPPVQAAYQKVVAGVANALAHKYH  147
```

Download GenPept Graphics

hemoglobin beta, partial [synthetic construct]

Sequence ID: AAX29557.1 Length: 148 Number of Matches: 1

Range 1: 1 to 147 GenPept Graphics Next Match Previous Match

Score	Expect	Method	Identities	Positives	Gaps
293 bits(750)	1e-100	Compositional matrix adjust.	144/147(98%)	144/147(97%)	0/147(0%)

```
Query  1    MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQR FESFGDL TPDAVMGNPK  60
            MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQR FESFGDL TPDAVMGNPK
```


- The alignment section shows the alignments of the query-matches with
  - Score
  - E-value
  - Identities
- The central sequence shows identical residues, conserved residues (“+” character) and mismatches (a gap)

The BLASTP algorithm

Phase 1: Setup: compile a list of words (w=3) above threshold T

- Query sequence: human beta globin NP_000509.1 (includes ...VTALWGKVNVD...).
  This sequence is read; low complexity or other filtering is applied; a "lookup" table is built.

- Words derived from query sequence (HBB): VTA TAL ALW LWG WGK GKV KVN VNV NVD

- Generate a list of words matching query (both above and below T). Consider LWG in the query and the scores (derived from a BLOSUM62 matrix) for various words.

- Generate similar lists of words spanning the query (e.g. words for WGW, GWG, WGK...).

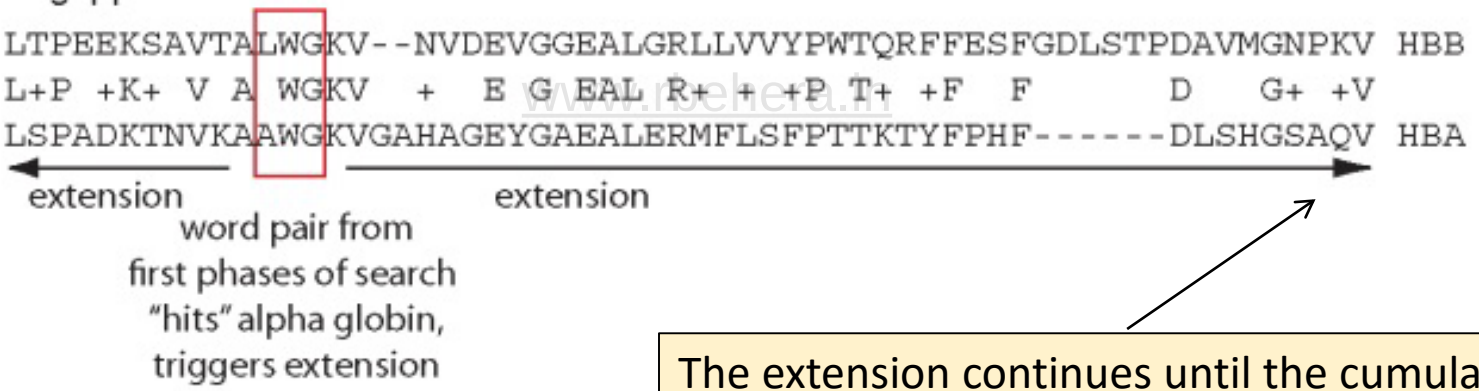
threshold →

examples of words >= threshold 12	LWG	4+11+6=21
	IWG	2+11+6=19
	MWG	2+11+6=19
	VWG	1+11+6=18
	FWG	0+11+6=17
	AWG	0+11+6=17
	LWS	4+11+0=15
	LWN	4+11+0=15
	LWA	4+11+0=15
	LYG	4+ 2+6=12
examples of words below threshold	LFG	4+ 1+6=11
	FWS	0+11+0=11
	AWS	-1+11+0=10
	CWS	-1+11+0=10
	IWC	2+11-3=10

The BLASTP algorithm

Phase 2: Scanning and extensions

- Select all the words above threshold T (LWG, IWG, MWG, VWG, FWG, AWG, LWS, LWN, LWA, LYG)
- Scan the database for entries ("hits") that match the compiled list
- Create a hash table index with the locations of all the hits for each word
- Perform gap free extensions
- Perform gapped extensions



The extension continues until the cumulative value drop below a "drop-off" value

Hash tables

- It is a table with a key that points to a storage location when a “hashing function” (H) is applied to the key
- Example of a H(K,n):
  - Storage location = K mod n, where K=key and n=size of storage
  - H(K,n) = mod(K,n)
  - A (ASCII=65) mod 9 = 2

	Storage locations								
Key	0	1	2	3	4	5	6	7	8
A			A						
C					C				
E							E		
G									G
I		I							

- If you have the key, you can quickly find the storage location, and recover its content

The BLASTP algorithm

Phase 3: Traceback
- Calculate locations of insertions, deletions, and matches (for alignments saved in Phase 2)
- Apply composition-based statistics (for BLASTP, TBLASTN)
- Generate gapped alignment

- For **BLASTN**, the word size is typically 7, 11, or 15 (EXACT match). Changing word size is like changing threshold of proteins. w=15 gives fewer matches and is faster than w=11 or w=7.

How BLAST calculates the significance of a match

$$E = Kmne^{-\lambda S}$$

S = the raw score

E = the expect value
the number of high-
scoring segment pairs (HSPs) expected
to occur with a score of at least S

m, n = the length of two sequences

λ, K = Karlin-Altschul statistics

Some properties of the BLAST equation

$$E = Kmne^{-\lambda S}$$

- The value of **E decreases** exponentially with **increasing S** (higher S values correspond to better alignments). Very **high scores** correspond to very **low E values**
- The E value for aligning a pair of random sequences must be negative! Otherwise, long random alignments would acquire great scores
- Parameter ***K*** describes the **search space** (database).
- For **E=1**, **one match** with a similar score is expected to **occur by chance**. For a very much larger or smaller database, you would expect E to vary accordingly

Bit scores

- There are two kinds of scores: **raw scores** (calculated from a substitution matrix) and **bit scores** (normalized scores)

- **Bit scores** are comparable between different searches because they are **normalized** to account for the use of different scoring matrices and different database sizes

- S' = bit score = $(\lambda S - \ln K) / \ln 2$

- The E value corresponding to a given bit score is:
- $E = mn2^{-S'}$

- Bit scores allow you **to compare results between different database searches**, even using different scoring matrices.

Specialised BLAST "flavours"

- When searching the "nr" dataset with human β-globin, the search does not return myoglobin (first 1000 hits)
- We saw that myoglobin was structurally almost identical to β-globin and clearly homologous
- BLASTp **is not sensitive enough**
- Thus studying evolutionary relations of a protein may **miss distant homologs**
- There are a number of adaptations to the classic BLAST algorithm to compensate for this.

PSI-BLAST

Position-specific iterated BLAST. Uses a position-specific scoring matrix (PSSM)

PHI-BLAST

Pattern-hit initiated BLAST

Delta-BLAST

Domain enhanced lookup time accelerated BLAST

PSI-BLAST

- Starts off with a BLASTP search, and then makes a **frequency matrix** of the number of occurrences of each residue at each position of the aligned sequences
- This is also known as a **position specific scoring matrix (PSSM)**

```
PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLYGRFT----IKSDVW
PDAGLYGRFTG--LKSDVW
PEAGVFGKFS----RSEVW
```

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

- What about the amino acid composition of the sequences?

Normalize the PSSM

- **Normalize** the matrix to the **frequency of occurrence** of each residue in the **population**
- Normalization **corrects** for the chance that we will **select** a specific amino acid **randomly** from the database
- You will typically use the **frequency observed** in the **database** that you are searching
- For instance, P was observed 5 times out of 5 at position 1
- Thus, the **raw** frequency of P is 5/5 = 1 (5 occurrences in 5 sequences)
- However, the frequency of P in the database that we are searching is **1/20** (assuming that all amino acids are equally represented)
- The frequency of P in the database is the **probability** that we will select a P in a **random selection** from the database
- Thus the **normalized frequency** for P at position 1 is:

- $\frac{\frac{5}{5}}{\frac{1}{20}} = \frac{1}{0.05} = 20$

- Thus, in the example above, P occurs **20× more frequently** than would be **expected** from a **random distribution**

Is “0” for some amino acids in a PSSM reasonable?

- A flipped coin can either be “heads” or “tails”
- Each toss gives an independent chance of $\frac{1}{2}$ that it will be "heads"
- There is a **real** (but **extremely small**) **chance** that you can flip 1000 “heads” in a row, **never observing a “tails”**
- The tally would then be “heads” = 1000, “tails” = 0
- Although you never observed a “tails” in your experiment, you know that it is **possible** (prior experience)
- Thus, to use your observation “tails” = 0 to indicate that “tails” is never observed, is incorrect
- To **adjust the chance** of an occurrence, **based on previous knowledge**, is an established statistical principle known as **pseudo-count**, or the **rule of succession**.
- This typically involves **adding 1** to the number of “heads”, and **adding 2 to the number of observations** (you have previously observed a “heads” and a “tails”)

Normalize matrix incorporating pseudo-counts

The normalized occurrence of P at position 1, normalized for the frequency of P in the database and corrected with a pseudo-count, is

$$P_{\text{normalized}} = \frac{Pobserved+1}{Number\ of\ sequences+20} \Big/ \frac{Pdatabase}{Database\ size}$$

The pseudo-count is 1

20×pseudo-count = 20
(there are 20 amino acids)

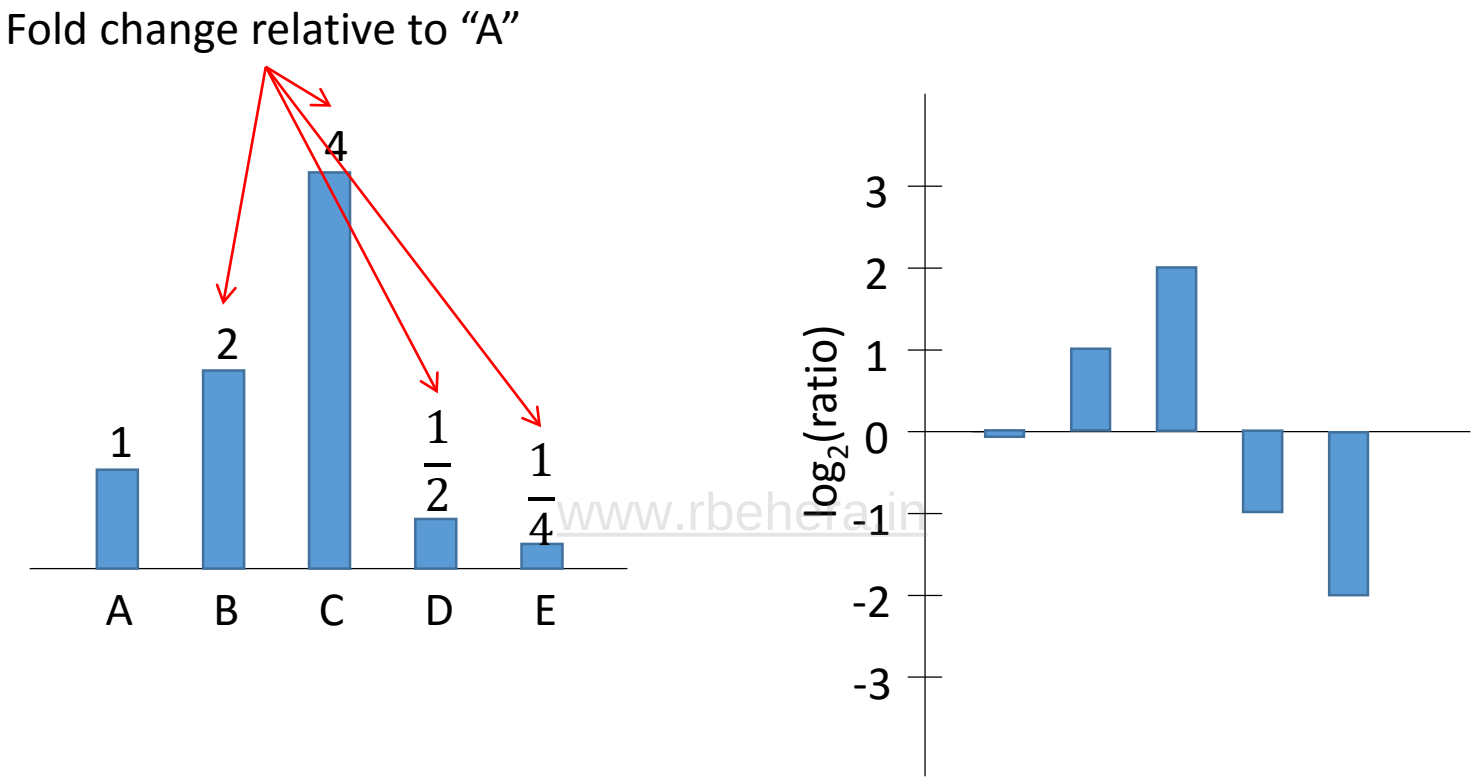
Assume each amino acid is equally represented ($\frac{1}{20}$)

Thus, for p = $\frac{5+1}{5+20} \Big/ 0.05 = 0.24 / 0.05 = 4.8$

Where we have a 0: p = $\frac{0+1}{5+20} \Big/ 0.05 = 0.04 / 0.05 = 0.8$

Slightly below our break-even level of 1

The value of using $\log_2$ space



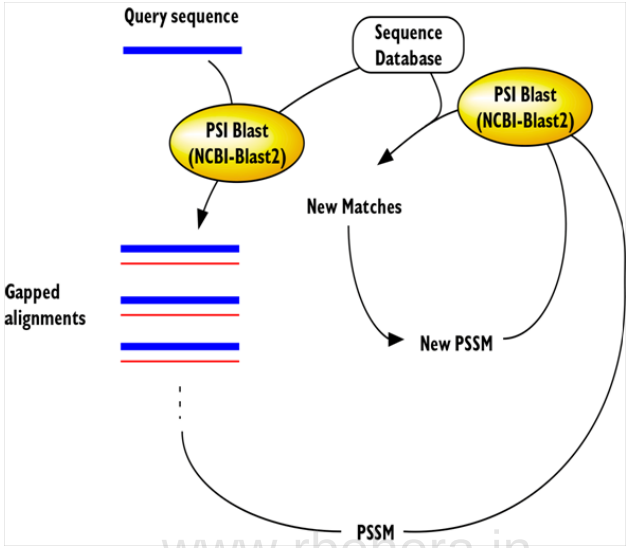
- $\log_2$ space gives **symmetrical distributions** for **identical fold changes**
- It is widely used in **matrices**, microarrays, RNA-seq, proteomics etc.

Sequence logos

```
PEAALYGRFT---IKSDVW
PEAALYGRFT---IKSDVW
PESLYGRFT----IKSDVW
PDAGLYGRFTG--LKSDVW
PEAGVFGKFS----RSEVW
```

- A sequence logo is a very informative way to display a multiple alignment
- The height of each letter in the stack is proportional to the observed frequency of the letter at that position
- The combined height of a stack corresponds to the “information content” (in bits) of the position
- You can made protein or DNA logos: weblogo.berkeley.edu

PSI-BLAST (Position-specific iterated BLAST)

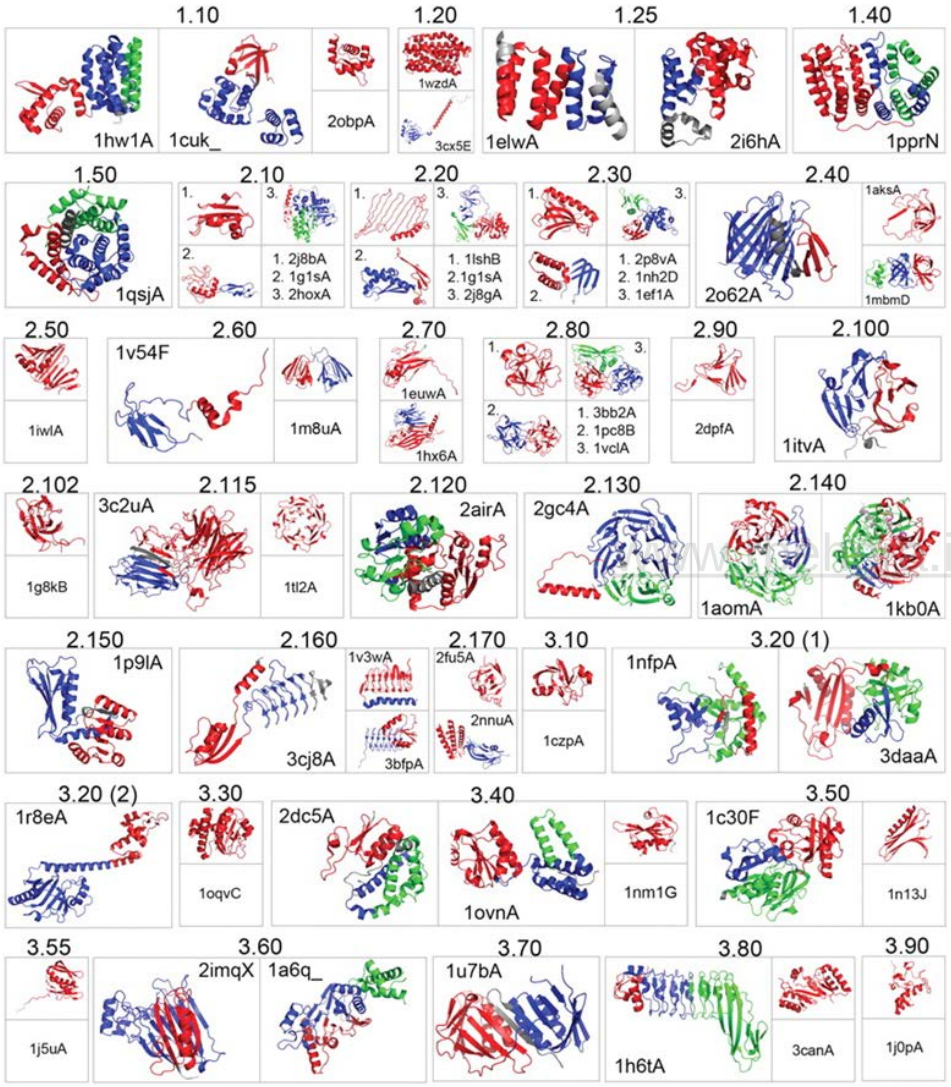


- A query is searched against the selected database with **BLASTP**
- The returned alignment is used to **construct a PSSM**
- The PSSM is used to **search the database** again
- The **PSSM is adjusted** to reflect the new returned matches
- This **iteration** (repetition) is typically repeated 5 times
- The **E-values** are estimated
- **More sensitive** than BLAST
- Will identify evolutionary **distant members of family**
- Iteration slows search -- **slower than BLAST**

PHI-BLAST (Pattern hit initiated BLAST)

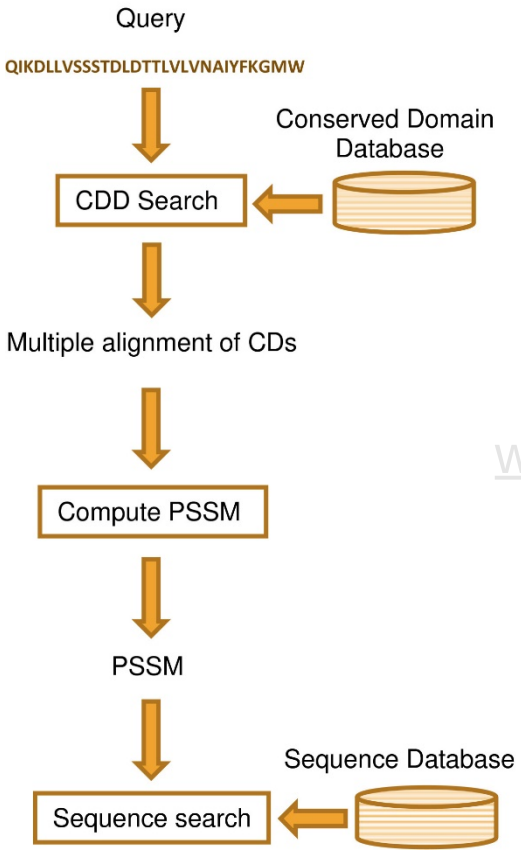
- **Searches** with a **pattern** against selected database
- PHI-BLAST uses the **Prosite pattern convention**:
  - Any valid residue one-character symbol ACDEFGHIKLMNPQRSTVWY (for DNA: GATC)
  - [ ] means any one of the characters in brackets e.g., [LFYT] means one occurrence of L or F or Y or T
  - - means nothing (this is a spacer for human readability)
  - x(5) means 5 positions in which any residue is allowed
  - x(2,4) means 2 to 4 positions where any residue is allowed
  - [LIVMF]-G-E-x-[GAS]-x(5,11)-R-[STAQ]-A-x-[LIVMA]-x-[STACV]
- Use when you know protein family has a **signature pattern**: **active site, structural domain**, etc.
- Better chance of eliminating false positives

Derive sequence patterns from protein domains



- We have seen that β-globin and myoglobin, although only 20% identical, fold into virtually identical structures
- It therefore seems reasonable to identify all known protein members with a specific domain structure, align the sequences of the domain, and use that alignment to identify possible unknown members
- DELTA-BLAST does this

DELTA-BLAST (Domain enhanced lookup time accelerated BLAST)



- DELTA-BLAST searches a database of pre-aligned **conserved domains**
- It uses the matched multiple alignment to compute a **PSSM**
- The PSSM it then used to **search** the selected database

Using HMMER

- HMMs have a **formal probabilistic basis** (unlike PSSMs)
- Use **probability theory** to guide how all the **scoring parameters** should be set
- **Consistent theory** for setting position-specific **gap and insertion scores**
- Allows **libraries** of hundreds of **profile HMMs** and apply them on a very large scale to whole **genome analysis**
- You can download Linux, Mac OSX and Windows binaries of HMMER and use it on your computer (http://hmmer.org/)
- HMMER is composed of **many programs** to build profiles, align to profiles, search profiles against databases etc.
- build a profile hmm from aligned sequences
- `> hmmbuild globins4.hmm tutorial/globins4.sto`

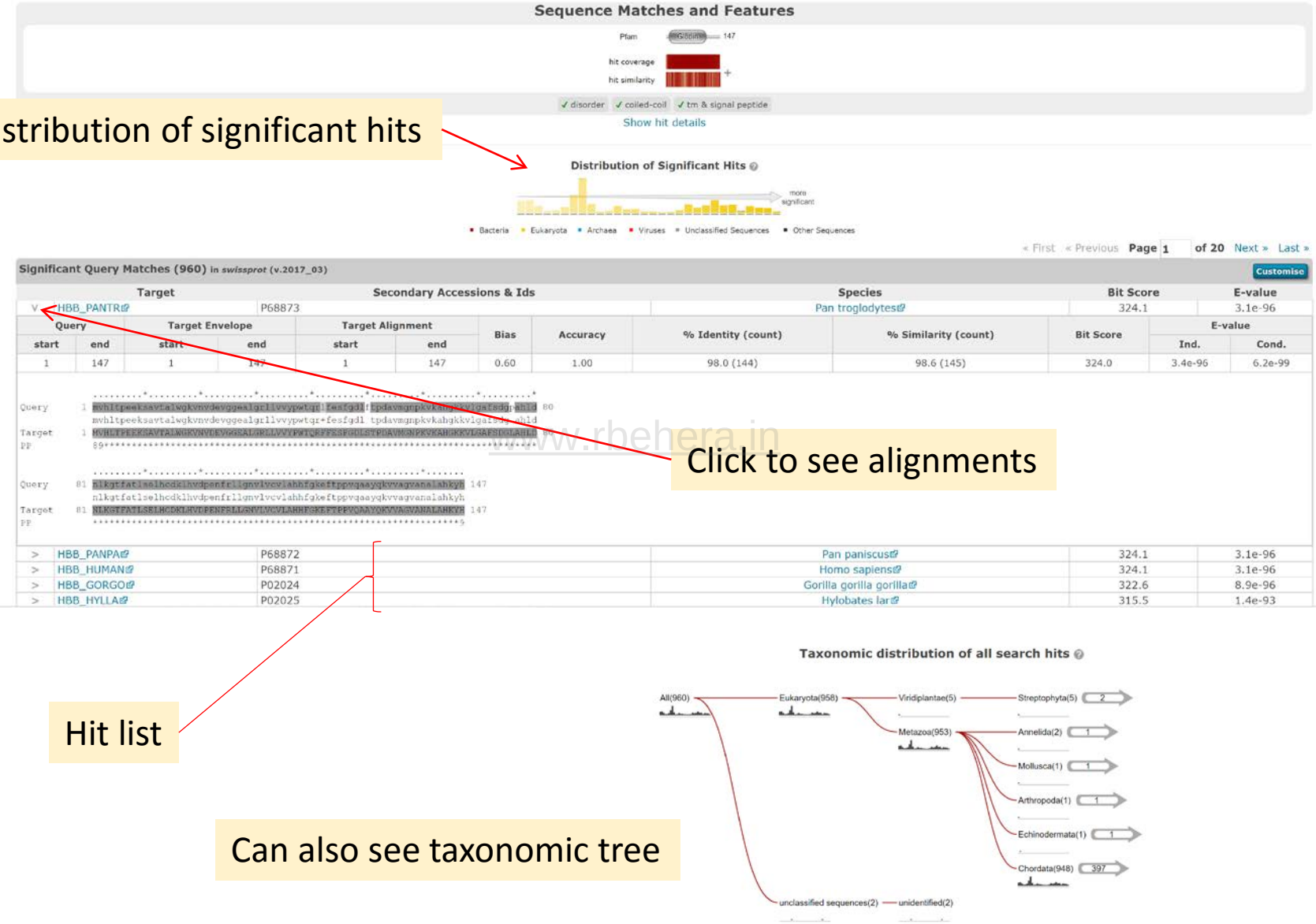
- Use the profile hmm to scan a fasta protein database
- `> hmmsearch globins4.hmm uniprot sprot.fasta > globins4.out`

```
--- full sequence --- --- best 1 domain --- -#domE-value
score bias E-value score bias exp N Sequence Description
------- ------ ----- ------- ------ ----- ---- -- -------- -----------
6.5e-65 222.7 3.2 7.2e-65 222.6 3.2 1.0 1 sp|P02185|MYG_PHYMC Myoglobin OS=Physeter macrocephalus GN
3.3e-63 217.2 0.1 3.7e-63 217.0 0.1 1.0 1 sp|P02024|HBB_GORGO Hemoglobin subunit beta OS=Gorilla gor
4.9e-63 216.6 0.0 5.4e-63 216.5 0.0 1.0 1 sp|P68871|HBB_HUMAN Hemoglobin subunit beta OS=Homo sapien
4.9e-63 216.6 0.0 5.4e-63 216.5 0.0 1.0 1 sp|P68872|HBB_PANPA Hemoglobin subunit beta OS=Pan paniscu
4.9e-63 216.6 0.0 5.4e-63 216.5 0.0 1.0 1 sp|P68873|HBB_PANTR Hemoglobin subunit beta OS=Pan troglod
7e-63 216.1 3.0 7.7e-63 216.0 3.0 1.0 1 sp|P02177|MYG_ESCGI Myoglobin OS=Eschrichtius gibbosus GN=
```

HMMer as a web service

- You can also access HMMER software as a **web service** (http://www.ebi.ac.uk/Tools/hmmer/)
- **Phmmer** - protein sequence against protein sequence database
  This is similar to BLASTp, using the input query and a BLOSUM62 matrix to derive a HMM profile, which is searched against a selected database
- **HMMscan** - protein sequence against profile-HMM database
- **HMMsearch** - protein alignment/profile-HMM against protein sequence database
- **Jackhmmer** - iterative search against protein sequence database, similar to PSI-BLAST

Phmmer output



Distribution of significant hits

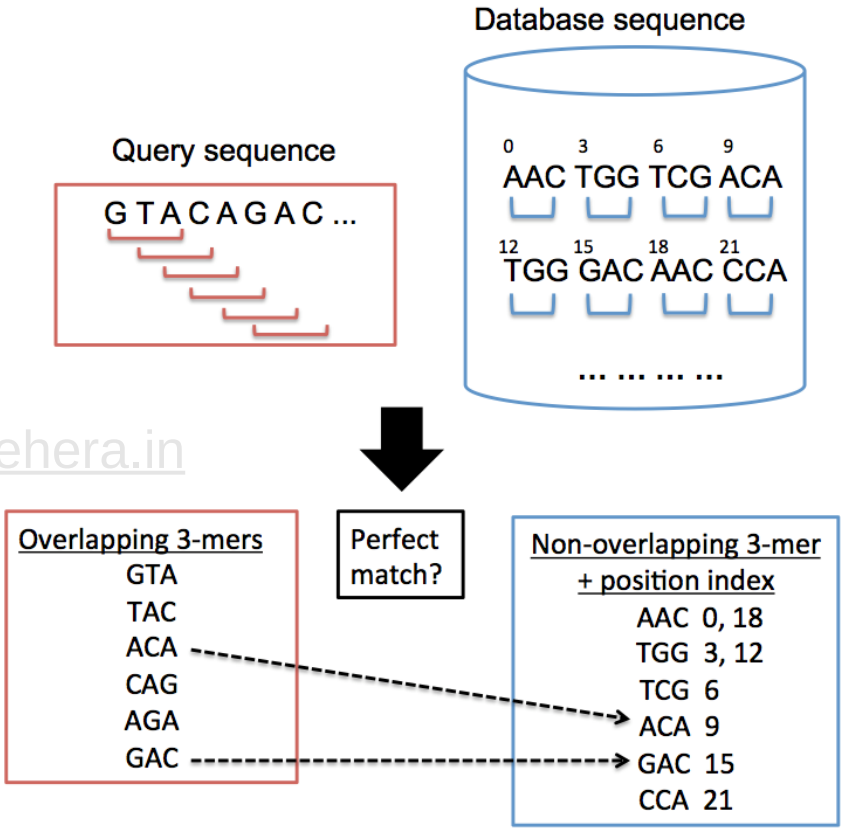
Click to see alignments

Hit list

Can also see taxonomic tree

Blast-like alignment tool (BLAT)

- BLAT **pre-indexes** (constructs a **hash table**) of the non-overlapping k-words of the **entire database**
- It keeps the entire **hash table** in **memory**
- It then searched for **1-character offset k-words** from the query sequence in the **hash table**
- **Two** nearby **hits** are **extended** and the sequence fused
- BLAT is **very efficient** at searching **genome-sized sequences**
- BLAT is **less sensitive** than BLAST



commons.wikimedia.org/w/index.php?curid=29016560