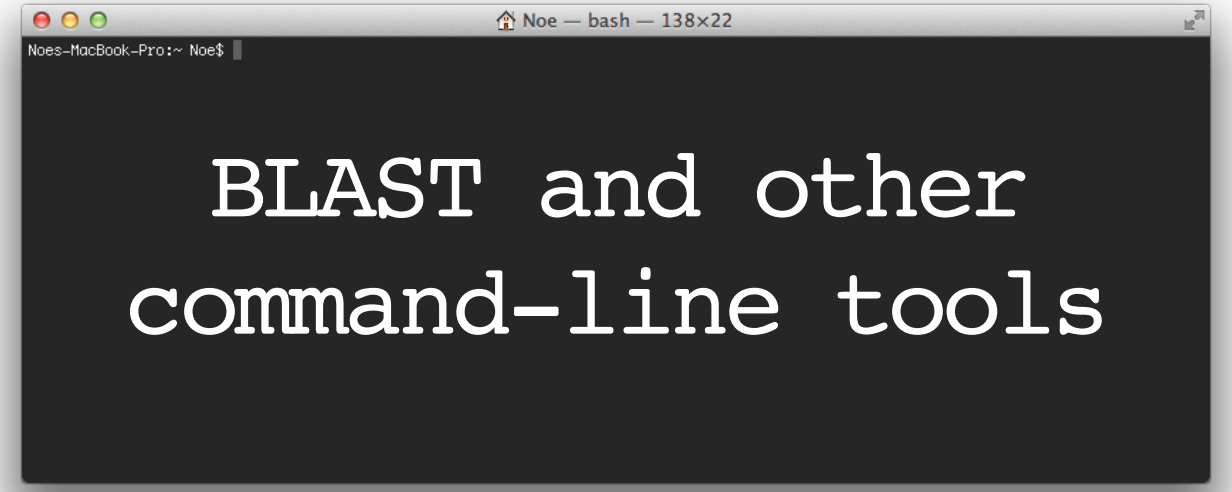


BLAST and other command-line tools

BTI

JUNE 29, 2017

Adrian Powell

Download class files

1. Go to your Desktop
2. Create a folder called blast_data
3. Download the file blast_data.tar.gz from:

ftp://ftp.solgenomics.net/bioinfo_class/interns/2017/

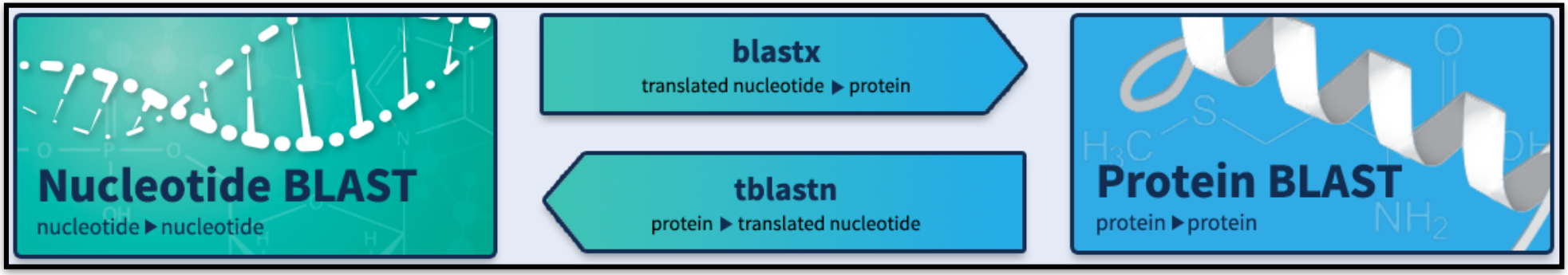
4.Decompress the file in blast_data

Class Content

- Introduction
- BLAST
- BLAST+
- AWK
- Custom Scripts

BLAST

https://blast.ncbi.nlm.nih.gov/Blast.cgi



The **Basic Local Alignment Search Tool** (BLAST) finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

FASTA format

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol at the beginning.

http://www.ncbi.nlm.nih.gov/

description line

sequence data

```
>sequence_ID1 description
ATGCGCGCGCGCGCGCGGGTAGCAGATGACGACACAGAGCGAGGATGCGCTGAGAGTA
GTGTGACGACGATGACGGAAAATCAGATGGACCCGATGACAGCATGACGATGGGACGGGA
AAGATTGGACCAGGACAGGACCAGGACCAGGACCAGGGATTAGA
>sequence_ID2 description
ATGGGGGGGACGACGATGGACACAGAGACAGAGACGACGACAGCAGACAGATTTACCTTA
GACGAGATAGGAGAGACGACAGATATATATATATAGCAGACAGACAGACATTTAGACGAG
ACGACGATAGACGATaaaaataa
```

BLAST classic output

```
BLASTN 2.2.26 [Sep-21-2011]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= Untitled_sequence
         (780 letters)

Database: ITAG_release_2.40_cdnas
           34,725 sequences; 41,981,568 total letters

Searching..................................................done
```

Sequences producing significant alignments:		Score (bits)	E Value
Solyc04g081490.2	beta-tubulin tub	1501	0.0
Solyc12g089310.1	Tubulin beta-1 chain (AHRD V1 ***- D7KT68_ARALY...	492	e-138
Solyc10g085020.1	Tubulin beta chain (AHRD V1 ***- B9HP96_POPTR);...	440	e-123
Solyc10g086760.1	Tubulin beta chain (AHRD V1 ***- B9HP96_POPTR);...	357	9e-98
Solyc06g076640.2	Tubulin beta chain (AHRD V1 ***- B9GKJ5_POPTR);...	315	3e-85
Solyc06g005910.2	Tubulin beta chain (AHRD V1 ***- B9GWG9_POPTR);...	274	1e-72
Solyc10g080940.1	Tubulin beta chain (AHRD V1 ***- B9HNJ2_POPTR);...	226	2e-58
Solyc03g118760.2	Tubulin beta chain (AHRD V1 ***- B9HNJ2_POPTR);...	220	1e-56
Solyc03g025730.2	Tubulin beta chain (AHRD V1 ***- B9HP96_POPTR);...	178	4e-44
Solyc06g035970.2	Tubulin beta chain (AHRD V1 ***- B9GKJ5_POPTR);...	159	4e-38
Solyc07g052040.1	Tubulin beta-1 chain (AHRD V1 ***- D7KT68_ARALY...	109	3e-23

BLAST classic output

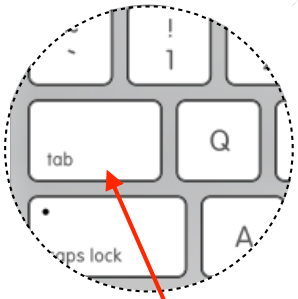
```
>Solyc04g081490.2 beta-tubulin tub
          Length = 1651

 Score = 1501 bits (757), Expect = 0.0
 Identities = 757/757 (100%)
 Strand = Plus / Plus

Query: 24   cttcatcttcatcttcatctcttcttcttcttattctctcattcctctcatcaatttttt 83
            |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 12   cttcatcttcatcttcatctcttcttcttcttattctctcattcctctcatcaatttttt 71


Query: 84   tcatcaaaaaaactaagagaaaatgagagaaattcttcacattcaaggaggacaatgtgg 143
            |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 72   tcatcaaaaaaactaagagaaaatgagagaaattcttcacattcaaggaggacaatgtgg 131
```

```
>Solyc12g089310.1 Tubulin beta-1 chain (AHRD V1 ***- D7KT68_ARALY); contains Interpro
           domain(s)  IPR002453  Beta tubulin
          Length = 1356

 Score =  492 bits (248), Expect = e-138
 Identities = 571/678 (84%), Gaps = 3/678 (0%)
 Strand = Plus / Plus

Query: 106 atgagagaaattcttcacattcaaggaggacaatgtgg-aaccaaatcggttccaaattc 165
           |||||||||||  | ||||||||||||| ||||| || ||||||||||||| ||||||
Sbjct: 1   atgagagaaatcttacacattcaaggaggccaatgcgggaaccaaatcggttcaaaattc 60
```

BLAST classic output

```
>Solyc04g081490.2 beta-tubulin tub
          Length = 451

 Score =  467 bits (1202), Expect = e-164
 Identities = 225/225 (100%), Positives = 225/225 (100%)
 Frame = +1

Query: 106 MREILHIQGGQCGNQIGSKFWEVICDEHGVDPTGRYKGTAAESDLQLERINVYFNEASGG 285
           MREILHIQGGQCGNQIGSKFWEVICDEHGVDPTGRYKGTAAESDLQLERINVYFNEASGG
Sbjct: 1   MREILHIQGGQCGNQIGSKFWEVICDEHGVDPTGRYKGTAAESDLQLERINVYFNEASGG 60



>Solyc10g080940.1 Tubulin beta chain  IPR002453  Beta tubulin
          Length = 452

 Score =  447 bits (1151), Expect = e-156
 Identities = 213/227 (93%), Positives = 222/227 (97%)
 Frame = +1

Query: 100 EKMREILHIQGGQCGNQIGSKFWEVICDEHGVDPTGRYKGTAAESDLQLERINVYFNEAS 279
           EKMREILHIQGGQCGNQIGSKFWEV+CDEHG+DPTGRY GT   SDLQLER+NVY+NEAS
Sbjct: 4   EKMREILHIQGGQCGNQIGSKFWEVVCDEHGIDPTGRYVGT---SDLQLERVNVYYNEAS 60
```

BLAST tabular output



Tab-delimited files are a very common format in scientific data. They consist in columns of text separated by tabs. Other file formats could have different delimiters.

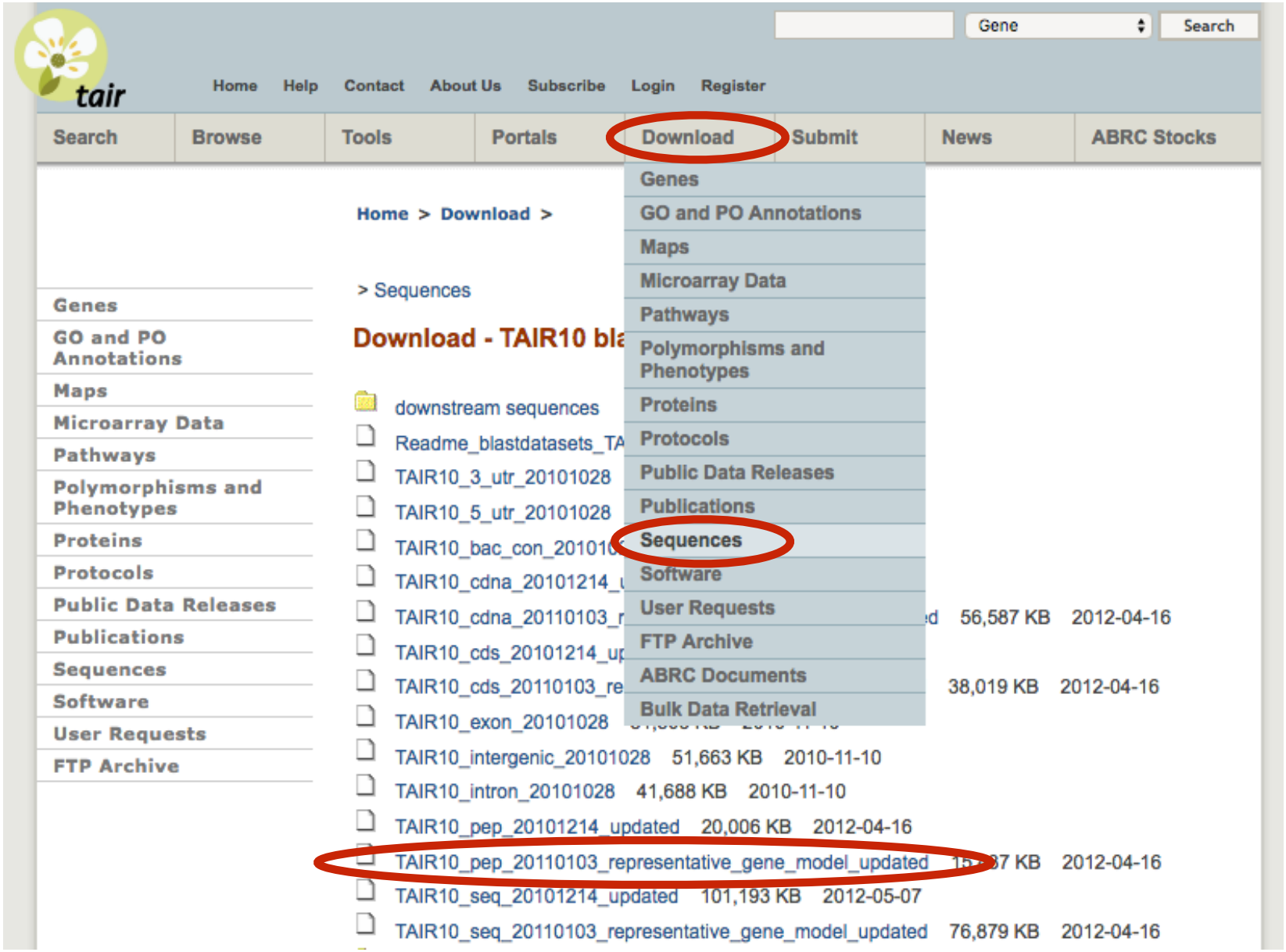
Query	Subject	id %	length	mismatch	gaps	qstart	qend	sstart	send	evalue	score
ATCG00500.1	PACid:23047568	64.88	299	64	2	220	477	112	410	5e-131	388
ATCG00500.1	PACid:23052247	58.88	321	69	3	220	477	381	701	3e-117	361
ATCG00890.1	PACid:16418828	90.60	117	11	0	18	134	1	117	1e-71	220
ATCG00890.1	PACid:16412855	90.48	147	14	2	41	387	27	173	1e-68	214
ATCG00280.1	PACid:24129717	95.99	474	19	0	1	474	1	474	0.0	847
ATCG00280.1	PACid:24095593	95.36	474	22	0	1	474	1	474	0.0	840
ATCG00280.1	PACid:20871697	94.94	474	24	0	1	474	1	474	0.0	837

Tabular blast output example

Blast, SAM (mapping), BED, VCF (SNPs), GTF, GFF ...

Class Content

- Introduction
- BLAST formatdb; fastacmd; blastall
- BLAST+
- AWK
- Custom Scripts

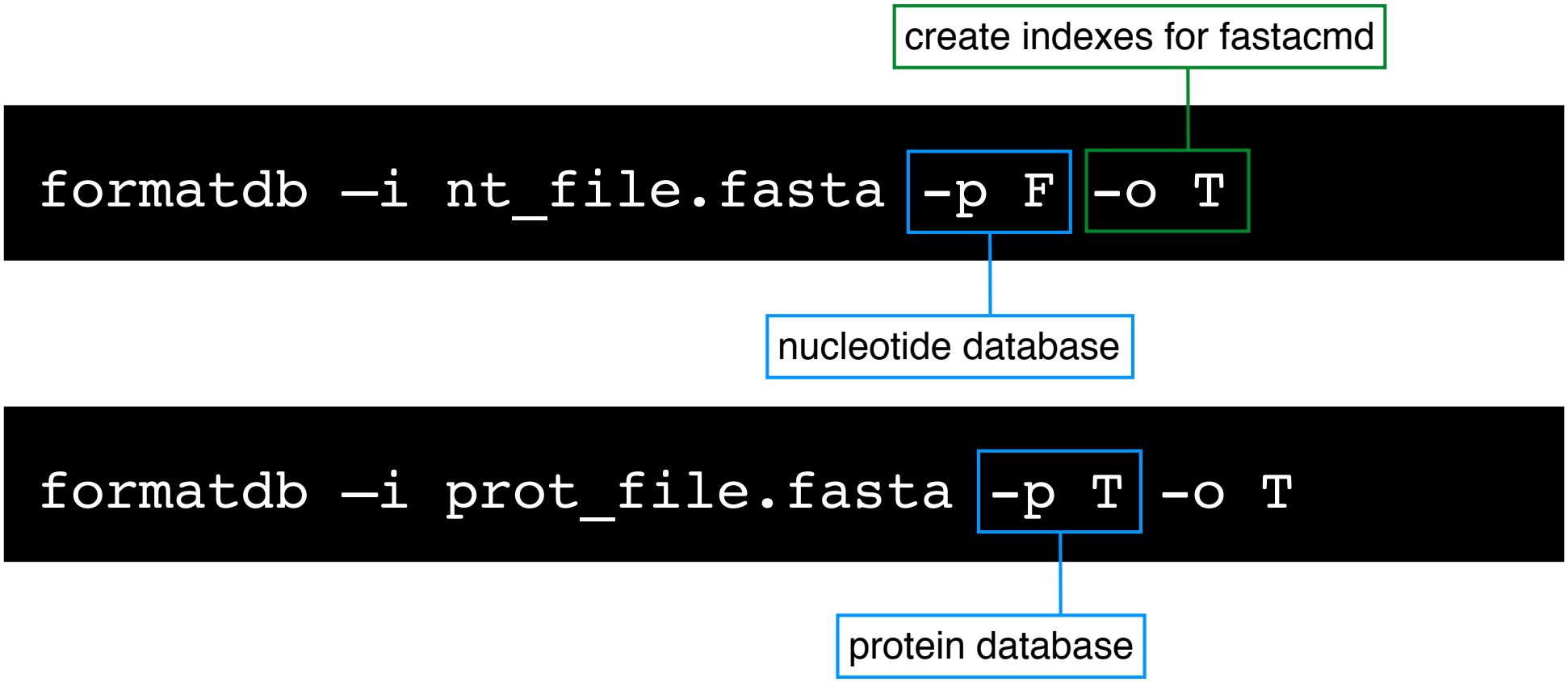

tair
Gene
Search
Home
Help
Contact
About Us
Subscribe
Login
Register
Search
Browse
Tools
Portals
Download
Submit
News
ABRC Stocks
Genes
GO and PO Annotations
Maps
Microarray Data
Pathways
Polymorphisms and Phenotypes
Proteins
Protocols
Public Data Releases
Publications
Sequences
Software
User Requests
FTP Archive
ABRC Documents
Bulk Data Retrieval
Home > Download >
> Sequences
Download - TAIR10 bla
downstream sequences
Readme_blastdatasets_TA
TAIR10_3_utr_20101028
TAIR10_5_utr_20101028
TAIR10_bac_con_20101
TAIR10_cdna_20101214_u
TAIR10_cdna_20110103_r
TAIR10_cds_20101214_up
TAIR10_cds_20110103_re
TAIR10_exon_20101028
TAIR10_intergenic_20101028 51,663 KB 2010-11-10
TAIR10_intron_20101028 41,688 KB 2010-11-10
TAIR10_pep_20101214_updated 20,006 KB 2012-04-16
TAIR10_pep_20110103_representative_gene_model_updated
TAIR10_seq_20101214_updated 101,193 KB 2012-05-07
TAIR10_seq_20110103_representative_gene_model_updated 76,879 KB 2012-04-16
56,587 KB 2012-04-16
38,019 KB 2012-04-16

formatdb

```
formatdb --help
```

```
-t  Title for database file [String]  Optional
-i  Input file(s) for formatting [File In]  Optional
-p  Type of file
        T - protein
        F - nucleotide [T/F]  Optional
   default = T
-o  Parse options
        T - True: Parse SeqId and create indexes.
        F - False: Do not parse SeqId. Do not create indexes.
[T/F]  Optional
   default = F
-n  Base name for BLAST files [String]  Optional
```

formatdb


create indexes for fastacmd
formatdb –i nt_file.fasta -p F -o T
nucleotide database
formatdb –i prot_file.fasta -p T -o T
protein database

Create BLAST databases

1. Create a BLAST database for TAIR10_pep.fasta

2. Create a BLAST database for tomato_1000cdna.fasta

fastacmd

fastacmd --help

```
-d  Database [String]  Optional
-p  Type of file
       G - guess mode (look for protein, then nucleotide)
       T - protein
       F - nucleotide [String]  Optional
-s  Comma-delimited search string(s).
    e.g. 555, AC147927, 'gnl|dbname|tag' [String]  Optional
-i  Input file with GIs/accessions/loci for batch
    retrieval [String]  Optional
-l  Line length for sequence [Integer]  Optional
  default = 80
-o  Output file [File Out]  Optional
  default = stdout
-D  Dump the entire database as (default is not to dump anything):
    1 FASTA

-L  Range of sequence to extract (Format: start,stop)
    0 in 'start' refers to the beginning of the sequence
    0 in 'stop' refers to the end of the sequence [String]  Optional
  default = 0,0
-I  Print database information only (overrides all other options) [T/F]
  default = F
```

fastacmd

```
fastacmd -d blast_prot_db -p T -i gene_list.txt
```

input file

```
fastacmd -d blast_nt_db -p F -s 'gene_id1,gene_id2'
```

input is a list of gene ids

Fastacmd Exercises

1. Get the sequences from the genes in 5_genes_list.txt and save them in a file called tomato5.fasta

2. Get the sequence for AT1G01190.1 and AT5G10980.1

3. Get the sequence AT1G01190.1 in a file called at1g01190.fasta with a line length of 60 AAs

4. Get the sequence AT5G10980.1 in a file called at5g10980.fasta with a line length of 100 AAs

5. Get the info from TAIR10_pep.fasta and tomato_1000cdna.fasta

6. Extract the sequence between coordinates 61,180 from AT1G01190.1

BLAST help

```
blastall --help
```

```
  -p  Program Name [String]
  -d  Database [String]
  -i  Query File [File In]
  -e  Expectation value (E) [Real]
  -m  alignment view options:
   0 = pairwise,
   8 = tabular,

  -o  BLAST report Output File [File Out]
Optional
    default = stdout
  -v  Number of database sequences to show one-
line descriptions for (V) [Integer]
    default = 500
  -b  Number of database sequence to show
alignments for (B) [Integer]
    default = 250
  -a  Number of processors to use [Integer]
    default = 1
```

print blast manual

```
man blastall
```

BLAST

```
blastall -p blastn -d blast_db -i input.fasta
```

program

```
blastall -p blastn -d blast_db -i input.fasta -m 8
```

tabular output

```
blastall -p blastn -d db -i input -e '1e-6' -v 10 -b 10 -a 8
```

e value

num descriptions

num alignments

num CPUs

BLAST Exercises

1. Find the 2 top hits between tomato5.fasta and tomato_1000cdna.fasta

2. Find the 5 top orthologous genes from tomato5.fasta in arabidopsis

3. Find the 10 most similar arabidopsis genes to at1g01190.fasta

4. Get the best arabidopsis hit for the genes from tomato_1000cdna.fasta in tabular format and save it in a file called blast_1000_res.txt

Class Content

- Introduction
- BLAST
- BLAST+ makeblastdb; blastn, blastp, blastx
- AWK
- Custom Scripts

makeblastdb

```
makeblastdb –help
```

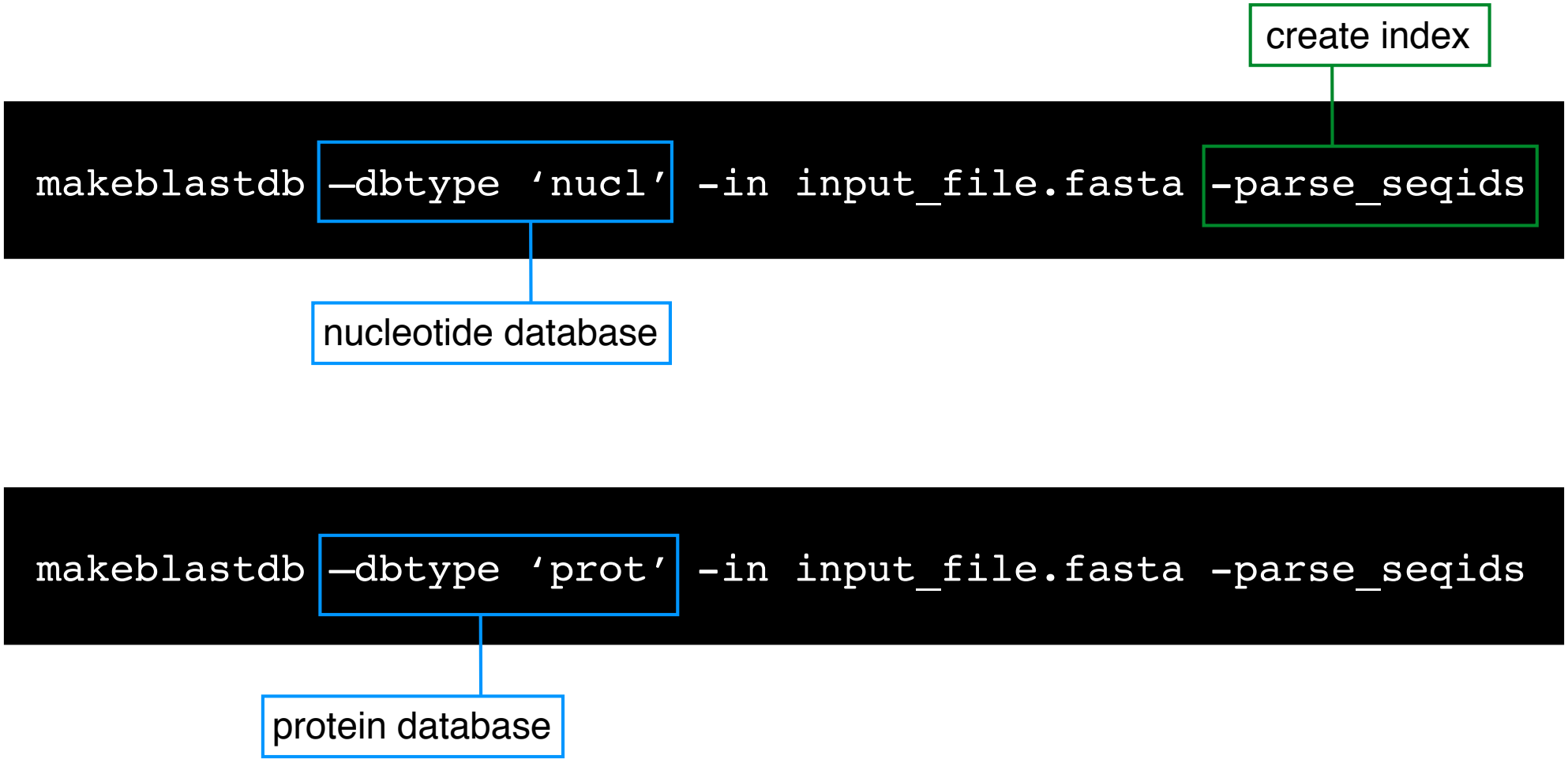
REQUIRED ARGUMENTS
 -dbtype <String, `nucl', `prot'>
   Molecule type of target db

OPTIONAL ARGUMENTS
 *** Input options
 -in <File_In>
   Input file/database name

 *** Configuration options
 -title <String>
   Title for BLAST database
   Default = input file name provided to -in argument
 -parse_seqids
   Option to parse seqid for FASTA input if set, for all other input types
   seqids are parsed automatically

 *** Output options
 -out <String>
   Name of BLAST database to be created
   Default = input file name provided to -in argumentRequired if multiple
   file(s)/database(s) are provided as input

makeblastdb



BLASTn help

blastn -help

```
-query <File_In>
   Input file name
-db <String>
   BLAST database name
 -out <File_Out>
   Output file name
 -evalue <Real>
   Expectation value (E)
threshold for saving hits
   Default = `10'

 *** Formatting options
 -outfmt <String>
   alignment view options:
     0 = pairwise,
     6 = tabular,
 -num_threads <Integer, >=1>
   Number of threads (CPUs) to use in the BLAST search
   Default = `1'
-remote
   Execute search remotely?
  num_threads
 -num_descriptions <Integer, >=0>
   Number of database sequences to show one-line
descriptions for
  Default = `500'
 -num_alignments <Integer, >=0>
   Number of database sequences to show alignments for
   Default = `250'
-perc_identity <Real, 0..100>
   Percent identity
 -max_target_seqs <Integer, >=1>
   Maximum number of aligned sequences to keep
  Default = `500'
```

BLAST+

```
blastn -db nt_blast_db -query nt_input.fasta -outfmt 6
```

tabular output

```
blastx -db prot_db -query nt_input.fasta -evalue '1e-6'
```

e value

```
blastp -db prot_db -query prot_input -num_threads 8
```

num CPUs

BLAST+ Exercises

1. Create a BLAST database for tomato_1000cdna.fasta with name tomato_1000cdna_b+ and title “Solanum lycopersicum 1000 cDNA”

2. Create a BLAST database for TAIR10_pep.fasta with name at_prots

3. Find the 2 most similar genes between tomato5.fasta and tomato_1000cdna.fasta in tabular format

4. Find the 5 top orthologous genes from tomato5.fasta in arabidopsis

5. Find the 10 most similar arabidopsis genes to at1g01190.fasta in tabular format with the columns 'qseqid sseqid pident evalue bitscore'

Class Content

- Introduction
- BLAST
- BLAST+
- AWK
- Custom Scripts

AWK

From the file blast_file, foreach line, if the value in column 3 is equal to or greater than 90, then print the line

```
awk '$3>=90 {print $0}' blast_file
```

From the file blast_file, foreach line, if the value in column 3 is equal to or greater than 95, then print columns 1, 2 and 3

```
awk '$3>=95 {print $1"\t"$2"\t"$3}' blast_file
```

AWK

From a GFF file, print the lines where column 3 is equal to gene

```
awk '$3 == "gene" {print $0}' gff_file
```

Print all the lines finding a pattern in column 2

```
awk '$2 ~ /pattern/ {print $0}' input_file
```

AWK Exercises

1. How many query genes in blast_1000_res.txt have matches with an identity percentage greater than 90%?

2. How many subject genes are arabidopsis mitochondrial genes?

3. Print these mitochondrial gene names and their alignment coordinates (columns 9 and 10)

Class Content

- Introduction
- BLAST
- BLAST+
- AWK
- Custom Scripts

Custom scripts

```
perl blast_filter_3.pl blast_file.txt min_id max_eval min_score

perl FastaExtract.pl -i ids_list.txt -f file.fasta > out.fasta

perl merge_2_lists.pl list1.txt list2.txt col1 col2
```

Scripts Exercises

1. Use the script blast_filter_3.pl to filter the blast result file blast_1000_res.txt for identity >=90, evalue <= 1e-10, and score >= 200

2. Add the descriptions from TAIR10_short_descriptions.txt to blast_1000_res.txt using the script merge_2_lists.pl

3. Extract the sequences for the 5 genes in 5_genes_list.txt from tomato_1000cdna.fasta using the script FastaExtract.pl