RedDog Tutorial

Authors: D. J. Edwards, B. J. Pope and K. E. Holt   Date: Mar. 25th, 2015

1. Tutorial Background

a) 2011 Outbreak of *E. coli* O104:H4

During the European spring and summer of 2011, an unusual outbreak of *Escherichia coli* serovar O104:H4 infections occurred, centered mainly on northern Germany. [1,2] The causative agent was found to be a particularly aggressive *E. coli* pathogen that caused bloody diarrhea in most of 75% of patients, and haemoretic fever in 25% of the patients, which lead to mortality in some cases. [3] Another distinction of the outbreak was that it was also one of the first chances to apply whole genome sequencing using high-throughput sequence platforms on a bacterial outbreak of unknown origin. [1,2] Different research groups have examined the whole genome of the causative agent in a number of isolates from both during and prior to the outbreak. [1-9] Here we will take the entire O104:H4 isolate collection available from the PATRIC website, create simulated reads, and then map them using RedDog to a suitable reference, to create the core SNP phylogeny for the O104:H4. Simulated reads will be used rather than making the user download the much larger short read files for each isolate (where available).

We will also use coverage information from reads mapped to the plasmids of the outbreak strain to create a heat-map, and combine this with the phylogeny.

b) Tutorial Assumptions and Limits

This tutorial assumes the user has access to a cluster environment, though some parts of the tutorial are carried out on software directly installed on a laptop. The tutorial also assumes the user has some familiarity with running programs on a cluster environment, and some familiarity using the other programs on a local machine, particularly the R statistics package. It is also assumed that the RedDog pipeline and associated dependencies have all been installed on the cluster system prior to the tutorial. There are also some programs that need to be installed on a local computer; these are detailed in the text and a summary list is provided towards the end of this tutorial.

Whilst this tutorial does cover the generation and filtering of core SNPs, and producing a phylogeny from these SNPs, the phylogenetic section is only a beginning of any further analysis of the data. For instance, only a quick maximum-likelihood tree will be produced (using FastTree), rather than a more exhaustive tree evaluation. For those who may not have worked with phylogenetic data, there is an excellent primer for beginners available. [10]

Finally, this tutorial is meant to be descriptive and certainly not prescriptive - how you analysis any other data set will depend on the outcomes required. For instance, the detection and removal of any bacteriophage is used in this tutorial - if your focus of attention should include the phage, you would most probably not want to remove them via filtering.

2. Inputs

First, log into your cluster account and create a folder for this tutorial in a suitable location.

```
mkdir RedDog_Tutorial
```

Take note of the full path to this folder (this will be referred to as `/full_path_to_target_folder/RedDog_Tutorial/` below).

a) Reference

The best reference to use for mapping reads is the closest related isolate available, preferably within the group being investigated. In this case there is an outbreak isolate with the whole genome available, str. 2011C-3493. To visual the SNPs later in the tutorial, we will use the NCBI version of this reference genome, along with the three plasmids that were found in this isolate.

Go to http://www.ncbi.nlm.nih.gov/assembly/ and enter ASM29945v1 into the search box and hit the search button:

NCBI Resources ⊙ How To ⊙			Sign in to NCBI
Assembly	Assembly ⧩	ASM29945v1	⊗ **Search**
	Advanced Browse by organism		Help

Scroll down to the bottom of the results page for ASM29945v1:

Assembly Definition	Assembly Statistics		

Global assembly definition   Download the full sequence report

Assembly Unit: Primary Assembly (GCF_000299465.1)

Molecule name	GenBank ID		RefSeq ID
Chromosome	CP003289.1	=	NC_018658.1
Plasmid pAA-EA11	CP003291.1	=	NC_018666.1
Plasmid pESBL-EA11	CP003290.1	=	NC_018659.1
Plasmid pG-EA11	CP003292.1	=	NC_018660.1

We need the GenBank references for the Chromosome (CP003298) and the three plasmids (CP003290-2), and the FASTA version of the Chromosome.

Click on the link to CP003289.1 and the GenBank file will be displayed. However, the sequence will not be displayed until the custom view is changed; in the `Customize view` box, click on the `Display options: Show sequence` box, then the `Update View` button:

Customize view ▲

Basic Features
○ Default features
◉ Gene, RNA, and CDS features only

Display options
☑ Show sequence
☐ Show reverse complement

[Update View]

Once the entire sequence has been downloaded (to screen), click on the down arrow next to `Send`. Click on `Choose Destination: File` then hit the `Create File` button.

Send: ⊙

◉ Complete Record
○ Coding Sequences
○ Gene Features

Choose Destination
◉ File   ○ Clipboard
○ Collections   ○ Analysis Tool

Download 1 items.

Format
[GenBank ⧩]

[Create File]

This will open the following dialogue box; hit `Save File`.

Once the file has finished downloading, rename it from `sequence.gb` to `CP003289.gbk`. [Note: RedDog does not care if the file extension is `.gbk` or `.gb`; the test for a GenBank reference within RedDog is not predicated on this extension.]

You can then upload it to you cluster account using the `scp` command.

```
scp CP003289.gbk
<username>@<cluster_machine_path>:/full_path_to_target_folder/RedDog_
Tutorial/
```

You will be asked for you password after you hit 'enter'.

Once the file has finished uploading, we also want to get the FASTA version of the Chromosome. Click on `FASTA` to change to the FASTA view of the genome:

Once the sequence has downloaded to screen, again click on the down arrow next to `Send`. Click on `Choose Destination: File` then hit the `Create File` button:
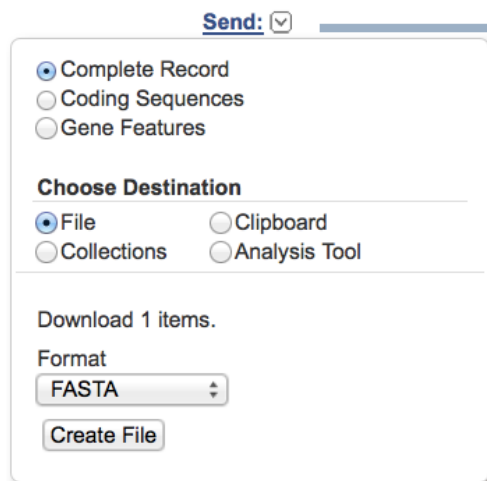
This will open the following dialogue box; hit `Save File`:
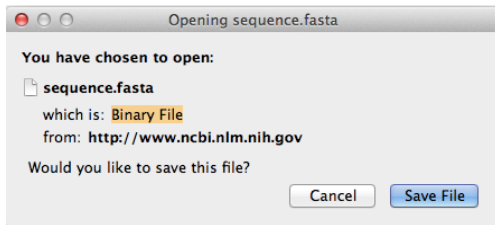
Once the file has finished downloading, rename it from `sequence.fasta` to `CP003289.fasta`. You can then upload it to you cluster account using the `scp` command.

We won't be using this FASTA version for mapping, however it will be used to locate duplicate sequences during SNP filtering later on, and to visualize the SNPs against the reference genome.

Now repeat the above to obtain the three plasmids in Genbank format (we don't need these in FASTA format for this tutorial). Remember to rename the Genbank files before the `scp` step (`CP003290.gbk`, `CP003291.gbk`, and `CP003292.gbk` respectively).

Once you have uploaded all four GenBank files (and the FASTA file for the chromosome) to your cluster account, we can join the four Genbank files to create a multiple sequence GenBank file. This will be the reference used by RedDog to map the simulated sequence reads. First, change to the RedDog_Tutorial folder you set up earlier:

```
cd /full_path_to_target_folder/RedDog_Tutorial/
```

Then we can concatenate the four GenBank files together to get the multiple sequence GenBank file for mapping:

```
cat CP003289.gbk CP003290.gbk CP003291.gbk CP003292.gbk > 2011C-
3493_full.gbk
```

b) Sequences
[NOTE: currently, the following method of downloading sequences from the PATRIC website is not working. Instead, you can obtain the full set of FASTA sequences (genomes or contigs) from RedDog_Extras. (The file can be downloaded without signing-in/up for a Dropbox account) Unlike the PATRIC downloads, these do not need to be renamed. However, the file does need to be unzipped before use.]

Go to the PATRIC genomes website.
http://wwww.patricbrc.org/portal/portal/patric/Genomes

Scroll down the page to `Download Data`

i. O104:H4 isolates

In the PATRIC Genomes Download search box, type in `O104:H4`. Then select all the available isolates.

Make sure that the `Annotation Source` is set to `PATRIC` and `File Type` is set to `Genomic Sequences in Fasta` as shown in the screen snapshot. Then hit `Download`. This will open the following download message.
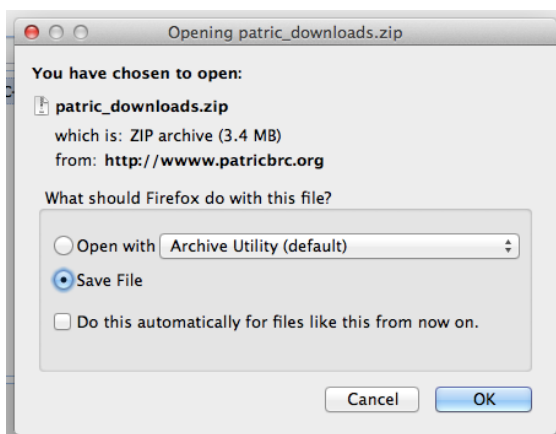
Hit `OK` to save the file (~89.5Mb).

Once the file has finished downloading, rename it from `patric_downloads.zip` to `O104_patric.zip`. You can then upload it to you cluster account using the `scp` command. Once the file has finished uploading, unzip the sequences: Change directory to the folder with the file and use the `unzip` command:

```
unzip O104_patric.zip
```

Unfortunately, this process does miss one isolate of O104:H4, namely str. 55989. To download this sequence, type `55989` into the PATRIC Genomes search box, then select the single isolate that results. Again, check that the `Annotation Source` and `File Type` are set correctly (as for O104:H4 above), then hit download (~1.5 Mb).

Once it has finished downloading, rename this file from `patric_downloads.zip` to `55989_patric.zip`, and `scp` the resulting file to the RedDog_Tutorial folder. Once the file has finished uploading, unzip the sequences: change directory to the folder with the uploaded file and use the `unzip` command.

ii. Outgroup (str. E1167)

To create the phylogeny, we should include an outgroup isolate. For this exercise, the closest available *E. coli* isolate will be used, *E. coli* str. E1167. (To find this isolate and its relationship to the O104:H4, use the 'phylogeny' feature on PATRIC – see the PATRIC website for details.)

To download this sequence, type `E1167` into the PATRIC Genomes Download search box, then select the single isolate that results. Again, check that the `Annotation Source` and `File Type` are set correctly (as for O104:H4 above), then hit download (~1.5 Mb).

Once it has finished downloading, rename this file from `patric_downloads.zip` to `E1167_patric.zip`, and `scp` the resulting file to the RedDog_Tutorial folder. Once the file has finished uploading, unzip the sequences: change directory to the folder with the uploaded file and use the `unzip` command.

The resulting files have rather cumbersome names so we will rename them. Enter the following command:

```
rename Escherichia_coli_O104-H4_str_ '' Escherichia_coli_O104-
H4_str_*.fna
```

Followed by:

```
rename Escherichia_coli_ '' Escherichia_coli_*.fna
```

iii. Shredding

Now that all the O104:H4 and outgroup sequences in FASTA format have been downloaded, unzipped, and renamed, we need to convert them into simulated high-throughput sequences in FASTQ format so they can be mapped against the reference. This will be accomplished by utilising the `wgsim` command in the SAMTools package (v1+).

For each isolate in FASTA format, run the following commands
(Note: this part should <u>definitely not be done on the head node</u>; either run them as separate batch jobs [one for each isolate], or initiate an interactive session)

```
module load samtools-intel/1.1

wgsim -e 0 -1 100 -2 100 -r 0 -R 0 -X 0 -N 1000000 <isolate>.fna
<isolate>_1.fastq <isolate>_2.fastq

gzip <isolate>_1.fastq

gzip <isolate>_2.fastq
```

Warning: this shredding of reads may take a while to complete, and is the longest part of this tutorial (time wise).

When finished, there should be a _1.fastq.gz and _2.fastq.gz file for each of the 57 O104:H4 isolates and the one outgroup. The pairs of files should also be about the same size (48-49 Mb each).

3. Cleanup after Creating Inputs

Once you finish preparing the input files, the contents of the RedDog_Tutorial folder should look like this:

01-09591_1.fastq.gz	11-4522_2.fastq.gz	CP003291.gbk	Ec11-9941_2.fastq.gz
01-09591_2.fastq.gz	11-4522.fna	CP003292.gbk	Ec11-9941.fna
01-09591.fna	11-4623_1.fastq.gz	E11210_1.fastq.gz	Ec11-9990_1.fastq.gz
04-8351_1.fastq.gz	11-4623_2.fastq.gz	E11210_2.fastq.gz	Ec11-9990_2.fastq.gz
04-8351_2.fastq.gz	11-4623.fna	E11210.fna	Ec11-9990.fna
04-8351.fna	11-4632_C1_1.fastq.gz	E1167_1.fastq.gz	Ec12-0465_1.fastq.gz
09-7901_1.fastq.gz	11-4632_C1_2.fastq.gz	E1167_2.fastq.gz	Ec12-0465_2.fastq.gz
09-7901_2.fastq.gz	11-4632_C1.fna	E1167.fna	Ec12-0465.fna
09-7901.fna	11-4632_C2_1.fastq.gz	E1167_patric.zip	Ec12-0466_1.fastq.gz
11-02030_1.fastq.gz	11-4632_C2_2.fastq.gz	E9211_1.fastq.gz	Ec12-0466_2.fastq.gz
11-02030_2.fastq.gz	11-4632_C2.fna	E9211_2.fastq.gz	Ec12-0466.fna
11-02030.fna	11-4632_C3_1.fastq.gz	E9211.fna	GOS1_1.fastq.gz
11-02033-1_1.fastq.gz	11-4632_C3_2.fastq.gz	Ec11-4984_1.fastq.gz	GOS1_2.fastq.gz
11-02033-1_2.fastq.gz	11-4632_C3.fna	Ec11-4984_2.fastq.gz	GOS1.fna
11-02033-1.fna	11-4632_C4_1.fastq.gz	Ec11-4984.fna	GOS2_1.fastq.gz
11-02092_1.fastq.gz	11-4632_C4_2.fastq.gz	Ec11-4986_1.fastq.gz	GOS2_2.fastq.gz
11-02092_2.fastq.gz	11-4632_C4.fna	Ec11-4986_2.fastq.gz	GOS2.fna
11-02092.fna	11-4632_C5_1.fastq.gz	Ec11-4986.fna	H112180280_1.fastq.gz
11-02093_1.fastq.gz	11-4632_C5_2.fastq.gz	Ec11-4987_1.fastq.gz	H112180280_2.fastq.gz
11-02093_2.fastq.gz	11-4632_C5.fna	Ec11-4987_2.fastq.gz	H112180280.fna
11-02093.fna	2009EL-2050_1.fastq.gz	Ec11-4987.fna	H112180282_1.fastq.gz
11-02281_1.fastq.gz	2009EL-2050_2.fastq.gz	Ec11-4988_1.fastq.gz	H112180282_2.fastq.gz
11-02281_2.fastq.gz	2009EL-2050.fna	Ec11-4988_2.fastq.gz	H112180282.fna
11-02281.fna	2009EL-2071_1.fastq.gz	Ec11-4988.fna	H112180283_1.fastq.gz
11-02318_1.fastq.gz	2009EL-2071_2.fastq.gz	Ec11-5536_1.fastq.gz	H112180283_2.fastq.gz
11-02318_2.fastq.gz	2009EL-2071.fna	Ec11-5536_2.fastq.gz	H112180283.fna

```
11-02318.fna           2011C-3493_1.fastq.gz    Ec11-5536.fna          H112180540_1.fastq.gz
11-02913_1.fastq.gz    2011C-3493_2.fastq.gz    Ec11-5537_1.fastq.gz   H112180540_2.fastq.gz
11-02913_2.fastq.gz    2011C-3493.fna           Ec11-5537_2.fastq.gz   H112180540.fna
11-02913.fna           2011C-3493_full.gbk      Ec11-5537.fna          H112180541_1.fastq.gz
11-03439_1.fastq.gz    55989_1.fastq.gz         Ec11-5538_1.fastq.gz   H112180541_2.fastq.gz
11-03439_2.fastq.gz    55989_2.fastq.gz         Ec11-5538_2.fastq.gz   H112180541.fna
11-03439.fna           55989.fna                Ec11-5538.fna          LB226692_1.fastq.gz
11-03943_1.fastq.gz    55989_patric.zip         Ec11-5603_1.fastq.gz   LB226692_2.fastq.gz
11-03943_2.fastq.gz    C227-11_1.fastq.gz       Ec11-5603_2.fastq.gz   LB226692.fna
11-03943.fna           C227-11_2.fastq.gz       Ec11-5603.fna          O104_patric.zip
11-04080_1.fastq.gz    C227-11_Broad_1.fastq.gz Ec11-5604_1.fastq.gz   ON2010_1.fastq.gz
11-04080_2.fastq.gz    C227-11_Broad_2.fastq.gz Ec11-5604_2.fastq.gz   ON2010_2.fastq.gz
11-04080.fna           C227-11_Broad.fna        Ec11-5604.fna          ON2010.fna
11-3677_1.fastq.gz     C227-11.fna              Ec11-6006_1.fastq.gz   ON2011_1.fastq.gz
11-3677_2.fastq.gz     C236-11_1.fastq.gz       Ec11-6006_2.fastq.gz   ON2011_2.fastq.gz
11-3677.fna            C236-11_2.fastq.gz       Ec11-6006.fna          ON2011.fna
11-4404_1.fastq.gz     C236-11.fna              Ec11-9450_1.fastq.gz   TY-2482_1.fastq.gz
11-4404_2.fastq.gz     CP003289.fasta           Ec11-9450_2.fastq.gz   TY-2482_2.fastq.gz
11-4404.fna            CP003289.gbk             Ec11-9450.fna          TY-2482.fna
11-4522_1.fastq.gz     CP003290.gbk             Ec11-9941_1.fastq.gz
```

There are a number of files we no longer need (specifically the `*.zip` and `*.fna` files) so we will remove them before moving onto the pipeline. *Note: you can save the zip files if you are likely to need these sequences again.*

```
rm *.zip
rm *.fna
```

The RedDog_Tutorial folder should now look like this with two read files for each isolate and the GenBank reference for mapping - in bold typeface, plus the five other reference files we downloaded:

```
01-09591_1.fastq.gz    11-4522_2.fastq.gz       CP003291.gbk           Ec11-9941_2.fastq.gz
01-09591_2.fastq.gz    11-4623_1.fastq.gz       CP003292.gbk           Ec11-9990_1.fastq.gz
04-8351_1.fastq.gz     11-4623_2.fastq.gz       E11210_1.fastq.gz      Ec11-9990_2.fastq.gz
04-8351_2.fastq.gz     11-4632_C1_1.fastq.gz    E11210_2.fastq.gz      Ec12-0465_1.fastq.gz
09-7901_1.fastq.gz     11-4632_C1_2.fastq.gz    E1167_1.fastq.gz       Ec12-0465_2.fastq.gz
09-7901_2.fastq.gz     11-4632_C2_1.fastq.gz    E1167_2.fastq.gz       Ec12-0466_1.fastq.gz
11-02030_1.fastq.gz    11-4632_C2_2.fastq.gz    E9211_1.fastq.gz       Ec12-0466_2.fastq.gz
11-02030_2.fastq.gz    11-4632_C3_1.fastq.gz    E9211_2.fastq.gz       GOS1_1.fastq.gz
11-02033-1_1.fastq.gz  11-4632_C3_2.fastq.gz    Ec11-4984_1.fastq.gz   GOS1_2.fastq.gz
11-02033-1_2.fastq.gz  11-4632_C4_1.fastq.gz    Ec11-4984_2.fastq.gz   GOS2_1.fastq.gz
11-02092_1.fastq.gz    11-4632_C4_2.fastq.gz    Ec11-4986_1.fastq.gz   GOS2_2.fastq.gz
11-02092_2.fastq.gz    11-4632_C5_1.fastq.gz    Ec11-4986_2.fastq.gz   H112180280_1.fastq.gz
11-02093_1.fastq.gz    11-4632_C5_2.fastq.gz    Ec11-4987_1.fastq.gz   H112180280_2.fastq.gz
11-02093_2.fastq.gz    2009EL-2050_1.fastq.gz   Ec11-4987_2.fastq.gz   H112180282_1.fastq.gz
11-02281_1.fastq.gz    2009EL-2050_2.fastq.gz   Ec11-4988_1.fastq.gz   H112180282_2.fastq.gz
11-02281_2.fastq.gz    2009EL-2071_1.fastq.gz   Ec11-4988_2.fastq.gz   H112180283_1.fastq.gz
11-02318_1.fastq.gz    2009EL-2071_2.fastq.gz   Ec11-5536_1.fastq.gz   H112180283_2.fastq.gz
11-02318_2.fastq.gz    2011C-3493_1.fastq.gz    Ec11-5536_2.fastq.gz   H112180540_1.fastq.gz
11-02913_1.fastq.gz    2011C-3493_2.fastq.gz    Ec11-5537_1.fastq.gz   H112180540_2.fastq.gz
11-02913_2.fastq.gz    2011C-3493_full.gbk      Ec11-5537_2.fastq.gz   H112180541_1.fastq.gz
11-03439_1.fastq.gz    55989_1.fastq.gz         Ec11-5538_1.fastq.gz   H112180541_2.fastq.gz
11-03439_2.fastq.gz    55989_2.fastq.gz         Ec11-5538_2.fastq.gz   LB226692_1.fastq.gz
11-03943_1.fastq.gz    C227-11_1.fastq.gz       Ec11-5603_1.fastq.gz   LB226692_2.fastq.gz
11-03943_2.fastq.gz    C227-11_2.fastq.gz       Ec11-5603_2.fastq.gz   ON2010_1.fastq.gz
11-04080_1.fastq.gz    C227-11_Broad_1.fastq.gz Ec11-5604_1.fastq.gz   ON2010_2.fastq.gz
11-04080_2.fastq.gz    C227-11_Broad_2.fastq.gz Ec11-5604_2.fastq.gz   ON2011_1.fastq.gz
11-3677_1.fastq.gz     C236-11_1.fastq.gz       Ec11-6006_1.fastq.gz   ON2011_2.fastq.gz
11-3677_2.fastq.gz     C236-11_2.fastq.gz       Ec11-6006_2.fastq.gz   TY-2482_1.fastq.gz
11-4404_1.fastq.gz     CP003289.fasta           Ec11-9450_1.fastq.gz   TY-2482_2.fastq.gz
11-4404_2.fastq.gz     CP003289.gbk             Ec11-9450_2.fastq.gz
11-4522_1.fastq.gz     CP003290.gbk             Ec11-9941_1.fastq.gz
```

Once you have confirmed all these files are present, you are ready to move on to using the RedDog pipeline for mapping.

4. Using Inputs with RedDog

a) Configuration File

First, we will make a copy of the configuration file to use with the pipeline. Change directory to the RedDog folder (not the RedDog_Tutorial folder we set up earlier, but the folder with the pipeline).

```
cd /full_path_to_folder/RedDog

cp RedDog_config.py o104_config.py
```

Then open `o104_config.py` in your favorite text editor. As we want to use the default values for the pipeline, we only need to change three out of the first four input variables: 'reference', 'sequences', and 'output'. Change the configuration file to the values below (remember to exchange `/full_path_to_folder/` for the actual path).

```
'''
Configuration file for RedDog.py V1.0
------------------------------
Essential pipeline variables.
'''

reference = "/full_path_to_folder/RedDog_Tutorial/2011C-3493_full.gbk"

sequences = "/full_path_to_folder/RedDog_Tutorial/*.fastq.gz"

output ="/full_path_to_folder/RedDog_Tutorial/RedDog_Output"

out_merge_target = ""
```

Once you have made the changes, save the file and exit the editor.

b) Running RedDog

If you are not already there, change directory to the RedDog program folder

```
cd /full_path_to_folder/RedDog
```

On a cluster system, you then need to load the Python module (this should be set up to include the Ruffus and Rubra dependencies as found in the RedDog manual). On our system, the command is:

```
module load python-gcc/2.7.5
```

We are now ready to try run the pipeline.

Note: RedDog needs to run on the head node, not via job submission or interactive session.

The first time we launch it, we will do a print run. This 'dummy' run will allow us to check the settings.

```
rubra RedDog --config o104_config --style print

RedDog V1.0 - phylogeny run

Copyright (c) 2015, David Edwards, Bernie Pope, Kat Holt
All rights reserved. (see README.txt for more details)

Mapping: Bowtie2 V2.2.3
Preset Option: --sensitive-local
4 replicon(s) in GenBank reference 2011C-3493_full
4 replicon(s) to be reported
58 sequence pair(s) to be mapped

Output folder:
/full_path_to_folder/RedDog_Tutorial/RedDog_Output/

Start Pipeline? (y/n)
```

Hit y and then the enter key and the pipe will print out to screen all the jobs that it will do when it actually runs. Whilst the entire list is much too long to print out here, you can check you have the right setting by looking at the very beginning of the print run.

```
Starting pipeline...
2005 jobs to be executed in total
```

You should have the same number of total jobs to be executed (total = 2005).

To actually run the pipeline, we just need to substitute the --style run command as follows:

```
rubra RedDog --config o104_config --style run
```

You should get the same message as before, but now when you hit y, RedDog will send jobs to the job queue for execution.

If the queue is not busy, the entire pipeline should take around 20 minutes or so to complete – if you want to know exactly how long it takes, just add time to the start of the rubra RedDog command.

c) Interpreting the Output

There is a large amount of information generated by the RedDog pipeline, to be found across a number of files in the output folder. However, we will limit ourselves to those necessary to obtain the core phylogeny and a heat-map of plasmid content. A more detailed look at the output files is provided at the end of this tutorial (Section 10. Understanding the Output - Examples).

The first output file to examine is `2011C-3493_full_run_report.txt` that gives a summary of the run. Open this file in your favorite text editor and read through the summary. For now we are most interested in this part:

```
Replicon: CP003289
None of the 57 isolates failed
Outgroup:
E1167
```

Replicon CP003289 is the genome of 2011C-3493, and the isolate we added as the outgroup, E1167, has a SNP count more than two standard deviations away from the mean SNP count for all isolates. Hence, the pipeline has also called E1167 as an outgroup.

Download the tree file, `2011C-3493_full_CP003289_alleles_var_cons0.95.tree`, to your local machine and open it with FigTree (http://tree.bio.ed.ac.uk/software/figtree/).
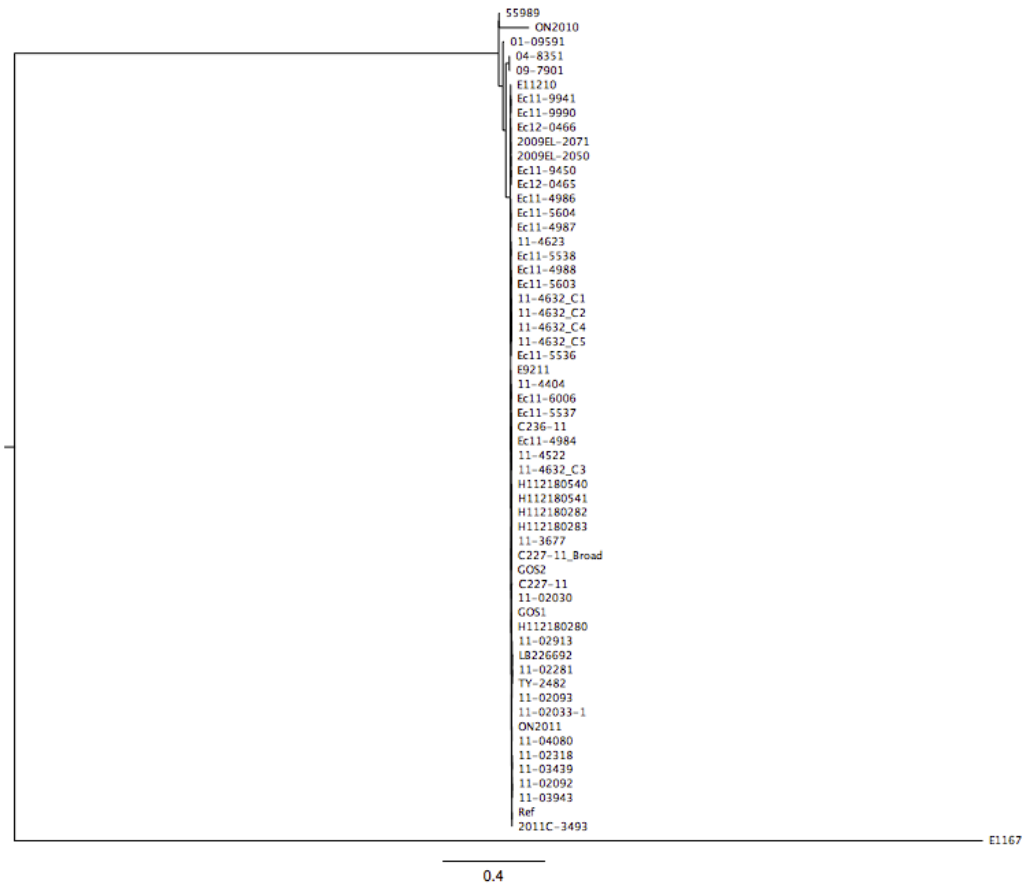
When it opens the following dialogue box, enter 'bootstrap', then hit OK.
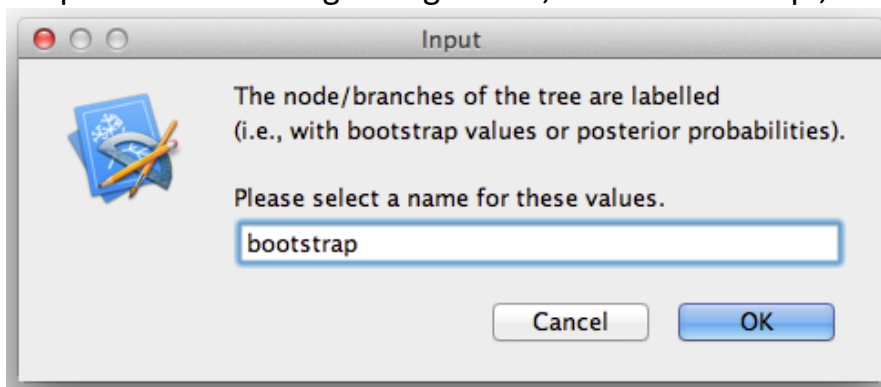
Select the branch leading to E1167, the outgroup, and then click on 'Reroot'.
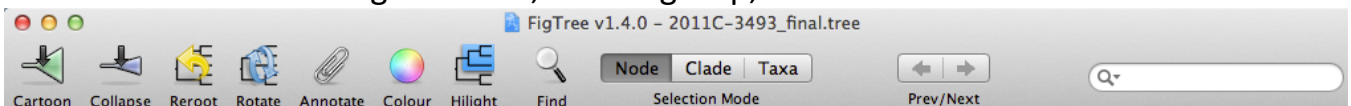
The resulting tree should look like this:

Notice that most of the variation is between the outgroup and the isolates of interest; as a result, it is difficult to observe detail within the O104:H4 clade.

5. Cleanup after RedDog

Once the pipeline has successfully finished, we need to clean out the log files. To quickly remove the folder and all its contents (assuming you are still in the RedDog folder), use:

```
rm –rf log
```

At the end of each run, there is one file, "finish.deleteDir.Success", that does normally need to be removed - especially so if the run will be merged with other read sets later. However, as we are not going to do any merging, the file can be ignored this time.

6. Post-RedDog Analysis

After checking the output from RedDog, the next step is to remove those SNPs that fall in sequences within the genome that can compromise the phylogenetic signal of the core genome. These include repeat sequences and prophage. Then the outgroup needs to be taken into account, as we need to remove any variation found solely in this outgroup to improve the resolution within the branches of interest. Also, any highly recombinant isolates within the data set need to be dealt with.

a) Identifying Reference Repeat Sequences and Prophage

Identifying those sequences within the reference that are direct repeats requires the use of the Mummer package (http://mummer.sourceforge.net/). In particular, we are

going to use the nucmer commands (see
http://mummer.sourceforge.net/manual/#identifyingrepeats for more details).

(Note: this part should <u>definitely not be done on the head node</u>; either run as a batch job or initiate an interactive session)

The reference genome we need to use is `CP003289.fasta`. We can examine the reference genome to identify any large inexact repeats within the genome.

```
cd /full_path_to_target_folder/RedDog_Tutorial/

module load mummer-intel/3.23

nucmer --maxmatch --nosimplify --prefix=CP003289 CP003289.fasta
CP003289.fasta

show-coords -r CP003289.delta > CP003289.coords
```

The identified repeats in `CP003289_CP003289.coords` then have to be converted to coordinates for the `parseSNPtable` script. There is a small script, `filterCoords` in the RedDog folder that will do exactly that.

```
python /full_path_to_folder/RedDog/filterCoords.py -i CP003289.coords
-o CP003289_filtered.coords -I 90
```

This will produce a file with the list of coordinates for all the sequences with 90% match within the genome. The coordinates of any potential prophage can now be added to this coordinates file.

Go to PHAST website (http://phast.wishartlab.com/) and enter `CP003289` into the Genbank accession number (GI) box. Then click on `Submit`.

The results page should look like this:

Escherichia coli O104:H4 str. 2011C-3493, complete genome.

	Summary result file
	Detailed file
	FLASH image of the result (Flash player 11 needed)
	Image in PNG format

Click on `Summary result file`. There are eight sequences within the genome that potentially are prophage. Rather than create a new file for these eight entries, we will add them to the bottom of the coordinates file from the last step.

Open `CP003289_filtered.coords` in your favorite text editor and add the following eight lines (these are straight from the PHAST output - you should check them!):

```
2052129,2071441
2133761,2204067
2349653,2393440
2525541,2592360
2892626,2939977
3254121,3316983
3567415,3625788
```

Then saved and close the coordinates file - it is now ready for use in parseSNPtable. *Note: parseSNPtable can take the coordinates in many different formats; see details on parseSNPTable in the RedDog manual for more information.*

b) Creating the Core Phylogeny Tree

i) parseSNPtable and FastTree
(Note: this part should <u>definitely not be done on the head node</u>; either run as a batch job or initiate an interactive session)
Now we can use the information about the outgroup and the repeat and phage regions within the genome to filter down the SNP table towards the core SNPs. If not already there, change directory to the RedDog_Output folder, then run:

```
python /full_path_to_folder/RedDog/parseSNPtable.py -s 2011C-
3493_full_CP003289_alleles_var.csv -m filter,cons,aln -o E1167 -x
/full_path_to_folder/RedDog_Tutorial/CP003289_filtered.coords -c 0.95
```

The parseSNPtable script will read in the SNP table from RedDog and filter out any SNPs found to be invariable in all but the outgroup; any SNPs that occur in the given repeat or phage regions will then be filtered out, followed by any SNPs with more than 3 isolates

with missing calls (*i.e.* 95% conservation of alleles). The filtered SNP table is then converted into a concatenated alignment of SNPs.

The resulting concatenated alignment of SNPs can then be passed to FastTree to produce the phylogeny:

```
module load fasttree-gcc/2.1.7

FastTree -gtr -gamma -nt 2011C-
3493_full_CP003289_alleles_var_1outgroup_var_regionFiltered_cons0.95.
mfasta > 2011C-
3493_full_CP003289_alleles_var_1outgroup_var_regionFiltered_cons0.95.
tree
```

Download the tree file, `2011C-3493_full_CP003289_alleles_var_1outgroup_var_regionFiltered_cons0.95. tree`, to your local machine and open it with FigTree as before. Again, select the branch leading to E1167, the outgroup, and then click on `Reroot`. The resulting tree should look like this:
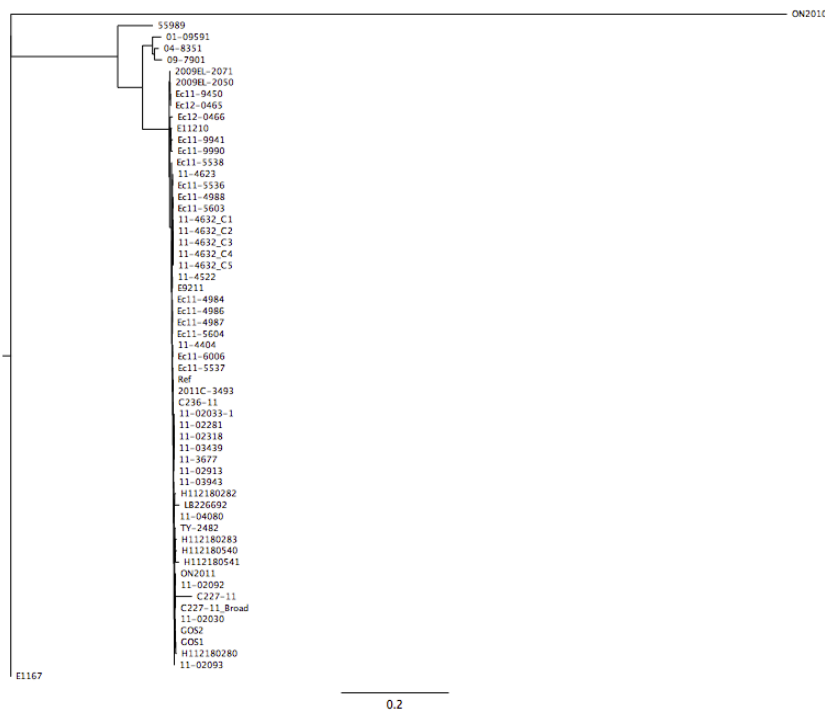
You should be able to clearly see that one of the branches, the one leading to ON2010, is very different to the rest of the O104:H4. A search of the literature will reveal this isolate to have undergone large amounts of recombination with very distantly related *E. coli*. [6] Thus, we need to deal with this organism - we could just remove it from the analysis, and you can if you like, but instead we will change the ON2010 to an outgroup call and reanalyse with FastTree. The phenomenon of long branch switching is discussed in [10].

Run parseSNPtable again, this time with two outgroups, and we will also produce a VCF file to examine the distribution of SNPs across the genome:

```
python /full_path_to_folder/RedDog/parseSNPtable.py -s 2011C-
3493_full_CP003289_alleles_var.csv -m filter,cons,vcf,aln -o
E1167,ON2010 -x /full_path_to_folder/RedDog_Tutorial/
CP003289_filtered.coords -c 0.95 -v CP003289
```

Before using FastTree, we will rename the filtered alignment to a more simplified name:

```
mv 2011C-
3493_full_CP003289_alleles_var_2outgroups_var_regionFiltered_cons0.95
.mfasta 2011C-3493_filtered.mfasta
```

Then open the 2011C-3493_filtered.mfasta file in your favorite text editor and remove the first sequence (with the header >Ref), then save and close it.
Followed by the FastTree command:
```
FastTree -gtr -gamma -nt 2011C-3493_filtered.mfasta > 2011C-
3493_filtered.tree
```

Download the revised tree file to your local machine and open it with FigTree as before. Again, select the branch leading to E1167, the outgroup, and then click on Reroot. At this stage, we would also like to look at the support on the branches (bootstrap values from FastTree). First, in the Layout box, select Align Tip Labels.
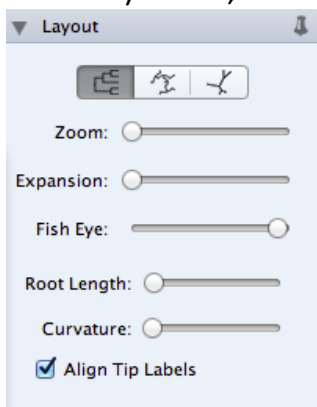
Then click on the Node Labels box, selecting bootstrap values in Display, and change the Sig. Digits to 2 (this will make the output a bit easier to read - you can also increase the Font Size a little).
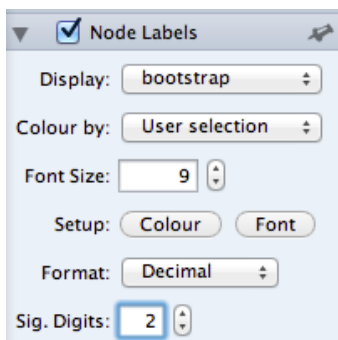
The resulting tree should look like this:
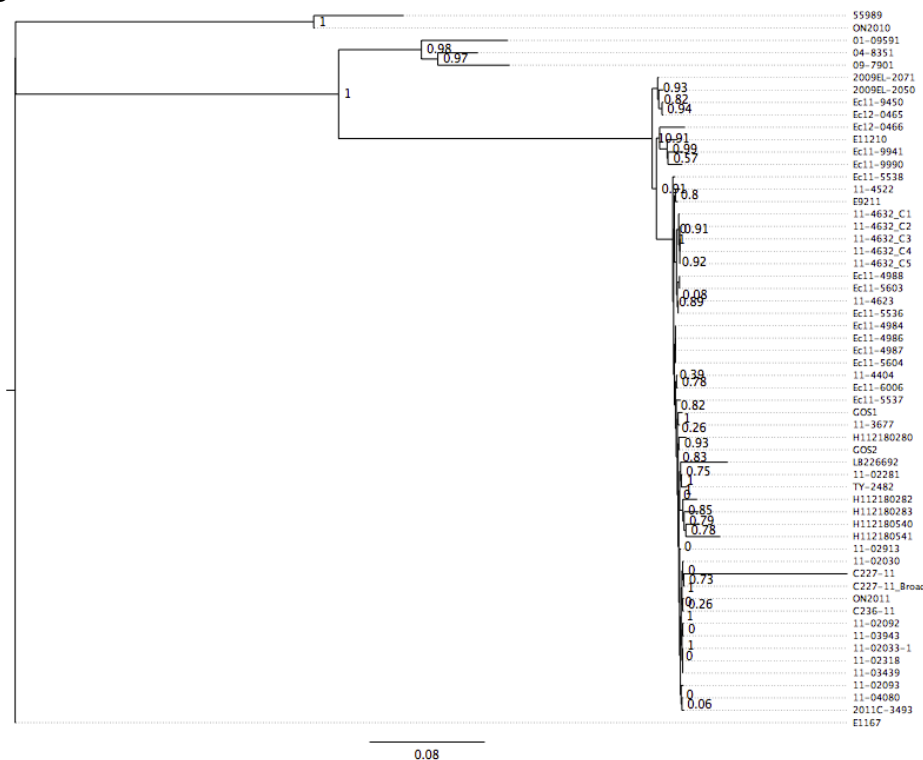
ii) Identifying any recombination

Using the Gingr GUI in the Harvest package (https://github.com/marbl/harvest), we can examine the position of SNPs across the genome. Download the VCF file `2011C-3493_full_CP003289_alleles_var_2outgroups_var_regionFiltered_cons0.95.vcf` to your local machine. When using Gingr, the loading order is important.

Open Gingr and click on `File` and select `Import alignment (VCF & Fasta)…`
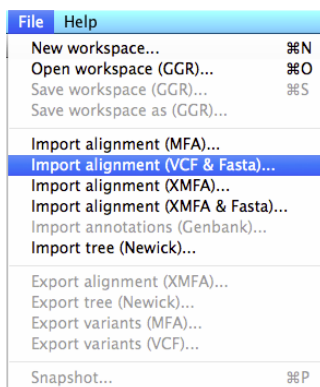
Find the VCF file you just downloaded and load that into Gingr. The program will then ask for the FASTA reference - select `CP003289.fasta` we obtained from NCBI earlier. Then click on `File` again, and this time select `Import annotations (Genbank)…`, this time selecting `CP003289.gbk`. For the third time, select File, and this time `Import tree (Newick)…`, choosing the filtered tree, `2011C-3493_filtered.tree`. Once loaded, right click on the outgroup isolate E1167 and choose `Set as outgroup`. The resulting output should look like this:
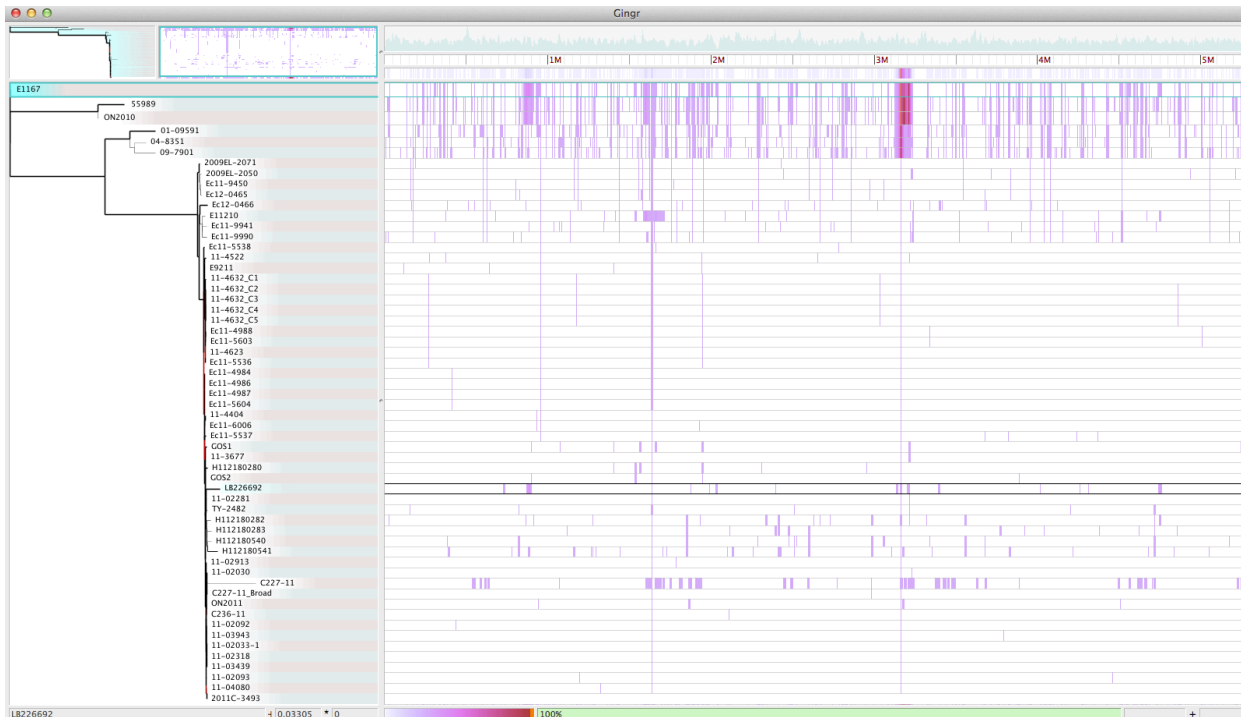
You can clearly see a number of clusters of SNPs along the genome, with a particularly large one around 3.2 million bases. These clusters are most likely due to recombination events that mask the core phylogenetic signal. However, we would like to use a more accurate method to test any clustering of SNPs within the genome.

One method is to examine the distribution of SNPs across 1000 bp windows; if any window has significantly more SNPs than the expected average, this window probably contains a recombinant region. In order to test the distribution of SNPs, we need to use the SNP table `2011C-3493_full_CP003289_alleles_var_2outgroups_var_regionFiltered_cons0.95.csv` to examine the distribution. This examination will be carried out with an R script, `getRecomb`, provided in the RedDog folder. Download both the R scripts (.R suffix) in the RedDog folder to your local machine (we will use the second one later).

Open R (http://www.r-project.org/) on your local machine and enter the following:

```
source('getRecomb.R')
alleles <- read.csv("CP003289_alleles.csv", header=T)
x <- getRecombBetweenStrains(alleles, 'X55989', 'X2011C.3493',
w=1000, multiplier=1, plotResult=T)
```

This should produce the following plot indicating the position of significant clusters of SNPs along the reference genome:

Histogram of snps
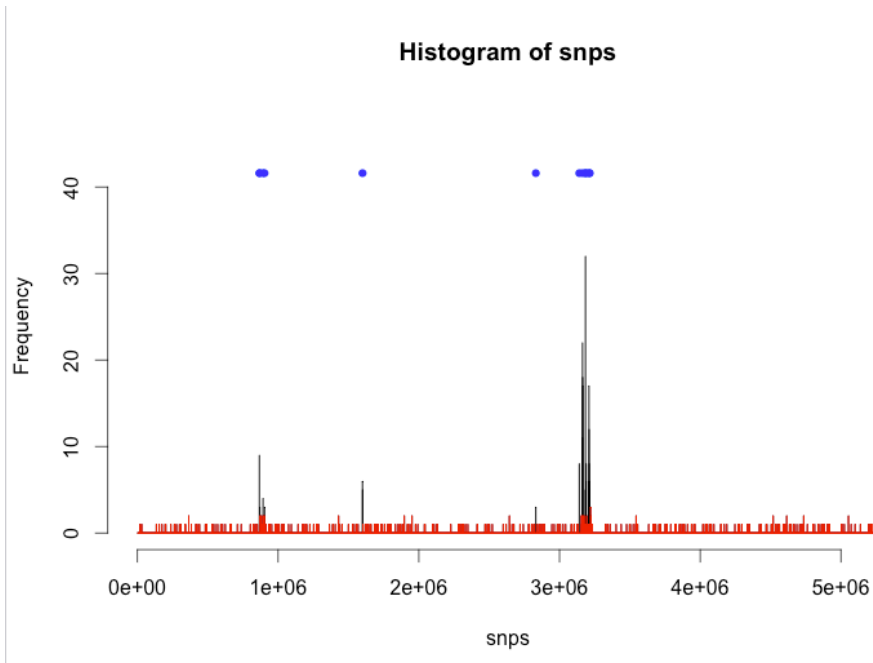
Notice that these peaks line up with the ones we saw using Gingr above. However, we need these regions as coordinates for parseSNPtable. In the same R session:

```
c <- data.frame(x$block[,1],x$block[,2])

write.table(c, file = "recomb.csv",row.names=FALSE,
na="",col.names=FALSE, sep=",")
```

Upload the resulting `recomb.csv` file to the RedDog_Tutorial folder on your cluster account. Then we need to add these regions to the other regions in `CP003289_filtered.coords`:

```
cat CP003289_filtered.coords recomb.csv > 2011C-3493_filtered.coords
```

Then we can run parseSNPtable one more time using this new coordinates file; this time we will also use parseSNPtable to produce a VCF file with information on what round of filtering each SNP was removed.

At this point, we will also introduce one another filter that can sometimes be useful, particularly when looking at clonal groups. We will clean out any SNP pairs that occur in a 3 bp window and any three or more SNPs in a 10 bp window (these are the default settings for the `clean` module of parseSNPtable). The man reason for using `clean` here is that some of the contigs used for shredding were produced from Ion Torrent reads (*e.g.* strains C227_11 and LB226692) and any artifact 'stutters' have not been removed (change to the RedDog output folder before you do the following):

```
python /full_path_to_folder/RedDog/parseSNPtable.py -s 2011C-
3493_full_CP003289_alleles_var.csv -m
filter,vcf,cons,vcf,clean,vcf,aln -o E1167,ON2010 -x
/full_path_to_folder/RedDog_Tutorial/2011C-3493_filtered.coords -c
0.95 -v CP003289 -A
```

Rename the `mfasta` and `_gnr.vcf` files (this is mainly for convenience):

```
mv 2011C-
3493_full_CP003289_alleles_var_2outgroups_var_regionFiltered_cons0.95
_cleanP3W10.mfasta 2011C-3493_final.mfasta
```

```
mv 2011C-
3493_full_CP003289_alleles_var_2outgroups_var_regionFiltered_cons0.95
_cleanP3W10_gnr.vcf 2011C-3493_final_gnr.vcf
```

Open `2011C-3493_final.mfasta` and again remove the first sequence (`>Ref`), then run FastTree:

```
FastTree -gtr -gamma -nt 2011C-3493_final.mfasta > 2011C-
3493_final.tree
```

Download the `2011C-3493_final.tree` and `2011C-3493_final_gnr.vcf` files to your local machine and open them up in Gingr, using the same FASTA and GenBank references as before.

This time we can also examine the SNPs filtered out during each round of filtering (in this case the `filter`, `cons`, and `clean` modules of parseSNPtable). To do this, select `Windows: Variants` from the Gingr menu. You can then use this to examine the distribution of discarded SNPs as well as those finally used to generate the tree. For instance, the figure below shows the SNPs filtered out using the regions defined in the file `2011C-3493_filtered.coords` (repeats, phage, and recombinant regions), after using the `Scale` button to rescale the relative SNP count.
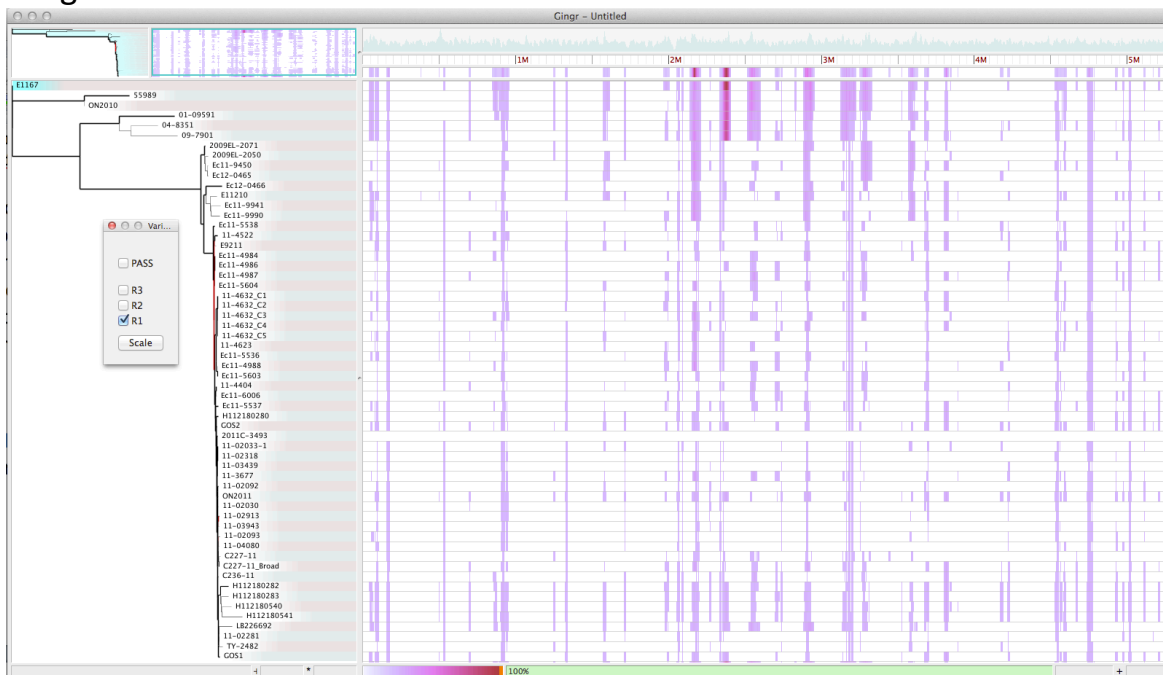
Whilst this view of the relationship between the tree and the SNPs allows us to examine the SNPs, and even see in which genes these SNPs fall (zoom into any SNP to see this - see the Harvest manual for more details on using Gingr), parseSNPtable can also

produce a table of SNP effects using the `coding` module - however, we will leave this as an optional task for the reader to pursue.

Instead, we will use the tree in conjunction with metadata about the strains and coverage statistics for the three plasmids to produce a more informative figure.

If any data set appears to contain large amounts of recombination, a more comprehensive method should be utilized, such as Gubbins. [11] Indeed, we have included a python script to convert the SNP table into the correct format for a Gubbins run (see `snpTable2GenomeAlignment.py` in the RedDog folder). The O104:H4 phylogeny in the paper was constructed using the Gubbins approach rather than the one described and used here to generate the final phylogeny.

c) Combining a Coverage Heat Map and the Phylogeny using R

The tree from above, `2011C-3493_final.tree`, provides information about the phylogenetics of the O104:H4. But we can add more information such as metadata about each isolate, along with a heat map showing plasmid coverage across the O104:H4 (at least for the three found in the outbreak reference).

The tree does need to be formatted before plotting. Open `2011C-3493_final.tree` in FigTree. Again, select the branch leading to E1167 and click on `Reroot`. Then select `Tree -> Increasing Node Order`.
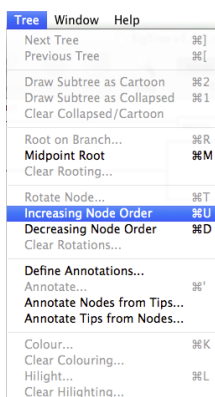
You should end up with a tree like this:
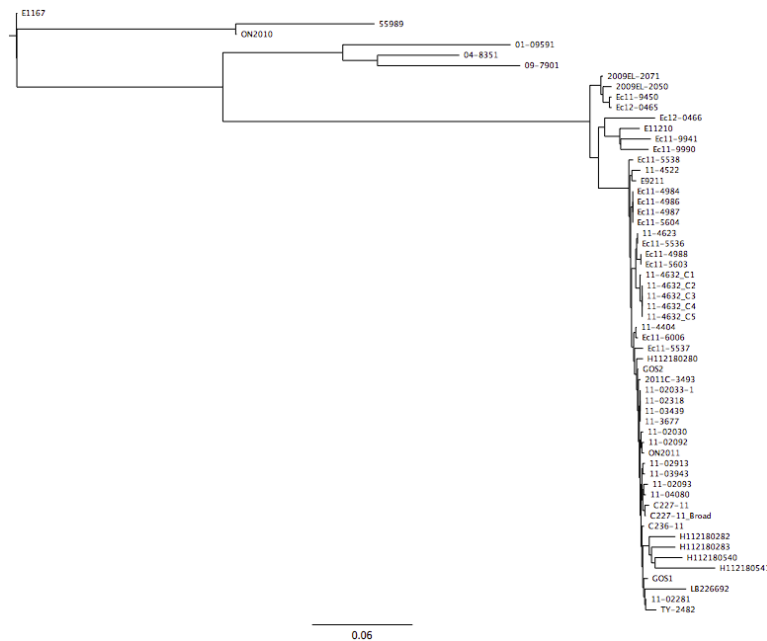
Save this tree. In FigTree, select File -> Export Trees…
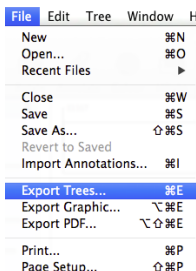
In the dialogue box that opens, for `Tree file format` select `NEXUS`, and click on `Save as currently displayed`. Then, click on `OK`.
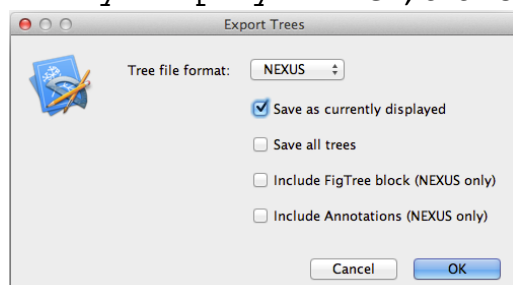
You will be asked for a name; name the new tree `2011C-3493_final.nxs`. Then, click on `Save`.

The data for the heat map in this case comes from the AllStats.txt and Gene Cover files. One of the statistics reported in the former is the coverage of reads for each isolate across each of the four sequences in the GenBank reference. The latter has the coverage for each isolate for all genes in the four reference sequences. To filter these complex tables into a simple text file to generate the coverage heat map in R, use the script, `get_cover.py`, found in the RedDog folder. This script will check the percentage of coverage. If the depth is less than 10 for any replicon, the coverage will be set to 0% for that isolate and replicon pairing. Gene cover is reported 'as is' In this case.

If not already there, change directory to the RedDog_Output folder, then run:

```
python /full_path_to_folder/RedDog/get_cover.py -i 2011C-
3493_full_AllStats.txt -I 2011C-3493_full_GeneSummary.csv -r
CP003290,CP003291,CP003292 -o plasmid_cover.csv
```

Download the resulting `plasmid_cover.csv` file to your local computer.

The final piece of information we need is the metadata for the group of isolates; in this case, whether the strain is a pre-outbreak, outbreak, or post-outbreak isolate with regards to the 2011 outbreak in Germany/France. This information is available at the end of this tutorial (Section 10. O104:H4 metadata). Cut, paste and save this information into a new text file, `metadata.csv`, on your local machine. **[For now you will have to add a trailing comma to each line of metadata.csv for it to work properly - hopefully this will be fixed before general release of the pipeline]**

Now that we have tree, `2011C-3493_final.tree`, the coverage file for the heatmap, `plasmid_cover.csv`, and the metadata file, `metadata.csv`, we can combine these three pieces of information into the one figure using the `plotTree.R` script. If you haven't already, download both the `plotTree.R` in the RedDog folder to your local machine. Then, open R on your local machine and enter the following:

```
source('plotTree.R');
install.packages('ape');
plotTree(tree='2011C-3493_final.nxs', infoFile='metadata.csv',
outputPDF='O104_H4_figure.pdf',
tipColours=c('red','black','darkorange','purple'), legend=T,
legend.pos='bottomleft', colourNodesBy="Group", tip.colour.cex=1.2,
heatmap.colours=colorRampPalette(c('white','orange','blue'),
space='rgb')(100), heatmapData='output_test.csv',treeWidth=50,
infoWidth=0.001, dataWidth=50, tip.labels = T, tipLabelSize = 0.7,
offset=0.03, colLabelCex=0.001)
```

This command will plot the tree against the heat map for plasmid cover across the three plasmids, using the metadata to colour the nodes in the tree by group. Note: `infoWidth` and `colLabelCex` are set to 0.001 to suppress their output; the coloured nodes represent `infoWidth`, and there are too many labels for the columns in the heatmap to be represented clearly.

The coloured lines above the heatmap (next page) represent the three plasmids (red: *pESL-EA11*; orange: *pAA-EA11*; and purple: *pG-EA11*). The first column is the coverage across the whole plasmid, whilst the other columns are the genes (the smallest plasmid has only one gene). These have been added within this document after generating the PDF in R.

Whilst this tree does seem to have good support for the nodes, remember that we generated this using FastTree that is designed to find the best approximation of the maximum likelihood tree. For a more thoroughly tested tree, use of an exhaustive

maximum likelihood method such as RAxML [12] is required. Indeed, the O104:H4 tree in the paper used to display the accessory genome from a pangenome run is generated from this same data set, but with recombination detected by Gubbins (as mentioned previously) and using RAxML instead of FastTree to generate the phylogeny.
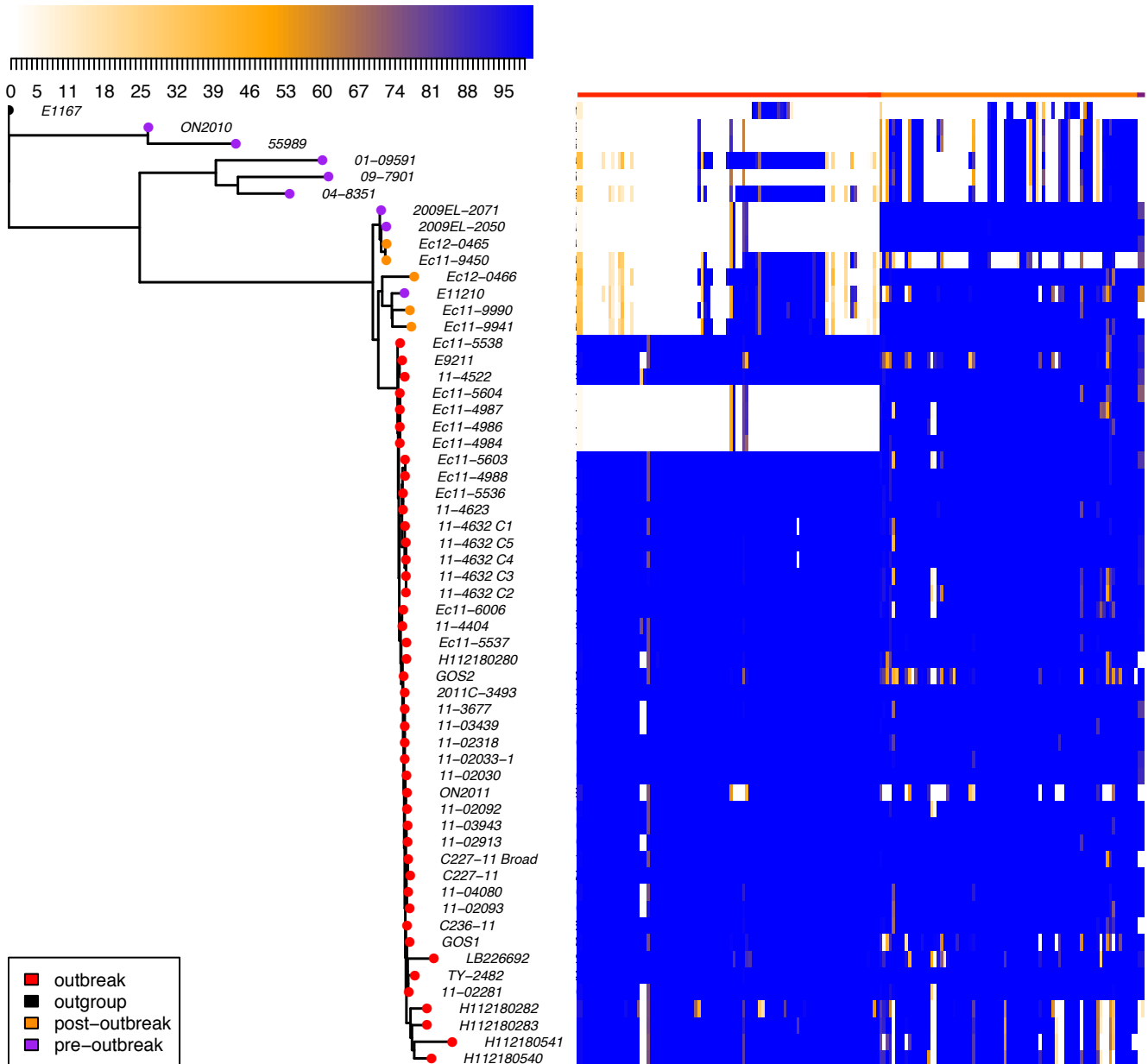
7. Isolate Information

Escherichia coli serotype O104:H4 st 678 isolates available on PATRIC

Any missing values were unavailable from the accompanying metadata, and could not be found through searching using the NCBI Taxon ID. The five isolates in bold are post-outbreak O104:H4 isolates

Genome Name	NCBI Taxon ID	Genome Status	Completion Date	Collection Date	2011 outbreak	Publication
Escherichia coli O104:H4 str. 2009EL-2050	1134782	Complete	2012-09-25	2009	n	[7]
Escherichia coli O104:H4 str. 2009EL-2071	1133853	Complete	2012-09-27	2009	n	[7]
Escherichia coli O104:H4 str. E112/10	1090928	Contigs	2013-04-01	2010	n	[3]
Escherichia coli O104:H4 str. 01-09591	1042803	Contigs	2011-07-20	2001	n	[5]
Escherichia coli O104:H4 str. ON2010	1136217	Contigs	2012-04-12	2010	n	[6]
Escherichia coli O104:H4 str. 09-7901	1048266	Contigs	2011-10-31	2009	n	[8]
Escherichia coli O104:H4 str. 04-8351	1048265	Contigs	2011-10-31	2004	n	[8]
Escherichia coli O104:H4 str. 2011C-3493	1133852	Complete	2012-09-27	2011	y	[7]
Escherichia coli O104:H4 str. 11-4404	1068614	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. 11-4522	1068615	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. 11-4623	1068616	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. Ec11-5536	1068623	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. Ec11-5537	1068624	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. Ec11-5538	1068622	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. GOS2	1073841	Contigs	2011-08-08	2011	y	[4]
Escherichia coli O104:H4 str. 11-02030	1240768	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02033-1	1240769	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02092	1240770	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02093	1240771	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02281	1240772	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02318	1240773	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-02913	1240774	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-03439	1240775	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-03943	1240777	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-04080	1240776	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. 11-4632 C1	1068617	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. 11-4632 C2	1068618	Contigs	2011-10-31	2011	y	[8]

Genome Name	NCBI Taxon ID	Genome Status	Completion Date	Collection Date	2011 outbreak	Publication
Escherichia coli O104:H4 str. 11-4632 C3	1068619	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. 11-4632 C4	1068620	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. 11-4632 C5	1068621	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. C227-11	1048254	Contigs	2011-07-15	2011	y	[1]
Escherichia coli O104:H4 str. E92/11	1090927	Contigs	2013-04-01	2011	y	[3]
***Escherichia coli* O104:H4 str. Ec11-9450**	**1240758**	**Contigs**	**2012-12-11**	**2011**	**n**	**[9]**
***Escherichia coli* O104:H4 str. Ec11-9941**	**1240759**	**Contigs**	**2012-12-11**	**2011**	**n**	**[9]**
Escherichia coli O104:H4 str. Ec11-4984	1240761	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-4986	1240762	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-4987	1240763	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-4988	1240764	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-5603	1240765	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-5604	1240766	Contigs	2012-12-11	2011	y	direct
Escherichia coli O104:H4 str. Ec11-6006	1240767	Contigs	2012-12-11	2011	y	direct
***Escherichia coli* O104:H4 str. Ec11-9990**	**1240760**	**Contigs**	**2012-12-11**	**2011**	**n**	**[9]**
***Escherichia coli* O104:H4 str. Ec12-0465**	**1240778**	**Contigs**	**2012-12-11**	**2012**	**n**	**[9]**
***Escherichia coli* O104:H4 str. Ec12-0466**	**1240779**	**Contigs**	**2012-12-11**	**2012**	**n**	**[9]**
Escherichia coli O104:H4 str. GOS1	1073985	Contigs	2011-08-08	2011	y	[4]
Escherichia coli O104:H4 str. H112180280	1042804	Contigs	2011-06-10	2011	y	direct
Escherichia coli O104:H4 str. H112180282	1052677	Contigs	2011-08-03	2011	y	direct
Escherichia coli O104:H4 str. H112180283	1069643	Contigs	2011-08-03	2011	y	direct
Escherichia coli O104:H4 str. H112180540	1069645	Contigs	2011-08-03	2011	y	direct
Escherichia coli O104:H4 str. LB226692	1040638	Contigs	2011-06-02	2011	y	[5]
Escherichia coli O104:H4 str. ON2011	1136218	Contigs	2012-04-12	2011	y	[6]
Escherichia coli O104:H4 str. TY-2482	1038844	Contigs	-	2011	y	[2]
Escherichia coli O104:H4 str. 11-3677	1048334	Contigs	2011-10-31	2011	y	[8]
Escherichia coli O104:H4 str. H112180541	1048765	Contigs	2011-06-23	2011	y	direct
Escherichia coli O104:H4 str. C227-11 (Broad)	1048254	Contigs	-	2011	y	[8]
Escherichia coli O104:H4 str. C236-11	1048256	Contigs	2011-10-31	2011	y	[8]
Escherichia coli 55989	585055	Complete	2008-12-18	1996-9	n	[13]

8. Tutorial References

[1] Rasko, D. A. *et al.* (2011) Origins of the *E. coli* strain causing an outbreak of hemolytic-uremic syndrome in Germany. *The New England Journal of Medicine* **365,** 709–17

[2] Rohde, H. *et al.* (2011) Open-source genomic analysis of shiga-toxin-producing *E. coli* O104:H4. *The New England Journal of Medicine* **365,** 718–24

[3] Guy, L. *et al.* (2013) Adaptive mutations and replacements of virulence traits in the *Escherichia coli* O104:H4 outbreak population. *PloS one* **8,** e63027

[4] Brzuszkiewicz, E. *et al.* (2011) Genome sequence analyses of two isolates from the recent *Escherichia coli* outbreak in Germany reveal the emergence of a new pathotype: Entero-Aggregative-Haemorrhagic *Escherichia coli* (EAHEC). *Archives of Microbiology* **193,** 883–91

[5] Mellmann, A. *et al.* (2011) Prospective genomic characterization of the German enterohemorrhagic *Escherichia coli* O104:H4 outbreak by rapid next generation sequencing technology. *PloS one* **6,** e22751

[6] Hao, W., Allen, V. G., Jamieson, F. B., Low, D. E. & Alexander, D. C. (2012) Phylogenetic incongruence in *E. coli* O104: understanding the evolutionary relationships of emerging pathogens in the face of homologous recombination. *PloS one* **7,** e33971

[7] Ahmed, S. *et al.* (2012) Genomic comparison of *Escherichia coli* O104:H4 isolates from 2009 and 2011 reveals plasmid, and prophage heterogeneity, including shiga toxin encoding phage stx2. *PloS one* **7,** e48228

[8] Grad, Y. H. *et al.* (2012) Genomic epidemiology of the *Escherichia coli* O104:H4 outbreaks in Europe, 2011. *Proceedings of the National Academy of Sciences of the United States of America* **109,** 3065–70

[9] Grad, Y. H. *et al.* (2013) Comparative genomics of recent Shiga toxin-producing *Escherichia coli* O104:H4: short-term evolution of an emerging pathogen. *mBio* **4,** e00452–12

[10] Sleator, R. D. (2013) A beginner's guide to phylogenetics. *Microbial Ecology* **66**, 1-4

[11] Croucher, N. J. *et al.* (2014) Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. *Nucleic Acids Research* **43**, e15

[12] Stamatakis, A. (2006) RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* **22**, 2688-90

[13] Touchon, M., *et al.* (2009) Organised genome dynamics in the *Escherichia coli* species results in highly diverse adaptive paths. *PloS Genetics* **5**, e1000344

9. Software Packages Used in Tutorial

Version number indicates the version used in preparing this tutorial

Local machine:

FigTree	v1.4.2	http://tree.bio.ed.ac.uk/software/figtree/
Harvest	v1.1.2	https://github.com/marbl/harvest
R	v3.1.1	http://www.r-project.org/

Cluster machine:

RedDog	v1.0	https://github.com/katholt/RedDog

plus dependencies - see RedDog manual

Mummer	v3.23	http://mummer.sourceforge.net/
FastTree	v2.1.7	http://www.microbesonline.org/fasttree/
SAMtools	v1.1	http://www.htslib.org/

The latter two are also dependencies of RedDog. Make sure to install the double-precision version of FastTree (See the FastTree FAQ and this blog-post for more details).

10. O104:H4 metadata

Cut and paste the following into a new text file and save it as `metadata.csv` on your local machine.

```
Group
E1167,outgroup
55989,pre-outbreak
ON2010,pre-outbreak
01-09591,pre-outbreak
04-8351,pre-outbreak
09-7901,pre-outbreak
2009EL-2050,pre-outbreak
2009EL-2071,pre-outbreak
Ec12-0465,post-outbreak
Ec11-9450,post-outbreak
E11210,pre-outbreak
Ec12-0466,post-outbreak
Ec11-9941,post-outbreak
Ec11-9990,post-outbreak
2011C-3493,outbreak
Ec11-5603,outbreak
H112180280,outbreak
11-4522,outbreak
C227-11,outbreak
C227-11_Broad,outbreak
TY-2482,outbreak
H112180282,outbreak
H112180283,outbreak
GOS1,outbreak
ON2011,outbreak
11-4632_C1,outbreak
Ec11-5538,outbreak
11-03943,outbreak
11-4632_C5,outbreak
11-02033-1,outbreak
11-02913,outbreak
11-4632_C4,outbreak
11-02093,outbreak
Ec11-4986,outbreak
H112180541,outbreak
Ec11-5604,outbreak
GOS2,outbreak
11-4632_C3,outbreak
Ec11-4984,outbreak
11-4632_C2,outbreak
11-3677,outbreak
11-4404,outbreak
11-02318,outbreak
11-03439,outbreak
Ec11-4987,outbreak
H112180540,outbreak
LB226692,outbreak
Ec11-5537,outbreak
Ec11-5536,outbreak
C236-11,outbreak
Ec11-6006,outbreak
Ec11-4988,outbreak
11-02281,outbreak
11-02030,outbreak
11-4623,outbreak
11-02092,outbreak
11-04080,outbreak
E9211,outbreak
```