BPSM Assignment 1
Due 1400, Mon 25 October 2021

Background to the technology
RNA sequencing (RNAseq), as a generic technique, enables us to assess the levels of gene transcripts (usually mRNA molecules) in a group of samples, perhaps to see how gene expression levels change over time, or, perhaps to answer a question like: "what happens when drug X is administered to cultured cells: how do the cells respond at the gene level, and what role do those genes play?".

Background to the biology
Trypanosoma brucei spp (comprising *Trypanosoma brucei brucei* and the human infective forms *T. b. rhodesiense* and *T. b. gambiense*) are eukaryotic protozoan parasites responsible for African sleeping sickness in 36 countries of sub-Saharan Africa, which are the poorest developing countries worldwide.

Trypanosoma congolense on the other hand, despite being closely related, does not infect humans, yet does infect cattle and game animals (where it causes the disease "nagana") in much the same way as the *Trypanosoma brucei* complex species do. An overview of relevant literature is available at this link.

The trypanosome is transmitted between mammalian hosts by the tsetse fly, *Glossina* spp, in which it initially establishes in the midgut after a bloodmeal but then migrates to the salivary glands in preparation for transmission to a new mammalian host. In mammals, the parasite survives free in the bloodstream, being able to evade antibody responses through antigenic variation. As the disease is prevalent in some of the poorer regions of the world, it can have a disproportionately negative effect on the welfare and livelihoods of local farmers.

We are fortunate in that Trypanosomes can be cultured *in vitro*, there are annotated genome sequences for many of the species (and field isolates) and genome manipulation tools have been developed that enable us to ask very focussed questions about individual gene function in the organism, typically using "knock-out" or "knock-down" technologies. An example of the latter is RNA interference (RNAi), in which expression of anti-sense transcripts hybridise/align to form double-stranded RNA molecules that can block gene expression and/or translation to protein. This is achieved by the use of an inducible

synthetic RNAi construct that has been transfected into the parasite; addition of the drug tetra-cycline activates the construct to transcribe the relevant interfering RNA molecule (see https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2794767/ and https://pubmed.ncbi.nlm.nih.gov/16963636/ for more detailed information about the use of RNAi in Trypanosomes).

Your task

You have been asked to have a look at some RNAseq data that have been generated from a *Trypanosoma congolense* RNAi experiment in the IL3000 laboratory strain of *T. congolense* that knocks down a specific gene. As the gene encodes an enzyme that is critical for energy metabolism, the hypothesis is that loss of function of this enzyme might illuminate other pathways that the parasite might activate to circumvent loss of the "default" pathway.

The experimental set up is as follows, and involved a time course, with samples taken at T=0h, T=24h and T=48h. Some of the samples are un-induced, while others have been treated with tetra-cycline to induce gene expression from the RNAi construct. There are three different sample types:

1. Wild type (WT) cultures are *Trypanosoma congolense* samples without the RNAi construct; there are three replicates of most conditions.

2. Clone1 is a *Trypanosoma congolense* cell line that has the RNAi construct in it; there are three replicates of each condition.

3. Clone2 is another *Trypanosoma congolense* cell line that has the RNAi construct in it; there are three replicates of each condition.

RNAseq was performed, and paired-end data generated in gzip compressed fastq format. You have been given access to 100,000 reads from each of the 45 samples.

The research lab is particularly interested in whether there are any differences in gene expression levels between the various sample groups. There are several possible "group-wise" comparisons that could be considered:

- over time relative to WT controls
- over time relative to uninduced controls
- over time between clones
- at specific timepoints
- etc., there are others!

There **might** be another dataset from another RNAi construct coming, so the analysis might need to be done all over again with the bigger dataset, but we don't know for sure yet. If there are more raw sequence data, they will get put in the same directory, and the sample details file updated as necessary, so your script will need to be able to deal with this eventuality.

There are online tools that could be used (at least in part) to process and analyse this kind of data (e.g. https://usegalaxy.eu/), but in this instance, we are going to do all the processing from scratch using the command-line interface (i.e. in a terminal) on our MSc course server: the goal for this assignment is to write a pipeline programme **in bash and/or awk** that, when executed in a Linux terminal on our MSc server, will process the RNAseq data on our server.

This should be all your own work, and will be Turnitin scanned for plagiarism; by all means chat with your friends about things, but the code etc should be written by you!

Please do **NOT** try to write a pipeline using Python, R, Nextflow, Snakemake etc.

During the development of your pipeline code, you should use git for version control as detailed in BPSM Lecture 4. Making your code available as a **PUBLIC** git repository in GitHub is a major part of this Assignment (it is worth 50% of the marks), so do not neglect this component...! Don't forget that you can check your GitHub account using a browser to see if it is all OK.

The paired-end RNAseq sequence data are provided in the directory `/localdisk/data/BPSM/AY21/fastq`

The sample details are also there, in a file called `100k.fqfiles`.

The pipeline should have the following minimum overall design components/modules:

1. perform a quality check on the paired-end raw sequence data (which are in gzip compressed fastq format; https://support.illumina.com/bulletins/2016/04/fastq-files-explained.html) using the installed programme `fastqc`

2. assess the numbers and quality of the raw sequence data based on the output of `fastqc`

3. align the read pairs to the *Trypanosoma congolense* genome using installed programme `bowtie2` **OR** `hisat2`, converting the output to indexed "bam" format with `samtools`; the *Trypanosoma congolense* IL3000 genome sequence is provided in fasta format in the directory `/localdisk/data/BPSM/AY21/Tcongo_genome/`

4. generate counts data: the number of reads that align to the regions of the genome that code for genes; this is to be done using the installed programme `bedtools` and the `TriTrypDB-46_TcongolenseIL3000_2019.bed` "bedfile" that contains the information about the gene locations in the genome that was assembled and annotated in 2019. The gene names are in the **4th** column of the "bedfile". The bedfile is provided in the directory `/localdisk/data/BPSM/AY21/`; for the purposes of this analysis, you should assume, **incorrectly**, that all genes have no introns.

5. generate plain text tab-delimited output files that give the statistical mean (average) of the counts per gene (i.e. expression levels) for each group; as the gene names are pretty uninformative to a biologist, the gene descriptions (provided in the bed file) should also be included.

6. use the mean expression levels to generate "fold change" data for the "group-wise" comparisons; these data are indicative only, as you are not being asked to do statistical modelling, but should be output such that the fold-changes are in decreasing order; as the gene names are pretty uninformative to a biologist, the gene descriptions (provided in the bed file) should also be included.

Considerations

1. don't panic: it isn't as bad as it first sounds, honestly!

2. **DO NOT** immediately try to start coding! This is a pipeline made up of various component "bits": try drawing it out as a flowchart first on a piece of paper, so that you can clearly understand what the inputs and outputs might be for each "bit" that gets "stuff" done to it. Remember my favourite picture?! This is time very well spent...

3. most programmes have many parameters/flags that can be set by the user: the defaults are **not** always the best, so see what is available: it might make it run faster/better

4. most programmes have a "-h" or "--help" for help

5. try to make it so the user doesn't actually have to do much, other than execute the programme

6. the "genome sequence" actually comprises the individual chromosome sequences/scaffolds, and will need to be made into an appropriate database before it can be used with `bowtie2` or `hisat2`

7. normally, when you get raw sequence data, you'd get **lots** more than what you have been given for the assignment, and each sample wouldn't have the same number of read-pairs...

8. Google is your friend...

9. the demonstrator and I, we are your friends too, but we really can't tell you the answers, no matter how politely you ask, sorry...!

10. doing this Assignment should use some/most/all/more of the things we cover in the Unix/awk/scripting/git parts of the course, so that includes variables, loops, and so on. The pipeline programme/code will be run by me on the bioinformatics server, not in your home space, and not your laptop, so think about whether things/files/whatever are going to be accessible to the person running the code...

What should be submitted for this assignment

There are two things that need to be completed for this Assignment.

1. **50% of the marks**: a **PUBLIC** GitHub repository called Assignment1 containing a pass-worded zip file called B*xxxxxx*-2021.Assignment1.zip, as detailed in the BPSM git and GitHub lecture. The zip file should **only** contain the pipeline programme/code for me to run on our server, as well as the full git log of this coding project (in a `all_the_things_I_did` file). The pipeline programme/code itself should contain lots of "comments" so than anyone looking at the code knows what each bit is doing

2. **50% of the marks**: a PDF file called B*xxxxxx*-2021.Assignment1.pdf, submitted via Learn Dropbox, that:
    - gives the link for your **PUBLIC** GitHub Assignment1 repository (e.g. https://github.com/*Bxxxxxx*-2021/Assignment1)
    - gives the **zipfile** password to open the B*xxxxxx*-2021.Assignment1.zip file ...!
    - has an overview/flowchart that describes how the pipeline processes the data
    - briefly indicates why you have chosen the programme parameters/flags that you have for each step
    - indicates any things the user will have to do to make things work (hopefully very few!)
    - highlights any difficulties, if any, that you have come across
    - indicates any additional/alternative features that you think might be beneficial to include

What I'll be looking for (in no particular order)

1. successful and appropriate use of git and GitHub: if I can't clone your repository, or can't open the passworded zip file because

you have given an incorrect password (it is the password for the zip file that you will need to provide!), **you will get no marks for the code section**. Full instructions on how to do the push to GitHub for the Assignment have been given already, so do please read and try them out soon! Test things out before the deadline: check you can do what is required, and see if one of you colleagues can clone your repository: if they can, it's public, and that means I can too!

2. a pipeline programme/code that works and produces the requested output in the correct format: note that it must be your own work (but by all means discuss things with your colleagues)

3. comments in the code so that I can see what it does

4. any additional "flexibility" or features that have been added/suggested that indicate to me that you understand what is being done/not done AND why your pipeline helps biology (this is **BIO**informatics after all!)

What I would rather not see...

1. (please) keep the PDF content "focussed"! It is quality, not quantity, that is more important ...

2. the pipeline itself can actually be written using quite a small number of (carefully constructed) commands, so if your programme is hundreds and hundreds of lines, you should probably check it is doing the "right" thing!

3. GitHub has a size limit of ~100Mb, so don't even think about including the raw data in your repository! We know where the raw data are/might be, accessible using the full path to the files!
