DENOVO

Introduction

De novo sequence assemblers are a type of program that assembles short nucleotide sequences into longer ones without the use of a reference genome. These are most commonly used in bioinformatic studies to assemble genomes or transcriptomes. Two common types of de novo assemblers are greedy algorithm assemblers and De Bruijn graph assemblers.

Just a note

Please don't rush through the tutorial, this session has only a few commands, but it's important to take your time to explore the results file, the different parameters and try other samples.

Before we start..

Denovo is computationally intensive so we will run the practical in a small subsamples of the data. If you are running out of time and your computer is struggling, I have saved all results in /home/training/denovo_results.zip

```
unzip /home/training/results/denovo_results.zip
```

and explore the data from there.

In this session we will use:

- **Spades**: a very popular tool used across many different organisms. http://cab.spbu.ru/software/spades/

Another good options for denovo with viral data is **IVA** (Iterative Virus Assembler). It is specifically designed for highly variable viral sequences with uneven coverage. http://sanger-pathogens.github.io/iva/

Following the assembly step, we will do some quality control on our assembly with:

- **QUAST**: A QC program used to generate a simple breakdown of assembly metrics. http://bioinf.spbau.ru/quast
- **BLAST**: The well-known online tool used to query input sequence(s) against global

sequence databases.

And then we will map back the original reads to our new denovo assembly:

- **Shiver**: a tool for mapping paired-end short reads to a custom reference sequence constructed using denovo assembled contigs https://github.com/ChrisHIV/shiver/

As usual, let's make a directory for this tutorial:

```
mkdir denovo
cd denovo
```

De novo assembly

SPAdes (St. Petersburg genome assembler) is an assembly toolkit containing various assembly pipeline. To check Spades argument:

```
/home/training/software/SPAdes-3.13.2-Linux/bin/spades.py
```

Basic options:

- -o : specify the output directory
- -1 sample.R1.fastq.gz
- -2 sample.R2.fastq.gz

One of the optional parameters for SPAdes is a range of kmer sizes. A kmer is a sequence of characters of length k sampled from longer strings. The default kmer sizes used by SPAdes are 21, 33, and 55, meaning that the program runs its assembly step three times, each time converting the sequence data into strings of a different length. After completion, it merges the results of all kmer sizes into a single output file. Data for each individual kmer are retained in folders within the output directory.

```
/home/training/software/SPAdes-3.13.2-Linux/bin/spades.py \
-k 21,33,55,77  \
-1 /home/training/Course_Data/PG15-BW001347_sub.R1.fastq.gz \
-2 /home/training/Course_Data/PG15-BW001347_sub.R2.fastq.gz \
-o spadesoutput
```

where:

- -k comma-separated list of k-mer sizes (must be odd and less than 128) [default: 'auto']
- -1 file with forward paired-end reads
- -2 file with reverse paired-end reads

There are a lot of files in the output directory:

```
ls -lhrt spadesoutput
```

but we are mostly interested in the contents of: contigs.fasta

We can look at the contigs:

```
less spadesoutput/contigs.fasta
```

Return each header from the contigs file

```
grep ">" spadesoutput/contigs.fasta
```

return the total number of contigs

```
grep -c ">" spadesoutput/contigs.fasta
```

```
grep -v ">" spadesoutput/contigs.fasta | wc | awk '{print $3-$1}'
```

Assembly statistics

We will now check the quality of the assembly using a tool called QUAST. This program takes as input the contigs.fasta files from the assemblies.

```
quast.py spadesoutput/contigs.fasta -o quast_spades
```

QUAST produces several files:

```
ls -lhrt quast_spades
```

let's check the **report.txt** with:

```
less quast_spades/report.txt
```

QUESTIONS

- How many contigs are there?
- How big is the largest contig?

Interrogating contig identity using BLAST

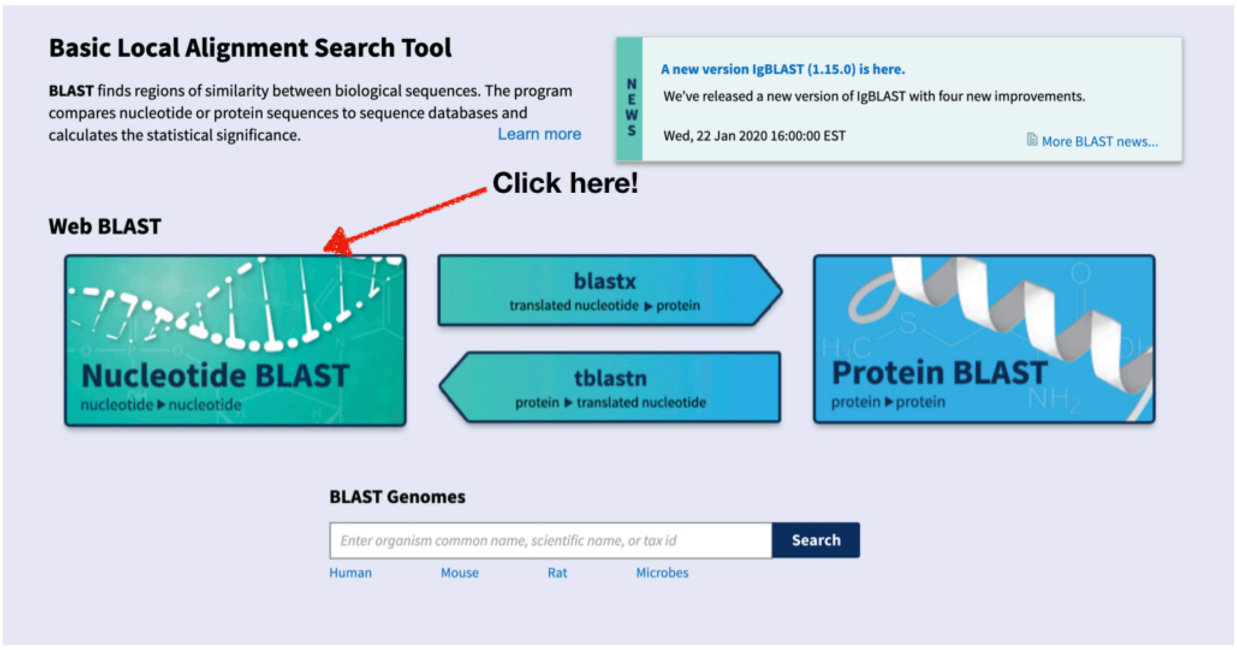
The identity of each conting could be:

- A genome or sub-genomic region of a target of interest
- Host genome
- a contaminant

To start, go to the Blast website:

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

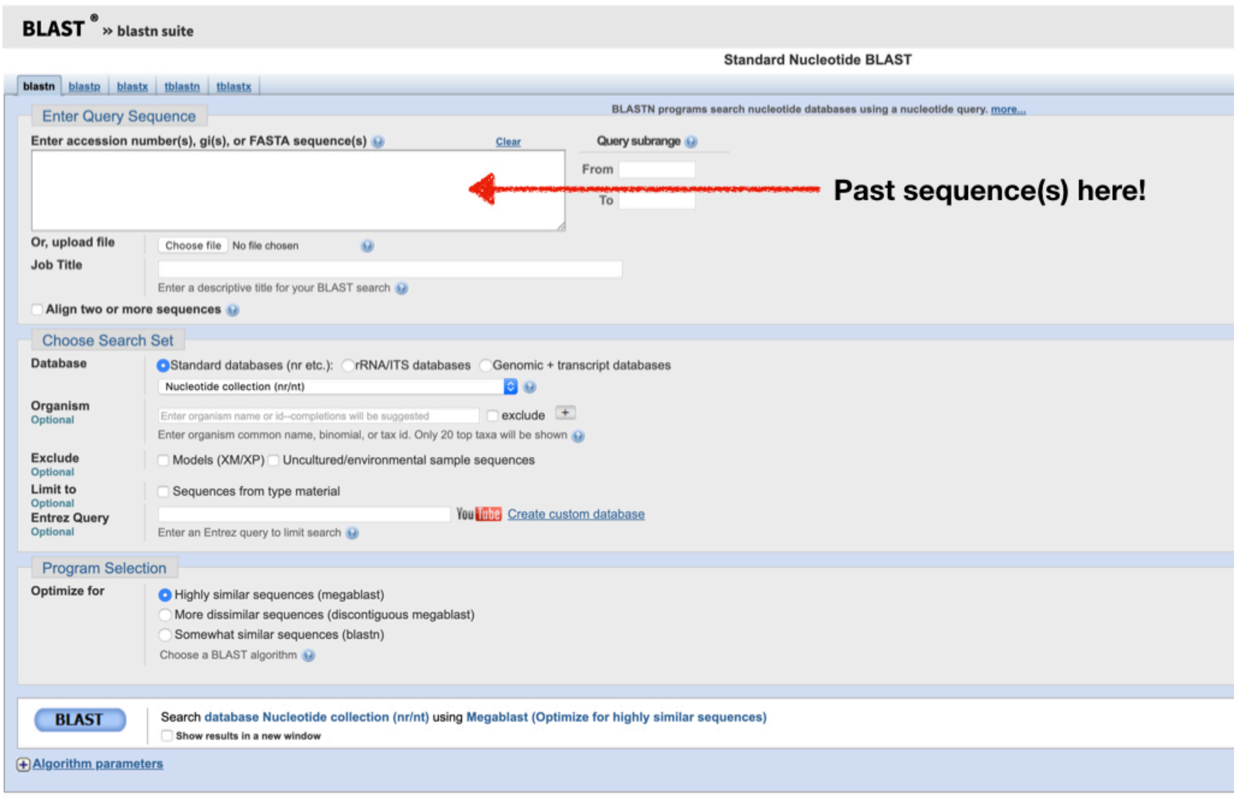
Click on **Nucleotide BLAST** (also known as Blastn) on the homepage:


Basic Local Alignment Search Tool
BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance.
Learn more
NEWS
A new version IgBLAST (1.15.0) is here.
We've released a new version of IgBLAST with four new improvements.
Wed, 22 Jan 2020 16:00:00 EST
More BLAST news...
Click here!
Web BLAST
Nucleotide BLAST
nucleotide ▶ nucleotide
blastx
translated nucleotide ▶ protein
tblastn
protein ▶ translated nucleotide
Protein BLAST
protein ▶ protein
BLAST Genomes
Enter organism common name, scientific name, or tax id
Search
Human
Mouse
Rat
Microbes

Open the contigs.fasta file from Spades

```
more spadesoutput/contigs.fasta
```

Copy & paste the contigs.fasta output into the **"Enter Query Sequence"** box, and then hit **BLAST** at the botton of the screen.


BLAST® » blastn suite

Standard Nucleotide BLAST

blastn | blastp | blastx | tblastn | tblastx

BLASTN programs search nucleotide databases using a nucleotide query. more...

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s) Clear

Query subrange From To

Past sequence(s) here!

Or, upload file Choose file No file chosen

Job Title

Enter a descriptive title for your BLAST search

Align two or more sequences

Choose Search Set

Database: [✓ Standard databases (nr etc.):] rRNA/ITS databases Genomic + transcript databases

Nucleotide collection (nr/nt)

Organism Optional: Enter organism name or id--completions will be suggested exclude

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown

Exclude Optional: Models (XM/XP) Uncultured/environmental sample sequences

Limit to Optional: Sequences from type material

Entrez Query Optional: Create custom database

Enter an Entrez query to limit search

Program Selection

Optimize for: [✓ Highly similar sequences (megablast)] More dissimilar sequences (discontiguous megablast) Somewhat similar sequences (blastn)

Choose a BLAST algorithm

BLAST Search database Nucleotide collection (nr/nt) using Megablast (Optimize for highly similar sequences)

Show results in a new window

Algorithm parameters

On the **BLAST** results page scroll down past the alignment graphic to see what sequences they have hit.

QUESTION :)

- Are these what you would expect?

Align contigs

This part of the tutorial is inspired by the documentation for Shiver, please read it here: https://github.com/ChrisHIV/shiver/blob/master/info/ShiverManual.pdf

Shiver is a tool for mapping paired-end short reads (like Illumina) to a custom sequence obtained with denovo assembly, in order to minimise the biased loss of information that occurs when you map to a reference that differs from the sample. From the mapped reads, base frequencies are quantified, and a consensus sequence is called. shiver was designed for HIV but can be used for other viruses.

The first step is to run the denovo assembly step as we did before with Spades or IVA

Let's make a folder:

```
mkdir shiver_results
cd shiver_results
```

Processing a sample requires two commands. The first one is:

```
/home/training/software/shiver/shiver_align_contigs.sh  \
/home/training/software/shiver/MyInitDir  \
/home/training/software/shiver/config.sh \
/home/training/denovo/spadesoutput/contigs.fasta PG15-BW001347
```

This command will produce:

- PG15-BW001347.blast detailing blast hits of your contigs to those existing reference supplied
- PG15-BW001347*raw*wRefs.fasta: an alignment of the HIV contigs (i.e. those with Blast hits) to your input existing reference genomes. In this alignment nothing has been done to the contigs (hence the name raw)
- PG15-BW001347*cut*wRefs.fasta: an alignment with contigs after correction (ie. trimming, cutting into pieces..)

!Important! We expect PG15-BW001347*cut*wRefs.fasta (if exists) to be a better alignment than PG15-BW001347*raw*wRefs.fasta, but both files should be inspected, the better one chosen, and edited if necessary. For simplicity here, we are just assuming PG15-BW001347*cut*wRefs.fasta is better

Let's map read back!

```
 /home/training/software/shiver/shiver_map_reads.sh  \
 /home/training/software/shiver/MyInitDir  \
 /home/training/software/shiver/config.sh  \
 /home/training/denovo/spadesoutput/contigs.fasta \
 PG15-BW001347 PG15-BW001347.blast PG15-BW001347_cut_wRefs.fasta \
 /home/training/Course_Data/PG15-BW001347_sub.R1.fastq.gz \
 /home/training/Course_Data/PG15-BW001347_sub.R2.fastq.gz

```

Output:

Output files include:

- **PG15-BW001347.bam**, **PG15-BW001347_ref.fasta**: the bam file of mapped reads, and the reference sequence to which they were mapped.
- **PG15-BW001347_BaseFreqs.csv**: the frequencies of A, C, G, T, '-' for a deletion, and N for unknown, at each position in the genome. SID*BaseFreqs*ForGlobalAln.csv is almost the same thing: it contains an extra column with a coordinate that can be used to compare different samples, at the cost of having to skip some rows for which this

  coordinate is ambiguous. Specifically, the extra coordinate is position in the global alignment (see section 8), and the rows that are skipped are positions where the reads had an insertion with respect to the mapping reference. Positions where the mapping reference has an insertion with respect to the global alignment (i.e. where the contigs had an insertion with respect to the existing references you gave shiver as input) are included at the appropriate position in this file, just with an undefined global coordinate. SID*BaseFreqs*WithHXB2.csv is the same thing as SID*BaseFreqs*ForGlobalAln.csv but using the coordinates of the HXB2 sequence instead of the coordinates of the global alignment (this is only produced if the config file variable GiveHXB2coords is left at its default value of true; you'll certainly want to change this to false if you're using shiver on non-HIV data).
- **PG15-BW001347*consensus*MinCov*15*30.fasta**: the pairwise alignment of the consensus genome and the reference used for mapping. In place of X and Y here you'll see the two coverage thresholds specified in config.sh: X is the minimum number of reads to call a base (if there are fewer than X reads we call ? instead of a base), Y the minimum number to use upper case for the base (to signal increased confidence). The mapping reference is included with the consensus here to give context to any ? characters.
- **PG15-BW001347*consensus*MinCov*15*30*ForGlobalAln.fasta, PG15-BW001347*coords.csv**: these files are useful when multiple samples are processed
- **PG15-BW001347_InsertSizeCounts.csv**: the insert-size distribution.

Explore them with more/less/cat!

If you have time..

You can run other samples to practise, all data are in /home/training/Course_Data/. I suggest you use a file with the suffix _sub.R*.fastq.gz for time issue, but any fastq.gz would work!