RNA-seq analysis
Practical workshop

Bilille - Plateforme de bioinformatique de Lille
UAR 2014 US 41 PLBS

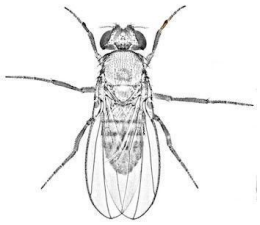
camille.marchet@univ-lille.fr, UMR 9189 CNRS (Centre de Recherche en Informatique, Signal et Automatique de Lille, CRIStAL), Université de Lille – FST

pierre.pericard@univ-lille.fr, Plateforme Bilille - UAR 2014 US 41 PLBS (Université de Lille, CNRS, Institut Pasteur de Lille, Inserm, Inria, CHU Lille)

Introduction

Connexion to the Galaxy interface

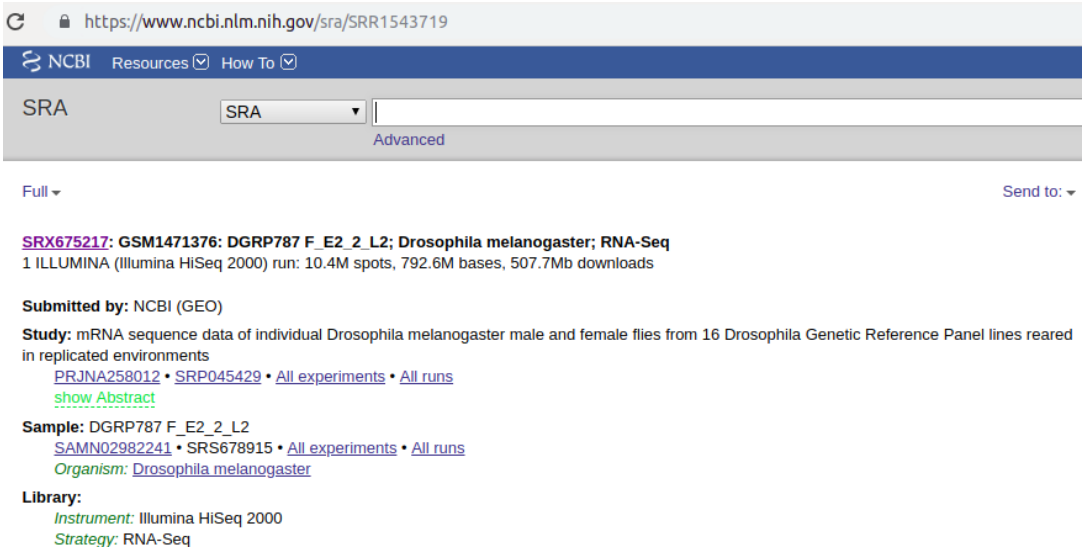
Datasets



We chose *D. melanogaster* for its relatively small genome (to scale for the time/memory requirements of our experiments), for its ability to produce alternative isoforms (as a Eukaryota), and because it is a well-known species: a reference genome is available.

This is where we got the data:

https://www.ncbi.nlm.nih.gov/sra/SRX675217



Write here the characteristics of the complete sequencing dataset:

- Run accession id:
- Sequencing technology:
- Library preparation:
- Single / Paired-end ?:
- Number of reads:

For this practical workshop, we built a reduced dataset (11421 SE reads):

1. 9 regions of interest from the 2L chromosome were selected
2. Reads that aligned onto those regions were extracted (= 7245 reads)
3. Reads with adapter sequences were added (+ 2176 reads)
4. rRNA reads were added (+ 2000 reads)

1. Cleaning - Pre-processing

This part will be done in the Galaxy environment

1.0. Datasets preparation

Import the shared history "`TP RNAseq bilille Initial datasets`" into a new history "`Initial datasets`".
Copy the following datasets from the "`Initial datasets`" history into a new history called "`Cleaning`":

- raw sequencing dataset: "`SRR1543719.sample.fq`"

1.1. Initial quality control

Step 1 - Run FastQC on the raw dataset

Q1 - How many reads are in this dataset **Q2 -** What sequencing length was used? **Q3 -** How would you qualify the sequencing quality of this run? **Q4 -** Are there any quality problems we should deal with ?

1.2 Quality trimming/filtering - Adapter sequences removal

Step 1 - Run Trimmomatic on the raw dataset

- Remove Illumina adapter sequences
- Quality trim reads with a 4bp sliding window and a quality threshold of 20
- Filter out reads with average quality < 25
- Filter out reads shorter than 50bp
- Output log messages

Q1 - Which adapter sequences should we use?

Step 2 - Rename the cleaned dataset "`SRR1543719.sample.trimmomatic.fq`"

Step 3 - Run FastQC on the cleaned dataset

Q2 - How many reads are left after this cleaning step ?

Q3 - Compare the quality graphs between the raw and cleaned datasets. You can use MultiQC to do that. What do you observe?

Q4 - What is the influence of the trimming/filtering parameters? Try re-running Trimmomatic after changing the quality thresholds (check with your neighbor so you can test a different set of parameters)

Q5 - What can you say about the read lengths?

Q6 - Are there any adapter sequences left?

1.3 rRNA sequences removal

Step 1 - Run SortMeRNA on the quality cleaned dataset

- use all the provided RFAM and Silva databases
- output the rejected reads in a file
- generate the statistics file

Q1 - Which file contains the non-rRNA reads?

Step 2 - Rename the non-rRNA reads file
"`SRR1543719.sample.trimmomatic.non_rrna.fq`"

Step 3 - Run FastQC on the non-rRNA reads file

Q2 - How many reads are left after rRNA removal?

Q3 - What changes in the FastQC metrics and graphs do you see after this step?

Q4 - Looking at SortMeRNA logs, what can you say about the taxonomic composition of this sample?

Bonus question - Was this a helpful step? Try to think ahead and imagine a scenario in which removing the rRNA sequences would greatly simplify or improve the downstream analysis.?

2. With-reference RNA-seq analysis

2.0. Datasets preparation

Create a new history called "`With reference RNA-seq`" and copy the non-rRNA reads file from the "`Cleaning`" history into this new history
Rename the non-rRNA reads file something like "`SRR1543719.sample.cleaned.fq`"

Copy the following datasets from the shared history "`Initial datasets`" into the "`With reference RNA-seq`" history:

- reference genome: "`D_melanogaster.BDGP6.22.dna.toplevel.fa`"
- genome annotation: "`D_melanogaster.BDGP6.22.96.gtf`"
- reference transcriptome: "`D_melanogaster.BDGP6.22.cdna.all.fa`"

2.1 Alignment against the reference genome

Step 1 - Run HISAT2 to align the cleaned reads against the reference genome

- print a machine-friendly alignment summary **to a file**

Q1 - How many reads were mapped?

Q2 - How many reads have multiple alignments?

Step 2 - Rename the BAM file "`SRR1543719.sample.cleaned.hisat2_genome.bam`"

Visualization - Let's visualize those alignments with IGV

Q3 - HISAT2 was designed to perform spliced alignments. Can you see some examples of spliced alignments?

Q4 - Can you differentiate between sequencing errors and SNPs?

Q5 - Can you see more than one transcript isoform for any gene? And using a Sashimi plot?

Bonus: What characteristics/features must you look for in a tool if you want to replace HISAT2 in this step? Can you recall an alternative tool? Is Bowtie2 a good fit?

2.2 Gene expression counting

Step 1 - Run featureCounts on the aligned reads file (BAM)

- Use the provided gene annotation file (GTF)
- Output has to be DESeq2/edgeR compatible

Step 2 - Rename the counts file something like
"`SRR1543719.sample.cleaned.hisat2_genome.gene_counts.tab`"

Q1 - How many reads were:
- used to count gene expression?
- excluded because of mapping quality?
- not aligned on any gene?
- aligned on multiple genes at once?

Q2 - How many genes are expressed at all?

Step 3 - Filter the counts file to keep only genes with count >= 10 (keep the headers)

Step 4 - Rename the filtered counts file
"`SRR1543719.sample.cleaned.hisat2_genome.gene_counts.filtered.tab`"

2.3 Transcriptome assembly with reference

Step 1 - Run StringTie on the aligned reads file

- in the banner on top of the form, select StringTie version 2.1.1 (click on the boxes icon)
- use the provided reference genome annotation file (you may have to use "browse datasets", click on the folder icon)
- output files for DESeq2/edgeR
- output gene abundance estimation file

Note! if no annotation file is available as input, check that the annotation file in your history has been correctly identified as GTF format. Otherwise, you should change its format.

Step 2 - Rename the assembled transcripts file
"`SRR1543719.sample.cleaned.stringtie.transcripts.gtf`"

Step 3 - Rename the gene and transcript count files
"`SRR1543719.sample.cleaned.stringtie.gene_counts.tab`" and
"`SRR1543719.sample.cleaned.stringtie.transcript_counts.tab`"

Q1 - How many transcripts were assembled by StringTie?

Q2 - How many genes do these transcripts represent?

Q3 - Where are these genes located? What do you think about it?

Q4 - How many genes are supported by at least 10 reads?

Main screen - Let's visualize StringTie assembled transcripts with IGV

Q5 - What can we tell looking at STRG.4?

Q6 - What can looking at STRG.1 tell us about how StringTie works?

Bonus - If you have time, re-run the StringTie assembly without the reference annotation file. This would reconstruct exons and transcript de-novo, using only the sequencing reads and the reference genome sequence.
Do we get as many assembled transcripts? How do they compare to the transcripts assembled using the annotation file?

2.4 Gene/Transcript expression estimation by pseudo-mapping

Step 1 - Run Salmon on the cleaned reads against the reference transcriptome

- provide the annotation file as a mapping of transcripts to genes

Step 2 - Rename the gene and transcript quantification files something like "`SRR1543719.sample.cleaned.salmon.gene_quantif.tab`" and "`SRR1543719.sample.cleaned.salmon.transcript_quantif.tab`"

Q1 - Why Salmon quantification files cannot be called "count" files?

Q2 - According to Salmon, how many genes are expressed at all?

Step 3 - Filter the gene quantification file to keep only genes with pseudo-count >= 10 (keep the headers)

Step 4 - Rename the filtered quantification file something like "`SRR1543719.sample.cleaned.salmon.gene_quantif.filtered.tab`"

Step 5 - Join the Salmon gene quantification file with the featureCounts gene count file

- join on the gene ID columns
- keep the headers
- keep lines that do not join or that are incomplete

- fill empty columns with “NA”

Step 6 - Rename the joined file something like “`SRR1543719.sample.cleaned.gene_quantif_matrix.tab`”

Q3 - What can you tell about the gene expression values estimated by mapping vs. pseudo-mapping?

Q4 - Why do you think Salmon identifies more genes expressed than the mapping pipeline?

Main screen - Visualise gene “FBgn0000055” and position “2L:20,846,256-20,944,010” from the annotation file in IGV

3. TP *de novo* transcriptome analysis

Goal: We will assemble the cleaned reads to obtain RNA transcripts and variants. We intend to obtain a reference transcriptome using assembly techniques, without relying on a reference genome (de novo). In our case, we study a sample of the reads. Therefore we will build a small part of the entire transcriptome and a short list of variants.

Tools: as mentioned in the course, we will use Trinity for full-length transcript assembly (https://github.com/trinityrnaseq/trinityrnaseq/wiki), and KISSPLICE for *de novo* variant calling (http://kissplice.prabi.fr/)

3.1 Assembly and contig visualization

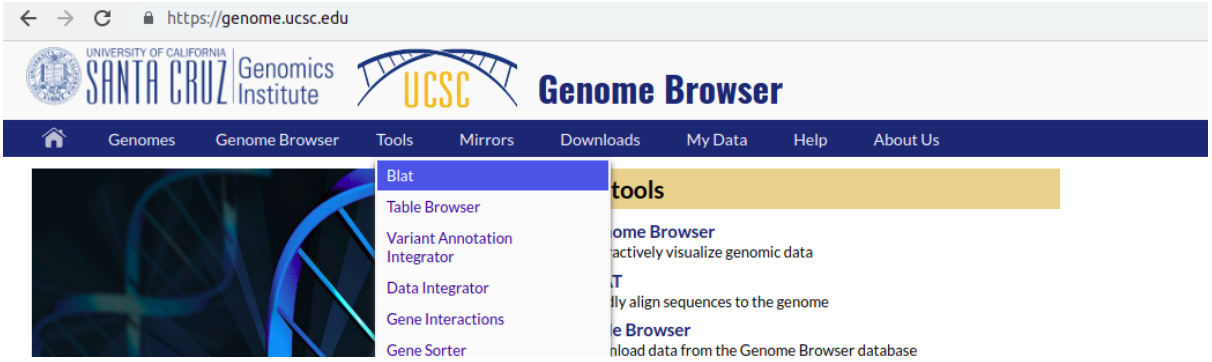
Q1 - Trinity assembly - Launch Trinity in Galaxy: which parameters should you set?
Notice that:

- reads are single-end
- the read bank is non-stranded

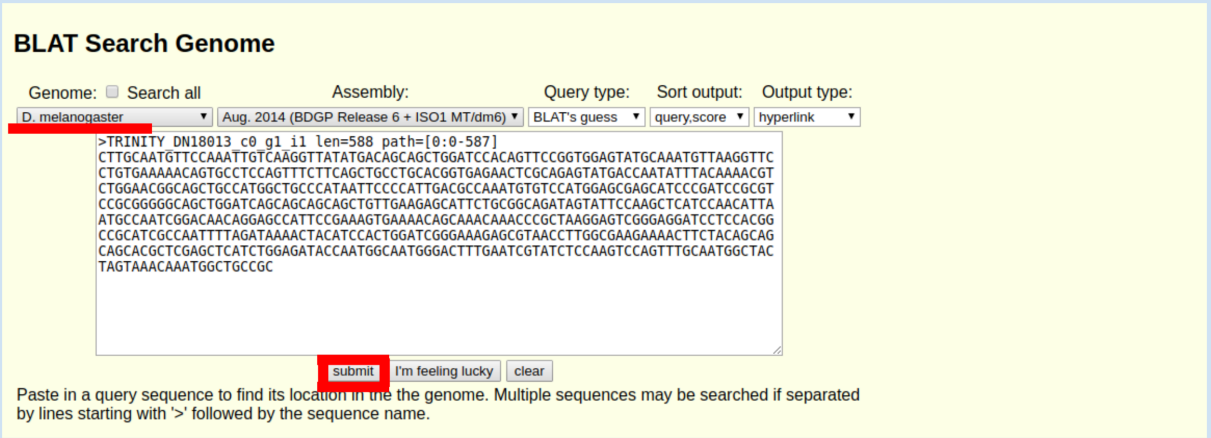
Q2 - Trinity assembly - When Trinity is finished, how many contigs are output?

To visualize Trinity contigs, we will use the BLAT tool from UCSC. Even though its index strategy differs, this tool is conceptually very close to the aligner BLAST. It will allow us to align the contigs to the reference genome and then display the alignments.
Here's a small how-to:

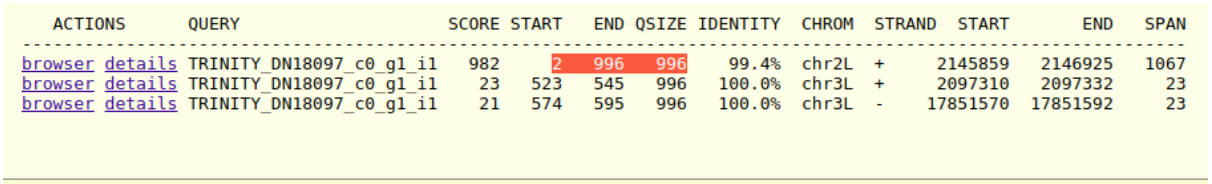
1. Copy the sequence you want to visualize.
2. Paste it in the BLAT field at UCSC's website (https://genome.ucsc.edu in your web browser), then check the parameters (genome)



3. Compute the contig to genome alignment using `Submit`



4. Select the most suited alignment if there are several of them. To do so, not only you will check the `IDENTITY`, but also `START, END, QSIZE` results



ACTIONS	QUERY	SCORE	START	END	QSIZE	IDENTITY	CHROM	STRAND	START	END	SPAN
browser details	TRINITY_DN18097_c0_g1_i1	982	2	996	996	99.4%	chr2L	+	2145859	2146925	1067
browser details	TRINITY_DN18097_c0_g1_i1	23	523	545	996	100.0%	chr3L	+	2097310	2097332	23
browser details	TRINITY_DN18097_c0_g1_i1	21	574	595	996	100.0%	chr3L	-	17851570	17851592	23

Q3 - Contig Visualization - What `START, END, QSIZE` columns stand for in the BLAT result? Why is identity result not always sufficient?

Q4 - Contig Visualization - How to distinguish exons and introns displays? Recognize your query, versus the annotated transcripts.

Q5 - Contig Visualization - Have a look at some contigs using BLAT.
Observe the following properties:

- On which chromosome do they align?
- Are they fully aligned?
- Do you notice multiple mapping (several loci)?
- How many exons are included in your contigs?
- Is your contig appearing in annotations?

Q6 (bonus if you have time) Trinity quantification per transcript - Use the tool to map the reads back to the Trinity assembly. Which information is obtained from the output?

Q7 (bonus if you have time) Trinity workflow for differential analysis - Let's say we intend to observe differential expression of a gene in drosophila. We want to verify a higher expression in a specific biological condition, C. We defined an experimental protocol composed of two individuals in the biological condition, C0 and C1, and two controls, T0 and T1
Propose a Galaxy workflow to:

- obtain contigs
- estimate contigs expression
- obtain the count table that will be used in the next session (statistics for sequencing data)

Q8 (bonus if you have time) - Can you think about reasons why a *de novo* transcriptome assembly would end up very fragmented?

To go further with assembly...

- we would rely on TransRate or RNAQuast to proof-check our assembly results (see course)
- we would annotate our contigs (for instance with BUSCO)

3.2 Local assembly for variant calling in RNA-seq data

Q9 - Kissplice - Launch the KISSPLICE tool in Galaxy.

Q10 - Kissplice - When KISSPLICE is finished, determine how many alternative variants were found in the data.
Which parameters can you change to increase KISSPLICE's sensitivity? Give it a try.
Has the number of alternative splicing changed?

Q11 - Kissplice - Describe the 4-line KISSPLICE variant format

Q12 - Variant visualization - Pick the variants and perform the same previous operations to display them with UCSC's BLAT (you can also choose some variants from the shared KISSPLICE output from the full dataset that we pre-computed for you)

- Can you notice differences with Trinity's sequences?

Q13 - Kissplice post-processing - Unfortunately, not everything is on Galaxy… Which tool should we use if we wanted to find:

- more information on the alternative variants (exon skipping? alternative exon start/end? ...) we found based on the reference

- the differential presence of isoforms in the data

You can refer to http://kissplice.prabi.fr/tools/

To go further with KISSPLICE…
visit http://kissplice.prabi.fr/training/ to reproduce a recent alternative splicing paper experiment, using command line.