RNA-seq data analysis in Galaxy

Hands on workshop: Buidling a workflow based on the Tuxedo protocol

Instructor: Hailiang (Leon) Mei
Leiden University Medical Center, The Netherlands

Introduction In this workshop we will show you a typical analysis done by a bioinformatician working with RNA-seq data using Galaxy. This involves quality control, aligning raw sequencing data to a known reference genome, doing expression analysis and visualization using the UCSC genome browser. We will also demonstrate the support of building workflows within Galaxy.

Tools and datasets All tools used in these exercises can be downloaded from the Galaxy toolshed.

- FastQC for quality: `https://toolshed.g2.bx.psu.edu/view/devteam/fastqc`
- Fastq groomer: `https://toolshed.g2.bx.psu.edu/view/devteam/fastq_groomer`
- Fastq trimming: `https://toolshed.g2.bx.psu.edu/view/nikhil-joshi/sickle`
- picard package: `http://toolshed.g2.bx.psu.edu/view/devteam/picard`
- Tophat2 + its dependencies (bowtie, etc) : `https://toolshed.g2.bx.psu.edu/view/devteam/tophat2`
- cufflinks package: `https://toolshed.g2.bx.psu.edu/view/devteam/all_cufflinks_tool_suite`

Datasets used in this practical is test data and not full size files. This is to reduce the time needed to run each step and make this analysis possible within the time permitted. The data was retrieved from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37918`, a description of the study can be found here `https://www.ncbi.nlm.nih.gov/pubmed/23580553`.

Preparations

- Open a browser and go to the Galaxy server assigned to you.
- Register to gain access to data libraries and workflows.

Exercise 1: Single sample expression analysis The input data is a small selection of reads that should align mostly to a small region on the human genome. After alignment, you can do expression analysis and visualisation.

Import the following files from the "ECCB2014" data library:

- `ucsc_refseq_20140619.gtf`
- `miR-23b_1.fq`
- `miR-23b_2.fq`

First look at one of the FASTQ files. Each read is represented by four lines: a header, the read itself, a "+" and the quality scores.

Do some standard QC on the FASTQ files:

- Run *FastQC* on both FASTQ files.
- *Hint*: When selecting input files, you can choose multiple datasets and run in parallel.

When looking at the output of the QC steps, you will notice a lot of warnings and errors, they arise partially from the fact that we work with a very small dataset.

NGS specific questions:

- Are there any other reasons for these warnings?
- What is the total number of sequences?
- What is the quality encoding?

Use *Sickle* for trimming low quality parts of the reads.

NGS specific questions:

- What do you see when you look at the newly generated FASTQ files?
- If you run *FastQC* again, which metrics are improved?

Align the trimmed reads to the human reference genome build `hg19` with *Tophat*.

- *Hint*: The data type should be fastqsanger. You can use *Fastq groomer* for converting the fastq quality encoding.

Visualise the aligned reads (BAM file) with the UCSC genome browser. Go to an area of interest. Note that splice junctions are most likely in an area of interest.

NGS specific questions:

- Can you find evidence for alternative splicing in region `chr16:15696870-15745667`?
  - *Hint*: Change the visualisation from "dense" to "pack".
- Can you find mismatches in the alignment (or possibly even variants)?

NGS specific questions: How many reads were aligned?

- *Hint*: Run *SAMTools flagstat* on the aligned reads and check tophat alignment summary.

Visualize the bam files in UCSC genome browser.
NGS specific questions: What do you see in a region of interest?

- *Hint*: Change the visualisation of the BAM track to "squish".

Inspect the insertion size metrics with *Picard* tools.

NGS specific questions:

- Can you explain the truncation at the left of the histogram?
- How could this be improved?

Use *Cufflinks* for transcript assembly and abundance estimation. Use the reference genes as guide for the assembly.

NGS specific questions:

- What is the most abundant gene?
  - *Hint*: Use the filter and sort tools.
- What is the most abundant transcript? Visualise it in the genome browser.

Extract a workflow, create a new history and apply the workflow on control sample files from the "ECCB2014" data library:

- `ucsc_refseq_20140619.gtf`
- `miR-nc_1.fq`
- `miR-nc_2.fq`

Exercise 2: Differential expression analysis. Now we have analysed two samples, one treated- and one control. Now we can do differential expression analysis to figure out what the effect of the treatment was.

Create a new history. Import the following datasets from the "ECCB2014" data library:

- `Control run: accepted_hits`
- `Treatment run: accepted_hits`
- `Control run: assembled transcripts`
- `Treatmen run: assembled transcripts`
- `ucsc_refseq_20140619.gtf`

Merge the control and treated transcript assemblies with *Cuffmerge*. Use the refseq genes as reference annotation.

Run *Cuffdiff* on the merged transcripts file, the control- and treated BAM files, use the "blind" dispersion estimation method.

For now, we are interested in the gene differential expression testing dataset. Filter this list based on the status column.

NGS specific questions:

- Which gene is most affected?
- Is it up- or down regulated?