LSM2241
Searching Sequence Databases with BLAST

Greg Tucker-Kellogg

2 Sept 2015

Outline

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

Dynamic programming for optimal sequence alignment

- Dynamic programming approaches
  - use a scoring scheme
  - identify the optimal pairwise alignment(s)
  - without comparing all possible alignments
- *Optimal* pairwise alignments
  - maximise a similarity measure
  - minimize a measure of evolutionary distance

The limits of dynamic programming

- Aligning more than two sequences together is *Multiple Sequence Alignment* (**MSA**)
- Dynamic Programming becomes intractable for MSA
- *Progressive* MSA methods
  1. Compare sequences in pairs
  2. Establish a “guide tree”
  3. Use the guide tree to *order* the addition of individual sequences to a growing alignment.

The risks of progressive alignment

- Early alignments (those chosen as *closest branches* in the guide tree) have a lot of influence in later stages of alignment
  - Aligning a third sequence to an alignment of the first two uses substitution scores drawn from *both* sequences in the alignment
  - Because new sequences are added in progressive alignment, early substitutions affect the scores of all later substitution possibilities
- "Once a gap, always a gap"

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

BLAST = Basic Local Alignment Search Tool

- A tool introduce in 1990 to identify similar sequences by database search. (This paper has been cited > 40,000 times!)
- Starting with one sequence (the **query**), identify sequences in a database (the **search**) that are similar to it
- This is for *local* alignments, like Smith-Waterman, except not guaranteed optimal
- Based on an approximate measure of local similarity

See (Altschul, Gish, et al. 1990) for the actual paper

What is BLAST used for?

Many, many applications!

- Discovering new genes or proteins, or sequence features
- Discovering gene variants
- Analyzing new sequence data using existing sequence knowledge
- Designing experiments
- Analyzing newly sequenced genomes
- Investigating expressed sequence tags (ESTs)

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

The steps of a basic BLAST search

1. Select a query sequence
2. Select a BLAST program from the family of BLAST programs
3. Select a database to search
4. Make any other needed parameter settings
5. Run it (click “BLAST”)

This can be done on the command line, but is usually done in a browser

1. Pick a query sequence

Our example sequence: *Homo sapiens* k-Ras, a GTPase important in cellular signaling

GTPase KRas isoform a precursor [Homo sapiens]

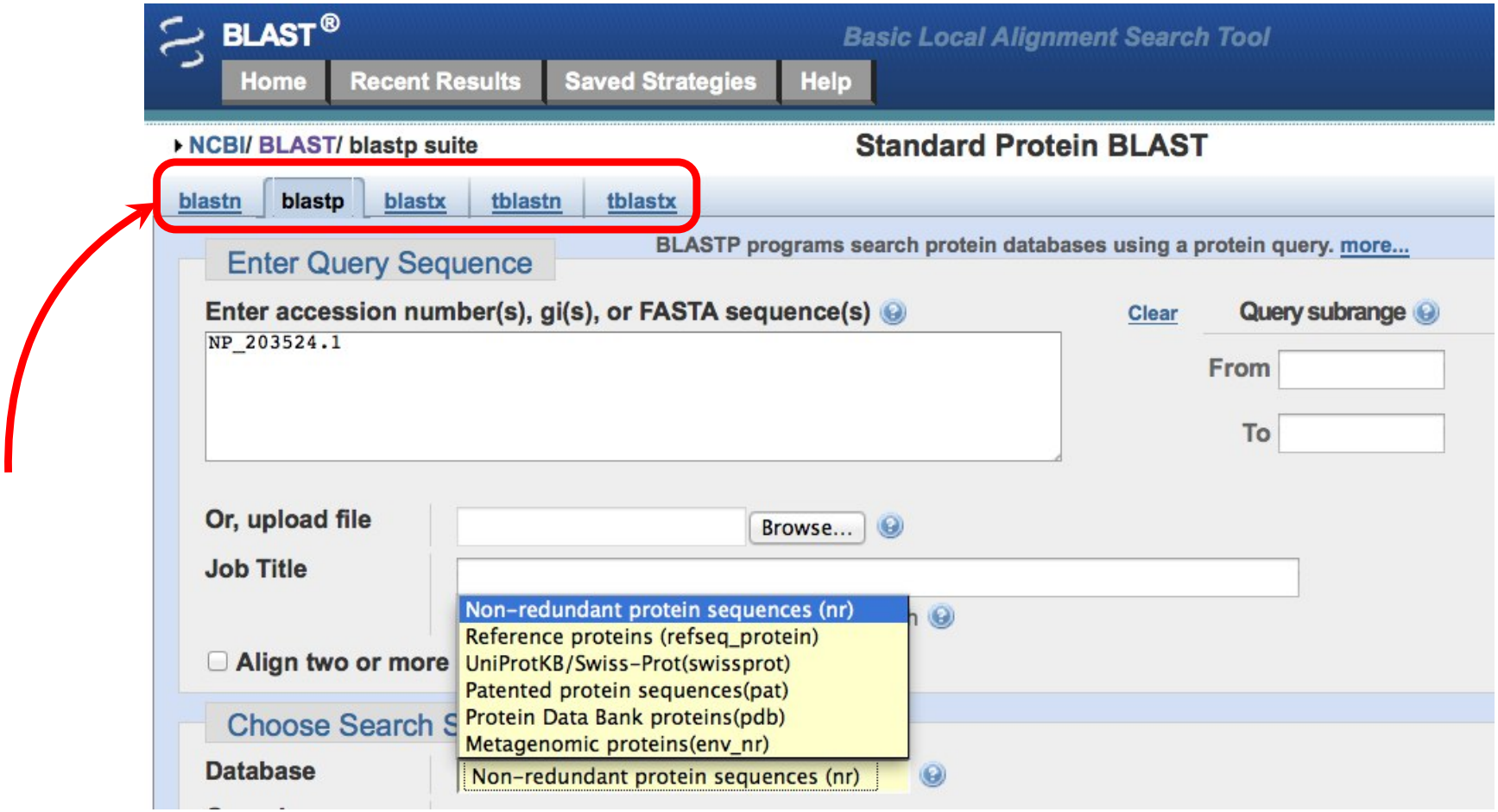
NCBI Reference Sequence: NP_203524.1

GenPept Graphics

```
>gi|15718763|ref|NP_203524.1| GTPase KRas isoform a precursor [Homo sapiens]
MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAGQEEYSAMRDQ
YMRTGEGFLCVFAINNTKSFEDIHHYREQIKRVKDSEDVPMVLVGNKCDLPSRTVDTKQAQDLARSYGIP
FIETSAKTRQRVEDAFYTLVREIRQYRLKKISKEEKTPGCVKIKKCIIM
```

We'll search this against a database of proteins from *Schizosaccharomyces pombe*, or fission yeast, a useful model organism.

2. Decide on the BLAST program



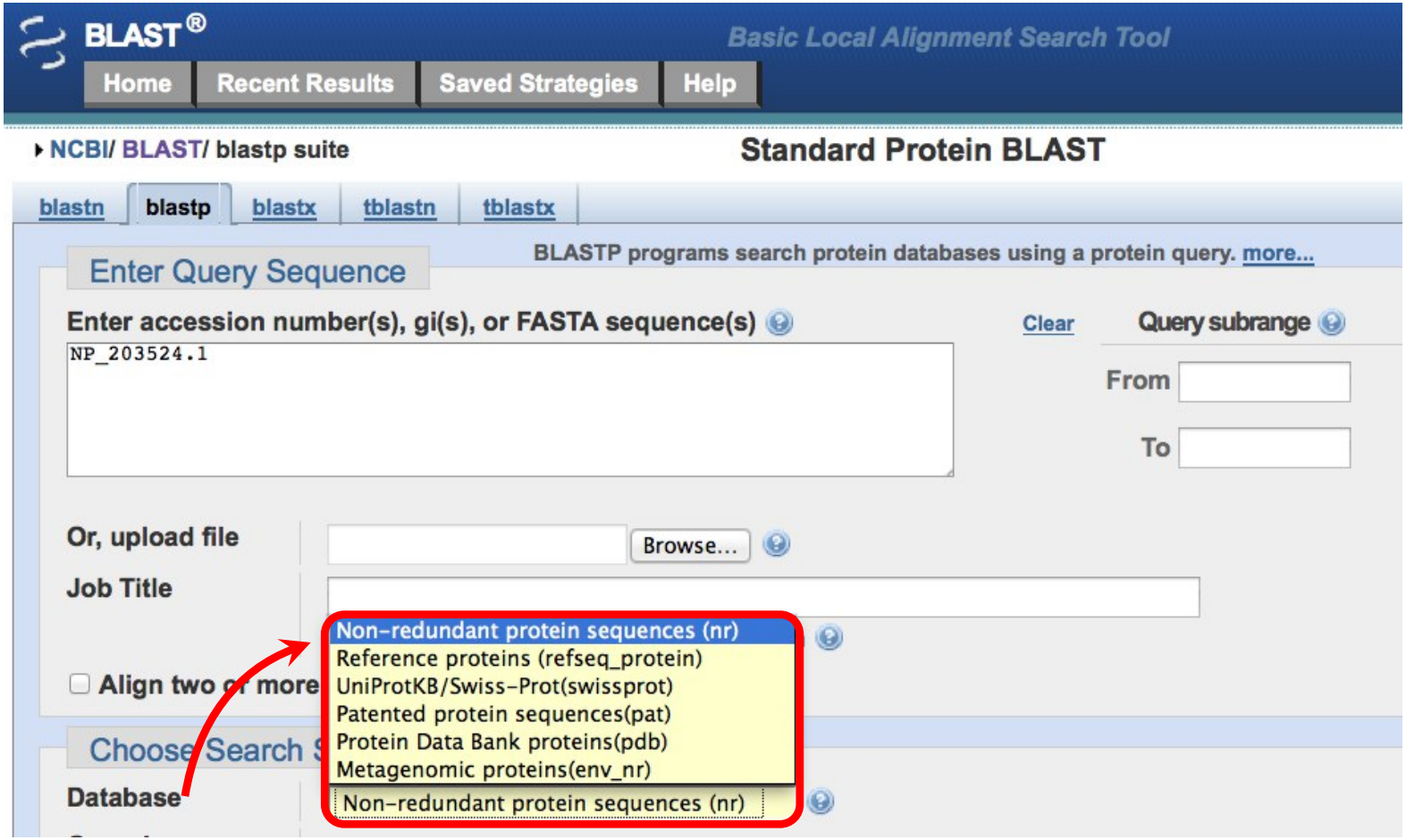
Standard BLAST programs

Program	input	search	database
`blastn`	nucleotide	1	nucleotide
`blastp`	protein	1	protein
`blastx`	nucleotide	6	protein
`tblastn`	protein	6	nucleotide
`tblastx`	nucleotide	36	nucleotide

Six-frame translation for protein-based searches of DNA

	input (5'-3')
DNA	`tctcgttttaggactcttcttccatattagtgt`
translation, frame 1	`_S R F R T L L P Y - C`
frame 2	`__L V L G L F F H I S`
frame 3	`___S F - D S S S I L V`
	reverse complement
DNA	`acactaatatggaagaagagtcctaaaacgaga`
translation, frame 1	`_T L I W K K S P K T R`
frame 2	`__H - Y G R R V L K R`
frame 3	`___T N M E E E S - N E`

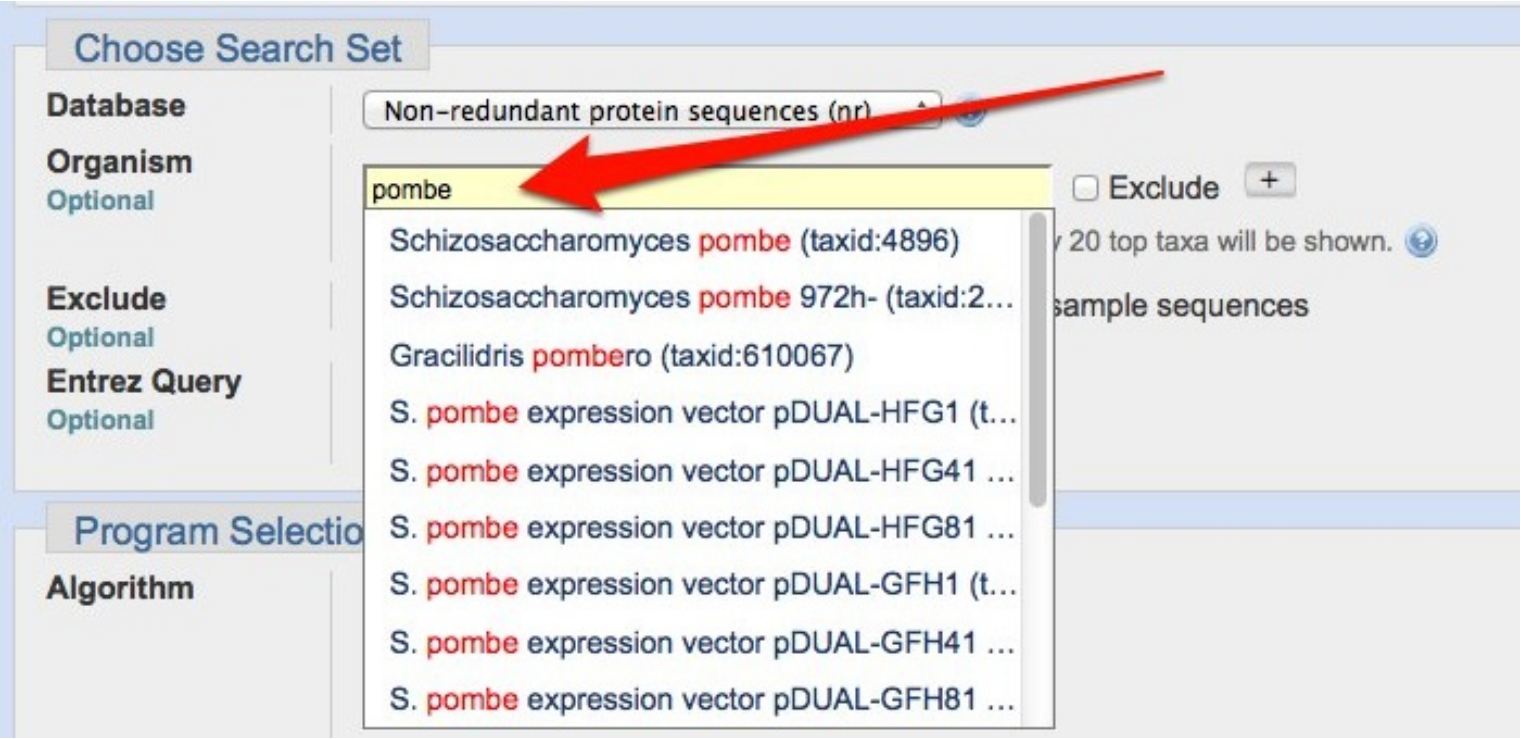
3. Pick the search database



4. Decide on any additional options

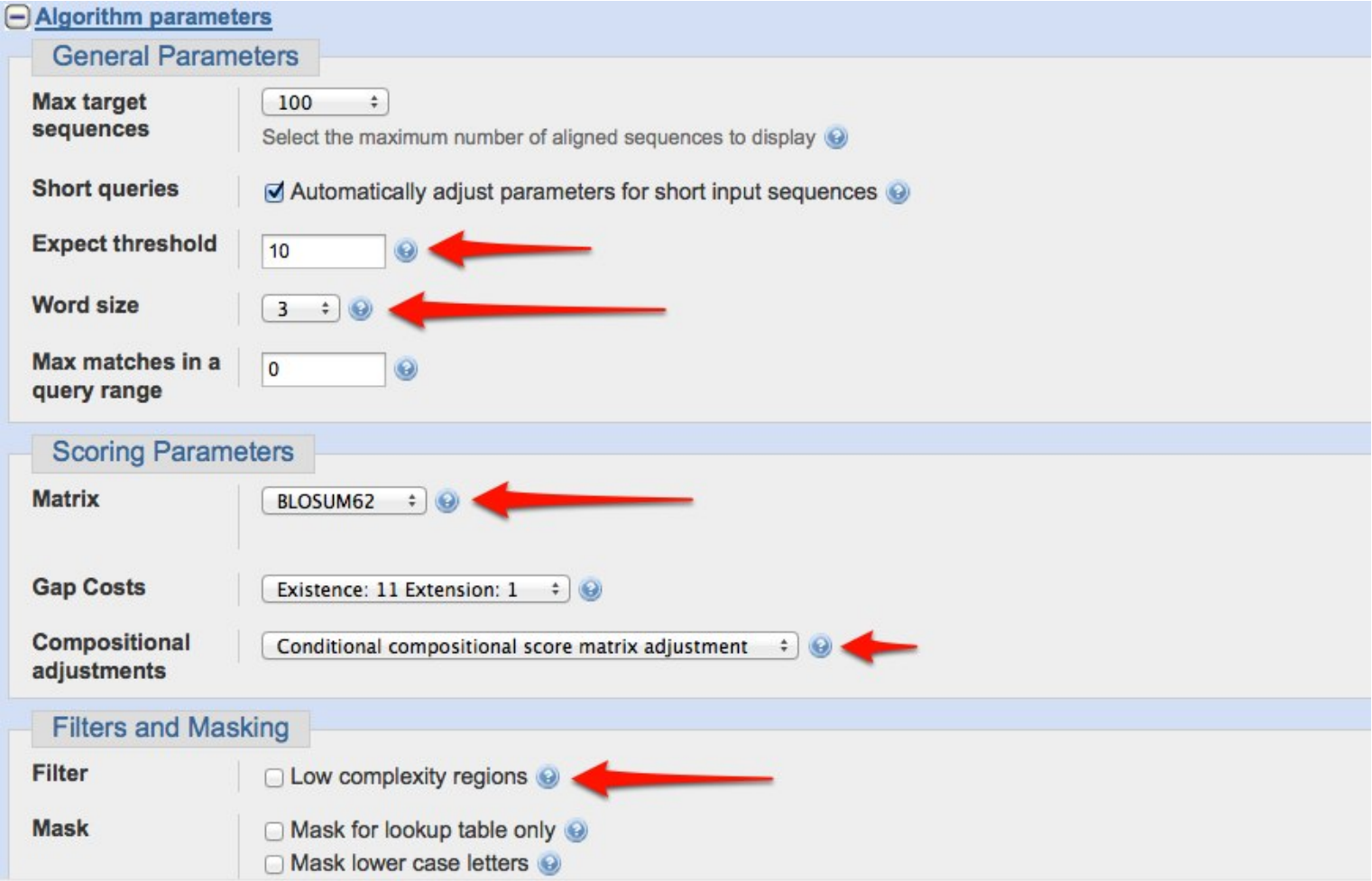
- There are a number of other options to BLAST
- Let's restrict our search to *S. pombe*
- Other options include
  - word size
  - masking and filtering
  - different scoring matrices
  - an "expect" threshold

Species-specific search



You can enter part of the species name, or a common name, and options will pop up

Even more parameters



Inspect the results

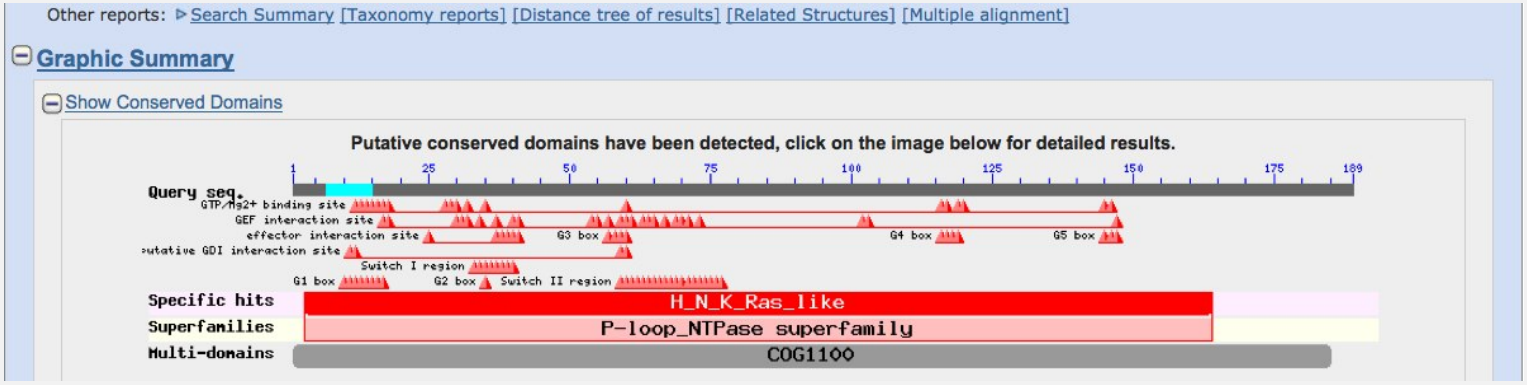
remember our search?

- Human k-ras protein
- searched against *S. pombe* protein sequence database
- BLOSUM62 matrix
- word size 3
- Conditional compositional score matrix adjustment

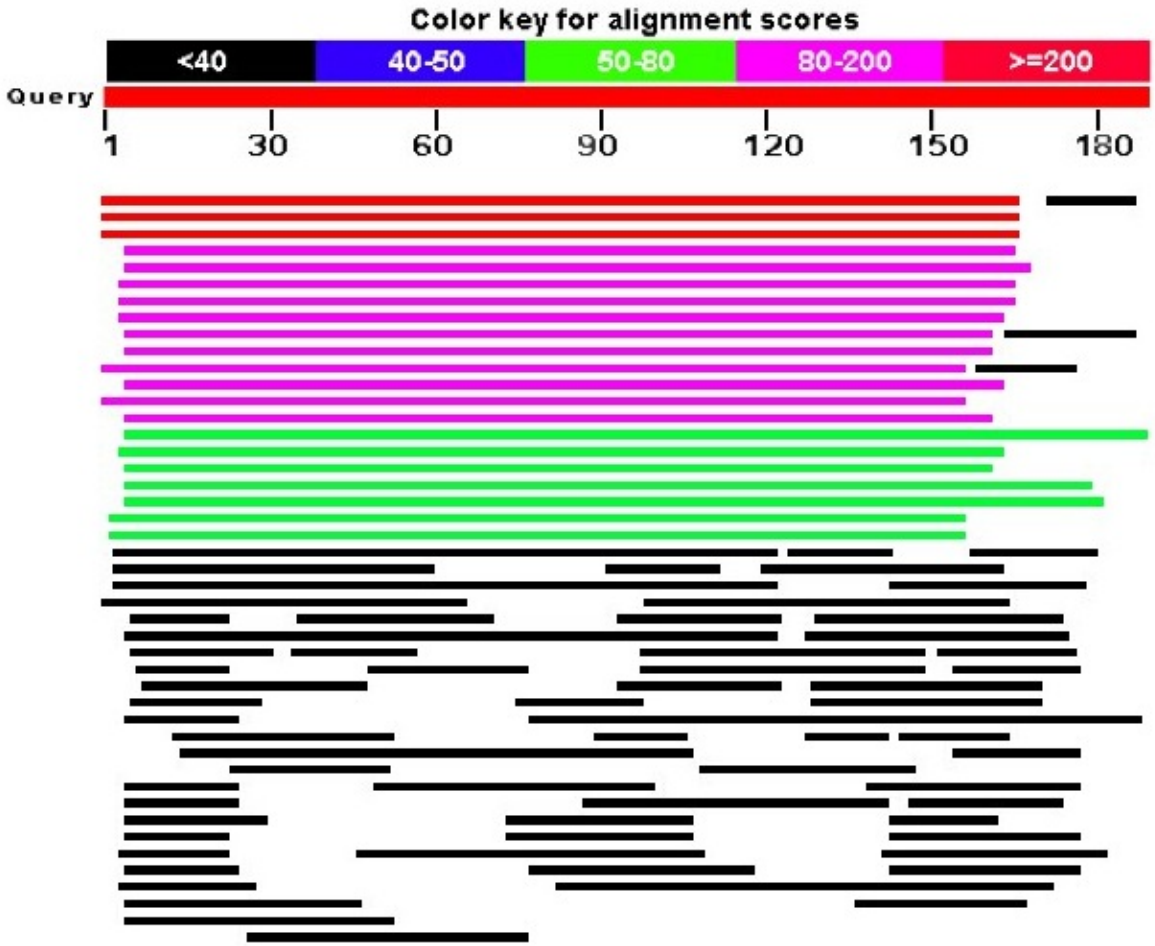
The results page

The very top

A graphical summary



The results page (2)



mouse over highlights and clickable details

The results page (3)

A tabular summary of results

Descriptions

Legend for links to other resources: U UniGene E GEO G Gene S Structure M Map Viewer PubChem BioAssay

bit scores

Expect values

Sequences producing significant alignments:

Accession	Description	Max score	Total score	Query coverage	E value	Max ident	Links
CAA27399.1	put. ras protein [Schizosaccharomyces pombe]	204	204	87%	1e-69	66%	
NP_593579.1	GTPase Ras1 [Schizosaccharomyces pombe 972h-] >sp|P08647.2|RA	203	203	87%	5e-69	66%	U G M
CAA26191.1	SPRAS-protein [Schizosaccharomyces pombe]	201	201	87%	5e-68	65%	
NP_595194.1	Rheb GTPase Rhb1 [Schizosaccharomyces pombe 972h-] >sp|O94363	119	119	85%	2e-36	36%	U G M
NP_594580.1	GTPase Ypt2 [Schizosaccharomyces pombe 972h-] >sp|P17609.1|YPT	110	110	86%	5e-33	36%	U G M
NP_596205.1	GTPase Ypt1 [Schizosaccharomyces pombe 972h-] >sp|P11620.2|YPT	104	104	85%	2e-30	35%	U G M
CAA33192.1	YPT1-related protein [Schizosaccharomyces pombe]	104	104	85%	2e-30	35%	
NP_593667.1	GTPase Ypt3 [Schizosaccharomyces pombe 972h-] >sp|P17610.1|YPT	102	102	84%	1e-29	38%	U G M
NP_596307.1	GTPase Ypt7 [Schizosaccharomyces pombe 972h-] >sp|O94655.1|YPT	94.0	94.0	83%	2e-26	32%	U G M
NP_593524.1	GTPase Ypt71 [Schizosaccharomyces pombe 972h-] >sp|Q9HDY0.1|Y	88.2	88.2	83%	4e-24	34%	U G M

Note that each aligned sequence may produce multiple alignments

The results page (4): the alignments

```
>emb|CAA27399.1|  put. ras protein [Schizosaccharomyces pombe]
Length=214

 Score =  204 bits (520),  Expect = 1e-69, Method: Compositional matrix adjust.
 Identities = 111/167 (66%), Positives = 132/167 (79%), Gaps = 1/167 (1%)

Query  1    MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSYRKQVVIDGETCLLDILDTAG  60
            M EYKLVVVG GGVGKSALTIQLIQ+HFVDEYDPTIEDSYRK+  IDGE  LLD+LDTAG
Sbjct  1    MREYKLVVVGDGGVGKSALTIQLIQSHFVDEYDPTIEDSYRKKCEIDGEGALLDVLDTAG  60

Query  61   QEEYSAMRDQYMRTGEGFLCVFAINNTKSFEDIHHYREQIKRVKDSEDVPMVLVGNKCDL  120
            QEEYSAMR+QYMRTGEGFL V+ I +  SF++I  + +QI RVKD +  P+VLV NKCDL
Sbjct  61   QEEYSAMREQYMRTGEGFLLVYNITSRSSFDEISTFYQQILRVKDKDTFPVVLVANKCDL  120

Query  121  PS-RTVDTKQAQDLARSYGIPFIETSAKTRQRVEDAFYTLVREIRQY  166
             + R V   + + LA+S    ++ETSAK R  VE+AFY+LVR IR+Y
Sbjct  121  EAERVVSRAEGEQLAKSMHCLYVETSAKLRLNVEEAFYSLVRTIRRY  167
```

Each alignment gives bit scores, *E* values, and other statistics for interpretation

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

The basic concepts

- In the BLAST model, alignments with similarity will have some short segments (words) with very high similarity
- These “hits” can be found quickly
- The hits are extended to segments that are able to exceed a threshold

How (original) BLAST works

- The original BLAST algorithm (1990) has three phases
  1. Compile a list of high scoring words above a threshold value
  2. Scan the database for all such matches (these are “hits”)
  3. Extend the hit in both directions, stopping when the accumulated score decreases X below maximum (Default X 20 for `blastn`, otherwise 7)

Step 1, compile a list of words and scores from the query sequence

Let's take a stretch of our query sequence

```
61 QEEYSAMRDQYMRTGEGFLCVFAINNTKSFEDIHHYR 98
```

If we are looking at the word **GFL**, the table below shows the words that would initiate hits with a threshold score of 11. Scores are taken from **BLOSUM62**

word	**pos 1**	**pos 2**	**pos 3**	**total score**
GFL	6	6	4	16
GFM	6	6	2	14
GFI	6	6	2	14
GFV	6	6	1	13
GYL	6	3	4	13
GFF	6	6	0	12

Step 2, find the "hits" (words with score > *T*)

- With 20 amino acids, there are only $20^3 = 8000$ three letter words.
- Positions of every word in a sequence database can be kept and looked up in constant time
- The default threshold for `blastp` is 11, but it can be changed
- This step can be very fast

Step 3, extend the hits in both directions

```
Query  1    MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSY  40
            + EYKLVVVG GGVGKSALTIQLIQ+HFVDEYDPTIEDS
Sbjct  6    LREYKLVVVGDGGVGKSALTIQLIQSHFVDEYDPTIEDSY  45
```

Hit!

- For each neighborhood hit (or pair of hits) start extending one residue at a time and keep score
- Stop when the score drops below a cutoff X below maximum

How about for `blastn`?

- For DNA databases, only 4 letters in the alphabet, so words can be larger
- For DNA searches, a word must match the query *exactly*, not just with a sufficiently high score
- So for DNA, increasing the word size makes the search *faster* but finds fewer matches
- Variants like **MEGABLAST** have very large word sizes and allow fast searching of whole genomes

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

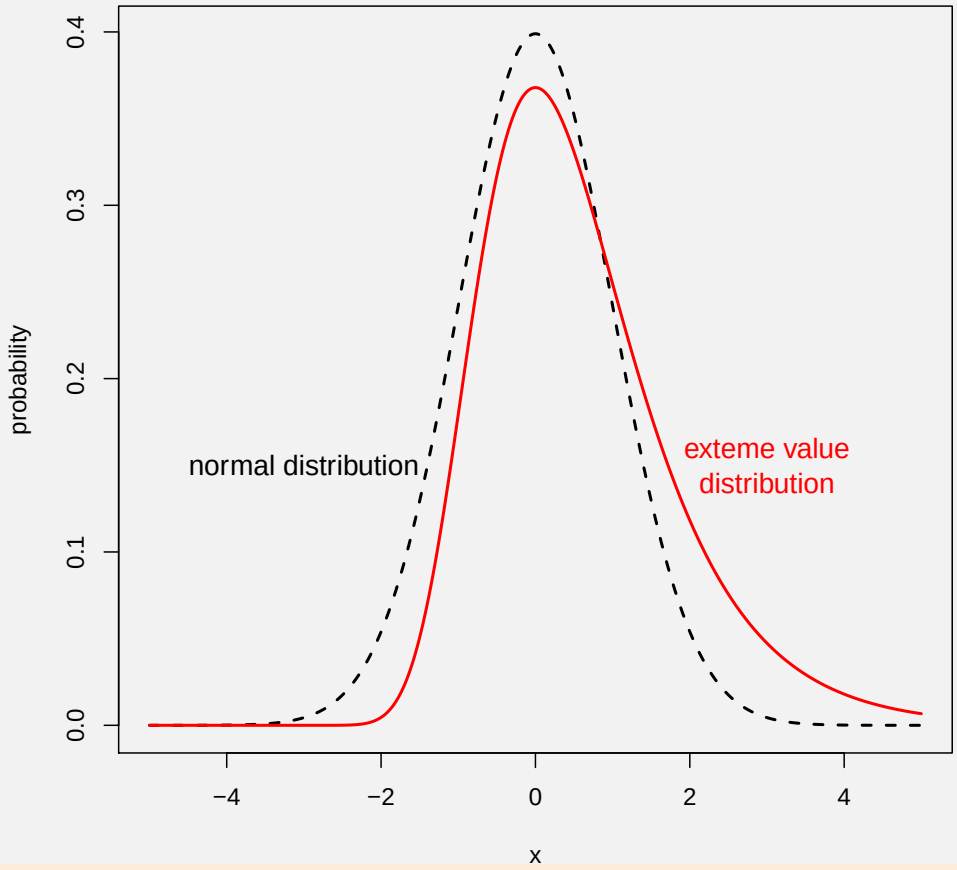
The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

Standard normal and extreme value distributions

The scores of random, ungapped local alignments are proved to follow an extreme value distribution



The *E* value

- From the expected distribution we can calculate the number of scores we expect to see by chance
- The Expect value E is the number of chance alignments one should expect to exceed a score S from a given database search
- The E value is related to the score S by the Karlin-Altschul equation

$$E = Kmn\mathrm{e}^{-\lambda S}$$

  - K and λ are the "Karlin-Altschul statistics", that help normalize the search size
  - m and n are the lengths of the aligned sequences

The use of bit scores

- BLAST uses the same substitution matrices as pairwise alignment schemes, but the scores can't be compared
- Bit scores normalize different searches to allow comparison using different substituion matrices and databases

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

so the E value for exceeding a given bit score is

$$E = mn2^{-S'}$$

What does *E* mean, then?

Relationship between *E* and *p*

p	E
≈ 1.0	2000
0.9999546	10
0.99326205	5
0.86466472	2
0.63212056	1
0.39346934	0.5
0.09516258	0.1
0.00995017	0.01
0.0009995	0.001
1×10^{-4}	1×10^{-4}

Interpreting the E scores

- With an *E* value of 10, you can expect 10 HSPs above the threshold from chance
- Looking at the *p* values, they would be hard to use!
- This is because of the number of opportunities for false positives
- We will see this problem later when looking at functional genomic data

The challenge with gaps

- BLAST provides an approximate model of local alignment, but not a comprehensive model guaranteed to find the best
- *Gaps* have even less
  - Gaps do not have the same probabilistic framework for understanding expectations
  - Models for gaps have been largely based on computational experiments

With short query sequences, many things change

- Short sequences cannot accumulate a high score!
- The *BLAST* web programs can make adjustments for short queries automatically

What about those other parameters?

- Low complexity
  - Low complexity regions (repeated sequences, etc.) can give many spurious hits. These can be *masked* in a search
- Composition-based statistics
  - Different sequences (and different databases) may have very different amino acid compositions
  - *Compositional adjustments* assign a scaling correction to account for these
  - In some cases, *compositional score matrix adjustment* may also be applied. If you click it, it will fall back to compositional adjustments if needed

Recent variations of BLAST

Two-hit BLAST requires two words, not just one, near each other "on the diagonal". These can be combined to use for a single extension.

Gapped Blast Define a score S_g that will trigger a gapped extension. Gapped extensions are costly, but parameters mean relatively few. (Altschul, Madden, et al. 1997)

Megablast Looks for slight variations (sequencing errors). Can be used for very large sequences, or even combining many queries if the alignments are expected to be good (Zhang et al. 2000)

Topic

Where we left off

Introducing BLAST

A BLAST run step by step

The BLAST search algorithm

Interpreting BLAST results

Roundup and next time

What we have learned (1)

1. BLAST is a family of tools available on the web for searching sequence databases
2. BLAST family programs rapidly search nucleotide and protein sequence databases, using nucleotide and protein sequence queries
3. The BLAST algorithm works though a three step process of constucting words above a threshold, scanning the database for matching words, and extending the hits to find high scoring pairs

What we have learned (2)

1. The BLAST results can be interpreted in terms of *E* values and scores
2. Scores for alignments are expected to follow an extreme value distribution, which is used in the calculation of *E* values
3. Many parameters can adjust BLAST searches for particular search needs

Next time

- Next time, we will look at how to find and describe sequences belonging to sequence families (much like we discussed in multiple sequence alignment).
- These methods of profiles and patterns extend the reach of sequence analysis

Bibliography

Altschul, S F, W Gish, et al. (1990). "Basic local alignment search tool." In: *Journal of Molecular Biology* 215.3, pp. 403–10. DOI: 10.1006/jmbi.1990.9999 (cit. on p. 8).

Altschul, S F, T L Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." In: *Nucleic acids research* 25.17, pp. 3389–402 (cit. on p. 40).

Zhang, Z et al. (2000). "A greedy algorithm for aligning DNA sequences." In: *Journal of computational biology* 7.1-2, pp. 203–14 (cit. on p. 40).