References

Abbot, J. C. et al. (2005) *Bioinformatics* 21(18) 3665-3666. WebACT – an online companion for the Artemis Comparison Tool.

Allen JE & Salzberg SL (2005). *Bioinformatics* 21: 3596-3603. JIGSAW: integration of multiple sources of evidence for gene prediction.

Alexa A. et al. (2006) *Bioinformatics* 22: 1600-1607. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure.

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

Anders S. HTSeq: Analysing high-throughput sequencing data with Python. 2010. Software. [http://www-huber.embl.de/users/anders/HTSeq/]

Assefa, S. et al. (2009) *Bioinformatics* 25 (15) 1968-9. ABACAS: algorithm-based automatic contiguation of assembled sequences.

Berriman, M., and K. Rutherford (2003) *Brief Bioinform* 4 (2) 124-132. Viewing and annotating sequence data with Artemis.

Bozdech Z. et al. (2003) *PLOS Biol* 1: E5.The transcriptome of the intraerythrocytic developmental cycle of Plasmodium falciparum.

Carver T. J. et al. (2010) *Bioinformatics* (doi:10.1093/bioinformatics/btq010) BamView: Viewing mapped read alignment data in the context of the reference sequence.

Carver T. J. et al. (2005) *Bioinformatics* 21: 3422-3. ACT: the Artemis Comparison Tool.

Conesa A, et al. (2005) *Bioinformatics* 21: 3674-3676. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research.

Delcher AL. et al. (1999) *Nucleic Acids Res* 27: 4636-4641. Improved microbial gene identification with GLIMMER.

Gardner et al. (2002). *Nature* 419(6906):498-511. Genome sequence of the human malaria parasite *Plasmodium falciparum*.

Grant GR, et al. (2011) *Bioinformatics* 27: 2518-2528. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM).

Hacker, J. et al. (1997) *Mol Microbiol* 23: 1089-97. Pathogenicity islands of virulent bacteria: structure, function and impact on microbial evolution.

Haas BJ, et al. (2008). *Genome Biol* 9: R7. Automated eukaryotic gene structure annotation using EVidenceModeler and the Program to Assemble Spliced Alignments.

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

Kozarewa, I., Z. Ning, et al. (2009). *Nature Met* **6**(4): 291-295. Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G+C)-biased genomes.

Langmead et al. (2009). *Genome Biol* 10:R25. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.

Li et al. (2008a). *Genome Res* 18:1851-8. Mapping short DNA sequencing reads and calling variants using mapping quality scores.

Li et al. (2008b). *Bioinformatics* 24(5):713-714. SOAP: short oligonucleotide alignment program.

Li et al. (2009). *Bioinformatics*, 25:1754-60. Fast and accurate short read alignment with Burrows-Wheeler Transform.

Majoros et al. (2003) *Nucleic Acids Res* 31 (13) 3601-3604. GlimmerM, Exonomy and Unveil: three *ab initio* eukaryotic genefinders.

Mortazavi et al. (2008). *Nature Met* 5: 621 – 628. Mapping and quantifying mammalian transcriptomes by RNA-Seq.

Ning et al. (2001). *Genome Res* 10:1725-9. SSAHA: a fast search method for large DNA databases.

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

Otto, T. D., G. P. Dillon, et al. (2011). *Nucleic Acids Res* **39**(9): e57. RATT: Rapid Annotation Transfer Tool.

Otto, T. D., M. Sanders, et al. (2010). *Bioinformatics* **26**(14): 1704-1707. Iterative Correction of Reference Nucleotides (iCORN) using second generation sequencing technology.

Parkhill, J. (2002) *Method Microbiol* 33: 1-26. Annotation of microbial genomes.

Robinson MD, et al. (2010) *Bioinformatics* 26: 139-140.edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.

Rutherford et al. (2000) *Bioinformatics* 16 (10) 944-945. Artemis: sequence visualization and annotation.

Simpson, J. T., K. Wong, et al. (2009). *Genome Res* **19**(6): 1117-1123. ABySS: a parallel assembler for short read sequence data.

Stephens et al. (1998). *Science* 282(5389): 754 – 759. Genome sequence of an obligate intracellular pathogen of humans: *Chlamydia trachomatis*.

Tsai, I. J., T. D. Otto, et al. (2010). *Genome Biol* **11**(4): R41. Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps.

Trapnell et al. (2009). *Bioinformatics* 25(9):1105-1111. TopHat: discovering splice junctions with RNA-Seq.

Wang et al. (2009). *Nat Rev Genet* 10(1):57-63. RNA-Seq: A revolutionary tool for transcriptomics.

Zerbino, D. R. and E. Birney (2008). *Genome Res* **18**(5): 821-829. Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs.

Appendices

Appendix I: Course Virtual Machine (VM) Quick Start Guide

Using a VM enables us to encapsulate the course data and software in such a way that you can still make use of them when you return to your own laboratory.

To use the VM on the USB stick provided, you will first need to download VirtualBox (http://www.virtualbox.org/). This software is required to run the VM on your machine, it is free and available for windows, MacOSX and linux,

For a detailed description of VirtualBox and the installation see the on-line manual (http://www.virtualbox.org/manual/).

Download and Install VirtualBox

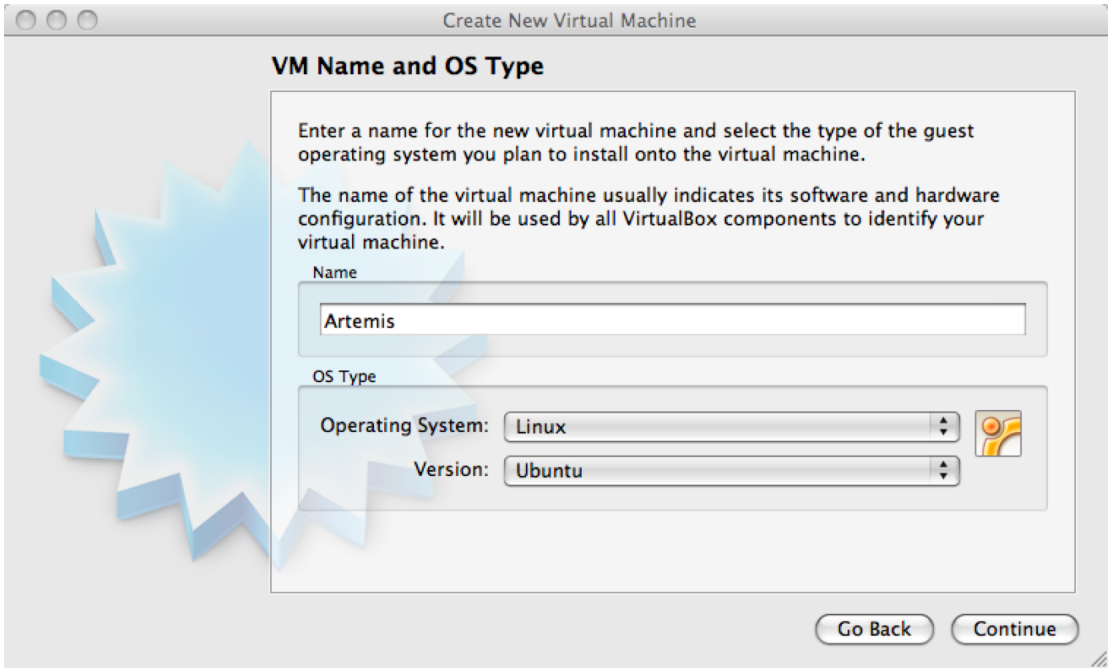
- Download VirtualBox for the type of workstation you are using (e.g. Windows) from http://www.virtualbox.org/wiki/Downloads.
- Double click on the executable file (Windows). The installation welcome dialog opens and allows you to choose where to install VirtualBox to, and which components to install. Depending on your Windows configuration, you may see warnings about "unsigned drivers" or similar. Please select "Continue" on these warnings; otherwise VirtualBox might not function correctly after installation.
- Launch the VirtualBox software from the desktop shortcut or from the program menu.

Setting up the VM

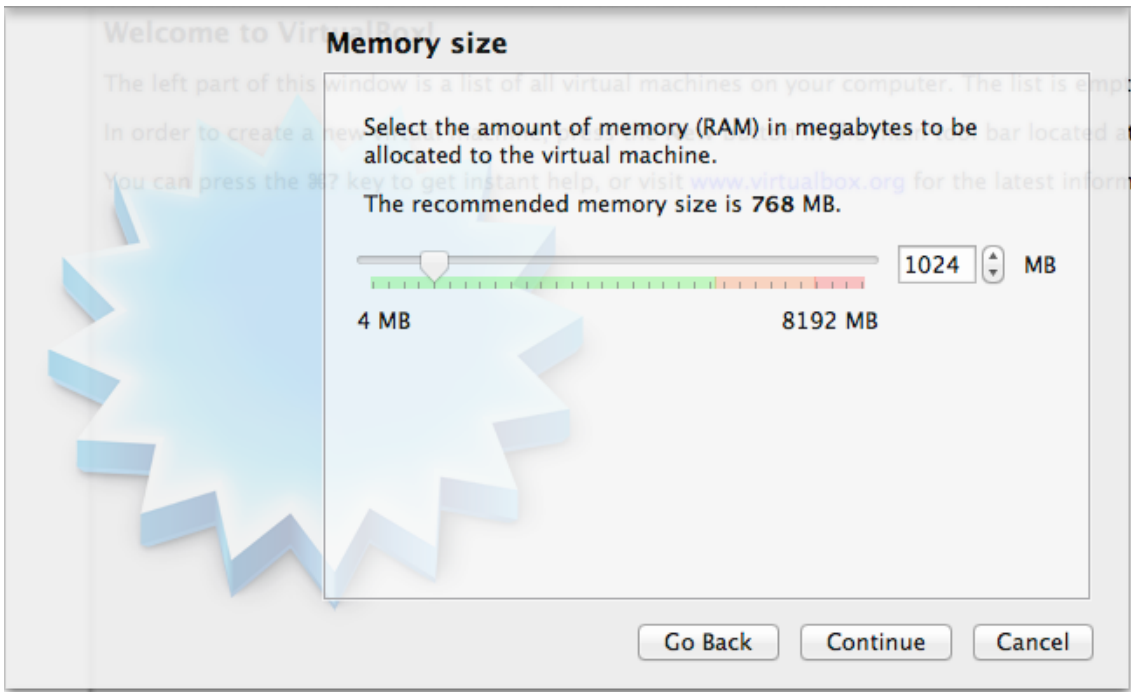
VirtualBox needs to be pointed at the VDI (This is the file that is on the memory stick used during the course) file as follows:

- Insert the USB memory stick provided. This contains a Virtual Disk Image (VDI) file.

Create a new virtual machine by selecting ‘New’ from the options at the top. Then fill the boxes in as shown below:
In the first window enter:
Name: **Artemis**
Operating System: **Linux**
Version: **Ubuntu**



Click ‘Continue’
In the next window set the memory to at least 1GB (as shown), but 2GB (2048 MB) will give you better performance. You can use more but no more than half the amount of memory on your PC.

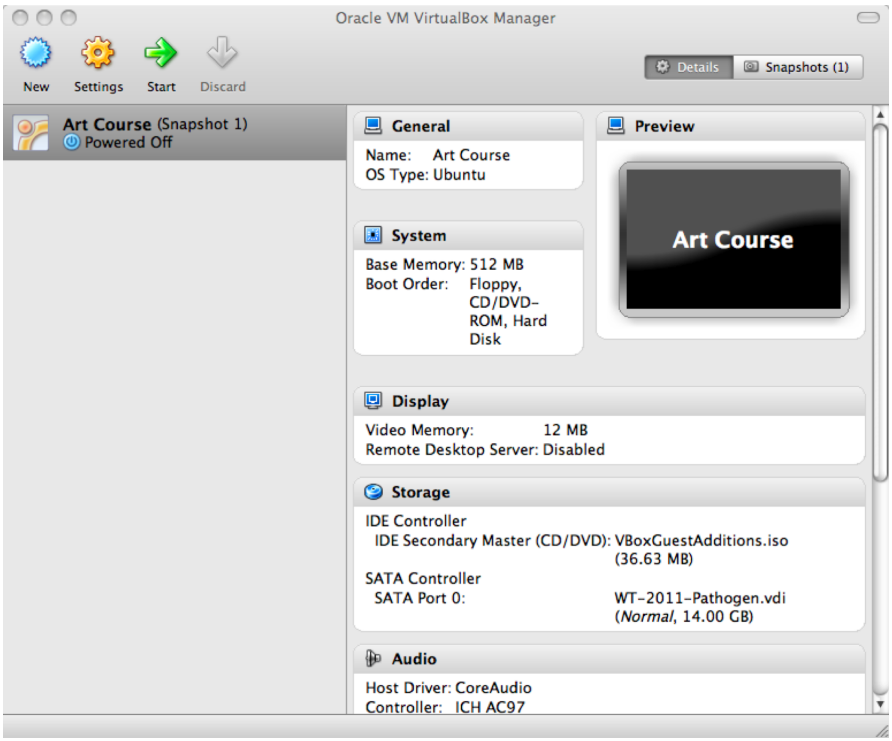


Click ‘Continue’.

In the next window select ‘Use existing hard disk’ and from the folder icon on the right hand side navigate to the memory USB stick and select the VDI file located on the memory stick

Click ‘Continue’.

There will now be an ‘Artemis’ (powered off) button in the left hand side of VirtualBox.



Double click on this new Artemis course power button to start the VM. It will then log you into the Ubuntu desktop.

Setting up a Shared Folder

This allows you to share a folder between the VM and your workstation. This means you can put files that you want to share between the operating systems in this folder.

Create a directory to share called ‘VMshare’ on your machine. With the VM shutdown select the ‘Artemis’ button in VirtualBox and click ‘Settings’ in the top menu bar. Go to ‘Shared Folders’ and select the ‘+’ button on the right. In the ‘Folder Path’ select ‘Other’ and navigate to and select the ‘VMshare’ folder that you have created. Then click on ‘OK’.

When the ‘Artemis’ VM is next started double click on the ‘mount’ icon in your home folder. This will open a window that you need to type the password into:

wt

It will show the contents of this folder in the /home/wt/host directory in Ubuntu.

A note on memory usage:

Some computing processes are very memory hungry. Should you find that your computer processes are killed without a clear reason, one aspect to check is the amount of memory allocated to the VM. The 1024MB you have allocated using this tutorial has been check and should be enough. Nonetheless, the amount of memory allocated to the VM can be changed at any time.

Appendix II: Artemis minimum hardware and software requirements.

Artemis and ACT will, in general, work well on any standard modern machine and with most common operating systems. It is currently used on many different varieties of UNIX and Linux systems as well as Apple Macintosh and Microsoft Windows systems.

Appendix III: ACT comparison files

ACT supports three different comparison file formats:

1) BLAST version 2.2.2 output: The blastall command must be run with the -m 8 flag which generates one line of information per HSP.
2) MegaBLAST output: ACT can also read the output of MegaBLAST, which is part of the NCBI blast distribution.
3) MSPcrunch output: MSPcrunch is program for UNIX and GNU/Linux systems which can post-process BLAST version 1 output into an easier to read format. ACT can only read MSPcrunch output with the -d flag.

Here is an example of an ACT readable comparison file generated by MSPcrunch -d.

```
1399 97.00 940 2539 sequence1.dna 1 1596 AF140550.seq
1033 93.00 9041 10501 sequence1.dna 9420 10880 AF140550.seq
828 95.00 6823 7890 sequence1.dna 7211 8276 AF140550.seq
773 94.00 2837 3841 sequence1.dna 2338 3342 AF140550.seq
```

The columns have the following meanings (in order): score, percent identity, match start in the query sequence, match end in the query sequence, query sequence name, subject sequence start, subject sequence end, subject sequence name.

The columns should be separated by single spaces.

Appendix IV: Feature Keys and Qualifiers – a brief explanation of what they are and a sample of the ones we use.

1 – Feature Keys: They describe features with DNA coordinates and once marked, they all appear in the Artemis main window. The ones we use are:

CDS: Marks the extent of the coding sequence.
RBS: Ribosomal binding site
misc_feature: Miscellaneous feature in the DNA
rRNA: Ribosomal RNA
repeat_region
repeat_unit
stem_loop
tRNA: Transfer RNA

2 – Qualifiers: They describe features in relation to their coordinates. Once marked they appear in the lower part of the Artemis window. They describe the feature whose coordinates appear in the 'location' part of the editing window. The ones we commonly use for annotation at the Sanger Institute are:

/class: Classification scheme we use "in-house" developed from Monica Riley's MultiFun assignments (see Appendix VI).
/colour: Also used in-house in order to differentiate between different types of genes and other features.
/gene: Descriptive gene a name, eg. ilvE, argA etc.
/label: Allows you to label a gene/feature in the main view panel.
/note: This qualifier allows for the inclusion of free text. This could be a description of the evidence supporting the functional prediction or other notable features/information which cannot be described using other qualifiers.
/product: The assigned possible function for the protein goes here.
/pseudo: Matches in different frames to consecutive segments of the same protein in the databases can be linked or joined as one and edited in one window. They are marked as pseudogenes. They are normally not functional and are considered to have been mutated.
/**locus_tag** : Systematic gene number, eg SAS1670, Sty2412 etc.

The list of keys and qualifiers accepted by EMBL in sequence/annotation submission files are list at the following web page:
http://www3.ebi.ac.uk/Services/WebFeat/

Appendix V: Generating ACT comparison files using BLAST

The following pages demonstrate how you can generate your own comparison files for ACT from a stand-alone version of the BLAST software. In Appendix X the NCBI BLAST distribution was downloading onto a PC with Windows XP. The exercises in this module are based on the Linux version of the BLAST software. Although the operating systems are different, the command lines used to run the programs are the same. One of the main differences between the two operating systems is that in Windows the BLAST program command line is run in the DOS Command Prompt window, whereas in Linux it is run from a Xterminal window.

In the exercises below you are going to download two small sequences (plasmids), and for two large sequences (whole genomes). You are then going generate files containing DNA sequences in FASTA format for these sequences, which will then be compared using two different programs from the NCBI BLAST distribution to generate ACT comparison files.

Exercise 1

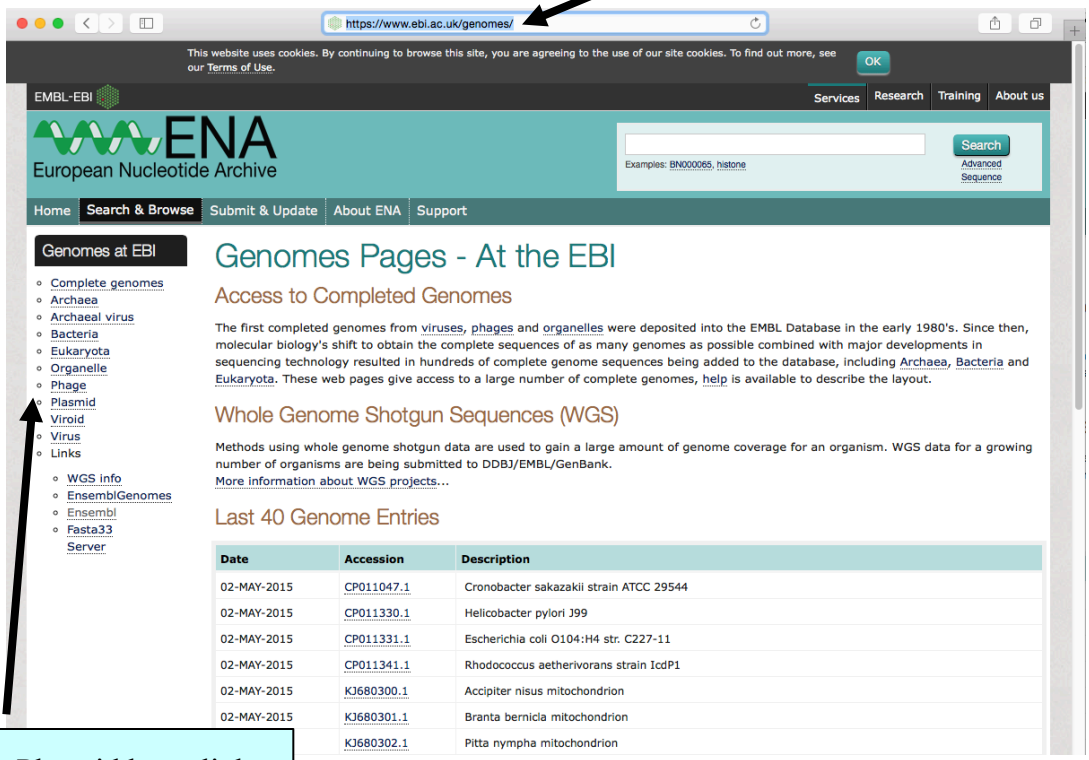
In this exercise you are going to download two plasmid sequences in EMBL format from the EBI genomes web page. You are then going to use Artemis to write out the DNA sequences of both plasmids in FASTA format. These two FASTA format sequences will then be compared using the blastall program from the NCBI BLAST distribution. Using blastall you can run BLASTN to identify regions of DNA-DNA similarity and write out a ACT readable comparison file. If required, blastall can also used to run other flavours of BLAST with the appropriate input files (i.e. DNA files for TBLASTX, protein files for BLASTP, and protein and DNA for BLASTX). For the purposed of generating ACT comparison files BLASTN and TBLASTX are appropriate.

In this example two relative small sequences have been chosen (<500 kb). BLAST running on a relatively modern stand alone machine can easily deal with required computations, and thus the comparison file should be produced in a matter of seconds. However as the size of the compared sequences increases the time taken to produce the output will dramatically increase. Therefore for very large sequences (several Mb) it will be impractical to run them using blastall. In **Exercise 2** you will use megablast, another program in the NCBI BLAST distribution, which is useful for comparing large sequence that are very similar.

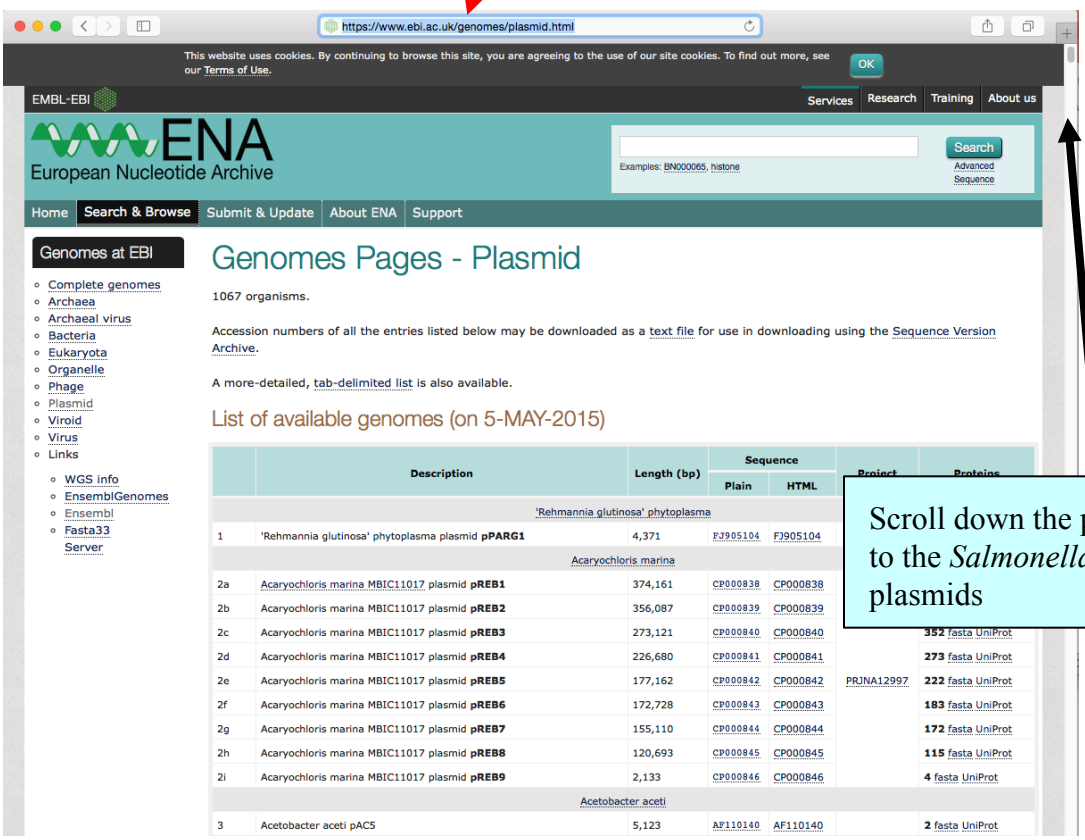
The plasmids chosen for this comparison are the multiple drug resistance incH1 plasmid pHCM1 from the sequenced strain of *Salmonella typhi* CT18 originally isolated in 1993, and R27, another incH1 plasmid first isolated from *S. typhi* in the 1960s.

Downloading the *S. typhi* plasmid sequences

Go to the EBI genomes web page (http://www.ebi.ac.uk/genomes)

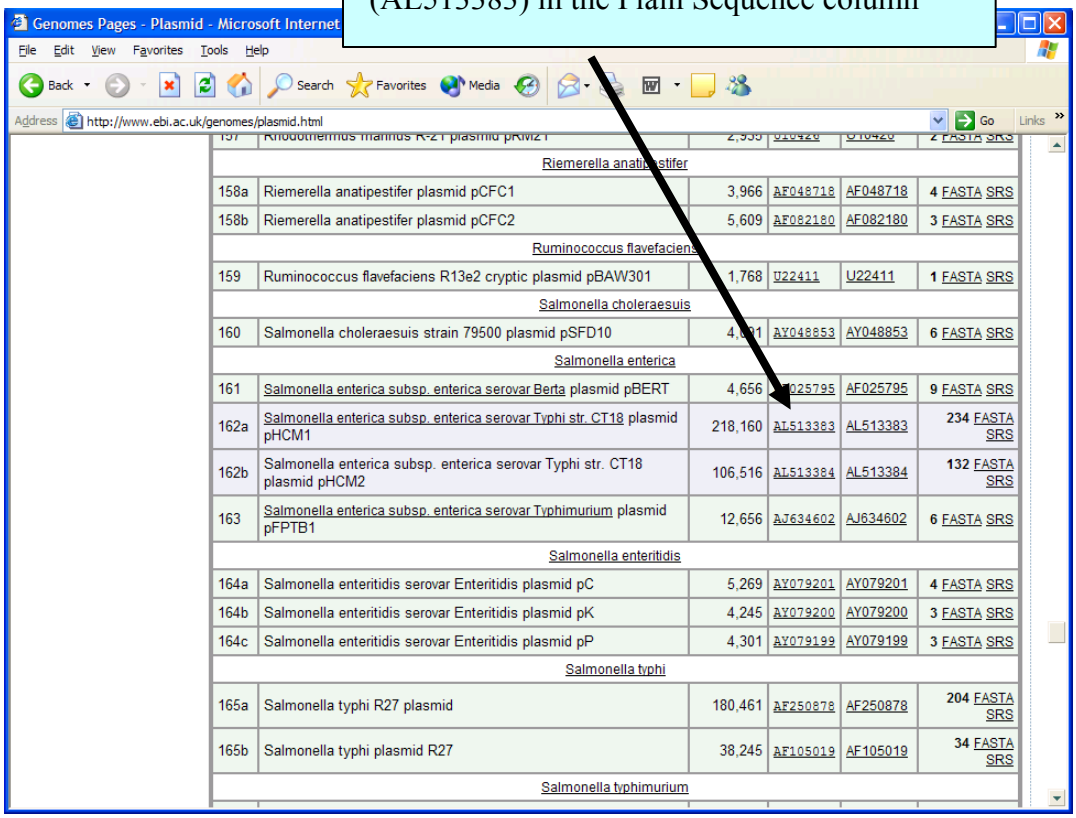


Click on the Plasmid hyperlink

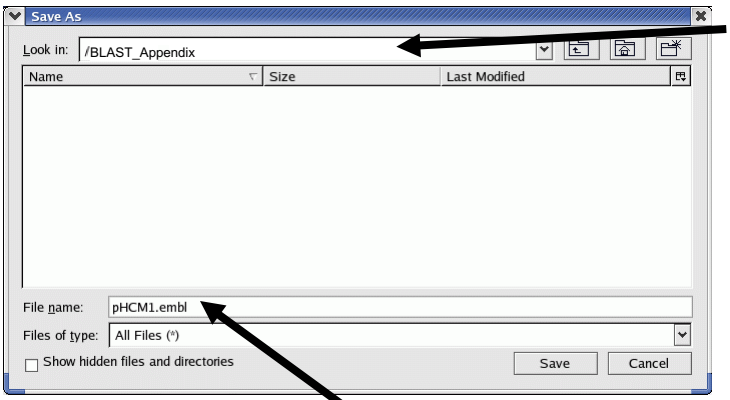


Scroll down the page to the *Salmonella* plasmids

Press the Shift key and left Click on the accession number hyperlink for pHCM1 (AL513383) in the Plain Sequence column



	Riemerella anatipestifer				
158a	Riemerella anatipestifer plasmid pCFC1	3,966	AF048718	AF048718	4 FASTA SRS
158b	Riemerella anatipestifer plasmid pCFC2	5,609	AF082180	AF082180	3 FASTA SRS
	Ruminococcus flavefaciens				
159	Ruminococcus flavefaciens R13e2 cryptic plasmid pBAW301	1,768	U22411	U22411	1 FASTA SRS
	Salmonella choleraesuis				
160	Salmonella choleraesuis strain 79500 plasmid pSFD10	4,[illegible]	AY048853	AY048853	6 FASTA SRS
	Salmonella enterica				
161	Salmonella enterica subsp. enterica serovar Berta plasmid pBERT	4,656	[illegible]025795	AF025795	9 FASTA SRS
162a	Salmonella enterica subsp. enterica serovar Typhi str. CT18 plasmid pHCM1	218,160	AL513383	AL513383	234 FASTA SRS
162b	Salmonella enterica subsp. enterica serovar Typhi str. CT18 plasmid pHCM2	106,516	AL513384	AL513384	132 FASTA SRS
163	Salmonella enterica subsp. enterica serovar Typhimurium plasmid pFPTB1	12,656	AJ634602	AJ634602	6 FASTA SRS
	Salmonella enteritidis				
164a	Salmonella enteritidis serovar Enteritidis plasmid pC	5,269	AY079201	AY079201	4 FASTA SRS
164b	Salmonella enteritidis serovar Enteritidis plasmid pK	4,245	AY079200	AY079200	3 FASTA SRS
164c	Salmonella enteritidis serovar Enteritidis plasmid pP	4,301	AY079199	AY079199	3 FASTA SRS
	Salmonella typhi				
165a	Salmonella typhi R27 plasmid	180,461	AF250878	AF250878	204 FASTA SRS
165b	Salmonella typhi plasmid R27	38,245	AF105019	AF105019	34 FASTA SRS
	Salmonella typhimurium				



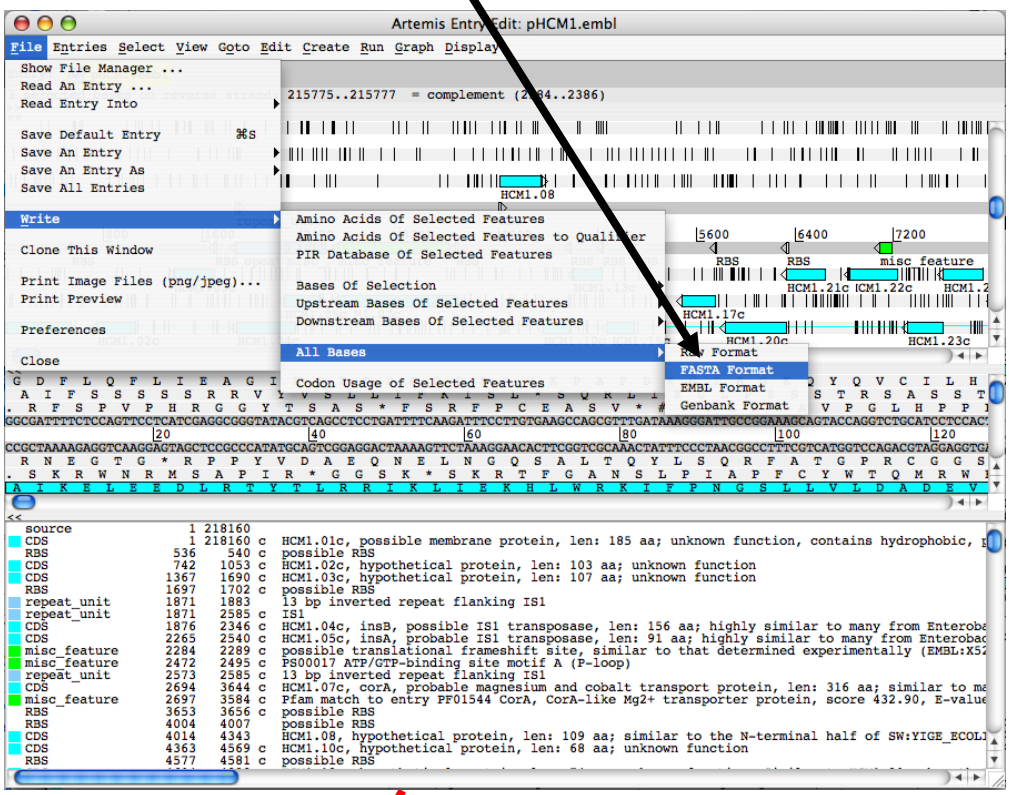
Save the EMBL sequence in a suitable directory. For example: BLAST_Appendix

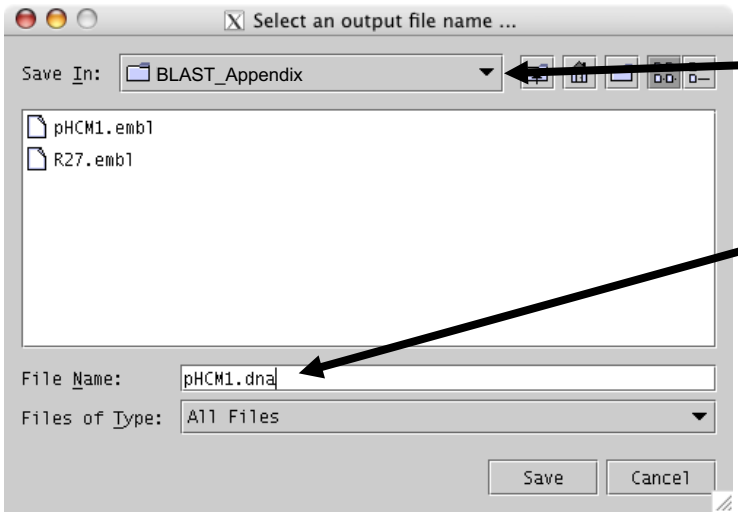
Save the file as pHCM1.embl

Repeat for the *Salmonella typhi* R27 plasmid (AF250878). Be careful when choosing the plasmid to download as there is also a *Salmonella typhi* plasmid R27 entry (AF105019), the one that you want is the larger of the two, 180,461 kb as opposed to 38,245 kb – make sure the accession number is correct. Save as R27.embl.

In order to run BLASTN you require two DNA sequences in FASTA format. The pHCM1 and R27 sequences previously downloaded from the EBI are EMBL format files, i.e. they contain protein coding information and the DNA sequence. In order to generate the DNA files in FASTA format, Artemis can be used as follows.

Load up the plasmid EMBL files in **Artemis** (each plasmid requires a separate Artemis window), select **Write, All Bases**, **FASTA format**.





Save the DNA sequence in the BLAST_Appendix directory

Save as pHCM1.dna

Also do this for R27.embl

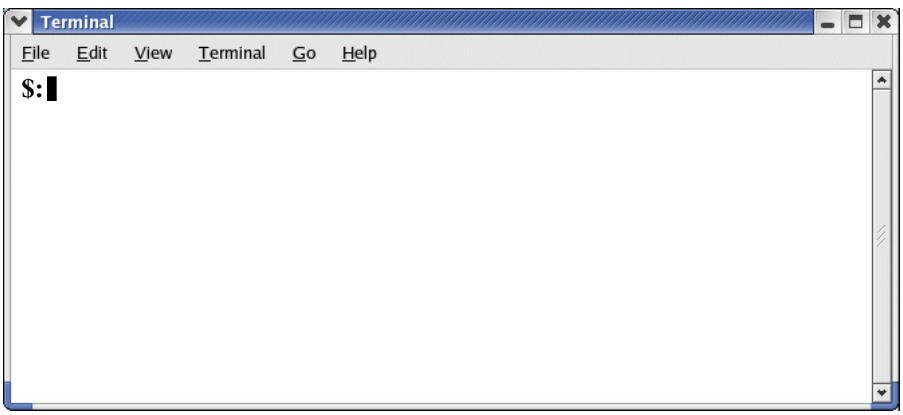
Running Blast

There are several programs in the BLAST package that can be used for generating sequence comparison files. For a detailed description of the uses and options see the appropriate README file in the BLAST software directory (see Appendix X).

In order to generate comparison files that can be read into ACT you can use the **blastall** program running either BLASTN (DNA-DNA comparison) or TBLASTX (translated DNA-translated DNA comparison) protocols.

As an example you will run a BLASTN comparison on two relatively small sequences; the pHCM1 and R27 plasmids from *S. typhi*. In principle any DNA sequences in FASTA format can be used, although size becomes and issue when dealing with sequences such whole genomes of several Mb (see **Exercise 2** in this module). When obtaining nucleotide sequences from databases such as EMBL using a server such as SRS (http://srs.ebi.ac.uk), it is possible to specify that the sequences are in FASTA format.

To run the BLAST software you will need an Xterminal window like the one below. If you do not already have one opened, you can open a new window by clicking on the Xterminal icon on the menu bar at the bottom of your screen.



Make sure you are in the appropriate directory (in this example it is BLAST_Appendix.) You should now see both the new FASTA files for the pHCM1 and R27 sequences in the BLAST_Appendix directory as well as their respective EMBL format files.
(Hint: You can use the **pwd** command to check the present working directory, the **cd** command to change directories, and the **ls** command will list the contents of the present working directory).

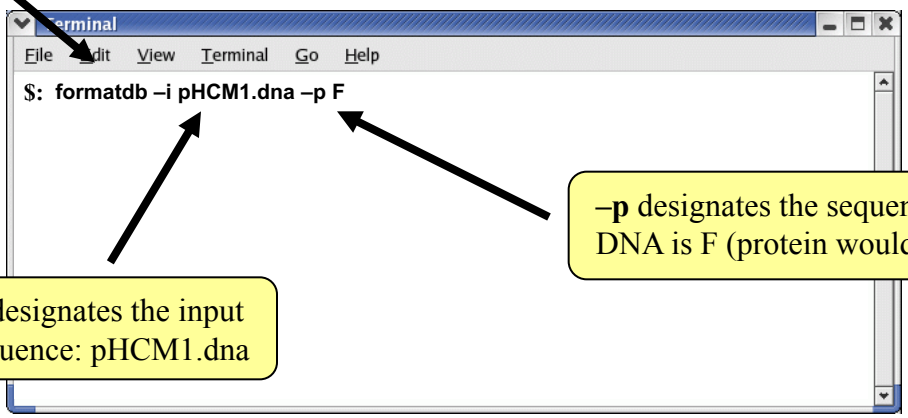
When comparing sequences in BLAST, one sequence is designated as a **database** sequence, and the other the **query** sequence. Before you run BLAST you have to format one of the sequences so that BLAST recognises it as a database sequence. **formatdb** is a program that does this and comes as part of the NCBI BLAST distribution.

You will treat pHCM1.dna as the **database** sequence and R27.dna as the **query** sequence

At the Command Prompt type:
formatdb –i pHCM1.dna –p F

Press **Return**

formatdb is the database format program



–p designates the sequence type: DNA is F (protein would be T)

–i designates the input sequence: pHCM1.dna

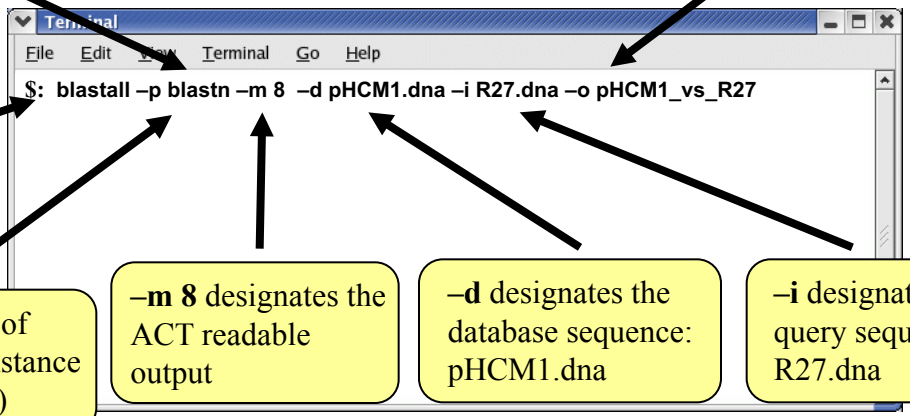
Now you can run the BLAST on the two plasmid sequences. The program that you are going to use is **blastall**. In addition to the standard command line inputs we have to add an additional flag (**-m 8**) to the command line so that the BLAST output can be read by ACT. This specifies that the output of BLAST is in one line per entry format (see appendix II).

At the Command Prompt type:
blastall –p blastn –m 8 –d pHCM1.dna –i R27.dna –o pHCM1_vs_R27

Press **Return**

tblastx could be substituted here if a translated DNA-translated DNA comparison was required

–o designates the output file: pHCM1_vs_R27



blastall is the BLAST program

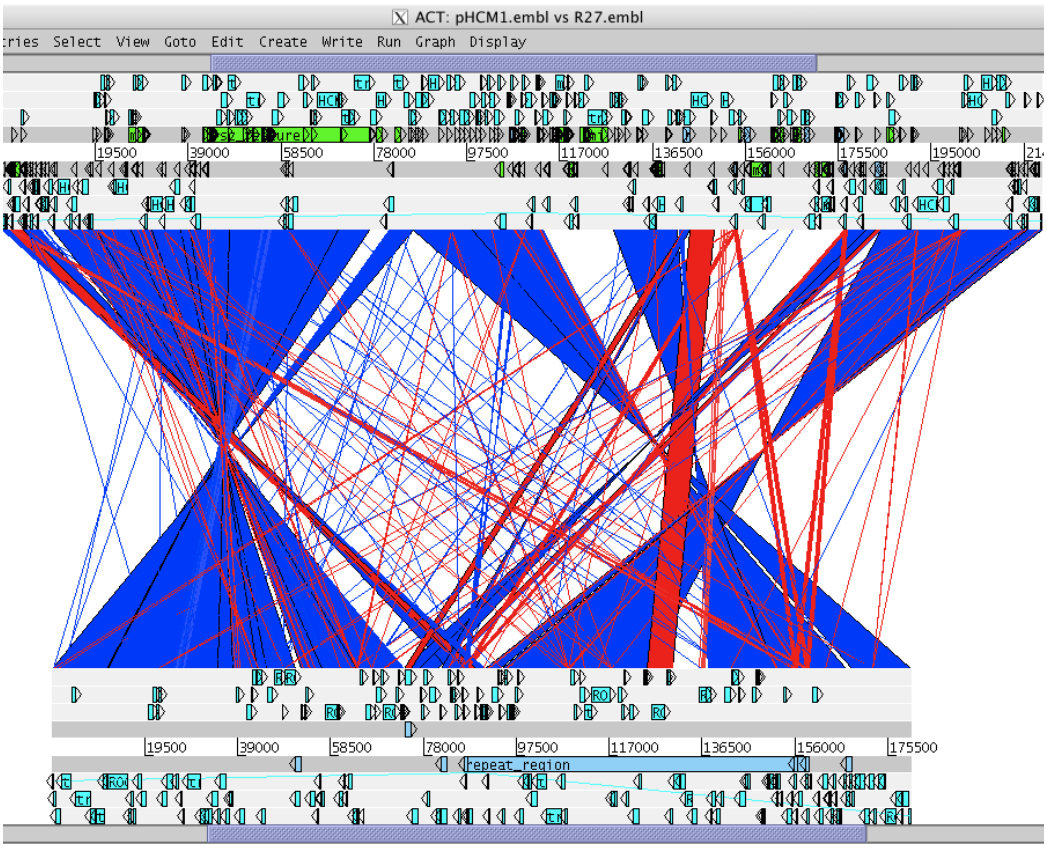
–p designates the flavour of BLAST: **blastn** (in this instance a DNA-DNA comparison)

–m 8 designates the ACT readable output

–d designates the database sequence: pHCM1.dna

–i designates the query sequence: R27.dna

The pHCM1_vs_R27 comparison file can now be read into ACT along with the pHCM1.embl and R27.embl (or pHCM1.dna and R27.dna) sequence files.



The result of the BLASTN comparison shows that there are regions of DNA shared between the plasmids; pHCM1 shares 169 kb of DNA at greater than 99% sequence identity with R27. Much of the additional DNA in the pHCM1 plasmid appears to have been inserted relative to R27 and encodes functions associated with drug resistance. What antibiotic resistance genes can you find in the pHCM1 plasmid that are not found in R27?

The two plasmids were isolated more than 20 years apart. The comparison suggests that there have been several independent acquisition events that are responsible for the multiple drug resistance seen in the more modern *S. typhi* plasmid.

Exercise 2

In the previous exercise you used BLASTN to generate a comparison file for two relatively small sequences (>500,000 kb). In the next exercise we are going to use another program from NCBI BLAST distribution, **megablast**, that can be used for nucleotide sequence alignment searches, i.e. DNA-DNA comparisons. If you are comparing large sequences such as whole genomes of several Mb, the **blastall** program is not suitable. The BLAST algorithms will struggle with large DNA sequences and therefore the processing time to generate a comparison file will increase dramatically.

megablast uses a different algorithm to BLAST which is not as stringent which therefore makes the program faster. This means that it is possible to generate comparison files for genome sequences in a matter of seconds rather than minutes and hours.

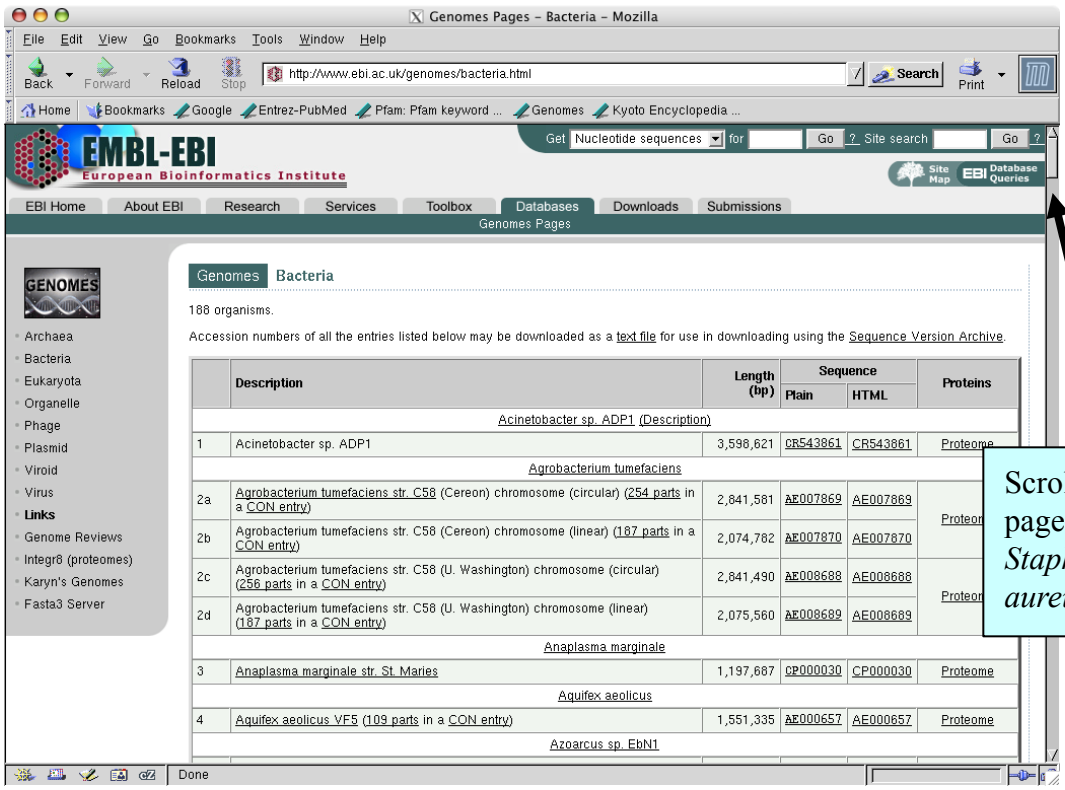
There are some drawbacks to using this program. Firstly, only DNA-DNA alignments (BLASTN) can be performed using **megablast**, rather than translated DNA-DNA alignments (TBLASTX) as can be using **blastall**. Secondly as the algorithm used is not as stringent, **megablast** is suited to comparing sequences with high levels of similarity such as genomes from the same or very closely related species.

In this exercise you are going to download two *Staphylococcus aureus* genome sequences from the EBI genomes web page and use Artemis to write out the FASTA format DNA sequences for both as before in **Exercise 1**. These two FASTA format sequences will then be compared using **megablast** to identify regions of DNA-DNA similarity and write out an ACT readable comparison file.

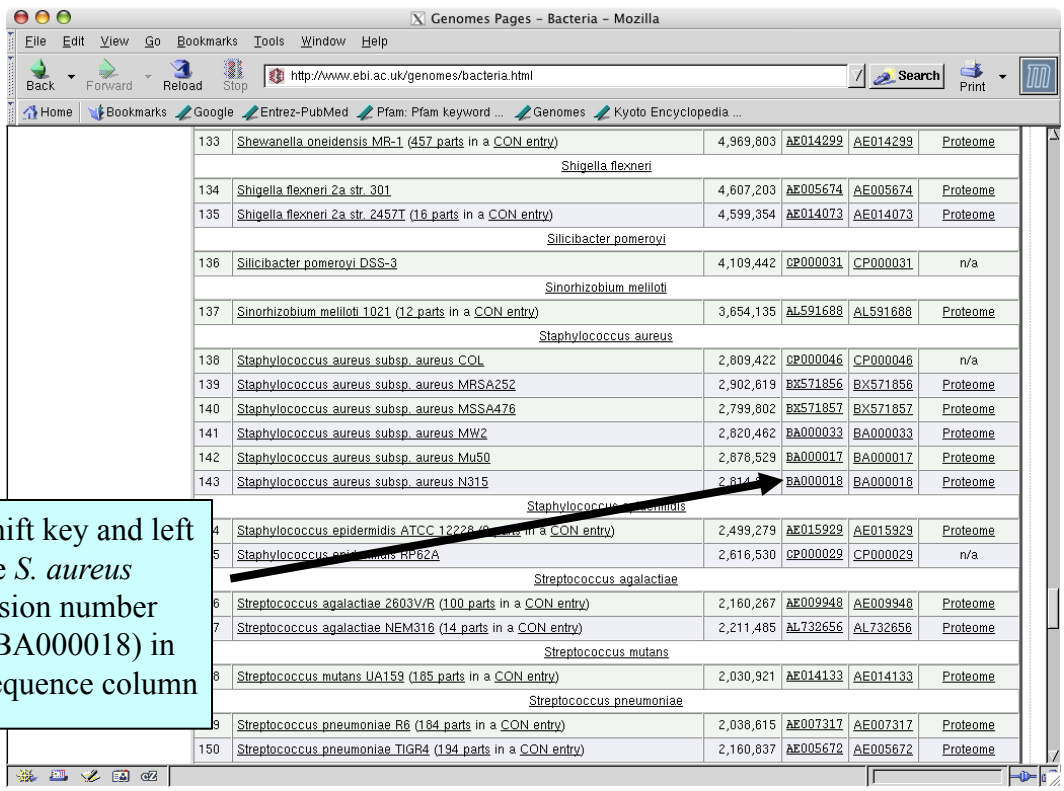
The genomes that have been chosen for this comparison are from a hospital-acquired methicillin resistant *S. aureus* (MRSA) strain N315 (BA000018), and a community-acquired MRSA strain MW2 (BA000033).

Downloading the *S. aureus* genomic sequences

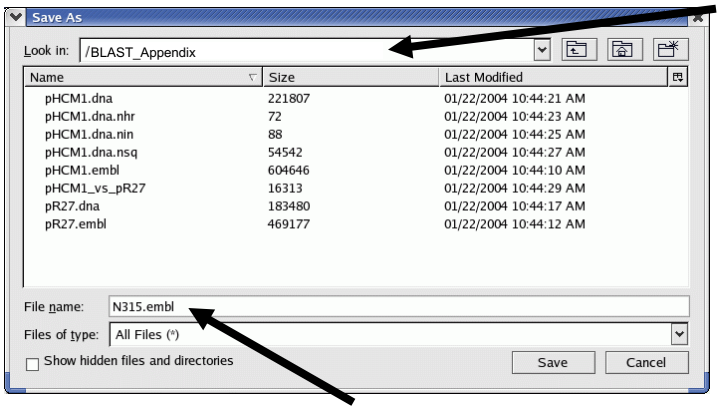
Go to the EBI genomes web page (http://www.ebi.ac.uk/genomes) as before in **Exercise 2**, and click on the **Bacteria** hyperlink



Scroll down the page to the *Staphylococcus aureus* genomes



Press the Shift key and left Click on the *S. aureus* N315 accession number hyperlink (BA000018) in the Plain Sequence column



Save the EMBL sequence in a suitable directory. For example: BLAST_Appendix

Save the file as N315.embl

Repeat for the *S. aureus* MW2 genome (BA000033). Be careful when choosing the genome to download as there is another *S. aureus* genome entry for strain Mu50 (BA000017). Save as MW2.embl.

Generate DNA files in FASTA format using Artemis for both the genome sequences as previously done in exercise 1.

(Hint: In **Artemis** (each genome requires a separate Artemis window), select **Write**, **Write All Bases**, **FASTA format**).

Save the DNA sequences as N315.dna and MW2.dna for the respective genomes.

Running Blast

In the previous exercise you used the **blastall** program to run BLASTN on two plasmid sequences. As the genome sequences are larger (~2.8 Mb) you are going to run **megablast**, another program from the NCBI BLAST distribution that can generate comparison files in a format that ACT can read (see Appendix II). For a detailed description of the uses and options in **megablast** see the megablast README file in the BLAST software directory (Appendix X).

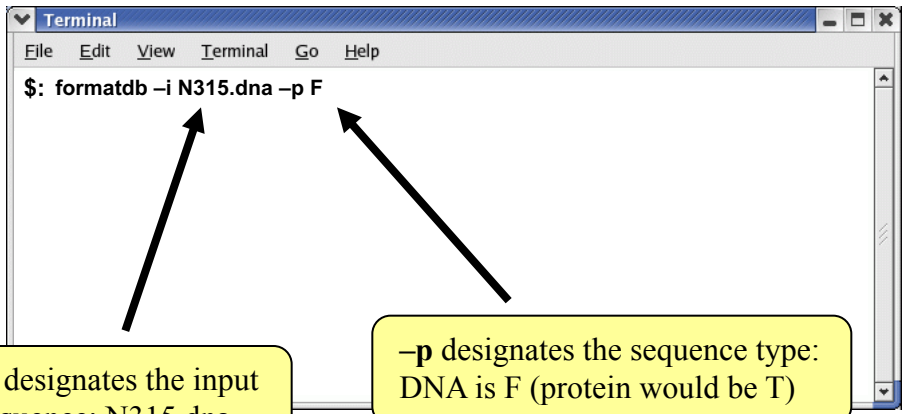
As before you will run the program from the command line in an Xterminal window.

Like BLAST, **megablast** requires that one sequence is designated as a **database** sequence and the other the **query** sequence. Therefore one of the sequences has to be formatted so that Blast recognises it as a database sequence. This can be done as before using **formatdb.**

We will treat N315.dna as the **database** sequence and MW2.dna as the **query** sequence

At the Command Prompt type:
formatdb –i N315.dna –p F

Press **Return**



–i designates the input sequence: N315.dna

–p designates the sequence type: DNA is F (protein would be T)

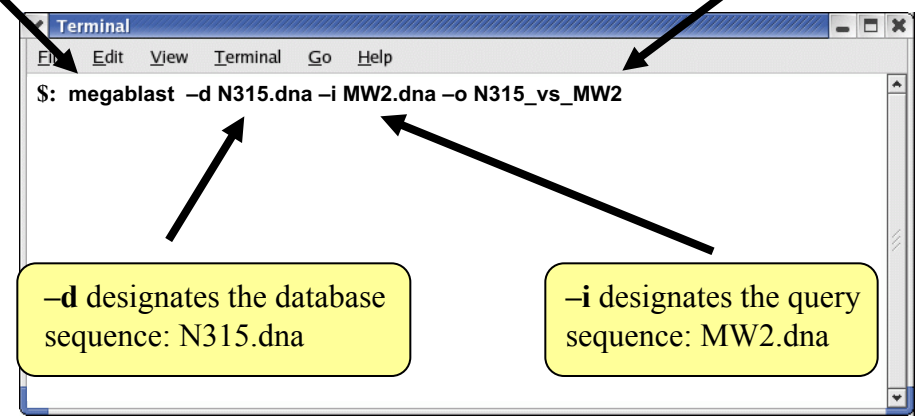
Now we can run the **megablast** on the two MRSA genome sequences. The default output format is one line per entry that ACT can read, therefore there is no need to add an additional flag (i.e. -m 8) to the command line (see appendix II).

At the Command Prompt type:
megablast –d N315.dna –i MW2.dna –o N315_vs_MW2

Press **Return**

megablast is the program

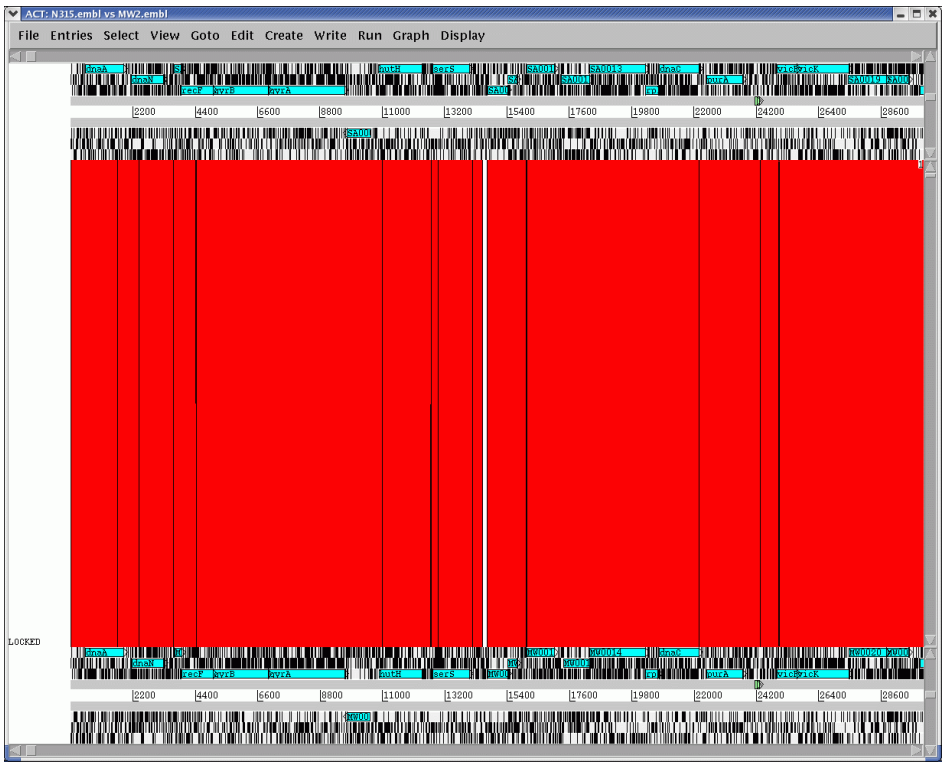
–o designates the output file: N315_vs_MW2



–d designates the database sequence: N315.dna

–i designates the query sequence: MW2.dna

The N315_vs_MW2 comparison file can now be read into ACT along with the N315.embl and MW2.embl (or N315.dna and MW2.dna) sequence files.



A comparison of the N315 and MW2 genomes in ACT using the **megablast** comparison reveals a high level of synteny (conserved gene order). This is perhaps not unsurprising as both genomes belong to strains of the same species. Using results of comparisons like these it is possible to identify genomic differences that may contribute to the biology of the bacteria and also investigate mechanisms of evolution.

Both N315 and MW2 are MRSA, however N315 is associated with disease in hospitals, and MW2 causes disease in the community and is more invasive. Scroll rightward in both genomes to find the first large region of difference. Examine the annotation for the genes in these regions. What are the encoded functions associated with these regions? What significance does this have for the evolution of methicillin resistance in these two *S. aureus* strains from clinically distinct origins?

Appendix VI – Generating Artemis comparison files using WebACT

Introduction

If you do not have access to BLAST software running on a local computer, there is a web resource WebACT (Appendix VII for the URL) that can be used for generating ACT comparison files. WebACT allows you to cut and paste, or upload, your own sequences, and generate ACT readable BLASTN or TBLASTX comparison files. WebACT also has a large selection of recomputed comparison files for bacterial genomes, which can be downloaded along with the EMBL sequence entries and viewed in ACT.

For the purposes of this exercise we are going to focus on the Gram-negative bacterial pathogens *Burkholderia pseudomallei* and *Burkholderia mallei.* Both of these organisms are category B bio-threat agents and cause the diseases Melioidosis and Glanders respectively. The two species are closely related (DNA-DNA identity is >99%, multi locus sequence typing (MLST) predicts that *B. mallei* is a clone of *B. pseudomallei*), however they differ markedly in the environmental niches that they occupy.

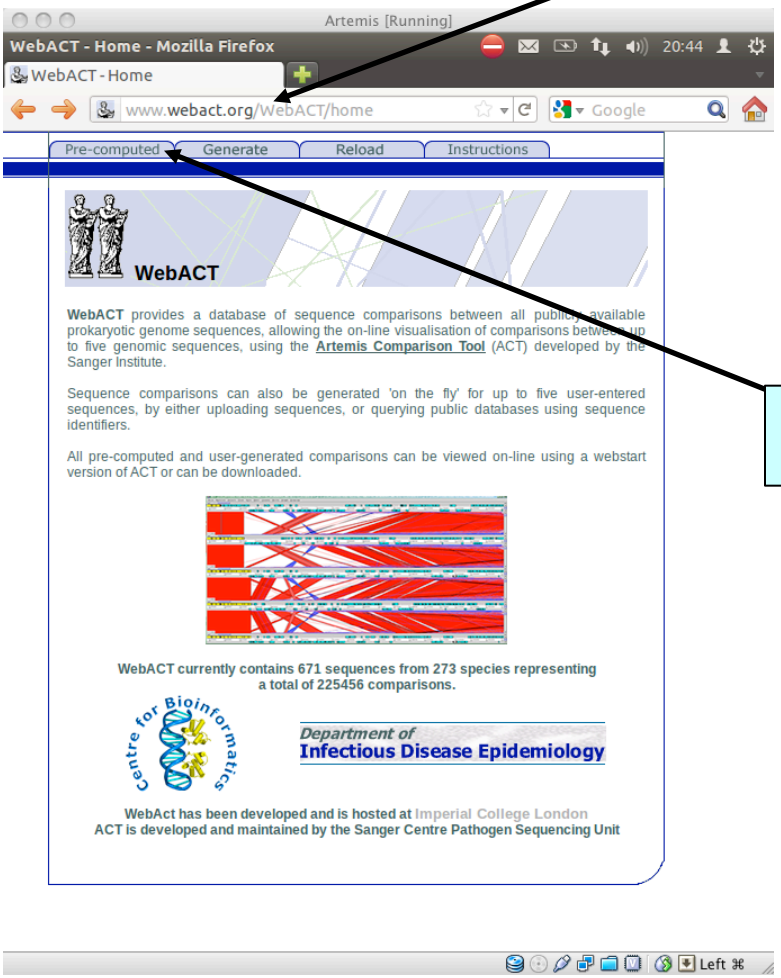
B. pseudomallei is found in S.E. Asia and northern Australia, and is prevalent in the soil in Melioidosis endemic areas. Inhalation, or direct contact with cuts or breaks in the skin, by soil-borne *B. pseudomallei* is the cause of Melioidosis in humans and higher mammals. In contrast, *B. mallei* is a zoonotic pathogen that is host restricted to horses and cannot be isolated from the environment. Comparative genomic analysis has provided insights into evolution of these two pathogens and the genetic basis for ecological and pathological differences of these two pathogens.

The genomes of these two organisms both consist of two circular chromosomes. Comparisons of the genomes reveals that the genome of *B. pseudomallei* is ~1.31 Mb larger than that of *B. mallei*; 16% of chromosome 1, and 32% of chromosome 2, are unique in *B. pseudomallei* with respect to *B. mallei*.

Aim

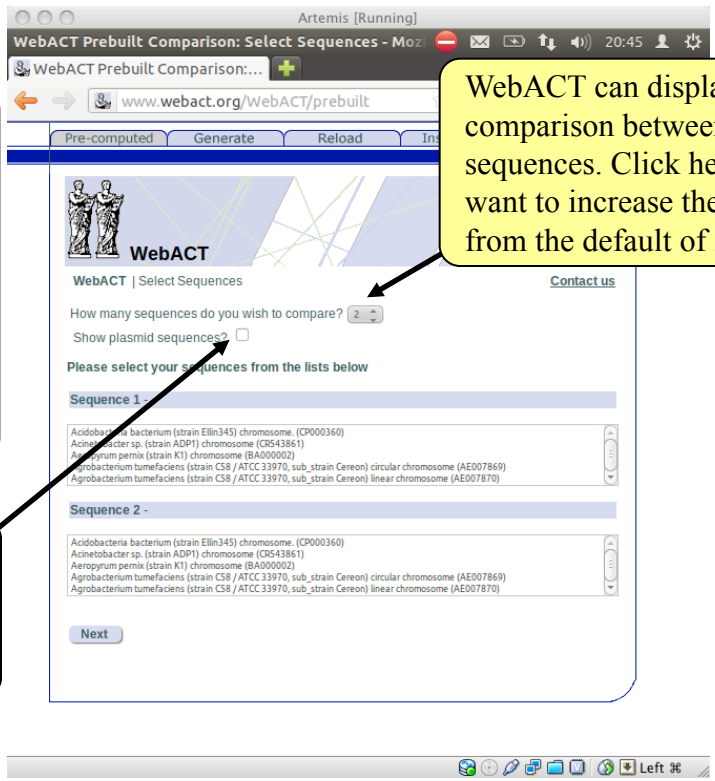
You are going to use a web resource, WebACT, to generate a comparison file of the smaller chromosomes of *B. pseudomallei* and *B. mallei*. From the WebACT site you will download a pre-computed ACT comparison comparison file, along with the appropriate EMBL sequence and annotation files, which you will then open in ACT. Using this comparison you can then investigate some of the the genotypic differences that differentiate these closely related pathogens, and look for the basis of structural differences in these chromosomes. We have not provided files for this exercise - you are on your own.

Open up a web browser and go to the URL: **www.webact.org**



Click on the Pre-computed tab.

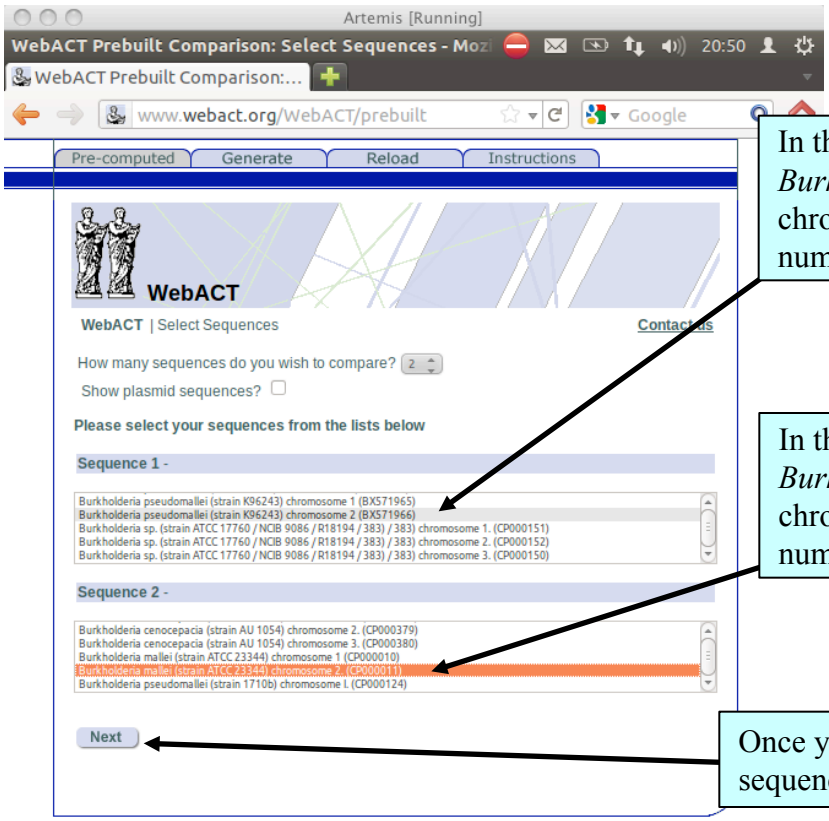
The 'Pre-computed' page contains genomic sequences that have been compared using BLASTN to each other. By selecting the desired sequences from the sequence lists, the appropriate sequence and comparison files can be downloaded



WebACT can display pairwise comparison between up to 5 sequences. Click here if you want to increase the number from the default of 2.

In addition to the chromosome sequences, plasmids can also be displayed by clicking in this box

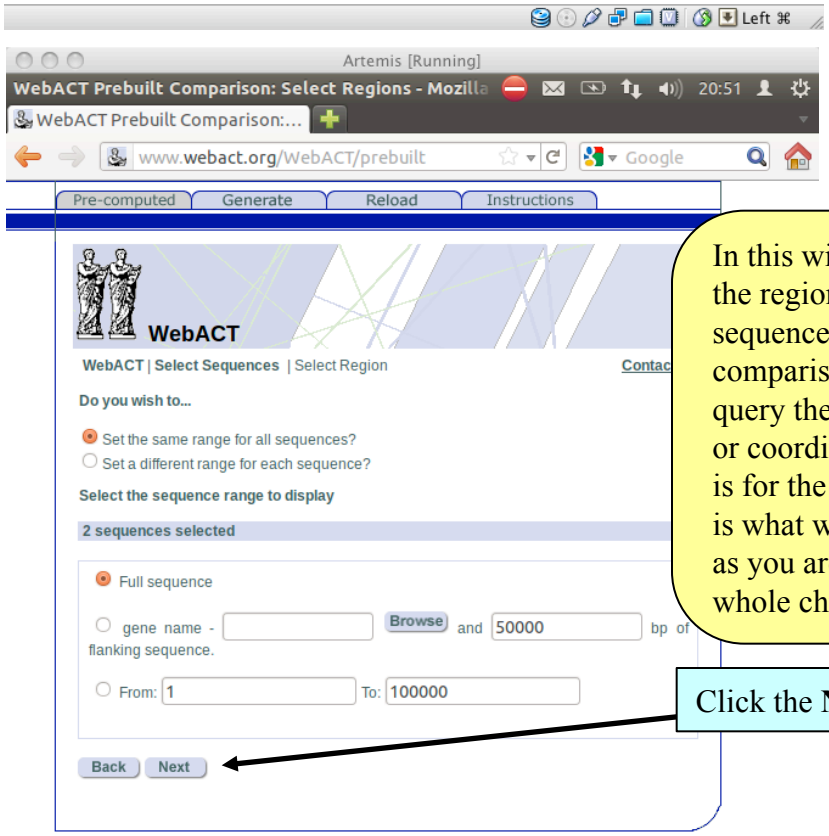
You are going to compare the smaller chromosomes of *B. pseudomallei* and *B. mallei*.



In the **Sequence 1** list select *Burkholderia pseudomallei* chromosome 2 (accession number BX571966)

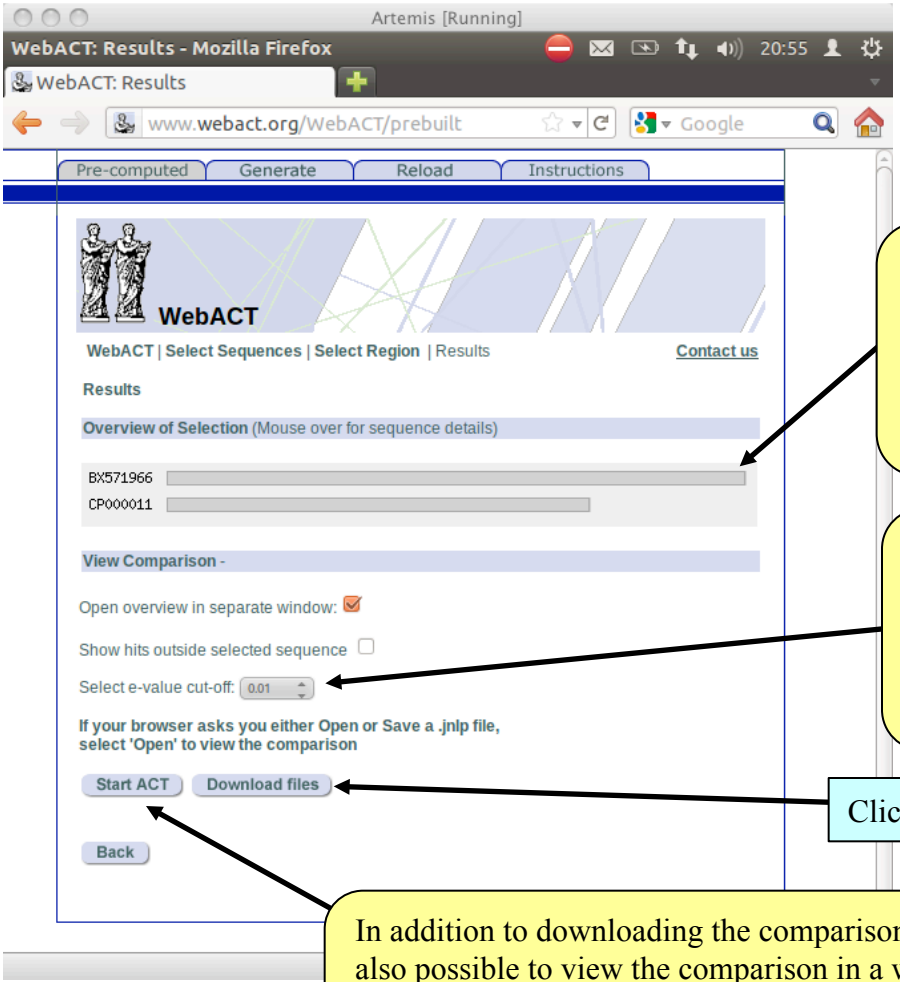
In the **Sequence 2** list select *Burkholderia mallei* chromosome 2 (accession number CP000011)

Once you have selected the sequences click the **Next** button



In this window you can specify the regions in the selected sequences to generate the comparison over. It is possible to query the sequences on gene name or coordinates. The default setting is for the whole sequence, and this is what we want for this exercise as you are going to compare the whole chromosomes.

Click the **Next** button



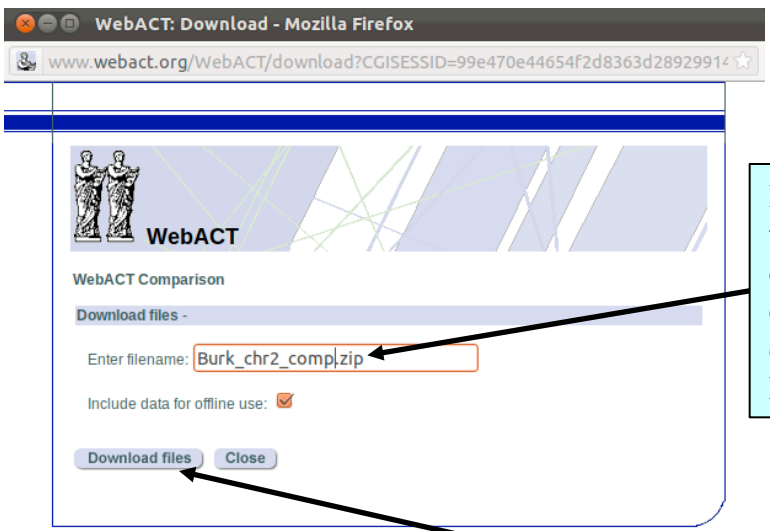
In the **Overview of Selection** you can see a schematic representation of the relative size of the two sequence that have been chosen to be compared.

The Expect (E) value cut-off can be changed in this box. The default value is 0.01, but the range is from 10.0 to 0.0001.

Click the **Download files** button

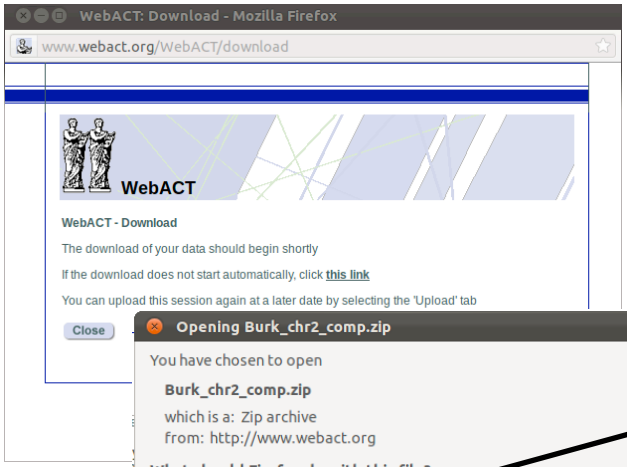
In addition to downloading the comparison files and sequence file it is also possible to view the comparison in a webstart version of ACT. This will run locally on your machine and does not require ACT to be previously loaded, as a webstart version of ACT will be included in the download. You are not going to use this option in this exercise.

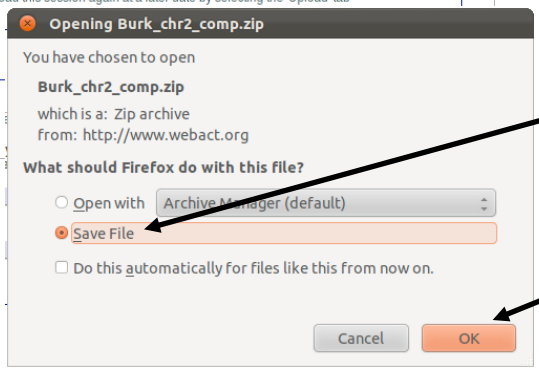
The comparison file and sequences files will contained in a folder. For the ease of downloading the folder is zipped.



In the filename box you can type the file name of the zip file containing the sequence and comparison files. For this exercise call the file: **Burk_chr2_comp.zip**

Click the **Download files** button





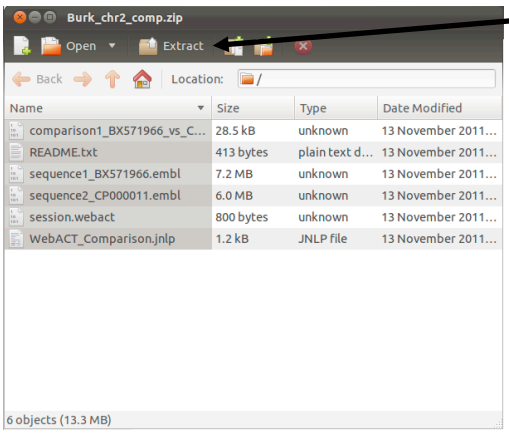
You may get a window appearing asking you what Firefox should do with the Burk_chr2comp.zip file? Save the file to disk.

Click the **OK** button

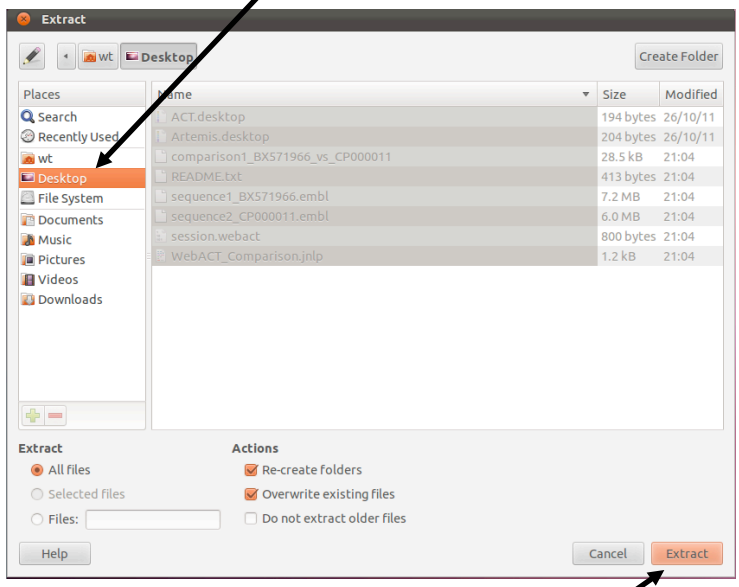
Burk_chr2_comp.zip should now be in the **Downloads** directory

To unzip the file, double click with the left mouse button on the file name

Click the **Extract** button



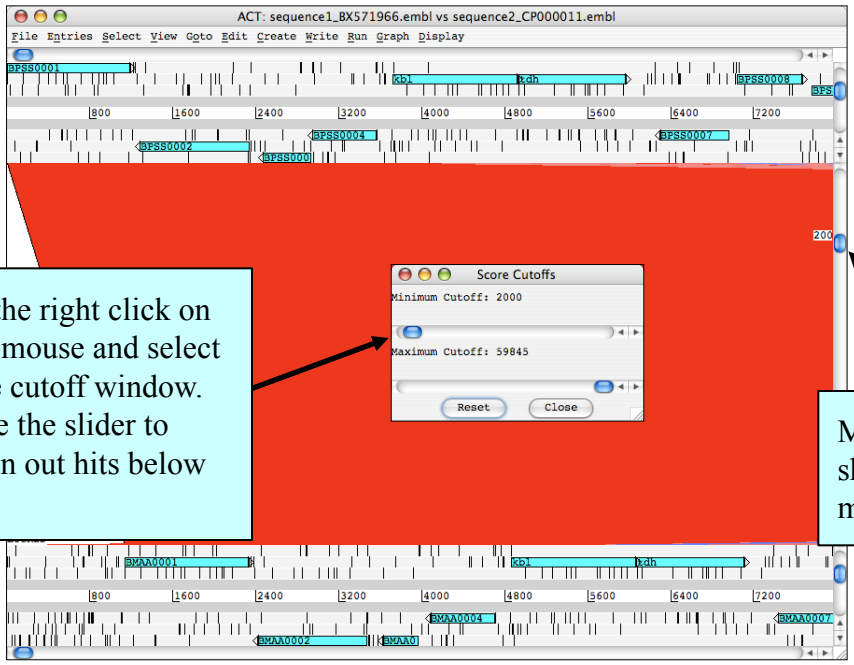
Select a location to extract the files to, such as **Desktop**



The files contained in the unzipped directory should include: comparison1_BX571966_vs_CP000011, sequence1_BX571966.embl and sequence2_CP000011.embl. These are the ACT comparison file and the *B. pseudomalllei* and *B. mallei* chromosome 2 EMBL annotation and sequence files respectively.

Click the **Extract** button

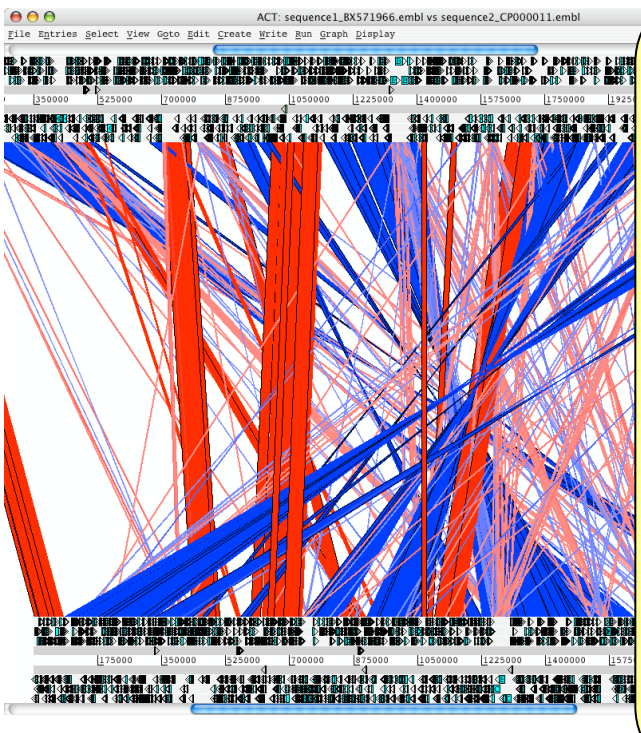
Open up ACT, and load up the comparison (comparison1_BX571966_vs_CP000011) along with the two EMBL sequence and annotation files (sequence1_BX571966.embl and sequence2_CP000011.embl). If you get a warnings window asking if you want to read warning, click **No**.



Use the right click on your mouse and select score cutoff window. Move the slider to screen out hits below 2000

Move the slider to 200 to show only BLASTN matches greater than 200

Now remove the stop codons for both entries, and then zoom out you will see the overall conservation of the structure of the small chromosomes is poor.



If you were to look at the comparison for the large chromosomes you would see a similar picture. The lack of conservation is the result of intra-chromosomal rearrangements. What do you think caused this? Zoom into the regions on the edge of the rearranged matches and look at the annotation in the *B. mallei* chromosome.

What is the function of the CDSs consistently found in these regions. Are there matches in the *B. pseudomallei* chromosome?

Try selecting CDSs in *B. pseudomallei* that match these regions and look how many matches there are in *B. mallei*. Are these regions repeated throughout the chromosome?

If you have time, you may want to generate, and view in ACT, comparisons for your own sequences. If you do not have any loaded on your workshop computer, why not try and download some. Sequence in various formats can be cut and pasted, or up loaded onto the WebACT site. In addition, if you know the accession number of the sequence that you want to compare, you can use that. As the web site will have to run BLAST to generate your comparison file, you may want to limit the size of the sequence that you submit for this exercise to <100 kb. The the web site can handle larger sequences, but it will just take longer.

Clicking on the 'Generate' tab will take you to this page

Number of sequences to compare

Cut and paste sequence

Upload file

Type accession number

Click here for BLAST options, such as changing from the default BlastN to TBlastX, and altering the BLAST cutoffs

Once you added the relevant sequence information, submit your query. The comparison file or files are down loaded as shown in the example, and can them be loaded in to ACT.

Appendix VII: Useful Web addresses

Major Public Sequence Repositories

DNA Data Bank of Japan (DDBJ)	http://www.ddbj.nig.ac.jp
EMBL Nucleotide Sequence Database	http://www.ebi.ac.uk/embl
Genomes at the EBI	http://www.ebi.ac.uk/genomes
GenBank	http://www.ncbi.nih.gov/Genbank

Microbial Genome Databases Resources

Sanger Microbial Genomes	http://www.sanger.ac.uk/Projects/Pathogens
GeneDB	http://www.genedb.org
Institute Pasteur GenoList databases *Including: SubtiList, Colbri, TubercuList, Leproma, PyloriGene, MypuList, ListiList, CandidaDB.*	http://genolist.pasteur.fr
Pseudomonas Genome Database	http://www.pseudomonas.com
Clusters of Orthologous Groups of proteins (COGs)	http://www.ncbi.nlm.nih.gov/COG
ScoDB (*S. coelicolor* database)	http://streptomyces.org.uk
GenProtEC	http://genprotec.mbl.edu

Protein Motif Databases

Prosite	http://www.expasy.ch/prosite/
Pfam	http://pfam.sanger.ac.uk
BLOCKS	http://blocks.fhcrc.org
InterPro	http://www.ebi.ac.uk/interpro/
PRINTS	http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/
SMART	http://smart.embl-heidelberg.de

Protein feature prediction tools

TMHMM Transmembrane helices prediction	http://www.cbs.dtu.dk/services/TMHMM-2.0/
SignalP Prediction Server	http://www.cbs.dtu.dk/services/SignalP/
PSORT protein prediction	http://psort.ims.u-tokyo.ac.jp/form.html

Metabolic Pathways and Cellular Regulation

EcoCyc	http://ecocyc.org/
ENZYME	http://www.expasy.ch/enzyme/
Kyoto Encyclopedia of Genes and Genomes (KEGG)	http://www.genome.ad.jp/kegg
MetaCyc	http://metacyc.org/

Miscellaneous sites

NCBI BLAST website	http://www.ncbi.nlm.nih.gov/BLAST/
EBI FASTA website	http://www.ebi.ac.uk/fasta33/index.html
The tmRNA website	http://www.indiana.edu/~tmrna/
tRNAscan-SE Search Server	http://selab.janelia.org/tRNAscan-SE/
Rfam	http://rfam.sanger.ac.uk/
Codon usage database	http://www.kazusa.or.jp/codon/
GO Gene Ontology Consortium	http://www.geneontology.org/
Artemis homepage	http://www.sanger.ac.uk/Software/Artemis/
ACT homepage	http://www.sanger.ac.uk/Software/ACT/
WebACT	http://www.webact.org/WebACT/home
Double ACT	http://www.hpa-bioinfotools.org.uk/pise/double_act.html
Glimmer	http://cbcb.umd.edu/software/glimmer/
EasyGene	http://www.cbs.dtu.dk/services/EasyGene/
String	http://string.embl.de
EMBOSS	http://emboss.sourceforge.net/

Appendix VIII: Prokaryotic Protein Classification Scheme used within the PSU

This scheme was adapted for in-house use from the Monica Riley's protein classification (http://genprotec.mbl.edu/files/Multifun.html).
More classes can be added depending on the microorganism that is being annotated (e.g secondary metabolites, sigma factors (ECF or non-ECF), etc).

0.0.0 Unknown function, no known homologs
0.0.1 Conserved in Escherichia coli
0.0.2 Conserved in organism other than Escherichia coli
1.0.0 Cell processes
- 1.1.1 Chemotaxis and mobility
- 1.2.1 Chromosome replication
- 1.3.1 Chaperones

1.4.0 Protection responses
- 1.4.1 Cell killing
- 1.4.2 Detoxification
- 1.4.3 Drug/analog sensitivity
- 1.4.4 Radiation sensitivity

1.5.0 Transport/binding proteins
- 1.5.1 Amino acids and amines
- 1.5.2 Cations
- 1.5.3 Carbohydrates, organic acids and alcohols
- 1.5.4 Anions
- 1.5.5 Other

1.6.0 Adaptation
- 1.6.1 Adaptations, atypical conditions
- 1.6.2 Osmotic adaptation
- 1.6.3 Fe storage

1.7.1 Cell division
2.0.0 Macromolecule metabolism
2.1.0 Macromolecule degradation
- 2.1.1 Degradation of DNA
- 2.1.2 Degradation of RNA
- 2.1.3 Degradation of polysaccharides
- 2.1.4 Degradation of proteins, peptides, glycoproteins

2.2.0 Macromolecule synthesis, modification
- 2.2.01 Amino acyl tRNA synthesis; tRNA modification
- 2.2.02 Basic proteins - synthesis, modification
- 2.2.03 DNA - replication, repair, restriction./modification
- 2.2.04 Glycoprotein
- 2.2.05 Lipopolysaccharide
- 2.2.06 Lipoprotein
- 2.2.07 Phospholipids
- 2.2.08 Polysaccharides - (cytoplasmic)
- 2.2.09 Protein modification
- 2.2.10 Proteins - translation and modification
- 2.2.11 RNA synthesis, modif., DNA transcrip.
- 2.2.12 tRNA

3.0.0 Metabolism of small molecules
3.1.0 Amino acid biosynthesis
- 3.1.01 Alanine
- 3.1.02 Arginine
- 3.1.03 Asparagine
- 3.1.04 Aspartate
- 3.1.05 Chorismate
- 3.1.06 Cysteine
- 3.1.07 Glutamate
- 3.1.08 Glutamine
- 3.1.09 Glycine
- 3.1.10 Histidine
- 3.1.11 Isoleucine
- 3.1.12 Leucine
- 3.1.13 Lysine
- 3.1.14 Methionine
- 3.1.15 Phenylalanine
- 3.1.16 Proline
- 3.1.17 Serine
- 3.1.18 Threonine
- 3.1.19 Tryptophan
- 3.1.20 Tyrosine
- 3.1.21 Valine

Appendix VIII (cont):

3.2.0 Biosynthesis of cofactors, carriers
- 3.2.01 Acyl carrier protein (ACP)
- 3.2.02 Biotin
- 3.2.03 Cobalamin
- 3.2.04 Enterochelin
- 3.2.05 Folic acid
- 3.2.06 Heme, porphyrin
- 3.2.07 Lipoate
- 3.2.08 Menaquinone, ubiquinone
- 3.2.09 Molybdopterin
- 3.2.10 Pantothenate
- 3.2.11 Pyridine nucleotide
- 3.2.12 Pyridoxine
- 3.2.13 Riboflavin
- 3.2.14 Thiamin
- 3.2.15 Thioredoxin, glutaredoxin, glutathione
- 3.2.16 biotin carboxyl carrier protein (BCCP)

3.3.0 Central intermediary metabolism
- 3.3.01 2'-Deoxyribonucleotide metabolism
- 3.3.02 Amino sugars
- 3.3.03 Entner-Douderoff
- 3.3.04 Gluconeogenesis
- 3.3.05 Glyoxylate bypass
- 3.3.06 Incorporation metal ions
- 3.3.07 Misc. glucose metabolism
- 3.3.08 Misc. glycerol metabolism
- 3.3.09 Non-oxidative branch, pentose pathway
- 3.3.10 Nucleotide hydrolysis
- 3.3.11 Nucleotide interconversions
- 3.3.12 Oligosaccharides
- 3.3.13 Phosphorus compounds
- 3.3.14 Polyamine biosynthesis
- 3.3.15 Pool, multipurpose conversions of intermed. metab.
- 3.3.16 S-adenosyl methionine
- 3.3.17 Salvage of nucleosides and nucleotides
- 3.3.18 Sugar-nucleotide biosynthesis, conversions
- 3.3.19 Sulfur metabolism
- 3.3.20 Amino acids
- 3.3.21 other

3.4.0 Degradation of small molecules
- 3.4.1 Amines
- 3.4.2 Amino acids
- 3.4.3 Carbon compounds
- 3.4.4 Fatty acids
- 3.4.5 Other
- 3.4.0 ATP-proton motive force

3.5.0 Energy metabolism, carbon
- 3.5.1 Aerobic respiration
- 3.5.2 Anaerobic respiration
- 3.5.3 Electron transport
- 3.5.4 Fermentation
- 3.5.5 Glycolysis
- 3.5.6 Oxidative branch, pentose pathway
- 3.5.7 Pyruvate dehydrogenase
- 3.5.8 TCA cycle

3.6.0 Fatty acid biosynthesis
- 3.6.1 Fatty acid and phosphatidic acid biosynthesis

3.7.0 Nucleotide biosynthesis
- 3.7.1 Purine ribonucleotide biosynthesis
- 3.7.2 Pyrimidine ribonucleotide biosynthesis

4.0.0 Cell envelop
- 4.1.0 Periplasmic/exported/lipoproteins
- 4.1.1 Inner membrane
- 4.1.2 Murein sacculus, peptidoglycan
- 4.1.3 Outer membrane constituents
- 4.1.4 Surface polysaccharides & antigens
- 4.1.5 Surface structures

4.2.0 Ribosome constituents
- 4.2.1 Ribosomal and stable RNAs
- 4.2.2 Ribosomal proteins - synthesis, modification
- 4.2.3 Ribosomes - maturation and modification

5.0.0 Extrachromosomal

5.1.0 Laterally acquired elements
- 5.1.1 Colicin-related functions
- 5.1.2 Phage-related functions and prophages
- 5.1.3 Plasmid-related functions
- 5.1.4 Transposon-related functions
- 5.1.5 Pathogenicity island-related function

6.0.0 Global functions
- 6.1.1 Global regulatory functions

7.0.0 Not classified (included putative assignments)

Appendix IX: List of colour codes

0 (white) - Pathogenicity/Adaptation/Chaperones
1 (dark grey) - energy metabolism (glycolysis, electron transport etc.)
2 (red) - Information transfer (transcription/translation + DNA/RNA modification)
3 (dark green) - Surface (IM, OM, secreted, surface structures
4 (dark blue) - Stable RNA
5 (Sky blue) - Degradation of large molecules
6 (dark pink) - Degradation of small molecules
7 (yellow) - Central/intermediary/miscellaneous metabolism
8 (light green) - Unknown
9 (light blue) - Regulators
10 (orange) - Conserved hypo
11 (brown) - Pseudogenes and partial genes (remnants)
12 (light pink) - Phage/IS elements
13 (light grey) - Some misc. information e.g. Prosite, but no function

Appendix X: List of degenerate nucleotide value/IUB Base Codes.

R = A or G

S = G or C

B = C, G or T

Y = C or T

W = A or T

D = A, G or T

K = G or T

N = A, C, G or T

H = A, C or T

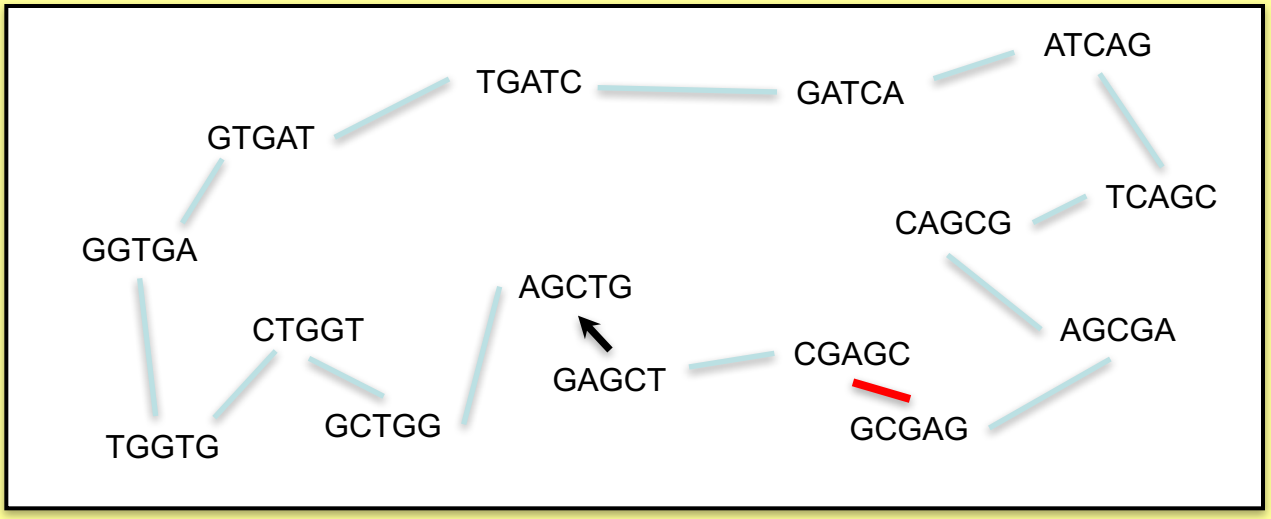
M = A or C

V = A, C or G

Appendix X - Assembly

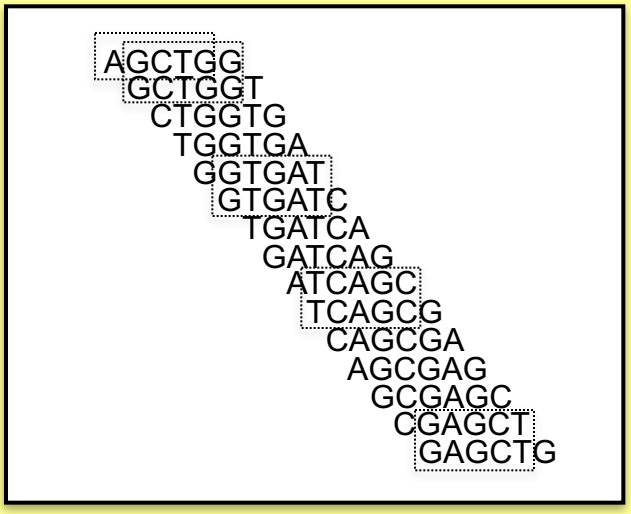
Here we present the solution for the de Bruijn graph exercise, as well as the code for the PERL scripts we mentioned in the assembly module.

The exercise is on page 2 of the assembly module. The first step of the solution would be to generate all the k-mers from the reads. For example the k-mers of length 5 for the read GCGAGC are GCGAG and CGAGC. Those two k-mers will be nodes in the de Bruijn graph, and moreover, will be connected (bold red edge). Doing this for all the reads, generates the following graph:



It is not always easy and might be confusing which node to connect. Remember, the concept of k-mers and the de Bruijn graph are needed to be able to process the large amount of short reads generated by the sequencing machines.

The graph can also be represented as a multiple alignment, as shown on the right hand site. All reads are aligned against each other. The dotted boxes are examples of k-mers.



Now just follow the path thought the graph. Starting at the arrow, the first k-mer is GAGCT, so this would be the start of our contig. The graph indicates the next k-mer AGCT**G**. So we add a G to the contig. The next k-mer is GCTG**G**. The new letter is another G. Doing this for the whole graph, we get**: GAGCTGGTGATCAGC.** As you see, the graph is circular. So depending where you start, you get a different contig! If you do a six frame translation, you might see which is a good starting point for the contig.

PERL: Find read pairs that map too far apart

For some applications it would be useful to know whether read pairs map too far apart or whether they don't map pointing to each other. This could be an indication of mis-assemblies, but also duplications or rearrangements, which are are looking for when comparing sequences of different strains.

To find read pairs (RPs) that map too far apart we just need columns 2 and 9 from the BAM file (mapping flag and insert size), and a PERL one-liner. We successively make the query more and more complex, until we find the mis-assembly. Please keep in mind that this is advanced programming! It should give you an idea how useful programming could be.

Assuming your BAM file is called IT_onDenovo.bam and you want to list RPs that map more than 2000bp apart:

```
$ samtools view IT_onDenovo.bam | perl -nle 'my
($read,$flag,$ref,$pos,$mappingQual,$cigar,$mateRef,$matePos,$insertSize,$seq,$
seqQual,$other)=split(/\t/); if($insertSize>2000){print}' | head
```

Here is also a shorter version, using an array (not as readable):

```
$ samtools view IT_onDenovo.bam | perl -nle 'my @ar=split(/\t/);
if($ar[8]>2000){print}' | head
```

There is a lot of output. Many read pairs map all over the place. We would like to bin those into chunks of 1kb, and then list of the most abundant:

```
$ samtools view IT_onDenovo.bam | perl -nle 'my
($read,$flag,$ref,$pos,$mappingQual,$cigar,$mateRef,$matePos,$insertSize,$seq,$
seqQual,$other)=split(/\t/); if($insertSize>2000){print
int($pos/1000)."\t".int($matePos/1000)}' | sort | uniq -c | sort -rn | head
```

This does look more complex! In the output the first column is the number of RPs that connect the first bin (2nd column) with the second bin (3rd column). For example `418 1360 1373` means that 481 RPs connect the region 1360000-1361000 of genome with the regions 1373000-13731000 of the genome. The list shows us that in the subtelomeric regions many RP map far apart!
The following command ignores the subtelomeric ends, by excluding 75kb at each end.

```
$ samtools view IT_onDenovo.bam | perl -nle 'my
($read,$flag,$ref,$pos,$mappingQual,$cigar,$mateRef,$matePos,$insertSize,$seq,$
seqQual,$other)=split(/\t/); if($insertSize>10000 && $pos>75000 && $matePos <
1300000){print int($pos/1000)."\t".int($matePos/1000)}' | sort | uniq -c | sort
-nr
```

The third line 21 81 323 shows us our mis-assembly. What are the other entries?

We are fully aware that is this a quite complex piece of code, and just used as a one liner. It uses the LINUX commands sort and uniq. But keep in mind that this command canfind you all mate pairs mapping too far apart for any bam file (if you adjust the insertSize parameter for your data)!

Getting non mapping reads and their mates

Here is an example of how to get the mates of non mapping reads. It is a good example of PERL one-liners.

First we are going to get reads that don't map with PERL. The original command is:

```
$ samtools view -f 0X4 IT.Chr5.bam | head
```

In PERL this would be:

```
$ samtools view IT.Chr5.bam | perl -nle 'my
($read,$flag)=split(/\t/); if ($flag & 0x4) {print }' | head
```

Now we need to get the reads where the mate is not mapped. Looking at the samtools manual:

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate

The 0x8 tells if the mate pair is not mapped. So if the read is not mapping (0x4) or the mate is not mapping (0x8) then print the sam line into a file:

```
$ samtools view IT.Chr5.bam | perl -nle 'my
($read,$flag)=split(/\t/); if ($flag & 0x4 or $flag & 0x8)
{print }' | sort > NonmappingReadsPlusmate.sam
```

This file can now be used in VELVET for *de novo* assembly as explained in the Assembly module.

We hope that this illustrates the power of PERL one-liners!

Appendix XI Splice site information

Gene	No.	Exon	Intron	Exon	Size (bp)
41-3	1	GAA	**GTA**CACA..CCTTCTTTTTCCATATT	**TAG**|CAA	152
	2	AAT	**GTT**AAAA...TTTTTTTTTTTAAACT	**TAG**|CCG	208
	3	GAG	**GTA**AGAA...ATTCATTATATATTTA	**TAG**|GGA	86
	4	TCG	**GTA**TGGA...TTTTGAAATACTTCCT	**CAG**|TTA	152
	5	ACT	**GTA**ATAT..TTTTTTTTTTTATTTCC	**TAG**|ATG	112
	6	CAG	**GTA**AATA..ATAATGACATTTTGATA	**CAG**|ATT	120
	7	AAT	**GTA**CATT..TTATTTTTATTTATTTA	**TAG**|AAA	81
	8	TAG	**GTA**TTTG..ATATTTTTTACTTATGA	**TAG**|TTA	96
RhopH3	1	AGG	**GTA**ATAT..TTTATTTTATTTTTTTTT	**TTA**|TTT	150
	2	GGA	**GTA**AGAG..TTTTTATTATTTTATTG	**TAG**|TCC	442
	3	GGA	**GTA**AGAG..TTTTTATTATTTTATTG	**TAG**|TCC	199
	4	CAG	**GTA**YGCT..TTTAATTTTTTTTTCCT	**TCA**|TCA	160
	5	AAA	**GTA**AGAA..TATTTTTTTACAATTTT	**TAG**|TTC	206
	6	AAG	**GTA**AAAG..TTTTTTTTTTTTTGTTT	**CAG**|TTT	142
RNA pol III	1	CAG	**GTA**CATA..TTTTTTTTTTTTTTTTTT	**TAG**|GTG	158
	2	CAA	**GTA**ATTA..TATATTTTATTTTTTCT	**TAG**|GTT	113
	3	TAC	**GTT**AGTT..TTTTTTTTTTTTTTTTTT	**TAG**|TGG	169
	4	ATT	**GTA**AGTT..TATTTTTTTTTTTTTTTT	**TAG**|TGA	112
SERA	1	TGT	**GTA**AGAA..TTGTCATTATTTTTTTTT	**TAG**|GTG	158
	2	AAA	**GTA**TAAA..TTTATTTATTTTTTTTTT	**TAG**|ATA	175
	3	CAG	**GTA**AATA..TTTTAATTTTTTTGTTT	**TAG**|AAA	129
SERP H	1	CTG	**GTT**TGTC..CATATATTTCTTTATTT	**TAG**|ATA	345
	2	AGA	**GTA**AAAA..TTTCTTATATTTTCTTT	**TAG**|GTG	92
	3	CTG	**GTT**TGTC..CATATATTTCTTTATTT	**TAG**|ATA	116
Ag15	1	ATG	**GTA**AGAG..TATTTTTGATACCTTTA	**TAG**|AGT	214
	2	AAA	**GTA**ATTA..CAATCATATTAACACAA	**AAG**|ATG	280
PfGPx	1	GAG	**GTA**TACA..TTATTATTCCCTTGCTT	**TAG**|ATC	208
	2	TCG	**GTT**AGTA..TATTTATCATTTTTTTC	**CAG**|ATG	168
Calmodulin	1	GAA	**GTA**AATC..TTTTTTATTTTTCTCAT	**TAG**|CTA	480
PfPK1	1	TAG	**GTG**TGTT..TCATTACATTTTTACCT	**TAG**|GAT	101
MESA	1	TTA	**GTA**AGTT..CGTAATATATTTTTTTTT	**TAG**|GAT	122
Aldolase	1	ATG	**GTA**AGAA..TATTTTTATATTTTTTT	**TAG**|GCT	452
KAHRP	1	AAC	**GTA**AGTT..TTATTTTTTTTTTCATA	**TAG**|TGC	430
GBPH2	1	TTG	**GTA**TGCC..TTTGTATTATTTAATTT	**TAG**|AAT	157
GBP	1	TTG	**GTA**TG....TGTGTATTGTTTATTTT	**TAG**|AAT	179
FIRA	1	TGT	**GTA**AGGA..TTTTTATATTTTTTCTT	**TAG**|CGA	175
GARP	1	AAG	**GTA**ACAA..TATATGTATTTTTTTTT	**TAG**|TGC	214

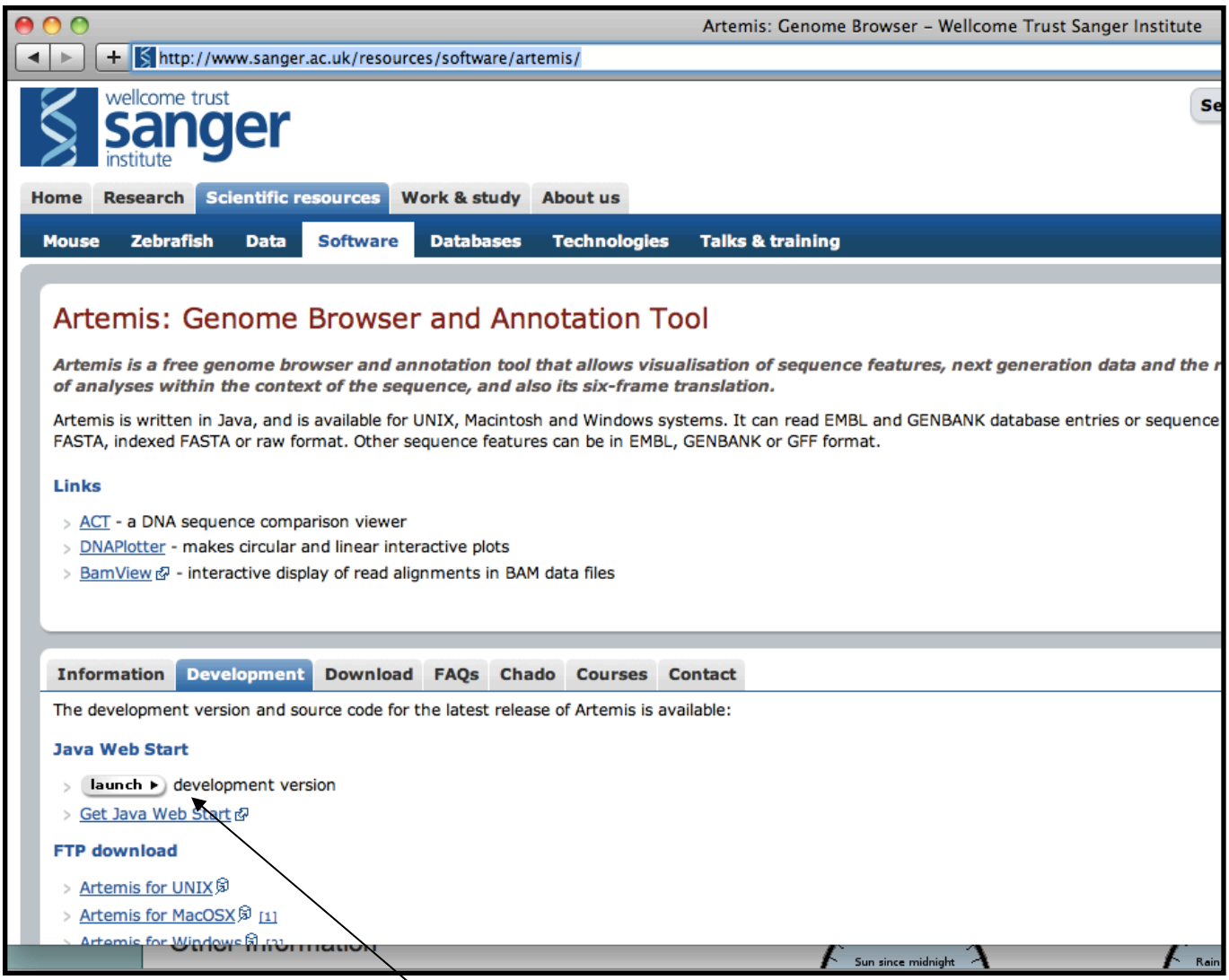
Donor motif **Acceptor motif**

The splice acceptor and donor sequences for several *P. falciparum* genes: adapted from Coppel and Black(1998). In "Malaria:Parasite Biology, Pathogenesis and Protection", I.W. Sherman (ed.); ASM Press; Washington DC; pp185-202

Appendix XII Running Artemis from the Web

To work this Artemis you don't necessary have to work with it from the VM. It can be run from the web:

http://www.sanger.ac.uk/resources/software/artemis/



Click on 'launch', accept and wait a bit…
Next you can load sequence, bam files etc.

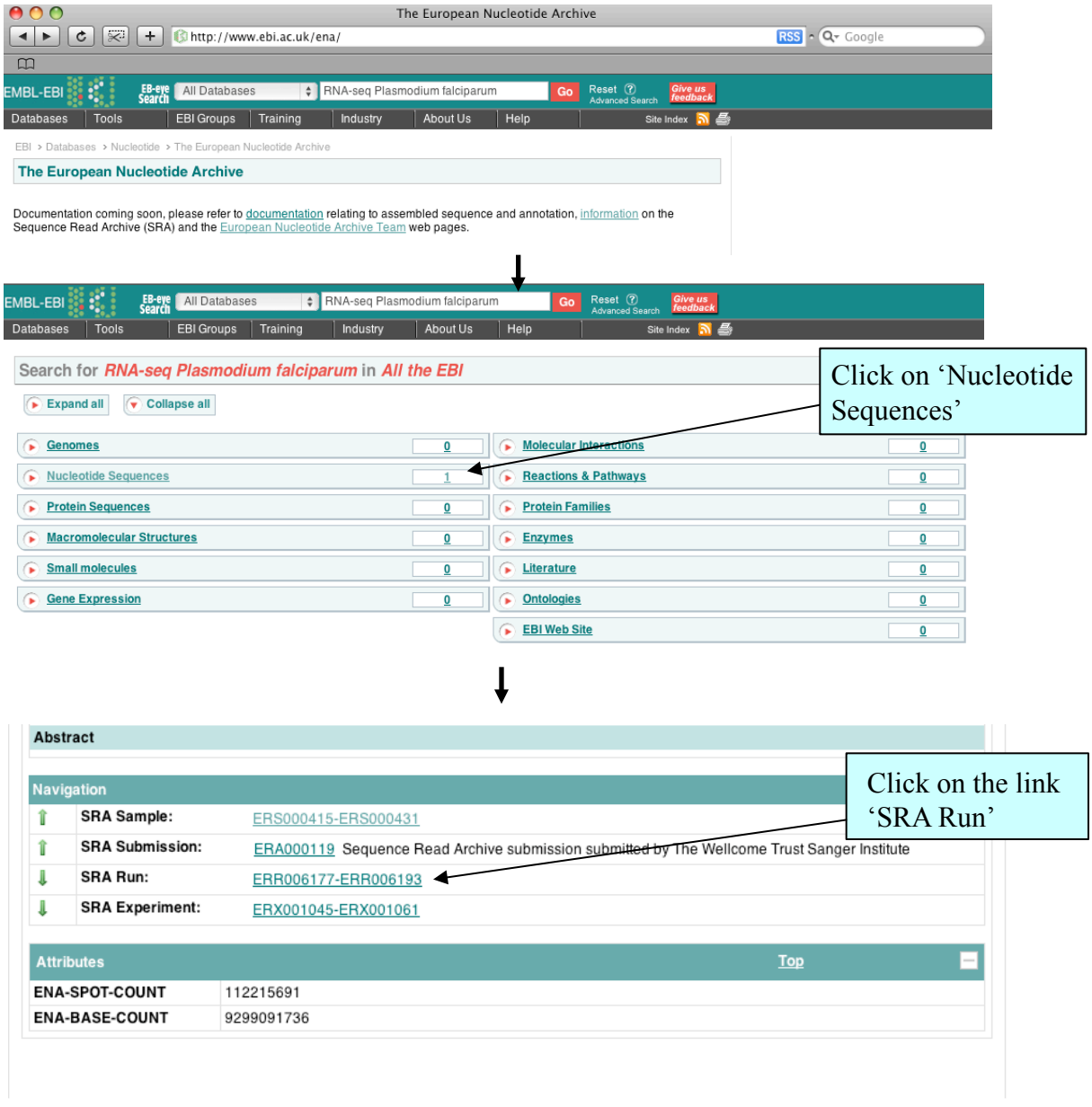
Appendix XIII Exploring the Sequence Read Archive

Exercise

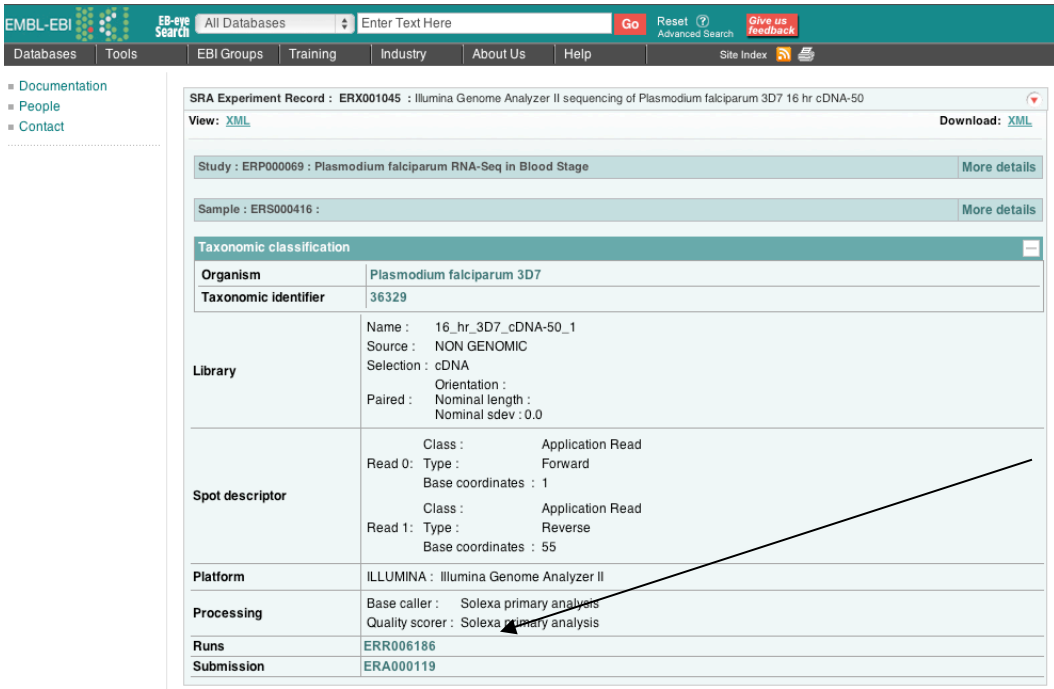
You can acceess the sequence read archive through the following sites:
http://www.ncbi.nlm.nih.gov/sra
http://www.ebi.ac.uk/ena

It's possible to download transcriptome data and other next generation sequencing data as follows. These instructions are given in the form of an exercise to help make it more interesting:

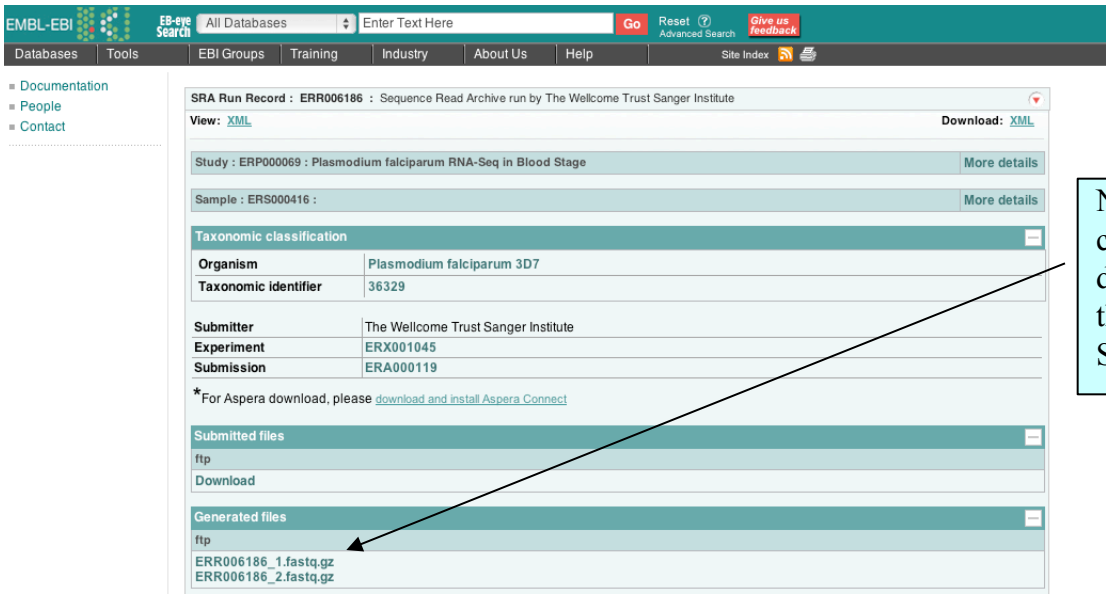
Go to the following website: http://www.ebi.ac.uk/ena and type in the search box:
RNA-seq, Plasmodium falciparum
Now follow the step-by-step instructions to download this data.



A list of all the RNA experiments will come up. Click on the red arrow to expand the window.



Click on 'Runs'



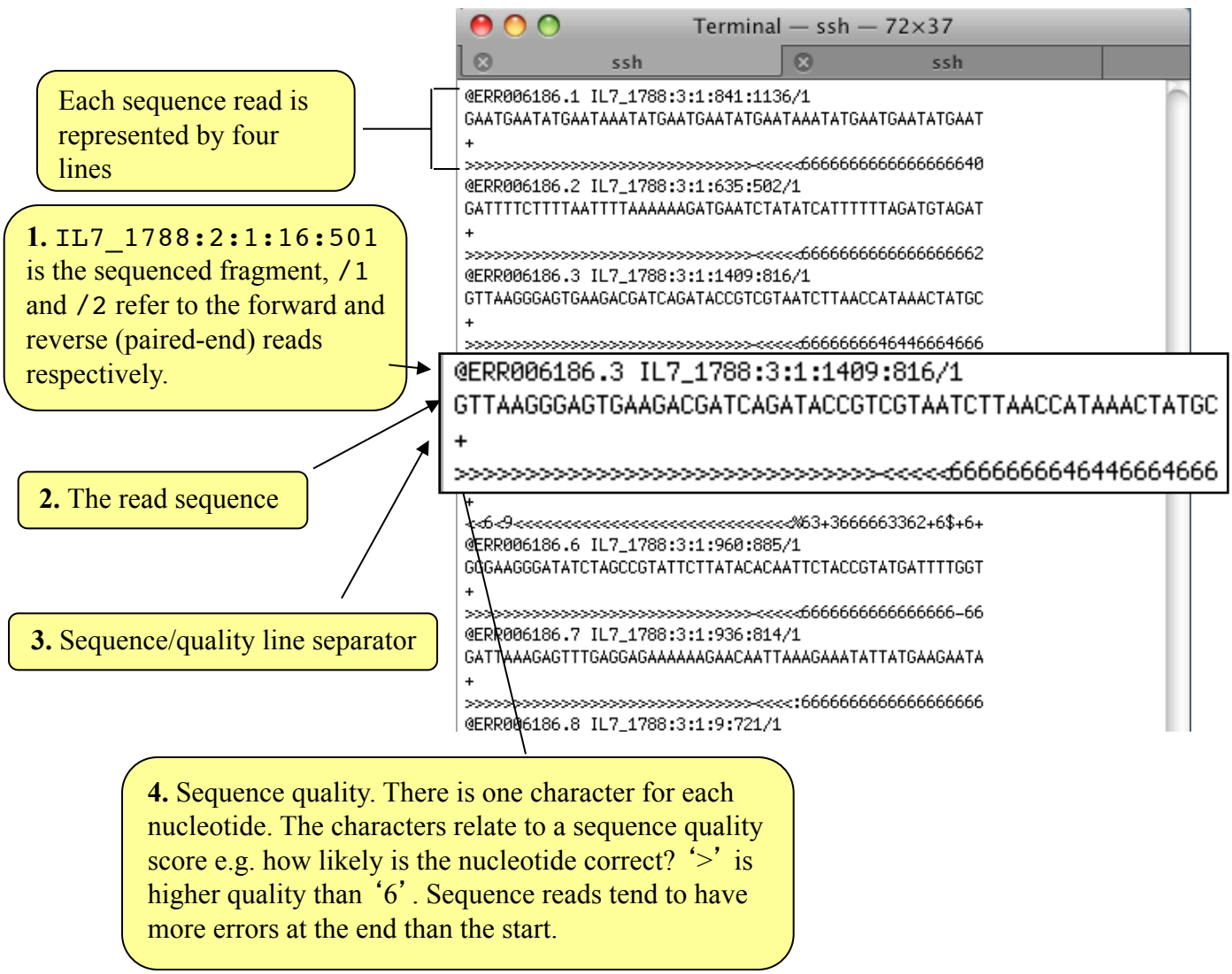
Now you can download the RNA-Seq data.

Appendix XIII continued.

Once downloaded the fastq file from the SRA, you can first have a look at the format.

Open up a terminal and navigate to where you saved the fastq file:

```
$ gunzip ERR006186_1.fastq.gz
$ more ERR006186_1.fastq
```



Each sequence read is represented by four lines

1. `IL7_1788:2:1:16:501` is the sequenced fragment, `/1` and `/2` refer to the forward and reverse (paired-end) reads respectively.

2. The read sequence

3. Sequence/quality line separator

4. Sequence quality. There is one character for each nucleotide. The characters relate to a sequence quality score e.g. how likely is the nucleotide correct? '>' is higher quality than '6'. Sequence reads tend to have more errors at the end than the start.

Appendix XIV

Here is compilations of the programs that we find useful.

Artemis & Act
http://www.sanger.ac.uk/science/tools/artemis
http://www.sanger.ac.uk/science/tools/artemis-comparison-tool-act

Mappers

SMALT: http://www.sanger.ac.uk/science/tools/smalt-0
BWA: http://sourceforge.net/projects/bio-bwa/files/
BOWTIE: http://bowtie-bio.sourceforge.net/index.shtml
TopHat: https://ccb.jhu.edu/software/tophat/index.shtml

SAMTOOLS & BCFtools
http://samtools.sourceforge.net/
https://samtools.github.io/bcftools/bcftools.html

Assemblers
Velvet: http://www.ebi.ac.uk/~zerbino/velvet/
ABYSS: http://www.bcgsc.ca/platform/bioinfo/software/abyss
SOAPdenovo: http://soap.genomics.org.cn/soapdenovo.html

Tools for automatic finishing / Annotation transfer
ABACAS: http://sourceforge.net/projects/abacas/files/
IMAGE: http://sourceforge.net/projects/image2/
iCORN: http://sourceforge.net/projects/icorn/files/
PAGIT: http://www.sanger.ac.uk/science/tools/pagit