

BS312: Practical 5: ChIP-seq Analysis

Vladimir Teif (vteif@essex.ac.uk)

In this practical we will learn how to analyse ChIP-seq data. Our practical will be in Linux. If you have forgotten about Linux have a look here <https://www.youtube.com/watch?v=XgaE4VlaJqI>, and here <http://manuals.bioinformatics.ucr.edu/home/linux-basics>. In addition, you may consult the guidelines for our departmental computer cluster at this link: <http://genomics.essex.ac.uk/cluster>. If you are already sitting at the practical, please just follow the instructions of the lecturer. Don't be scared, we will guide you step by step through all the process ☺

Objectives:

- Understand ChIP-seq analysis workflow
- Practice working with Linux
- Understand peak calling
- Understand intersection of genomic regions
- Understand enrichment analysis
- Get main ideas of how to work with Bowtie, HOMER, BedTools, UCSC Genome Browser

Introduction. Our practical will be based on the data reported in the study entitled “Integrative genomic analysis reveals widespread enhancer regulation by p53 in response to DNA damage” (Younger et al. (2015) *Nucleic Acids Res.* 43 (9): 4447-4462). The full text of this article is available at <http://nar.oxfordjournals.org/content/43/9/4447.long>. This paper is about chromatin binding of the tumour suppressor protein p53. The authors determine genome-wide p53 binding profiles in human and mouse cells. Their main finding is that p53 binding occurs predominantly within transcriptional enhancers. The authors report both human and mouse ChIP-seq datasets, but mostly analyse the human data in the paper. Today we will perform analysis based on their mouse data.

Finding the data on the Internet. After carefully reading the paper’s abstract we scroll down the bottom of the manuscript to find where the authors have deposited their data. We find the following:

ACCESSION NUMBERS

.....

The Gene Expression Omnibus accession number for the RNA-Seq and ChIP-Seq data reported in this paper is **GSE55727**.

SUPPLEMENTARY DATA

.....

Supplementary Data are available at NAR Online.

Using the Gene Expression Omnibus (GEO) accession number GSE557227 reported by the authors, we find their data at the following link:

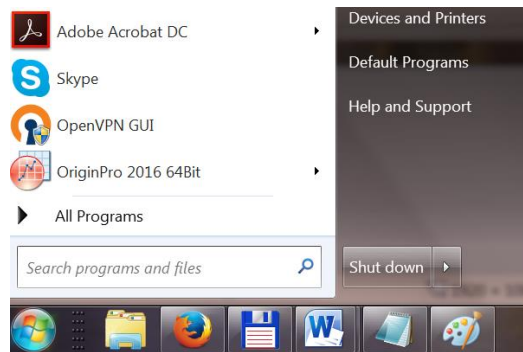
<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE55727>

Opening this link in the browser, we can see the complete description of the experimental details of this study, and the list of the samples which they have deposited (you have to click on “more” next to the sample list):

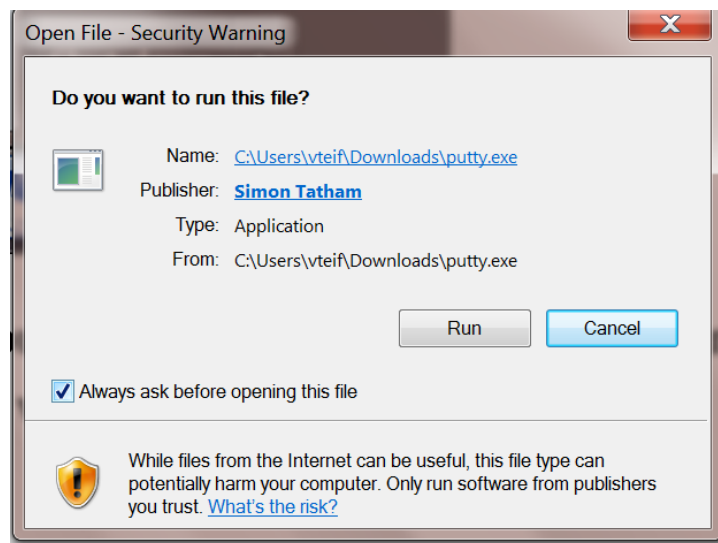
Samples (24)	GSM1342483	GM06170_RNA_unt_rep1
Less...	GSM1342484	GM06170_RNA_unt_rep2
	GSM1342485	GM06170_RNA_dox_rep1
	GSM1342486	GM06170_RNA_dox_rep2
	GSM1342487	GM06170_ChIP_input
	GSM1342488	GM06170_ChIP_p53
	GSM1342489	GM00011_RNA_unt_rep1
	GSM1342490	GM00011_RNA_unt_rep2
	GSM1342491	GM00011_RNA_dox_rep1
	GSM1342492	GM00011_RNA_dox_rep2
	GSM1342493	GM00011_ChIP_input
	GSM1342494	GM00011_ChIP_p53
	GSM1342495	MEF_WT_RNA_unt_rep1
	GSM1342496	MEF_WT_RNA_unt_rep2
	GSM1342497	MEF_WT_RNA_unt_rep3
	GSM1342498	MEF_WT_RNA_dox_rep1
	GSM1342499	MEF_WT_RNA_dox_rep2
	GSM1342500	MEF_WT_RNA_dox_rep3
	GSM1342501	MEF_ChIP_input
	GSM1342502	MEF_ChIP_p53
	GSM1375967	MEF_KO_RNA_unt_rep1
	GSM1375968	MEF_KO_RNA_unt_rep2
	GSM1375969	MEF_KO_RNA_dox_rep1
	GSM1375970	MEF_KO_RNA_dox_rep2

We will be working with the samples MEF_ChIP_p53 and MEF_ChIP_Input. “MEF” stands for mouse embryonic fibroblasts. “p53” stands for the sample which has undergone ChIP-seq with antibody against p53 protein, and “Input” is the same sample, but sequenced without antibody. I have already downloaded these files to our computer cluster and unpacked them. Our task for this practical will be to analyse these data: check whether the conclusions of the authors of the paper are correct (or may be suggest new scientific conclusions and make a discovery!)

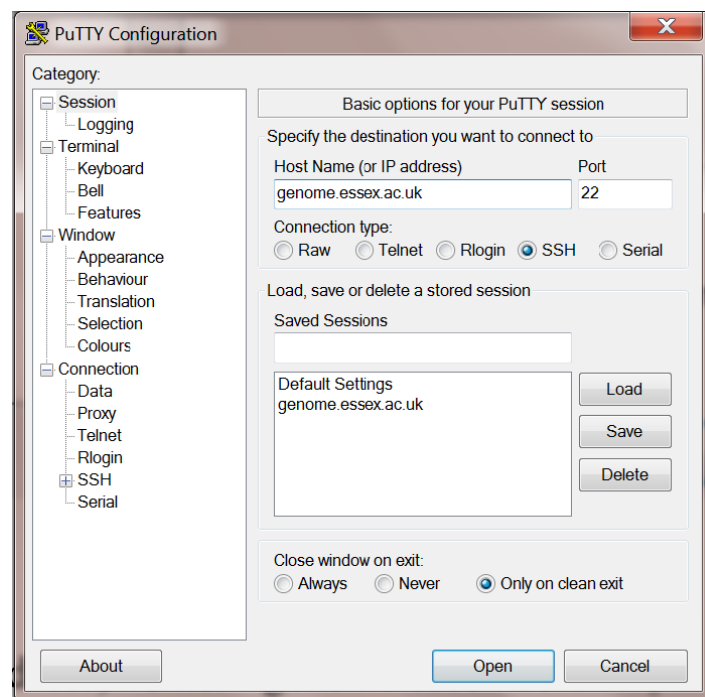
A reminder how to connect to the computer cluster using Putty. Our calculations deal with large files, and therefore have to be performed on the computer cluster. This is exactly how most serious sequencing analysis is being performed nowadays. Firstly, we need to connect from your computers to the cluster. We will do this using a program called **Putty**. A detailed description of this program can be found here: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. Let’s open the “Start” menu of your Windows computers and type “Putty” in the “Search programs and files” field:



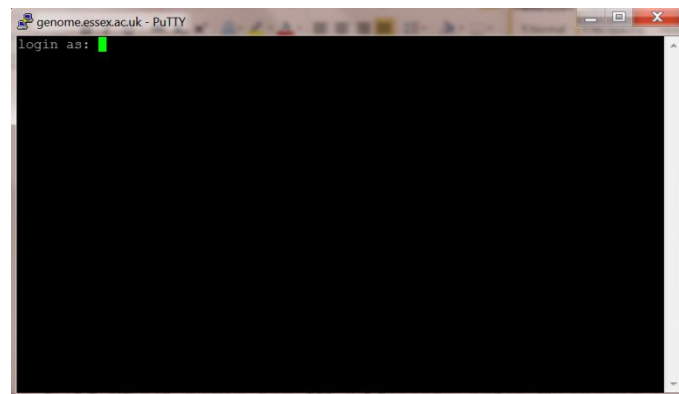
If Putty already exists on your computer, it will show up in the search results, and after clicking on this program it will open the following window:



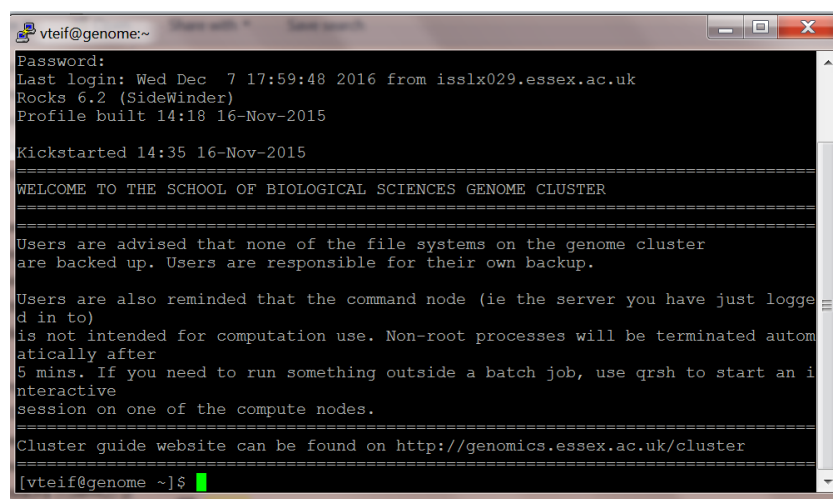
Click “Run” and then agree to add the security key when it asks for it. Then Putty opens like this:



In the field “host name (or IP address)” instead of “genome.essex.ac.uk” as shown in the picture enter “ceres.essex.ac.uk”. Then click “Open”. It will open the black terminal screen:



Enter your university user name (the same as you use for your email), and press Enter. Then enter your university password and again click Enter. Note that when you are typing your password the cursor will not move on the screen, but this is fine, the computer is reading what you are writing. If you have correctly entered your password the welcome screen appears:



You are now located in your home directory on the cluster. Today we will be working in the “interactive mode”, that is, everything typed in the terminal will be executed immediately as we type. To switch to the interactive mode type the following command:

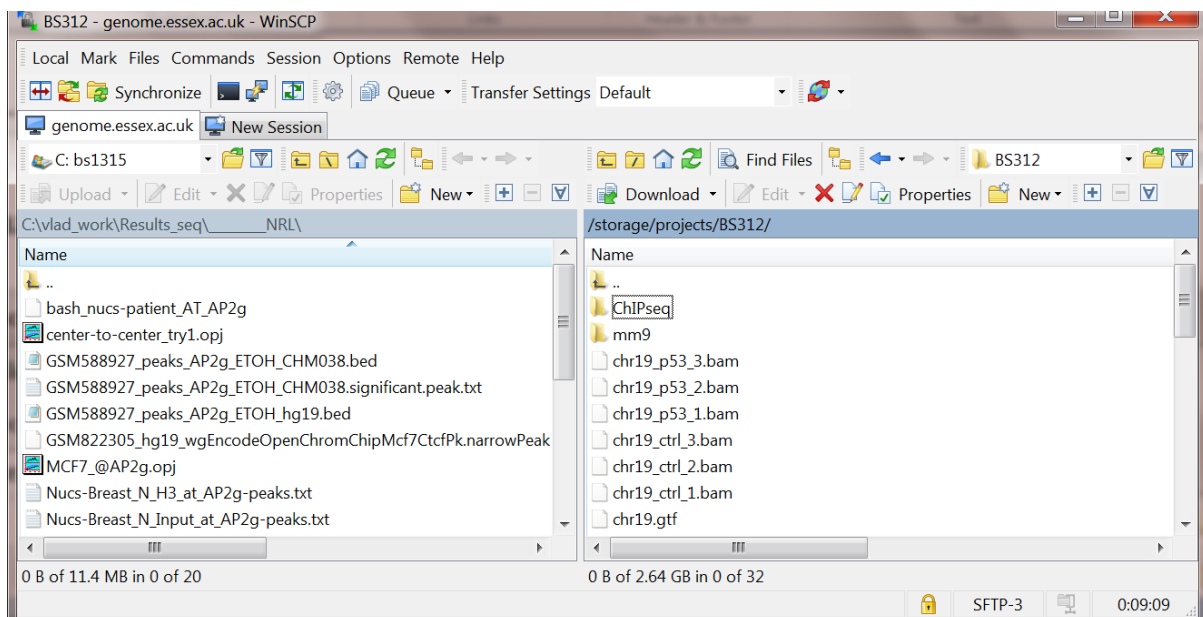
```
qcrsh
```

We have now entered the interactive mode.

A simple command “`ls`” will show you the content of the current directory. Just type “`ls`” and press [Enter]. We will be doing all the calculations in the home directory. The module materials including the data that we need for the analysis are stored in another directory which is situated at the following path: `/storage/projects/BS312/ChIPseq`. In order to go to any directory we can type the command `cd` followed by the path to the desired directory.

In order to go back to your home directory you can type the command “`cd`” without any parameters, or alternatively “`cd ~`”. In both cases it will bring you home.

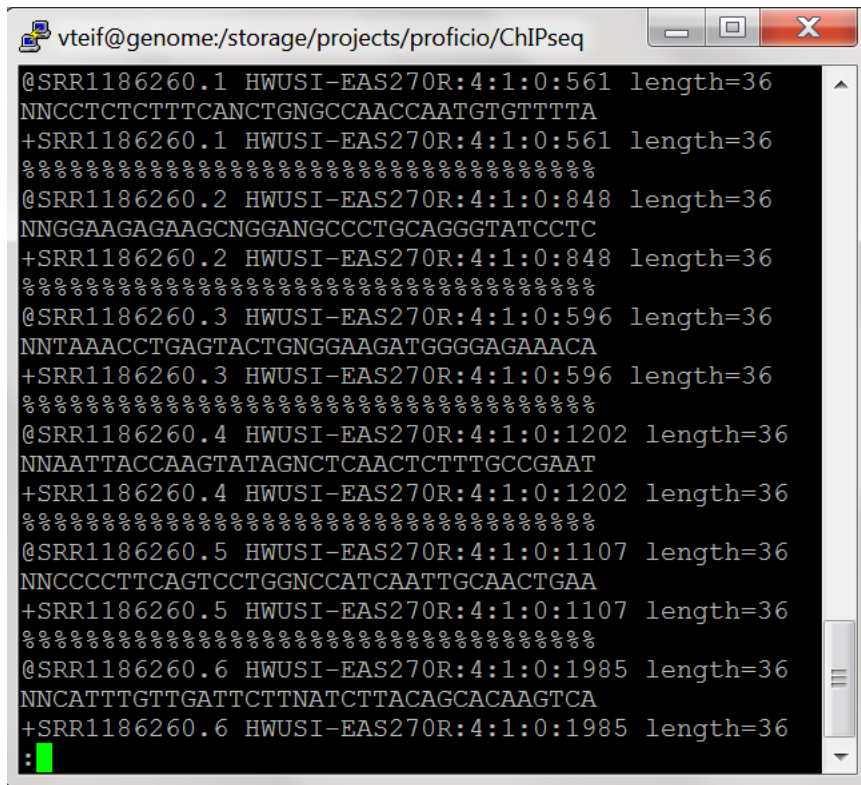
Connecting to the computer cluster using WinSCP. As you have seen previously, there is also user-friendly software called WinSCP that allows to manage files on the cluster without typing any Linux commands. The same directory can be viewed in WinSCP as follows:



Here the right panel in WinSCP shows the directory on the cluster, and the left panel shows the directory on your local computer. You can copy files between the cluster and your computer by just dragging them by mouse. You can also view the content of the files by double-clicking on them. The files will then open in a text editor, where you can view and edit them. This can be only done for small files. Please do not attempt to do this for large files, as their opening on your computer can take ages. Large files can be viewed in Putty using the command “less”.

Task 1. Map p53 ChIP-seq reads. Our first task is to map the DNA fragments obtained after p53 ChIP-seq. We can view the raw files with unmapped ChIP-seq reads by typing in Putty the following command:

```
less /storage/projects/BS312/ChIPseq/p53.fastq
```



This is the FASTA file with raw DNA reads that are obtained after the sequencing. The complete sequence of each DNA read is listed here. It is a large file, because it contains information of several million DNA reads. In this case, the length of each read is 36 nucleotides.

In order to close the file view in Putty you need to press letter [Q] on the keyboard (meaning “quit”).

Se we want to map all these reads to the mouse genome. That is, we want to know where each of these reads is situated in the genome, what is its genomic coordinate.

If you’ve been travelling around the cluster in Putty let’s return to the home directory by typing

```
cd ~
```

We will be mapping reads with the program Bowtie. You have already done this before, but now we will look in detail in the parameters. The command that we need to run is the following:

```
bowtie -t -v 2 -p 2 -m 1 --solexa-quals mm9  
/storage/projects/BS312/ChIPseq/p53.fastq p53.map
```

The execution of this command will take ~17 minutes. Let’s run this command, and meanwhile let’s think what we just did. The line starting with “bowtie” as you already know executes the software called Bowtie, which maps DNA reads to the genome. A detailed description of the program Bowtie is available at its web site: <http://bowtie-> <http://bowtie-bio.sourceforge.net/manual.shtml>

Bowtie has many parameters. You need to understand just those which we use today:

-t tells Bowtie to print the amount of wall-clock time taken by each phase;

-v tells Bowtie that alignments may have no more than V mismatches, where V may be a number from 0 through 3 set using the -v option. This is an important parameter! What is its value?

-p tells Bowtie to launch a specified number of parallel search threads. Each thread runs on a different processor/core and all threads find alignments in parallel, increasing alignment throughput by approximately a multiple of the number of threads;

-m suppresses all alignments for a particular read or pair if more than <the number which follows m> reportable alignments exist for it.

--solexa-quals tells Bowtie that our data are obtained with the Illumina Solexa sequencer;

"mm9" tells the program that we are working with the mouse genome assembly called mm9;



Finally, the parameter `"/storage/projects/BS312/ChIPseq/p53.fastq"` is the path to the Input file which we want to map (it is in FASTQ format), and the output file `"p53.map"` is the name of the mapped Input reads in the Bowtie format.

Task 2. Map ChIP-seq Input reads. At the previous step we have mapped all reads corresponding to the p53 ChIP-seq (`p53.fastq`). Now we need to repeat the same procedure, but for the reads corresponding to the control experiment called "Input" (`Input.fastq`), which is conducted in the same cells in the same manner, just without the antibody. In order to make it faster, we do not need to wait until the calculation of the step 1 is finished, but we can run another calculation in parallel. To do so, open another Putty window, switch to the interactive mode:

```
qrsh
```

Then type the following command:

```
bowtie -t -v 2 -p 2 -m 1 --solexa-quals mm9  
/storage/projects/BS312/ChIPseq/Input.fastq Input.map
```

The calculations for the Task 1 and 2 that we have submitted will be running for about 17 minutes.

Task 3. Convert mapped reads to BED format. The Bowtie "map" format which stored mapped reads in the file that we have obtained at the previous steps, reports all reads and the genomic coordinates to which it maps, and several other parameters which we do not need. (Remember, that in order to see how each file looks we need to print "less" followed by the file name). For the purpose of further analysis we need to convert the Bowtie format to a simpler BED format (which starts with the following columns: chromosome, region start, region end, strand).

In each of the two instances of Putty that we have (after the previous calculation has finished), we can type the following commands (correspondingly for p53 and for Input):

In the first Putty window (where you did mapping of the p53 file), type the following:

```
perl -w /storage/projects/BS312/ChIPseq/bowtie2bed.pl p53.map p53.bed
```

In the second Putty window (where you previously did mapping of Input), type the following:

```
perl -w /storage/projects/BS312/ChIPseq/bowtie2bed.pl Input.map Input.bed
```

You can view your output files by typing the following:

```
less Input.bed
```


Task 4. Convert BED files to chromosome-wide occupancy files (create HOMER tag directory).

The next program that we will be using for our analysis is called HOMER. A detailed description of this program is available at <http://homer.salk.edu/homer/>. HOMER allows us to do several things today: Firstly, we will be calculating protein binding maps (genome-wide occupancy based on ChIP-seq, which will be done by creating HOMER directories inside your home directory). Secondly, we will use HOMER to call peaks (find genomic regions significantly enriched with our protein of interest, this is called “peak calling”). Finally, later we will use HOMER to identify DNA sequence motifs characteristic for these peaks (e.g. transcription factor binding sites). Our current task is to calculate the genome-wide occupancy of ChIP-seq reads in order to call peaks later. The genome-wide occupancy profiles per each chromosome are stored by HOMER in so called tag directory. Here is what you need to do:

In the first Putty window (where you did calculation of the p53 files), type the following:

```
makeTagDirectory HOMER_p53 p53.bed -genome mm9
```

In the second Putty window (where you did calculations of the Input files), type the following:

```
makeTagDirectory HOMER_Input Input.bed -genome mm9
```

What have we done? In these commands we have told HOMER to create the tag directory with name HOMER_p53 for the p53 ChIP-seq based on the mapped file p53.bed and HOMER_Input based on the mapped BED file named Input.bed. We have also told HOMER that we are working with the mouse genome (mm9).

Task 5. Find p53 peaks (genomic locations of bound p53 protein).

OK, so at the previous steps HOMER has calculated its tag directories for both the Input and p53 ChIP-seq, and now we can find p53 peaks. If all the previous calculations finished, you can close one of the Putty windows, we will just need one Putty window. At the next step, in order to identify p53 binding sites, we need to know that a peak appears at a given site in the p53 sample, but not in the Input sample. This will be done by so called peak calling, an algorithm which is also realised in HOMER. We need to locate peaks of high density of ChIP-seq reads, and to make sure that these are not just noise (this will be achieved by comparing them with the control experiment, the “Input”).

If you are outside of the home directory, go first home (type “cd ~”). Our next command is this:

```
findPeaks HOMER_p53 -style factor -o auto -i HOMER_Input
```

The command above tells HOMER that we want to determine peaks based on the HOMER directory with ChIP-seq data called HOMER_p53 (which we have created before), and it will expect sharp peaks (`-style factor`), and it will compare these peaks with the peaks found based on the Input sample in the directory called HOMER_Input (which we have also created before). A detailed description of HOMER parameters can be found at <http://homer.salk.edu/homer/ngs/peaks.html>

This command will be executing for about 15 minutes, and after that the resulting `peaks.txt` file will be located inside the folder `/HOMER_p53`.

Task 6. Interpreting the results of the peak calling

Let us look at our results and understand them. Change your directory to /HOMER_p53, and then read the file `peaks.txt`. (You can do this in WinSCP by double-clicking on it).

How many peaks did you find? What is the average width of the peak?

Now let's compare the number of peaks that you have found with the number of peaks reported by the authors of this study. Remember where the data came from? We can look in the GEO database, where we took the data from <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE55727>). At the bottom of the entry, we can see the following:

Relations			
BioProject	PRJNA240784		
SRA	SRP039598		
Download family		Format	
SOFT formatted family file(s)		SOFT ?	
MINIML formatted family file(s)		MINIML ?	
Series Matrix File(s)		TXT ?	
Supplementary file		Size	Download
SRP/SRP039/SRP039598			(ftp)
GSE55727_Human_ChIP_peaks.bed.gz		24.2 Kb	(ftp) (http)
GSE55727_Human_RNA_Expression_Matrix.txt.gz		1000.0 Kb	(ftp) (http)
GSE55727_MEF_ChIP_peaks.bed.gz		27.6 Kb	(ftp) (http)
GSE55727_MEF_KO_RNA_Expression_Matrix.txt.gz		570.5 Kb	(ftp) (http)
GSE55727_MEF_WT_RNA_Expression_Matrix.txt.gz		784.4 Kb	(ftp) (http)
Raw data provided as supplementary file			
Processed data is available on Series record			

We are particularly interested in the file “GSE55727_MEF_ChIP_peaks.bed.gz”. This is the file with the peaks determined by the authors. You can right-click on this file to copy its Internet address, and then download it to the cluster using the following command:

```
wget
ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE55nnn/GSE55727/suppl/GSE55727%5FMEF%5FChIP%5Fpeaks%2Ebed%2Egz
```

Now we need to unpack this file. The file is packed as suggested by the extension “.gz” at the end. To unpack the file we use the following command:

```
gunzip GSE55727_MEF_ChIP_peaks.bed.gz
```

After the file has been unpacked, the “.gz” disappeared from the end of the file name, and the file size increased. We can then refresh the directory content in WinSCP, double-click on this file `GSE55727_MEF_ChIP_peaks.bed` and check how many p53 binding sites are found there.

*We can also look inside this file in Putty by typing the following command:

```
less GSE55727_MEF_ChIP_peaks.bed
```

We can also count the number of lines in this file in Putty by typing the following command:

```
wc GSE55727_MEF_ChIP_peaks.bed
```

The command “wc” stands for “word count”. It outputs the number of lines, words and symbols. We are interested in the number of lines, because each line corresponds to one peak. The number of lines is given by the first value in the output. This is what I’ve got when I typed this command:

```
[vteif@chromosome-0-2 ~]$ wc GSE55727_MEF_ChIP_peaks.bed
3100  9300 74213 GSE55727_MEF_ChIP_peaks.bed
[vteif@chromosome-0-2 ~]$
```

It means there are 3100 lines in this file. There is one peak per line. How many peaks are there? ☺

Why is the number of our peaks different from the number of peaks reported by Younger et al?

Now that we know that the number of peaks that we have identified and the number of peaks reported by the authors is different, we can do more detailed analysis. We can ask questions such as how many peaks are the same between these two files, and so on.

Peaks are genomic regions (defined by the chromosome, region start, region end, etc). In the BED format (the format supplied by the authors), we have columns in exactly this order (chromosome, region start, region end). The format of the file “peaks.txt” that we obtained after HOMER analysis is different (for example, the first column is the peak name). We need to convert it to the standard BED format. To do so, open the file `HOMER_p53/peaks.txt` in WinSCP (just double-click on it). As you can see, it starts with a lot of descriptive lines and the columns with peak coordinates follow only after these header lines. The header lines need to be deleted to convert it to the proper BED format. Delete all the header lines using the WinSCP text editor. Close the file in the text editor in WinSCP. Say “Yes” to save the changes. Enter your university password when prompted for the password.

Another thing that we need to change in our peaks file is the order of columns. There is an easy one-liner command to print only certain columns. For example, to make our peaks file in exactly the same format as the file deposited by the authors of this paper to the GEO database, we want to print only columns 2, 3 and 4 (chromosome name, region start and region end). Here is the command to do so:

```
awk -v OFS='\t' '{ print $2, $3, $4 }' HOMER_p53/peaks.txt
> peaks_formatted.bed
```

If you did it correctly, the new file `peaks_formatted.bed` contains only three columns (chromosome name, region start and region end).

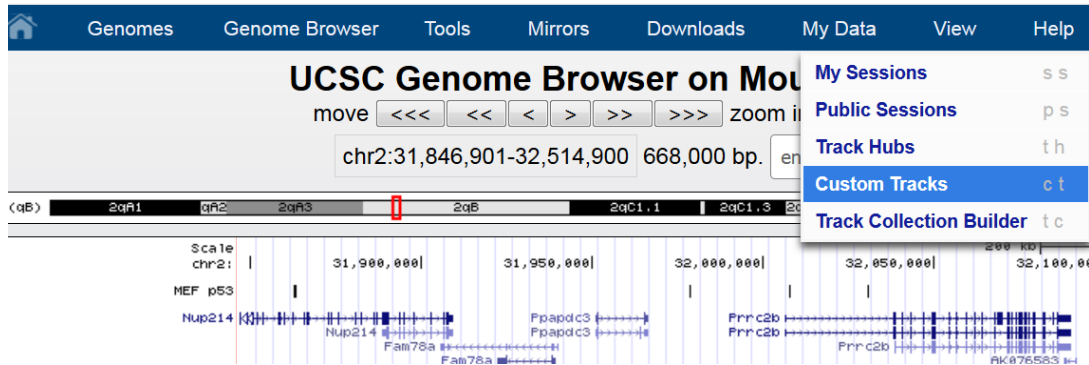
Task 7. Visualisation in genome browser. Now that we have p53 peaks (one dataset determined by us, called `peaks_formatted.bed` and another one reported by the authors of the paper `GSE55727_MEF_ChIP_peaks.bed`), we can visually compare them in a genome browser.

***The description below explains how you can upload your data to the UCSC Genome Browser and I strongly recommend you to try doing this on your own. However, if you just want to look at the data that I have uploaded you can simply click on this link (which I do not recommend☺)

https://genome.ucsc.edu/cgi-bin/hgTracks?db=mm9&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr8%3A13460513-13637512&hgid=697036851_2oAUMwYSDxCaXYOdoJk3AND2CJj

Here is how to upload you custom track to the UCSC Genome Browser:

Let's go to the UCSC Genome Browser on the Internet (<https://genome-euro.ucsc.edu/index.html>), select mouse genome (NCBI37/mm9) Assembly), and then upload our two files with p53 peaks as custom tracks. Here is how to do this. Go to menu [My data] >[Custom Tracks]:



Then we need to select mouse genome mm9 and upload file GSE55727_MEF_ChIP_peaks.bed:

The screenshot shows the 'Add Custom Tracks' form. At the top, there are dropdown menus for 'clade' (set to Mammal), 'genome' (set to Mouse), and 'assembly' (set to July 2007 (NCBI37/mm9)). Below these, there is a paragraph of text explaining the data format requirements (bigBed, bigWig, BAM, barChart, VCF, BED, BED detail, bedGraph, broadPeak, CRAM, GFF, GTF, interact, MA) and a link to the User's Guide. Another paragraph mentions 'Track Hubs'. At the bottom, there are two input fields: 'Paste URLs or data:' and 'Or upload:'. The 'Or upload:' field contains the file name 'GSE55727_MEF_ChIP_peaks.bed' and a 'Browse...' button. A 'Submit' button is also present.

After uploading this file it will appear online with the name “User track”, which we can change to a more informative caption “p53 published”:

The screenshot shows the 'Update Custom Track: User Supplied Track [mm9]' form. It contains a paragraph of text explaining the data format requirements and a link to the User's Guide. Another paragraph mentions 'Track Hubs'. At the bottom, there is an 'Edit configuration:' section with a text input field containing the code 'track name='p53 published' description='p53 published''. A 'Submit' button is located to the right of the input field.

Then we can similarly upload the second file with our own peaks ([peaks_formatted.bed](#)) and change the name of the corresponding track e.g. to “p53 our peaks”. We now have two tracks:

Genomes Genome Browser Tools Mirrors Downloads My Data Help

Manage Custom Tracks

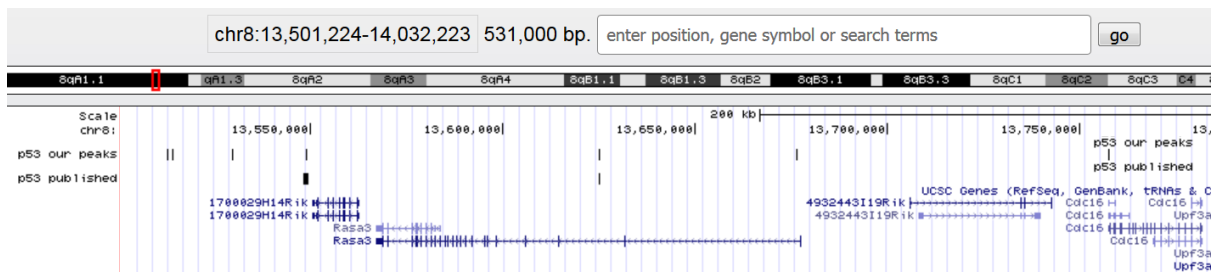
genome Mouse assembly July 2007 (NCBI37/mm9) [mm9]

Name	Description	Type	Doc	Items	Pos	delete
p53 our peaks	p53 our peaks	bed		15377	chr8:	<input type="checkbox"/>
p53 published	p53 published	bed		3100	chr2:	<input type="checkbox"/>

view in Genome Browser go

add custom tracks

Next we can just click “go” to display both these tracks in the genome browser. The genome browser allows to change magnification and move to any genomic regions, e.g. this one:



As we can see, some our peaks coincide with the published peaks. But we also have additional peaks. How many of our peaks intersect with the published peaks? Let's see.

Task 8. Intersect genomic regions using BedTools

In the next task we want to intersect the genomic regions which we have identified as p53 binding sites with those reported by the authors of the original paper, and also with the regions corresponding to mouse enhancers and promoters. Here is a schematic picture which explains the “intersection” between two sets of genomic regions:



Intersection is one of the main concepts in ChIP-seq analysis. To do this we will use command `IntersectBed` from the software package `BedTools`.

A detailed description of all parameters of this command is provided at the following link: <http://bedtools.readthedocs.io/en/latest/content/tools/intersect.html>

Let us first look at the following command:

```
intersectBed -a GSE55727_MEF_ChIP_peaks.bed -b peaks_formatted.bed -u
> intersection_GSE55727_peaks_with_our_peaks.bed
```

It intersects two files, “-a” and “-b”. In our case, it intersects the file `GSE55727_MEF_ChIP_peaks.bed` (the peak coordinates provided by the authors of the manuscript) with the file `peaks_formatted.bed` (peak coordinates which we have determined). The results are in the file `intersection_GSE55727_peaks_with_our_peaks.bed`.

Similarly, we can also intersect our peaks with the coordinates of the enhancer regions which I have collected for you from the FANTOM consortium (they have mapped all mouse enhancers), and the annotated mouse enhancers. Now we can do the following intersections:

8.1. Intersect p53 peaks reported by Young et al with p53 peaks that we have found:

```
intersectBed -a GSE55727_MEF_ChIP_peaks.bed -b peaks_formatted.bed -u  
> intersection_GSE55727_peaks_with_our_peaks.bed
```

Count the number of peaks in the resulting file:

```
wc intersection_GSE55727_peaks_with_our_peaks.bed
```

8.2. Intersect our p53 peaks with mouse enhancers:

```
intersectBed -a peaks_formatted.bed -b  
/storage/projects/BS312/ChIPseq/enhancers_mm9.bed -u >  
intersection_our_peaks_with_enhancers.bed
```

Count the number of peaks in the resulting file:

```
wc intersection_our_peaks_with_enhancers.bed
```

8.3. Intersect our p53 peaks with mouse promoters:

```
intersectBed -a peaks_formatted.bed -b  
/storage/projects/BS312/ChIPseq/promoters_mm9.bed -u >  
intersection_our_peaks_with_promoters.bed
```

Count the number of peaks in the resulting file:

```
wc intersection_our_peaks_with_promoters.bed
```

8.4. Intersect p53 peaks reported by Young et al with mouse enhancers:

```
intersectBed -a GSE55727_MEF_ChIP_peaks.bed -b  
/storage/projects/BS312/ChIPseq/enhancers_mm9.bed -u >  
intersection_GSE55727_peaks_with_enhancers.bed
```

Count the number of peaks in the resulting file:

```
wc intersection_our_peaks_with_promoters.bed
```

8.5. Intersect our p53 peaks reported by Young et al with mouse promoters:

```
intersectBed -a GSE55727_MEF_ChIP_peaks.bed -b  
/storage/projects/BS312/ChIPseq/promoters_mm9.bed -u >  
intersection_GSE55727_peaks_with_promoters.bed
```

Count the number of peaks in the resulting file:

```
wc intersection_our_peaks_with_promoters.bed
```

The results are the following files, which will be created in your home directory:

```
intersection_GSE55727_peaks_with_our_peaks.bed
intersection_our_peaks_with_enhancers.bed
intersection_our_peaks_with_promoters.bed
intersection_GSE55727_peaks_with_enhancers.bed
```

For each of these files, if we forgot the numbers of peaks in them, we can retrieve these numbers using the “wc” command. This will give the number of peaks in each corresponding intersection.

How many peaks reported by Young et al intersect with our peaks? How many of our peaks intersect with promoters? How many of our peaks intersect with enhancers? Is it a lot?

Task 9. Understanding enrichments of peaks at regulatory regions: How many is really many?

At the previous step we have learned that the authors of the paper have determined 3,100 p53 peaks, out of which 2,709 intersect with the peaks which we have determined. Is it a lot? It means that 87% (2709/3100) of the peaks determined by the authors of the paper are identical to our peaks. So we have found most of their peaks, plus some additional peaks.

In total we have found 15,377 p53 bound peaks. Out of these,

- 1,069 intersect with enhancers
- 4,854 intersect with promoters

In other words, $1069/15,377 = 7\%$ of our peaks intersect with enhancers, and $4854/15377 = 32\%$ of our peaks intersects with promoters. Is it a lot? To understand whether this is a lot, we need to compare these values with those given by chance. Say, let us select 15377 random regions with about the same width of 167 nucleotides as the average width of our p53 peaks. What would be the percentage of overlapping of those regions with genomic features such as enhancers and promoters?

9.1. We can ask BedTools to create for us a random file with the same number of regions and the same region length as in our p53 peaks. To do so, we will use a comand “shuffleBed”, which randomly shuffles regions contained in our file across the whole genome:

```
shuffleBed -i peaks_formatted.bed -g
/storage/projects/BS312/ChIPseq/mm9.genome >
shuffled_peaks_formatted.bed
```

9.2. Then we can ask BedTools, how many of these random peaks would intersect with enhancers:

```
intersectBed -a shuffled_peaks_formatted.bed -b
/storage/projects/BS312/ChIPseq/enhancers_mm9.bed -u >
intersection_random_peaks_with_enhancers.bed
```

Now let’s count the number of peaks in the resulting file:

```
wc intersection_random_peaks_with_enhancers.bed
```

9.3. Then we can ask how many of these random peaks would intersect with promoters:

```
intersectBed -a shuffled_peaks_formatted.bed -b  
/storage/projects/BS312/ChIPseq/promoters_mm9.bed -u >  
intersection_random_peaks_with_promoters.bed
```

Now let's count the number of peaks in the resulting file:

```
wc intersection_random_peaks_with_promoters.bed
```

The results appear in the following two new BED files:

```
intersection_random_peaks_with_enhancers.bed  
intersection_random_peaks_with_promoters.bed
```

If we forgot how many peaks are in each file we can retrieve these numbers using command “wc”.

How many peaks are in these files? What is the percentage of random regions overlapping with promoters? What is the percentage of random regions overlapping with enhancers?

9.4. Now let's calculate enrichment of p53 sites at promoters. To do so, we just need to divide the number of p53 sites overlapping with promoters by the number of randomly shuffled p53 sites overlapping with promoters. If the resulting ratio is >1 then p53 sites overlap with promoters more frequently than it is expected by chance for random regions. If the resulting ratio is <1 then p53 sites overlap with promoters less frequently than it is expected by chance. If the resulting ratio is around 1 then p53 sites overlap with promoters to the same degree as expected by chance for random regions.

9.5. Now we can do the same enrichment calculation for p53 sites overlapping with enhancers.

Do we have enrichment of p53 binding at promoters? At enhancers?

Finally, are the conclusions of Younger et al. about p53 enrichment at enhancers correct?