Module 7
Genome Annotation and Differential Expression

Introduction to Genome Annotation

One of the key goals of producing a draft genome is to define the genes encoded in it. This is the first step in answering many important questions. How many genes does this organism have? What metabolic pathways are present? Are there novel gene families encoded in the genome? Subsequent uses of the genome, such as proteomics and transcriptomics experiments rely on accurate gene models. We concentrate here on protein coding genes, however one would also try to identify non-protein coding RNAs such as tRNAs, rRNAs, miRNAs, transposons etc.

Producing accurate **gene models** is just as hard as producing a good assembly. You have seen that one way of producing a set of gene models is to transfer them from a closely related organism. However, if there is not a closely related genome, or the most closely related genome is not well annotated, this may not be an option. Furthermore, even if there is a closely related, well annotated reference genome as in the case of the malaria parasite - *Plasmodium falciparum* strain 3D7 and *P. falciparum* strain IT, there may be regions of your genome of interest which are not syntenic to the reference. Indeed, this is the case here, as the subtelomeric regions of *Plasmodium* chromosomes are highly variable and cannot be used to transfer gene models, even between strains of the same species.

When there is no reference genome, or for regions which are not syntenic to the reference, we can use ***ab initio* gene finding** methods. These identify genes based on properties of the genome sequence independent of whether they show homology to known genes. They can be trained and it is common to identify the most well conserved genes by homology, then to train an *ab inito* gene finding algorithm using these well conserved genes so that it can learn what a gene looks like in the particular genome you are interested in. In this module we will use the program **Augustus** to predict gene models *ab initio*.

Another approach to identifying gene models is to determine those regions of the genome which are transcribed. Prior to the advent of second generation sequencing technologies Sanger capillary sequencing was used to sequence mRNAs and generate Expressed Sequence Tags (ESTs). These could be used to identify the most highly expressed genes and improve some gene models. With second generation sequencing technologies we are able to sequence mRNA transcripts from essentially all the genes which are expressed. This is known as **RNA sequencing** or RNA-seq (Mortazavi et al., 2008; Wang et al, 2009) and the resulting data provides incredible resolution of gene structure. This method is not biased by which genes are present in previously sequenced genomes (as with RATT) or by how much their structure reflects that of other genes in the genome (as with Augustus), but rather by how highly expressed they are and how extensively we sequence the transcriptome.

In this module you will generate a set of gene models *ab initio* using the gene prediction tool Augustus. It has been trained using the highly accurate, manually curated gene models of *P. falciparum* 3D7. These gene models will be used to fill in those regions of the *P. falciparum* IT assembly which could not be annotated using RATT because they are not syntenic to *P. falciparum* 3D7. You will then map RNA-seq data to your assembly and use this to improve the gene models. There will be inaccuracies from the RATT transfer due to technical error and due to real differences in the gene structures. There will be inaccuracies in the Augustus predictions due to gene models which do not follow the expected pattern of a *Plasmodium* gene and due to inaccuracies in the genome assembly. These can be addressed using the RNA-seq mapping.

Below is a model of the eukaryotic protein-coding gene highlighting features relevant to their annotation. Note that compared to bacteria, eukaryotic genes frequently have multiple exons, separated by un-translated introns and 5` and 3` Un-Translated Regions (UTRs) which do not encode part of the protein sequence.

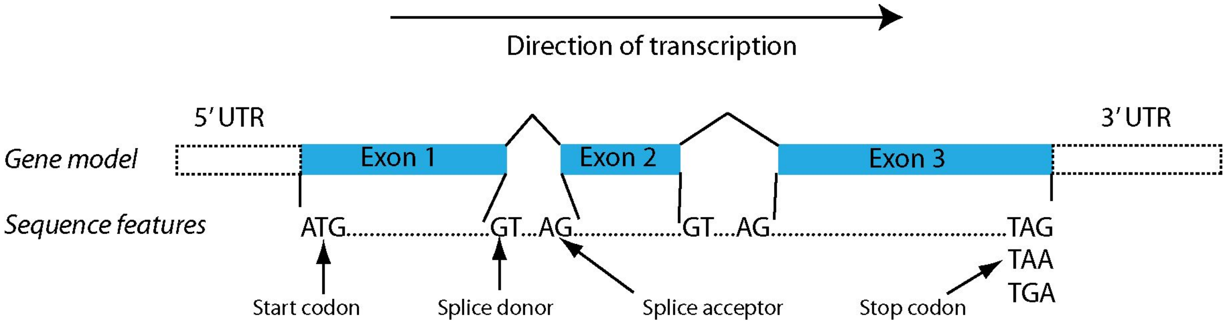

Direction of transcription
5′UTR
3′UTR
Gene model
Exon 1
Exon 2
Exon 3
Sequence features
ATG.........................GT...AG...................GT....AG..TAG
TAA
TGA
Start codon
Splice donor
Splice acceptor
Stop codon

Figure showing the key features of a eukaryotic gene. The exact DNA sequence of the splice sites may vary.

Module Summary

1. Generating an initial set of gene models (merging RATT and Augustus)
2. Mapping RNA-seq data to a reference
3. Viewing RNA-seq mapping in Artemis
4. Correcting gene models based on RNA-seq data
5. Automatically generating gene models based on RNA-seq data
6. Using RNA-Seq to improve annotation

1. Generating an initial set of gene models

A. Generate gene models *ab initio*

The *ab initio* gene prediction algorithm Augustus has already been trained with gene models from *Plasmodium falciparum* 3D7. You will now run it on your *Plasmodium falciparum* IT strain chromosome to predict a set of gene models.

Navigate to the module 7 data directory. On the command line, type:

```
augustus --species=pfalciparum IT.genome.fa > augustus.gtf
```

The file `augustus.gtf` now contains your predicted gene models

Next we are going to convert the Augustus GFF to EMBL format. On the command line, type:

```
cat augustus.gtf | augustus2embl.pl > augustus.embl
```

Although Artemis can display gff files, the visualization is better for embl files (if you want you can also try loading the gff file into Artemis).

It is typically much more challenging to train an *ab initio* gene finder than to run it. Most *ab initio* gene finders require a set of very accurate models to train them, which may be predicted from highly conserved genes, RNA-Seq and typically manual curation of a few hundred gene models. The chosen training set can influence which types of gene models get better or worse predictions, so should be carefully chosen.

B. Examine gene model predictions

We will open some gene models which we transferred from *P. falciparum* 3D7 using the tool RATT. We can compare them to the gene models predicted *de novo* by Augustus.

On the command line, type:

```
art Transfer.ordered_Pf3D7_05.final.embl &
```

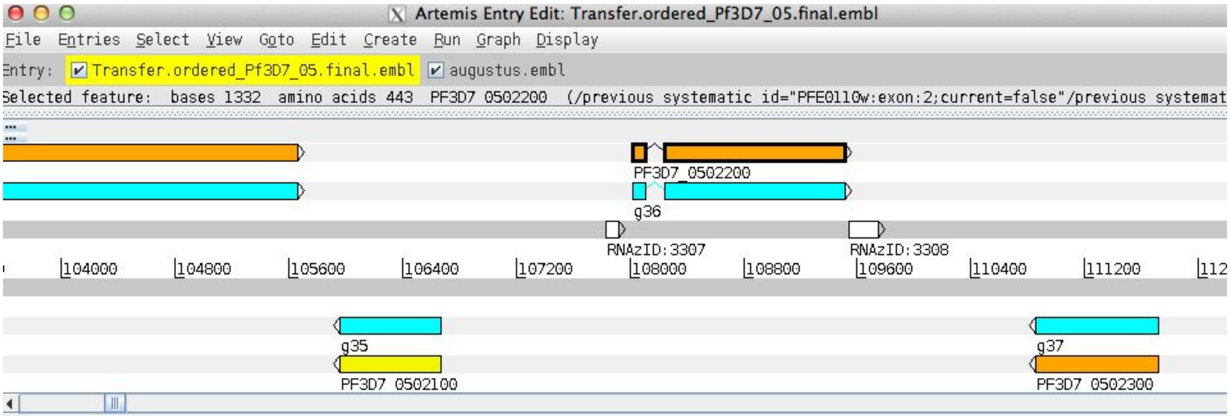
Load in Augustus models. In Artemis:

File -> Read An Entry and select the file `augustus.embl`.

Right click in genome window, select "One Line Per Entry".

The transferred models have different colours and annotation, while the newly predicted genes have the default colour (blue), and are named g1, g2, g3…

Check out different loci in this chromosome to appreciate the difference in the annotation files

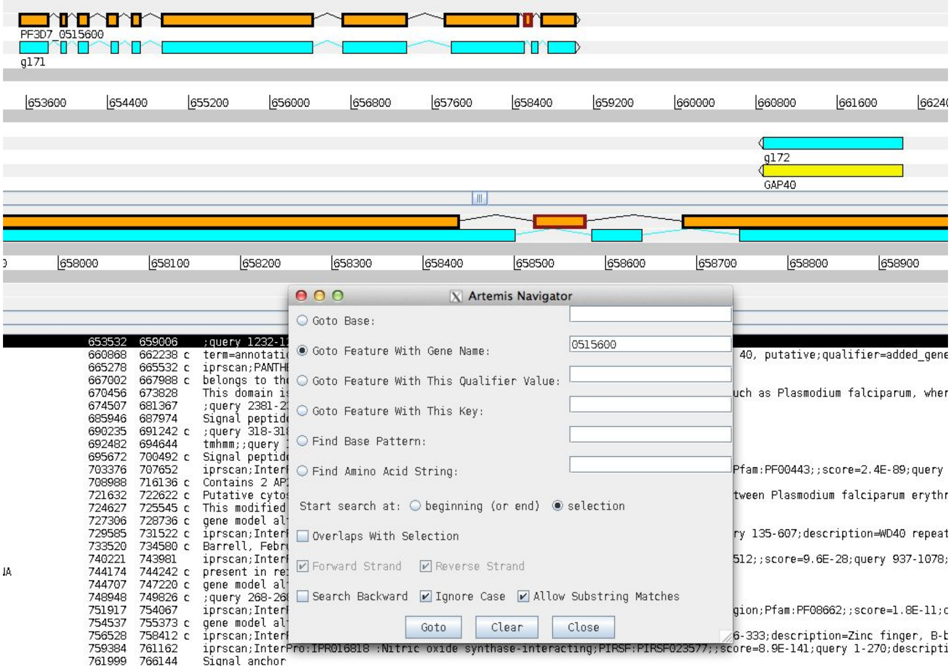


Artemis Entry Edit: Transfer.ordered_Pf3D7_05.final.embl

Entry: ☑ Transfer.ordered_Pf3D7_05.final.embl ☑ augustus.embl

```
Selected feature:  bases 1332  amino acids 443  PF3D7 0502200  (/previous systematic id="PFE0110w:exon:2;current=false"/previous systemat
```

Have a look at the gene PF3D7_0515600. Does Augustus perform better than RATT? What has gone wrong with the prediction? Go to the gene by Goto -> Navigator -> “Goto Feature With Gene Name”. Tip: you don’t have to type in the complete gene name.



○ Goto Base:

● Goto Feature With Gene Name: 0515600

○ Goto Feature With This Qualifier Value:

○ Goto Feature With This Key:

○ Find Base Pattern:

○ Find Amino Acid String:

Start search at: ○ beginning (or end) ● selection

□ Overlaps With Selection

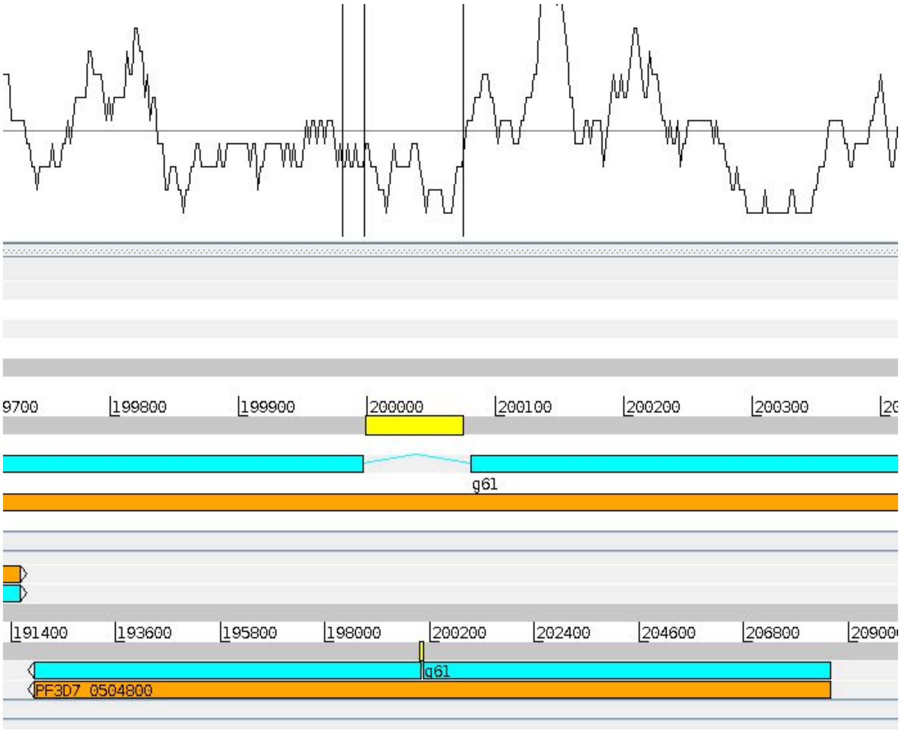
☑ Forward Strand ☑ Reverse Strand

□ Search Backward ☑ Ignore Case ☑ Allow Substring Matches

Goto Clear Close

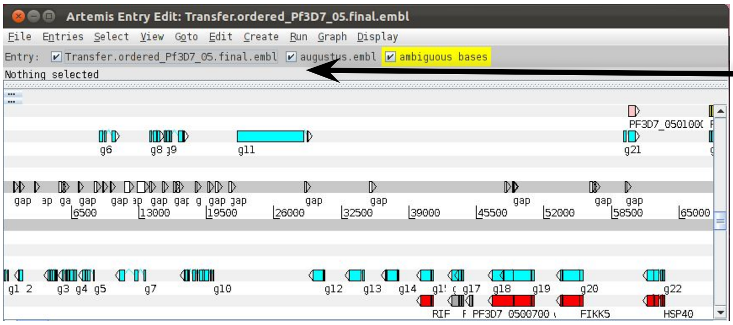
```
      653532  659006    ;query 1232-1
      660868  662238 c  term=annotati                                                   40, putative;qualifier=added_gene
      665278  665532 c  iprscan;PANTHE
      667002  667988 c  belongs to the
      670456  673828    This domain is                                                  uch as Plasmodium falciparum, wher
      674507  681367    ;query 2381-2
      685946  687974    Signal peptide
      690235  691242 c  ;query 318-31
      692482  694644    tmhmm;;query
      695672  700492 c  Signal peptide
      703376  707652    iprscan;Inter                                                   Pfam:PF00443;;score=2.4E-89;query
      708988  716136 c  Contains 2 AP
      721632  722622 c  Putative cytos                                                  tween Plasmodium falciparum erythr
      724627  725545 c  This modified
      727306  728736 c  gene model al
      729585  731522 c  iprscan;Inter                                                   ry 135-607;description=WD40 repeat
      733520  734580 c  Barrell, Febru
      740221  743981    iprscan;Inter                                                   512;;score=9.6E-28;query 937-1078;
JA    744174  744242 c  present in re
      744707  747220 c  gene model al
      748948  749826 c  ;query 268-26
      751917  754067    iprscan;Inter                                                   gion;Pfam:PF08662;;score=1.8E-11;c
      754537  755373 c  gene model al
      756528  758412 c  iprscan;Inter                                                   6-333;description=Zinc finger, B-b
      759384  761162    iprscan;InterPro:IPR016818 :Nitric oxide synthase-interacting;PIRSF:PIRSF023577;;score=8.9E-141;query 1-270;descripti
      761999  766144    Signal anchor
```

Look at the gene PF3D7_0504800. Perhaps due to the low GC content of this region Augustus decided that it is intronic (Graph -> GC content (%)). How could you verify the prediction?



C. Combine RATT and Augustus gene models

Augustus is able to predict gene models in regions of the IT genome which have no synteny with the 3D7 genome, principally the subtelomeres. However, on the whole, the RATT-transferred gene models ought to be more accurate because the IT and 3D7 genome are otherwise very similar. Therefore it is perhaps most useful to add in only those Augustus gene models which do not overlap RATT-transferred models. To do this, the Augustus gene models must be converted into EMBL format.



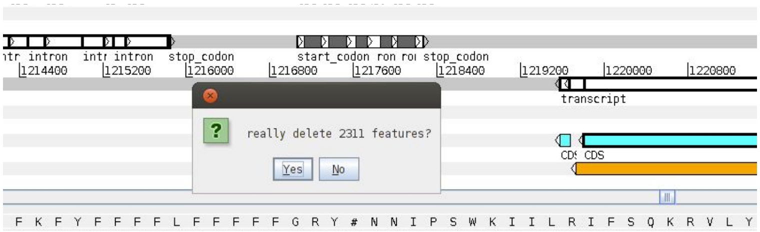
Artemis Entry Edit: Transfer.ordered_Pf3D7_05.final.embl

Entry: ☑ Transfer.ordered_Pf3D7_05.final.embl ☑ augustus.embl ☑ ambiguous bases

Nothing selected

Unselect the "augustus.embl" entry.

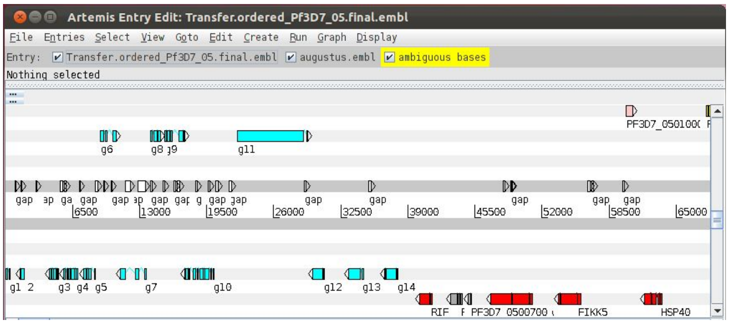
Go to Select -> All CDS features



really delete 2311 features?

Re-select the "augustus.embl" entry.

Go to Select -> Features Overlapping Selection



Artemis Entry Edit: Transfer.ordered_Pf3D7_05.final.embl

Entry: ☑ Transfer.ordered_Pf3D7_05.final.embl ☑ augustus.embl ☑ ambiguous bases

Nothing selected

n.b. deselect any augustus (blue) gene models which you have edited and want to keep by shift-clicking that model.

The Augustus gene models which overlap RATT models should now be selected. Delete them! Edit -> Selected Features -> Delete

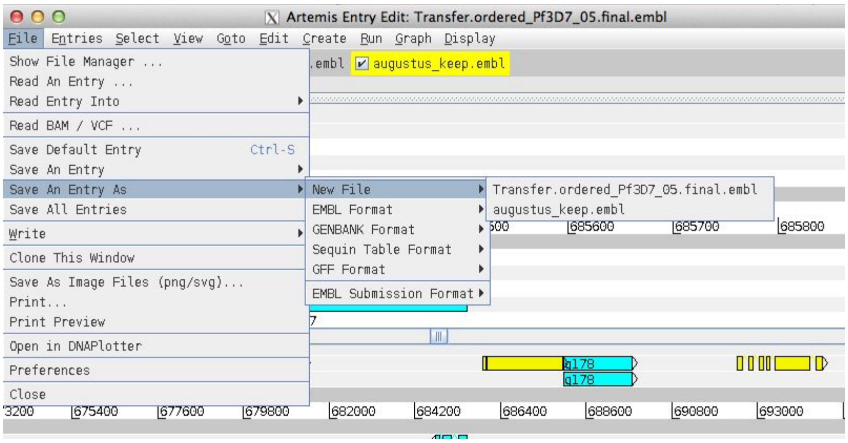
If you want to keep any augustus gene model you have edited, then delete the overlapping RATT model. How does the annotation look? How many extra gene models do we have compared to using the RATT-transferred ones alone? Use the function View -> Overview with and without the "augustus.embl" entry selected to see some stats. In the next section we are going to show how RNA-Seq data can be used to correct gene models.

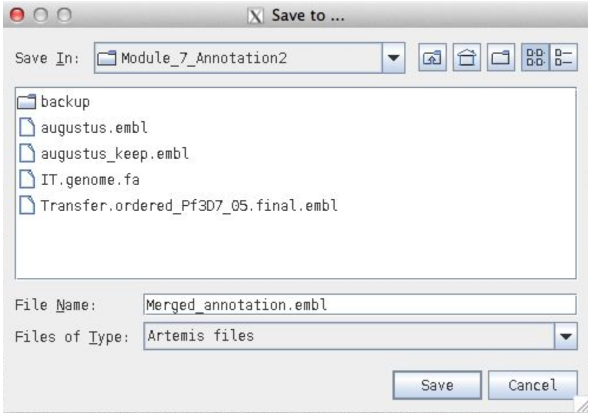
We need to save the merged annotation:
File -> Save An Entry As -> New File -> augustus.embl
Save to … "augustus_keep.embl"

Now merge it into the original file:
File -> Read Entry Into -> Transfer… Transfer.ordered_Pf3D7_05.final.embl Select a file … "augustus_keep.embl"

And now save the merged file:
File -> Save An Entry As -> New File -> Transfer.ordered_Pf3D7_05.final.embl
Save to … "Merged_annotation.embl"

This file "Merged_annotation.embl" can now be used as a basis for further annotation. Always remember to save your work! You can deselect the "augustus_keep.embl" entry.





Save In: Module_7_Annotation2

File Name: Merged_annotation.embl

Files of Type: Artemis files

Gene models can also be merged/overlapped/removed bioinformatically using custom scripts or the free command-line software BEDtools http://bedtools.readthedocs.org/en/latest/. The most convenient formats for manipulating annotation are GFF/GFF3/GTF and BED. Artemis works best with EMBL files, but we hope to improve compatibility with GFF in the future

D. Functional annotation

For those gene models transferred by RATT, you will have a range of functional information which will help you identify the types of genes present in your genome. This functional information has been manually curated for *P. falciparum* 3D7 based on the literature. For those genes predicted *de novo* by Augustus there is no such information. It is beyond the scope of this module to present a solution for assigning functional annotation for all these extra genes, but you can annotate a few of them yourself. Product calls for genes are usually defined by looking for orthologues or best BLAST hits in other organisms for which a gene has been annotated with a useful name. Many annotation databases exist for different purposes; Pfam for functional domains pfam.sanger.ac.uk, Gene Ontology for functional terms http://www.geneontology.org/ and KAAS for enzymes www.genome.jp/tools/kaas/. There are often specialized databases for specific classes of genes/organisms you might be particularly interested in.

For *Plasmodium* annotation, the best place to look is PlasmoDb, a large resource of comparative genomics data for these species.

Right click on an augustus gene model (blue ones), View -> Amino Acids of Selection as Fasta -> Ctrl-A -> Ctrl-C

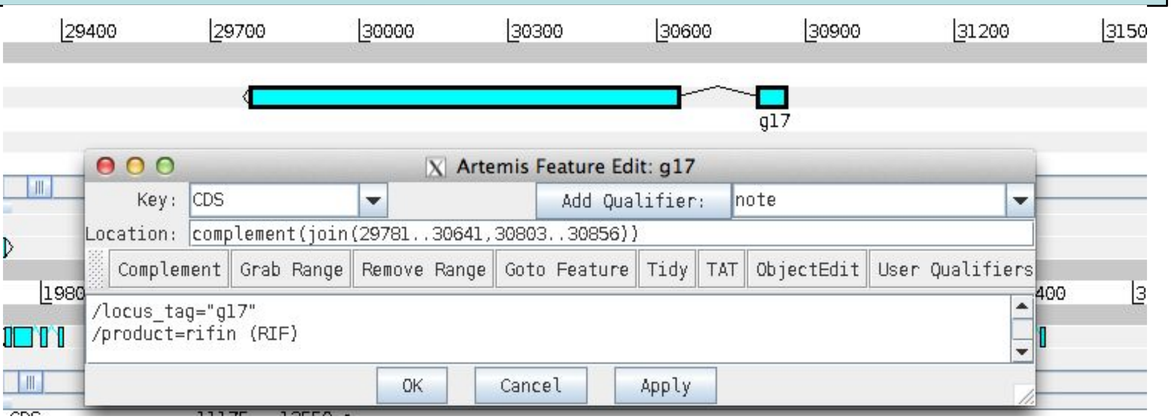
In a web browser, navigate to http://plasmodb.org/

Under the Tools menu, select BLAST. In the web form, select “Proteins”, “blastp”, select all Target Organisms, paste in your sequence (Ctrl-V) and “Get Answer”. It is essential that your BLAST-search parameters match the search you are trying to do (protein versus protein in this case).

If the top hit has a good E-value (e.g. less than 1e-20), select it and copy the description. Then select the gene model, press “Ctrl+e”, and add a new line as below. In the example below Augustus prediction g17 is a gene from the rifin (RIF) family.

```
/product=“rifin (RIF)”
```

Click OK to save the annotation. Now, when someone comes across this gene, they will know what sort of gene it is - well done!



Key: CDS

Add Qualifier: note

Location: complement(join(29781..30641,30803..30856))

```
/locus_tag="g17"
/product=rifin (RIF)
```

2. Mapping RNA-Seq data to a reference

We will use the program TopHat, part of the Tuxedo suite, to map RNA-Seq reads to our references genome, chromosome 5 of *P. falciparum* IT. In this case we are mapping single-end reads generated from RNA extracted during the blood stage of malaria.

TopHat requires an index of the reference. Create the index:

```
bowtie2-build IT.genome.fa IT.genome
```

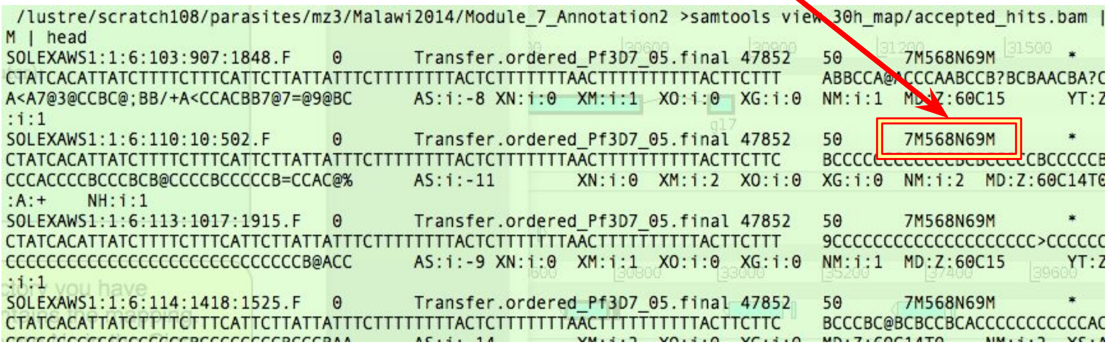
Now run TopHat (n.b. this may take several minutes depending on computer):

```
tophat -o 30h_map -I 10000 IT.genome blood_stage_30h.fastq.gz
```

To read the BAM file:

```
samtools view 30h_map/accepted_hits.bam | head
```

TopHat will generate several files in the new “30h_map” directory you have specified. The most important is accepted_hits.bam. This contains the mapping. Briefly read through the file if you like. For some reads, there are Ns in the Cigar line (column 6, see below). What might these mean?



```
 /lustre/scratch108/parasites/mz3/Malawi2014/Module_7_Annotation2 >samtools view 30h_map/accepted_hits.bam |
M | head
SOLEXAWS1:1:6:103:907:1848.F    0       Transfer.ordered_Pf3D7_05.final 47852   50      7M568N69M       *
CTATCACATTATCTTTTCTTTCATTCTTATTATTTCTTTTTTTTACTCTTTTTTTAACTTTTTTTTTTTACTTCTTT    ABBCCA@ACCCAABCCB?BCBAACBA?C
A<A7@3@CCBC@;BB/+A<CCACBB7@7=@9@BC      AS:i:-8 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:60C15      YT:Z
:i:1
SOLEXAWS1:1:6:110:10:502.F      0       Transfer.ordered_Pf3D7_05.final 47852   50      7M568N69M       *
CTATCACATTATCTTTTCTTTCATTCTTATTATTTCTTTTTTTTACTCTTTTTTTAACTTTTTTTTTTTACTTCTTC    BCCCCCCCCCCCCBCBCCCCCBCCCCCB
CCCACCCCBCCCBCB@CCCCBCCCCCB=CCAC@%      AS:i:-11        XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:60C14T0
:A:+    NH:i:1
SOLEXAWS1:1:6:113:1017:1915.F   0       Transfer.ordered_Pf3D7_05.final 47852   50      7M568N69M       *
CTATCACATTATCTTTTCTTTCATTCTTATTATTTCTTTTTTTTACTCTTTTTTTAACTTTTTTTTTTTACTTCTTT    9CCCCCCCCCCCCCCCCCCCCC>CCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCB@ACC     AS:i:-9 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:60C15      YT:Z
:i:1
SOLEXAWS1:1:6:114:1418:1525.F   0       Transfer.ordered_Pf3D7_05.final 47852   50      7M568N69M       *
CTATCACATTATCTTTTCTTTCATTCTTATTATTTCTTTTTTTTACTCTTTTTTTAACTTTTTTTTTTTACTTCTTC    BCCCBC@BCBCCBCACCCCCCCCCCCAC
```

We have to index the bam using SAMtools, in order to view the data in Artemis.
`samtools index 30h_map/accepted_hits.bam`

The output index file is called `30h_map/accepted_hits.bam.bai`

So that the mapping would go faster, you have only mapped reads to part of the genome. We will now view the much larger, pre-computed file which does cover the whole genome: `blood_stage_30h.bam`

3. Viewing RNAseq mapping in Artemis

We will now examine the read mapping in Artemis using the BAM view feature.

If you have closed the Artemis window, then first type:

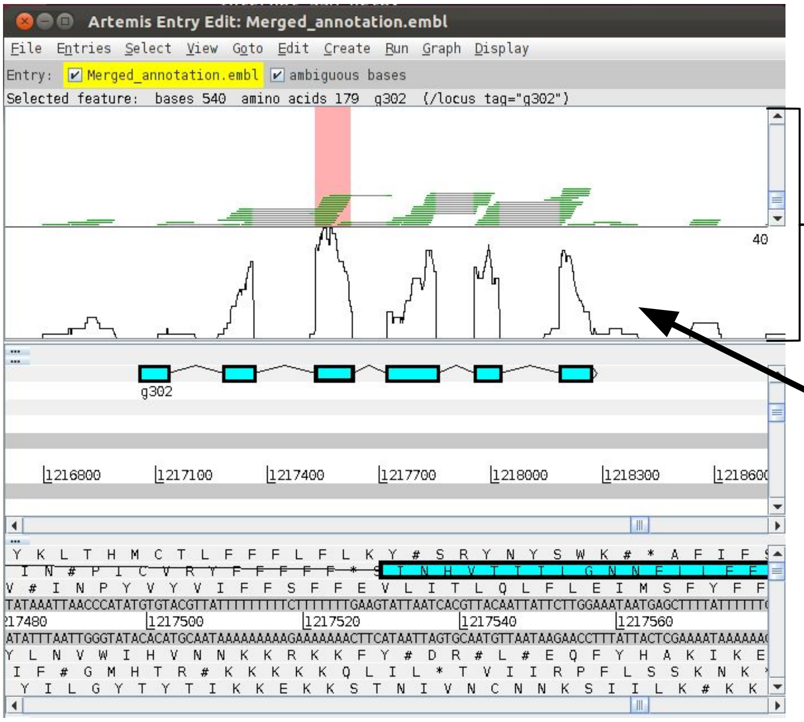
```
art Merged_annotation.embl &
```

In Artemis, highlight gaps in the assembly which might mislead you about the meaning of the RNAseq data:

Create -> Mark Ambiguities

Load the BAM file:
File -> Read BAM / VCF -> Select, “blood_stage_30h.bam” -> “OK”

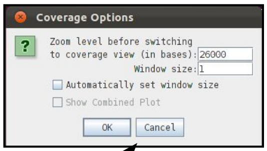
This opens a new window at the top. Right click on the BAMview window, select Graph -> coverage.


Artemis Entry Edit: Merged_annotation.embl
File Entries Select View Goto Edit Create Run Graph Display
40
g302
1216800
1217100
1217400
1217700
1218000
1218300
1218600
BAM view

Entry: ☑ Merged_annotation.embl ☑ ambiguous bases

Selected feature: bases 540 amino acids 179 g302 (/locus_tag="g302")

```
 Y  K  L  T  H  M  C  T  L  F  F  F  L  F  L  K  Y  #  S  R  Y  N  Y  S  W  K  #  *  A  F  I  F
  I  N  #  P  I  C  V  R  Y  F  F  F  F  F  *  S  T  N  H  V  T  T  T  L  G  N  N  F  L  L  F  F
V  #  I  N  P  Y  V  Y  V  I  F  F  S  F  F  E  V  L  I  T  L  Q  L  F  L  E  I  M  S  F  Y  F  F
TATAAATTAACCCATATGTGTACGTTATTTTTTTTTCTTTTTTTGAAGTATTAATCACGTTACAATTATTCTTGGAAATAATGAGCTTTTATTTTTT
217480          1217500         1217520         1217540         1217560
ATATTTAATTGGGTATACACATGCAATAAAAAAAAAGAAAAAAACTTCATAATTAGTGCAATGTTAATAAGAACCTTTATTACTCGAAAATAAAAAA
Y  L  N  V  W  I  H  V  N  N  K  K  R  K  K  F  Y  #  D  R  #  L  #  E  Q  F  Y  H  A  K  I  K  E
 I  F  #  G  M  H  T  R  #  K  K  K  K  K  Q  L  I  L  *  T  V  I  I  R  P  F  L  S  S  K  N  K
  Y  I  L  G  Y  T  Y  T  I  K  K  E  K  K  S  T  N  I  V  N  C  N  N  K  S  I  I  L  K  #  K  K
```



Zoom level before switching to coverage view (in bases): 26000

Window size: 1

☐ Automatically set window size

☐ Show Combined Plot

Right click on the coverage plot and select Options... Set the window size to 1 to see the exon boundaries (You have to disable “Automatically…”)

Examine the exon boundaries of a couple of genes. Can you see how RNA-seq helps us to identify exon boundaries and UTRs?

What do the grey lines in the middle of some of the mapped reads mean? Right click on one of these reads, select “Show details of” and examine the cigar string.

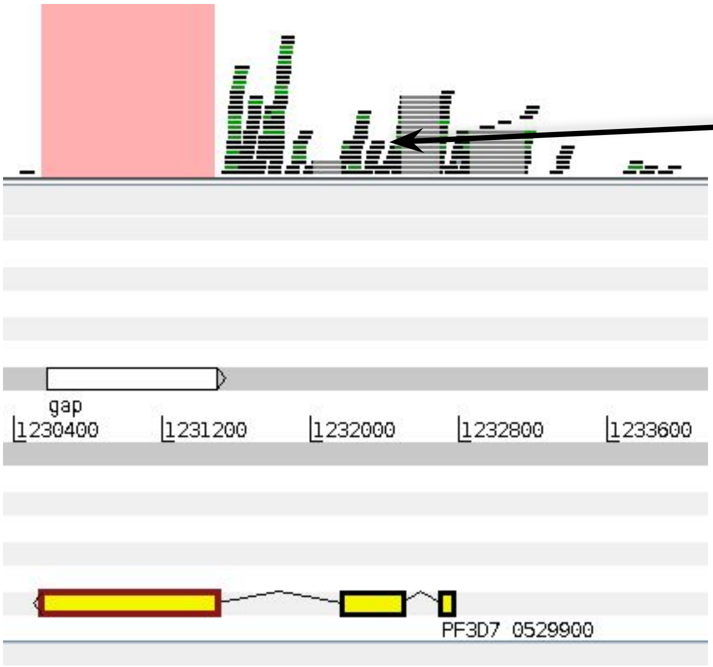
4. Correcting gene models based on RNA-seq data

Scroll along the chromosome and examine the read coverage. How well does it agree with the gene models? Notice how different genes have different numbers of reads mapping. Why do some genes have little or no reads mapping? What does this mean for annotation?

Why do some reads map where there are no genes models?

Scroll along the chromosome. Can you see any gene models which might be incorrect based on the RNAseq data? PF3D7_0529900 is a good example.

Bonus question: Can you figure out why RATT got this gene model so wrong!



Can you see why this model isn’t correct? The reads should map over exons and not introns.

Is the start of the gene model correct? There seem to be spliced reads confirming another exon.

5. Automatically generating gene models based on RNA-seq data

Cufflinks is another program in the Tuxedo suite. It can be used to generate gene models based on the RNA-seq mapping. Rather than fix each gene model by hand, we could replace them with RNA-seq based predictions if these are better.

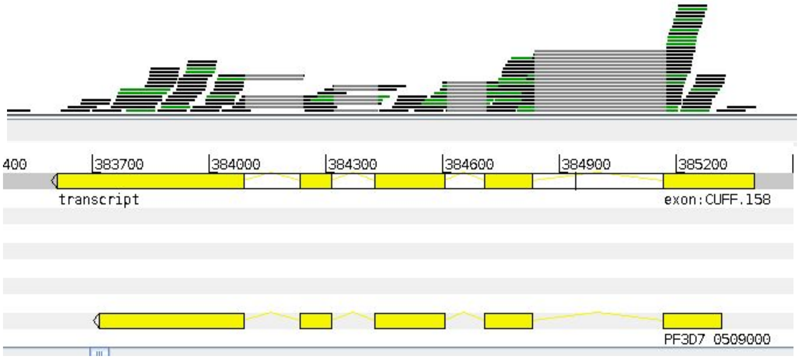
Run Cufflinks:

```
cufflinks blood_stage_30h.bam
```

The key results file from Cufflinks is transcripts.gtf. When the output file `transcripts.gtf` has been created, change the format to display better in Artemis:

Read this into Artemis and compare the results to the RNAseq coverage plots and to the existing gene predictions. In Artemis,
File -> Read An Entry, select "Files of type: All files", select "transcripts.gtf"

Look through the annotation. Does cufflinks confirm the Augustus predictions? What can we say where a gene model has no coverage? How can we remedy this?



Why are the Cufflinks predictions often longer than the gene model predictions? Remember we are examining a eukaryote. Why does this make it difficult to incorporate these models directly into our gene set? Would this explain the spliced reads of the gene PF3D7_0529900 on page 11? What does the Cufflinks gene model look like?

6. Using RNA-seq to improve predictions

Perhaps the best option would be to use the RNA-seq to guide the *ab initio* predictions? Augustus allows you to to create hints files from RNA-seq. These files tell Augustus about intron/exon boundaries observed in the RNA-seq data. You can read more about how to create your own hints on augustus help pages: augustus.gobics.de/binaries/readme.rnaseq.html

On the command line, you first have to copy a suitable configuration-file from the augustus folder:

```
cp /opt/augustus-3.2.1/config/extrinsic/extrinsic.M.RM.E.W.cfg
.
```

Make RNA-seq hints from the bam-file. On the command-line, type:

```
bam2augustusHints.pl blood_stage_30h.bam > hints.introns.gff
```

Now predict new models using RNA-seq hints, you'll notice it takes a bit longer than running without hints, expect a few minutes. On the command-line, type (as one line).

```
augustus --species=pfalciparum
  --extrinsicCfgFile=extrinsic.M.RM.E.W.cfg
  --hintsfile=hints.introns.gff IT.genome.fa >
augustus.hints.gtf
```

Convert the Augustus GFF to EMBL format, so you can view it nicely in Artemis. On the command line, type:

```
cat augustus.hints.gtf | augustus2embl.pl >
augustus.hints.embl
```

Load the new predictions into Artemis like you did before, and compare this prediction to your earlier prediction without hints, to see which one you think is better. Look for instance at PF3D7_0528500. What additional information is Augustus now able to uncover?

Tip: if you want to see the difference between the models more clearly, you can in Artemis un-tick all files except for augustus.hints.embl at the bar at the top, choose Select -> "All CDS features". Click Edit -> "Qualifier of selected feature(s)" -> Change. In the drop-down menu in the pop-up box choose "colour", and then click "Insert qualifier:". Change the text in the box to /colour=3 and click "Add". All your models predicted using hints are now green.

Optional 1: Learn more augustus; a) learn how to train augustus with your favorite species from the online manual, b) check out the useful scripts that come with augustus; /usr/local/augustus.2.7/scripts/ , c) What does the Augustus option --alternatives-from-evidence do?

Optional 2: Try to look at all the differences between augustus predictions with and without hints: 1. using programming, 2. in Artemis. How would you choose which predictions are the best? If you had to write a script to automatically choose between the models, what would that script contain?

Key aspects of genome annotation

Quality

The better the genome prediction is, the better the gene-models will be. If the genome is miss-assembled that can lead to partial or chimeric gene-models. The gold standard for gene-models is manual gene-model curation, but for draft genomes there often not resources available to do this. So for draft genomes you may have to accept that you will not have a perfect set of genes. Ten years of annotating the malaria genome by hand using all possible lines of evidence has not resulted in a perfect annotation (although it is a very good one)! How do you think the quality of the gene-models affect the analysis you can do, and the conclusions you can draw?

Several lines of independent evidence are best

Predicting gene-models, you will find that one of the hardest things to do is to choose which set of models are the best; all methods are good at some types of genes and bad at others. Augustus has a built-in quality check, in which you can compare your training models with your predicted models. Use a variety of prediction approaches, as you have done here, to capture as many of the genes as possible and to improve their accuracy. It is always a good idea to try different programs for any particular problem in computational biology: if they all produce the same answer you can be more certain it is correct. In the case of gene model predictions they will frequently disagree. If several predictions are of similar high quality, perhaps the best option is to combine different sets of gene using tools such as Jigsaw (Allen & Salzberg, 2005) and EVM (Haas et al., 2008)?

Over-prediction

There is a balance to strike between having almost all the genes and lots of erroneous ones as well, or to miss some genes but have relatively few incorrect ones. It may be important to find as many of the real genes as possible. However once you have published an erroneous model to the public sphere, for instance by submitting it to GenBank, it can be very hard to retract it later, and it may cause problems for other people using that gene for their analyses.

Bacteria are simpler

If you work on bacteria you will encounter fewer problems with accurately predicting gene models as they almost always have single-exon genes. The program Glimmer3 (Delcher et al., 1999) is an alternative to Augustus for *ab initio* gene prediction in bacteria, but since version 2.7 Augustus has improved its prediction methods for bacterial genomes. Another alternative for both gene prediction and functional annotation for bacteria is Prokka www.vicbioinformatics.com/software.prokka.shtml

Alternative splicing

Many genes in more complex organisms have several alternative splice-forms, including/excluding different UTRs and exons. Predicting alternative splicing is much harder than predicting a “canonical”; most complete, gene-model (like the ones you have just predicted). It is currently only possible to predict alternative transcripts reliably for model genomes which already are very well assembled and annotated, and have a wealth of supporting evidence (like human and *C. elegans*).

Key aspects of RNA-seq mapping

Non-unique/repeat regions

A sequence read may map equally well to multiple locations in the reference genome. Different mapping algorithms have different strategies for this problem, so be sure to check the options in the mapper. A low GC content, such as in *Plasmodium falciparum* (81% AT) means that reads are more likely to map to multiple locations in the genome by chance.

Insert size

When mapping paired reads, the mapper (e.g. TopHat/Hisat2) takes the expected insert size into account. If the fragments are expected to be 200bp on average, and the reads are 50bp, then the insert between the paired reads should be ~100bp. If the paired reads are significantly further apart than expected, we can suspect that the reads have not mapped properly and discard them. Removing poorly mapping reads can produce a more reliable mapping.

Spliced mapping

Eukaryotic mRNAs are processed - the introns are spliced out after transcription. Therefore some reads (those crossing exon boundaries) should be split when mapped to the reference genome sequence in which intron sequences are still present. TopHat, Hisat2, STAR and Gmap are examples of mappers which can split reads while mapping them, making them suitable for mapping RNA-seq.

Alternative mappers

Alternative short read mappers which do not split reads include SOAP (Li et al., 2008b), SSAHA (Ning et al., 2001), BWA (Li et al., 2009) and Bowtie2 (Langmead B, Salzberg S. 2012), SMALT (https://www.sanger.ac.uk/science/tools/smalt-0). All of these may be appropriate for bacterial RNA-seq.

New tools for mapping sequence reads are continually being developed. This reflects improvements in mapping technology, but it is also due to changes in the sequence data to be mapped. Long reads from PacBio and Oxford Nanopore platforms are being used more and more. The tools best for these reads include minimap2 and Gmap.

Beware of the genes!

In spite of our very best efforts, it is not always possible to predict genes accurately. There are many phenomena which can throw both automatic and manual predictions off track. For instance (but not limited to): seleno-proteins containing "stop-codons" as part of the coding sequence, polycistronic genes, splice-leader trans-splicing, long non-coding RNAs, repetitive genomic regions and pseudogenes. Before publishing, it is always a good idea to try to estimate how correct the models are, and do some sanity-checking of the gene-models.

Differential Expression

Introduction to Differential Expression analysis

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 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 bacterium uses its genome differently when exposed to stress, such as excessive heat or a drug. Alternatively we could ask what genes make human livers different from human kidneys.

In this module we will address the effect of vector transmission on gene expression of the malaria parasite. Is the transcriptome of a mosquito-transmitted parasite different from one which has not passed through a mosquito? The key reason for asking this question is that parasites which are transmitted by mosquito are less virulent than those which are serially blood passaged in the laboratory. Figure 1A shows the malaria life cycle, the blue part highlighting the mosquito stage. Figure 1B shows the difference in virulence, measured by blood parasitemia, between mosquito-transmitted and serially blood passaged parasites. The data in this exercise, as well as figures 1B and 1C are taken from Spence et al. (2013).

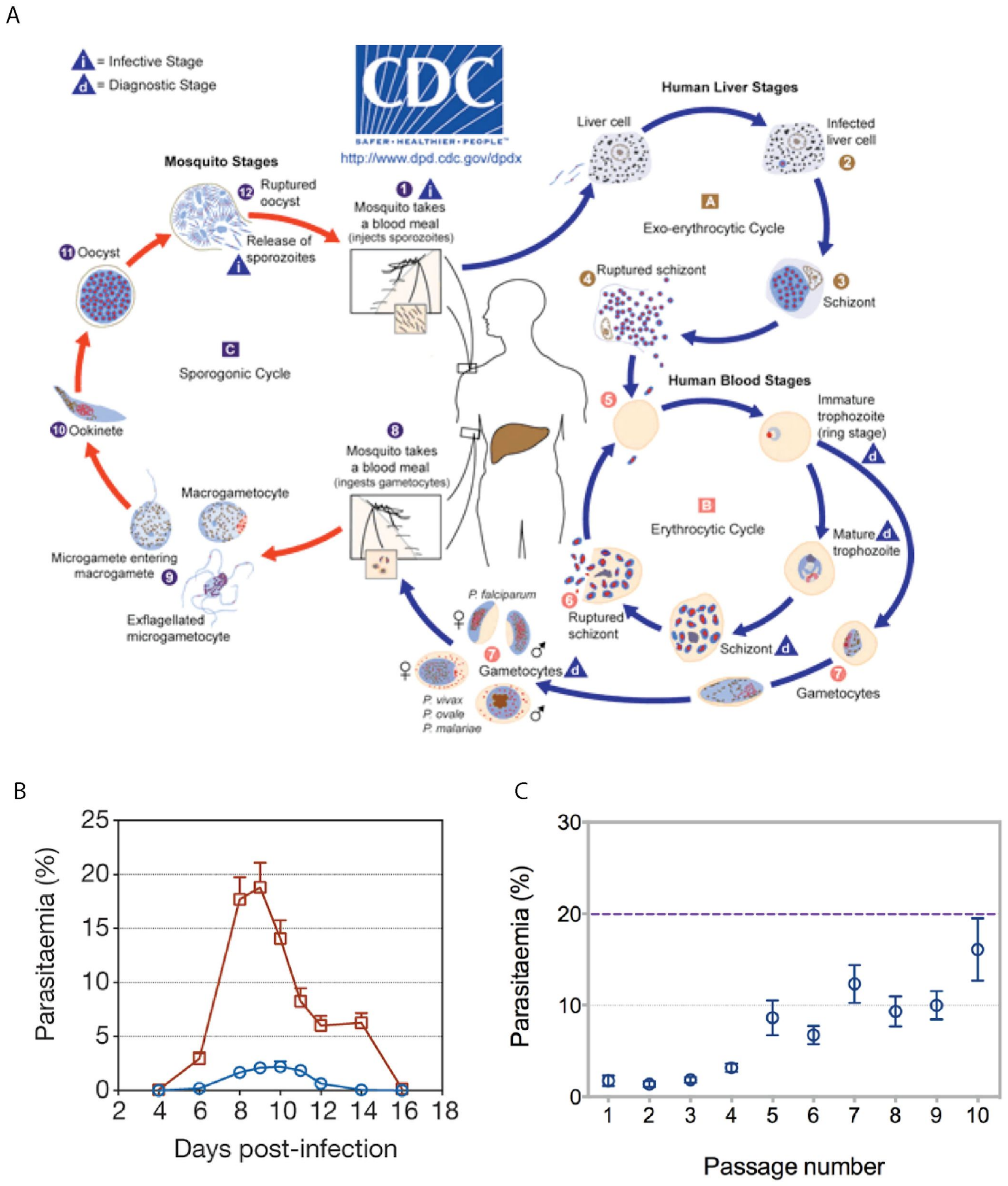


Figure 1. Serial blood passage increases virulence of malaria parasites. (A) The lifecycle of plasmodium parasites involves mammalian and mosquito stages. Experiments in the lab often exclude the mosquito stage (red) and instead remove parasites from the blood of a mouse to infect another mouse (serial blood passage). (B) Serially blood passaged parasites (red) are more virulent than mosquito-transmitted parasites (blue) as shown by their higher parasitemia over the course of infection. (C) As mosquito transmitted parasites are serially blood passaged an increasing number of times, they return to a higher level of parasitemia.

Figure 1C shows that increasing numbers of blood passage post mosquito transmission results in increasing virulence, back to around 20% parasitemia. Subsequent mosquito transmission of high virulence parasites renders them low virulence again. We hypothesise that parasites which have been through the mosquito are somehow better able to control the mosquito immune system than those which have not. This control of the immune system would result in lower parasitemia because this is advantageous for the parasite. Too high a parasitemia is bad for the mouse and therefore bad for the parasite. Are there any differences between the transcriptomes of serially blood passaged parasites and mosquito-transmitted parasites which might explain how they are able to do this?

Module Summary

1. Mapping RNA-seq reads to the genome using *HISAT2*
2. Using *Artemis* to visualise transcription
3. Using *Kallisto* and *Sleuth* to identify differentially expressed genes
4. Using *Sleuth* to quality check the data
5. Interpreting the results

1. Mapping RNA-seq reads to the genome using *HISAT2*

We have two conditions: serially blood-passaged parasites (SBP) and mosquito transmitted parasites (MT). One with three biological replicates (SBP), one with two (MT). Therefore we have five RNA samples, each which has been sequenced on an Illumina HiSeq sequencing machine. For this exercise we have reduced the number of reads in each sample to around 2.5m to reduce the mapping time. However this will be sufficient to detect most differentially expressed genes.

Firstly, make a HISAT2 index for the *P. chabaudi* genome reference sequence.

```
hisat2-build PccAS_v3_genome.fa PccAS_v3_hisat2idx
```

Map the reads for the MT1 sample using HISAT2. Each of the following steps will take a couple of minutes.

```
hisat2 --max-intronlen 10000 -x PccAS_v3_hisat2idx -1
MT1_1.fastq -2 MT1_2.fastq -S MT1.sam
```

Convert the SAM file to a BAM.

```
samtools view -b -o MT1.bam MT1.sam
```

Sort the BAM file (otherwise the indexing won't work)

```
samtools sort -o MT1_sorted.bam MT1.bam
```

Index the BAM file so that it can be read efficiently by Artemis

```
samtools index MT1_sorted.bam
```

Now map, convert SAM to BAM, sort and index with the reads from the MT2 sample.

Note the BAM files and .bai index files provided for the SBP samples:

```
ls *bam*
```

2. Using *Artemis* to visualise transcription

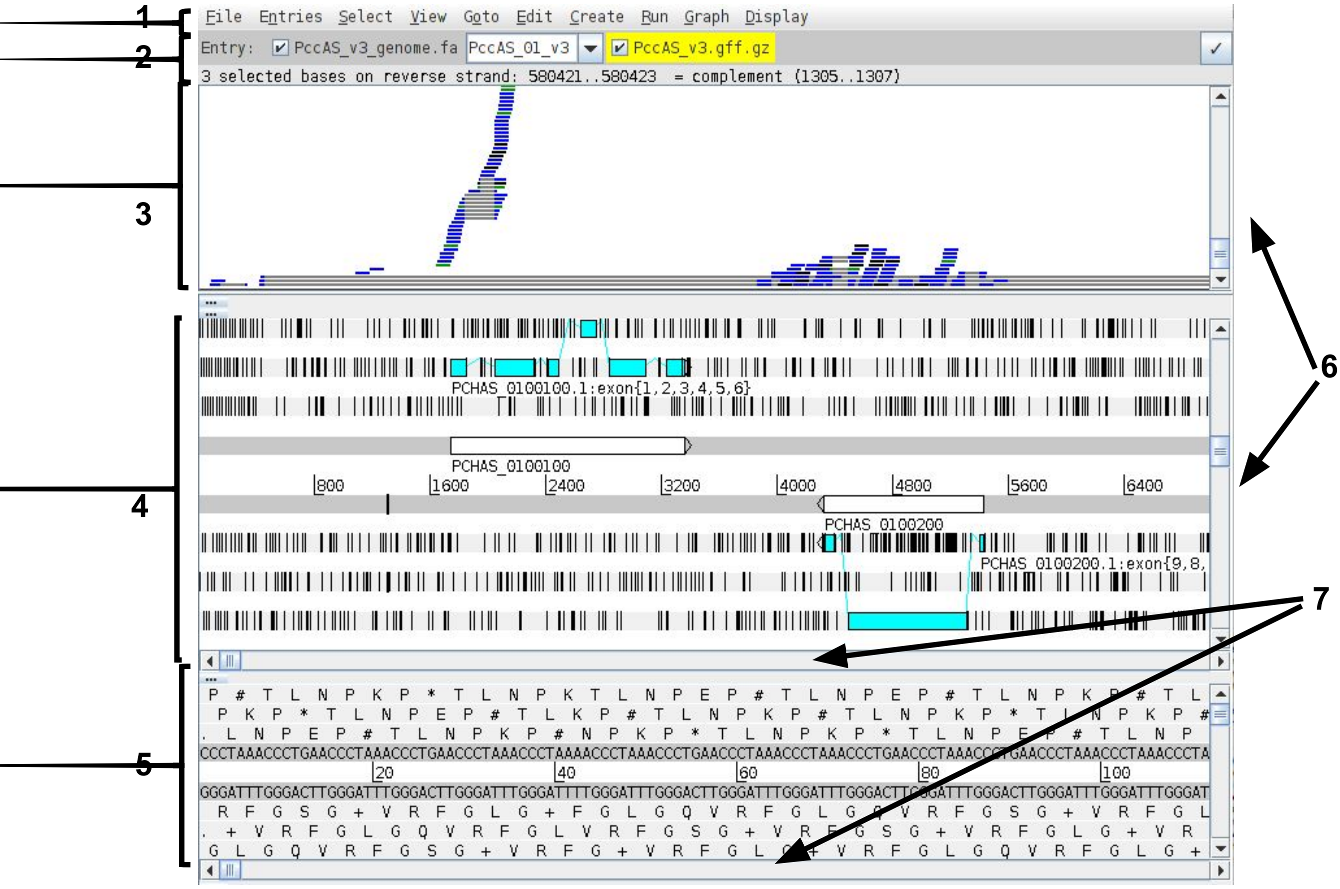
Index the fasta file so Artemis can view each chromosome separately

```
samtools faidx PccAS_v3_genome.fa
```

Load chromosome 14 into Artemis from the command line, displaying the mapped reads from each sample:

```
art
-Dbam="MT1_sorted.bam,MT2_sorted.bam,SBP1_sorted.bam,SBP2_sorted.
bam,SBP3_sorted.bam" PccAS_v3_genome.fa +PccAS_v3.gff.gz &
```

Select ”Use index” so Artemis will show individual chromosomes.


1
2
3
4
5
6
7
File Entries Select View Goto Edit Create Run Graph Display
PCHAS_0100100.1:exon{1,2,3,4,5,6}
PCHAS_0100100
800
1600
2400
3200
4000
4800
5600
6400
PCHAS_0100200
PCHAS_0100200.1:exon{9,8,
20
40
60
80
100

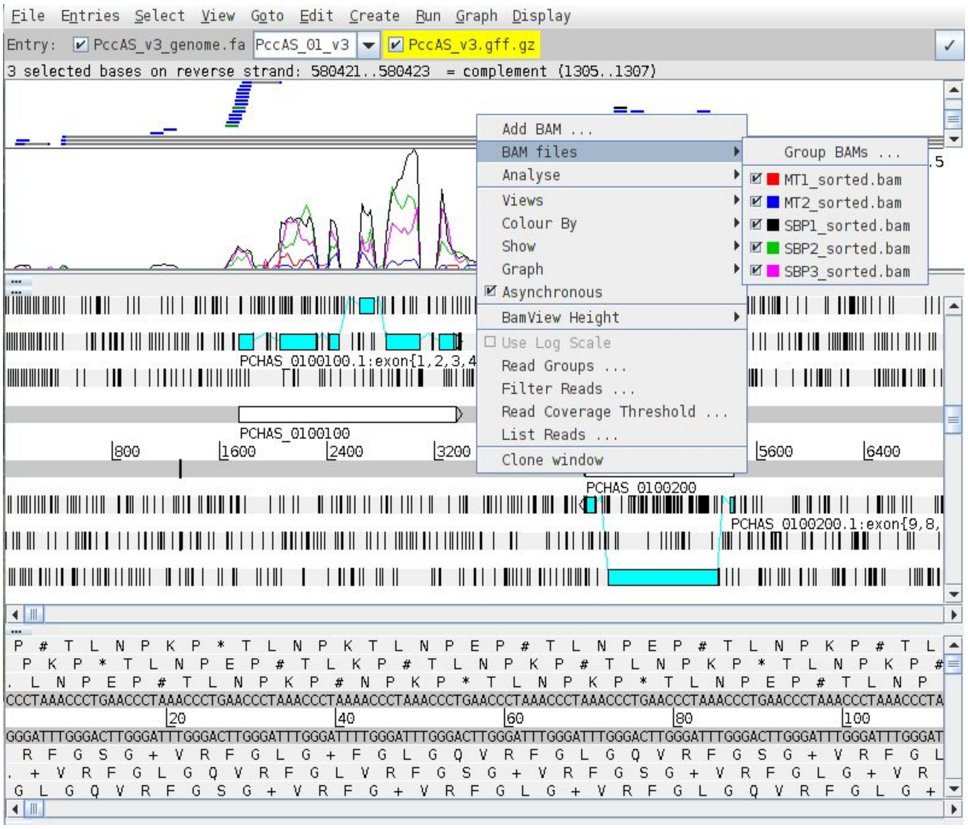
Entry: ☑ PccAS_v3_genome.fa PccAS_01_v3 ☑ PccAS_v3.gff.gz

3 selected bases on reverse strand: 580421..580423 = complement (1305..1307)

1. **Drop-down menus**
2. **Entry (top line):** shows which entries are currently loaded with the default entry highlighted in yellow You can select different chromosomes to view here.
3. **BAM view**: Displays reads mapped to the genome sequence. Each little horizontal line represents a sequencing read. Some reads are blue indicating that they are unique reads. Green reads represent multiple reads mapped to exactly the same position on the reference sequence. Grey lines in the middle of reads mean that the read has been split and this usually means it maps over an intron. If you click a read its mate pair will also be selected. If you want to know more about a read right-click and select ‘Show details of: READ NAME’.
4. **Sequence view panel**. The central two grey lines represent the forward (top) and reverse (bottom) DNA strands. Above and below these are the three forward and three reverse reading frames (theoretical translations of the genome). Stop codons are marked as black vertical bars. Genes and other annotated features are displayed as coloured boxes. We often refer to predicted genes as coding sequences or CDSs.
5. This panel has a similar layout to the main panel but is zoomed in to show nucleotides and amino acids.
6. **Sliders** for zooming view panels.
7. **Sliders** for scrolling along the DNA.

Right click on the BAM view, select *Graph*, then *Coverage*.

Right click on the BAM window showing the reads and hover over *BAM files*. This will show you which colours in the coverage plot relate to which samples. Scroll through the chromosome and see if you can identify genes which might be differentially expressed between SBP and MT parasites. Is looking at the coverage plots alone a reliable way to assess differential expression? Hint: what is the difference between read count and RPKM? Are the libraries all the same size?



```
3 selected bases on reverse strand: 580421..580423 = complement (1305..1307)
```

```
 P  #  T  L  N  P  K  P  *  T  L  N  P  K  T  L  N  P  E  P  #  T  L  N  P  E  P  #  T  L  N  P  K  P  #  T  L
  P  K  P  *  T  L  N  P  E  P  #  T  L  K  P  #  T  L  N  P  K  P  #  T  L  N  P  K  P  *  T  L  N  P  K  P  #
.  L  N  P  E  P  #  T  L  N  P  K  P  #  N  P  K  P  *  T  L  N  P  K  P  *  T  L  N  P  E  P  #  T  L  N  P
CCCTAAACCCTGAACCCTAAACCCTGAACCCTAAACCCTAAAACCCTAAACCCTGAACCCTAAACCCTAAACCCTGAACCCTAAACCCTGAACCCTAAACCCTAAACCCTA
GGGATTTGGGACTTGGGATTTGGGACTTGGGATTTGGGATTTTGGGATTTGGGACTTGGGATTTGGGATTTGGGACTTGGGATTTGGGACTTGGGATTTGGGATTTGGGAT
  R  F  G  S  G  +  V  R  F  G  L  G  +  F  G  L  G  Q  V  R  F  G  L  G  Q  V  R  F  G  S  G  +  V  R  F  G  L
.  +  V  R  F  G  L  G  Q  V  R  F  G  L  V  R  F  G  S  G  +  V  R  F  G  S  G  +  V  R  F  G  L  G  +  V  R
 G  L  G  Q  V  R  F  G  S  G  +  V  R  F  G  +  V  R  F  G  L  G  +  V  R  F  G  L  G  Q  V  R  F  G  L  G  +
```

Select chromosome PccAS_14_v3 from the drop down box on the Entry line.

Press Ctrl-g and use “Goto Feature With Gene Name” to navigate to the gene PCHAS_1402500.

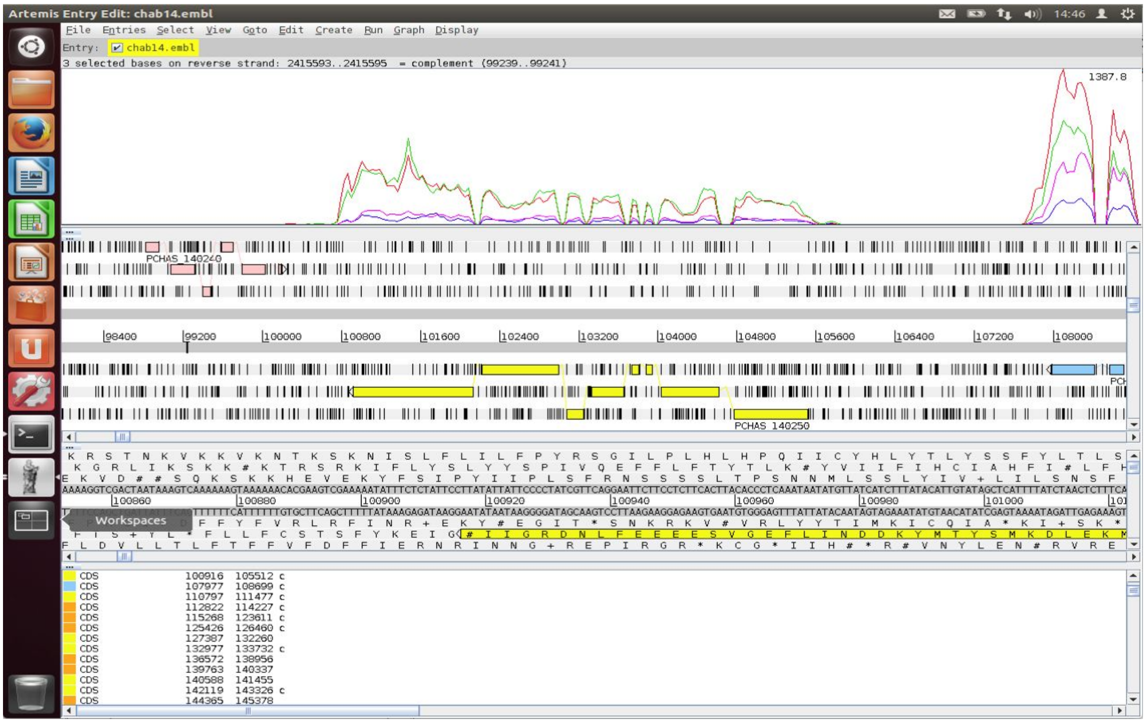
Investigate the coverage for this gene. Does the RNA-seq mapping agree with the gene model in blue?

You can determine read counts and RPKMs for individual genes within Artemis.

Click on the blue gene model, right click on the BAMview window, select *Analyse*, then *RPKM value of selected features*.

Artemis asks whether you want to include introns in the calculations. We are only interested in reads mapping to the spliced transcript, so you should exclude these. Select *Use reads mapped to all reference sequences*. Is this important?

After the analysis is done a window will appear behind Artemis.



```
3 selected bases on reverse strand: 2415593..2415595  = complement (99239..99241)
```

```
CDS          100916  105512 c
CDS          107977  108699 c
CDS          110797  111477 c
CDS          112822  114227 c
CDS          115268  123611 c
CDS          125426  126460 c
CDS          127387  132260
CDS          132977  133732 c
CDS          136572  138956
CDS          139763  140337
CDS          140588  141455
CDS          142119  143326 c
CDS          144365  145378
```

This gene looks to be up-regulated in serially blood passaged parasites; SBP samples have RPKMs several times greater than the MT samples. Is it statistically significant? In the next section we will find out.

3. Using *Kallisto* 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 pseudoalignment rather than true read alignment.

We first need to make a fasta file containing out transcript sequences, which we can extract from our reference genome using a gff annotation file

```
gffread -g PccAS_v3_genome.fa -w PccAS_v3_transcripts.fa
PccAS_v3.gff3
```

Kallisto needs an index of the transcript sequences.

```
kallisto index -i PccAS_v3_kallisto PccAS_v3_transcripts.fa
```

Quantify the expression levels of your transcripts for the MT1 sample.

```
kallisto quant -i PccAS_v3_kallisto -o MT1 -b 100 MT1_1.fastq
MT1_2.fastq
```

The results are contained in the file MT1/abundance.tsv

Use the *kallisto quant* command four more times, for the MT2 sample and the three SBP samples.

Sleuth uses the output from *Kallisto* to determine differentially expressed genes. It is written in the R statistical programming language, as is a lot of other RNA-seq analysis software. Helpfully, 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 not as hard to get started with as full-blown programming languages such as 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. 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
```

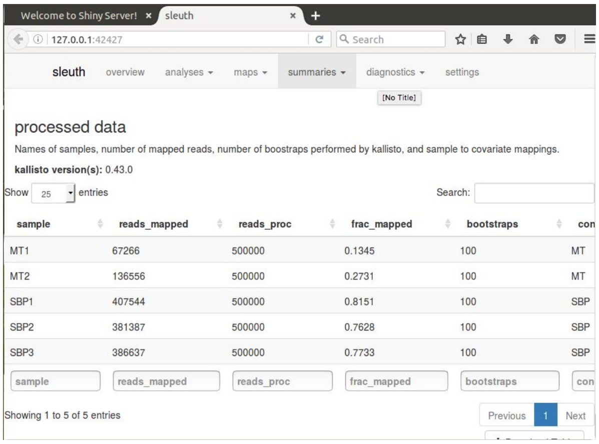
And then copy and paste commands from the file *sleuth.R*

4. 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?



processed data

Names of samples, number of mapped reads, number of boostraps performed by kallisto, and sample to covariate mappings.

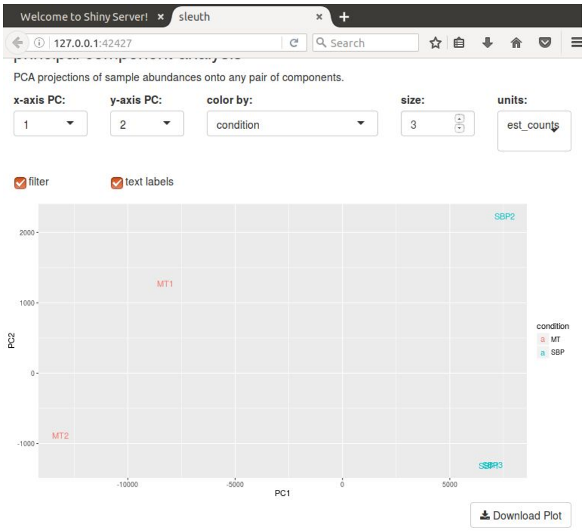
kallisto version(s): 0.43.0

Show 25 entries

Search:

sample	reads_mapped	reads_proc	frac_mapped	bootstraps	con
MT1	67266	500000	0.1345	100	MT
MT2	136556	500000	0.2731	100	MT
SBP1	407544	500000	0.8151	100	SBP
SBP2	381387	500000	0.7628	100	SBP
SBP3	386637	500000	0.7733	100	SBP
sample	reads_mapped	reads_proc	frac_mapped	bootstraps	con

Showing 1 to 5 of 5 entries



PCA projections of sample abundances onto any pair of components.

x-axis PC: 1

y-axis PC: 2

color by: condition

size: 3

units: est_counts

☑ filter

☑ text labels

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 MT samples are well separated from the SBP samples, meaning that the replicates are more similar to each other than they are to samples from the different condition. This is good.

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.

5. 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 descriptionof the gene than its id)

With a little Linux 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?

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

The most highly up-regulated genes in MT samples are from the *cir* family. This is a large, malaria-specific gene family which had previously been proposed to be involved in immune evasion (Lawton et al., 2012). Here however we see many of these genes up-regulated in a form of the parasite which seems to cause the immune system to better control the parasite. We now think that expression of these genes is epigenetically reset during mosquito transmission, whereas serially blood passaged parasites end up expressing one particular *cir* gene which results in more rapid growth in this artificial situation. Expressing a wide range of *cir* genes may allow the parasite to cause less harm to its host, maximise the length of infection and increase transmissibility.

Remember PCHAS_1402500? Of course you do. It was the gene we looked at in Artemis that seemed absolutely definitely differentially expressed.

What does Sleuth think about it?

```
grep PCHAS_1402500 kallisto.results | cut -f1,3,4,12
```

Although this gene looked like it was differentially expressed from the plots in Artemis our test did not show it to be so. This might be because some samples tended to have more reads, so based on raw read counts, genes generally look up-regulated in the SBP samples. Alternatively the reliability of only two biological replicates and the strength of the difference between the conditions was not sufficient to be statistically convincing. In the second case increasing the number of biological replicates would give us more confidence about whether there really was a difference.

In this case, the lower number of reads mapping to MT samples mislead us in the Artemis view. Luckily careful normalisation and appropriate use of statistics saved the day!

If you want to read more about the work related to this data it is published: Spence et al. 2013 (PMID: 23719378). A follow up paper explores the role of cir genes in chronic infections: Brugat et al. 2017 (PMID: 28165471).

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.