Part 1 - from raw data to differential expression

Aim:

(a) From raw data to gene-count table: You will use a NewTuxedo cascade of `scythe, sickle, hisat2, samtools, stringtie` and `prepDE.py` software to convert raw data in form of fastq files into gene-based and transcript-based count tables needed for downstream Differential expression analysis.

(b) Differential expression: Here you will run the Differential Expression analysis using DESeq2 R package using the count tables generated under aim (a) above.

Rules for aim (a)

- You will NOT copy the raw data fastq files into you area on HPC. Instead, you will make soft links in your working area on HPC.
- You will use the University's HPC for this task and you have already rehearsed all the steps when working with *D.melanogaster* RNAseq samples.
- You will finally perform the HPC calculations **in one step** that means by **submitting a single job** to HPC
- *Read pre-processing* - You will use the cascade of `scythe` and `sickle` - you will need to work out the offset of the base-qualities and use quality flags accordingly.
- *New Tuxedo* - Here you will rely entirely on the GTF annotation provided, and therefore you will use the NewTuxedo in "non-discovery mode". The RNA-samples were prepared using Illumina "stranded" protocol (Lecture 1) - this is important for a proper selection of RNA strandedness flags in hisat2 and stringtie. Don't forget to remove trimmed fastq files after hisat execution as well as both SAM and unsorted BAM files as soon as they were converted to sorted BAM.
- *Converting stringtie generated GTFs to count table* - Run `prepDE.py` the same way as you did for the practical to generate `gene_count_matrix.csv` and `transcript_count_matrix.csv`

Rules for aim (b)

- Transfer both gene and transcript count matrices to your local computer
- Generate experiment design table and store in a csv file.
- Generate a **`dds`** object for gene counts and another for transcript counts.
- Make dispersion plots for both objects and compare
- Make rlog-based PCA plots for both objects and compare

The remaining part below requires **`dds`** object for gene counts only:

- Make "SD versus mean" plots using meanSdPlot for log(normalised counts) and rlog (*please read about `meanSdPlot` function in DESeq2 manual*)
- Perform differential expression for all contrasts
- Generate MA (MvA) plots for standard NULL hypothesis (LFC=0) and NULL hypothesis of LFC<1
- Generate MA (MvA) plots for standard NULL hypothesis (LFC=0) for shrunken log2 fold-changes (*here you will have to first generate DE results using log fold-change shrinkage, please read about `lfcShrink` function in the DESeq2 manual*)

Deliverables (proof of work) - a single compressed archive with a prefix **part1** should contain the following:

Aim (a)

- the final SINGLE bash scripts for the job submitted to HPC.
- the **error** and **output** files from your final HPC execution
- the final `gene_count_matrix.csv` table

Aim (b)

- single R-code file for all work performed under aim (b)
- csv file containing experiment design table

Deliverables (report)
Aim (a)
- present and discuss sample-specific reads' statistics for raw, preprocessed and aligned data

Aim (b)
- Present dispersion plots and discuss the effect of dispersion shrinkage for both gene- and transcript-level data
- Present and discuss the PCA plot for gene-level data only.
- Investigate the effect of rlog transformation on the variance by comparing plots generated with `meanSdPlot` function for two gene count transformations namely log2(normalised count) and rlog - which of them is more *homoskedastic*
- discuss the meaning of NULL hypotheses of LFC=0 and LFC<1 and illustrate the discussion with associated MA plots and recorded numbers of significant genes.
- Compare MA plots for standard NULL hypothesis generated for shrunk and unshrunk log2 fold-changes and discuss the differences as well as the reason for the log2 fold-change shrinking.

For those who feel they need more challenge:
The deliveries related to running DESeq2 [(aim (b)], namely the R-code and the written report can be bundled together into a single R-markdown file, where chunks of R-code for tasks listed under "Rules for aim (b)" are interspersed with the text related the requirements specified under "Deliverables (report), Aim (b)". You will submit the .Rmd file together with its .html version obtained upon rendering. It will require an additional unsupervised learning about R-markdown (*start with creating a new R-markdown file in R-studio rendering to html with a Knit button*).

Marking:
- *Report:* You will be marked on this section for clarity of presentations of the results as well as for the quality of discussion.
- *Proof of work:* You will be equally marked in this section for your codes (how functional, neat and clear they are) as well as for the correctness of the results.

RESOURCES
Software
- All the programs required for HPC work are already installed on HPC in the directory
`/export/projects/polyomics/App/`
and you have already installed soft links in your ~/bin/ and the latter is included in your $PATH.
- The DESeq2 Bioconductor module will have been installed on your virtual machine by the time you start working on part b. If not go to Practical-6 instructions.

Raw data files
The raw data in form of fastq files are stored in the directory:
`/export/projects/polyomics/buzz/biostuds`
`s1.c2.fq` –> group "A" replicate 1
`s2.c2.fq` –> group "A" replicate 2
`s3.c2.fq` –> group "A" replicate 3
`s4.c2.fq` –> group "A" replicate 4
`s5.c2.fq` –> group "B" replicate 1
`s6.c2.fq` –> group "B" replicate 2
`s7.c2.fq` –> group "B" replicate 3
`s8.c2.fq` –> group "B" replicate 4
`s9.c2.fq` –> group "C" replicate 1
`s10.c2.fq` –> group "C" replicate 2
`s11.c2.fq` –> group "C" replicate 3
`s12.c2.fq` –> group "C" replicate 4

These data sets contain sequencing reads generated with single-end sequencing using NextSeq500 sequencer. The reads are preselected so that they originate from chromosome 2 of mouse genome. You will have to identify the offset of the base qualities.

Reference genome
The reference genome in form of fasta file represents chromosome 2 of mm10 mouse genome:

Reference hisat2 indexes
`/export/projects/polyomics/Genome/Mus_musculus/mm10/Hisat2Index/chr2`

Reference transcriptome annotation
`/export/projects/polyomics/Genome/Mus_musculus/mm10/annotations/chr2.gtf`

Illumina adapter sequences
`/export/projects/polyomics/biostuds/data/illumina_adapter.fa`