Bioinformatics Tool Basics

Running Analysis

- How to run BLAST on command line
- How to setup data files and process
- Development of workflows

Sequence search tools - BLAST

- BLAST is by far the most taught tool in Bioinformatics. I am not going to rehash this for
- See NCBI's Introduction to BLAST
- One of 7 Million pages by Googling "blast introduction tutorial"

BLAST on Biocluster

There are multiple flavors of BLAST (implementations). Focus on the latest version from NCBI (2.7.1+). Default on the cluster is 2.2.30+

We will make links to two files which are ORFs from two yeast species

```
# setup some files to do some searches
$ mkdir BLAST_demo
$ cd BLAST_demo
$ ln -s /bigdata/gen220/shared/data_files/sequences/yeast_chr1_orfs.fa .
$ ln -s /bigdata/gen220/shared/data_files/sequences/C_glabrata_orfs.fa .
```

Now we have some files, set them up for running BLAST. Our question is, what ORFs are similar at the DNA level between these two species.

```
$ module load ncbi-blast/2.7.1+ # load the module on the biocluster
$ makeblastdb -dbtype nucl -in C_glabrata_orfs.fa
$ ls
C_glabrata_orfs.fa      C_glabrata_orfs.fa.nhr
C_glabrata_orfs.fa.nin  C_glabrata_orfs.fa.nsq
yeast_chr1_orfs.fa
$ head -n 15 yeast_chr1_orfs.fa  > YAL027W.cds # get 1st seq for an example
$ blastn -query YAL027W.cds -db C_glabrata_orfs.fa
```

BLAST Running

Change the output format to tab delimited with `-outfmt 6` or `-outfmt 7`

```
$ blastn -query YAL027W.cds -db C_glabrata_orfs.fa \
  -evalue 0.001 -outfmt 7 -out yeast_chr1-vs-Cglabrata.BLASTN.tab
```

This will query the 1 sequence and produce a tab delimited file.

If you provide a multi-FASTA format file with many sequences, each one will be queried and all the results contatanted toegther.

```
$ blastn -query yeast_chr1_orfs.fa -db C_glabrata_orfs.fa \
  -evalue 0.001 -outfmt 7 -out yeast_chr1-vs-Cglabrata.BLASTN.tab
```

BLAST: what are the tools

- `makeblastdb` - index a database (required to do once before searching)
- `blastn` - DNA/RNA to DNA/RNA search
- `blastp` - protein to protein search
- `blastx` - translated query (DNA/RNA) against protein database
- `tblastn` - protein query against translated (DNA/RNA) database
- `tblastx` - translated query and database (both in DNA/RNA but search in protein space)
- `blastdbcmd` - retrieve a sequence from a blast formatted DB

BLAST: what are the cmdline options?

All the tools have documented command line options. Use -h or -help to get detailed info. Sometimes with no arguments will print documentation, other times will not.

```
$ makeblastdb
USAGE
makeblastdb [-h] [-help] [-in input_file] [-input_type type]
-dbtype molecule_type [-title database_title] [-parse_seqids]
[-hash_index] [-mask_data mask_data_files] [-mask_id mask_algo_ids]
[-mask_desc mask_algo_descriptions] [-gi_mask]
[-gi_mask_name gi_based_mask_names] [-out database_name]
[-max_file_sz number_of_bytes] [-logfile File_Name] [-taxid TaxID]
[-taxid_map TaxIDMapFile] [-version]

DESCRIPTION
Application to create BLAST databases, version 2.2.30+

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

BLAST: what are the cmdline options?

```
$ blastn -h
USAGE
blastn [-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]
[-perc_identity float_value] [-qcov_hsp_perc float_value]
[-xdrop_ungap float_value] [-xdrop_gap float_value]
[-xdrop_gap_final float_value] [-searchsp int_value] [-max_hsps int_value]
[-sum_stats bool_value] [-penalty penalty] [-reward reward] [-no_greedy]
[-min_raw_gapped_score int_value] [-template_type type]
[-template_length int_value] [-dust DUST_options]
[-filtering_db filtering_database]
[-window_masker_taxid window_masker_taxid]
[-window_masker_db window_masker_db] [-soft_masking soft_masking]
[-ungapped] [-culling_limit int_value] [-best_hit_overhang float_value]
[-best_hit_score_edge float_value] [-window_size int_value]
[-off_diagonal_range int_value] [-use_index boolean] [-index_name string]
[-lcase_masking] [-query_loc range] [-strand strand] [-parse_deflines]
[-outfmt format] [-show_gis] [-num_descriptions int_value]
[-num_alignments int_value] [-line_length line_length] [-html]]
```

BLAST: some key arguments

- -query - query file name (required)
- -db - database file name (require)
- -evalue - set the evalue cutoff
- -max_target_seqs - max number of hit seqs to show
- -num_alignments - max number of alignments to show
- -num_threads - number of threads (parallel processing to run, 8 will be faster than 2)
- -outfmt - specify a simpler format than the text format, try '-outfmt 6' for tabular format
- -subject - instead of doing a DB search, search for alignments between query sequence and 1 to many subject sequences. Useful when want to just see the alignment of 2 sequences already picked out from other analyses

BLAST: Putting it all together

This is a script. e.g. `run_blast.sh`

```
#!/usr/bin/bash
#SBATCH --nodes 1 --ntasks 4 --mem 2G --job-name=BLASTN
#SBATCH --output=blastn.%A.log
module load ncbi-blast/2.7.1+
CPUS=$SLURM_CPUS_ON_NODE
if [ ! $CPUS ]; then
    CPUS=1
fi
if [ ! -f  C_glabrata_orfs.fa.nhr ]; then
  makeblastdb -in C_glabrata_orfs.fa -dbtype nucl
fi
blastn -query yeast_chr1_orfs.fa -db C_glabrata_orfs.fa \
-evalue 1e-5 -outfmt 6 -out yeastORF-vs-CglabrataORF.BLASTN.tab -num_thread
```

Now submit this script

```
$ sbatch run_blast.sh
$ squeue -u $USER # check on your submitted job
```

Other types of search tools

- **HMMER**
  - Identify conserved domains in a protein
  - Sensitive searches for distant homologs
  - phmmer can be of comparable speed to BLASTP
  - HMMs are a way to not just match a single sequence but match a pattern
- **FASTA**
  - Another tool like BLAST
  - Doesn't require formatting the database
  - FASTA/SSEARCH are more full length optimal alignments instead of individual scoring pairs, a single best alignment generated
  - Global alignment also with ggsearch

Other seq search tools

- **Exonerate**
    - Another aligner useful for cDNA to genome alignment and protein to genome alignment
    - splice-site aware
    - output harder to parse but there is a GFF-flavor output and parsers in some toolkits
- **USEARCH / VSEARCH**
    - fast, near-exact search tool
    - useful in microbiome short-read
- **DIAMOND**
    - fast, near-exact short read search tool
    - translated BLASTX search option to search proteins against a short read database