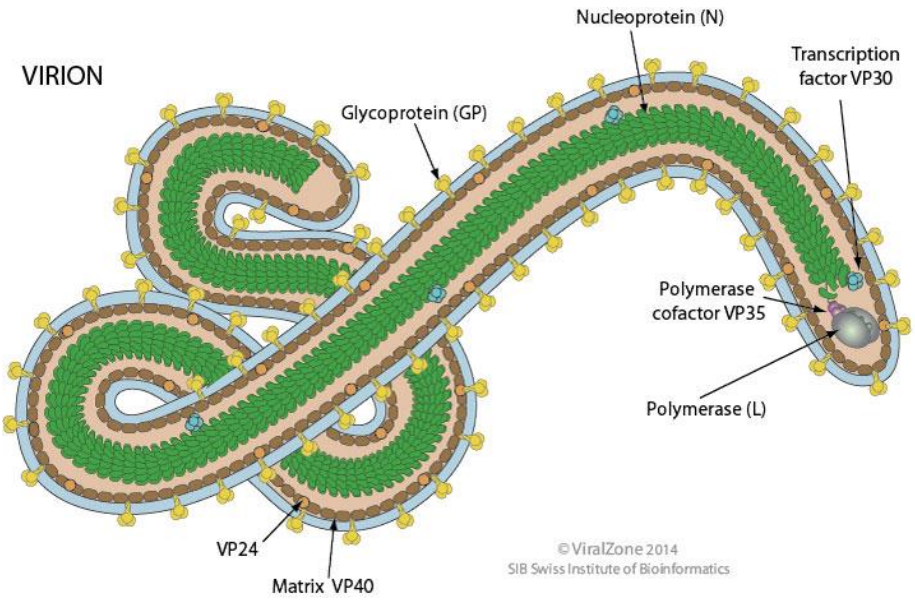
GENOMICS NORTH AUSTRALIA (GNA) 2015 WORKSHOP SERIES

Workshop #1: Whole Genome Sequence Analysis: Case Study of the Ebola Outbreaks



Workshop presenters: Dr Derek Sarovich and Dr Erin Price, Menzies School of Health Research, Royal Darwin Hospital campus

derek.sarovich@menzies.edu.au; erin.price@menzies.edu.au

GNA workshop coordinator: Dr Robyn Marsh, Menzies School of Health Research, Royal Darwin Hospital campus

robyn.marsh@menzies.edu.au

Date and time: 29th September 1–4pm and 30th September 8:30am–4.30pm

Location: CDU Campus, Building Red 9, Room 1.45

Workshop description

Next-Generation Sequencing (NGS) has revolutionised the biological sciences. The introduction of the first NGS technology a decade ago has paved the way for inexpensive, high-throughput genomic platforms that are now readily accessible to laboratories across the globe. The ability to manipulate and interpret NGS data is emerging as an essential and highly sought-after skill for molecular biologists.

This introductory workshop is aimed at teaching participants how to use NGS data analysis tools in the Linux environment; specifically, to perform whole genome sequence (WGS) analysis. You will be shown how to run basic commands in Linux to analyse and interpret WGS data, using the Ebola outbreak as a case study.

By the end of the workshop, you will know how to:

- Setup and access a Linux environment via the CDU-Menzies supercomputer, Cheetah.
- Use basic command line syntax in a Linux environment, including how to use the Portable Batch System resource manager (PBS).
- Analyse and interpret WGS data outputs from the African Ebola outbreaks using the genomics pipeline SPANDx and downstream bioinformatics tools.
- Use these tools to analyse your own NGS data.

No prior experience with NGS or WGS data is required!

Common basic commands when working with in the Cheetah Linux environment

Note: This list is by no means comprehensive. Please use Google to find Linux commands not listed here.

`cd <path>`	Change from one directory (i.e. folder) to another.
`pwd`	Present working directory.
`ls`	List files and folders in your `pwd`.
`mkdir <dirname>`	Create a new directory.
`rmdir <dirname>`	Remove (delete) directory.
`rm <filename>`	Remove file.
`ln –s <path/filename>`	Create a single file symbolic link (i.e. shortcut) from another directory in your `pwd`.
`ln –s <path>/*gz .`	Create a symlink to all files with the prefix '`gz`'.
`qsub -I -l walltime=96:00:00`	Generic interactive qsub command recommended for small jobs, or when you want to watch the process of your jobs.
`exit`	Exit interactive qsub mode (warning: will exit you out of PuTTY/Terminal if you aren't in interactive qsub mode!) .
`qstat` (and `qstat|more`)	`qstat` tells you the status of your jobs in the Cheetah queue; adding `|more` gives you a page-by-page list (sometimes multiple pages long) .
`showq` (and `showq|more`)	`showq` tells you the status of *all* jobs in the Cheetah queue, not just your own. Useful for determining current queuing load on the system.
`clear`	Clear any text in your command prompt, giving you a fresh line to work with.
`head -n20 <filename>`	Shows the first 20 lines of the file.
`tail -n50 <filename>`	Show the last 50 lines of the file.

For larger or time-consuming jobs not suitable for interactive qsub, use the following command:

```
qsub –v command="<command to run>" /home/dsarovich/bin/Header.pbs
```

Some helpful pointers when working in the Cheetah Linux environment

- Don't use special characters, including spaces, in directory or file names. Stick with underscores and alphanumeric numbering (e.g. Don't name a file `60294NP 1 sequence.fastq.gz`; instead, name it `60264NP_1_sequence.fastq.gz`). Addition of special characters will almost certainly cause unintended errors/failures during data analysis.
- The Tab key is exceptionally helpful for filling in command line text so you don't have to type whole path names and files out in full, which gets tiring very quickly. Tab will autofill text only when the directory/file is unique. Hit Tab twice if the text is not unique; the command prompt will list options. Tabbing through your commands is also strongly recommended for double-checking your spelling; if it won't autofill, it means you've got a typo that you need to fix.
- Use the 'up' key to bring up previous commands that you've typed.
- The 'home' and 'end' keys can be helpful for navigating more quickly through long lines of command line text.
- Simply highlighting text in the command line will copy text; don't use Ctrl+c (or Command+c for Macs), which will kill a running command and give you a new line.
- To paste text at the command prompt, use Shift+Insert, or the right-click mouse button.
- You can use * as a wildcard character for your commands, just like with DOS in the old days!
- There are two main locations for software installed on Cheetah: `/usr/local/` and `/home/dsarovich/bin`. Derek is happy for people to use software installed in his bin, but encourages users to install their own software in their `/home/usr/bin` directories, especially specialised software.
- Use Ctrl+r (or Command+r for Macs) to refresh your WinSCP or Fugu window. Contents displayed in WinSCP and Fugu do not update automatically.

TOPIC 1. How to perform quality assessment of next-generation sequencing (NGS) data

NGS data quality control (QC) is sometimes (frequently?!) overlooked, but it cannot be stressed enough how important it is that you use QC tools to understand any shortcomings of your data before you begin doing genomic analyses. Low-quality NGS data leads to poorer alignments, lower-quality *de novo* assemblies and potentially even incorrect assumptions being made about your data.

In Exercise 1 of this workshop, you will learn how to use the free program FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) to assess the quality of Illumina and Ion Torrent NGS reads. We will examine two paired-end Illumina datasets from the same strain: a simulated *Haemophilus influenzae* Illumina dataset (`60294NP_simulated*`) generated using the program ART (you'll learn how to do this in the next part of the workshop), and a real *H. influenzae* Illumina dataset (`60294NP_real*`). We'll also examine a real single-end Ion Torrent dataset (`60294NP_IonTorrent_1_sequence.fastq.gz`) to see how NGS data differs between sequencing platforms.

FastQC tells you lots of useful info about your read data so is especially helpful if you're new to NGS.

a. Launch WinSCP and PuTTY (or Fugu and Terminal if using a Mac).
b. Log into Cheetah using your username and password.
c. In the PuTTY/Terminal window, open an interactive qsub session: `qsub -I -l walltime=96:00:00`
d. Create a new directory by typing: `mkdir FastQC_analysis`
e. Change into this new directory by typing: `cd FastQC_analysis`
f. Symbolically link (i.e. 'create a shortcut') to the *H. influenzae* Illumina reads by typing: `ln -s /data/Ebola/FastQC_datasets/NTHi/*gz .` (yes, the full stop at the end is important!). This command will symlink all files with the suffix 'gz'.
g. Type `ls` to list the contents of your `FastQC_analysis` directory - you should see the following files:

```
eprice@cheetah:~/FastQC_analysis$ ls
60294NP_IonTorrent_1_sequence.fastq.gz  60294NP_simulated_1_sequence.fastq.gz
60294NP_real_1_sequence.fastq.gz        60294NP_simulated_2_sequence.fastq.gz
60294NP_real_2_sequence.fastq.gz
```

h. Next, type in the path to FastQC on Cheetah, followed by the file/s you wish to analyse. You can either analyse a single read, a pair of reads, or all reads in your directory as follows:

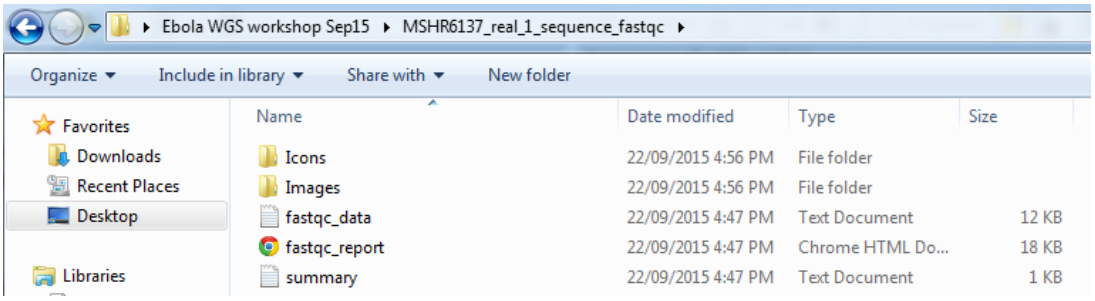
Single read: `/home/dsarovich/bin/fastqc 60294NP_real_1_sequence.fastq.gz`
Pair of reads: `/home/dsarovich/bin/fastqc 60294NP_real*`
All reads (batch analysis [recommended]): `/home/dsarovich/bin/fastqc *gz`

i. Once all the reads have been processed by FastQC (will take ~5 min), copy across the five `*_fastqc` directories (shown below) to your laptop.

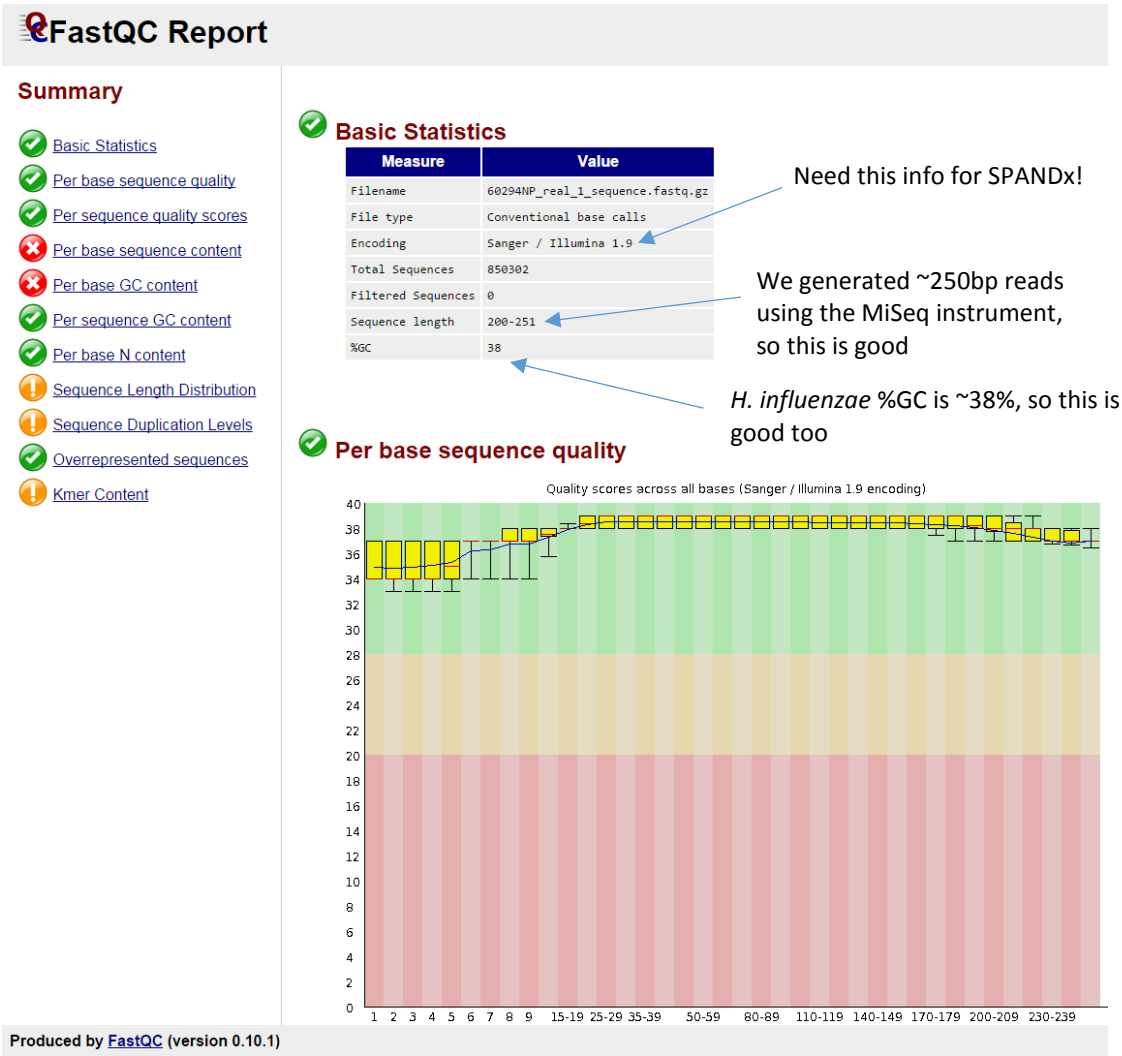
Name	Size	Changed	Rights	Owner
..		22/09/2015 4:16:18 PM	rwxr-xr-x	eprice
60294NP_IonTorrent_1_sequence_fastqc		22/09/2015 6:42:01 PM	rwxr-xr-x	eprice
60294NP_real_1_sequence_fastqc		22/09/2015 6:41:56 PM	rwxr-xr-x	eprice
60294NP_real_2_sequence_fastqc		22/09/2015 6:41:51 PM	rwxr-xr-x	eprice
60294NP_simulated_1_sequence_fastqc		22/09/2015 6:41:46 PM	rwxr-xr-x	eprice
60294NP_simulated_2_sequence_fastqc		22/09/2015 6:41:43 PM	rwxr-xr-x	eprice

Note: The forward (_1) and reverse (_2) paired-end Illumina reads are analysed by FastQC separately. You'll see why later when you run through the FastQC exercises.

j. Open the `fastqc_report` files with your favourite internet browser by double-clicking on the html icon.



The beginning of the report for `60294NP_real_1_sequence.fastq.gz` should look like this:



k. Scroll through the rest of the FastQC report to learn more about the NGS data.

CLASS EXERCISES TO DO DURING WORKSHOP

Class exercise #1a: Compare the forward and reverse read FastQC reports from the 'real' Illumina 60294NP datasets. What are the commonalities and differences between them?

Class exercise #1b: Compare the Illumina reads for the 'real' and 'synthetic' datasets for 60294NP. What are the commonalities and differences between them?

Class exercise #1c: Compare the Illumina and Ion Torrent reads for 60294NP. What are the commonalities and differences between them (NB. The Ion Torrent reads were generated by a particularly inept sequence provider who shall remain nameless, and should thus not be considered representative of Ion Torrent read data)?

Class exercise #1d: Perform quality filtering of the real paired 60294NP Illumina reads, and compare these filtered reads with the original unfiltered reads. What has improved post-filtering?

Below is the command for filtering the paired Illumina reads (will take ~5-10min to process):

```
java -jar /home/dsarovich/bin/Trimmomatic-0.33/trimmomatic-0.33.jar PE -
threads 1 -phred33 60294NP_real_1_sequence.fastq.gz
60294NP_real_2_sequence.fastq.gz 60294NP_real_pairedoutput_1.fq.gz
60294NP_real_1_unpaired.fq.gz 60294NP_real_pairedoutput_2.fq.gz
60294NP_real_2_unpaired.fq.gz ILLUMINACLIP:/home/dsarovich/bin/Trimmomatic-
0.33/adapters/TruSeq2-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15
MINLEN:36
```

The filtered reads will be named: `60294NP_real_pairedoutput_1.fq.gz` and `60294NP_real_pairedoutput_2.fq.gz`.

TOPIC 2. How to create simulated NGS data

The SPANDx comparative genomics pipeline (http://sourceforge.net/projects/spandx/) you'll be using throughout this workshop requires `.fastq.gz` files as input. However, sometimes you'll want to include genomes in your analysis that aren't in this format; for example, you may wish to include publicly available genomes on GenBank (`.fasta` format). In Exercise 2 of this workshop, you'll learn how to create synthetic NGS reads from `.fasta` files using the ART package (http://www.niehs.nih.gov/research/resources/software/biostatistics/art/).

> All the Ebola genomes you'll be analysing in this workshop comprise synthetic reads that were generated from `.fasta` files using ART!

a. Follow steps a-c from Exercise 1 above if you are not already logged onto Cheetah and in an interactive qsub session.
b. Create a new directory by typing: `mkdir simulated_reads`
c. Change into this new directory by typing: `cd simulated_reads`
d. Browse the Ebola `.fasta` genomes available on Cheetah by typing: `ls /data/Ebola/fasta_files/*fasta`. The top of the list should look like this:

```
eprice@cheetah:~/simulated_reads$ ls /data/Ebola/fasta_files/*fasta
/data/Ebola/fasta_files/1Eko_Gabon_1996.fasta
/data/Ebola/fasta_files/1Luebo_Congo_2007.fasta
/data/Ebola/fasta_files/23Luebo_Congo_2007.fasta
/data/Ebola/fasta_files/2Nza_Gabon_1996.fasta
/data/Ebola/fasta_files/4Luebo_Congo_2007.fasta
/data/Ebola/fasta_files/Boende_Lokolia_Congo_2014.fasta
```

e. From this list, choose between two and four Ebola genomes you'd like to convert to synthetic reads, and symlink them into your `pwd` by typing: `ln -s /data/Ebola/fasta_files/<filename>`. Create symlinks for each of the genomes you want to convert to `.fastq` format.
f. Next, run ART. We'll be creating synthetic 100bp paired-end Illumina reads at 85X coverage with an insert size of 500bp and quality shifts of 10. You can either run ART across a single `.fasta` genome, or across a batch of `.fasta` genomes. The commands for both are listed below:

Single (for your info only; please don't run today):

```
/home/dsarovich/bin/ART/art_illumina -i 1Eko_Gabon_1996.fasta -p -l 100
-f 85 -m 500 -s 10 -qs 10 -qs2 10 -na -o 1Eko_Gabon_1996.fastq
```

Batch (recommended) – keep in mind you must exit interactive qsub before running this command:

```
for f in *.fasta; do qsub -N batch_ART -v
command="/home/dsarovich/bin/ART/art_illumina -i $f -p -l 100 -f 85 -m
500 -s 10 -qs 10 -qs2 10 -na -o ${f}_out"
/home/dsarovich/bin/Header.pbs; done;
```

g. ART should quickly process these small Ebolavirus genomes (<10 sec) to generate "Illumina" FASTQ files. Check the status of your jobs by typing `qstat`. Once jobs are completed, your `qstat` list should be blank. You can then check your output files by typing `ls`. Your directory should look something like this:

```
eprice@cheetah:~/simulated_reads$ ls
1Eko_Gabon_1996.fasta           Makona_Kouroussa_Guinea_531_2014.fasta
1Eko_Gabon_1996.fasta_out1.fq   Makona_Kouroussa_Guinea_531_2014.fasta_out1.fq
1Eko_Gabon_1996.fasta_out2.fq   Makona_Kouroussa_Guinea_531_2014.fasta_out2.fq
batch_ART.o228864               Zaire_Yambuku_Mayinga_1976.fasta
batch_ART.o228865               Zaire_Yambuku_Mayinga_1976.fasta_out1.fq
batch_ART.o228866               Zaire_Yambuku_Mayinga_1976.fasta_out2.fq
```

h. These files are a little ugly-looking and not yet fully compatible with SPANDx, so we'll do some clean-up on them using the '`for f in`' command, which is a useful command for batch jobs (we also used it for running ART above). Run the following two commands to clean up your directory (you can paste them both in at the same time; should take <5sec to complete):

```
for f in *out1.fq; do mv $f ${f//.fasta_out1.fq/_1_sequence.fastq};
done;
for f in *out2.fq; do mv $f ${f//.fasta_out2.fq/_2_sequence.fastq};
done;
```

i. Finally, `gzip` these files to compress them, and use `qstat` to determine when they're done (should take <10sec). Compression saves space on the Cheetah hard drives. The `.gz` format is a common file format on Linux, and is similar to `.zip` and `.rar` compression formats.

```
for f in *fastq; do qsub -v command="gzip ${f}"
/home/eprice/bin/Header.pbs; done;
```

j. That's it! Your final files should look something like this:

```
eprice@cheetah:~/simulated_reads$ ls
1Eko_Gabon_1996_1_sequence.fastq.gz  Header.pbs.o229039  Makona_Kouroussa_Guinea_531_2014_1_sequence.fastq.gz
1Eko_Gabon_1996_2_sequence.fastq.gz  Header.pbs.o229040  Makona_Kouroussa_Guinea_531_2014_2_sequence.fastq.gz
1Eko_Gabon_1996.fasta                Header.pbs.o229041  Makona_Kouroussa_Guinea_531_2014.fasta
batch_ART.o228864                    Header.pbs.o229042  Zaire_Yambuku_Mayinga_1976_1_sequence.fastq.gz
batch_ART.o228865                    Header.pbs.o229043  Zaire_Yambuku_Mayinga_1976_2_sequence.fastq.gz
batch_ART.o228866                    Header.pbs.o229044  Zaire_Yambuku_Mayinga_1976.fasta
```

SUGGESTED EXERCISES TO DO AFTER WORKSHOP (OR DURING IF YOU'RE KEEN!)

Exercise #2a: Run FastQC across your synthetic Ebolavirus genomes. This analysis should tell you that you have 100bp Illumina reads, and the other metrics should look very similar to the `60294NP_simulated*` reads that you examined in Exercise 1.

Exercise #2b: Try running ART with different parameters, or even with different NGS technologies! Note: the current version of ART installed on Cheetah (VanillaIceCream) will only simulate data for Illumina, SOLiD and 454 reads.

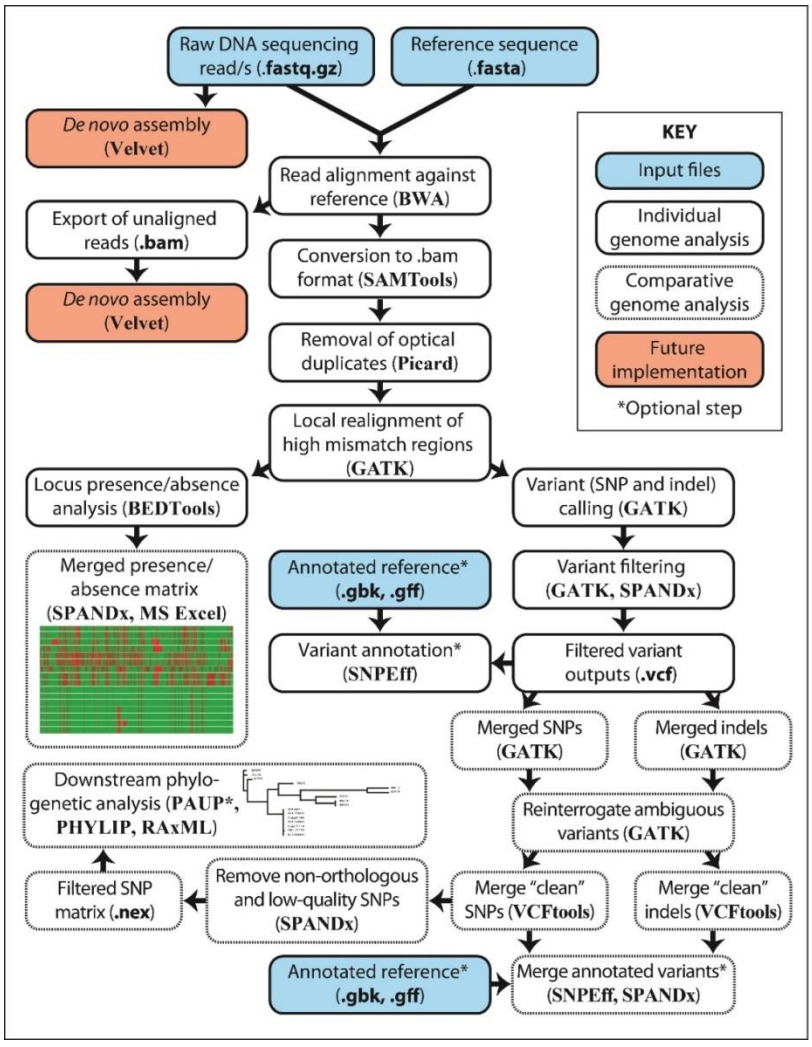
Exercise #2b: Try ART out on public genomes of interest to you in your research. Note: The VanillaIceCream version of ART currently installed on Cheetah will not be able to create synthetic reads from genomes with contigs less than ~800bp in length using the workshop parameters. You can get around this issue by deleting these short contigs from your reference before running ART.

TOPIC 3. How to perform comparative genomic analysis of NGS data

This exercise is the major component of today's workshop. Now that you've learned how to assess and filter NGS data, and how to simulate NGS reads, you are ready to identify genetic differences amongst an Ebolavirus genome dataset!

You will be using the comparative genomics pipeline SPANDx (Synergised Pipeline for Analysis of NGS Data in Linux) to identify differences amongst Ebolavirus strains from all major outbreaks affecting human populations since 1976. SPANDx will identify almost all genetic variants (single-nucleotide polymorphisms (SNPs), insertions-deletions (indels) and larger deletions) from your NGS data in comparison to a reference genome – only a small number of variants cannot be detected with this method, for reasons we'll discuss at the end of the workshop. SPANDx manages the resource request for you on Linux cluster systems, so before running SPANDx, make sure that you have exited out of interactive qsub (`exit`).

SPANDx was written at Menzies by Derek and Erin, so we know a lot about it! An overview of what SPANDx does is shown below:



From Sarovich and Price, 2014: http://www.biomedcentral.com/1756-0500/7/618

TOPIC 3.1: How to run SPANDx across the Ebolavirus outbreak genomes

Today we are going to compare Ebolavirus genomes from the most recent Ebola outbreak with previous Ebolavirus outbreaks (*n*=76). By the end of the analysis you will know how to identify virtually all genetic differences amongst these strains.

To save time, we have already downloaded the Ebolavirus genomes that you will use for your analysis. They are in `/data/Ebola` (*`gz` files). These 'Illumina' `.fastq.gz` reads were simulated from downloaded `.fasta` files using the same methods we just covered in Topic 2.

The reference genome for our analysis is in `.fasta` format and is in the same directory as the NGS reads: `Makona_Kissidougou_Guinea_C15ref_2014.fasta`. This Guinean strain was the same reference used in the recent Ebola paper (Gire *et al*., 2014: http://www.sciencemag.org/content/345/6202/1369.long). We have chosen this genome because it is annotated, which means that we can use this information to identify mutational effects on protein function. The GenBank link for this genome is: http://www.ncbi.nlm.nih.gov/nuccore/KJ660346.

a. Create a new analysis directory: `mkdir ~/analysis/Ebola`
b. Change into this directory by typing: `cd ~/analysis/Ebola`
c. Symlink the NGS reads into your analysis directory (including the full stop!): `ln -s /data/Ebola/*.gz .`
d. Symlink the reference genome into the same directory (no full stop required when symlinking a single file): `ln -s /data/Ebola/Makona_Kissidougou_Guinea_c15ref_2014.fasta`
e. At this stage, it is good practice to check your directory structure and the files contained in the current directory to make sure that you have all of the `.fastq.gz` files that you want to process, to ensure you have the correct reference sequence in `.fasta` format, and to make sure that none of your symlinks are 'dead' links. Type: `ls` to view your symlinked files. Your directory should look something like the screenshot below:

```
dsarovich@cheetah:~/analysis/Ebola$ ls
1Eko_Gabon_1996_1_sequence.fastq.gz                Makona_G3809_SierraLeone_2014_1_sequence.fastq.gz
1Eko_Gabon_1996_2_sequence.fastq.gz                Makona_G3809_SierraLeone_2014_2_sequence.fastq.gz
1Luebo_Congo_2007_1_sequence.fastq.gz              Makona_G3810.1_SierraLeone_2014_1_sequence.fastq.gz
1Luebo_Congo_2007_2_sequence.fastq.gz              Makona_G3810.1_SierraLeone_2014_2_sequence.fastq.gz
23Luebo_Congo_2007_1_sequence.fastq.gz             Makona_G3814_SierraLeone_2014_1_sequence.fastq.gz
23Luebo_Congo_2007_2_sequence.fastq.gz             Makona_G3814_SierraLeone_2014_2_sequence.fastq.gz
2Nza_Gabon_1996_1_sequence.fastq.gz                Makona_G3816_SierraLeone_2014_1_sequence.fastq.gz
2Nza_Gabon_1996_2_sequence.fastq.gz                Makona_G3816_SierraLeone_2014_2_sequence.fastq.gz
4Luebo_Congo_2007_1_sequence.fastq.gz              Makona_G3817_SierraLeone_2014_1_sequence.fastq.gz
4Luebo_Congo_2007_2_sequence.fastq.gz              Makona_G3817_SierraLeone_2014_2_sequence.fastq.gz
Boende_Lokolia_Congo_2014_1_sequence.fastq.gz      Makona_G3818_SierraLeone_2014_1_sequence.fastq.gz
Boende_Lokolia_Congo_2014_2_sequence.fastq.gz      Makona_G3818_SierraLeone_2014_2_sequence.fastq.gz
Conacry_192_Guinea_2014_1_sequence.fastq.gz        Makona_G3819_SierraLeone_2014_1_sequence.fastq.gz
Conacry_192_Guinea_2014_2_sequence.fastq.gz        Makona_G3819_SierraLeone_2014_2_sequence.fastq.gz
GAB_Gabon_1994_1_sequence.fastq.gz                 Makona_G3820_SierraLeone_2014_1_sequence.fastq.gz
GAB_Gabon_1994_2_sequence.fastq.gz                 Makona_G3820_SierraLeone_2014_2_sequence.fastq.gz
```

f. You should now be ready to run SPANDx! Check to make sure that you aren't in interactive qsub (the command prompt should say `$cheetah;` if you are still in interactive qsub you will see `$cheetah` followed by a number e.g. `$cheetah06:`). To run SPANDx, type the full path to the SPANDx script: `/home/dsarovich/bin/SPANDx_v2.7/SPANDx.sh`. You should see the following flag options available in SPANDx:

```
dsarovich@cheetah:~/analysis/Ebola$ /home/dsarovich/bin/SPANDx_v2.7/SPANDx.sh
USAGE: SPANDx.sh <parameters, required> -r <reference, without .fasta extension> [parameters, optional] -o [organis
m] -m [generate SNP matrix yes/no] -i [generate indel matrix yes/no] -a [include annotation yes/no] -v [Variant gen
ome file. Name must match the SnpEff database] -s [Specify read prefix to run single strain or none to just constru
ct SNP matrix] -t [Sequencing technology used Illumina/Illumina_old/454/PGM] -p [Pairing of reads PE/SE] -w [BEDToo
ls window size in base pairs]
```

g. We will only be using some of the possible SPANDx options for the current tutorial. If you get stuck while performing your own analyses, please refer to the SPANDx manual (available at https://sourceforge.net/projects/spandx/) for more information.
    - The only essential flag in the SPANDx command line is `-r` (**r**eference genome), which specifies your reference genome file. This flag will generate a locus presence-absence matrix, which is basically a heatmap showing read mapping across the genome; however, it will not generate any other matrices. For most analyses, you will generally also want to use `-m` (generates a SNP **m**atrix).
    - By default, SPANDx will assume all NGS reads are paired-end and in Illumina v1.8+ quality encoding format. Strains need to be named `<strainname1>_1_sequence.fastq.gz` for the forward read and `<strainname1>_2_sequence.fastq.gz` for the reverse read, and these files need to be in the analysis directory. If you have single-end data, you need to use the `-p` (read **p**airing) flag. We don't have any single-end reads so you don't need to use this flag today.
    - Use `-t` (NGS **t**echnology) to process NGS data in old Illumina encoding (pre-v1.8) or other NGS formats (Ion Torrent and 454). As we're using Illumina v1.8+ encoding for the Ebolavirus analysis, you won't need to specify this parameter today.
    - SPANDx doesn't process indels by default, so we'll switch on `-i yes` for the current dataset (generates an **i**ndel matrix). We'll also switch on the annotation feature using the `-a` (**a**nnotation) and `-v` (reference for **v**ariant annotation) flags.
    - SPANDx will handle all the resource requests for you so you don't have to worry about the `qsub` command for SPANDx.

h. To run SPANDx across the entire Ebolavirus dataset, type in the following command:
```
/home/dsarovich/bin/SPANDx_v2.7/SPANDx.sh -r
Makona_Kissidougou_Guinea_C15ref_2014 -m yes -i yes -a yes -v
ebola_zaire
```
   If there is no queue for resources on Cheetah, SPANDx should only take ~10-15 minutes to process as the Ebolavirus genome is very small. In contrast, a typical bacterial genome analysis takes between 30 minutes to ~48h to process depending on genome complexity, NGS depth and the number of strains, and a single human genome (analysed using a diploid-compatible version of SPANDx) typically takes between 2-3 weeks to complete.

i. To check the status of your jobs, type `qstat` to display your currently queued and running jobs, or `showq` to display the entire user load on Cheetah. Due to the large number of jobs that get run by SPANDx, Cheetah can sometimes have a hard time keeping up with demand. If your jobs get stuck, you can either wait for them to complete (~30 minutes) or you can delete your stuck jobs (`qdel` followed by the job number or `qdel all` to kill everything) and repeat your previous `SPANDx.sh` command.

TOPIC 3.2: How to analyse SPANDx outputs from the Ebolavirus outbreak genomes

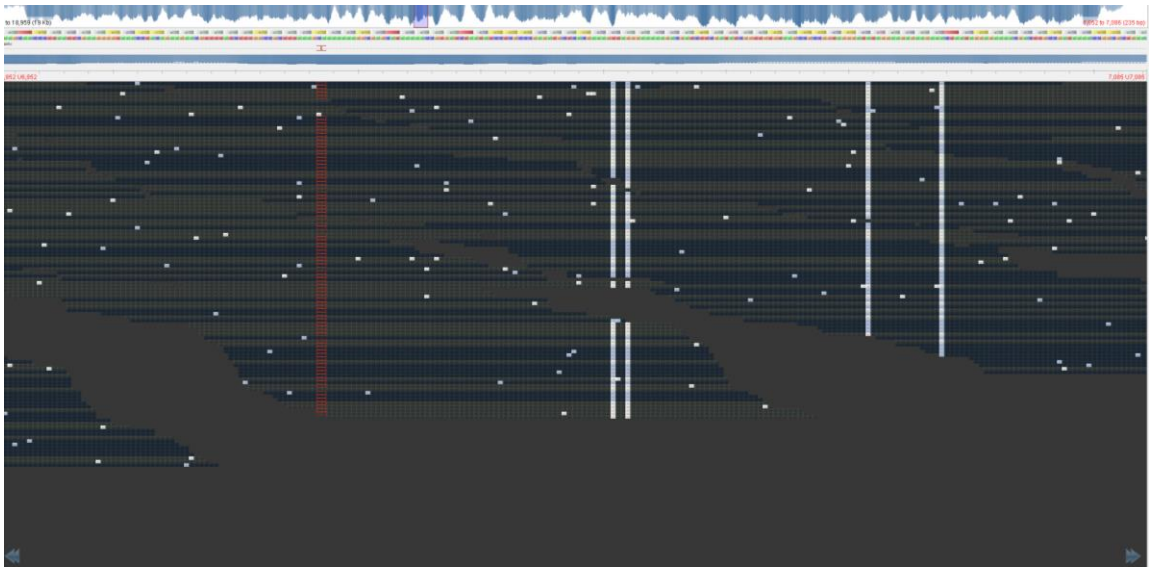
3.2.1 Individual SNP and indel variants. The initial step SPANDx will perform is SNP and indel identification for each genome. This process produces a binary alignment file (`.bam`), which is an alignment of NGS reads against your reference genome. This `.bam` file contains useful information about your alignment, including depth of coverage and read quality. You can visualise `.bam` files in the NGS viewer Tablet.

a. Using Fugu or WinSCP, navigate to the individual genome directory for `Zaire_1995` (`analysis/Ebola/Zaire_1995/unique/`). Copy across the `Zaire_1995.realigned.bam` and `Zaire_1995.realigned.bai` files to your laptop.

b. Next, navigate to `analysis/Ebola/` and copy across the `Makona_Kissidougou_Guinea_C15ref_2014.fasta` reference file.

CLASS EXERCISES TO DO DURING WORKSHOP

Class exercise #3.2.1a: Open up the above files in Tablet and go to the genomic position 6917. What do you see?



Class exercise #3.2.1b: Try opening a second alignment file by copying the `.bam` files across to your laptop and see if you can find a SNP or indel.

c. The individual SNPs and indels generated by SPANDx are also listed in the `analysis/Ebola/Outputs/SNPs_indels_PASS/` directory (when they pass the default SPANDx filters) or in `analysis/Ebola/Outputs/SNPs_indels_FAIL/` (when they have been removed from the analysis because they didn't meet one of the SPANDx filtering requirements). The format of these files is variant call format, or `.vcf`. These files are not very human friendly (a snapshot of a typical `.vcf` file is shown below) but they can contain a lot of information about the quality and reliability of your variant calls. If you are doing analyses on your own data and you are seeing unexpected results, the `.vcf` files are the first outputs you should check. All of the filters in SPANDx are completely customisable to be able to cope with varied datasets, so use your `.vcf` files to optimise parameters for your dataset if the default SPANDx parameters don't work well with your data.

```
KJ660346    6719    .    T    C    2858    PASS    AC=1;AF=1.00;AN=1;DP=77;Dels=0.00;FS=0.000;HaplotypeScore=12
KJ660346    6734    .    C    T    2493    PASS    AC=1;AF=1.00;AN=1;DP=74;Dels=0.00;FS=0.000;HaplotypeScore=12
KJ660346    6842    .    C    T    3368    PASS    AC=1;AF=1.00;AN=1;DP=92;Dels=0.00;FS=0.000;HaplotypeScore=11
KJ660346    6980    .    C    G    2941    PASS    AC=1;AF=1.00;AN=1;BaseQRankSum=1.653;DP=87;Dels=0.00;FS=0.00
KJ660346    7029    .    A    G    2804    PASS    AC=1;AF=1.00;AN=1;DP=73;Dels=0.00;FS=0.000;HaplotypeScore=6.
KJ660346    7044    .    A    C    3337    PASS    AC=1;AF=1.00;AN=1;DP=80;Dels=0.00;FS=0.000;HaplotypeScore=5.
KJ660346    7112    .    A    G    2669    PASS    AC=1;AF=1.00;AN=1;BaseQRankSum=1.552;DP=77;Dels=0.00;FS=0.00
KJ660346    7168    .    T    C    2863    PASS    AC=1;AF=1.00;AN=1;DP=80;Dels=0.00;FS=0.000;HaplotypeScore=8.
KJ660346    7181    .    A    C    3422    PASS    AC=1;AF=1.00;AN=1;BaseQRankSum=1.318;DP=94;Dels=0.00;FS=0.00
KJ660346    7204    .    T    C    3780    PASS    AC=1;AF=1.00;AN=1;DP=106;Dels=0.00;FS=0.000;HaplotypeScore=1
KJ660346    7251    .    G    A    3427    PASS    AC=1;AF=1.00;AN=1;DP=101;Dels=0.00;FS=0.000;HaplotypeScore=1
KJ660346    7264    .    T    C    3079    PASS    AC=1;AF=1.00;AN=1;DP=94;Dels=0.00;FS=0.000;HaplotypeScore=14
KJ660346    7268    .    G    A    2941    PASS    AC=1;AF=1.00;AN=1;DP=91;Dels=0.00;FS=0.000;HaplotypeScore=10
```

CLASS EXERCISE TO DO DURING WORKSHOP

Class exercise #3.2.1c: What do the different columns in the `.vcf` file mean?

Class exercise #3.2.1d: Using WinSCP or Fugu, open the `Zaire_1995.snps.PASS.vcf` file. Can you find all of the variants in both your Tablet viewer and in the `.vcf` file? Do their quality scores make sense based on the visual output in Tablet?

Class exercise #3.2.1e (optional): Using WinSCP or Fugu, open the `Zaire_1995.snps.FAIL.vcf` file. Can you see why these SNPs failed default SPANDx quality filters?

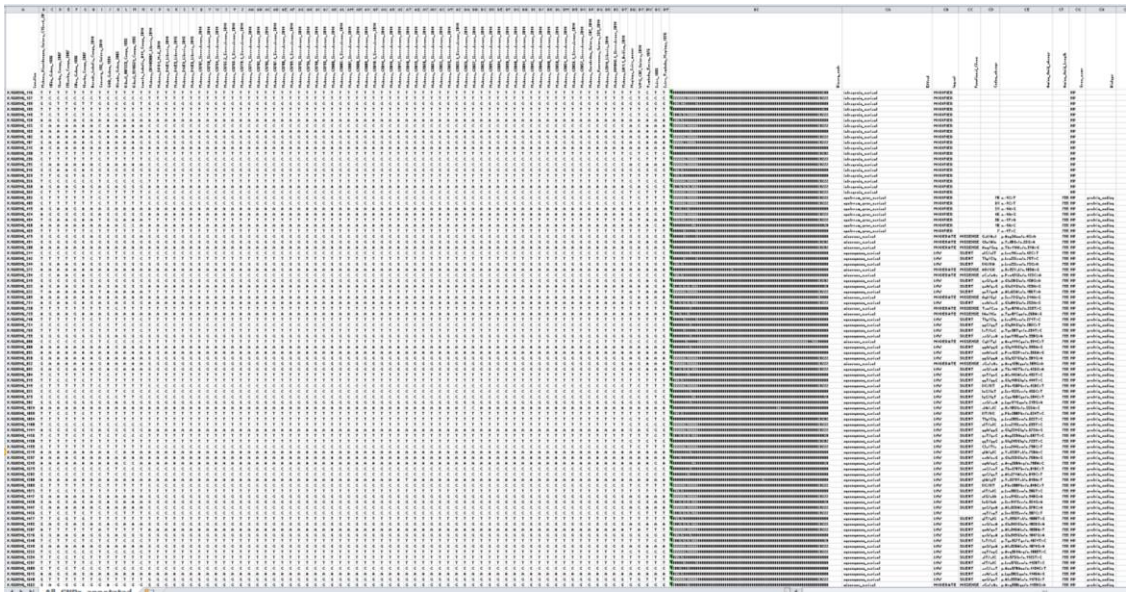
3.2.2 Annotation of individual SNP calls. Following identification of SNPs and indels, the next question usually asked is: what is the functional consequence of this SNP or indel i.e. does it result in an amino acid change, a frameshift mutation, or is it synonymous ('silent')? SPANDx performs annotation of all identified variants to determine their putative effect on protein and transcriptional function. There are several annotation outputs provided by SPANDx, including annotation on an individual genome level. Looking at all annotated variants on a per-strain basis can quickly become a cumbersome task, so we'll also take a look at annotated SNP and indel matrices generated by SPANDx across the entire Ebolavirus dataset.

a. Using WinSCP or Fugu, navigate to the annotated genome directory for `Zaire_1995` (`/analysis/Ebola/Zaire_1995/unique/annotation/`).

b. Copy across to your laptop the `snpEff_summary_SNPs.html` annotation summary file. This file can be opened with your favourite internet browser. See Class exercise #3.2.2a below for a screenshot of this file.

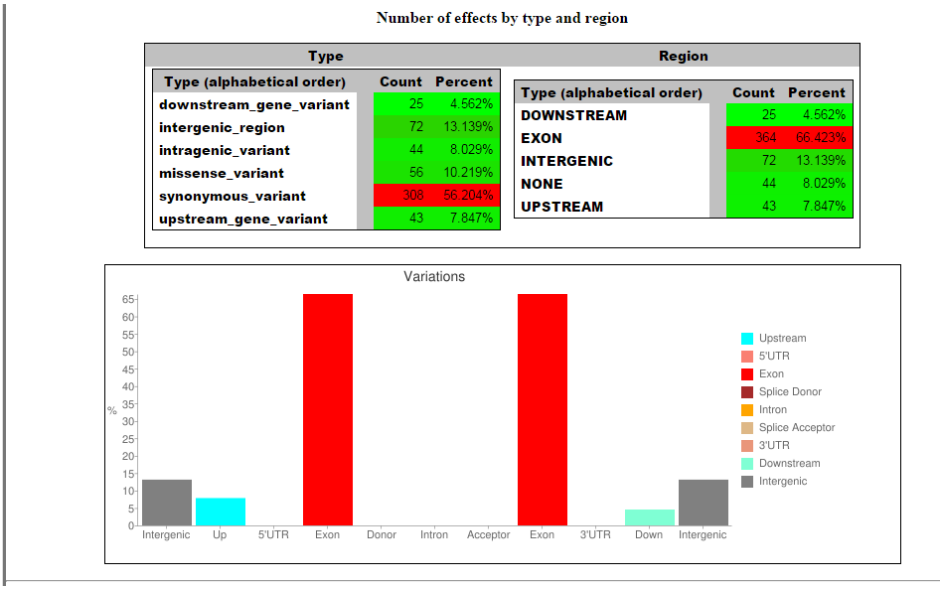
c. Navigate to: `~/analysis/Ebola/Outputs/Comparative`, and copy across the `All_SNPs_annotated.txt` and `All_indels_annotated.txt` files to your laptop. The best way to view these files is to import them into Excel as follows:

- Go to File > Open > navigate to your `.txt` file, then click 'Open'.
- A "Text Import Wizard" dialogue box will appear. Keep as 'Delimited' and click on 'Next' to go to Step 2.
- Keep as 'Tab' for the Delimiters, and click on 'Next' to go to Step 3.
- Scroll across to the 'Binary_code' column and change the 'Column data format' from 'General' to 'Text'.
- Click 'Finish'. A zoomed-out view of the `All_SNPs_annotated.txt` file is shown below:



CLASS EXERCISES TO DO DURING WORKSHOP

Class exercise #3.2.2a: From the `snpEff_summary_SNPs.html` annotation summary file for `Zaire_1995`, what percentage of SNPs are silent? What about nonsynonymous SNPs? How many SNPs are in coding regions vs. other regions? Do these numbers make biological sense and what might they be (broadly) telling you about Ebolavirus evolution?



Number of effects by type and region

Type

Type (alphabetical order)	Count	Percent
downstream_gene_variant	25	4.562%
intergenic_region	72	13.139%
intragenic_variant	44	8.029%
missense_variant	56	10.219%
synonymous_variant	308	56.204%
upstream_gene_variant	43	7.847%

Region

Type (alphabetical order)	Count	Percent
DOWNSTREAM	25	4.562%
EXON	364	66.423%
INTERGENIC	72	13.139%
NONE	44	8.029%
UPSTREAM	43	7.847%

Class exercise #3.2.2b: From the `All_SNPs_annotated.txt` and `All_indels_annotated.txt` files, what proportion of variants are synonymous vs. non-synonymous? Why do you think some of the annotation fields are missing? For what purpose/s might you use the 'binary_code' column?

TOPIC 3.3: How to perform phylogenetic analysis of the Ebola outbreak genomes

Phylogenetic reconstruction is a very common downstream application of NGS data analysis. A whole genome SNP phylogeny gives you an excellent overview of the differences between individual strains in your dataset with a single tree, not to mention it makes for nice publication figures! In this workshop, we'll show you how to reconstruct phylogenetic trees from the Ebolavirus genomes using two SPANDx-generated SNP matrices.

There are three types of phylogenetic methods commonly used to examine genomic data: maximum parsimony (MP), maximum likelihood (ML) and Bayesian inference. Phylogenetics is a HUGE field and would require its own course, so we'll just keep it simple today. As a general rule, it's always worth running MP and ML methods on your datasets, especially if you're viewing them for the first time. In our experience, both methods will give you similar outputs; however, there are some differences, mainly branch lengths. It is important to point out that none of these methods are the best at population reconstruction in all situations. Below is more info on MP and ML.

According to Wikipedia:

"*…**maximum parsimony (MP)** is an optimality criterion under which the phylogenetic tree that minimizes the total number of character-state changes is preferred. Under the MP criterion, the optimal tree will minimize the amount of homoplasy (i.e., convergent evolution, parallel evolution, and evolutionary reversals). In other words, under this criterion, the shortest possible tree that explains the data is considered best. The principle is akin to Occam's razor, which states that—all else being equal—the simplest hypothesis that explains the data should be selected*."

"*…**maximum-likelihood (ML)** estimation is a method of estimating the parameters of a statistical model given data. In practice, ML tends to favor trees that are very similar to the most parsimonious tree(s) for the same dataset; however, ML allows for complex modelling of evolutionary processes, is statistically consistent and is not susceptible to long-branch attraction. Note, however, that the performance of likelihood is dependent on the quality of the model of evolution employed; an incorrect model can produce a biased result. In addition, ML is still quite slow relative to parsimony methods, sometimes requiring weeks to run large datasets. Most of these methods have particularly avid proponents and detractors…."*

- A tabulated summary of some of the pros and cons of MP and ML is summarised below:

MP	ML
Fast	Slow – some complex datasets won't complete within your lifetime!
Weighs all nucleotide changes equally; doesn't use models to factor in less frequent events like transversions	Uses models to weigh the impact of nucleotide changes when determining branch lengths e.g. transitions vs. transversions
Provides no. of SNPs along branches	Does not provide no. of SNPs along branches
Susceptible to long-branch attraction	Not susceptible to long-branch attraction
No model used; only consideration is size of dataset (i.e. need to use heuristic search on larger datasets)	Choice of model VERY important and incorrect model can dramatically impact accuracy of tree
Easy to ascertain proportion of homoplastic SNPs	Difficult (impossible?) to ascertain homoplastic SNPs
Calculates shortest possible tree; may underestimate complexity of evolution, particularly homoplastic events	Does not calculate best tree based on shortest distance; theoretically better at factoring in complex evolution

First, we will reconstruct an MP tree using the program PAUP (Phylogenetic Analysis Using Parsimony).

> Note: The version of PAUP currently installed on Cheetah has known bugs that will not affect the Ebola genome analysis; however, it should not be used on more complex datasets, particularly those planned for publication. We eagerly await an updated, debugged version of PAUP to be released by the developer.

a. Follow steps a-c from Exercise 1 if you are not already logged onto Cheetah and in an interactive qsub session.
b. `cd` to your `analysis/Ebola/Outputs/Comparative` directory.
c. Have a look at the top of your PAUP-compatible SNP matrix by typing: `head Ortho_SNP_matrix.nex`. Here you will see a summary of the number of phylogenetic 'characters' (genetic variants) and taxa you are about to analyse. SPANDx generates this file automatically; if you want to edit to e.g. change taxon names, you can open this file in a text editor (e.g. NotePad++ or TextWrangler), then paste the changes back into the `.nex` file.
d. Execute PAUP by typing: `/home/dsarovich/bin/paup4a146_centos64`
e. Load your SNP matrix file into PAUP by typing: `execute Ortho_SNP_matrix.nex`
f. To perform a heuristic MP search, type: `hsearch`
g. Once this search is complete (should take <10sec), you'll want to determine the consistency index (CI) of your tree - the CI is a measure of the amount of homoplasy in your tree and is used to get a feel for the accuracy of the topology: `describetrees`. Record the CI value.
h. To save your tree file, type: `savetrees brlens=yes file=Ebola_MP_tree.tre`
i. Exit PAUP by typing: `quit`
j. Copy your tree file across to your laptop and load the `.tre` file into FigTree v1.4.0.

Next, we'll reconstruct an ML tree using ExaML (Exascale Maximum Likelihood):

a. Exit interactive qsub by typing: `exit`
b. `cd` to your `analysis/Ebola/Outputs/Comparative` directory.
c. Run the following wrapper script: `/home/dsarovich/wrapper_scripts/ExaML-AVX.sh`
d. Rename the `ExaML_result.T1` file by typing: `mv ExaML_result.T1 Ebola_ML_tree.tre`, and then copy this file across to your laptop. Load the `.tre` file into FigTree v1.4.0.

CLASS EXERCISES TO DO DURING WORKSHOP

Class exercise #5a: Compare the Ebola MP and ML trees that you have generated. What are the similarities and differences? Which tree is "best"? What does the CI value tell you about homoplasy in your MP tree? How does changing the rooting method change the interpretation of your trees?

Class exercise #5b: Compare your trees with a recently published, comprehensive tree from all seven Ebola outbreaks (Gire *et al*., 2014: http://www.sciencemag.org/content/345/6202/1369.long). Due to copyright permissions, we are not able to provide a copy of their trees in this tutorial, but you can find the figure link here: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4431643/figure/F2/). Did they use MP or ML to construct their trees? How can you tell? Why do you think they chose one type of tree over the other?

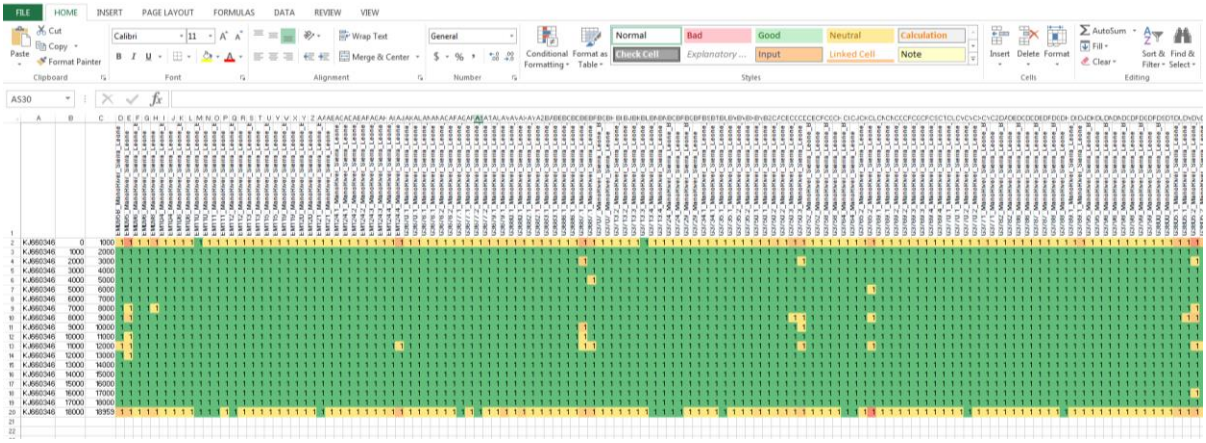
EXERCISE TO DO AFTER WORKSHOP

Although we've not used identical strains to those reported in the Gire *et al.* study, you might be interested in comparing the number of SNPs in their MP tree to the MP tree you constructed. Because we've also included strains from all Ebola outbreaks, your SNP and indel numbers should be very similar to theirs. You may also want to compare indels and non-synonymous SNP variants to learn more about how the virus has changed over time.

TOPIC 3.4: How to perform presence/absence analysis of the Ebola outbreak genomes

In this final tutorial session, we will briefly cover how to look at locus presence-absence (P-A) using comparative genomic data. P-A analysis in Ebola (and pretty much all other viruses) is quite boring, as you will soon discover for yourself. However, this tool can be very useful when analysing more complex organisms such as bacteria, which often do not carry the same complement of genetic material amongst strains. For instance, P-A matrices can be used to identify genes specific to a certain clade or even a particular species. P-A matrices can also be used in combination with statistical testing in the program PLINK to perform microbial genome-wide association studies (mGWAS), which enable you to identify genes or genetic variants significantly associated with e.g. virulence (Note: mGWAS will not be covered in this workshop).

a. Navigate to your `analysis/Ebola/Outputs/Comparative` directory in WinSCP or Fugu.
a. Copy across the `Bedcov_merge.txt` file to your laptop.
b. Open this text file in MS Excel (can also be opened and copied in NotePad/NotePad++/TextWrangler first and then pasted into Excel without loss in formatting). This file is best viewed as a heatmap – to do so in Excel, select those cells containing values between 0 and 1 (usually D2 onwards), then select 'Conditional Formatting', 'Color Scales', 'Green – Yellow – Red Color Scale'. The heatmap should look similar to that shown below:



CLASS EXERCISES TO DO DURING WORKSHOP

Class exercise #3.4a: Examine the P-A matrix for the Ebola genomes. What does it mean? What are the strengths and shortcomings of this approach? What could you use this matrix for?

Class exercise #3.4b (optional): Erin and Derek can provide more complex P-A matrices from bacterial analyses for anyone who is interested.

Class exercise #3.4c: Now that you've come to the end of this tutorial, can you think of any circumstances where you wouldn't be able to identify a SNP, indel or P-A variant using the SPANDx reference-based alignment method we've done today? What strategies could you employ to get around this issue?

HINT 1: SPANDx performs short-read alignment using the program Burrows-Wheeler Aligner, which does not map NGS reads to paralogous regions by default.

HINT 2: How would the choice and even quality of your reference genome affect what SNPs/indels/P-A loci are identified in your NGS reads?

HINT 3: How would a single poor-quality `.fastq.gz` pair affect your overall SNP/indel matrix outputs (keeping in mind that SPANDx has default filters in place to remove regions with poor mapping quality)?