RNA-Seq & Differential Expression

Introduction

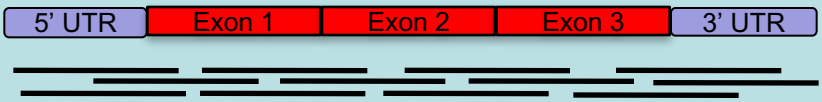
Understanding the genome is not simply about understanding which genes are there. Understanding when each gene is used helps us to find out how organisms develop and which genes are used in response to particular external stimuli. The first layer in understanding how the genome is used is the transcriptome. This is also the most accessible because like the genome the transcriptome is made of nucleic acids and can (indirectly) be sequenced using the same technology. Arguably the proteome is of greater relevance to understanding cellular biology however it is chemically heterogeneous making it much more difficult to assay.

Over the past decade or two microarray technology has been extensively applied to addressing the question of which genes are expressed when. Despite its success this technology is limited in that it requires prior knowledge of the gene sequences for an organism and has a limited dynamic range in detecting the level of expression, e.g. how many copies of a transcript are made. RNA sequencing technology, using for instance Illumina HiSeq machines, can sequence essentially all the genes which are transcribed and the results have a more linear relationship to the real number of transcripts generated in the cell.

The aim of differential expression analysis is to determine which genes are more or less expressed in different situations. We could ask, for instance, whether a pathogen uses its genome differently when exposed to stress, such as excessive heat or a drug. What happens if a gene gets silences (this exercise), and how does that change the transcription profile? Alternatively we could ask what genes make human livers different from human kidneys.

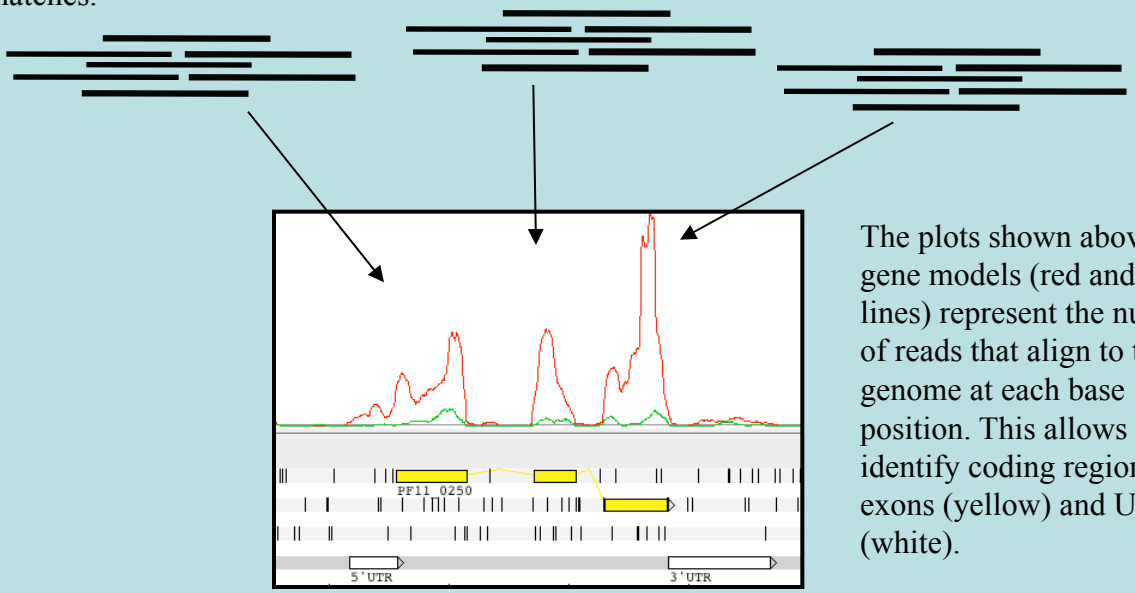
In close future we are going to be able to sequence routinely sequence single cells from organisms, understanding the heterogeneity of complex beings… but let’s focus on this exercise and malaria.

Here how RNA-Seq is working in a more visual way … imagine this transcript is present in the sample



Reads belonging to the transcript are produced by the sequencing process.

When the reads come out as raw data, there is no information about where they belong on the reference genome. What is more, all reads from several different transcripts come out together. An alignment algorithm finds where they belong in the reference genome based on similarity matches.



The plots shown above the gene models (red and green lines) represent the number of reads that align to the genome at each base position. This allows us to identify coding regions: exons (yellow) and UTRs (white).

The first RNA-Seq study in *Plasmodium* parasites focused on *P. falciparum* (Otto et. al. 2010). The aim was to show the viability of the RNA-Seq protocol in comparison to microarrays and also to improve the genome annotation and find alternative splicing. Recently a group used RNA-seq to identify differentially expressed genes, showing that parasites from vector transmitted infections are less virulence than serially blood passaged in the laboratory (Spence et al. 2013). In PlasmoDB there are more than 20 transcriptome for Plasmodium, and many more for the other parasites. Let's start to have a look at one:

Exercise

In this exercise you will need to determine the function of a gene that was knocked out. Your task is to find out the function of this gene, off you go!

0. Before we start

As we are going to use the HPC, you will need to login , start a session and copy the relevant files over.

At the end, we would appreciate if you could delete the files, latest two weeks after the end of the course.

As before, to work with the command line of Linux you will first need to login into the cluster and open a qsub session on the server. Next go to the directory you generated on /export/projects/bioinfo3/[your ID] and create the directory, Module_RNA-Seq and enter
-> ssh into headnode03.cent.gla.ac.uk

```
$ ~to16r/bioinfo/start.sh ## wait and see
$ ~to16r/bioinfo/setpath.sh

$ pwd # check if you are on bioinfo3, in the directory
```

If you haven't copied the data files, please do it now:

```
$ mkdir Module_RNA-Seq1
$ cd Module_RNA-Seq1
```

Now copy the files over: (use the tab key – faster and correct

```
$ cp –r $data/Module_RNA-Seq1 .
```

if you already copied the data, just go into the directory Module_RNA-Seq1 with the cd command.

A. Mapping with HiSat2

We have two conditions, wildtype (WT) and knock out (KO). First we are going to focus on the WT and map the reads against the reference genome of *Plasmodium berghei.* The tool we are using, HiSat2 is rather fast, but we did decrease the amount of reads to 30% for speed and space issues, compared to the original data.

In the directory of the module you can find the *Plasmodium* chromosome genome reference sequence (`PbANKA_v3.fasta`) as well as the two files of RNA-seq reads of the WT: `WT1_1.fastq.gz` and `WT1_2.fastq.gz`.

For the mapping, first an index of the reference (`PbANKA_v3.fasta`) must be constructed with hisat2-build. On the command syntax is: ***(Remember the names in italic are variables that you have to set!)***

```
$ hisat2-build reference_file index_name
```

As index name you can use for example **`PbANKA_v3`**.

This will generate the index need for the mapping. Most of the output you can ignore. Hisat2 is an improved version of tophat which is several times faster. To start the command you should type:

```
$ hisat2 -p 2 --max-intronlen 10000 -x index_name -1 reads_1 -2
reads_2 –S WT1.sam
```

The read file are called `WT1_1.fastq.gz WT1_2.fastq.gz` and the *index_name* you gave, see above.

The results of the mapping is in WT1.sam. The –p 2 option runs the mapping on four processors - why not using them if they are there?

Now you have to transform the WT1.sam file into a bam file and index it. Do you remember the command from yesterday? It was something with samtools, sort and index….

Check that your bam files exists by getting the mapping stats:

```
$ samtools flagstat WT1.bam
```

If that worked, we need to clean up, as those files are pretty huge. Now that we have the bam file, we don't need the sam file anymore. Let's have a look at the files

```
$ ls -lrS
```

You can see the large file called WT1.sam. You can delete it with

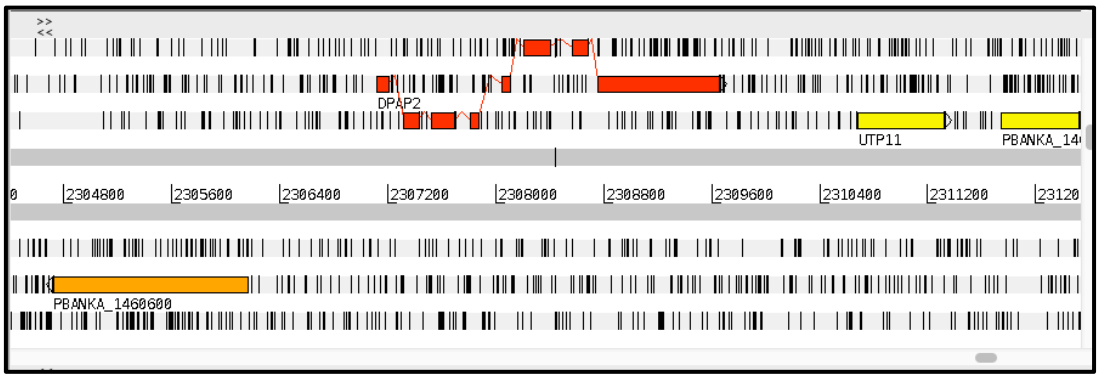
```
$ rm WT1.sam
```

B. Viewing the mapped reads in Artemis

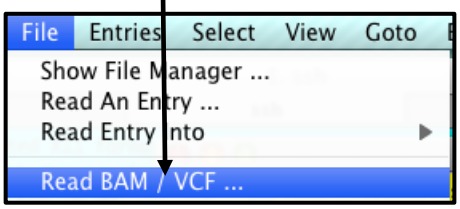
We will now examine the read mapping in Artemis using the BAM view feature. Be sure to be in the same directory as before. Open Artemis and load `PbANKA_14_v3.embl`. This file contains the sequence of chromosome 14 and the annotation. Here is also the gene that was knocked out.

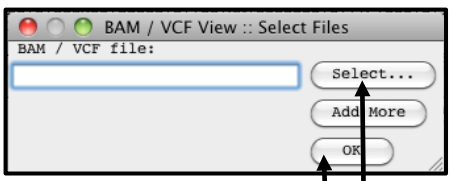
`$ art PbANKA_14_v3.embl &` ### to open Artemis

First go to the position 2308000 (Goto -> navigator).

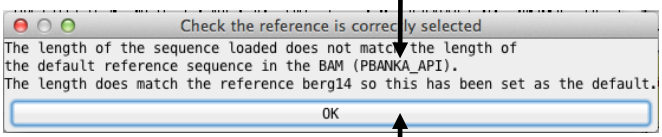


1. Now click on File -> "Reads BAM/ VCF…"



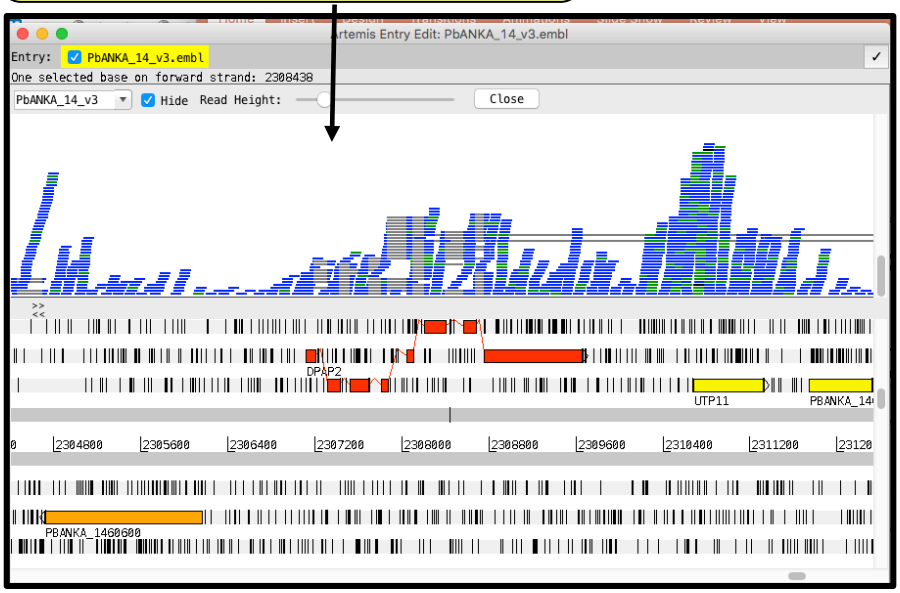


2. Select here the bam file you just generated (**WT1.bam**?) and then press ok.

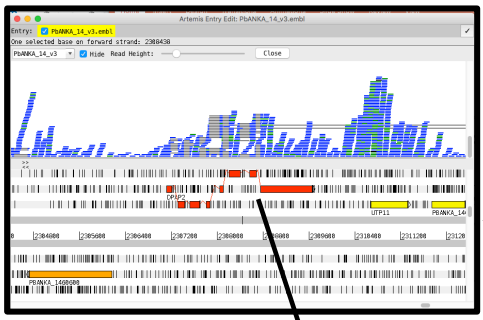


3. Confirm that the correct chromosome is chosen.

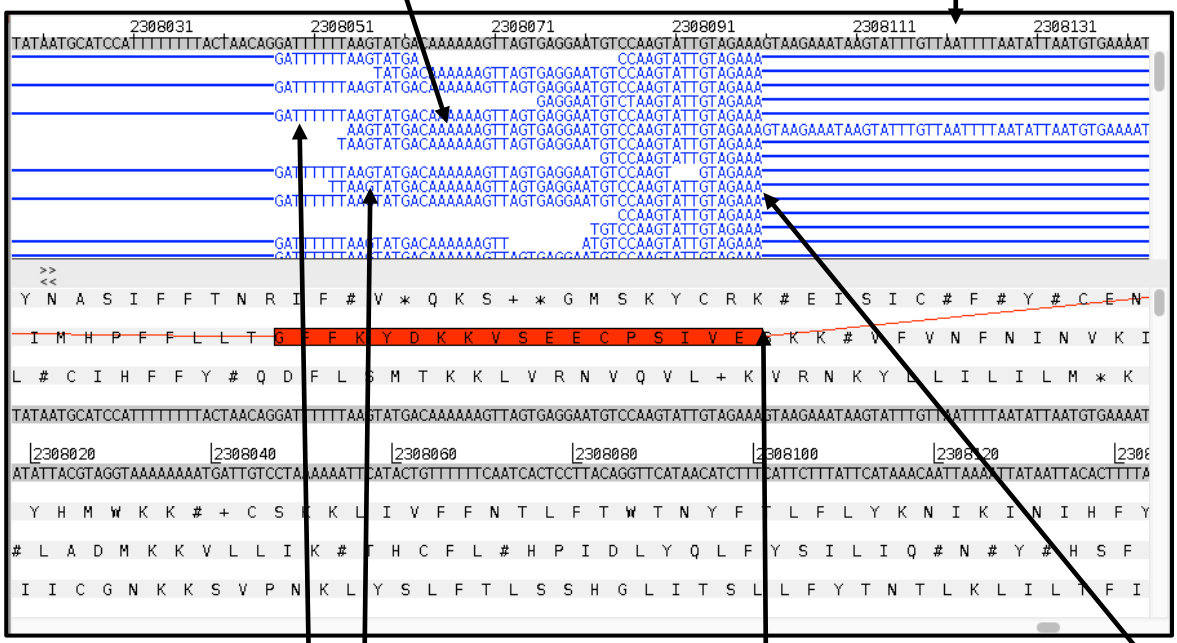
4. You should see following window… any idea what it means?



Congratulations, you have opened a Malaria chromosome with RNA-Seq mapping on it! The horizontal blue/green lines are sequencing reads, mapped against the reference. Let's have a look how the reads are "mapped" against the reference.



1. Zoom in as much as you can

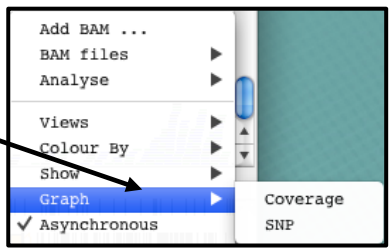


Each sequence represents a read. It is very similar to the genomic sequence at this regions, and therefore was mapped at this position. The abundance of reads represents the amount of mRNA of this gene.

Those reads are mapped over a splice site. The bar shows the intronic regions, which should be skipped in the reads. Can you see where the other parts of the reads are mapping?

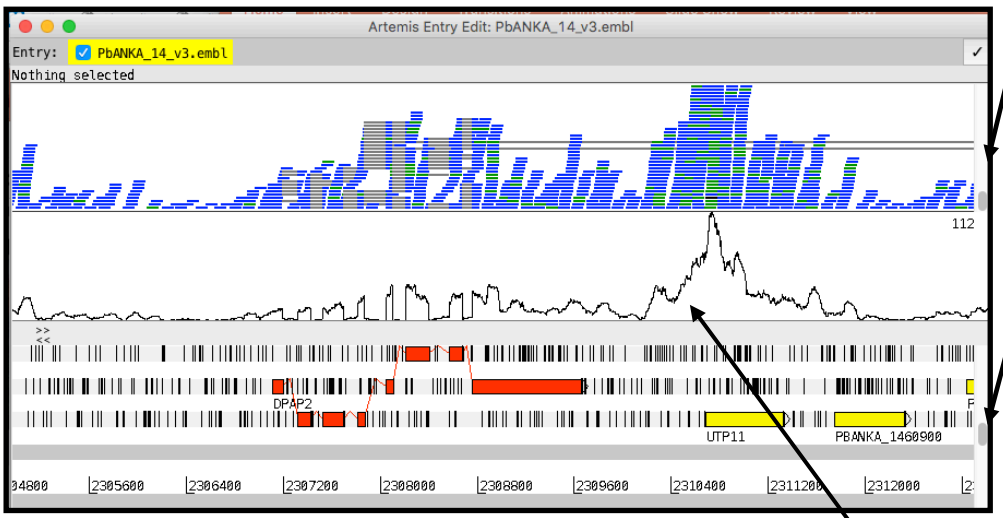
This is the so-called one-base pair resolution of RNA-Seq!

Right click in "BAM view,, select Graph -> Coverage. Then zoom out again



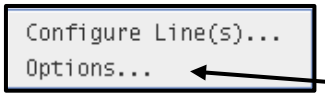
C. Interpreting the mapping

Zoom out until you have the same view as below:

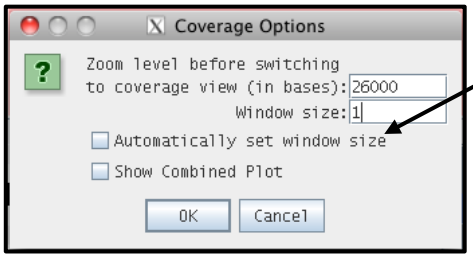


You can move the reads up and down, on the right scroll bar.

You can increase the size of the bam view, by dragging down with the mouse.



To better see the splice sites, do right click.
Select “Options…”:
Set the window size to 1 (before unselect “Automatic…”)



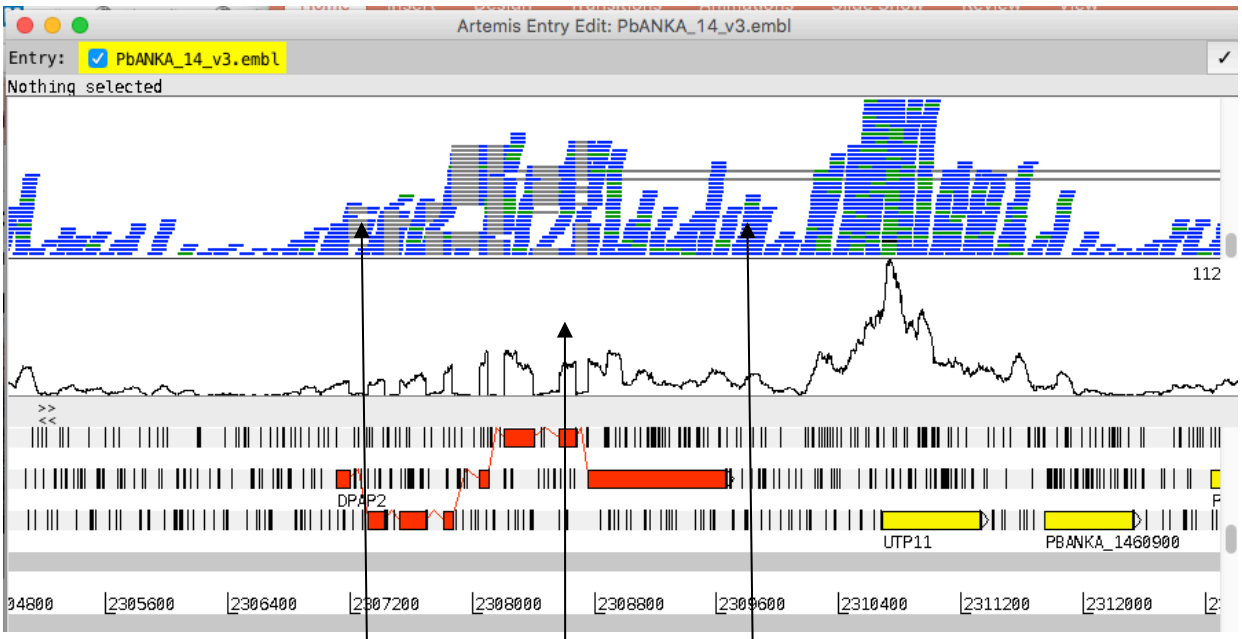
Please discuss following aspects with your neighbour:

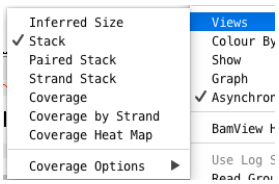
The coverage represents the amount of reads mapped over each position. Why are reads mapped where no exons are? Can you distinguish transcription starts and stops of genes?

Notice that different genes have different depths of coverage. What does this means?

Scroll through the genome and look at half a dozen genes, also some longer ones.
Why do some genes have less coverage? Have some genes no reads mapped to them? Is the coverage very even over longer genes?

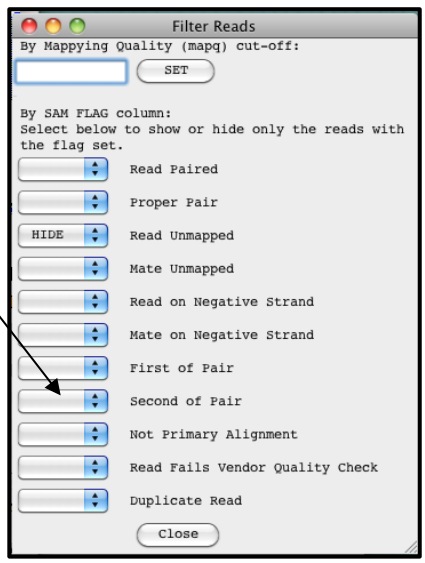
Actually, these data are strand specific! This means that from the reads you can determine the orientation of the transcript. This is useful to find RNA reverse tanscribed of genes. That might help also answer some questions, like were genes start, when they are head to head.





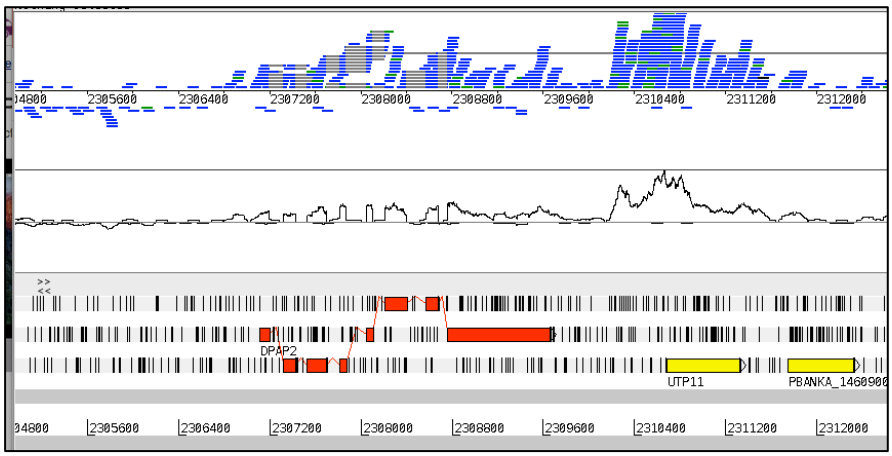
1. Right click, ->View -> strand stack

2. Right click, ->Filter Reads…



3. **Show** the second pair

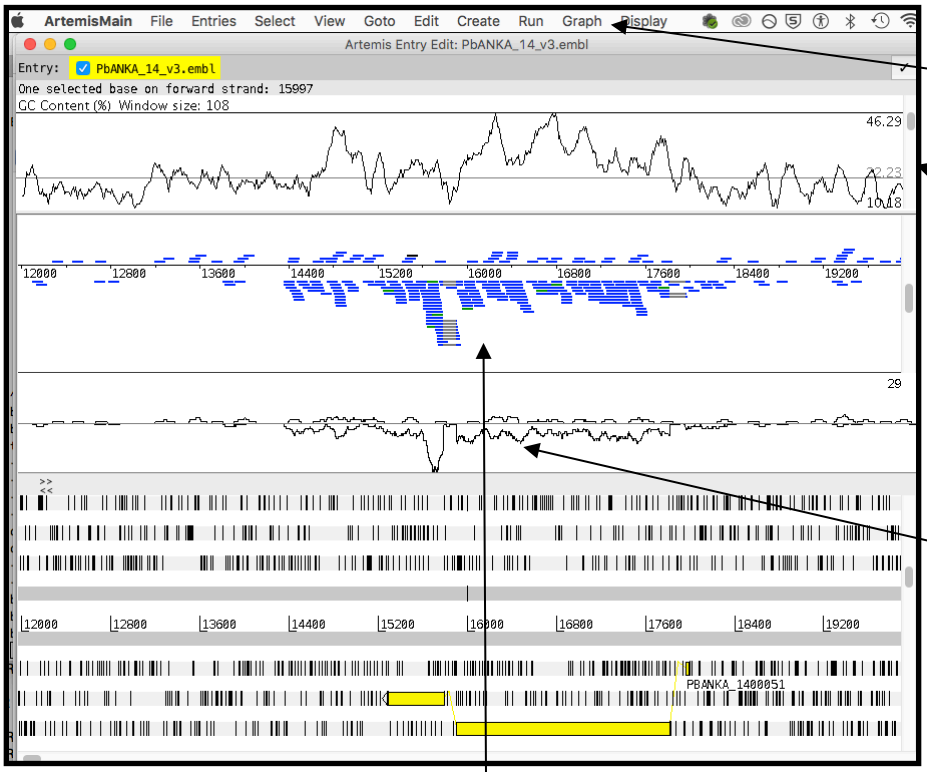
4. Right click, on plot, -> Options -> plot by strand!



NICE!

D. Uniqueness and GC content

Go to the position 16000 (Goto -> navigator).



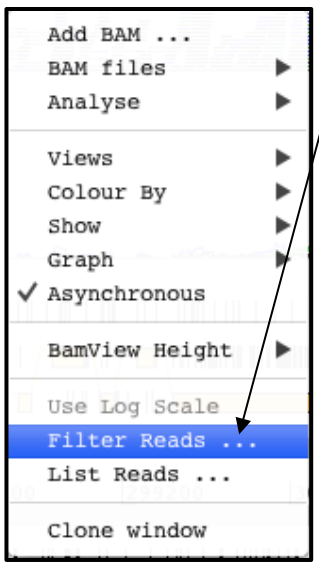
1. Enable the GC content, Graph -> GC Content.

2. Change the window size

3. What are those peaks? Is there a correlation to the GC content?

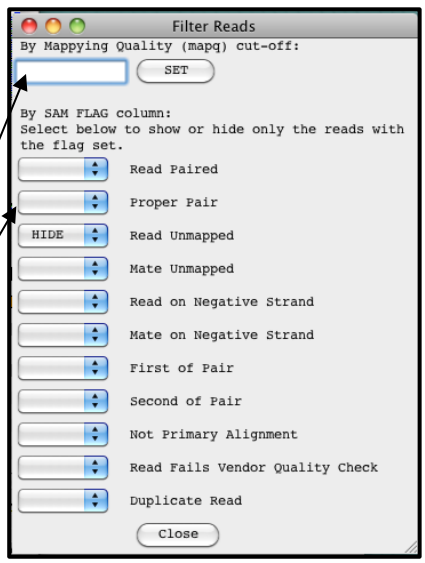
4. You can filter reads by mapping quality and if they are mapped as proper mate pairs.

5. Right click, than Filter Reads…



6. Set the mapping quality to 10 and show proper pairs. What happens?

What does that mean for the expression values of those genes?

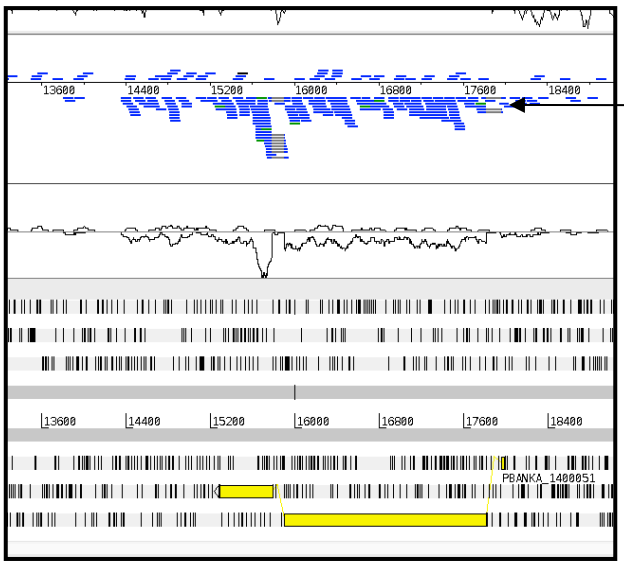


Variation in coverage can have many reasons, one is GC content. Also important, reads can be placed more than once, when they are mapped repetitively. More conservative mapping is to just look at proper pairs, and ignore reads with a mapping quality score below 5.

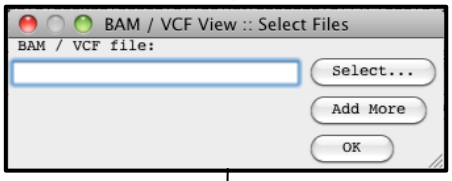
E. Including the mutant data set

Next we want to include the mutant (knock out) data set.

The reads of the KO parasite are in directory bam.



Right click here, select add BAM

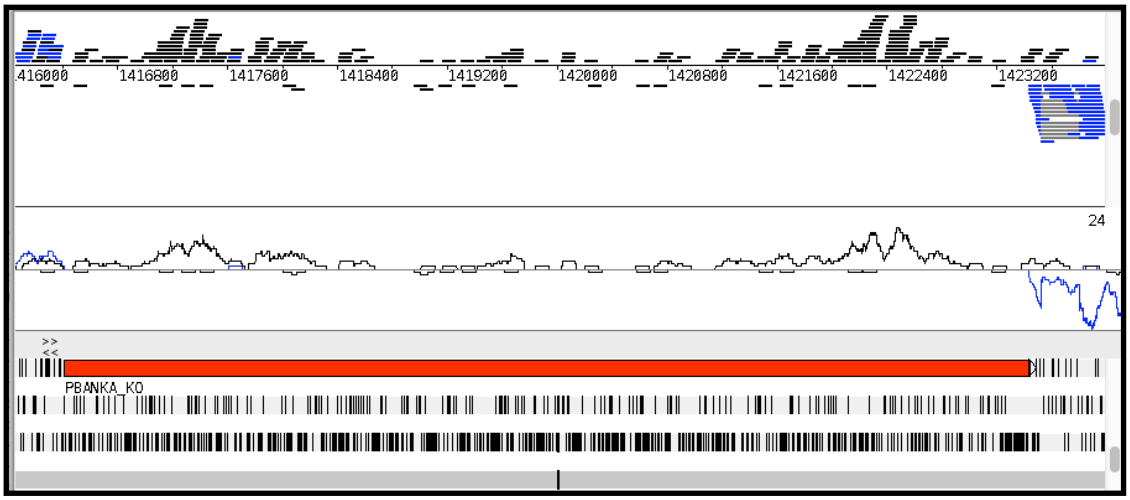


Include the file KO1.bam from the bams directory.

In the BAM view of the reads, it might be difficult to distinguish the differences between the two different BAM files (data sets). But in the coverage plot, one can see the differences in coverage by the color. You can color the read by the coverage plot (right click BAMview -> color by -> Coverage plot colors.

First have a look at the knock out gene (PBANKA_KO). Is it really knocked out?

It seems quite convincing that this gene is not expressed at all in the mutant (blue coverage plot). So the knock out seem to have worked. Interestingly, it is just a little bit expression in the WT… can that be an important gene after all?



Skim through the genome and compare the expression (coverage plots) between the two conditions. Again discuss the following questions with your neighbour or a tutor:

Which genes have extreme different coverage? Find a few and write the gene id numbers down.

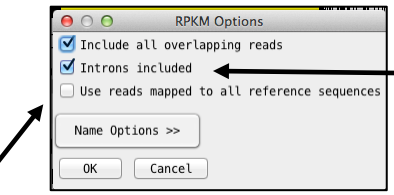
Is it enough to look at raw coverage, or would you need some kind of normalization?

F. Normalization - RPKM

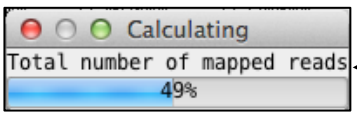
You have seen that the coverage varies between the two samples and it is difficult to compare. One possibility of normalizing the data is to generate the RPKM for each gene. RPKM stands for **r**eads per **k**ilobase **p**er **m**illion mapped reads. It is a measure of how many reads map to a gene, normalized by the gene length and by the amount of mapped reads in the run.

1. Select all genes by: Click on Select -> All CDS Features

2. Right click on the BAM view -> Analysis -> RPKM values of selected features…



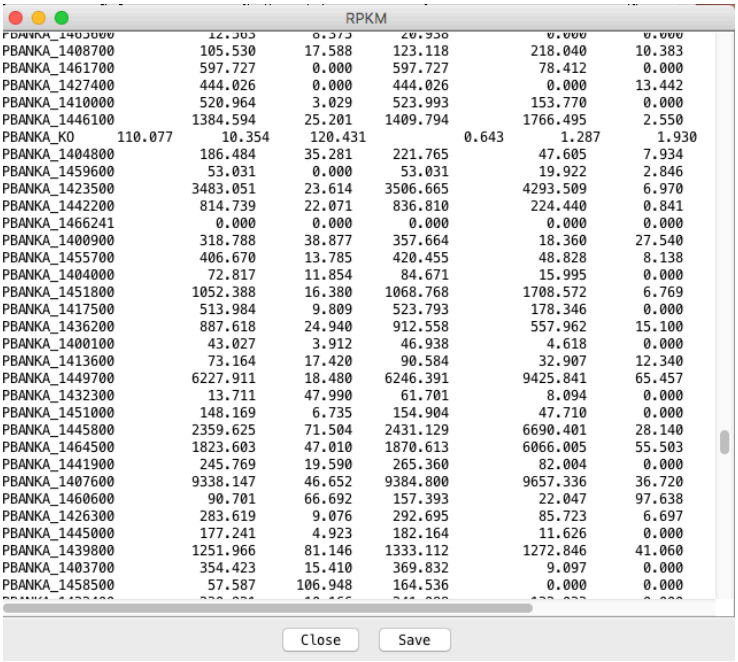
3. Unselect "Intron included"



4. Wait until the box says it is done.
Maybe take a break to do some stretching for your back… at home this will take longer. It is faster to use local copies of the BAM files!

5. The upcoming window will have RPKM values for each gene, for both the WT and the mutant. This will be split by strand of the DNA and a total score for both strands of DNA.

Save the file as Pb_RPKM.csv. This one you could load into LibreOffice (Excel), but here we are going to use a Linux "one-liner".



Now we would like to know which genes have the biggest difference in terms of expression between them. One way is to generate the ratio of the RPKM of WT and KO and look at the most extreme values. This can be done very easily on the command line:

```
$ awk '$4>100{print $1,$4,$7,($4/($7+0.001))}' Pb_RPKM.csv |
sort -rnk 4 | head -n 20
```

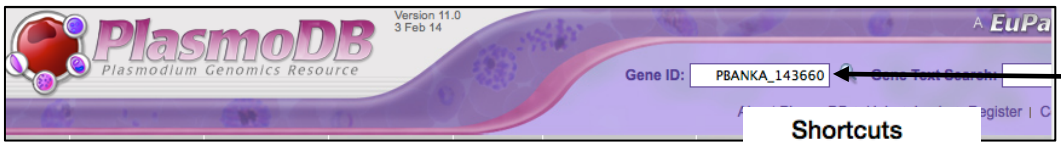
The `awk` commands can access columns in a file (like Excel) and do mathematical operations in this case the ratio. We just want genes that are expressed ($4>100).The output is piped into the `sort` program, that sort **n**umeric **r**everse and column 4 (k). And we are just interested in the top 20 lines (head -n 20).

What happened if you try `tail` instead of `head`?

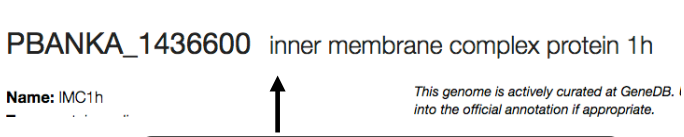
```
PBANKA_1419500 352.344 0.000 352344
PBANKA_1431500 315.246 0.000 315246
PBANKA_1458500 164.536 0.000 164536
PBANKA_1453900 157.680 0.000 157680
PBANKA_1402400 118.764 0.000 118764
PBANKA_1446400 112.513 0.000 112513
PBANKA_1422000 105.384 0.000 105384
PBANKA_1436600 1195.609 2.935 407.224
PBANKA_1421700 1388.498 6.691 207.486
PBANKA_1430300 1541.091 8.051 191.392
PBANKA_1461300 2466.177 26.137 94.3522
PBANKA_1449300 749.267 8.242 90.8974
PBANKA_1414800 962.378 14.203 67.754
PBANKA_1419300 739.357 11.337 65.2105
PBANKA_KO 120.431 1.930 62.3672
```

If your values are different - maybe you filtered the reads differently, and that is no problem at all!

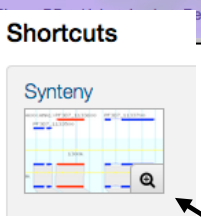
Open at least the two marked genes in PlasmoDB (http://plasmodb.org) and enter the first (yellow) gene id.



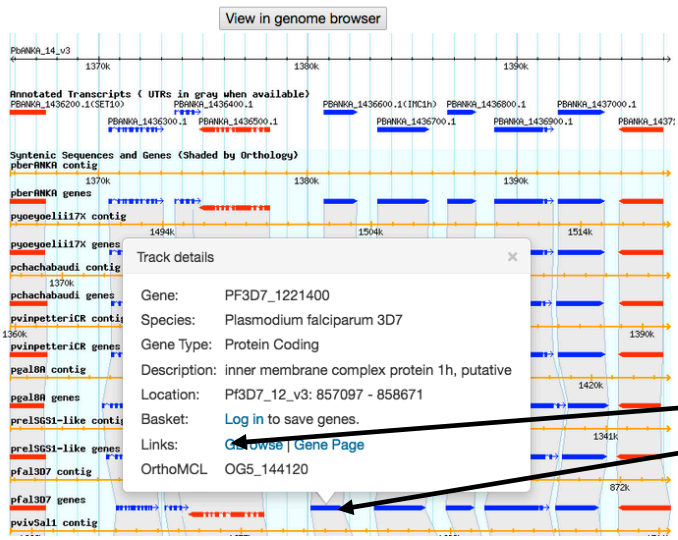
1. Type the gene IDs in here.



2. Read the gene page. Does it tell you about the function of the down regulated gene?

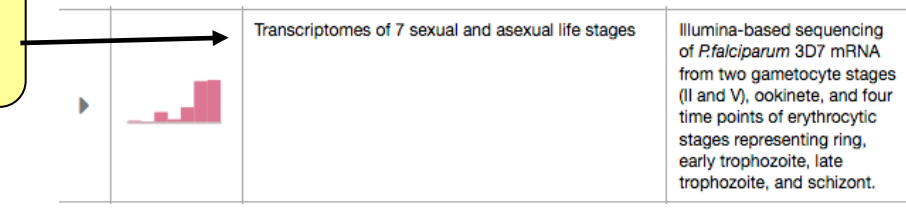


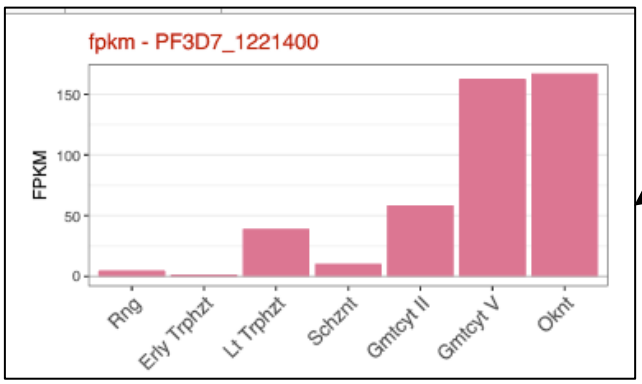
3. Select the synteny



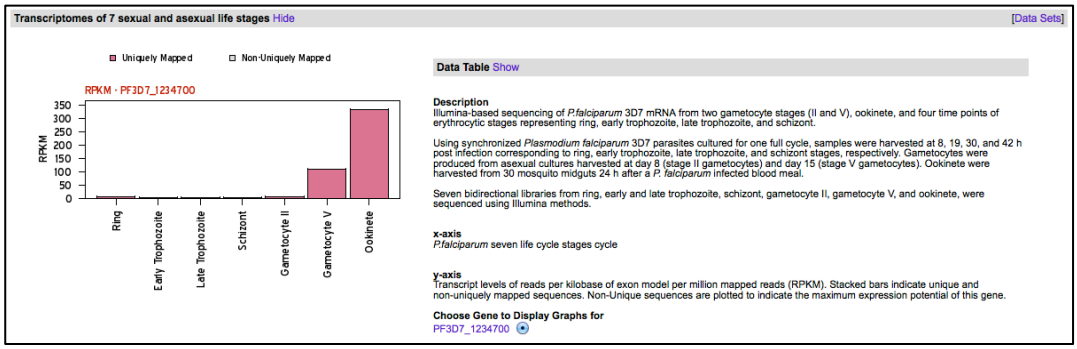
4. The genome of *P. falciparum* 3D7 has a far richer annotation, so let's look at the orthologue.

Scroll down until you come to the transcriptome data for expression in the sexual stages the 17th or so.





Doing the same with the following gene (PBANKA_1449300), that has the annotation "CPW-WPC family protein, putative", returns a similar pattern.



When are those genes mostly expressed? Could you formulate a hypothesis what kind of genes the knocked out gene might control?

What genes would you expected to be up regulated in the mutant?

Conversely, how much can you trust those results? Could the variation be down to noise, or natural variation?

What extra data would be useful to help us to be more confident about our conclusions?

Differential Expression

Introduction

Understanding the genome is not simply about understanding which genes are there. Understanding when each gene is used helps us to find out how organisms develop and which genes are used in response to particular external stimuli. The first layer in understanding how the genome is used is the transcriptome. This is also the most accessible because like the genome the transcriptome is made of nucleic acids and can be sequenced relatively easily. Arguably the proteome is of greater relevance to understanding cellular biology however it is chemically heterogeneous making it much more difficult to assay.

Over the past decade or two microarray technology has been extensively applied to addressing the question of which genes are expressed when. Despite its success this technology is limited in that it requires prior knowledge of the gene sequences for an organism and has a limited dynamic range in detecting the level of expression, e.g. how many copies of a transcript are made. RNA sequencing technology using, for instance Illumina HiSeq machines, can sequence essentially all the genes which are transcribed and the results have a more linear relationship to the real number of transcripts generated.

The aim of differential expression analysis is to determine which genes are more or less expressed in different situations. We could ask, for instance, whether a bacterium uses its genome differently when exposed to stress, such as excess heat or a drug. Alternatively we could ask what genes make human livers different from human kidneys.

In this module we will try to gain more understanding of the genes differentially expressed between the wild type and knock out of our experiment. We are going to use three biological replicates of the WT and three biological replicates of the mutant to get more statistical power. And we are going to use new tools, and just for Omar something to click, once the analysis is done.

G. Using *Kalliso* and *Sleuth* to identify differentially expressed genes

Kallisto is a read mapper, but instead of mapping against the genome it is designed to map against the transcriptome, i.e. the spliced gene sequences inferred from the genome annotation. Rather than tell you where the reads map it's aim is in quantifying the expression level of each transcript. It is very fast because it uses pseudo-alignment rather than true read alignment.

Kallisto needs an index of the transcript sequences (`Pb.CDS.fasta`).

```
$ kallisto index -i Pb.transcript transcript_sequences
```

Quantify the expression levels of your transcripts for the MT1 sample. The read file are again **WT1_1.fastq.gz WT1_2.fastq.gz.**

```
$ kallisto quant -t 3 --rf-stranded -i index_name -o MT1 -b
100 read_1 read_2
```

The results are contained in the file MT1/abundance.tsv. Kind of we are, the other 5 samples are already mapping with the same command: the command looks like:

```
for x in WT2 WT3 KO1 KO2 KO3 ; do kallisto quant -t 3 --rf-stranded -i
Pb.transcript -o $x -b 100 $x\_1.fastq.gz $x\_2.fastq.gz; done
```

But again, this is already done!

Sleuth uses the output from *Kallisto* to determine differentially expressed genes. It is written in the R statistical programming language, as is almost all RNA-seq analysis software. Helpfully however it produces a web page that allows interactive graphical analysis of the data. However, I would recommend learning R for anyone doing a significant amount of RNA-seq analysis. It is nowhere near as hard to get started with as full-blown programming languages such as Perl or Python!

We have provided a series of R commands which will get *Sleuth* running. These are in the file *sleuth.R*. Open the file and have a look.

```
$ cat sleuth.R
```

It is not as hard as it seems, I copied most of this from the manual! To run this R script, you will have to open R:

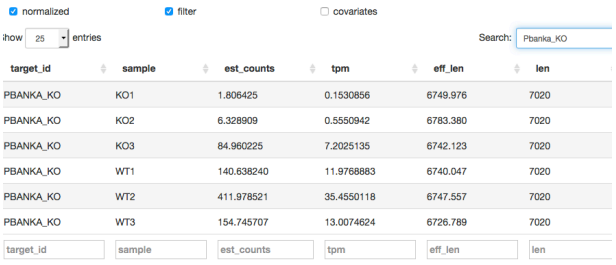
$ R
And then copy and paste commands from the file *sleuth.R*
This should open a browser….

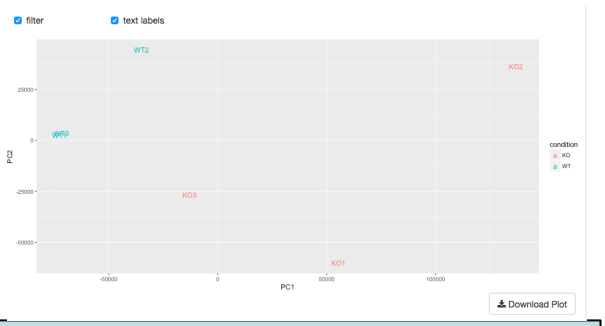
H. Using *Sleuth* to quality check the data

Sleuth provides several tabs which we can use to determine whether the data is of good quality and whether we should trust the results we get.

In the web page which has been launched click on Summaries->processed data.

Even though we have used the same number of reads for each sample, there are large differences in the number of reads mapping for each one. Why might this be? Is it a problem?





Click on map->PCA.

The Principal Components Analysis plot shows the relationship between the samples in two dimensions (PC1 and PC2). In this case almost all the variation between the samples is captured by just Principal Component 1. The WT samples are well separated from the WT samples, meaning that the replicates are more similar to each other than they are to samples from the different condition. This is good. But it seems that one KO is a bit distant… hm...

In some cases we identify outliers, e.g. samples which do not agree with other replicates and these can be excluded. If we don't have many replicates, it is hard to detect outliers and our power to detect differentially expressed genes is reduced.

BTW, we for sleuth we needed to generate two more files! `hiseq_info.txt` and `Pb.CDS.fasta`, have a look.. if you once need to generate them, drop me a line, thomasdan.Otto@glasgow.ac.uk!

I. Interpreting the results

In the R script we printed out a file of results describing the differentially expressed genes in our dataset. This is called “kallisto.results”.

The file contains several columns, of which the most important are:

Column 1: target_id (gene id)
Column 2: pval (p value)
Column 3: qval (p value corrected for multiple hypothesis testing)
Column 4: b (fold change)
Column 12: description (some more useful description of the gene than its id)

Go back to Linux. With a little of magic we can get the list of differentially expressed genes with only the columns of interest as above. The following command will get those genes which have an adjusted p value less than 0,01 and a positive fold change. These genes are more highly expressed in SBP samples.

```
$ cut -f1,3,4,12 kallisto.results | awk '$2 < 0.01 && $3 > 0'
```

These genes are more highly expressed in MT samples:

```
cut -f1,3,4,12 kallisto.results | awk '$2 < 0.01 && $3 < 0'
```

How many genes are more highly expressed in each condition? (use `| wc –l` command!)

Do you notice any particular genes that come up in the analysis?

Now let's compare this list to the one before. What are the differences? Is the list similar to your first list of differentially expressed genes?

Do you understand each column?

Which results would you trust more (this or the ratio in the Excel table)?

If time permits lookup more genes up in plasmodb…

What other datasets would help in the interpretation of the results?

gene_id	q-value	fold expression	product
PBANKA_1431400	6.28573573826137e-15	2.90541334960253	conserved Plasmodium protein, unknown function
PBANKA_1458800	5.88973337957171e-12	2.41338602186776	kinesin, putative
PBANKA_1419300	3.17796604134042e-10	3.63543283541383	conserved Plasmodium protein, unknown function
PBANKA_1421500	9.94515679668602e-09	2.8444049121292	conserved Plasmodium protein, unknown function
PBANKA_1445700	1.72292404298525e-07	2.15214104518609	conserved Plasmodium protein, unknown function
PBANKA_1421400	7.15835281629491e-07	3.7918779791166	C-Myc-binding protein, putative
PBANKA_1455800	2.00880355908585e-06	1.97989601893411	GAS8-like protein, putative
PBANKA_1449000	8.4373055659006e-06	3.4288280805515	aspartyl protease, putative
PBANKA_1405000	1.39265260604041e-05	2.03202423522353	MORN repeat protein, putative
PBANKA_1461300	2.20725821001193e-05	3.29140785393363	conserved Plasmodium protein, unknown function
PBANKA_1421000	5.32004223642822e-05	2.20121966902428	calmodulin, putative
PBANKA_1463300	5.98153928630341e-05	2.17624218138509	conserved Plasmodium protein, unknown function
PBANKA_1448000	6.0374884324577e-05	1.44426682481122	pentatricopeptide repeat domain-containing protein, putative
PBANKA_1432400	6.49922638605398e-05	2.94787483175709	perforin-like protein 2
PBANKA_1460700	0.000160661046661276	3.4674617405814	dipeptidyl aminopeptidase 2
PBANKA_1463700	0.000197119395073462	1.66910330280852	DNA repair protein rhp16, putative
PBANKA_1430900	0.000225344469051259	1.30635422585721	conserved Plasmodium protein, unknown function
PBANKA_1466181	0.000287734943841492	3.36472542625359	BIR protein, pseudogene // PIR protein, pseudogene
PBANKA_1437700	0.000423587811520067	1.25811367768951	conserved Plasmodium protein, unknown function
PBANKA_1414500	0.000507995668623349	3.08778913002022	protein kinase, putative
PBANKA_1420700	0.000507995668623349	1.49075353351279	MAATS1 domain-containing protein, putative
PBANKA_1431500	0.000544939833660155	3.68436798976629	conserved Plasmodium protein, unknown function
PBANKA_1436600	0.00100991573190885	3.69354470377221	inner membrane complex protein 1h

The list is a bit different as we are looking now at the complete genome. But if you would do a `| grep _14`, you would see that there are not tooooo many differences.

Anyhow, this does not necessarily help us to find the function of the gene we knocked out… let's write out the differential expressed genes, with a log fold change of 2, and do a GO enrichment in PlasmoDB.

```
$ cut -f1,3,4,12 kallisto.results | awk '$2 < 0.01 && $3 > 0'
| cut -f 1 > UpRegulated.txt
```

GO enrichment

Maybe some of you have already determined the function of the transcription factor. But this would have been done manually. A more automated method would be to do a GO enrichment. Basically, statistics are used to test if a function (or GO term) is enriched in the down or up regulated genes compared to all of the GO terms associated to the genes that are expressed.

Gene Ontology or GO, is a major bioinformatics initiative to unify the representation of gene and gene product attributes across all species, see http://en.wikipedia.org/wiki/Gene_ontology. GO terms are represented in directed acyclic graph, so functions can be further specified in a sub node. The GO enrichment test we will use takes the structure of this hierarchy into account.

But the association of GO terms to genes depend on the known functions and level of curation. For example, in *P. berghei*, less than half of the genes have GO terms associated!

In this exercise we will do a GO enrichment of the differentially expressed genes of the complete gene set (not just chromosome 14).

Change the directory and have a look at the files:

```
$ cd ~/Module_RNA-Seq/GO
$ ls
```

The file `full.gene_exp.diff` has the same format as the output of `cuffdiff` you produced. But it was generated for all of the genes in the genome, not just for chromosome 14.

The next command will get all the gene ids of genes that are
- differentially expressed (`grep yes`)
- down regulated in the mutant (`$10<0` - log fold change),
- have a FPKM of at least 40 (`$8>40` - FPKM of WT),
- are three times more expressed in the WT (`$8 > (3*$9)`).

To do the filtering, we are using the command `awk`. The "$" refers to the i-th column in the text file. As the first row contains the id's, it is returned with `cut -f 1` and then saved in a file, using the ">" command. (You could do that in excel, but it might take a bit more time…)

```
$ grep yes full.gene_exp.diff | awk '$10<0 && $8>40 &&
$8>(3*$9)' | cut -f 1 > list.down.txt
$ head list.down.txt
$ head Pb.GOterms.txt
```

The two head commands give you an idea of the format of the two files we are going to use.

Though the enrichment test is done in R, using the bioconductor class topGO, we are going to call it directly from the command line. Maybe have a quick look at the code to see how the enrichment is done.

```
$ cat doGO.R
```

So next we are going to call the program, looking for the biological process (BP), see http://en.wikipedia.org/wiki/Gene_ontology.

```
$ R CMD BATCH "--args list.down.txt Pb.GOterms.txt BP “
doGO.R
```

This command tells R to run from the command line the program `doGO.R`. Three parameters are given:

1. Genes of interest - which you generated
2. GO database
3. The domain search: BP (biological process, e.g. cell cycle), MF (molecular function, e.g. kinase) or CC (cellular component, e.g. nucleus, cytoplasm)

The result is in file Result.txt

```
$ cat Result.txt
```

Google the first hit, “microtubule-based movement” including “malaria” as further search term. What paper pops out first? Does this help to understand which genes the knocked out transcription factor might regulate?

Can you repeat the analysis with with the other GO domains (CC and MF)?

Would you be able to repeat the analysis with up regulated genes in the mutant? Which processes are enriched. Are the results expected?

Would it make sense to change the criteria to generate the list of up and down regulated genes? If so, how and why?

Do not panic…

… if you don’t understand everything! This is a very advanced methodology. It involved bioinformatics, statistics and deep knowledge into the parasite. At the same time, the results depend on many parameters like, experiment setup, quality of your RNA-Seq data, parameter used in the different steps and the quality of the GO database.

Important: In the end you got several enriched functions as result of your experiment that characterize the function of the knocked out gene! *Well done!*

Key aspects of differential expression analysis

Replicates and power

In order to accurately ascertain which genes are differentially expressed and by how much it is necessary to use replicated data. As with all biological experiments doing it once is simply not enough. There is no simple way to decide how many replicates to do, it is usually a compromise of statistical power and cost. By determining how much variability there is in the sample preparation and sequencing reactions we can better assess how highly genes are really expressed and more accurately determine any differences. The key to this is performing biological rather than technical replicates. This means, for instance, growing up three batches of parasites, treating them all identically, extracting RNA from each and sequencing the three samples separately. Technical replicates, whereby the same sample is sequenced three times do not account for the variability that really exists in biological systems or the experimental error between batches of parasites and RNA extractions.

n.b. more replicates will help improve power for genes that are already detected at high levels, while deeper sequencing will improve power to detect differential expression for genes which are expressed at low levels.

P-values vs. q-values

When asking whether a gene is differentially expressed we use statistical tests to assign a p-value. If a gene has a p-value of 0.05 we say that there is only a 5% chance that it is not really differentially expressed**.** However, if we are asking this question for every gene in the genome (~5500 genes for *Plasmodium*), then we would expect to see p-values less than 0.05 for many genes even though they are not really differentially expressed. Due to this statistical problem we must correct the p-values so that we are not tricked into accepting a large number of erroneous results. Q-values are p-values which have been corrected for what is known as **multiple hypothesis testing**. Therefore it is a q-value of less than 0.05 that we should be looking for when asking whether a gene is differentially expressed.

Alternative software

If you have a good quality genome and genome annotation such as for model organisms e.g. human, mouse, *Plasmodium*, I would recommend mapping to the transcriptome for determining transcript abundance. This is even more relevant if you have variant transcripts per gene as you need a tool which will do its best to determine which transcript is really expressed. As well as Kallisto (Bray et al. 2016; PMID: 27043002), there is eXpress (Roberts & Pachter, 2012; PMID: 23160280) which will do this.

Alternatively you can map to the genome and then call abundance of genes, essentially ignoring variant transcripts. This is more appropriate where you are less confident about the genome annotation and/or you don't have variant transcripts because your organism rarely makes them or they are simply not annotated. Tophat2 (Kim et al., 2013; PMID: 23618408), HISAT2 (Pertea et al. 2016; PMID: 27560171), STAR (Dobin et al., 2013; PMID: 23104886) and GSNAP (Wu & Nacu, 2010; PMID: 20147302) are all splice-aware RNA-seq read mappers appropriate for this task. You then need to use a tool which counts the reads overlapping each gene model. HTSeq (Anders et al., 2015; PMID: 25260700) is a popular tool for this purpose. Cufflinks (Trapnell et al. 2012; PMID: 22383036) will count reads and determine differentially expressed genes.

There are a variety of programs for detecting differentially expressed genes from tables of RNA-seq read counts. DESeq2 (Love et al., 2014; PMID: 25516281), EdgeR (Robinson et al., 2010; PMID: 19910308) and BaySeq (Hardcastle & Kelly, 2010; PMID: 20698981) are good examples.

What do I do with a gene list?

Differential expression analysis results is a list of genes which show differences between two conditions. It can be daunting trying to determine what the results mean. On one hand you may find that that there are no real differences in your experiment. Is this due to biological reality or noisy data? On the other hand you may find several thousands of genes are differentially expressed. What can you say about that?

Other than looking for genes you expect to be different or unchanged, one of the first things to do is look at Gene Ontology (GO) term enrichment. There are many different algorithms for this, but you could annotate your genes with functional terms from GO using for instance Blast2GO (Conesa et al., 2005; PMID: 16081474) and then use TopGO (Alexa et al., 2005; PMID: 16606683) to determine whether any particular sorts of genes occur more than expected in your differentially expressed genes.

References

Alexa A, Rahnenfuhrer J, Lengauer T. 2006. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. Bioinformatics 22(13): 1600-1607.

Anders S, Huber W. 2010. Differential expression analysis for sequence count data. Genome Biol 11(10): R106.

Conesa A, Gotz S, Garcia-Gomez JM, Terol J, Talon M, Robles M. 2005. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. Bioinformatics 21(18): 3674-3676.

Hardcastle TJ, Kelly KA. 2010. baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. BMC Bioinformatics 11: 422.

Lawton J, Brugat T, Yan YX, Reid AJ, Bohme U, Otto TD, Pain A, Jackson A, Berriman M, Cunningham D et al. 2012. Characterization and gene expression analysis of the cir multi-gene family of Plasmodium chabaudi chabaudi (AS). BMC Genomics 13: 125.

Otto et al. (2010) Mol Microbiol Apr;76(1):12-24. New insights into the blood stage transcriptome of Plasmodium falciparum using RNA-Seq.

Robinson MD, McCarthy DJ, Smyth GK. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26(1): 139-140.

Spence PJ, Jarra W, Levy P, Reid AJ, Chappell L, Brugat T, Sanders M, Berriman M, Langhorne J. 2013. Vector transmission regulates immune control of Plasmodium virulence. Nature 498(7453): 228-231.

Trapnell C, Pachter L, Salzberg SL. 2009. TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 25(9): 1105-1111.

Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L. 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nature biotechnology 28(5): 511-515.