CS310: Computer Science Project

Analysis of Ovarian Cancer Single-Cell RNA-seq Data

Student: Ava Spataru
Project Supervisor: Dr. Sascha Ott
Project Advisor: Annika Stechemesser

Department of Computer Science
University of Warwick

Abstract

One in five cancer patients undergo the long process of chemotherapy without any positive results. This is due to uncertainty around the nature of cancer and its numerous patient-specific mutations. Improving treatment for one patient would mean tailoring it to the genetic composition of their tumour. To understand this genetic composition of the tumour, the cells inside it can be grouped together into types, based on the genes that are highly expressed in each. This separation of tumour cells into types is the first step towards patient-specific treatment.

Recently developed RNA-seq techniques allow scientists to analyse genetic information from large datasets of single-cell samples. One of these techniques, Drop-seq, was used to extract the data used in this project. This data consists of ovarian cancer cells, which were analysed with the aim of classifying cells and tracking the changes that occur on cell-types after treatment. Being able to track the influence of chemotherapy drugs on specific cell-types will mean assessing their effect and therefore determining the most efficient drug for individual patients.

Despite the complexity of the task, the project achieved its purpose, by making use of information about gene expression within single cells from one patient's tumour, before and after treatment. The analysis done has successfully identified 6 cell-types before treatment and 4 cell-types after. A new mathematical formula was developed to facilitate the mapping of cell-types across datasets. This has shown much more conclusive results than any of the currently existing methods. It is now possible to identify and understand similarities between clusters in a much faster and precise way.

Using this newly developed method, the cancer stem cells were identified even after treatment, showing that their representation in the tumour has grown significantly. It is known that cancer stem cells survive chemotherapy, but since they go through significant genetic changes during treatment, it is often hard to clearly identify them. This project has created methods and tools to help gain a deeper understanding of the effects that chemotherapy has on tumours.

Keywords: ovarian cancer, chemotherapy, RNA-seq, gene expression, cell classification, tumour progression

Acknowledgements

My supervisor, dr. Sascha Ott, has guided me at every step of this project. He has gone to great lengths of selfless involvement, always answering my simple questions and reviewing my work. Under his mentorship, I have gained a considerable amount of novel knowledge from fields of biology, statistics and data science. I could never thank him enough.

I owe a large debt of gratitude to Annika, whose numerous advice was always insightful and helped me understand critical parts of the project. I have been extremely lucky to have an advisor who was so involved and who responded so promptly to all my questions.

I would like to thank to dr. Nigel Dyer, who has so kindly provided me with crucial software and much-appreciated support in setting it up.

I thank the Ovacome team at the University of Birmingham for providing the data and especially Dr. Rachel Pounds for meeting with me to discuss my findings.

Lastly, I would like to thank my family and peers for their endless support and valuable advice. I am forever grateful.

Contents

1 Introduction

According to Cancer Research UK [1] 7,270 women are diagnosed with ovarian cancer each year. Around 54% of them undergo chemotherapy as part of their treatment. Unfortunately, in some cases the chemotherapy has no positive effects on the tumour. The causes of this are unknown at the moment, since patients with the same diagnosis might have different reactions to the same drugs. An extensive amount of research is now conducted around this area, to determine the best treatment specific to individual patients. Precision medicine is a new method that allows doctors to determine the most efficient treatment for a patient, based on the genetic understanding of their disease [6]. This methods implies that some day, doctors will be able to extract data from patient tumours, look at the *composition* of the tumour and assign a personalised combination of drugs, which will give the best results. The first step towards personalised treatment is to analyse this *composition* of tumours and create methods for assessing the efficiency of chemotherapy drugs on different tumours or parts of tumours.

This project makes use of two datasets coming from the same patient, one showing genetic information about the tumour before treatment and one after treatment. The aim of the project was to classify cells in each dataset based on their genetic expression and develop methods for assessing the effect of chemotherapy on each class. Under the assumption that patients with the same classes (types) of cells will react similarly to the same drugs, the methods developed in this project become crucial to predicting the outcome of chemotherapy drugs on different patients.

The two datasets consist of single-cell RNA-seq data, which contains information about genetic information captured for individual cells. This project employs machine learning algorithms for the classification of cells into types and analyses the classifications obtained using mathematical and statistical methods. Along with the classification, the project resulted in the development of two tools that improve the overall analysis pipeline. These tools assist with determining the effectiveness of chemotherapy treatments on specific types and allow for investigation of genetic mutations occurring in individual types. Exploring genetic mutations has helped assess the effect of chemotherapy on the identified cell-types and further validate the classifications made.

This report details the analysis done, highlights the main results and talks about why and how the tools were developed. It begins with some essential background information and details on the development of the tools used to improve the overall analysis pipeline. These two sections together also provide a good understanding of how the datasets were obtained and how the analysis was conducted. Section 4 gives a brief overview of the datasets used with a discussion of their relevance to the project's aim. The report then has three main parts: section 5, which presents the classifications; section 6, which explains the methods used for assessing the effect of chemotherapy and shows an understanding of the influence it had on the presented datasets; section 7, which investigates gene mutations. The report concludes with the main results of the project and a discussion on the impact they will have into the future.

This project has made it possible to begin understanding how the cancer stem cells survive chemotherapy and why in so many cases, the cancer relapses. In the particular case of the patient data analysed in this project, the cancer stem cells could clearly be identified in the dataset before the treatment, but not afterwards. This might be because of the many genetic changes that occur on the tumour cells during chemotherapy. The method designed as part of this project has helped not only identify the cancer stem cells in the post treatment dataset, but also observe that these cells account for a significantly larger part of the tumour after the treatment than they did before. Overall, this project has designed new methods to create mappings in cell-types across different datasets and assess the effectiveness of treatment on each cell-type.

2 Background information

In every data analysis project, such as this one, understanding the data to be analysed is very important. This can help give meaning to the numbers and classifications. Understanding how the data is collected helps identify possible errors and supports the data filtering stage. Analysis methods should be carefully selected and tailored to the data. Researching the existing software helps the analysis process, by providing fast computational methods already implemented, which can be used instead of re-developing.

This section will give details about data provenience, data extraction method, data analysis methods and existing software. All these factors have influenced the direction of the project and are relevant to understanding the key results.

2.1 Cancer research

Cancer is one of the leading causes of death, being responsible for about one in six deaths globally [2]. Cancer is not a single disease, but is a group of diseases, that can affect various parts of the human body. The main aspect of cancer is that abnormal cells are created at a very fast pace and they grow outside their normal boundaries, spreading to different organs and invading the body [2].

In order to efficiently treat the cancer, it is important that the diagnosis made is correct. The diagnosis refers to the type of cancer and the stage. Some cancers, such as breast, cervical and oral cancer, if detected early have a high cure rate [3]. There are a couple of treatments for cancer and most people are prescribed a combination of them. The most common treatments targeting cancer are surgery, chemotherapy and radiation therapy, but recently there have been many advances is immunotherapy, stem cell transplants and precision medicine [4]. Chemotherapy is a drug-based treatment that aims to kill the cancer cells. There are more than 100 types of chemotherapy drugs and new ones are being developed constantly [5]. According to Cancer Research UK, these chemotherapy drugs are assigned to patients based on the area of the body where the cancer first appeared, the stage of the disease, the patient's general health and how the cancer cells look under the microscope. However, different people respond differently to these drugs, despite having the same general type of cancer, in the same stage. Often patients go through chemotherapy with no positive effects on the tumour. This should be avoided, mainly since the chemotherapy is a very painful and long process, which can have various side-effects on the patient. Unfortunately, at the moment there is no definite way of predicting what will happen with a specific patient, under a specific treatment.

This project is only a small part of the wide research area around cancer, which focuses on assessing the effect on chemotherapy with regards to genetic information of an ovarian cancer tumour. It can be considered as a step towards advancements in precision medicine (treatments tailored to genetic changes in individual tumours) [6].

2.2 Ovarian Cancer

Ovarian cancer is a type of cancer which manifests in the ovary. The cells begin to grow and divide in an abnormal uncontrollable manner, which results in a tumour. There are 4 stages of ovarian cancer, based on the size and spread of the tumour. The stage of the cancer upon detection has a very high impact on the effectiveness of treatment. Out of 100 women, if the cancer is discovered in the 1st stage, then around 90 will survive. However, if the cancer is discovered in the 4th stage, then only 5 will survive [7]. Figure 1 shows the rate of survival for 5 years of patients diagnosed with ovarian cancer, including those who have been cured.

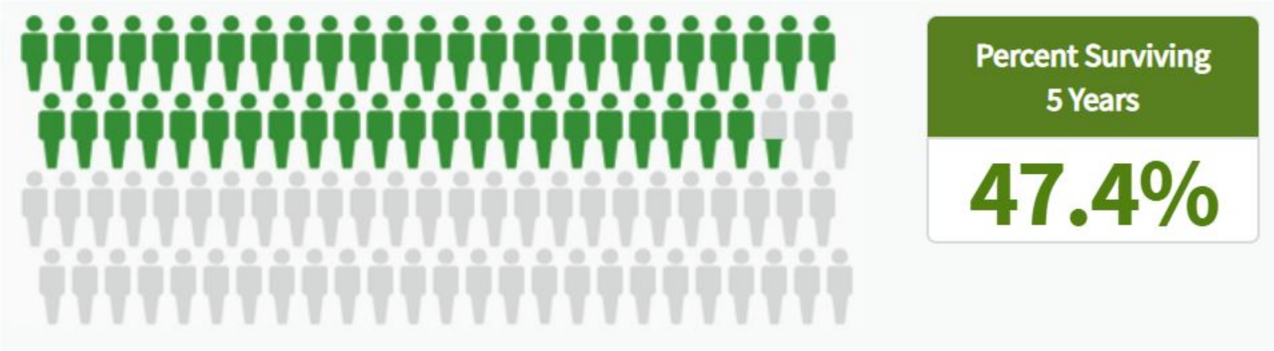


Figure 1: Based on data from SEER 18 2008-2014. Grey figures represent those who have died of ovarian cancer. Green figures represent those who have survived 5 years of more [8].

There have been many efforts to identify efficient treatments specific to ovarian cancer, mostly by attempting to specialize the cancer on various types. There are four main types of ovarian cancer, epithelial ovarian cancer being the most common [9]. Primary peritoneal cancer and fallopian tube cancer are less common and there is much fewer information around what possible causes are or how it can be treated. According to Cancer Research UK [10] these are treated in the same way with ovarian cancer since they are *similar*.

2.3 General Biology of Cells

It has been mentioned above that cancer manifests as a number of abnormal cells, which grow outside their normal boundaries. The mass that they form is called a tumour and it is what the cancer "looks like" within the body. The tumours differ in size, but on average, in a $1cm^3$ tumour, there are around one billion cancer cells [11]. Some of these billion cancer cells will perform different roles to the organism. Perhaps the most important cancer cells to be targeted by treatments are the cancer stem cells. These cells have characteristics similar to normal stem cells and have shown the ability to give rise to more cancer cells, they are tumour-forming cells, which have a critical role to the growth of the tumour. According to the Ovarian Cancer Research Alliance, about 70% of patients diagnosed with ovarian cancer, will have a recurrence after treatment [12]. The reason behind this is that some of the cancer stem cells survive chemotherapy and are able to form a tumour once again.

In order to separate the tumour cells into types (potentially identify a type corresponding to the cancer stem cells) and determine their roles for the tumour, the *composition* of each cell must be investigated. To understand what is meant by *composition* of a cell, it is worth looking at the structure of a single cell as shown in figure 2.

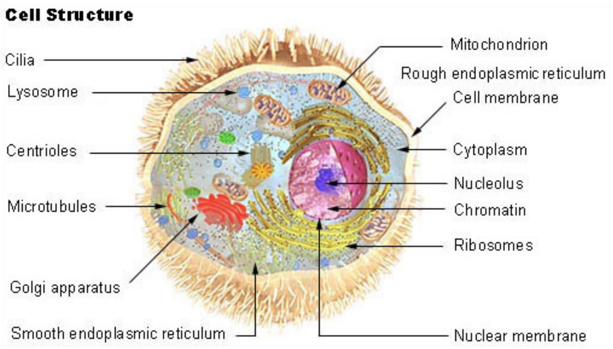


Figure 2: Diagram showing the structure of a single cell [13].

The most important components of the cell are the membrane, the nucleus and the cytoplasm. The membrane separates the material inside the cell from the material outside and controls the passage of materials from the cell to the rest of the organism. The nucleus is the control center of the cell and determines how the cell will function, containing the genetic material of the cell - deoxyribonucleic acid (DNA). The cytoplasm is a fluid inside the cell which facilitates the transport of genetic materials, by diffusion [13].

To understand the "composition" of a cell, it is necessary to understand what the nucleus is made up of. The nucleus contains chromosomes, for a human cell there are 23 pairs of chromosomes. One chromosome is simply a long strain of DNA. The DNA is comprised of molecules, called nucleotides. Each nucleotide contains one of four nitrogen bases: A (adenine), T (thymine), G (guanine) and C (cytosine). One stretch (sequence) of DNA forms a gene and each gene represents information for the body to know what protein to generate. The combination of proteins found in a cell determines its function. Boiling it down to general terms, the sequence of adenine, thymine, guanine and cytosine represents a gene, which encodes a protein and the combination of proteins found inside of a cell determines the *composition* of the cell [14]. This will be used for analysis throughout the project as a way of identifying a specific cell with a cell type.

The DNA cannot leave the nucleus, so in order to communicate genetic information to the outside of the nucleus, a copy of it must be sent. Transcription is the process of copying genetic information from DNA into mRNA (messenger Ribonucleic acid) as shown in figure 3. After transcription, the mRNA can leave the nucleus through the cytoplasm and membrane, carrying information outside the nucleus [15].

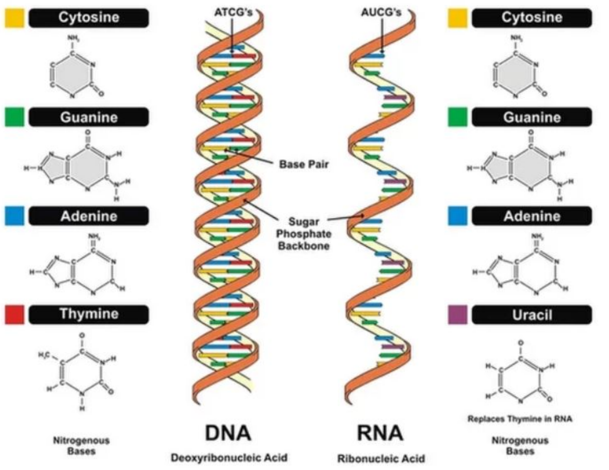


Figure 3: The structure of DNA and RNA. Both are comprised of genetic information [15].

The mRNA can be intercepted outside the cell nucleus and sequenced to be ready for analysis. In order to capture this genetic information, various technologies were created. The next section describes Drop-seq, which is the method used for extraction of the datasets used as part of this project.

2.4 Drop-seq

Drop-seq is an open-source technology that was developed in 2015 by the McCaroll Lab, allowing scientists to capture genome-wide RNA expression in thousands of individual cells at once. The technique was first described in a paper [16] published in *Cell*.

The main process consists of separating the cells in tiny droplets (nanoliter scale sized) to allow for parallel analysis. Afterwards, the mRNA of each cell is intercepted and linked with an uniquely barcoded bead, mapping it to the cell of provenience. The Drop-seq process can be separated into different stages: (1) generation of barcoded beads, (2) co-encapsulating cells with beads, (3) sequencing and analysis of the cells in one reaction and (4) creating the digital format of the data.

In order to generate thousands of uniquely barcoded beads, a split-pool synthesis approach was used. A pool of millions of microparticles is iteratively divided into four equally-sized groups, one corresponding to each DNA base (A, C, G, T). The DNA base corresponding to each group is then added to the microparticles inside. This process is repeated by re-pooling for 12 rounds, thus obtaining 4^{12} possible barcodes of length 12. Primers on any microparticle will have the same barcode, but different microparticles will have different barcodes.

For encapsulating cells with beads a micro-fluidic device was designed, which combined two flows of liquid, one containing the beads and one containing the cells. The device will generate some droplets with no beads and some with no cells. For the ones that contain both, STAMPS (Single-cell Transcriptomes Attached to Micro-Particles) are generated.

In the stage of sequencing and analysis of these STAMPs, the droplets are broken and the mRNA-bound microparticles are collected. These are then reverse transcribed, the resulting molecules then being sequenced from both ends. Unique molecular identifiers are used to avoid counting the same RNA sequence twice.

The output from this process is as shown in figure 4. Each read consists of a cell barcode, a unique molecular identifier and a sequence of cDNA (from the reverse transcribed RNA sequence). Using the barcodes, each read can be assigned to a cell. Rearranging the reads by the cell they belong to and then aligning each sequence to the genome, results in a BAM file. Counting the number of unique molecular identifiers corresponding to each cell-gene pair gives an idea of how expressed a gene is in a certain cell. The matrix from figure 4 is known as a DGE matrix and together with the BAM files it composes the datasets used in this project. The details of what each file looks like and how they were used in analysis are described in the section below, titled "Project Datasets".

Even though Drop-seq has been proven successful in the experiments described in the paper [16], there are a number of potential issues and challenges with the extraction of cells. Some of these challenges are cell duplets (cells that pair together with the same bead), single-cell impurity (mostly cells which have been broken during preparation) and sequencing errors, which inflate the number of unique molecular identifiers. These possible errors create noise and outliers in the data and therefore should be ignored during the analysis. This has been taken into consideration and explained in further sections of this report.

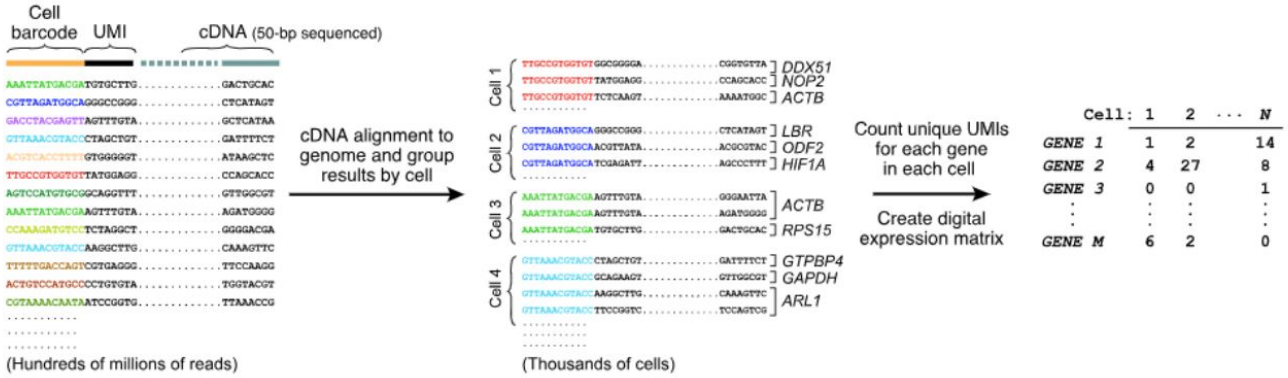


Figure 4: The pipeline for formatting the output of the Drop-seq technology, starting from individual reads of cDNA (obtained by reverse transcription of RNA) to a DGE matrix (showing expression of genes in individual cells). Figure from [16].

2.5 Analysis methods

The data analysed in this project is very large, containing information about over 20,000 genes in 3,000 cells. It is almost impossible for humans to analyse this data by hand and therefore computational methods are needed to pick up on certain trends in the data. Machine Learning is a very fast growing field of research that consists of methods for statistically modelling datasets, with the idea of creating software which can *understand* data better than humans can.

Classification is a known machine learning problem which consists of taking data samples with multiple attributes and assigning classes to each sample, according to the list of attributes. The machine needs to pick up on trends in the data and separate the samples into groups/ classes of similar elements with respect to those trends. In the case of this project, each cell represents a sample and each gene represents an attribute. As described in the following subsections, PCA (Principal Component Analysis) was used to reduce the dimensionality of the data and t-SNE (t-Distributed Stochastic Neighbouring Embedding) was used to perform the actual classification.

The learning performed in this project is unsupervised. This means that for a sample, the true class isn't known and therefore there is no way of telling if a classification given by the algorithm is correct. There is also no prior knowledge of what the classes should be or how many classes should be in one dataset. To overcome this issue, two distinct algorithms (t-SNE and Clustering) were used for classification. If the resulting classes obtained by each algorithm were similar, then there is some confidence that the classification is correct.

The following subsection describe the algorithms used as part of this project. Understanding how they work was essential for applying them to the project datasets, because it aided with parameter tweaking and interpretation of results.

2.5.1 PCA (Principal Component Analysis)

As explained in the above sections, the classification of cells into types will be done using a DGE matrix. This matrix contains each cell as a column and each gene as a row, giving a value of "expression" for each gene in each cell, as an entry of the matrix. This data can be viewed as a set of cells, for which we have multiple points of information, attributes (the expressions of all genes). Since there are over 20,000 genes in each dataset, there will be over 20,000 attributes for each cell. One way of classifying the cells would be to plot all of the attributes in a graph and observe which cells are nearby the others. However, this is not feasible with the number of attributes in this project. Plotting each gene would mean plotting a graph of over 20,000 dimensions and this is not only impossible to do, but also impossible to interpret. Since the dataset is highly dimensional, it needs to be reduced, for easier interpretation, but without losing any relevant trends in the data.

Principal Component Analysis (PCA) is a statistical method for reducing data that consists of values for linearly uncorrelated variables and a high number of dimensions to a lower dimensional space while preserving as much of the variance in the data as possible. PCA takes as input data points with a very large number of attributes and reduces it to a smaller number of attributes which are *metafeatures* that combine information across a correlated set of initial attributes. These metafeatures are called principal components and their value summarizes the values across a subset of initial attributes, which are correlated. Two attributes are correlated when the value of one is dependent upon the value of the other one. A set of attributes S is correlated if the value of each attribute in S depends on the values of a subset of S. Assuming that the metafeatures have been determined, these then can be used as dimensions for plotting the data and therefore the data has

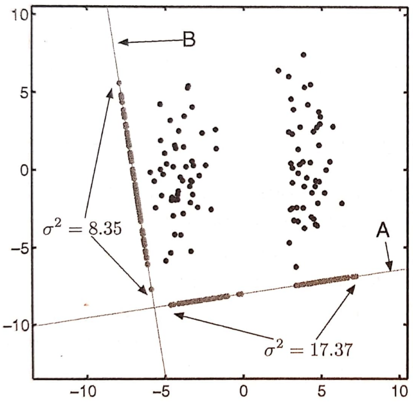


Figure 5: Example of two projections (A,B) of the same data points with different variance. Data is sampled from two Gaussians [17].

been projected into a lower dimensional space.

To determine which dimensions to use, it is important to maintain as many trends in the data as possible. These dimensions should have a form of measurement to decide which one "says more" about the data. PCA uses variance as a proxy for this, assuming that dimensions with more variance maintain more trends from the initial data than dimensions with less variance. Figure 5 shows how for the same data (sampled from two different Gaussians) the projection with more variance (A) represents the trend in data - that it is a cluster structure - better than the projection with lower variance (B). In this case, projection A "says more" about the data and therefore would be a better dimension to use than projection B.

PCA uses variance in the projection space to determine which projections to choose as principal components. If the reduction is performed from M to D dimensions, PCA defined D vectors, each of them M-dimensional. These vectors intuitively determine the weights of each existing feature in the metafeature. These vectors are determined in a decreasing order of variance, i.e. the vector corresponding to the first dimension will give the highest variance in the data and the D^{th} vector will give the lowest variance in the data. Another property of these vectors is that each vector i is orthogonal to all vectors from 1 to (i-1). Due to this property, the set of vectors is linearly independent and therefore the new dimensions are not correlated to each other, therefore creating a set of relevant features, with no redundancy.

The learning part of the problem becomes choosing the number of dimensions to which the feature space should be reduced. To determine the number of principal components, the different variance scores were investigated, to only include principal components which have a statistically significant variance, as compared to the other components.

As tailored to this project, the new dimensions will represent values that summarize certain subsets of genes. For example, genes that are expressed in cells which are going through mitosis will have correlated expression. Hence it is worth summarizing the set of such genes as one single metafeature in the new dimensional space. This is the intuition behind dimensionality reductions, however it is not entirely sure if this is exactly what happens when applying PCA. It is almost impossible to check and give meaning to the very numerous values behind the weights in each dimension. Throughout this project, PCA was used before applying any classification algorithms to ensure that the classification can identify significant trends, rather than focus on values of individual genes.

2.5.2 t-SNE (t-distributed Stochastic Neighbouring Embedding)

After the dimensions of the data are reduced using PCA, it is still hard to visualise the data and identify separated classes. The hope after reducing the dimensions would be to plot the data in the D dimensions and clearly identify clusters of data points. These clusters would be the different classes, because if the points are clustered together, then they have similar values for the D dimensions and can be considered similar (belonging to the same class). However, it is really hard for human minds to plot and understand and correctly interpret more than 3 dimensions. For this reason, the classification algorithm explained in this section, plots the points in a 2-dimensional space, while considering all D dimensions of the input data.

T-Distributed Stochastic Neighbouring Embedding is a technique for visualising highly dimensional data in 2 dimensions, with the crucial property that it preserves clusters. Any points that were clustered in D dimensions, will be clustered in the 2 dimensional plot outputted by the algorithm. At each iteration of the t-sne algorithm, the groups of similar points cluster more closely together.

To understand how t-SNE works, it is important to look at each individual iteration. The first iteration puts the dots in a random order in the 2-dimensional plot, as shown in figure 6. Afterwards, each step attempts to bring together points that were clustered in the original D-dimensional plot. The algorithm should terminate when stability is reached and the 2-dimensional plot is a faithful representation of the clusters in the initial D-dimensional plot.

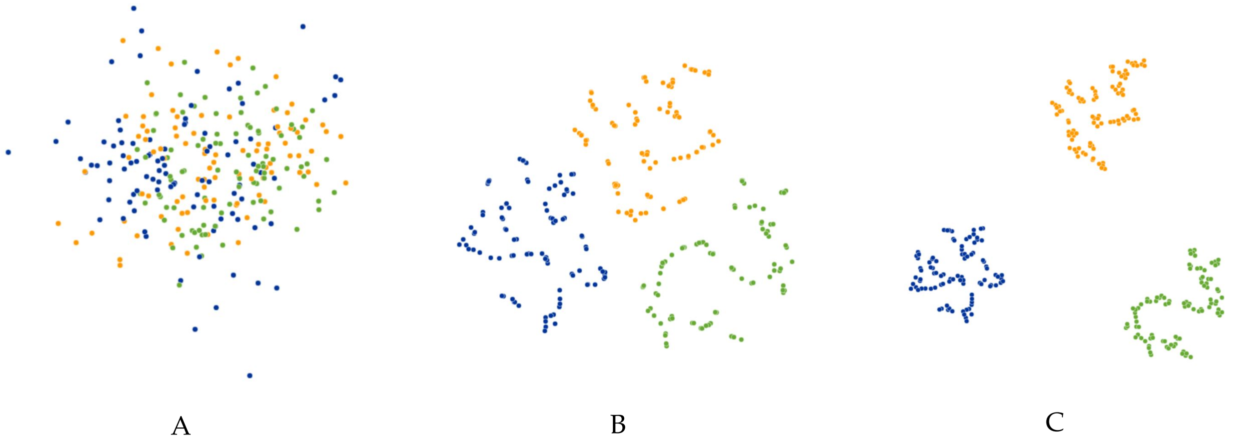


Figure 6: Run of t-SNE on sample data from [18], at different iterations. Colours show true classes and distance shows t-SNE clustering. (A) shows the data at iteration 1, with positions randomly projected in the 2D space. (B) and (C) show iterations 150 and 1,000 respectively.

To bring together intially clustered points, the algorithm computes the "distance" from one point to all of the others in the original plot. In this way, each point is attracted to the points that are near it on the original plot and repelled by points that are far from it on the original plot. Determining the actual "distance" from one point to another is done by first measuring it, using a simple distance measure such as Euclidian distance in multidimensional space and then determining the distance from the point to the normal curve with mean 0. The intuition behind this curve is that it represents the distribution of distances from the point of reference, which should be a normal distribution. The distance from the point to the curve, as shown in figure 7 is denoted by f(A,B) and represents the *similarity* between points A and B. This means that because the chosen distribution is normal, as points are further away from A, the *similarity* value will approach zero.

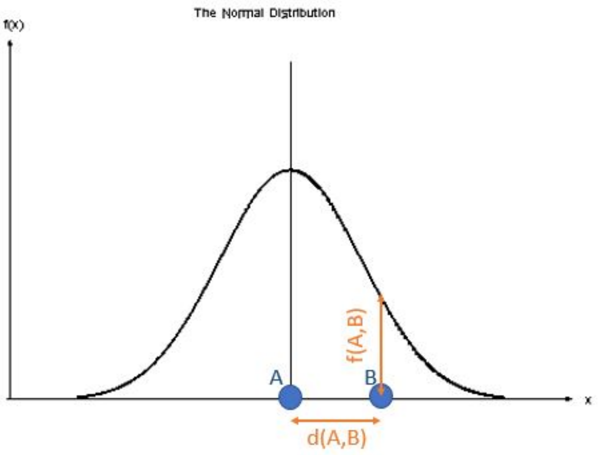


Figure 7: Shows how the distance between two points is calculated by t-SNE. A is the point to be moved, for which the distance is calculated, B is the point of reference for calculating the distance. d(A,B) is the distance between the points on the original D-dimensional plot. f(A,B) is the distance from B to the Normal curve of distances from A, with mean at 0.

The next step would be to select the points that are nearest and move the reference point towards them. Using the raw similarity to determine the nearest points will not work as expected if the reference cluster is less dense than others. This is because if the cluster is wider in diameter than the others, the distribution itself should be wider, with a larger standard deviation. To take this into account, all the similarity values are scaled in such a way that they add up to 1. To overcome this issue in practice, t-SNE has a *perplexity* parameter which is an estimated density of clusters. Another detail of implementation is that the similarity from point A to point B is not always equal to the similarity from point B to point A. The algorithm has an elegant solution to this problem, by simply averaging the two distances to obtain a single reflexive similarity between any two points.

The last step of the iteration is to calculate the similarities from any two points on the plot generated at this specific iteration. This is done in the exact same manner as described above, with a couple of exceptions. The distances used are the ones on the iteration's plot, for iteration one, the random scatter-plot will give the distances. Another difference is that instead of using a normal distribution for calculating the similarity, the algorithm uses a t-distribution. Without using a t-distribution, the clusters would clump up in the middle and be much harder to see.

After calculating all similarities, from the original plot and this iteration's plot, t-SNE nudges one point at a time. This is done in such a way that the similarity matrix obtained from this iteration's plot, will be closer to resembling the similarity matrix of the original plot. In this iterative manner, t-SNE projects the actual clusters formed in the D-dimensional space onto a 2-dimensional one.

In this project, t-SNE was used to classify the cells into types, in which each cluster corresponded to a type. This was done after applying PCA, for the reason that t-SNE should focus on clusters made by a smaller number of dimensions. If t-SNE was used on the original dimensions of the datasets, the 20,000 genes, very few points would have clustered together, thus giving a classification of too many classes with very few cells in each. If only PCA were used, the clusters would not have been maintained when reducing the dimensions, making it hard to determine which cells for a type. The combination of PCA and t-SNE ensures that the classification is made using a smal number (D) of relevant metafeatures and that the 2-dimensional projection conserves the clusters determined by these D dimensions.

2.5.3 Clustering

Using PCA and t-SNE provides a classification of cells, but since this is unsupervised learning, there is no way of telling straight away if the classification is correct. To verify the classification, another algorithm is employed - the clustering algorithm. Using two classification methods can give some confidence in the resulting classes. If by using two different methods, the same number of classes and with the same cells were obtained, then the classification is likely ″correct“. The correctness of the classification would simply mean that the trends in the data, which have helped identify the classes are relevant and in fact actual trends, not just noise.

The clustering algorithm is another classification algorithm, which works a bit different to t-SNE. It assigns actual classes to the data points, instead of plotting them and letting the human interpret the formed clusters. The clustering algorithm used throughout this project is the one described in [**?**]. It is based off of graph-like plots, for example KNN graphs (K nearest neighbours). The algorithm takes as parameter an estimate of how many "near" neighbours each point should have, meaning an average of how many points should be in a cluster. In the original KNN algorithm [21], the class of a point is estimated based on aggregate functions of the k nearest neighbours of it, which are determined by the Euclidian distance metric. However, there is no known classes in this dataset, so the algorithm has to determine communities in an unsupervised manner. The community determination in distance based graphs is a known problem, called modularity optimisation [20]. The modularity is a value (between -1 and 1) which determines the density of edges inside communities to edges outside communities. The algorithm consists of finding communities such that the modularity value is optimised, the edges inside communities are much more dense than outside communities. The modularity is determined by the following formula, as described in [22]:

$$Q = \frac{1}{m} \times \sum_{ij} [A_{ij} - \frac{k_i k_j}{m}] \delta(c_i, c_j)$$

where the following notation is maintained:

- m is the sum of all edge weights in the graph;
- A_{ij} is the edge weight between nodes i and j;
- k_x is the sum of all edge weights of edges adjacent to node x;
- $\delta(c_i, c_j)$ is a $1/0$ function which is 1 if the two communities c_i and c_j are equal and 0 otherwise (the function is 1 if and only if the two nodes belong to the same community).

The optimisation of the modularity value (Q) is done using the Louvain method [22]. This method consists of creating a community for every single node and calculating the modularity if a node were to be moved to a different community. After computing these values and moving to nodes to the closest community (with the highest modularity increase), all the nodes inside a community are merged into a super-node and the algorithm is repeated. This is done until no change results in modularity increase, i.e. no nodes can be moved into different communities.

For this project, the algorithm is applied after PCA with nodes representing all data points (cells) and edges with weights as Euclidian distances between cells in the PCA dimensions space. Because this algorithm assigns classes to cells, it has been used as a primary algorithm for classification and verified using t-SNE. The plot used for interpretation shows the cells as data points, with the following properties: the distances between points represent the t-SNE similarity; the colours of the points represent the cluster they belong to, according to the Clustering algorithm. If the positional clusters are the same as the coloured clusters, then the classification has been successful.

2.6 Existing software

Many of the algorithms described above have available implementations that can be used instead of writing new versions. The advantages that code re-use brings are avoiding unnecessary writing of code, which would result in time wastefulness and ensured correctness, since the implementations have already been validated by extensive testing. Some of these implementations are available as part of various software packages, such as Seurat, which will be described below. However, there are other software systems used throughout this project, some which help investigate large datasets in a visual manner, such as IGV and some which organise data based on different criteria, such as bamCleave. Making use of existing software has ensured the timely progress of this project.

2.6.1 Seurat

Seurat [23] is an R package, developed and mainained by the Satija lab, which aims to facilitate the exploration of single-cell RNA-seq data. Throughout this project, Seurat has been used for the classifications of cells into types.

Seurat provides in-built implementations of PCA (Principal Component Analysis), t-SNE (t-distributed stochastic neighbouring embedding) and of the clustering algorithm. The package also contains tools for filtering and normalising the data, with methods tailored to single-cell genomics analysis. All the plots used to understand the data and aid the filtering and classification have been made using built-in Seurat tools.

Using Seurat for the classification has helped focus more on tweaking parameters and ensuring a correct classification, rather than on implementation details of known algorithms.

2.6.2 IGV (Integrated Genomics Viewer)

The Integrated Genomics Viewer [24] is a high-performance tool that helps visualises large genomic datasets. Throughout this project, IGV has been used to explore gene mutations as detailed in section 7 of this report, in order to confirm the classifications made and the mapping from before chemotherapy to after chemotherapy. Figure 8 shows the interface of IGV. This is not easy to read and interpret at first glance, but the documentation is extensive and has helped understand and use the interface. The main observations to be made for understanding the contents of this report is that the gray areas represent reads aligned to the genome sequence and the colours represent mutations - reads that deviate from the normal sequence.

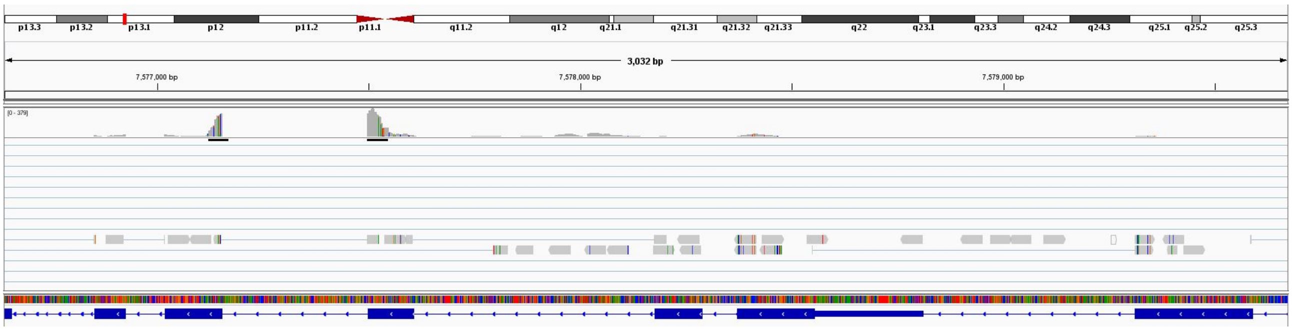


Figure 8: The Integrated Genomics Viewer interface, showing a sample dataset. The bottom line shows the introns and exons. Above them is the normal sequence of nitrogen bases. The gray forms show the distribution of reads across the genome. The coloured lines over the reads are mutations from the normal sequence.

The files read and visualised by the IGV software are extremely large (up to 20 GB) and would be very hard to analyse, since already reading and processing such files, even without any automatic analysis is computationally expensive. The alternative to using IGV for this project would be writing a script that aligns the reads and flags any mutations. This would be time-consuming and would require extensive prior knowledge of genomic analysis. Using IGV had helped understand the data in a visual way and identify relevant mutations much faster.

2.6.3 bamCleave

The bamCleave software was developed by Dr. Nigel Dyer at the University of Warwick, with the aim of separating reads from individual cells out of a file with reads from all cells.

The software operates on BAM files, which are binary files of SAM (Sequence Alignment Map) files. SAM files contain alignment maps of genome sequences, in a text format. More details on the file formats will be presented further in this report, when discussing the Project Datasets (Section 4).

The main aim of the software is to take as input a bam file and output multiple bam files, corresponding to the cells with most reads. Each output file contains reads coming from a single cell. The software takes as parameter the number of cells for which to create files, as well as the name of the input file. The parameters are given as part of the command. For example, the following line:

```
bamCleave -c 30 -n ':' -b 'subset.bam' -o 'subset_split'
```

will create files for the top 30 individual cells from the file subset.bam and prefix all the output files with the string 'subset_split'. The -n parameter specifies that the tag ':' should be used to separate single cell identity, rather than the default XC.

Even though the software did not have an immediate applicability to the project, it has been extended to contain extra functionality. The extension is the bamCleave Splitter and will be explained in the next section of this report.

3 Software development

The main aim of the project has been to analyse the data from Patient 9, by classifying the cells and drawing some conclusions about the effect of chemotherapy on each individual type of cells. However, throughout the project, some gaps were identified in the current flow of analysis. In order to fill those gaps, new methods and software were developed. These are automatic and speed up the process of analysis, while also increasing the level of understanding around the data.

One gap identified is that of lacking software for understanding the meaning behind cell classification. The Seurat classification allows scientists to spot marker genes, but gives no support when it comes to understanding what those genes *mean* on their own or as a group. The ClusterInsight Tool which was developed as part of this project was designed to overcome this issue.

Another issue with the current analysis flow is that the .bam files with cell reads are not separated by cell-types. It is useful to understand the gene mutations that are specific to a certain dataset, but even more so those that are specific to a certain cell type. The .bamCleave Splitter provides means to divide these files with respect to a cell-type classification.

One significant advancement that this project has brought to the analysis flow is a mathematical method for identifying similar clusters. This method has been implemented as part of the ClusterInsight Tool, but the details of how it works are described in section 6.4 of this report.

3.1 ClusterInsight Tool

ClusterInsight is a web application built to fill some gaps that were identified during the development of this project and to automate parts of the data analysis. One of the features it has is based on a mathematical formula developed as part of this project. This formula will be explained and applied in Section 6.4 *Similarity quantification*, whereas this section will only explain why and how the software was created. It will not go into detail about the development of the formula.

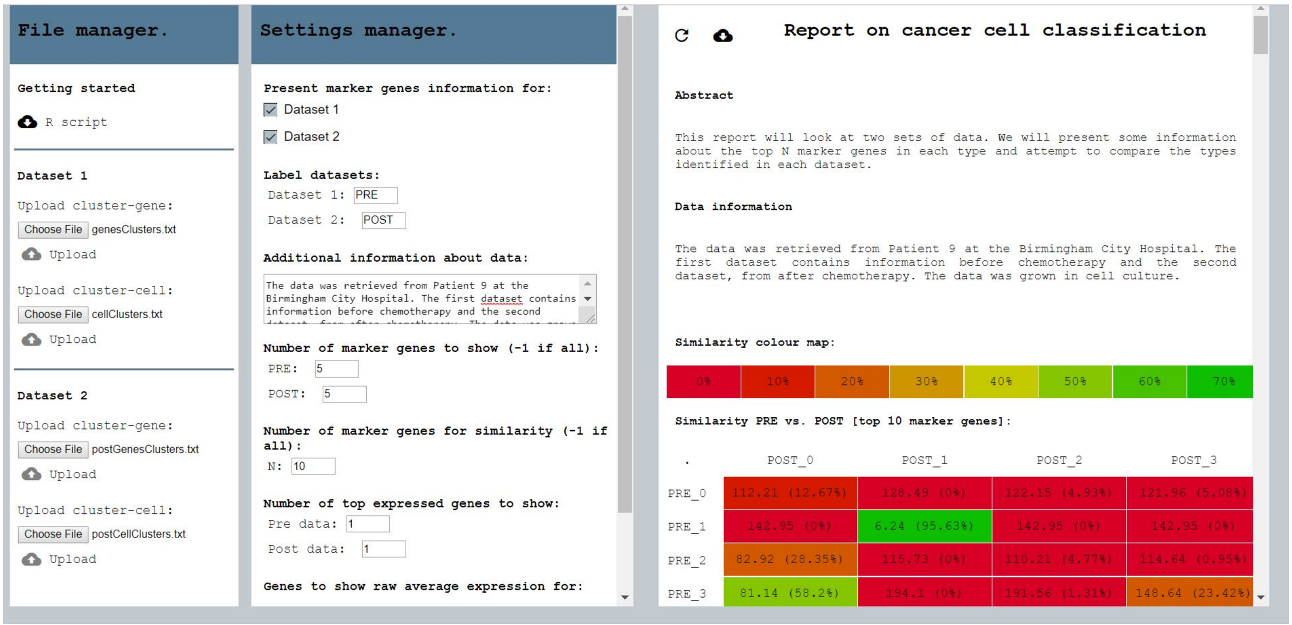
The system has a simple architecture, with three main components, which communicate with a local cache and an online gene definition library. The main components of the system are as below.

1. **File manager.** This allows the user to download a ”Getting started” script, which takes in the Seurat object with the classification and outputs formatted files for ClusterInsight to then use. It allows for upload of these files for one or two datasets.

2. **Settings manager.** Contains a list of settings that can be changed by the user to manipulate which parts of the data are looked at and to tweak parameters later used by the program.

3. **Report writer.** This takes the data from the file manager, the settings from the settings manager and writes a report accordingly, which the user can then download.

These three components together form a new analysis pipeline that is fully automated. The way ClusterInsight works is that the user downloads an R script, which takes in a Seurat object and outputs relevant data about marker genes and clusters to specific files. The user then uploads the files obtained and adjusts some settings. Upon the press of a button, the application will generate a report based on the files uploaded and the set settings. The user can upload key information about one or two datasets to analyse. If the user uploads two datasets, not only will the report contain information about each individual dataset, but it will also compare the two based on various criteria.



File manager.

Getting started

R script

Dataset 1

Upload cluster-gene:

Choose File genesClusters.txt

Upload

Upload cluster-cell:

Choose File cellClusters.txt

Upload

Dataset 2

Upload cluster-gene:

Choose File postGenesClusters.txt

Upload

Upload cluster-cell:

Choose File postCellClusters.txt

Upload

Settings manager.

Present marker genes information for:

☑ Dataset 1

☑ Dataset 2

Label datasets:

Dataset 1: PRE

Dataset 2: POST

Additional information about data:

The data was retrieved from Patient 9 at the Birmingham City Hospital. The first dataset contains information before chemotherapy and the second

Number of marker genes to show (-1 if all):

PRE: 5

POST: 5

Number of marker genes for similarity (-1 if all):

N: 10

Number of top expressed genes to show:

Pre data: 1

Post data: 1

Genes to show raw average expression for:

Report on cancer cell classification

Abstract

This report will look at two sets of data. We will present some information about the top N marker genes in each type and attempt to compare the types identified in each dataset.

Data information

The data was retrieved from Patient 9 at the Birmingham City Hospital. The first dataset contains information before chemotherapy and the second dataset, from after chemotherapy. The data was grown in cell culture.

Similarity colour map:

0%	10%	20%	30%	40%	50%	60%	70%

Similarity PRE vs. POST [top 10 marker genes]:

.	POST_0	POST_1	POST_2	POST_3
PRE_0	112.21 (12.67%)	128.49 (0%)	122.15 (4.93%)	121.96 (5.08%)
PRE_1	142.95 (0%)	6.24 (95.63%)	142.95 (0%)	142.95 (0%)
PRE_2	82.92 (28.35%)	115.73 (0%)	110.21 (4.77%)	114.64 (0.95%)
PRE_3	81.14 (58.2%)	194.1 (0%)	191.56 (1.31%)	148.64 (23.42%)

Figure 9: The interface of ClusterInsight (tool developed for fast report writing in this project). The **File manager** where the user uploads input; The **Settings manager** where the user changes parameters for the report formulas and sets the quantity of information display; The **Report** is the final output of the application, which can be downloaded by the user.

The interface (shown in figure 9) is simple to understand with three column sections representing the three components. The user interacts with the file manager and the settings manager, to give input about how the report should be written and retrieves the report from the report writer as final output.

ClusterInsight has a number of features aimed at helping the user understand the data and the classification in a faster manner than having to manually write scripts to compute certain applications and research meaning of genes. These features are outlined below.

1. **Additional information.** The user is allowed to put personalized labels on the datasets and add any information they wish to add to the report, such as provenience of the data and extraction dates. The purpose of this feature is to make the report clear to read.

2. **Similarity measurements.** This feature is aimed at facilitating a mapping between clusters of the two uploaded datasets. The need for this feature came from the fact that, to map clusters, most analysts will only glance over marker genes and determine if they are the same. A more detailed analysis of this problem can be found in section 6 (Cell-type progression) of this report. This section also covers the formula used by ClusterInsight in detail, while also providing a template for interpretation of results with the aid of an example.

ClusterInsight produces heatmaps showing percentages of similarity between pairs of clusters from the two datasets. Since the formula used is not symmetric, two such heatmaps are produced, showing comparison from dataset 1 to dataset 2 and viceversa. Alongst with the application of the formula, the produced report shows the sizes of each cluster in terms of cells, to help interpret the results.

The formula used to determine the similarity requires a parameter N (explained in section 6.4 of this report). This parameter can be adjusted in the Settings manager component of ClusterInsight.

The “Similarity” section of the outputted report is written by simply calling upon a script in the back-end which runs the similarity formula on the uploaded files. The results are then returned as a pair of matrices to the front-end, which parses them and displays them in the shape of heatmaps. The script has also been made available, in a version that prints out the results to the console, without any friendly user interface.

3. **Marker gene expression.** This feature focuses on helping the user investigate the marker genes of each cluster, by allowing them to print a number of the top marker genes and their raw average expression per cluster cell. ClusterInsight also allows for the user to input names of possible marker genes and look up the raw average expression of them in each cluster. By default, the report will show raw average expressions of gene TP53 (strongly correlated to cancer).

The parameters for these features - the number of top marker genes to show and the gene names to check as markers - can be modified using the Settings manager. If the user wishes to display all marker genes for each cluster, rather than a number of top ones, they can set the parameter to value -1.

To generate this section, the report writer sends a request to the back-end, which analyses the uploaded files and sends the results back to the front-end. The analysis of the files in the back-end is not complicated, since it only looks at expression of genes and aggregates results to then display.

4. **Gene definitions.** During the development of the project, one problem identified with the current analysis pipeline is that after the classification is done, most analysts will glance over names of genes important to each cell-type, without attempting to find correlations between these.

A recent study [25] shows that there are at least 46,831 genes in the human genome, without counting small RNAs. Each gene has a different function to the organism and it is impossible to memorise all of them. There are online libraries which show definitions of genes, but some of them are incomplete or not studied in enough depth to clearly understand their function. Most scientists will know the definitions of some very important genes and recognize the names, but often it takes a long time to understand marker genes of each cell type.

After identifying the cell types and looking at the top marker genes corresponding to each of them, it might be immediately obvious what the role of the cell type is. For example, if a cell type has a combination of known marker genes for ovarian cancer stem cells, such as IFI27 and POSTN, it can be labelled intuitively as cancer stem cells. Unfortunately, this is very often not the case and scientists continue to work with the cell types without knowing what the *meaning* of them is. In the case of this project the definitions behind each marker gene have helped provide an intuitive classification and progression of cell-types, which has further validated the results obtained by more mathematical methods.

ClusterInsight automatically provides definitions for a number of top marker genes for each cluster. These are printed in the final section of the output report and the number representing how many genes to search for definitions can be adjusted in the Settings manager component.

The system back-end contains a web-scraper for a known online gene library, named GeneCards. The application will send requests to the GeneCards library and retrieve the definition for the marker genes. In order to make this process faster and stop the application from sending numerous repetitive requests, all the definitions already retrieved are cached. ClusterInsight first requests the definition for each gene name to the caching system and if the gene was never queried before, it will send an HTTP request to GeneCards. The use-case diagram in figure 10 shows how the gene definitions are retrieved and the actions of the system.

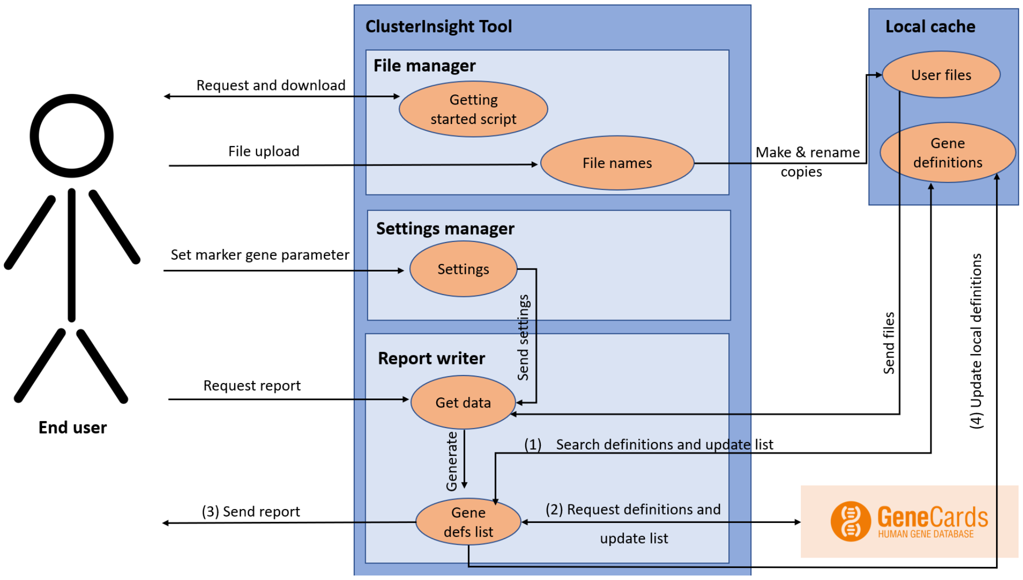


Figure 10: Diagram showing how the ClusterInsight system works for automatic retrieval of definitions for the marker genes for each cluster from a user classification.

Technologies used

The application was developed in two parts: front-end and back-end, which communicate through HTTP requests. One relevant hidden component of the system is the local cache, which holds the files uploaded by the user and a dictionary of gene definitions that were already retrieved from GeneCards.

The front-end was written using Angular and its purpose is to interact with the user and transmit formatted requests to the back-end. The front-end keeps track of all the settings set by the user and renames the files uploaded to conform to the standards of the back-end. The reason behind using Angluar as the main technology is that it was very well suited to the component structure, since it has an in-build component system. The interface is user-focused, allowing for fast generation of the report and hiding all the heavy-computation in the back-end. In terms of interface design, the application structure was inspired from other report writing web application, such as Overleaf.

The back-end was written using Python and its purposes is to parse and manipulate data obtained from the files uploaded by the user. The back-end consists of a main "manager" script which listens from requests from the front-end. When a request is retrieved, it contains information about the type of request (similarity result, marker gene name, gene definitions list etc) and the corresponding parameters. The "manager" parses the request and sends the relevant information to the script specific to the type of request. After the script is run, the results are formatted and sent to the front-end by the "manager". The reason behind using Python for the back-end is twofold. Firstly, it allows for fast and easy manipulation of large dataframes (such as the ones that need to be analysed). Secondly, it has in-built packages for sending retrieving HTTP requests, as well as packages to facilitate caching (such as pickle).

The local cache consists of copies of the user uploaded files and a pickle file for the gene definitions. The user files are parsed upon request to generate the needed information. The pickle file consists of a dictionary with the key being the name of the gene and the value being the definition. This allowed for fast retrieval of gene definition as well as fast checking for already defined genes, before sending the HTTP request to genecards.

The application is not yet deployed, but it will be used by the ovarian cancer research project at the University of Warwick. The reason behind not deploying at the moment is that it fulfills very specific purposes and it is aimed at specialised users, which already know and understand the similarity formula and all the other features offered by ClusterInsight.

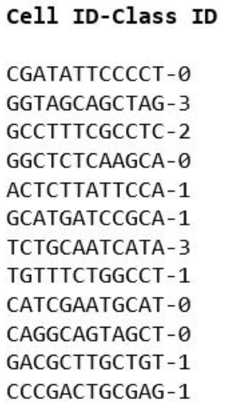
3.2 .bamCleave Splitter

Investigating genome wide mutations is a common analysis done by scientists, through uploading .bam files in IGV (Integrated Genomics Viewer). This allows them to look at specific genes and see how they are mutated in specific patients. For example, one common method of analysis for cancer patients is to look at gene TP53. This gene generates a protein known to regulate sporadic growth of cells and therefore act as a tumour suppressor. Heavy mutations of this gene could be a reason for tumour genesis. In the context of this project, it would be useful to see if some of the cell-types contain more mutations than others or if mutations are specific to different cell-types. To investigate these mutations, the .bam file with genome-wide information from all the cells should be split into multiple files, each one corresponding to a cell-type. These files could then be uploaded in IGV and analysed.

The .bam files with information about all the captured cells and genes that are outputted by Drop-seq are extremely large and could not be split manually. To allow for cell-type specific genome mutations analysis, the .bamCleave Splitter tool was developed.

The bamCleave Splitter is an extension to the bamCleave software, which separates the .bam files with respect to the classifications made. The bamCleave software project was provided by Dr. Nigel Dyer, a member of the ovarian cancer research group at the University of Warwick as a visual studio project. The extension was added to the base code as another option given to the user. The .bamCleave software, as described in the Existing Software subsection of the Background information section above, creates separated .bam files with the reads corresponding to single cells. The extension added was aimed to replace the single cell files with files specific to each group from the classification. The task allowed for the extension to make use of already existing code from the source of .bamCleave and consisted of mostly manipulating pointers of writers to different files.

The .bamCleave software takes as input a .bam file with the reads for all the cells in the dataset and outputs separate files for a number of cells (given as parameter). The Splitter extension is called with an extra command parameter and takes as input a file with the classification (-g "classification.txt"). To ensure that the input files are in the correct and expected format for the software, the following sensible rules must be followed when generating the input file containing the classification:



```
Cell ID-Class ID

CGATATTCCCCT-0
GGTAGCAGCTAG-3
GCCTTTCGCCTC-2
GGCTCTCAAGCA-0
ACTCTTATTCCA-1
GCATGATCCGCA-1
TCTGCAATCATA-3
TGTTTCTGGCCT-1
CATCGAATGCAT-0
CAGGCAGTAGCT-0
GACGCTTGCTGT-1
CCCGACTGCGAG-1
```

Figure 11: The standard format of the group table file that should be fed as input to the .bamCleave Splitter. The example has cell IDs of 12 base and 4 classes.

1. All cell identifiers must be of the same length. The Drop-seq standard is of 12 base, but the software is not restricted to a specific length, as long as all cell IDs are of the same format.

2. Group IDs from the classifications must be integers ranging from 0 to N inclusively, where N is the number of classes - 1.

3. A cell-class mapping is a string that contains the cell identifier, the separator ″-“ and the integer denoting the class to which the specific cell belongs to.

4. Every line corresponds to a single cell-class mapping.

5. The classification must be correct, i.e. the same cell cannot be repeated with two assignments to different classes.

There were two main approaches towards building the Splitter extension. The first and simplest approach was to use the existing bamCleave software to create files for all the cells and then merge the files corresponding to cells from the same group. This merging of bam files can be done by employing the package samTools, which is software created specifically for the manipulation of bam and sam files. The second approach consisted of creating writer pointers for each class file and redirecting reads for specific cells, to the corresponding class file. This method was more difficult to implement, but had the advantage of not performing redundant computations. This is due to the fact that there is no creation of separate files that are then merged, but instead the files are created as they should from the start. This method also holds the advantage of not creating more files than needed, and therefore saving memory as well as time.

One important aspect that had to be taken into consideration when developing the extension was related to the possible differences between cells in the bam file and cells in the classification file. The classification is done after filtering the data and therefore some cells which have reads in the input bam file will not have a corresponding class in the classification file. The reads from these cells were printed to another bam file, separated from the class specific files.

The extension followed a number of clearly separated steps, which made use of the already implemented methods of the bamCleave software.

1. Parsing the classification file to obtain a HashTable, in which the key is the identifier of the cell and the value is the group it belongs to. This allowed for fast mapping of reads to files in further parts of the application.

2. Creating a file and a bamWriter for each group identified in the previous stage. The bamWriters are kept as values in an indexed array, where the position represents the identifier of the class (since all class IDs are positive integers).

3. Creating a list of cells for which reads should be found. This was done using the existing code, but instead of finding the top cells, it adds all the cells found, that are also in the classification file.

4. Parsing the reads from the input bam file and mapping them to a specific output bam file. This was mostly done making use of existing code, but by using the bamWriter corresponding to the class of the cell.

The biggest challenge when developing and testing the Splitter extension is that the input files to be handled are incredibly large and therefore a run of the bamCleave software takes up to 15 hours. This was a problem when trying to debug and test the code, because of how time-consuming it is to do a run. To overcome this, new test files were written from the original input files, with much smaller sizes (about 10% of the original size). The application was then tested to meet the following criteria:

- The input is parsed correctly and generates the cellIDs that can be found in the input bam file as well.

- Every cell is mapped to the correct class file.

- The class specific files contain reads from all the cells in the class, not only one. This was relevant for checking if a new cell added to a group was not overriding already existing information in the output class file.

- None of the cells from the classification file are missing from the original bam file. This is equivalent to checking that there are some reads for every cell.

- Indexing the printed files corresponding to each class.

Technologies used

The Splitter extension was added to the already existing code of the bamCleave software, which was written in C++. The advantages of using C++ are that it allows for very fast low-level computation, as well as the existing APIs specific to operations of bam files, such as : bamtools and bioinformaticsLib.

3.3 Overall pipeline improvements

The analysis pipeline for single cell RNA-seq data is now improved in two ways. Firstly, drawing conclusions about the cell-types identified after classification is now faster and requires no repetitive work or code writing, but only interpretation of presented results. Secondly, investigating mutations is now also possible for cell-type specific genomes.

The main advantage brought by the development of these tools is firstly speed, since analysts can employ an application to do a part of the work, which would normally be tedious and time consuming. Other advantages of these tools are that they can be used together on any datasets, the group specific mutations found using the bamCleave Splitter can confirm some labelling of cell-types and they help assess the effect of chemotherapy on each specific type. Looking at which mutations from the genome have disappeared in each group can help understand which types of cells a specific treatment had impact on and which ones it did not change.

4 Project Datasets

The analysis conducted in this project was mainly focused on two datasets, each comprised of a DGE matrix and a BAM file, which will be detailed further in this section. The project also made use of two other datasets containing information about germline and cancerous data, taken in bulk instead of single-cells, but coming from the same patient as the main datasets. These were taken with the purpose of comparison with the main single-cell datasets.

4.1 Meaning

The first dataset contains information from tumour cells before chemotherapy from Patient 9. The cells were extracted from a biopsy, which was performed on the patient before any chemotherapy treatment.

The second dataset contains information from tumour cells after chemotherapy from Patient 9. The cells were extracted from surgery, which was performed immediately after the chemotherapy treatment.

Given that the two datasets come from the same patient and the same tumour, they are likely to contain the same cells, but which have changed during chemotherapy. Due to the same provenience of cells, the analysis conducted below was able to draw some likely conclusions on the changes that have occured and efficiency of chemotherapy on this tumour.

In both cases the data was extracted using Drop-seq. The data was then raised in cell culture, in such a way that all healthy cells were eliminated. This came with the advantage that the analysis was focused on cancer cells, but the disadvantage of not knowing with certainty if the relevant information is kept in cell culture. This poses a great question of whether the change that occurs in cells represents a problem to the classification. It could be the case that each cell culture changes the cells in a different way and therefore any mapping found from one dataset to another may not be relevant. However, the results obtained in this project show that both datasets seem to have developed in the same way, since there are clear connections between the clusters identified during both classifications.

The two datasets, germline and cancerous, with bulk collected data have also been extracted from Patient 9, from healthy and diseased tissue respectively. These two bam files have only been used when reasoning about gene mutations. The aim was to compare the mutations found in the main datasets, with the mutations found in healthy or cancerous tissue of the same patient.

4.2 Notations

This document will continue to reference the two datasets, by following the below notations.

1. PRE = the dataset taken from Patient 9 during biopsy, before chemotherapy.
2. PRE_i = the i^{th} cell type (cluster) identified for the PRE dataset.
3. POST = the dataset taken from Patient 9 during surgery, after chemotherapy.
4. POST_i = the i^{th} cell type (cluster) identified for the POST dataset.

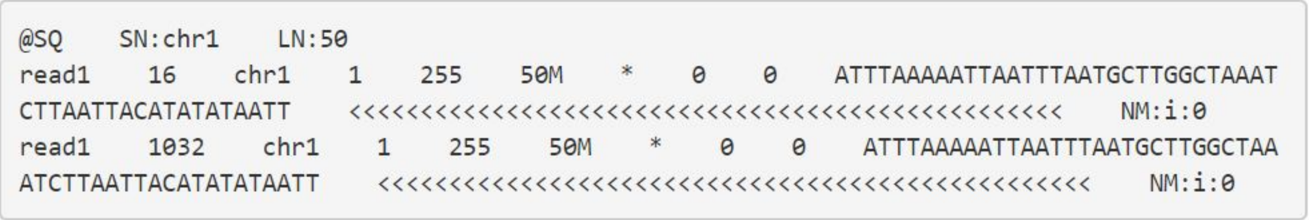
Both datasets, PRE and POST are comprised of a DGE matrix and a BAM file. The DGE matrices will be used for the classification and cell type progression mapping. The BAM files will be used for the identification of gene mutations with respect to each cell type.

4.3 Formatting as input

This section will explain how the dataset is formatted, to give an overview of what the input for this project looked like.

A BAM file is the binary compressed respresentation of a SAM file (Sequence Alignment Map) and contains a header and a sequence section. The sequence section is formed of a list of biological sequences, which are aligned to a reference sequence. Each alignment has 11 mandatory fields, containing information about the segment sequence, the starting position, the next read and the quality of the mapping. [26] These files can be read using software packages such as SAMtools. Figure 12 shows an example of a SAM file, viewed with SAMtools.

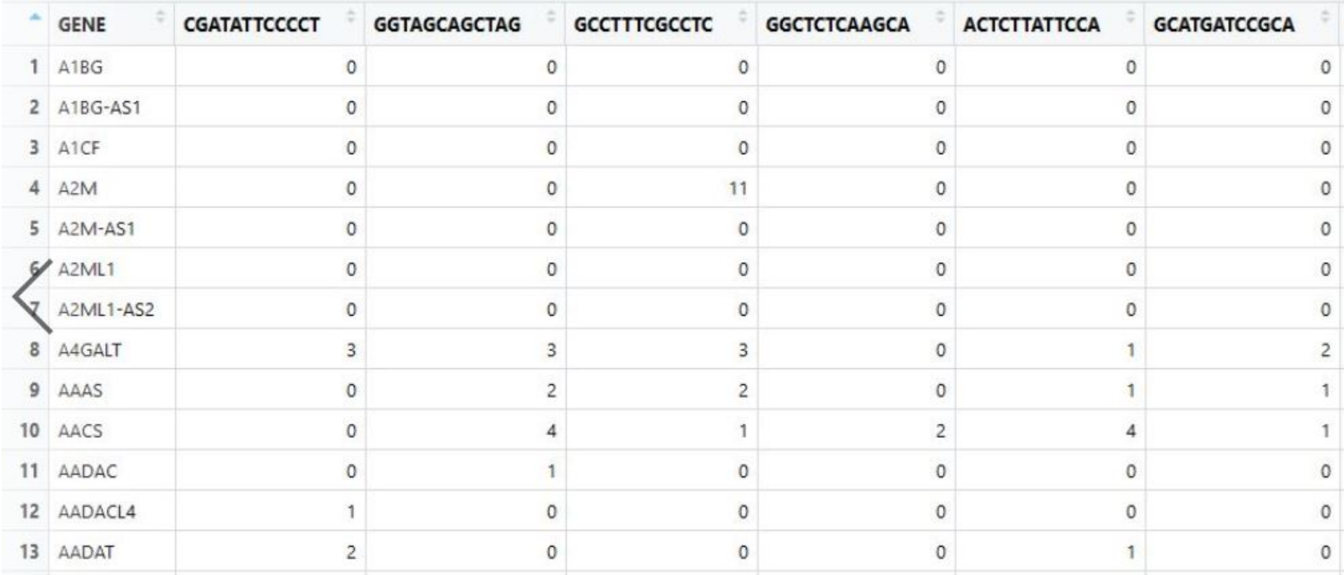
BAM files are binary files, meaning that their content is impossible to read as such. There are various tools to open and parse a BAM file, before analysing it. BAM files contain large amounts of data in a difficult to understand format and therefore programs are needed to analyse them. The BAM files for the PRE and POST datasets analysed in this project have sizes of 10GB and 20GB respectively.



```
@SQ     SN:chr1     LN:50
read1     16     chr1     1     255     50M     *     0     0     ATTTAAAAATTAATTTAATGCTTGGCTAAAT
CTTAATTACATATATAATT     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     NM:i:0
read1     1032     chr1     1     255     50M     *     0     0     ATTTAAAAATTAATTTAATGCTTGGCTAA
ATCTTAATTACATATATAATT     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<     NM:i:0
```

Figure 12: Example of SAM file viewed with SAMtools. [27]

A DGE matrix is a large matrix with integer entries, in which the columns represent cells and the rows represent genes. Each entry corresponds to the number of unique molecular identifiers found for a specific cell in a specific gene and represent *how expressed the gene is in the cell*. Figure 13 shows an example of a small DGE matrix of sample data.



	GENE	CGATATTCCCCT	GGTAGCAGCTAG	GCCTTTCGCCTC	GGCTCTCAAGCA	ACTCTTATTCCA	GCATGATCCGCA
1	A1BG	0	0	0	0	0	0
2	A1BG-AS1	0	0	0	0	0	0
3	A1CF	0	0	0	0	0	0
4	A2M	0	0	11	0	0	0
5	A2M-AS1	0	0	0	0	0	0
6	A2ML1	0	0	0	0	0	0
7	A2ML1-AS2	0	0	0	0	0	0
8	A4GALT	3	3	3	0	1	2
9	AAAS	0	2	2	0	1	1
10	AACS	0	4	1	2	4	1
11	AADAC	0	1	0	0	0	0
12	AADACL4	1	0	0	0	0	0
13	AADAT	2	0	0	0	1	0

Figure 13: Example of a DGE matrix with 13 genes across 6 samples.

Real-life DGE matrices are very large, meaning that most of the entries are 0 values, the data being very sparse. The PRE and POST DGE matrices analysed in this project have sizes of 25,465 x 2,000 entries and 22,641 x 999 entries respectively.

5 Data Analysis and Classifications

The datasets analysed during this project were assessed in terms of quality, filtered and then classified. This was needed in order to identify possible cell types and assess the effect of chemotherapy on each of these. This section will give an overview of the process of identifying cell types for PRE and POST data from Patient 9.

5.1 Pre treatment data

The first dataset to be analysed as part of this project has been obtained from a biopsy on Patient 9. This dataset consists of a DGE matrix of tumour cells, before chemotherapy.

5.1.1 Data quality and filtering

Initially, the dataset was comprised of information for 25,465 genes across 2,000 samples. The DGE matrix was very large and sparse, containing a total of 25,373,463 reads across the 50,930,000 entries (cell x gene). In order to reduce the sparsity and eliminate any possible outlier cells or genes, two rounds of filtering were applied to the dataset.

The first round of filtering was aimed at reducing the sparsity of the DGE matrix, by eliminating genes expressed in less than 3 cells and cells that have less than 200 genes captured. These cells and genes are likely to be partially captured information and not relevant for the classification and analysis. The bounds of 3 and 200 are the constant default filtering from Seurat. After this filtering was applied, the dataset contained a total of 20,681 genes across 2,000 samples. The number of cells was not reduced at all, but the number of genes decreased by approximately 19%.

The second round of filtering was aimed at eliminating all possible outlier cells, such as broken cells, cells with uncaptured reads and duplet cells. In order to achieve this, metadata had to be calculated: nGene (the number of genes captured for each cell); nUMI (the number of unique molecular identifiers for each cell); percent.mito (the percentage of mitochondrial genes captured for each cell). To identify possible outliers, the distribution of these metadata features was plotted along with the ratio of percent.mito and nGene to nUMI.

From figure 14 it can be observed that: The number of genes captured is distributed across the cells in the dataset, with most cells clustered under 3,000 nGene; The number of unique molecular identifiers is mainly low, below 100,000 per cell; The percentage of

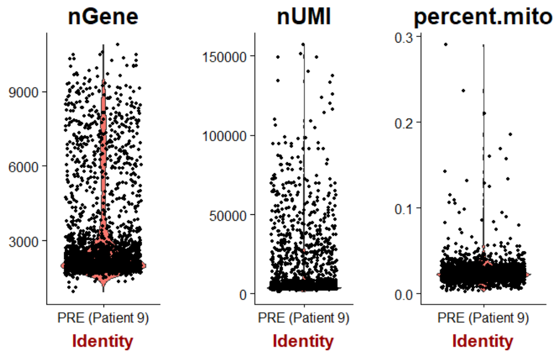


Figure 14: The distribution of the number of genes captured, number of unique molecular identifiers and percentage of mitochondrial genes captured for the cells across the PRE dataset for Patient 9.

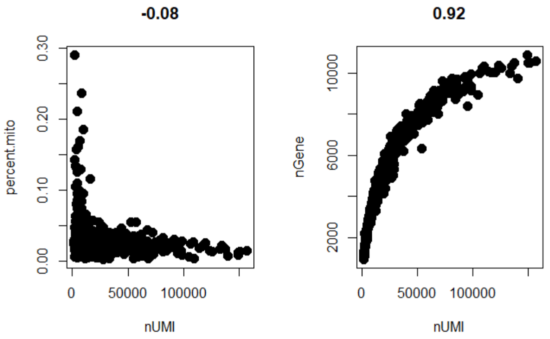


Figure 15: Left: The ratio of percentage of mitochondrial genes captured against the number of unique molecular identifiers, for cells in the PRE dataset for Patient 9. Right: The ratio of number of genes captured agains the number of unique molecular identifiers, for cells in the PRE dataset for Patient 9.

mitochondrial genes is generaly low, below 0.1, with some outliers above this bound.

From figure 15 it can be deduced with confidence that a low number of unique molecular identifiers is related to a high percentage of mitochondrial genes and that a high number of unique molecular identifiers in a cell can be related to a high number of genes captured for that cell. This gives a certainty over the quality of the data being suitable for analysis, since patterns can be identified.

After inspecting the above distributions, the filtering was done by eliminating cells with a number of unique molecular identifiers outside the range (20,000-100,000) and cells with a percentage of mitochondrial genes captured above 0.1. This resulted in a datased comprised of 20,681 genes across 333 samples.

5.1.2 Classification

The cell classification was done using the Seurat clustering algorithm, with a k parameter of 10 and 5,000 iterations. In order to verify the classification t-SNE was used, with a perplexity of 10 and 10,000 iterations. This was done after data scaling and dimensionality reduction with PCA (to 10 dimensions).

The classification identified 6 clusters, PRE_0, PRE_1, PRE_2, PRE_3, PRE_4 and PRE_5 with sizes of 155, 87, 45, 27, 11 and 8 respectively. The top 10 expressed marker genes for each cluster in figure 16 have been determined using the Seurat formula for marker genes [28] and are presented in figure 17. The expression of these genes clearly delimits

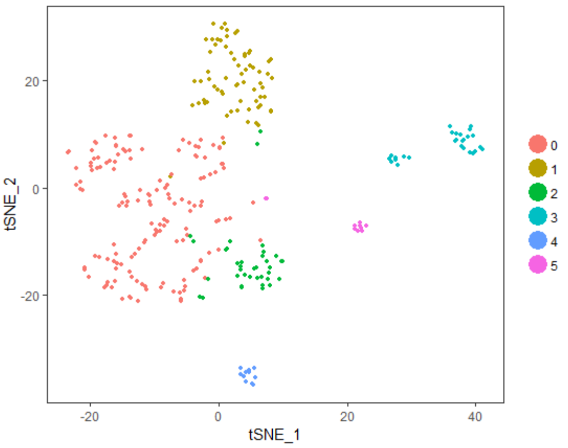


Figure 16: Classification of cells in PRE, Patient 9. The distance between data points is given by t-SNE and the colours are corresponding to the Seurat clustering algorithm.

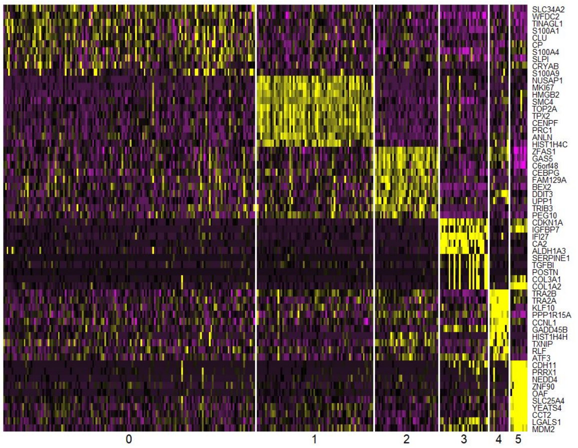


Figure 17: HatMap showing the expression of the top 10 marker genes (rows) for cells in the clusters from figure 16. The purple to yellow scale represents low to high expression of genes.

the clusters, thus providing confidence that the classification is relevant to the expressed genes. There are some observations that can be made on cluster PRE_3. The cluster seems to be separated in two parts, both on the t-SNE plot (figure 16) and on the HeatMap (figure 17). Upon adjustment of the resolution of the clustering algorithm, PRE_3 divides into 2 separate clusters. However, these clusters are of very small sizes and there is not much relevant difference in terms of marker genes. For this reason, when continuing the analysis, the two parts have been considered a singular cluster. Another notable observation is that 4/10 top expressed marker genes of cluster PRE_3 (CDKN1A, IGFB7, COL3A1, COL1A2) are also highly expressed in all cells of cluster PRE_5.

One problem that can arise when running algorithms such as t-SNE is that seldom cells with a very low number of unique molecular identifiers or other metadata features, cluster together. To verify that this was not the case with the above classification, the metadata was plotted by clusters.

Figure 18 shows that the distribution is similar in each cluster. The one difference is cluster PRE_5, which has cells which have very few genes captured, few unique molecular identifiers and a low mitochondrial percentage. It is noticeable that the classification did not exclusively separate the cells with the lowest mitochondrial percentage (see cluster PRE_3 in figure 18) not the cells with the lowest number of genes captured (see the overlap with clusters PRE_3 and PRE_4 in figure 18). Since the marker genes for PRE_5 are not highly expressed in any of the other clusters, the classification clearly distinguished the clusters based on gene expression. It is worth noticing here that cluster PRE_5 is very small in size

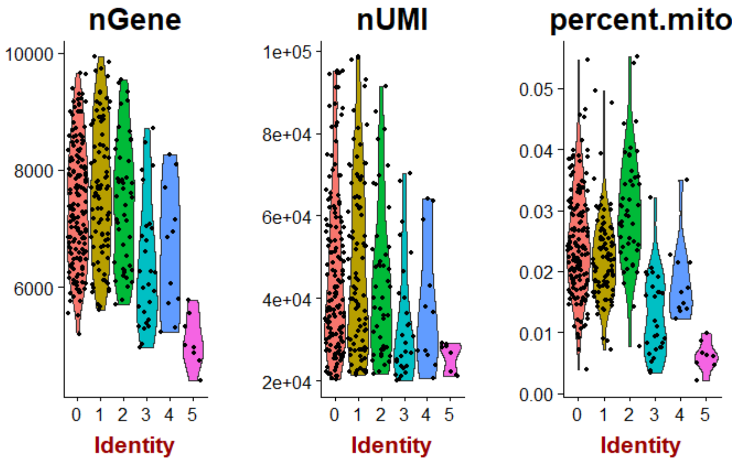


Figure 18: The distribution over the number of genes captured, number of unique molecular identifiers and the percentage of mitochondrial genes captured in PRE, Patient 9, based on the clustering in figure 16

in comparison with the other PRE clusters, having a size of 8 cells.

5.1.3 Cell type identities

It is important to understand what each cell-type actually represents for the tumour. For example some cell-types might be handling specific cell-cycle processes, some cells might be handling tumour growth etc. To place labels on the role of the cell-type in the tumour, the genes that are important for each cluster had to be understood. For each cluster, the top 10 marker genes were selected. Looking at the definitions of these genes has provided some insight into the roles of each cell-type. One of the tools developed as part of this project, ClusterInsight, allows for automatic retrieval of gene definitions for identified marker genes, thus facilitating this process. By focusing on the definitions and roles of the top 10 marker genes of each cluster, the following observations were made:

- **PRE_0** (155 cells) presented high expression of three genes, encoding calcium-binding proteins: S100A4, S100A1 and S100A9. Out of these three, S100A4 is known to be implicated in tumour metastasis. These genes were expressed along with genes WFDC2 and CLU, which are correlated with cancer and tumour progression. In particular WFDC2 has been identified as a highly expressed gene in ovarian cancer.

- **PRE_1** (87 cells) is likely to be formed of cells undergoing mitosis, because of the high expression of genes such as NUSAP1, MKI67, ANLN and others. These are protein encoding genes with roles in the assembly of the mitotic spindle or chromosome segregation during mitosis. It should be noted that this cluster showed high expression of gene TOP2A, whose mutations are associated with cancer drug resistance. The current theory is that this cluster represents a state, not a type and that cells from different types go through this state at certain points in the cell cycle. However, this is not a definite statement, since there are expressions of genes such as TOP2A which also indicate some similarities in terms of cancer-related affections on the cells.

- **PRE_2** (45 cells) contained high expression of genes such as ZFAS1 and PEG10, which are related to cell proliferation and differentiation. High expression of such genes is strongly linked to cancer progression. The cluster also contained some tumour supressor genes, with high expression: BEX2, GAS5 and FAM129A. The other top marker genes were connected with apoptosis, controlled cell death.

- **PRE_3** (27 cells) is likely to be formed of ovarian cancer stem cells. This is due to high expression of known marker genes for ovarian cancer stem cells, such as IFI27 and POSTN. The cluster presented these genes with high expression, along with ALDH1A3 and CDKN1A, which are also strongly linked to tumour cells.

- **PRE_4** (11 cells) contained marker genes which encode proteins with roles in mRNA splicing, such as TRA2B, TRA2A and CCNL1. The cluster also presented TXNIP and ATF3 with high expression, which are genes encoding proteins that respond to cellular stress, seldom associated with cancer cells. TXNIP has been known to have a role as tumour suppressor and is required for the maturation of natural killer cells.

- **PRE_5** (8 cells) presented marker genes that are related to cancer and are regulating expression of immune system cells. Gene NEDD4 actively regulates expression of the tumour suppressor PTEN. Gene LGALS1 is a strong inducer of T-cell apoptosis, thus preventing any attack of the immune system onto the cancer. Gene MDM2 promotes tumour formation by targeting tumour suppressors proteins and accelerates tumour formation. All these three genes are highly expressed marker genes of this cluster. PRE_5 also presented high expression of gene CDKN1A (marker gene for cluster PRE_3), which is a regulator of cell cycle progression, whose expression is controlled by P53, thus having an important part in the execution of apoptosis. This combination of cells suggests that cluster PRE_5 is formed of cells which are actively fighting the immune system to promote the spread of cancer.

All the cell clusters present marker genes which are strongly linked with cancer and different stages of it. The cell type identities that can be placed on these clusters with a high level of certainty are: PRE_1 as highly proliferative cells; PRE_3 as cancer stem cells; PRE_5 as tumour cells which play a role in the subversion of the immune system. These cell type identities seem to be very likely after investigating the marker of genes of each cluster. However, the labels are not a certainty and should be treated as a likely classification.

5.2 Post treatment data

The second dataset to be analysed throughout this project has been obtained from surgery on Patient 9. This dataset consists of a DGE matrix of tumour cells, after chemotherapy.

5.2.1 Data quality and filtering

Initially, the dataset was comprised of information for 22,641 genes across 999 samples. The DGE matrix was sparse, containing a total of 14,047,794 reads. Data was filtered, to eliminate any broken cells, partially captured cells and duplets.

The first round of filtering was aimed at reducing the sparsity of the DGE matrix. The thresholds were the default Seurat constants; eliminating all genes that appear in less than 3 cells and all cells that have less that 200 detected genes. After this filtering, the number of cells was only reduced by 3, to a total of 996 samples. However, the number of genes was reduced to 17,744, eliminating approximately 21% of the initial set of genes. The remaining number of reads was 14,040,670, reduced by only 1%.

The second round of filtering was aimed at eliminating the broken cells and duplets. To investigate the overall quality of the dataset, the number of genes captured for a cell was plotted, along with the number of unique molecular identifiers and the percentage of mitochondrial genes. These metadata fields were investigated in terms of their distribution across the dataset, as well as their relation to eachother.

From Figure 19, it can be observed that: The number of genes captured in cell is well-distributed across the dataset with no obvious outliers; The number of unique molecular identifiers is concentrated below 50,000; There are some cells with a percentage of mitochondrial cells higher than 0.02, which are breaking this distribution.

From Figure 20, it can be observed with high certainty that: The cells which have a high percentage of mitochondrial genes captured, have a low number of unique molecular

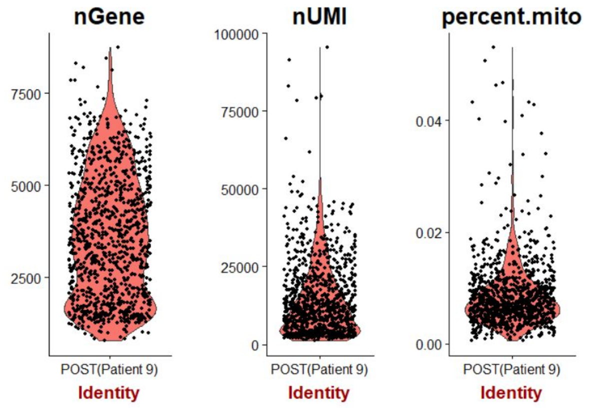


Figure 19: The distribution of the number of genes captured, number of unique molecular identifiers and percentage of mitochondrial genes captured for the cells across the POST dataset for Patient 9.

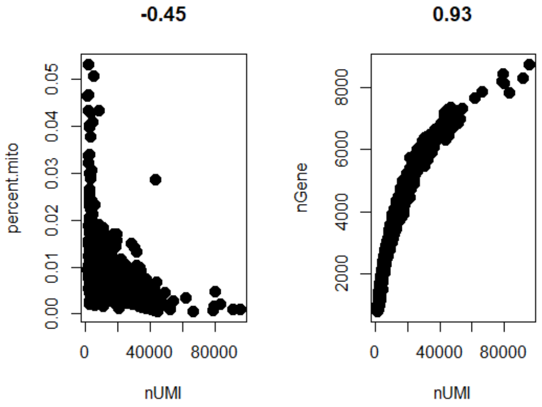


Figure 20: Left: The ratio of percentage of mitochondrial genes captured against the number of unique molecular identifiers, for cells in the POST dataset for Patient 9. Right: The ratio of number of genes captured agains the number of unique molecular identifiers, for cells in the POST dataset for Patient 9.

identifiers; The number of genes captured per cell is mostly increasing with the number of unique molecular identifiers. These patters give confidence that the dataset is of good enough quality to be further analysed.

Upon inspection of the distributions presented above, only the cells with a number of unique molecular identifiers in the range (15,000-50,000) and with a percentage of mitochondrial genes below 0.02 were kept. This resulted in a dataset of 17,744 genes over 341 samples.

5.2.2 Classification

The classification of the cells was done using the clustering algorithm embedded in Seurat, with a k parameter of 20 and 5,000 iterations and t-SNE, with a perplexity of 50 and 5,000 iterations. This was done after data scaling and dimensionality reduction with PCA (to 10 dimensions).

The classification identified 4 clusters, POST_0, POST_1, POST_2 and POST_3 with sizes of 165 cells, 67 cells, 61 cells and 48 cells respectively. The top 10 expressed marker genes for each cluster, shown in figure 22 have been determined using the Seurat formula for marker genes [28] and denote a clear separation between clusters. One notable observation is that the top 10 expressed marker genes of POST_3 are also expressed in POST_1.

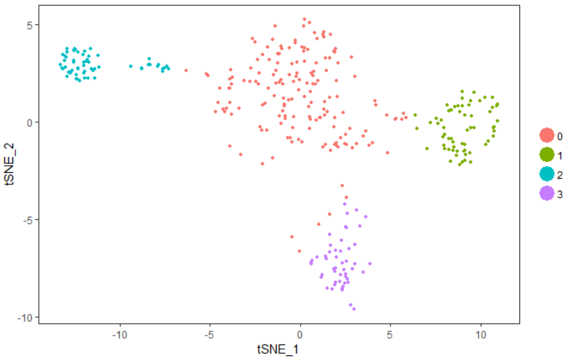


Figure 21: Classification of cells in POST, Patient 9. The distance represents t-SNE clusters and the colours are corresponding to the Seurat clustering algorithm.

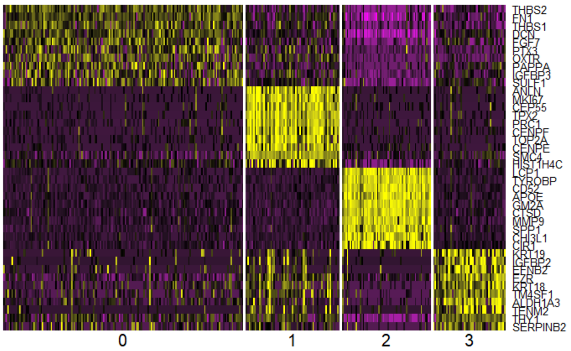


Figure 22: HeatMap showing the expression of the top 10 marker genes (rows) for cells in the clusters from figure 21. The purple to yellow scale represents low to high expression of genes.

One problem that could arise when classifying cells is that the tendency of the algorithms to group them by the amount of captured information rather than the genes expressed. The clusters should have no significant differences in terms of number of genes expressed, number of unique molecular identifiers or percentage of mitochondrial genes. To verify that this is not the case with the above classification, the distribution of the three metadata features should be similar in each cluster.

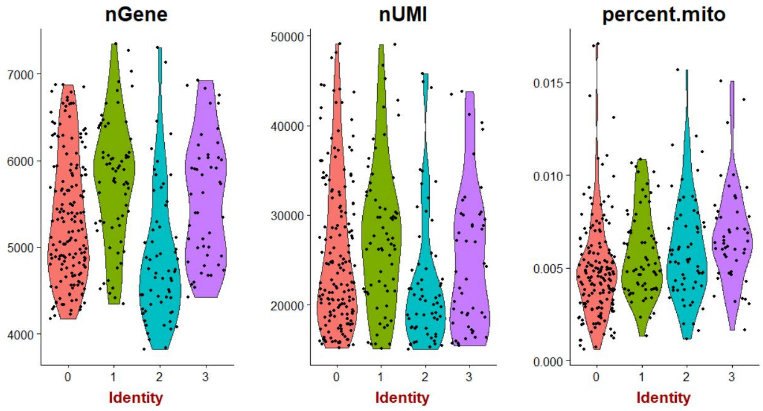


Figure 23: The distribution over the number of genes captured, number of unique molecular identifiers and the percentage of mitochondrial genes captured in POST, Patient 9, based on the clustering in figure 21

5.2.3 Cell type identities

In order to place some intuitive cell type identities on the clusters, the combination of marker genes had to be understood. By focusing on the role of the top 10 expressed marker genes of each cluster, some observations were made.

- **POST_0** (165 cells) contained high expression of some genes, such as FN1, THBS2 and DCN, which are known to have inhibitor effects on the tumour growth. The cluster also contained high expression of genes correlated with tumour growth, such as FGF7 and PAPPA.

- **POST_1** (67 cells) is likely to be formed of cells undergoing mitosis. This is because of the high expression of NUSAP1, MKI67, SMC4, ANLN and others, which are protein encoding genes with roles in the assembly of the mitotic spindle or chromosome segregation during mitosis. The cells in this cluster are more likely to be in a *state* of a cell cycle, rather than a separated type of cells. However, this is a hypothesis, which needs further verification. It is noticeable that this group of cells presents high expression of gene TOP2A, whose mutations are associated with cancer drug resistance.

- **POST_2** (61 cells) contained the genes TYROBP, SPP1 and CHI3L1 with high expression. These genes are associated with the immune system (T-helper cells and killer cells). The cluster also contained markers of tumour-associated tissue remodelling and tumour genesis, such as LCP1 and MMP9.

- **POST_3** (48 cells) presented marker genes related to kinases with role in cell surface structural adhesion and angiogenesis, such as EFNB2, EZR and THY1. The cluster contained high expressions of genes IGFBP2 and ALDH1A3 which are associated with tumour growth. EZR and KRT19 are also known to be highly expressed in cancerous cells.

All the clusters have some marker genes which are strongly linked to cancer and some present tumour supressor encoding genes. However, the only clear cell-type identity that can be inferred from the gene definitions is that of cluster POST_1 as being composed of highly proliferative cells.

6 Cell-type progression

Following the classifications of both PRE and POST datasets, the cell types were compared. The purpose of the comparison was to identify the progession of a cell type from the PRE dataset into one or more cell types from the POST dataset. If the progression is identified, then the efficiency of chemotherapy on each type can be assessed.

This section will go through the influence of data quality and quantity differences over the process of computing a progression and present three methods for comparing the datasets. The first method is based on the intuition behind the data; the second method consists of using t-SNE on the combined datasets; the third method is a more mathematical approach to differences in gene expressions.

6.1 Data quality and quantity

In order to make a meaningful mapping of cell types from the PRE dataset to the POST dataset, the data must be compared in quality in quantity. Significant differences might influence the progression, which is why their impact should be minimised.

In terms of data quality, both PRE and POST have similar a ratio of number of unique molecular identifiers to number of genes and number of unique molecular identifiers to percentage of mitochondrial genes (see figures 15 and 20).

In terms of data quantity, the PRE dataset is almost twice as large as the POST dataset (in number of cells). This is quite a significant difference which might influence the result of the mapping. In order to overcome this issue, all entries in the DGE matrices have been scaled to library size and multiplied by a large constant. Library size scaling is a known method for normalizing two datasets, to allow for comparison independent of their size [30].

The method involved dividing all DGE matrix entries by the total number of reads. In this way the read counts for every cell and gene were scaled with respect to the total number of reads identified in the dataset. The numbers obtained after applying library size scaling are very small and were multiplied by an arbitrary large constant (100,000) for clarity.

6.2 Intuitive cell type progression

Before beginning to make any clear mapping, regardless of the method, it is important to place some intuition on what the cell type progression is. Using the information about the genes expressed in each cell type and some background knowledge, some of the clear mappings can be made. For example, if a cluster from PRE has the same marker genes as a cluster from POST, it can be deduced that the specific type of cells survived chemotherapy.

However, this intuitive mapping will not provide very accurate insight. Most times, the connections between clusters are hidden behind large amounts of data and therefore hard to see at a first glance (without any thorough analysis). The purpose of using the intuition to first map the cell types is to be able to verify the results of the next methods.

On the PRE and POST datasets of Patient 9, looking at the data gave one clear cluster to cluster mapping. The types PRE_1 and POST_1 share 8/10 of the Seurat marker genes, with very high expression in both cases. They are also very close in size, 87 and 67 cells respectively. The first conclusion that can be drawn from this is that the chemotherapy had little to no effect on type PRE_1, because the same type can be identified in POST_1 and it is 22% smaller. During the classification, both of these clusters were identified with a possible state of the cell cycle, meaning that the cells themselves do not form a *type*, but are in a certain *state* of a cell cycle. This would mean that the clusters are not necessarily formed by the same cells. Even if they represent only a state, any mapping based on the marker genes should identify them as being very close together. Clusters PRE_1 and POST_1 can therefore be used as a verification of the mapping.

Another observation that is worth keeping in mind when placing an intuitive mapping on the cells is that cancer stem cells survive chemotherapy. This is the main reason why most cancers return [29]. Intuitively, since PRE_3 was labelled as cancer stem cells, the same cells should have survived chemotherapy and should be found in POST. One issue with intuitive mapping is that not all the data can be analysed, it is too large of a dataset for a person to simply observe, without the aid of computational methods. Looking at just the marker genes of each POST cluster, there are no immediate labels of cancer stem cells, nor similarities with PRE_3. However, this is simply the beginning of the mapping and more insight into the data should be gained after using more advanced methods.

The only mappings that can be made intuitively are:

$$PRE\_1 \implies POST\_1$$
$$PRE\_3 \implies POST\_?$$

6.3 Data merging and re-classification

One known method for comparing datasets is to combine them and then re-classify, using the same classification algorithms as before. If two clusters from different datasets are actually from the same type, they will be classified together. Applying this method on PRE and POST together gave the results shown in figure 24. The datasets were filtered separately and the combined into a new dataset, thus ensuring that the same cells from the individual classifications are kept into the combined one.

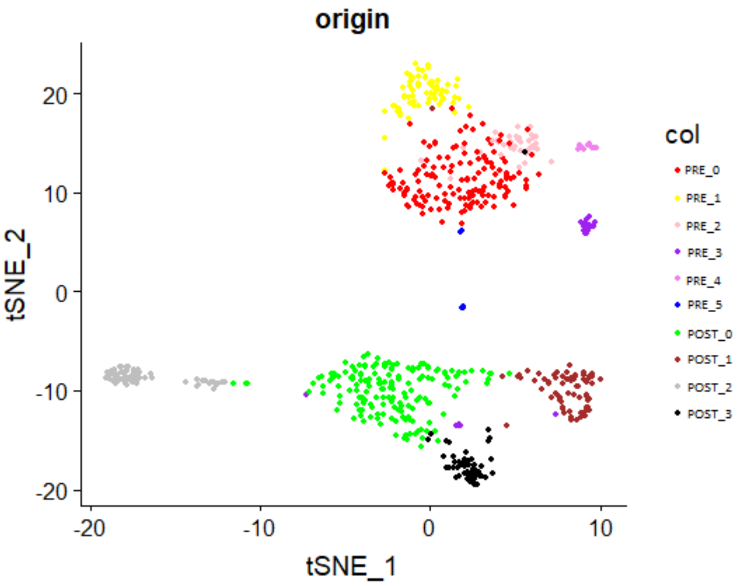


Figure 24: t-SNE run for Patient 9 data, after combining the filtered cells from the PRE and POST analysis with library size scaling. Colours show true provenience of data and position shows re-classification

Figure 24 shows that running t-SNE on the combined datasets does not identify any similar trends in PRE and POST data. The two datasets were clustered separately, maintaining the initial classifications. This gave a total of 10 clusters, which are exactly the 6 PRE clusters and the 4 POST clusters identified in the individual classifications.

The re-classification shows that there are no connections between any clusters from PRE to clusters from POST. However, the intuitive mapping placed in the above section shows that PRE_1 and POST_1 are very similar and should have been clustered together. There could be many reasons for why this did not occur, but it is likely that there are other trends in each dataset, not relevant to the marker genes, which the algorithm intercepts and therefore separates the datasets.

There have been previous experiments, prior to the start of this project, which combined the datasets taken from the same patient, at the same time, with different resolutions. In this case t-SNE completely separated the two datasets, but identified the same classification in each. This information already supported the idea that the re-classification method might not be very efficient when combining datasets.

The re-classification method for the combined dataset (PRE and POST) showed no conclusive mapping or cell progression.

6.4 Similarity quantification

The intuition behind the data showed that there is at least one pair of "similar" clusters: PRE_1 and POST_1, because they share almost the same markers. However, the method described in the previous section does not find any similarities between clusters from the PRE dataset and the POST dataset. For this reason, a new approach was taken at determining the similarity between clusters, by looking at raw average expressions of marker genes.

In order to reason about the mappings of one cell-type from PRE to a cell-type from POST, the notion of how similar each pre-cluster is to each post-cluster, must be defined. Hence, computing the cell-type progression has been reduced to the following problem, done for each pair of clusters.

6.4.1 Method development

Question: Given two cell clusters, C1 and C2, how similar are they?
Input:

- DGE_C1 = DGE matrix corresponding to C1.
- DGE_C2 = DGE matrix corresponding to C2.
- MARKERS_C1 = The list of marker genes identified by the Seurat analysis for C1.
- MARKERS_C2 = The list of marker genes identified by the Seurat analysis for C2.

Parameters:

- N = number of top marker genes to be considered for computing the similarity. This parameter can be set to -1, if all marker genes should be considered.

Output:

- L = loss. The amount of expression that is lost from C1 to C2.
- S% = similarity %. The percentage of similarity between C1 and C2. How similar is C2 to C1?

Similarity formula

The formula for quantifying the similarity between clusters C1 and C2 goes through 3 main steps. Firstly, the raw average expressions for each of the marker genes must be calculated in each one of the clusters and the top N most expressed genes in C1 must be determined. Then the differences in raw average expression for each gene from C1 to C2 are summed, giving an overall loss. Since the actual value of the loss is quite difficult to interpret, based on the maximum possible loss, the percentage of actual loss is computed. The similarity is constructed by reversing the percentage of loss.

Calculating Average Expression of Genes

For each gene from MARKERS_C1, the average expression over all cells in C1 and C2 respectively must be calculated.

$$\mu_{C1}[GENE] = \frac{1}{|C1|}\sum_{c \in C1}(DGE_{C1}[GENE][c]) \tag{1}$$

Equation (1) is simply saying that the average expression for a gene across C1 is the sum of the expression of that gene in all the cells from C1, divided by the total number of cells in C1.

Similarly we can calculate the average expression of a gene in C2.

$$\mu_{C2}[GENE] = \frac{1}{|C2|}\sum_{c \in C2}(DGE_{C2}[GENE][c]) \tag{2}$$

At the end of this section, a table with the following headers should be computed. This contains information about the average expression of all marker genes of C1, in C1 and C2 respectively.

$GENE$	$\mu_{C1}[GENE]$	$\mu_{C2}[GENE]$

Top N marker-genes of C1

The formula explained takes a parameter N, the number of genes to be considered when computing the similarity. This section should explain how to generate the list of N genes, that have the highest average expression, across all marker genes in C1.

Let TOP_N be the list of the top N marker-genes of C1. In Seurat these are the genes which will have the highest avg_logFC and can be obtained by running the following code, where dataobj is the Seurat object containing the analysed data.

```
topN <- dataobj.markers %>% group_by(cluster) %>% top_n(N, avg_logFC)
```

Note that these genes are not necessarily the ones with highest μ_{C1} (average expression in C1), since the formula used to calculate avg_logFC is not linear in terms of average raw expressions.

At the end of this section, a list of genes, TOP_N has been calculated, containing the genes with the top average expression over all marker genes in C1.

Calculating the loss L

The loss L denotes the actual quantity of gene expression that was lost from C1 to C2, with respect to the marker genes of C1. In other words, L is a quantification over the average number of reads that are in C1, but are missing from C2.

Firstly we must calculate the loss in expression for every gene separately, as shown in equation (3). In order to do so, we must calculate the difference in expression of every specific gene, from C1 to C2, by using the average expressions calculated above. One important note is that this difference will not always be a positive number, because some genes might be more expressed in C2, than in C1. In order to overcome this issue, we could either compute the squared differences, or simply round-up all negative numbers to 0.

$$loss[GENE] = max(0, \mu_{C1}[GENE] - \mu_{C2}[GENE]) \tag{3}$$

The above formula will consider all genes which have a higher expression in C2 than in C1, to constitute of 0 loss. The main reason behind this assumption is that in a case of such a gene, there was no actual expression “lost” from C1 to C2, in fact there is now more expression of that gene in C2. Another reason for considering negative values as 0 is because this allows for computing a percentage of similarity, whereas negative values would have inconvenienced the calculations.

The overall loss L is calculated as the sum over the loss of all the genes in TOP_N, as shown in equation (4) below.

$$L = \sum_{G \in TOP\_N} loss[G] \tag{4}$$

At the end of this section, a number L denoting the loss of expression from C1 to C2 has been calculated. It is important to notice that L is a positive number, that will always be less than the sum of all average expressions of the top N genes in C1.

Calculating the similarity S%

In order to compute the similarity S for C1 and C2, the range of loss must be determined. Intuitively, knowing what interval L belongs to should be enough for matching L to a percentage.

The best-case scenario is that there was no lost expression from C1 to C2 (L=0), which can happen if and only if:

$$\mu_{C2}[g] >= \mu_{C1}[g], \forall g \in TOP\_N \tag{5}$$

In this case the loss L will always be 0, because the difference $\mu_{C1}[g] - \mu_{C2}[g]$ will be 0, for all $g \in TOP\_n$, thus having a final sum $L = 0$.

The worst-case scenario is that there is no expression retained from C1 to C2, which can happen if and only if:

$$\mu_{C2}[g] = 0, \forall g \in TOP\_N \tag{6}$$

In this case the difference $\mu_{C1}[g] - \mu_{C2}[g]$ will be $\mu_{C1}[g]$ for all $g \in TOP\_n$, thus having a final sum L as the sum of all average expressions in C1 of genes in TOP_N.

$$L = \sum_{g \in TOP\_N} (\mu_{C1}[g] - \mu_{C2}[g]) = \sum_{g \in TOP\_N} (\mu_{C1}[g] - 0) = \sum_{g \in TOP\_N} \mu_{C1}[g] \tag{7}$$

After calculating the best and worst case scenario, a lower bound and upper bound can be set for L.

$$L \in (0, \sum_{g \in TOP\_n} \mu_{C1}[g]) \tag{8}$$

Since there is a bound on L, the percentage of loss can be determined, i.e. what percentage of the total possible loss is L:

$$L\% = \frac{L * 100}{MAX_L} = \frac{L * 100}{\sum_{g \in TOP\_n} \mu_{C1}[g]} \tag{9}$$

After calculating L %, the percentage of expression that is lost out of the total expression of C1, a natural way to define the similarity would be to consider the amount of retained expression (not lost) to be the value of similarity.

$$S = 100 - L\% \tag{10}$$

At the end of this subsection, the number S has been calculated, denoting the percentage of similarity between clusters C1 and C2. This value gives raise to a natural and simple interpretation since it can be compared across different pairs of clusters with no scaling needed.

Final expression

After following all the above steps, the formula for computing the similarity between C1 and C2 is:

$$\begin{aligned}
S &= 100 - L\% \\
S &= 100 - \frac{L * 100}{\sum_{g \in TOP\_n} \mu_{C1}[g]} \\
S &= 100 - \frac{(\sum_{g \in TOP\_N} loss[g]) * 100}{\sum_{g \in TOP\_n} \mu_{C1}[g]}
\end{aligned}$$

$$S = 100 - 100 * \frac{(\sum_{g \in TOP\_N} max(0, \mu_{C1}[g] - \mu_{C2}[g])}{\sum_{g \in TOP\_n} \mu_{C1}[g]}$$

$$S = 100 * \left(1 - \frac{(\sum_{g \in TOP\_N} max\,(0, \mu_{C1}[g] - \mu_{C2}[g])}{\sum_{g \in TOP\_n} \mu_{C1}[g]}\right)$$

Observations

1. **Non-symmetric.** Since the TOP_N list, which is used in the formula, will vary from cluster to cluster, the results of the similarity are non-symmetric. In other words the S% from C1 to C2 is different from the S% from C2 to C1.

2. **Influence of parameter N.** As the parameter N is increased, the similarity will most likely decrease, since we are considering more genes, which will probably bring an addition to the overall loss. Based on testing, the most efficient use of the parameter N is with the value of 10.

3. **Library-size scaling.** One problem that might arise when applying the formula as previously explained is that sometimes one of the clusters might come from a dataset with an unexpectedly large/small number of reads. In this case the loss will be very large, but most of the difference could come from the difference in data extraction methodologies. To overcome this problem, the DGE matrices should be scaled by library size. (All read counts should be divided by the number of total reads in the dataset and multiply them by a large constant, which should be consistent across datasets).

4. **Optimisation.** One optimisation that could be done when implementing the above-described method comes from the observation that raw average expressions need not be computed for all genes, but only for the genes in TOP_N.

5. **Average over total.** The reasoning behind using average expression values over actual total expression values is that the similarity should be independent of cluster sizes.

6.4.2 Results

The pairwise similarities for clusters from the pre-treatment dataset and post-treatment dataset of Patient 9 can be seen in the tabels below. As explained in the previous section, the similarity percentages differ based on the choice of TOP_N genes. Figure 25 uses to top 10 marker genes from each PRE cluster and compares the expression of the same genes in POST clusters.

Since the formula is not symmetric, there will be different results for using the same formula on the top 10 marker genes from POST clusters and their expression in the PRE clusters. These results are shown in figure 26.

	POST_0	POST_1	POST_2	POST_3
PRE_0	12.67	0	4.93	5.08
PRE_1	0	95.63	0	0
PRE_2	28.35	0	4.77	0.95
PRE_3	58.2	0	1.31	23.42
PRE_4	3.65	1.53	2.83	0
PRE_5	2.59	1.42	0	1.37

Figure 25: Table of similarity percentages, using gene expression from clusters in PRE to clusters in POST; Patient 9 mapping of cell types, using the top 10 marker genes from PRE clusters.

	PRE_0	PRE_1	PRE_2	PRE_3	PRE_4	POST_5
POST_0	0	2.94	0	18.74	0	6.69
POST_1	0	72.51	0	0	0	0
POST_2	0	0	0	0.93	1.11	0
POST_3	13.69	0	0	29.4	0	2.19

Figure 26: Table of similarity percentages, using gene expression from clusters in POST to clusters in PRE; Patient 9 mapping of cell types, using the top 10 marker genes from POSt clusters.

The most relevant mappings are the one given by figure 25 because the question asked is *"What form does PRE_i take in POST?"*. However, the mappings in figure 26 are used as a form of confirmation that the mappings do stand. For eaxample, cluster POST_0 maps to PRE_0 (12.67%), PRE_2 (28.35%) and PRE_3 (58.2%) (see figure 25). But in figure 26, it maps to PRE_3. This could be because POST_0 has a wide range of genes expressed. For this reason, POST_2 is only mapped to PRE_3 further, since it is the only mapping maintained from PRE to POST and vice-versa.

Likely mapping of clusters

For POST-treatment clusters:

- POST_0 came from PRE_3.
- POST_1 is PRE_1.
- POST_3 came from PRE_3.

The only POST-treatment cluster that remains unaccounted for is POST_2. Comparing every cell of POST_2 with the average cell of each PRE-treatment cluster, the composition below was obtained.

$$POST\_2 = (4.85\% * PRE\_0) + (0\% * PRE\_1) + (4.61\% * PRE\_2) + (1.31\% * PRE\_3) \\ + (2.86\% * PRE\_4) + (0\% * PRE\_5)$$

These results are within a 0.5 error margin from the similarity presented at the start of this results section, from the pre-treatment clusters to POST_2. The hope of computing this cell-by-cell composition was to determine if each cell from POST_2 came from a different cluster of pre-treatment data. Unfortunately, the results show that each cell has an homogenous composition of pre-treatment clusters, meaning that it is likely that the cells are changed beyond recognition and therefore cannot be mapped to any cells in the pre-treatment dataset.

For PRE-treatment clusters:

- PRE_0 and PRE_4 together form POST_2.
- PRE_1 is POST_1.
- PRE_2 was removed.
- PRE_3transformed and split into POST_0 and POST_3.
- PRE_5 was removed.

Final mapping

Using the newly designed mathematical method, the following mapping was determined:

$$\begin{aligned} PRE\_1 &\implies POST\_1 \\ PRE\_3 &\implies POST\_0; POST\_3 \end{aligned}$$

This mapping confirms the intuition that PRE_1 and POST_1 are similar, as well as confirming the fact that cancer stem cells can still be found in post-treatment cells.

6.5 Method comparison

Method I (Intuitive mapping) gives some results, but very few. It can only identify the really obvious connections, such as identical clusters. The only consistent information obtained after applying this method to Patient 9 data is that PRE_1 and POST_1 are the same type or state of the cell cycle.

Method II (Data merging and re-classification) gives no real results regarding the mapping. It completely separates the data, making it difficult to see any connections between clusters from different datasets. These connections are overshadowed by difference between the datasets themselves.

Method III (Similarity quantification) gives more results than the other methods and verifies the intuition behind the data. It maps PRE_1 and POST_1 with high similarity and relates POST_0 and POST_3 to PRE_3, with higher similarities than any other types. This method also adds exact numbers on how similar any two clusters are, providing a good mathematical way of assessing chemotherapy efficiency.

The data merging and re-classification method that has been used before was ineffective. The similarity quantification method adds more information onto the cell type progression and the results are verified by the intuition behind the data.

6.6 Assessment of chemotherapy effectiveness

In order to assess the effect of chemotherapy on a patient, a number of situations must be considered. The effect of chemotherapy is an overall value of effects on each cell type contained in the PRE dataset. In order to determine the effect on a cell type, it is important to look over all possibilities regarding what happened to that cell type, while linking it to the existing POST cell types.

- PRE_i exactly maps to $POST\_j$.
    - $|PRE\_i| \approx |POST\_j| \implies$ The chemotherapy stopped the growth of type PRE_i.
    - $|PRE\_i| << |POST\_j| \implies$ The chemotherapy had no effect on type PRE_i
    - $|PRE\_i| >> |POST\_j| \implies$. The chemotherapy shrunk type PRE_i

- PRE_i maps to no types in $POST$.
  - Type PRE_i changed beyond recognition.
  - $|PRE\_i|$ is large. $\implies$ The chemotherapy removed type PRE_i.
  - $|PRE\_i|$ is small. $\implies$ The chemotherapy either removed type PRE_i or type PRE_i was represented an oddity during extraction/ classification.
- PRE_i maps to types: $POST\_j_1, POST\_j_2....POST\_j_k$.
  - $|PRE\_i| \approx \sum_{q=1}^{k} |POST\_j_q| \implies$ The chemotherapy had no effect on type PRE_i, but the classification of $POST$ separated the cells in multiple types; or groups of cells in PRE_i mutated in different ways during chemotherapy.
  - $|PRE\_i| << \sum_{q=1}^{k} |POST\_j_q| \implies$ Type PRE_i continued to grow and mutate under chemotherapy, creating the different types $POST\_j_1, POST\_j_2....POST\_j_k$.
  - $|PRE\_i| << \sum_{q=1}^{k} |POST\_j_q| \implies$ The chemotherapy had an impact on type PRE_i, but did not eradicate it completely. Some of the remaining cells mutated during treatment.

All of the above are intuitive interpretations of cell progression, based on classification similarities. However, these are not the only possibilities and should not be treated as facts.

In the case of Patient 9, the mapping suggests that there was no impact on type PRE_1, but the growth stagnated and that type PRE_3 continued to grow under treatment and develop 2 types of mutations (POST_0 and POST_2). Type PRE_1 seemed to be closer in meaning to a state of cells, not a type. If this is indeed the case, there are no conclusions to be drawn about cells in PRE_1, since there is little information about what type each cell belongs to.

Taking into account the cluster labelling described in sections 5.1.3 and 5.2.3 and the cell type progression described in section 6.4, some conclusions can be drawn about the effect of chemotherapy on Patient 9. In the PRE-chemotherapy data, there were only 27 cancer stem cells, out of a total 333 cells. These cells are only a subset of the set of tumour cells from PRE, but it can be assumed that the distribution of the cancer stem cells across the subset is consistent with the tumour. Extrapolating, there are approximately 8% cancer stem cells in the tumour. These cancer stem cells were not immediately obvious in the POST-chemotherapy data, but after applying the similarity quantification formula, two POST clusters were labelled as cancer stem cells variants. One of them, POST_0 contained about 58.2% of the gene expression in the cancer stem cells (PRE_3). The second one, POST_3 retained approximately 23.42% of the gene expression of the same cancer stem cells.

The POST-chemotherapy data had 165 cells which retained 58.2% cancer stem cell expression and 48 cells which retained 23.42% cancer stem cell expression. If generalised to label these two clusters as cancer stem cells, it can be noticed that there was a significant increase in the percentage of cancer stem cells from before treatment to after treatment data. These cells used to account for only 8% of the tumour, but after treatment they represent 62% of the tumour.

Considering the percentage of expression that these cells retain from the cancer stem cells, different calculations can be performed. If 165 cells contain 58.2% of the cancer stem cells, it can be deduced that the genomic information retained in them is equivalent to about 96 cells. Similarly, for the 48 cells with 23.42% maintained expression, it can be said that the amount of cancer stem cells genomic information in them is the equivalent of 11 cells. In total this would mean that the tumour contains cancer stem cell expression that is equivalent to 107 cells, showing that the tumour is composed of approximately 31.37% cancer stem cell expression. However, the most likely interpretation is however that both clusters from the POST set represent derivatives of the cancer stem cells, but which have partly changed in expression. This means that the cancer stem cells represent more than half of the tumour cells in the POST-chemotherapy data.

The interpretation of these numbers clearly shows that in the case of Patient 9, the cancer stem cells account for a much larger part of the tumour after chemotherapy than they did before.

6.7 Summary

The two datasets, PRE and POST chemotherapy, have been extracted from the same tumour. It is a fair assumption that these sets of data actually show the same cells, but which have gone through changes during treatment. This section was concerned with understanding how the cell-types from the PRE dataset have progressed into the cell-types from the POST dataset. Based on this *cell-type progression*, the effect of chemotherapy can be measured in terms of the number of surviving cancer stem cells.

To create a mapping of PRE cell-types to POST cell-types, both datasets were scaled down to library size (to eliminate differences due to average read count). The first method used consisted of creating an intuitive mapping between cell-types, based on the top 10 marker genes expressed in each. If two clusters share most of the marker genes, then they are mapped. Observing these genes resulted in one conclusive result: PRE_1 is very similar to POST_1. There are obvious drawbacks of this method since there are most likely more patterns to be observed than can be spotted at a first glance. However, the similarity between PRE_1 and POST_1 has been used as a form of validation for further methods.

The second method used is a simple approach of merging and re-classifying the data. This consists of taking the two datasets, after filtering was applied and merging the DGE matrices into a single matrix. Using the same algorithms, another classification is obtained, this time consisting of clusters with cells from either or both datasets. In this particular case, each cluster was entirely corresponding to individual cell-types from the previous classifications. T-sne was distinguishing between cells coming from PRE and cells coming from POST treatment. This method failed to see that PRE_1 and POST_1 are very similar clusters and therefore it could not be used to draw any conclusions.

Because of the inconclusive results obtained with the second method, a new one was developed as part of this project, by inventing a new mathematical formula. The reasoning behind it and how it was developed for this project are described in section 6.4, along with the main results after applying it. This formula takes two clusters and quantifies the amount of genetic expression that is lost from the average cell of a cluster in the average cell of the other cluster. Having an exact number of lost expression can trivially give a percentage of loss between clusters and therefore a percentage of similarity between them. Applying this formula pairwise resulted in a clear mapping of approximately 95% similarity between PRE_1 and POST_1. This validated the method, since it was perfectly aligned with the intuition behind the data. Some other very interesting results have surfaced from using this method, which were otherwise not discovered. The cancer stem cells (PRE_3) seem to have remained after chemotherapy and separated into two other cell-types (POST_0 and POST_3). This also confirms the existing intuition that cancer stem cells survive chemotherapy, while also shedding some light onto *how* they do so.

Using the results from the newly designed method, an assessment of chemotherapy effectiveness can be performed. Because the cancer stem cells are the main cause of cancer relapse, an effective treatment would eliminate those. However, in the case of Patient 9 it appears that the percentage that the cancer stem cells represent in the tumour has grown significantly, from 8% to 31%. Without having the method designed in section 6.4, it would be impossible to determine if the treatment has eliminated the cancer stem cells, since these are not easily identifiable in the POST dataset.

7 Gene mutations

Using the bamCleave Splitter (described in the Method Development section of this report) the BAM files corresponding to PRE and POST datasets were split into cell-type specific files. This allowed for the exploration of gene mutations in each type, with the hope of identifying some type-specific mutations or confirming the labelling and cell-type progression of the identified clusters.

There were two other BAM files made available as part of this project, with information from healthy (germline) and cancerous tissue taken in bulk from the same patient. The purpose of using these two datasets was to determine which mutations should disappear after treatment. For example, if a certain mutation is in the cancerous tissue, but not in the healthy tissue, it is highly likely that targeting the mutation will be an improvement to the treatment. Identifying such mutations will help understand and assess the effect of chemotherapy on a given patient. Identifying such mutations in each type will help understand and assess the effect of chemotherapy on each specific type and therefore start designing type specific treatments.

The genome is incredibly wide and investigating all mutations in all genes would be incredibly time-consuming and it will likely not lead to many relevant conclusions. It is important to focus the resources on exploring known genes that have strong correlations to cancer and one of these genes in P53, which will be the starting point for the identification of gene mutations. Gene P53, also known as TP53 (Tumour Protein 53) is a gene located on the 17th chromosome, which encodes a protein with the role of regulating division of cells, keeping them from growing too fast, to slow or in an uncontrolled way [31]. Therefore, this gene should be the "guardian of the genome", keeping tumours from forming. Since tumours have in fact formed in the organism, it is likely that the gene has been damaged or mutated in ways that keep it from performing it's normal functions.

7.1 P53 in cancerous and germline data

As an initial part of this section, certain mutations have been identified in the P53 gene of the cancerous data and then searched in the germline data. It has been noted that not many mutations have been consistent across the two datasets, but in fact both display various mutations that cannot be found in the other. One other important observation is that the cancerous dataset has more reads of gene P53 than the germline dataset, but this could be down to the extraction of data.

There are many small mutations which occur in only two or three reads, so these have not been taken into consideration, since they could simply be noise. However, there are a number of mutations with much wider expression, which have been detailed below and investigated in both the cancerous and the germline datasets. Even though this occur with a high number of reads, it is possible that they are simple P53 variants which have no connection to the tumour growth.

Mutation 1 (position 7,578,115 - intron - with Cytosine instead of Thymine)

This mutation was found in both datasets, cancerous and germline. The position of the genome had a number of 744 reads in the cancerous dataset and only 21 in the germline dataset. Out of the 744 reads in the cancerous data, 733 (99%) were consisting of Cytosine and only 8 (1%) were consisting of Thymine. In the germline data, all 21 (100%) reads were of Cytosine.

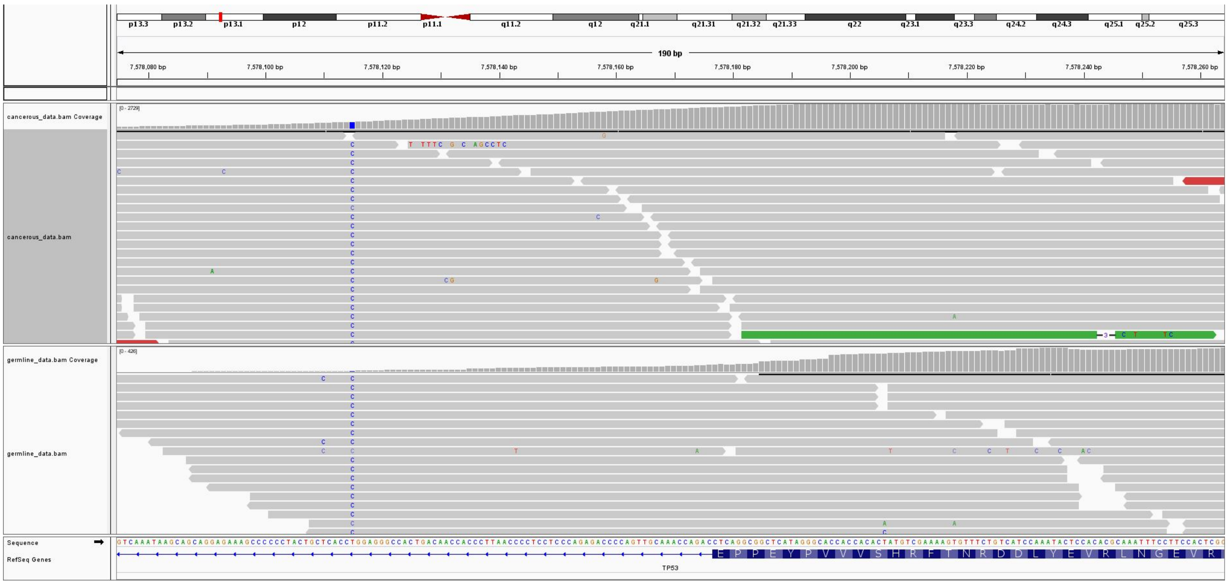


Figure 27: P53 mutation (blue) in cancerous (above) and germline (below) datasets from the same patient. Data viewed with IGV. The same mutation is consistent across datasets, but more prominent in the cancerous data.

Figure 27 shows the mentioned genome section viewed with IGV (Integrated Genomics Viewer). It can be noticed that the same mutation, marked with the blue C appears in an area of many reads for both datasets, but otherwise mostly clean. This shows a reduction of the mutation in the healthy tissue, meaning that the treatment should aim to reduce this specific mutation.

Mutation 2 (position 7,578,645 - exon - with Thymine instead of Cytosine)

This mutation was found in both datasets, cancerous and germline. The position of the genome had a number of 2,765 reads in the cancerous dataset and 183 in the germline dataset. Out of the 2,765 reads in the cancerous data, 2,727 (99%) were consisting of Thymine and only 12 (0%) were consisting of Cytosine. In the germline data, out of the 183 reads, 182 (99%) were of Thymine and only 1 (1%) of Cytosine.

This mutation should also be reduced considerably by the treatment, since it is found in large quantities in the cancerous data and smaller quantities in the germline data.

Mutation 3 (position 7,578,837 - intron - with Guanine instead of Adenine)

This mutation was only found in the cancerous dataset, with 258 reads. Out of these reads, 257 were of Guanine (100%) and only 1 (0%) of Adenine.

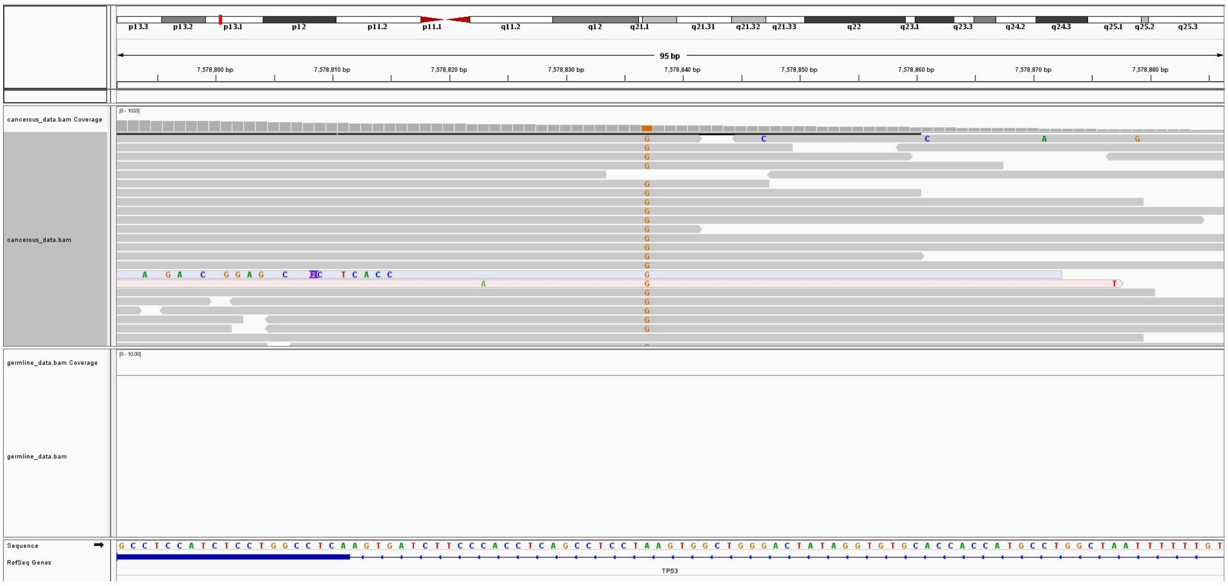


Figure 28: P53 mutation (yellow) in cancerous (above) and germline (below) datasets from the same patient. Data viewed with IGV. The mutation is present in the cancerous dataset, but not existent in the germline dataset.

The mutation is present in the cancerous dataset and not present in the germline dataset. This could mean that the mutation should be entirely cured by the treatment. However, since there are no reads for this sequence of the genome in the healthy data, it is possible that the mutation existed, but was not intercepted by the extraction method.

7.2 P53 in PRE-chemotherapy data

The BAM file analysed in this section has genome-wide reads and mutations from the same patient, from the tumour before chemotherapy. The analysis was focused on observing if the same three mutations identified in the healthy and cancerous tissue above are also found in this PRE dataset and in which specific groups. If the PRE - POST mutations correspond to the Cancerous - Germline mutations, then the chemotherapy treatment has worked as expected.

Mutation 1 (position 7,578,115 - intron - with Cytosine instead of Thymine)

This mutation was also found in the PRE-chemotherapy dataset, showing that it is indeed a sign of unhealthy P53. In the PRE dataset, this mutation was only manifested in type PRE_0 with 14 reads, all of them being of Cytosine. None of the other types had any reads at this position, so it is impossible to say if the mutation was manifested in the others as well.

Mutation 2 (position 7,578,645 - exon - with Thymine instead of Cytosine)

There were no reads at this position in any of the 6 PRE cell-types.

Mutation 3 (position 7,578,837 - intron - with Guanine instead of Adenine)

There were no reads at this position in any of the 6 PRE cell-types.

Other mutations (positions 7,577,112 - 7,577,155)

The PRE dataset had reads in this area of the gene P53, most of them in type PRE_0. This could be because type PRE_0 is larger in size than all the others. There were some reads in the other types as well, except for PRE_2 which had no reads in this area.

In figure 29 four mutations expressed in four of the six PRE types can be noticed. These are on the same position, so most cells present it, regardless of their type. PRE_2 has no reads in this area and PRE_3 simply has no mutations. Types PRE_0, PRE_4 and PRE_5 have the exact same four mutations, but type PRE_1 has them in much lower quantities and more impure. This mutation should be observed in the POST types to determine the effect of chemotherapy on it in the different types of cells.

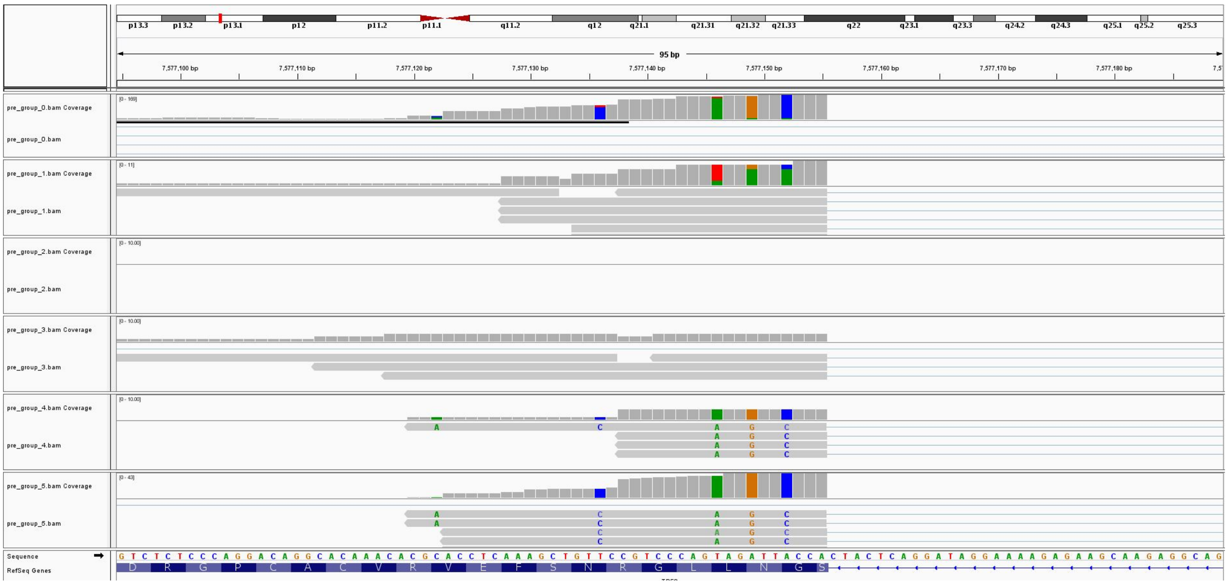


Figure 29: P53 mutations in the PRE dataset, specific to each type (0 above to 5 below). There are four mutations (blue, green, orange and blue) which distinguish between individual types.

7.3 P53 in POST-chemotherapy data

The BAM file analysed in this section has genome-wide reads and mutations from the same patient, from the tumour after chemotherapy. The analysis was focused on observing if the same three mutations identified in the healthy and cancerous tissue above are also found in this POST dataset and in which specific groups. Especially it was important to see if the mutations identified in the PRE dataset have persevered after treatment or have disappeared.

Mutation 1 (position 7,578,115 - intron - with Cytosine instead of Thymine)

This mutation was also found in the POST-chemotherapy dataset, showing that it is indeed a sign of unhealthy P53. In the POST dataset, this mutation only manifested in type POST_0 with a total of 3 out of 3 mutated reads. There are no reads for this position in any of the other types, so it is unsure if the mutation is non existent or just not captured.

Mutation 2 (position 7,578,645 - exon - with Thymine instead of Cytosine)

There were no reads at this position in any of the 4 POST cell-types.

Mutation 3 (position 7,578,837 - intron - with Guanine instead of Adenine)

There were no reads at this position in any of the 4 POST cell-types.

Other mutations from PRE (positions 7,577,112 - 7,577,155)

The POST dataset has reads in this area of the gene P53, most of them from POST_0. There one mutation present (position 7,577,155), but none of the ones identified in the PRE dataset (see figure 29). This could mean that the mutations have disappeared as an effect of chemotherapy or other factors. It is relevant to mention that the number of reads for this area of the genome is much smaller in the POST dataset. There are a maximum of 27 reads in POST, whereas PRE had around 170.

Other mutations from PRE (positions 7,573,927 - 7,574,003)

This section of the gene did not seem particularly interesting when analysing the PRE dataset, since the mutations are not present in many reads, but looking at the POST dataset, it becomes obvious that all these mutations have been removed by the treatment. This is a good example of showing taht the effect of chemotherapy was positive on all types.

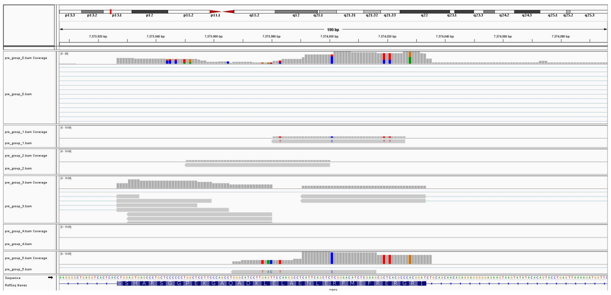


Figure 30: Sequence of gene P53 in the PRE dataset, with reads separated by types (0 above to 5 below). The colours represent mutations in the genome and the grey areas represent reads.

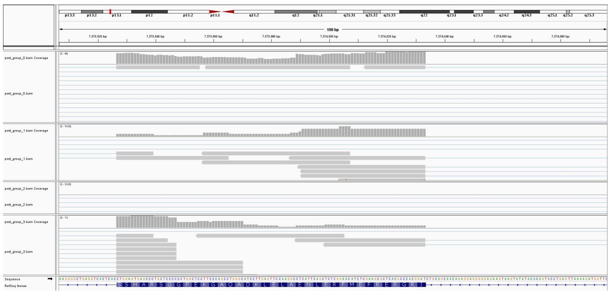


Figure 31: Sequence of gene P53 in the POST dataset, with reads separated by types (0 above to 3 below). The colours represent mutations in the genome and the grey areas represent reads. No visible mutations.

Other observations

It has been observed that throughout the genome there are very few reads of gene P53 for types POST_1 and POST_2, almost none in comparison to POST_0 and POST_3. This could be due to the size of POST_0 (being much larger than the other types), but it is relevant to mention that these are the types that have been mapped to the cancer stem cells by the cell progression mapping from previous sections. This gives some confidence in the cell progression, because it could mean that the protein generated by gene P53 is more active in these types as response to the uncontrollable growth of cells, as a suppressor. This would

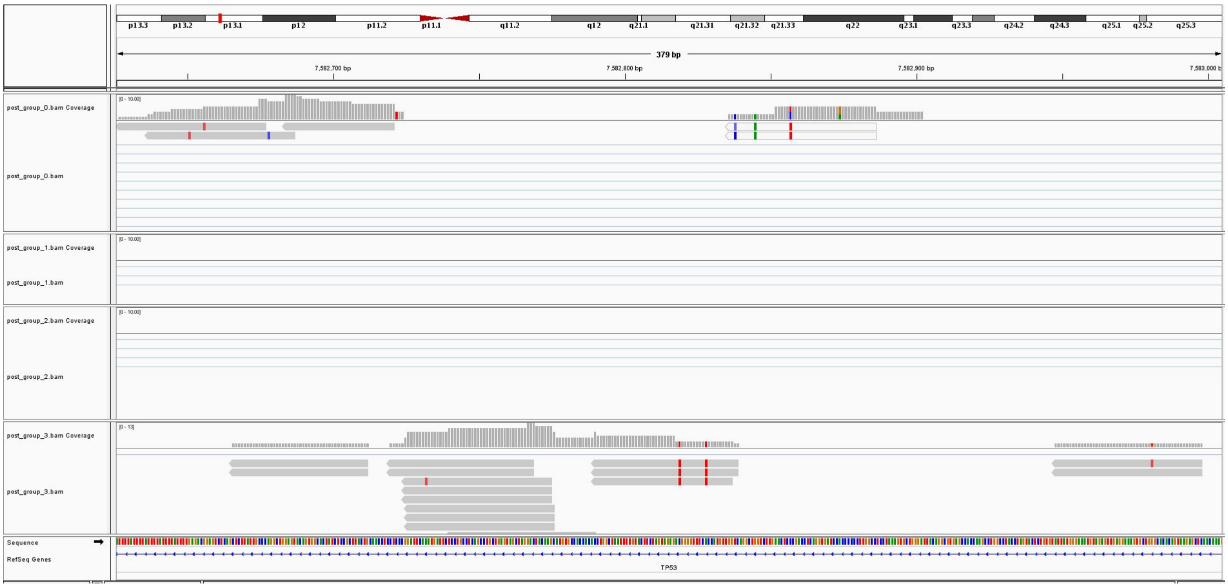


Figure 32: Sequence of gene P53 in the POST dataset, with reads separated by types (0 above to 3 below). The colours represent mutations and the grey areas represent reads. There are no reads for types POST_1 and POST_2 and reads for POST_0 and POST_3 are alternating and contain small mutations.

imply that the types POST_0 and POST_3 are in fact still fast growing and correspond to the cancer stem cells.

Another observation to be made is that towards the end of the gene, there are alternating reads between types POST_0 and POST_3. Even though these reads do not contain significant mutations, it shows that there are some differences between the two clusters in terms of the expression of gene P53 and therefore ensures that they are in fact different types. Figure 32 shows that the reads of P53 in POST_0 and POST_3 are somehow alternating along the genome, whereas POST_1 and POST_2 have no reads of P53. Some of these reads contain mutations, but only small ones, of under 10 reads.

7.4 Summary

Mutations of genes might prevent them from fulfilling their normal functions and therefore analysing them can help better undestand the data. Mutations in the P53 gene (a known tumour suppressor) are strongly correlated to tumour growth. Investigating P53 mutations in the tumour from Patient 9, before and after chemotherapy, can help assess the effect of the treatment. Intuitively, a successful treatment will eliminate the mutations. However, in practice this is not very easily identified. Some mutations can be simple errors from the extraction process and some mutations can be normal variants of the gene.

Besides the PRE and POST datasets from Patient 9, there were two other .bam files, containing genetic information of healthy and cancerous tissue separately. Comparing these gives an idea of which mutations should in fact be targeted by the chemotherapy and which are irrelevant. These mutations are then searched for in the PRE and POST datasets, to evaluate the actual outcome of the chemotherapy.

There were 3 main mutations found in the cancerous tissue data, which appeared reduced or missing from the healthy tissue data. Considering that these could be simple variants of P53 and that there was no significant relation between these and the PRE - POST mutations, no serious conclusions can be drawn. There were some other mutations from the PRE dataset, that were missing or reduced in the POST dataset. This shows that the chemotherapy has in fact targeted P53 mutations, strengthening the fact that there is a correlation between these mutations and tumour growth.

A new type of analysis has been performed on this data, by investigating gene mutations specific to each cell-type. This was possible after the development of the .bamCleave Splitter described in section 3.2 of this report. Exploring the mutations in each type has provided a stronger argument to the cell-type progression from section 6.4, where clusters POST_0 and POST_3 are labelled as cancer stem cells. There were multiple sequences of continuous P53 reads in both of these cell-types, but in none of the others. This implies that they are in fact different from the two other types (with respect to their role in tumour growth) and different from each other, since the sequences are mostly interleaving.

8 Ethical Considerations

The data to be analysed throughout the project is real data from patients at the Birmingham City Hospital, which involves some ethical considerations. However, the patients are anonymised and have given their informed consent.

Some of the analysed data will eventually be made public, when the results of the research are published, but the patients have all consented to this, which means there are no more further considerations for the scope of this project.

9 Conclusions

This final section of the report will highlight some of the main project findings and discuss what impact these can have in the future, as well as how the project could be extended.

9.1 Project findings

The project was mainly focused on gaining an understanding of what happens to different types of cells during chemotherapy and how this effect can be assessed. It is known that cancer stem cells survive chemotherapy and suspected that they are the reason why in so many cases the tumour grows back after treatment [29]. In datasets with cells from before chemotherapy, such as the one in this project, it might be clear which type can be labelled as cancer stem cells. However, after chemotherapy, these cells undergo many changes and it might be much harder to label them. The method developed in this project, the similarity quantification of clusters, is a mathematical formula tailored to working with genetic expression in types of cells to provide more insight into which cells in the post-chemotherapy dataset are similar to the cancer stem cells. This method allowed for the identification of cancer stem cells in the tumour after treatment, despite all the changes the cells have gone through. These results are strengthened by gene mutations and validate the intuition that cancer stem cells survive chemotherapy.

Throughout the project, two tools were developed to aid the analysis. These are available and can be used in the future to help gain a wider understanding of the data, in a faster manner. They provide support for labelling and comparing cell types, by automatically retrieving gene definitions and applying the mathematical methods described in this project. These tools will speed up the analysis process for others who wish to separate cells into types and investigate their gene mutations or understand differences between types.

Although all the aims of the project were achieved, the analysis was only done on data coming from one patient. The results provide insight into how the cancer evolved for Patient 9, but the methods described could be applied on other datasets. Re-applying the methods will allow for more rigurous conclusions to be drawn.

9.2 Future work

The tools developed as part of this project could always be extended with more functionality. The main improvements that could be done are multithreading the bamCleave Splitter, to reduce the running time of 15 hours; and connecting the ClusterInsight tool to existing analysis packages, such as Seurat.

The most important aspect of the similarity quantification method is that it makes no assumptions based on where the datasets to compare come from. This means that the method could be used on datasets from different patients and if they have similar types of cells, then it is very likely that the treatment will have the same output on both of them. In this way the result of chemotherapy can be predicted for different patients, based on others. The same method can be applied to different types of cancer as well, since at no point it assumes that the tumour is of ovarian cancer. It could be applied to any single-cell RNA-seq data.

The project has improved the analysis pipeline of single-cell RNA-seq data, in ways which now allow for scientists to identify cancer stem cells and understand how these survive chemotherapy for specific patients. The software and methods described in this report can be used in future analysis projects, to help us gain a better understanding of cancer tumours and chemotherapy treatments.

References

[1] Cancer Research UK, *Ovarian Cancer Statistics* (2015). [Online] (https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/ovarian-cancer).
(Accessed 27 April 2019).

[2] World Health Organization, *Cancer. Key facts.* (2018). [Online]
(https://www.who.int/news-room/fact-sheets/detail/cancer).
(Accessed 27 April 2019).

[3] World Health Organization, *Cancer. Treatments.* (2018). [Online]
(https://www.who.int/news-room/fact-sheets/detail/cancer).
(Accessed 27 April 2019).

[4] National Cancer Institute, *Types of Cancer Treatment.* at the National Institutes of Health. [Online]
(https://www.cancer.gov/about-cancer/treatment/types).
(Accessed 27 April 2019).

[5] Cancer research UK, *What is chemotherapy?* (2017) [Online]
(https://www.cancerresearchuk.org/about-cancer/cancer-in-general/treatment/chemotherapy/what-chemotherapy-is).
(Accessed 27 April 2019).

[6] National Cancer Institute, *Precision medicine.* (2017) [Online]
(https://www.cancer.gov/about-cancer/treatment/types/precision-medicine).
(Accessed 27 April 2019).

[7] Cancer Research UK, *Ovarian Cancer. Survival.* (2018) [Online]
(https://www.cancerresearchuk.org/about-cancer/ovarian-cancer/survival).
(Accessed 27 April 2019).

[8] Howlader N, Noone AM, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, Cronin KA (eds). *SEER Cancer Statistics Review* (1975-2016) National Cancer Institute. Bethesda, MD. [Online]
(https://seer.cancer.gov/csr/1975_2016/).
(Accessed 27 April 2019).

[9] Cancer Research UK, *Types of Ovarian Cancer.* (2018) [Online]
(https://www.cancerresearchuk.org/about-cancer/ovarian-cancer/types).
(Accessed 27 April 2019).

[10] Cancer Research UK, *Epithelial Ovarian Cancer.* (2018) [Online]
(https://www.cancerresearchuk.org/about-cancer/ovarian-cancer/types/epithelial-ovarian-cancers).
(Accessed 27 April 2019).

[11] U. Del Monte, *Does the cell number 10(9) still really fit one gram of tumor tissue?* (2009) in Cell Cycle. 2009 Feb 1;8(3):505-6. Epub 2009 Feb 11. PMID: 19176997. [Online]
(https://www.ncbi.nlm.nih.gov/pubmed/19176997).
(Accessed 27 April 2019).

[12] Ovarian Cancer Research Aliance, *Recurrence in ovarian cancer* [Online]
(https://ocrahope.org/patients/about-ovarian-cancer/recurrence/).
(Accessed 27 April 2019).

[13] National Cancer Institute, *Cell Structure.* SEER Modules. [Online]
(https://training.seer.cancer.gov/anatomy/cells_tissues_membranes/cells/structure.html).
(Accessed 27 April 2019).

[14] Genetics Home Reference, *What is DNA?* (2019) in NIH: U.S. National Library of Medicine. [Online]
(https://ghr.nlm.nih.gov/primer/basics/dna).
(Accessed 27 April 2019).

[15] Rettner R., *DNA: definition, structure discovery* (2017) in Live Science. [Online]
(https://www.livescience.com/37247-dna.html).
(Accessed 27 April 2019).

[16] Evan Z.Macosko et. al., *Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets.* (2015) Cell, Volume 161, Issue 5, Pages 1202-1214. [Online] (https://www.sciencedirect.com/science/article/pii/S0092867415005498). (Accessed 27 April 2019).

[17] Rogers, S., Girolami M., *A First Course in Machine Learning* (2017) ISBN: 978-1-4987-3848-4, Chapter 7. Principal Component Analysis and Latent Variable Models, p. 236.

[18] Wattenberg, M., ViÃ©gas, F., Johnson, I., *How to use tSNE effectively?* (2016) Distill. DOI: 10.23915/distill.00002. [Online] (https://distill.pub/2016/misread-tsne/). (Accessed 27 April 2019).

[19] Chen X.,, Zhengchang S., *Identification of cell types from single-cell transcriptomes using a novel clustering method*, (2015) Bioinformatics, Volume 31, Issue 12, Pages 1974â1980. [Online] (https://doi.org/10.1093/bioinformatics/btv088). (Accessed 27 April 2019).

[20] Clauset, A., Newman, M., Moore, C., *Finding community structure in very large networks* (2014) Physical review E, Volume 70, Number 6, pages 66-111, APS. [Online] (https://arxiv.org/abs/cond-mat/0408187). (Accessed 27 April 2019).

[21] Rogers, S., Girolami M., *A First Course in Machine Learning* (2017) ISBN: 978-1-4987-3848-4, Chapter 5 Non-probabilistic Classifiers. 5.3.1 K-nearest neighbours. p.181.

[22] Blondel, V. D., Guillaume, J., Lambiotte, R. Lefebvre, Et. *Fast unfolding of communities in large networks*. (2008) Journal of Statistical Mechanics: Theory and Experiment. P10008. arXiv:0803.0476.doi:10.1088/1742-5468/2008/10/P10008. [Online] (https://arxiv.org/abs/0803.0476). (Accessed 27 April 2019).

[23] Butler, A., Hoffman, P., Smibert, P., Papalexi, E., Satija, R., *Integrating single-cell transcriptomic data across different conditions, technologies, and species* (2018) Nature Biotechnology. https://doi.org/10.1038/nbt.4096. [Online] (https://www.nature.com/articles/nbt.4096). (Accessed 27 April 2019).

[24] Robinson J. T., Thorvaldsdottir H., Winckler W., Guttman M., Lander E. S., Getz G., Mesirov J. P., *Integrative Genomics Viewer.* (2011) Nature Biotechnology 29, 24â26. [Online] (https://www.nature.com/articles/nbt.1754).
(Accessed 27 April 2019).

[25] Hesman Saey T., *A recount of human genes ups the number to at least 46,831.* (2018) Science News. Vol. 194, No. 7, p. 5. [Online]
(https://www.sciencenews.org/article/recount-human-genes-ups-number-least-46831).
(Accessed 27 April 2019).

[26] The SAM/BAM Format Specification Working Group, *Sequence Alignment/Map Format Specification.* (2019) [Online]
(http://samtools.github.io/hts-specs/SAMv1.pdf).
(Accessed 27 April 2019).

[27] Bioinformatics Explained, *Small SAM examples.* [Online]
(https://www.biostars.org/p/150010/).
(Accessed 27 April 2019).

[28] Kharchenko, P.V., Silberstein, L., Scadden, D.T., *Bayesian approach to single-cell differential expression analysis.* (2014) Nat. Methods 11, 740â742. [Online]
(https://www.nature.com/articles/nmeth.2967).
(Accessed 27 April 2019).

[29] Liu H., Lv L., Yang K., *Chemotherapy targeting cancer stem cells.* (2015) Am J Cancer Res. 2015;5(3):880â893. [Online]
(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4449424/).
(Accessed 27 April 2019).

[30] Robinson M.D., Oshlack A., *A scaling normalization method for differential expression analysis of RNA-seq data.* (2010) Genome Biol. 11(3):R25. doi:10.1186/gb-2010-11-3-r25. [Online] (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2864565/).
(Accessed 27 April 2019).

[31] Zilfou J.T., Lowe S.W., *Tumor suppressive functions of p53.* (2009) Cold Spring Harb Perspect Biol. 1(5):a001883. doi:10.1101/cshperspect.a001883. [Online] (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2773645/). (Accessed 27 April 2019).