Introduction

The cancer silencer p53 is a record factor that facilitates the cell reaction to DNA harm. Here we give a coordinated examination of p53 genomic inhabitance and p53-subordinate quality guideline in the splenic B and non-B cell compartments of mice presented to entire body ionizing radiation, giving knowledge into general standards of p53 action in vivo. In unstressed circumstances, p53 bound not many genomic targets; enlistment of p53 by ionizing radiation expanded the quantity of p53 bound locales, prompting profoundly covering profiles in the different cell types. Examination of these profiles with chromatin highlights in unstressed B cells uncovered that, upon enactment, p53 confined at dynamic advertisers, distal enhancers, and a more modest arrangement of plain distal locales. At advertisers, acknowledgment of the accepted p53 theme as well as restricting strength were related with p53-subordinate transcriptional initiation, yet not constraint, it was no doubt circuitous to demonstrate that the last option. p53-enacted targets comprised the center of a cell type-free reaction, superimposed onto a cell type-explicit program. Center reaction qualities included the greater part of the known p53-directed qualities, as well as numerous new ones. The information addresses an exceptional portrayal of the p53-controlled reaction to ionizing radiation in vivo. The mice were presented to entire body ionizing radiation and successions were separated from both Bcells and non-B cells from the spleens of the mice. Two genotypes of mice were utilized: mice with p53 took out and the wild-type C57/Bl6. There were 4 different gathering mixes including the 2 unique genotypes; every genotype was exposed to the ionizing radiation as well as control/mock.

Treatment of the mice that were either controls or treated with ionizing radiation to decide response of p53.

	Genotype	**Treatment**
Group 1	p53	Mock
Group 2	C57/Bl6	IR
Group 3	p53	IR
Group 4	C57/Bl6	IR

High-throughput mRNA sequencing (RNA-seq) offers the capacity to find new qualities and records and measure record articulation in a solitary examine. In any case, even little RNA-seq tests including just a solitary example produce huge volumes of crude sequencing peruses — current instruments create more than 500 gigabases in a solitary run. Besides, sequencing costs are decreasing dramatically, making the way for reasonable customized sequencing and welcoming examinations with product figuring and its effect on society. Albeit the volume of information from RNA-seq tests is frequently troublesome, it can give tremendous knowledge. Similarly as cDNA sequencing with Sanger sequencers radically extended our inventory of known human qualities, RNAseq uncovers the full collection of elective graft isoforms in our transcriptome and reveals insight into the most extraordinary and most cell- and setting explicit records. Moreover, in light of the fact that the quantity of peruses delivered from a RNA record is a component of that record's overflow, read thickness can be utilized to gauge record and quality articulation with similar or better precision than articulation microarrays. Processing data and performing differential analysis can be overwhelming for scientists with restricted insight in programming. We want to assist specialists with figuring out how to break down mass RNA-sequencing information by giving code, clarifications, and the normal result for each progression of the cycle.

Finding and downloading data from GEO utilizing NCBI SRA apparatuses and Python
Planning FASTQ documents utilizing STAR

Performing differential analysis can be overwhelming for scientists with restricted insight in programming. I want to assist specialists with figuring out how to examine mass RNA-sequencing information by giving code, clarifications, and the normal result for each progression of the cycle. Here, I present an illustration of a total mass RNA-sequencing pipeline which incorporates:

Finding and downloading crude information from GEO utilizing NCBI SRA apparatuses and Python

1.Mapping FASTQ documents utilizing STAR

2.Differential analysis using DESeq2

3.Visualizations

4.Representations for mass RNA-seq results.

Snakemake is a Bioinformatics instrument for dealing with a work process. This instrument demonstrates significant while dissecting a lot of information with numerous devices. This content was made as a learning instrument for work process director. There is likewise Nextflow to oversee huge examination work processes. Here, Snakemake was utilized to run all that is generally run-on Linux with RNA-Seq Examinations

Method

- The packages I am using are as follows
- FastQC
- Trimmomatic 0.39
- STAR
- MultiQC
- featureCount
- Samtools
- Bam2Fastx
- MultiQC

Downloading the Raw Data.

1. The sequences from the sequence read archive (SRA), to download the sequences the SRA toolkit was used. Then we use the split files command to split the forward read

from the reverse read for paired end read trimming through Trimmomatic. Trimmomatic is quick, multithreaded tool that can be utilized to trim and edit Illumina (FASTQ) information as well as to eliminate adaptors.

2. After splitting the files, it gives 2 per SRR, one is forward and other one is reverse. The files will be named as _1 and _2 ending with.fastq.gz.

FastQC (conda install -c bioconda fastqc)

1 For the quality control of raw sequencing data FastQC is used. The output of FastQC is in a zip folder and an HTML file connected to it. The figure below shows the example of HTML file.

2 The image shows how the HTML looks like and ever category has a graph attached under it which shows the read quality of the sequence. These reports give insights about the sequences i.e., Bad or Good.

MultiQC (*conda install -c bioconda multiqc*)

1. We run MultiQC with the folder which has QC files . After running MultiQC there will be a new folder namely multiqc data where the summaries are stored.
2. Trimmomatic. (conda install -c bioconda trimmomatic)
3. When using trimmomactic to trim the paired-end reads with the adapter TruSeq3-PE-2. We will remove first and last 3 bases. Trimmed reads can be examined once more with FastQC to perceive how well the managing attempted to improve the document quality. Subsequent to running FastQC on the managed records we see that the nature of those that were downright horrendous quality were moved along.

STAR conda install (*conda install -c bioconda star*).

1. To run STAR. I will utilize a similar genome and GTF record as recently downloaded from NCBI. With the record documents made, we can begin lining up with STAR. It's significant here than we just pick the matched end peruses and not utilize the peruses in general.
2. In a FASTQ record, each read should be "planned" against a reference genome. Finding the region in the reference genome that best matches the fragment of the mRNA record caught in the read might be described as "planning."
3. To undertake bulk RNA sequencing, you must first prepare a "library" from the RNA of your target cells. To create FASTQ records, this library is sequenced. To create a library, mRNA from the cells is first captured and then converted to cDNA using a switch. To build the library, this is then broken into little pieces and sequencing connectors are added to the ends of each portion. These components will be "read" by a sequencer, which will save the groups in a FASTQ document. We can see how the groupings resemble each other.

4. After these have been downloaded and unzipped, we may create the genome list record by following the steps below.

Differentially Expressed Sequence Identification

1. In order to perform differential analysis, we will utilise R and the DESeq2 package. This package contains a set of data standards and handling tools designed specifically for mass RNA-seq data and differential expression analysis..

The packages we have used here are:

- readr (*install.packages('readr')*)
- limma (*BiocManager::install('limma')*)
- DESeq2 (*BiocManager::install('DESeq2')*)
- dplyr (*install.packages("dplyr")*)
- ggplot2 (*install.packages("ggplot2")*)
- gplots (*install.packages("gplots")*)
- gage (*BiocManager::install('gage')*)
- gageData (*BiocManager::install('gageData')*)

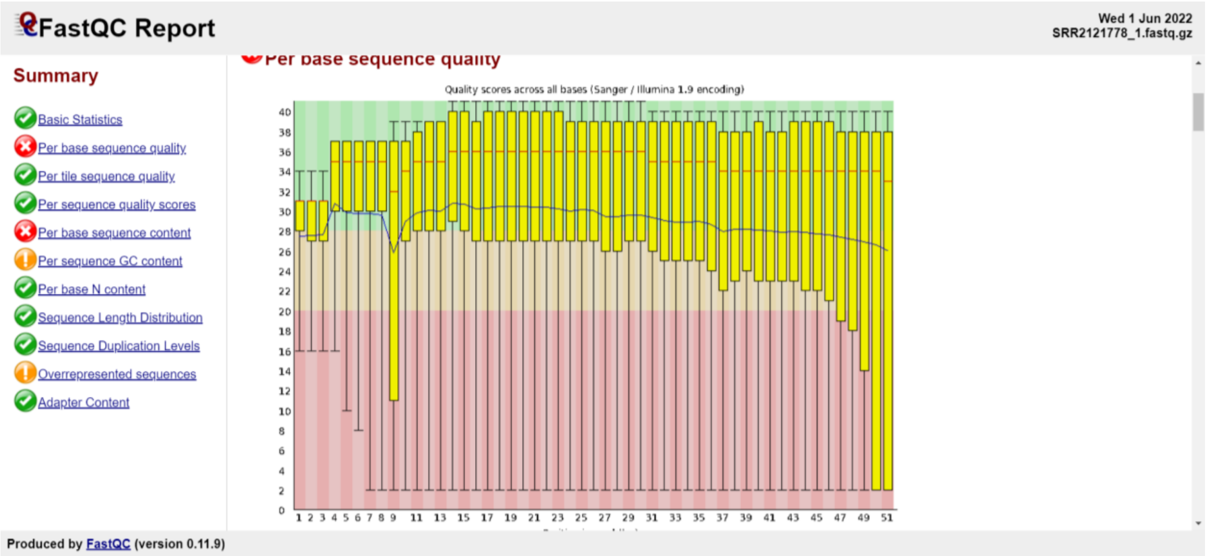
2. First, we will import the file into R. This will give us the data about the file. Then we will do the data exploration which will tell us about the data. We will analyze the data.
3. To make the DESeq2 object, we can utilize the capacity DESeqDataSetFromMatrix(), which requires three things:
4. countData - The information as a whole number lattice, where the rownames relate to the quality IDs. This is the information object that we made from raw_counts.csv.
5. colData - A table that gives the example gatherings to be looked at. It ought to likewise relegate each example to a gathering (for example natural imitas).
6. design - An equation that directs how the counts for every quality rely upon the factors in colData. We will utilize the gatherings characterized in colData.

Visualizations

1. For visualizations we use PCA plot which is a multivariate dataset, we will create a heatmap, MA plot.
2. A foremost parts plot is one more method for seeing how different the examples are. As usual, the examples which are generally comparable with one another as far as their quality articulation values will be nearer to one another on the plot. This plot utilizes a Cartesian direction framework, and the tomahawks that are shown relate to the main two head parts that make sense of most of the fluctuation in the information

Results and Discussion

1. The crude peruses we have here completely passed for connector content and arrangement length appropriation and everything fizzled per base succession content. SRR2121770, SRR2121771, SRR2121774, SRR2121775, SRR2121788, SRR21217812.2, and SRR2121789.1 were genuinely good quality peruses. SRR2121778, SRR2121779, SRR2121780, SRR2121786, SRR2121787, SRR2121781-1, and SRR2121789.2 were of reasonably lower quality (bombing at least 3 in the two peruses). Every one of them fizzled both per base arrangement quality and per tile grouping quality.



FastQC Report

Wed 1 Jun 2022
SRR2121778_1.fastq.gz

Summary

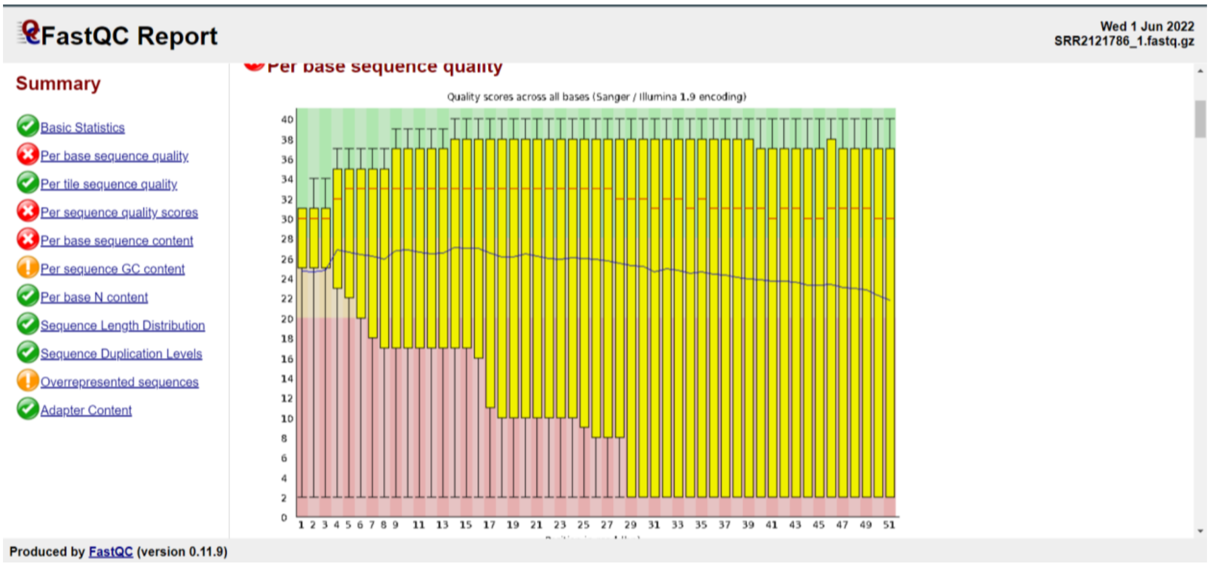
- ✓ Basic Statistics
- ✗ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ! Overrepresented sequences
- ✓ Adapter Content

Per base sequence quality

Produced by FastQC (version 0.11.9)

Figure shows the quality of SRR2121778.

The largest number of peruses dropped was from managing SRR2121786, where 20.55% dropped. Most peruses were somewhere in the range of 5% and 10% dropped. SRR2121786, SRR2121787, and SRR2121779 had grouping drops more noteworthy than 15%.



FastQC Report

Wed 1 Jun 2022
SRR2121786_1.fastq.gz

Summary

- ✓ Basic Statistics
- ✗ Per base sequence quality
- ✓ Per tile sequence quality
- ✗ Per sequence quality scores
- ✗ Per base sequence content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ! Overrepresented sequences
- ✓ Adapter Content

Per base sequence quality

Produced by FastQC (version 0.11.9)

Figure shows the quality of SRR2121786.

Visualization

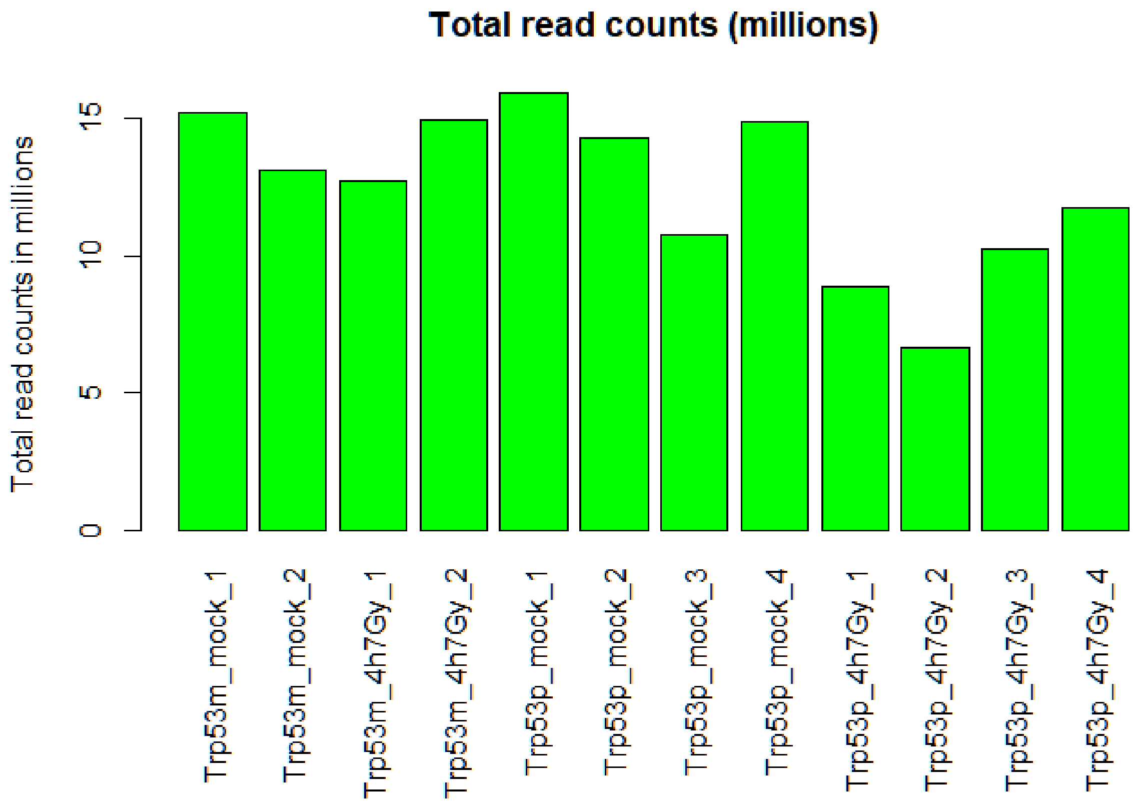
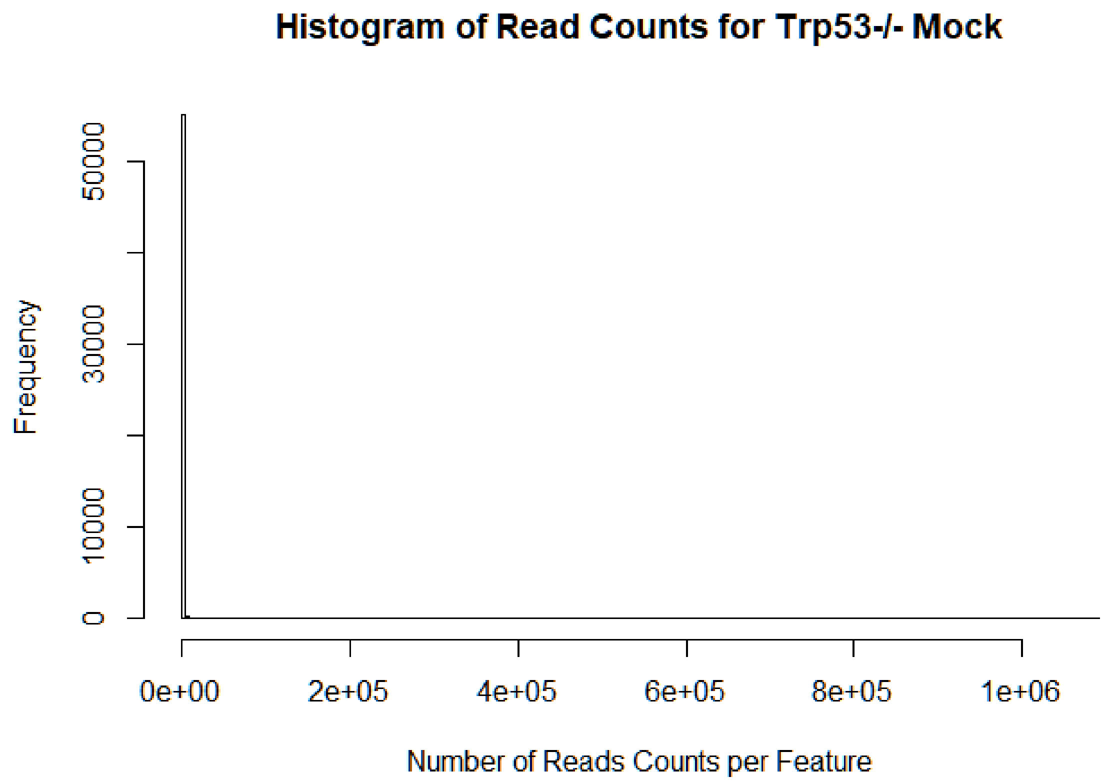


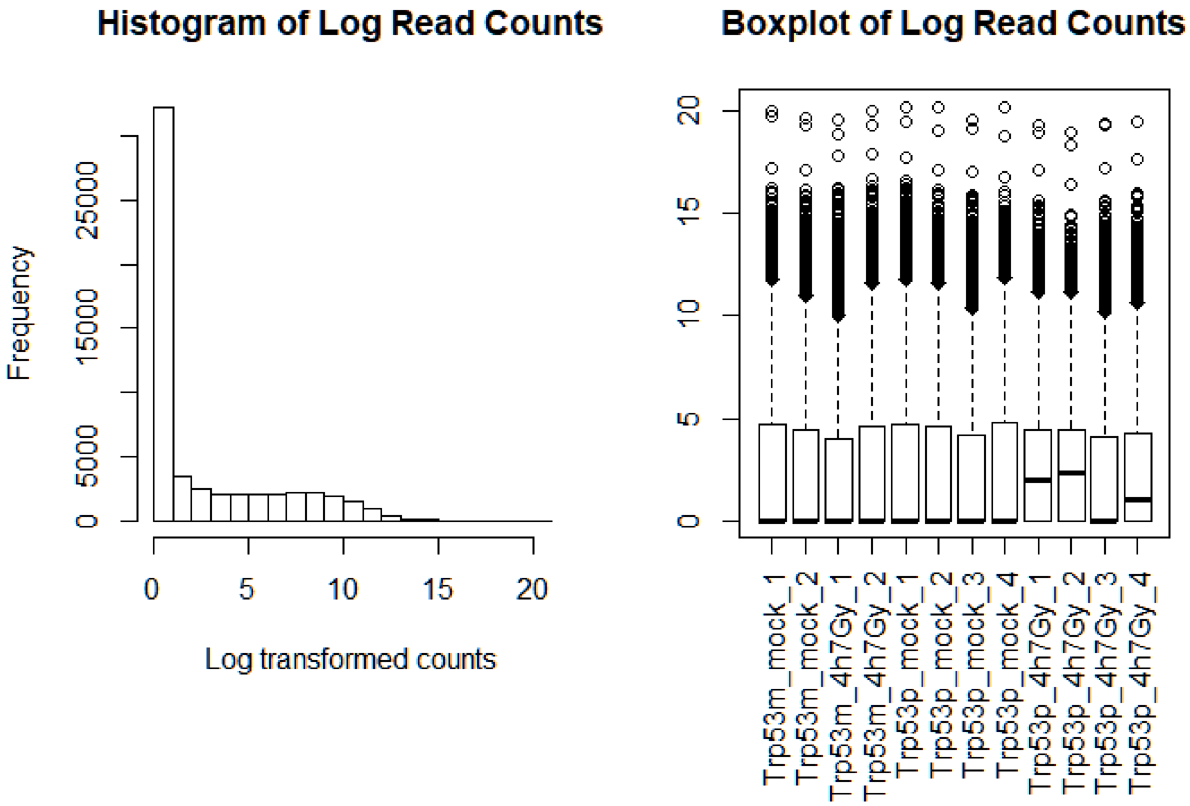
Figure 1: Shows the total read counts in (millions)

2. So as we can see Trp53p_mock_1 has more than 15 million reads in total meanwhile Trp53p_4h7G4_2 has the lowest in total reads which is lower than 10 million.

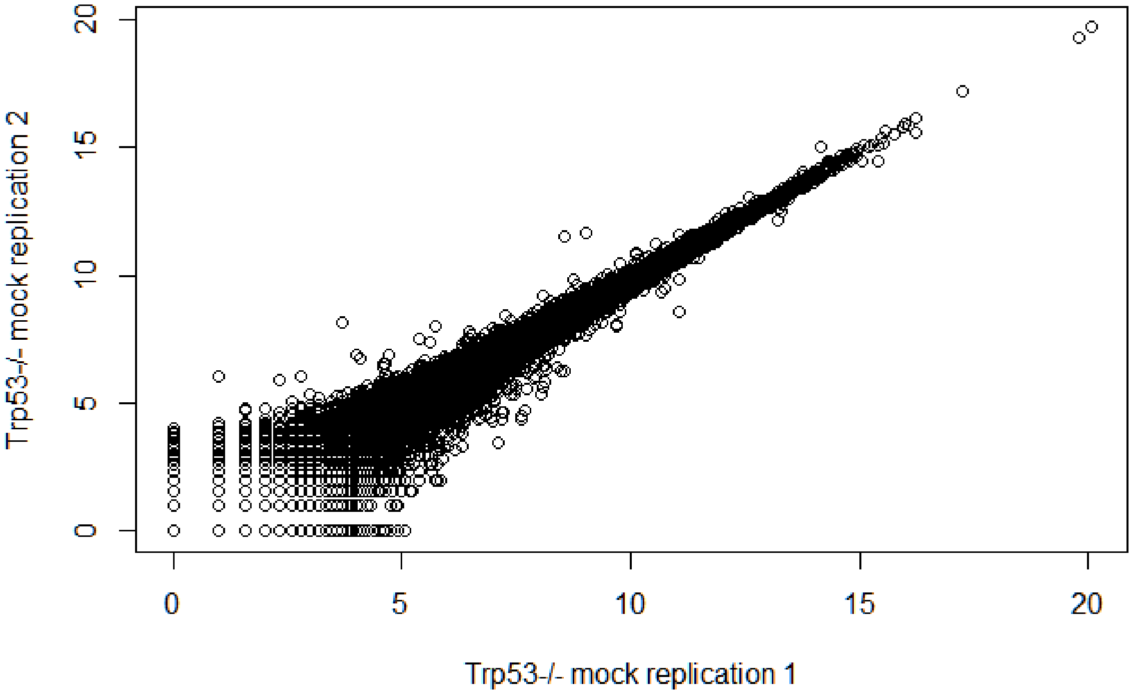
3. To visualize the number of 0's in our data we plot the histogram for Tro53m_mock_1



So we can see, most of them are staying around 0 mark. However, not all the samples are actually 0 by suing the log function we can easily skew the data like this.



Now we are trying to replicate the data between two columns. Coheeration is pretty strong between logCountData[,1] and logCountData[,2]



Trp53p_mock replication 1 and Trp53p_mock replication 2 are more similar to each other.

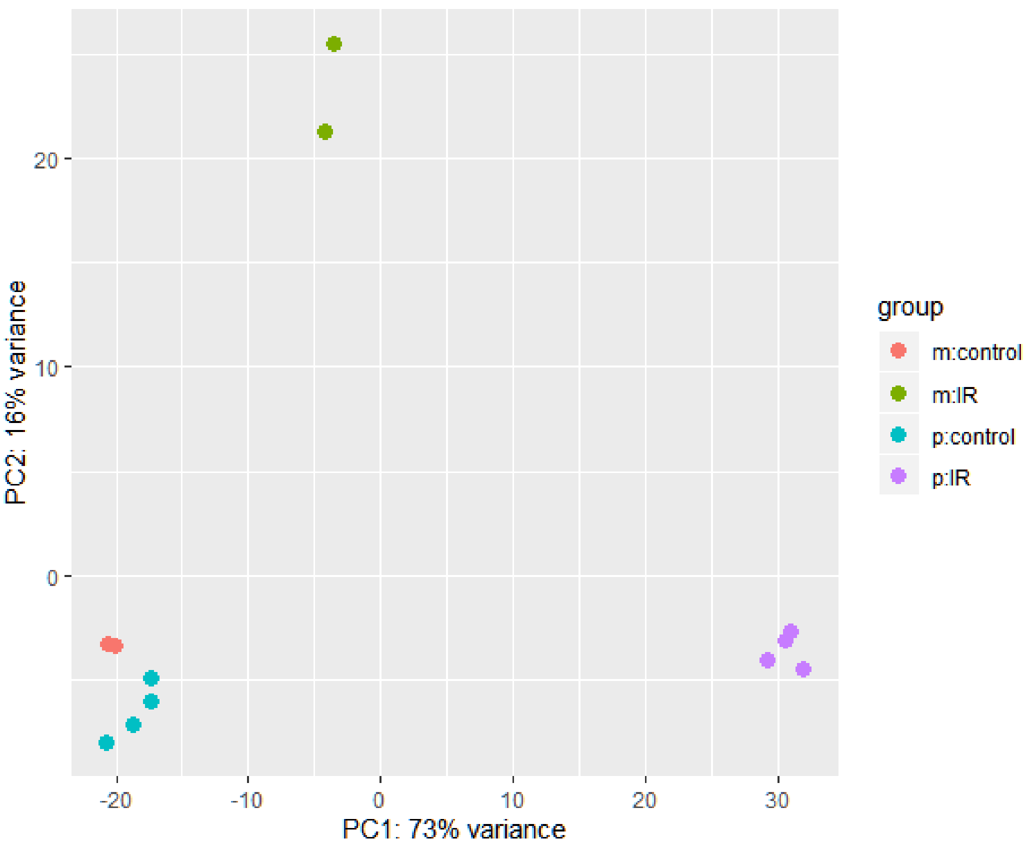
DESeq2 objects

1. Because the distinction behind the scenes of the cell lines accounts for a lot of the variation in the data, we should create separate DESeq2 objects for the LNcaP and PC3 tests. It will use grep() to extract segments from the raw data that meet the gathering assignments, create the colData framework based on your gathering assignments, then generate the DESeq2 object using DESeqDataSetFromMatrix() and DESeqDataSetFromMatrix() ().

1. The capacity ought to print the colData that was utilized to produce each DESeq2 object, so we can twofold check to ensure that the fitting example sections were chosen and that the gathering was right.

2. we can begin to evaluate test to-test changeability. This is significant for deciding if you have anomalies in your dataset, as well as whether your information seems OK. For instance, the quality articulation examples of natural/specialized repeats ought to look genuinely like each other. In this part, we will cover a few perceptions that can be utilized for this reason. To utilize these capacities, we initially play out a fluctuation settling change.

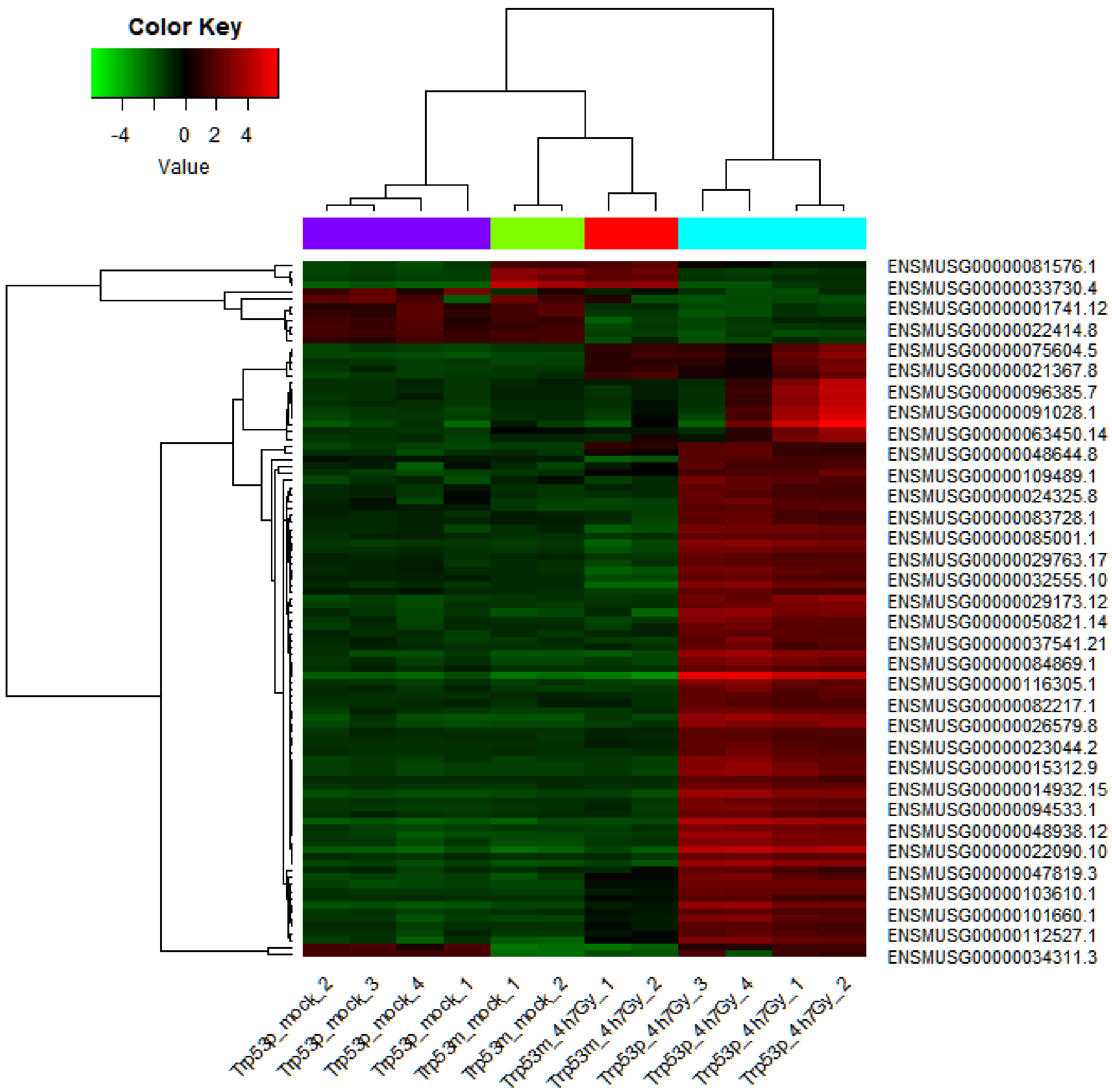
PCA plot.



1. PC1 is on the minus side vs p=PC2 is on the positive side. On the left sides are control. There are no irradiated. So this the combination of p53 and the treatment which are merged together. P53 is on the other side of control which is eventually pushing it far. Therefore, a lot genes are based on the impacts of p53.

Variable Genes Heatmap

We can recognize the qualities that are driving the grouping among tests and show them in a heatmap. To do as such, we initially acquire the top most factor qualities across tests by choosing the qualities with the most elevated esteem. Then, at that point, we plug the standardized counts for these qualities into a heatmap . The heatmap will group the examples in light of articulation similitude as well as show the qualities that are related with each bunch on the right-hand side. Note that the qualities are just picked in light of fluctuation across tests. No factual tests are performed to decide if they are quite unique between gatherings.



The top 100 genes are shown below. Trp53m 4h7Gy 2 emits a lot of radiation, which causes gene expression differences, whereas Tro53p mock 3 emits very little radiation. The grouping looks a lot like what we saw in the distance map. It's hardly unexpected that the grouping is so well defined, given that we're working with data from cell lines. Knowing which genes define sample differences is useful for determining whether sample preparation concerns such as low cell purity or tissue contamination occurred..

Refrences

https://anaconda.org/

https://www.ncbi.nlm.nih.gov/

https://web.expasy.org/cgi-bin/cellosaurus/search

2022. [online] Available at: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0190152#sec006> [Accessed 2 June 2022].

(Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks | Trapnell, Cole; Roberts, Adam; Goff, Loyal; Pertea, Geo; Kim, Daehwan; Kelley, David R; Pimentel, Harold; Salzberg, Steven L; Rinn, John L; Pachter, Lior | download, 2022)

Booksc.org. 2022. *A comparison of methods for differential expression analysis of RNA-seq data | Charlotte Soneson, Mauro Delorenzi | download*. [online] Available at: <https://booksc.org/book/21944662/8b8dc5> [Accessed 2 June 2022].

Tarek Khorshed, Mohamed N. Moustafa, Ahmed Rafea, "Multi-Tissue Cancer Classification of Gene Expressions using Deep Learning", *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp.128-135, 2020.

Hyunjin Park, Seungyeoun Lee, Ye Jin Kim, Myung-Sook Choi and Taesung Park, "Multivariate approach to the analysis of correlated RNA-seq data," 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2016, pp. 1783-1786, doi: 10.1109/BIBM.2016.7822789.