An Introduction to Bioinformatics Tools

Part 2: BLAST

The James Hutton Institute

Leighton Pritchard and Peter Cock

Table of Contents

Introduction

Alignment

BLAST

BLAST Statistics

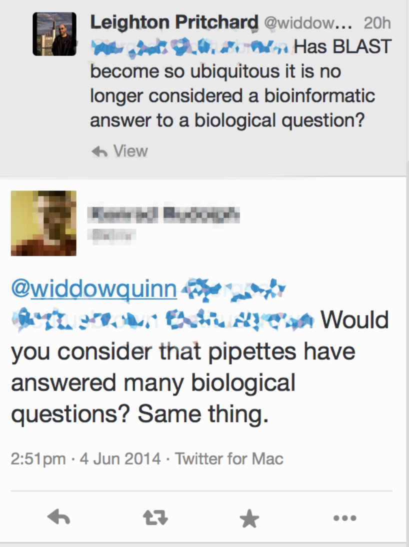
Using BLAST

Learning Outcomes

- How BLAST searches work
- How the way BLAST searches work affects your results
- Why search parameters matter
- Setting search parameters

About Bioinformatics Tools

A Recent Twitter Conversation



Leighton Pritchard @widdow… 20h

[redacted] Has BLAST become so ubiquitous it is no longer considered a bioinformatic answer to a biological question?

View

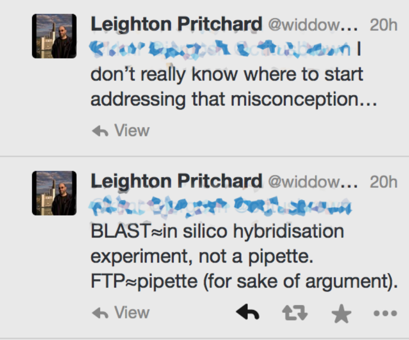
[redacted]

[redacted]

@widdowquinn [redacted] Would you consider that pipettes have answered many biological questions? Same thing.

2:51pm · 4 Jun 2014 · Twitter for Mac

A Recent Twitter Conversation



Leighton Pritchard @widdow… 20h

[redacted] I don’t really know where to start addressing that misconception…

Leighton Pritchard @widdow… 20h

[redacted]

BLAST≈in silico hybridisation experiment, not a pipette. FTP≈pipette (for sake of argument).

Why So Much Detail?

- You're going to go away and do lots of BLAST searches
- Everyone uses BLAST - not everyone uses it well
- Easier to fix problems if you know how it works
- Understanding what's going on helps avoid misuse/abuse
- Understanding what's going on helps use the tool more effectively
- Not so much detail, really
  - like knowing about T_m and ion concentration effects, not molecular orbitals or thermodynamics (but ask if you're interested ;))

Table of Contents

Introduction

Alignment

BLAST

BLAST Statistics

Using BLAST

What BLAST Is

- BLAST:
  - Basic (it's actually sophisticated)
  - Local Alignment (what it does: local sequence alignment)
  - Search Tool (what it does: search against a database)

What BLAST Is

- BLAST:
  - Basic (it's actually sophisticated)
  - Local Alignment (what it does: local sequence alignment)
  - Search Tool (what it does: search against a database)
- The most important software package in bioinformatics?
- Fast, robust, sequence similarity search tool
- Does not necessarily produce optimal alignments
- Not foolproof.

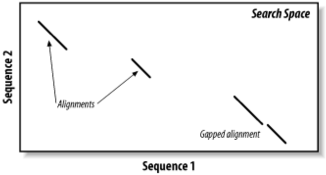
What A BLAST Search Is

- Every BLAST search is an *in silico* hybridisation experiment
- BLAST search = identification of similar sequences in a given database
- Results depend on:
  - query sequence
  - BLAST program (including version and BLAST vs BLAST+)
  - database
  - parameters

Alignment Search Space

Consider two biological sequences to be aligned. . .

- One sequence on the x-axis, the other on the y-axis
- Each point in space is a pairing of two letters
- Ungapped alignments are diagonal lines in the search space, gapped alignments have short 'breaks'
- There may be one or more "optimal" alignments



Global vs Local Alignment

- Global alignment: sequences are aligned along their entire lengths
- Local alignment: the best subsequence alignment is found

Global vs Local Alignment

- Global alignment: sequences are aligned along their entire lengths
- Local alignment: the best subsequence alignment is found
- Consider an alignment of the same gene from two distantly-related eukaryotes, where:
  - Exons are conserved and small in relation to gene locus size
  - Introns are not well-conserved but large in relation to gene locus size
- Local alignment will align the conserved exon regions
- Global alignment will align the whole (mostly unrelated) locus

Our Goal

- We aim to align the words
  - `COELACANTH`
  - `PELICAN`

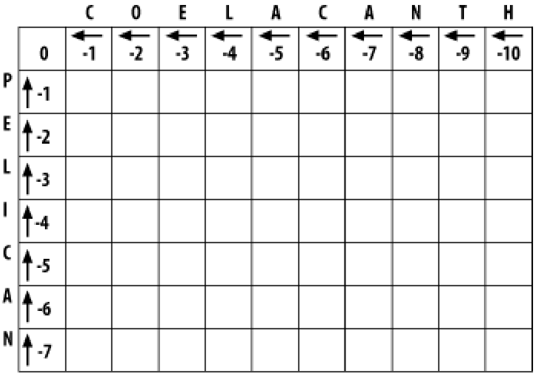
Our Goal

- We aim to align the words
  - `COELACANTH`
  - `PELICAN`
- Each identical letter (match) scores $+1$
- Each different letter (mismatch) scores -1
- Each gap scores -1

Our Goal

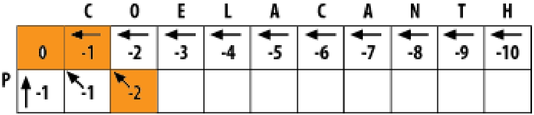
- We aim to align the words
  - `COELACANTH`
  - `PELICAN`
- Each identical letter (match) scores $+1$
- Each different letter (mismatch) scores -1
- Each gap scores -1
- *All sequence alignment is maximisation of an alignment score* - a mathematical operation.

Initialise the matrix

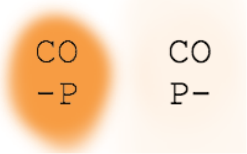


		C	O	E	L	A	C	A	N	T	H
	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1										
E	↑ -2										
L	↑ -3										
I	↑ -4										
C	↑ -5										
A	↑ -6										
N	↑ -7										

Fill the cells



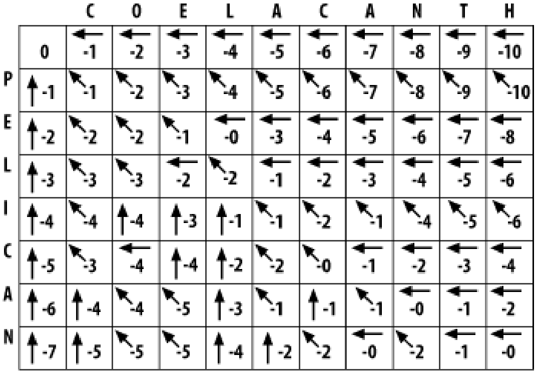
		C	O	E	L	A	C	A	N	T	H
	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1	↖ -1	↖ -2								



```
CO
-P
```

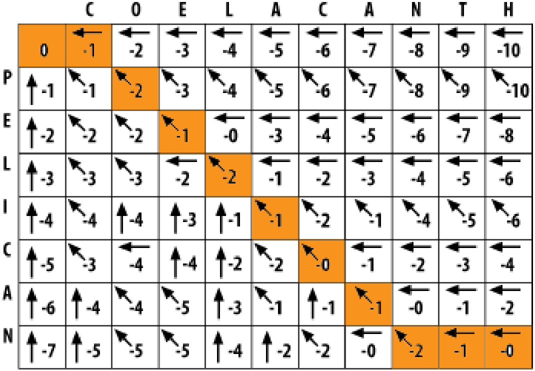
```
CO
P-
```

Fill the matrix – represents all possible alignments & scores



		C	O	E	L	A	C	A	N	T	H
	0	← -1	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1	↖ -1	↖ -2	↖ -3	↖ -4	↖ -5	↖ -6	↖ -7	↖ -8	↖ -9	↖ -10
E	↑ -2	↖ -2	↖ -2	↖ -1	← -0	← -3	← -4	← -5	← -6	← -7	← -8
L	↑ -3	↖ -3	↖ -3	← -2	↖ -2	← -1	← -2	← -3	← -4	← -5	← -6
I	↑ -4	↖ -4	↑ -4	↑ -3	↑ -1	↖ -1	↖ -2	↖ -1	↖ -4	↖ -5	↖ -6
C	↑ -5	↖ -3	← -4	↑ -4	↑ -2	↖ -2	↖ -0	← -1	← -2	← -3	← -4
A	↑ -6	↑ -4	↖ -4	↖ -5	↑ -3	↖ -1	↑ -1	↖ -1	← -0	← -1	← -2
N	↑ -7	↑ -5	↖ -5	↖ -5	↑ -4	↑ -2	↖ -2	← -0	↖ -2	← -1	← -0

Traceback



		C	O	E	L	A	C	A	N	T	H
	0	**← -1**	← -2	← -3	← -4	← -5	← -6	← -7	← -8	← -9	← -10
P	↑ -1	↖ -1	**↖ -2**	↖ -3	↖ -4	↖ -5	↖ -6	↖ -7	↖ -8	↖ -9	↖ -10
E	↑ -2	↖ -2	↖ -2	**↖ -1**	← -0	← -3	← -4	← -5	← -6	← -7	← -8
L	↑ -3	↖ -3	↖ -3	← -2	**↖ -2**	← -1	← -2	← -3	← -4	← -5	← -6
I	↑ -4	↖ -4	↑ -4	↑ -3	↑ -1	**↖ -1**	↖ -2	↖ -1	↖ -4	↖ -5	↖ -6
C	↑ -5	↖ -3	← -4	↑ -4	↑ -2	↖ -2	**↖ -0**	← -1	← -2	← -3	← -4
A	↑ -6	↑ -4	↖ -4	↖ -5	↑ -3	↖ -1	↑ -1	**↖ -1**	← -0	← -1	← -2
N	↑ -7	↑ -5	↖ -5	↖ -5	↑ -4	↑ -2	↖ -2	← -0	**↖ -2**	**← -1**	**← -0**

```
COELACANTH
-PELICAN-
```

Algorithms

- Global: Needleman-Wunsch (as in example)
- Local: Smith-Waterman (differs from example)

Algorithms

- Global: Needleman-Wunsch (as in example)
- Local: Smith-Waterman (differs from example)
- Biological information encapsulated *only* in the scoring scheme (matches, mismatches, gaps)

Algorithms

- Global: Needleman-Wunsch (as in example)
- Local: Smith-Waterman (differs from example)
- Biological information encapsulated *only* in the scoring scheme (matches, mismatches, gaps)
- NW/SW are *guaranteed* to find the optimal match *with respect to the scoring system being used*
- BUT the optimal alignment is a biological approximation: no scoring scheme encapsulates biological “truth”
- Any pair of sequences can be aligned: finding meaning is up to you

Table of Contents

Introduction

Alignment

BLAST

BLAST Statistics

Using BLAST

BLAST Is A Heuristic

- BLAST does not use Needleman-Wunsch or Smith-Waterman
- BLAST *approximates* dynamic programming methods
- BLAST is not guaranteed to give a mathematically optimal alignment

BLAST Is A Heuristic

- BLAST does not use Needleman-Wunsch or Smith-Waterman
- BLAST *approximates* dynamic programming methods
- BLAST is not guaranteed to give a mathematically optimal alignment
- BLAST does not explore the complete search space

BLAST Is A Heuristic

- BLAST does not use Needleman-Wunsch or Smith-Waterman
- BLAST *approximates* dynamic programming methods
- BLAST is not guaranteed to give a mathematically optimal alignment
- BLAST does not explore the complete search space
- BLAST uses heuristics (loosely-defined rules) to refine High-scoring Segment Pairs (HSPs)

BLAST Is A Heuristic

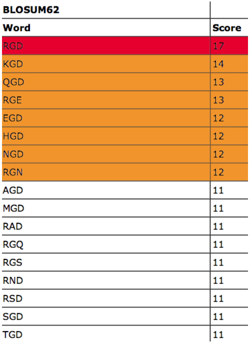
- BLAST does not use Needleman-Wunsch or Smith-Waterman
- BLAST *approximates* dynamic programming methods
- BLAST is not guaranteed to give a mathematically optimal alignment
- BLAST does not explore the complete search space
- BLAST uses heuristics (loosely-defined rules) to refine High-scoring Segment Pairs (HSPs)
- BLAST reports only “statistically-significant” alignments (dependent on parameters)

Steps in the Algorithm

1. Seeding
2. Extension
3. Evaluation

Word Hits

- A *word hit* is a short sequence and its *neighbourhood*
- *neighbourhood*: words of same length whose aligned score is greater than or equal to a threshold value T
- Three parameters: scoring matrix, word size W, and T

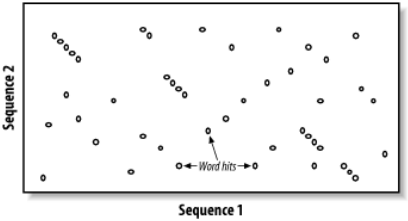


BLOSUM62

Word	Score
RGD	17
KGD	14
QGD	13
RGE	13
EGD	12
HGD	12
NGD	12
RGN	12
AGD	11
MGD	11
RAD	11
RGQ	11
RGS	11
RND	11
RSD	11
SGD	11
TGD	11

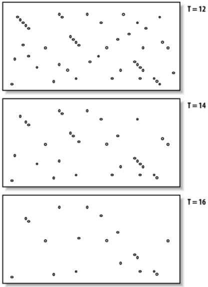
Seeding

- BLAST assumption: significant alignments have *words* in common
- BLAST finds word (*neighbourhood*) hits in the database index
- Word hits are used to *seed* alignments



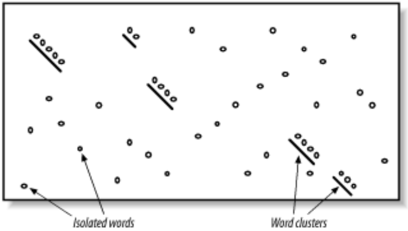
Seeding Controls Sensitivity

- Word size W controls number of hits (smaller words $\implies$ more hits)
- Threshold score T controls number of hits (lower threshold $\implies$ more hits)
- Scoring matrix controls which words match



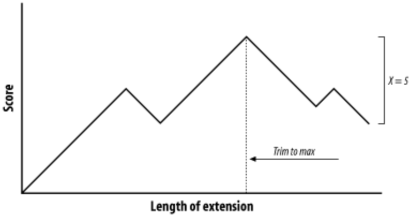
The Two-Hit Algorithm

- BLAST assumption: word hits cluster on the diagonal for significant alignments
- The acceptable distance *A* between words on the diagonal is a parameter of your model
- Smaller distances isolate single words, and reduce search space



Extension

- The best-scoring seeds are extended in each direction
- BLAST does not explore the complete search space, so a rule (heuristic) to stop extension is needed
- Two-stage process:
  - Extend, keeping alignment score, and *drop-off* score
  - When drop-of score reaches a threshold X, trim alignment back to top score



Example

- Consider two sentences (match=+1, mismatch=-1)
  - `The quick brown fox jumps over the lazy dog.`
  - `The quiet brown cat purrs when she sees him.`

Example

- Consider two sentences (match=+1, mismatch=-1)
    - `The quick brown fox jumps over the lazy dog.`
    - `The quiet brown cat purrs when she sees him.`
- Extend to the right from the seed `T`
    - `The quic`
    - `The quie`
    - `123 4565 <- score`
    - `000 0001 <- drop-off score`

Example

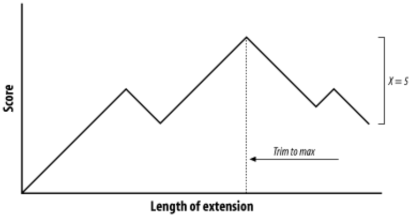
- Consider two sentences (match=+1, mismatch=-1)
  - `The quick brown fox jumps over the lazy dog.`
  - `The quiet brown cat purrs when she sees him.`
- Extend to drop-off threshold
  - `The quick brown fox jump`
  - `The quiet brown cat purr`
  - `123 45654 56789 876 5654 <- score`
  - `000 00012 10000 123 4345 <- drop-off score`

Example

- Consider two sentences (match=+1, mismatch=-1)
  - `The quick brown fox jumps over the lazy dog.`
  - `The quiet brown cat purrs when she sees him.`
- Trim back from drop-off threshold to get optimal alignment
  - `The quick brown`
  - `The quiet brown`
  - `123 45654 56789 <- score`
  - `000 00012 10000 <- drop-off score`

Notes on implementation

- X controls termination of alignment extension, but dependent on:
    - substitution matrix
    - gap opening and extension parameters



Evaluation

- The principle is easy: use a score threshold S to determine strong and weak alignments
  - S is monotonic with E, so an equivalent threshold can be calculated
- Score S is independent of database size and search space. E values are not.
- Alignment consistency of HSPs is also a factor in the report

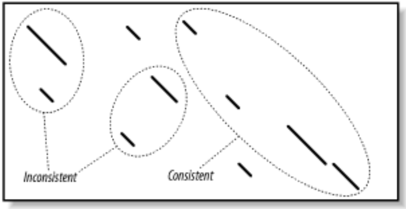


Table of Contents

Introduction

Alignment

BLAST

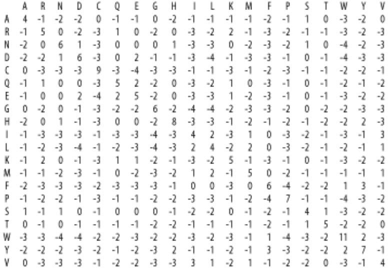
BLAST Statistics

Using BLAST

Log-odds Matrices

- Substitution matrices are your model of evolution
- Substitution matrices are log-odds matrices
    - Positive numbers indicate likely substitutions/similarity
    - Negative numbers indicate unlikely substitutions/dissimilarity

BLOSUM62



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

Choice of Matrix

- Substitution matrix determines the raw alignment score S
  - S is the sum of pairwise scores in an alignment
- BLAST provides, for proteins:
  - `BLOSUM45 BLOSUM50 BLOSUM62 BLOSUM80 BLOSUM90`
  - `PAM30 PAM70 PAM250`
- BLOSUM matrices empirically defined from multiple sequence alignments of $\geq n\%$ identity, for `BLOSUMn`
- For nucleotides: ‘matrix’ defined by match/mismatch (reward/penalty) parameters

Definition

- The Karlin-Altschul equation

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

- Symbols:
    - k: minor constant, adjusts for correlation between alignments
    - m: number of letters in query sequence
    - n: number of letters in the database
    - λ: scoring matrix scaling factor
    - S: raw alignment score

Interpretation

- The Karlin-Altschul equation

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

- E is the number of alignments of a similar score expected by chance when querying a database of the same size and letter frequency, where the letters in that database are randomly-ordered
- Small changes in score S can produce large changes in E
- BUT biological sequence databases are not random!

Table of Contents

Introduction

Alignment

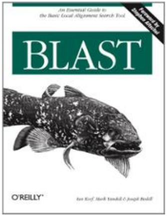
BLAST

BLAST Statistics

Using BLAST

Multiple BLAST tools

- BLASTN *vs* MEGABLAST *vs* TBLASTX *vs* ...?
- Korf *et al.* (2003) BLAST is really good for theory part, but practical examples dated due to changes with BLAST+


An Essential Guide to
the Basic Local Alignment Search Tool
Foreword by Stephen Altschul
BLAST
O'REILLY
Ian Korf, Mark Yandell & Joseph Bedell

Multiple flavours of BLAST

- NCBI “legacy” BLAST
  - Now obsolete and not being updated
  - Spawned offshoots including:
    - WU-BLAST aka AB-BLAST (commerical)
    - MPI-BLAST for use on clusters
    - Versions to run on graphics cards
- NCBI BLAST+
  - Re-written in 2009 using C++ instead of C
  - Many improvements
  - Slightly different output
  - Different commands used to run it

Multiple ways to run BLAST

- BLAST+ at the command line (today)
- Via a script or programming language
- Via a graphical tool like BioEdit, CLCbio, Blast2GO
- Via the NCBI website
- Via a genome consortium website
- Via a Galaxy web server
- etc
- Offers flexibility *but* different settings/options/versions

Multiple places to run BLAST

- On the NCBI servers, e.g. via website or tool
- On 3rd party servers, e.g. via websites
- On your own computer
- On our Linux cluster

Core BLAST tools: Query sequences vs Database

- Nucleotide *vs* Nucleotide:
    - `blastn` (covering blastn, megablast, dc-megablast)
- Translated nucleotide *vs* Protein:
    - `blastx`
- Protein *vs* Translated nucleotide:
    - `tblastn`
- Protein *vs* Protein:
    - `blastp`, `psiblast`, `phiblast`, `deltablast`

See `http://blast.ncbi.nlm.nih.gov/` for a reminder ;)

The BLAST tools have built in help

```
$ blastp -h
USAGE
  blastp [-h] [-help] [-import_search_strategy filename]
    [-export_search_strategy filename] [-task task_name] [-db database_name]
    [-dbsize num_letters] [-gilist filename] [-seqidlist filename]
    [-negative_gilist filename] [-entrez_query entrez_query]
    [-db_soft_mask filtering_algorithm] [-db_hard_mask filtering_algorithm]
    [-subject subject_input_file] [-subject_loc range] [-query input_file]
    [-out output_file] [-evalue evalue] [-word_size int_value]
    [-gapopen open_penalty] [-gapextend extend_penalty]
    [-xdrop_ungap float_value] [-xdrop_gap float_value]
    [-xdrop_gap_final float_value] [-searchsp int_value] [-max_hsps int_value]
    [-sum_statistics] [-seg SEG_options] [-soft_masking soft_masking]
    [-matrix matrix_name] [-threshold float_value] [-culling_limit int_value]
    ...
    [-max_target_seqs num_sequences] [-num_threads int_value] [-ungapped]
    [-remote] [-comp_based_stats compo] [-use_sw_tback] [-version]

DESCRIPTION
   Protein-Protein BLAST 2.2.29+

Use '-help' to print detailed descriptions of command line arguments
```

Minimal example of BLAST+ at the command line

```
$ blastp -query my_input.fasta -db my_database -out my_output.txt
```

- Replace blastp with the appropriate tool, e.g. blastn
- Replace my_input.fasta with your actual filename
- Replace my_database with your actual database, e.g. nr
- Replace my_output.txt with your desired output filename
- Best to avoid spaces in your folder and filenames!

e.g.

```
$ blastp -query query.fasta -db dbA -out my_output.txt
```

Setting the BLAST+ output format

```
$ blastp -help
USAGE
...

 *** Formatting options
 -outfmt <String>
   alignment view options:
     0 = pairwise,
     1 = query-anchored showing identities,
     2 = query-anchored no identities,
     3 = flat query-anchored, show identities,
     4 = flat query-anchored, no identities,
     5 = XML Blast output,
     6 = tabular,
     7 = tabular with comment lines,
     8 = Text ASN.1,
     9 = Binary ASN.1,
    10 = Comma-separated values,
    11 = BLAST archive format (ASN.1)

   ...
   Default = ‘0’
...
```

Setting the BLAST+ output format

Default is plain text pairwise alignments, for humans:

```
$ blastp -query query.fasta -db dbA -out my_output.txt
...
```

XML output can be useful (e.g. for BLAST2GO):

```
$ blastp -query query.fasta -db dbA -out my_output.xml -outfmt 5
...
```

Tabular output is easiest to filter, sort, etc:

```
$ blastp -query query.fasta -db dbA -out my_output.tab -outfmt 6
...
```

Setting the e-value threshold

Check the built in help:

```
$ blastp -help
USAGE
...
 -evalue <Real>
   Expectation value (E) threshold for saving hits
   Default = ‘10’
...
```

Example using 0.0001 or 1×10^{-5} in scientific notation (1e−5)

```
$ blastp -query query.fasta -db dbA -out my_output.txt -evalue 1e-5
...
```

In Conclusion

- Every BLAST search is an experiment
- Badly-designed searches can give you bad results
- Knowing how BLAST works helps improve search design
- BLAST results still require inspection and interpretation