Practical: Using LAST and MEGAN to get a quick view of a metagenome

Daniel Lundin

Linneaeus University

November 14, 2014

A GIT archive in your virtual machine

- You will use the virtual machine again
- This presentation is available in a Git archive:
  workshop/bils-shared-slides-and-practicals/practicals/last_megan.d/last-megan.pdf
- Open with e.g. `evince` (a pdf reader)
  - To update to most current version, and open from the command line:

```
$ cd workshop/bils-shared-slides-and-practicals/practicals/last_megan.d
$ git pull
$ evince last-megan.pdf
```

Read-based metagenome annotation

- Quick view at a metagenome
- Sufficient for some analyses
  - Especially for quantitative purposes
- Assembly required in other cases
  - Careful description of key organisms

The task

- Compare individual reads to a large database (NCBI Refseq)
- Import alignment into MEGAN, a graphical analysis tool
- Analyse, compare two samples
- Samples taken in the Baltic Sea
  - 80m depth: suboxic
  - 400m depth: anoxic

The data

- You get reduced data sets to have a chance of completing the assignment

```
$ cd courses/bils-shared-slides-and-practicals/practicals/last_megan.d/data
$ ll
```

- The samples are called `80m_round2` and `400m_round2`
- Each sample exists in four different sizes: 100, 1000, 10000 and 100000 sequences denoted by `d100` etc. in file names
- Two files for each combination of sample and size
  - read1: Forward
  - read2: Reverse

General instructions

- Try each step with at least one input file at the 10000 level (i.e. 10000 original sequences)
- If it takes too long (¿15 minutes): cancel and try a smaller datafile
- Keep track of (create a spreadsheet (e.g. Libre Office, installed in the VM)):
  - How long each step takes
  - What files were input and output in each step
  - Any parameter you set
  - How much data passes through the tool, e.g. how many sequences were lost in quality control, how many aligned to RefSeq etc.
- When you come to the MEGAN analysis you can work with larger files:
  - `80m_round2.d100000.sickle.refseq.rma`
  - `400m_round2.d100000.sickle.refseq.rma`

Pipeline

1. Quality trimming: Sickle
2. Convert fastq to fasta
3. Alignment to Refseq:
   - LAST
   - BLAST
   - (Compare performance)
4. Convert format maf to tabular BLAST
5. Import into MEGAN
6. Make comparison MEGAN data file
7. Analyse!

Quality-trimming: Sickle

- Trim off poor quality bases
- Sickle uses a sliding window (0.1 times the length of the read by default)
- One mode for single reads (`sickle se`), one for pairs (`sickle pe`)
- A command line for Illumina paired end data:

```
$ sickle pe -f sequences_1.fastq -r sequences_2.fastq -t sanger \
  -o sequences.sickle.fwd.fastq -p sequences.sickle.rev.fastq \
  -s sequences.sickle.unpaired.fastq
```

Aside: Using subprocesses to decompress/compress data

- Gzip can save a lot of space
- Especially if you only read/write directly from/to compressed files
- Pipe lines is one solution

```
$ gunzip -c large_file.fastq.gz| grep -c '^@HWI'
```

- Pipes can only handle one input and one output
- If you have more, subprocesses in parentheses is an option:

```
$ diff <(gunzip -c large_file_1.gz) <(gunzip -c large_file_2.gz) > >(gzip -c > large_output
```

Sickle 2: Subprocesses

- Sickle reads gzipped files, but doesn't write:

```
$ sickle pe -f sequences_1.fastq.gz -r sequences_2.fastq.gz -t sanger \
  -o >(gzip -c > sequences.sickle.fwd.fastq.gz) \
  -p >(gzip -c > sequences.sickle.rev.fastq.gz) \
  -s >(gzip -c > sequences.sickle.unpaired.fastq.gz)
```

Converting fastq files to fasta

- The LAST aligner doesn't read fastq files, but fasta
- The FastX toolkit contains the `fastq_to_fasta` tool
- Examples, uncompressed and compressed respectively:

```
$ fastq_to_fasta sequences.sickle.read1.fastq > sequences.sickle.read1.fna
$ gunzip -c sequences.sickle.read1.fastq | \
  fastq_to_fasta | \
  gzip -c > sequences.sickle.read1.fna.gz
```

Creating a LAST formated RefSeq database 1 – download RefSeq

You don't need to do this, just for later reference

- NCBI distributes BLAST formated database at: `ftp://ftp.ncbi.nih.gov/blast/db/`
- You can extract sequences to fasta and format for LAST
- To download the RefSeq:

```
$ cd directory_with_lots_of_space
$ wget "ftp://ftp.ncbi.nih.gov/blast/db/refseq_protein*tar.gz"
```

Creating a LAST formated RefSeq database 2 – extract sequences

You don't need to do this, just for later reference

- The BLAST utility command `fastacmd` extracts sequences from databases (assuming you're in the same directory as earlier):

```
$ fastacmd -d refseq_protein -D 1 -o refseq_protein.faa
```

Creating a LAST formated RefSeq database 3 – format the database

You don't need to do this, just for later reference

- The command to create a LAST formated database is `lastdb`
- Create a database named `refseq_protein.lastdb`:

```
$ lastdb -p -c refseq_protein.lastdb refseq_protein.faa
```

Running the LAST aligner

- LAST is around 10000 times faster than BLAST
- A lane of Illumina HiSeq data, around 100 million read pairs, can be aligned to RefSeq in less than 24h on a fast computer
- You can't distribute work over more than one computer
- To use a parallel computer, use the GNU parallel library
  - I will not show you, see the LAST manual: `http://last.cbrc.jp/` (especially the "Tutorial for smarties")
- An example, aligning sequences against the RefSeq installed in your VM:

```
$ gunzip -c 80m_round2.d10000.sickle.read2.fna.gz | \
  lastal -e120 -F15 /scratch/refseq_protein.lastdb | \
  gzip -c > 80m_round2.d10000.sickle.read2.refseq.maf.gz
```

Converting maf files to MEGAN import format

- The output from LAST is not readable by MEGAN, so it needs to be converted
- I have included a Makefile (see below) with a conversion recipe for the conversion
- Assuming a file `sequences.sickle.read1.maf.gz` you just need to:

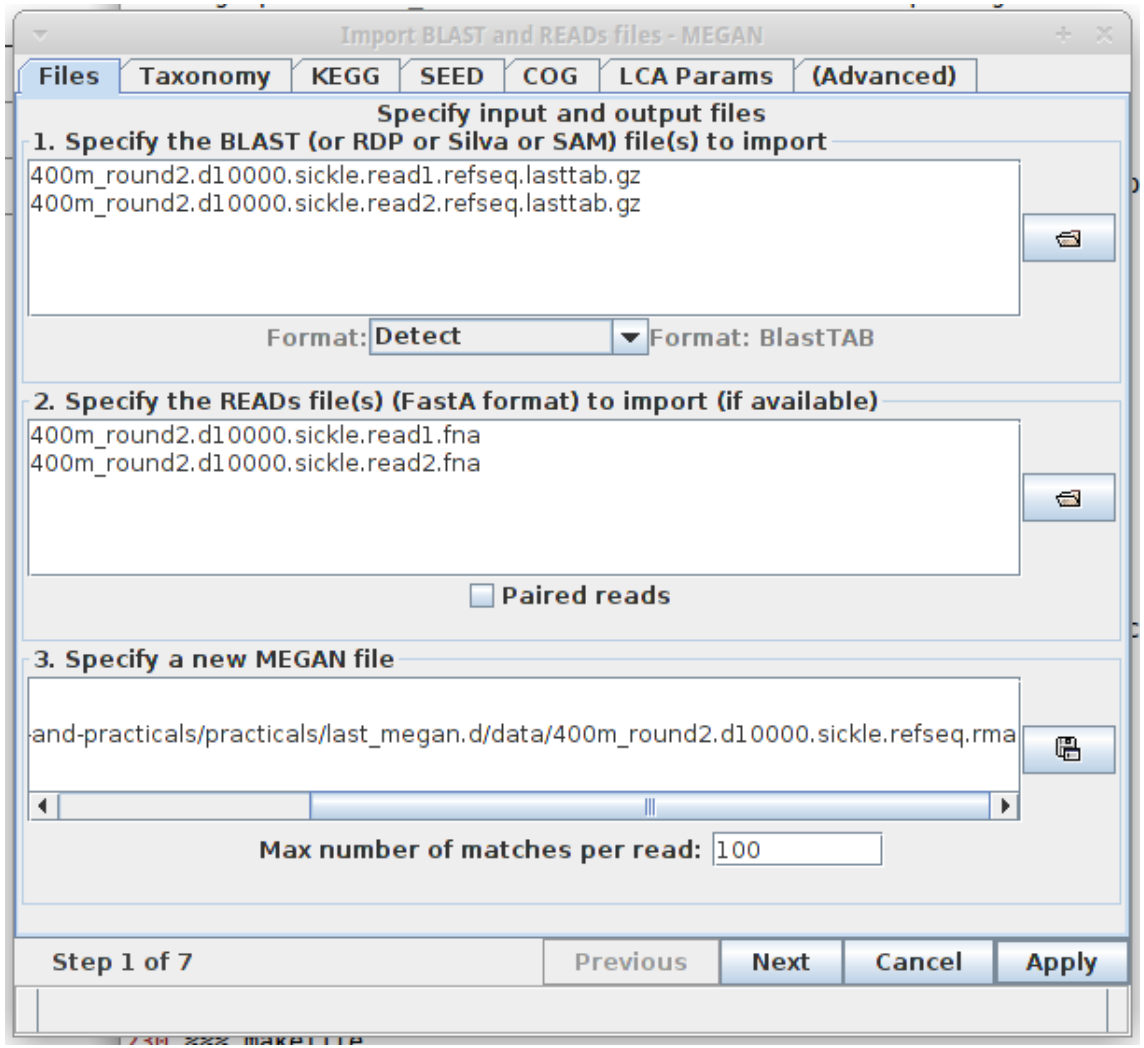
```
$ make sequences.sickle.read1.lasttab.gz
```

Importing into MEGAN 1 – Prerequisites

- Make sure you have created the .lasttab.gz files
- You also need fasta files in uncompressed format
- Start MEGAN from the menu or the command line
- When you're asked for a license, supply `/data/megan/MEGAN5-academic-license.txt`

Importing into MEGAN 2 – Running the import

- Choose *File*, *Import_from_BLAST*, and fill in values, making sure to supply both forward and reverse reads:



Importing into MEGAN 3 – Taxonomy and function

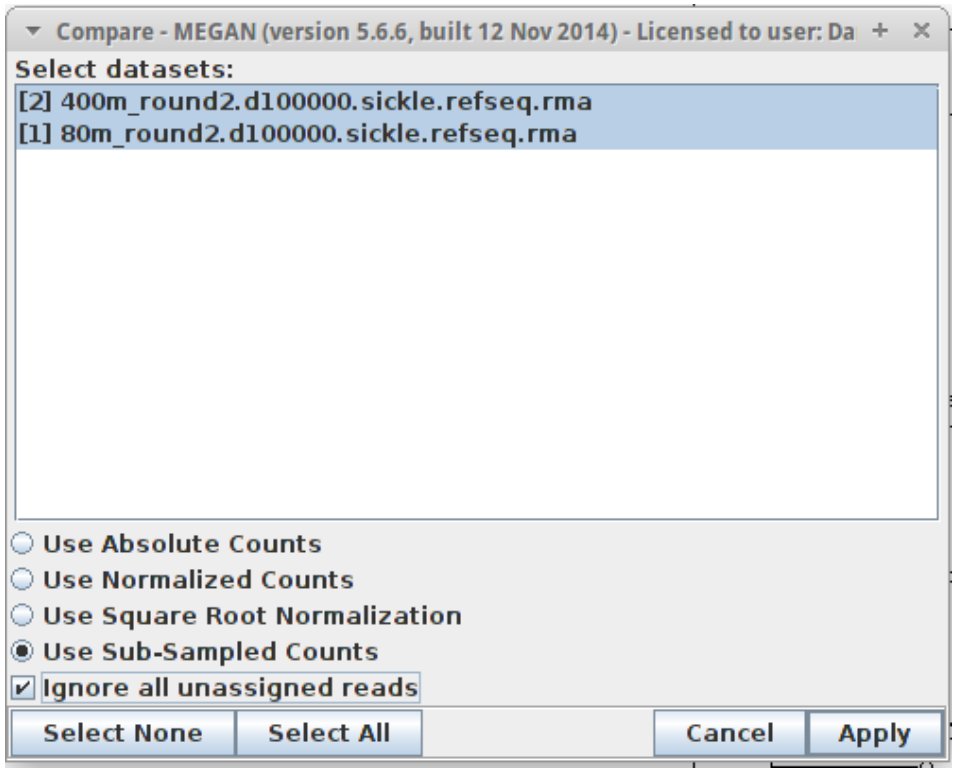
- Click *Next* or *Taxonomy and make sure you click _Load_GI_mapping_file*
- The GI mapping file can be found in `/data/megan/gi_taxid_prot.bin`
- Do similarly for SEED and KEGG
- Choose *Use_builtin_RefSeq_Map* for COG
- LCA params are quite good by their defaults
- Under *Advanced*, make sure you click *Load_all_reads_into_memory*
- Click *Apply*
- You can watch progress in a separate window

Importing into MEGAN 4 – open the larger files

- Now you can switch to the larger files I've created in `workshop/bils-shared-slides-and-practicals/practicals/last_megan.d/data`
- Open both the 400m and 80m d100000 files

Make a comparison

- Select all leaves (*Select*, *All_leaves*), uncollapse subtree (*Tree* menu item)
- Choose *Options*, *Compare* to create a comparison dataset:



- Play around and see who's there and what they can do!

Using Make (useful, NOT REQUIRED)

- Make is a tool to "cook a file" from other files
- Recipe: List of ingredients (dependency files) + instructions (commands)

```
sequences.maf.gz: sequences.faa refseq.lastdb
	lastal -e120 -F15 $(word 2,$^) < $< | gzip -c > $@
```

```
$ make sequences.maf.gz
```

- Excellent for automation:
  - Recursive, makes dependent files if necessary
  - Only remakes files when dependencies are newer than target
  - Can be parallelised simply

Make 2: pattern rules

- Pattern rules for generalisation:

```
%.maf.gz: %.faa refseq.lastdb
	lastal -e120 -F15 $(word 2,$^) < $< | gzip -c > $@
```

```
$ make sequences.maf.gz more_sequences.maf.gz
```

Make 3: A full example

Here's a full example of make targets from `.fastq.gz` files to finished `.maf.gz`

```
# This target creates all possible maf files (LAST alignment format), one for
# each existing .fastq.gz file.
all_mfas: $(subst .fastq.gz, .maf.gz, $(wildcard *.fastq.gz))

# Create a fasta formated file with sequences from a fastq formated, both
# gzipped.
%.fna.gz: %.fastq.gz
	gunzip -c $< | fastq_to_fasta | sed '/^>/s/$$/\\1/' | gzip -c > $@

# Run the LAST alignment against the refseq.lastdb database
%.maf.gz: %.faa refseq.lastdb
	lastal -e120 -F15 $(word 2,$^) < $< | gzip -c > $@
```

```
$ make all_mfas
```

Make 4: A Makefile in the data directory

In the data directory there's a Makefile to perform most of what you will do here:
/workshop/bils-shared-slides-and-practicals/practicals/last_megan.d/data