Global Genomic and Single Cell Data

Sandra Alonso Paz

February 1, 2022

1 Practice 1

Calculate PCA among ground tissue samples for shr mutant, the wild type and the complemented lines with the transcription factors BLUEJAY (BLJ), JACKDAW (JKD), MAGPIE (MGP), NUTCRACKER(NUC), IMPERIAL EAGEL (IME) and SCARECROW (SCR). File = table.csv. Plot variance captured by each component and loadings for most significant components.

What do the dots represent? Can we infer any relationship among samples?

In order to answer this question, firstly it is mandatory to upload the given table (table.csv) with the gene expression and keep only the useful columns for this section (shr mutant, the wild type and the complemented lines with the transcription BLJ, JKD, MGP, NUC, IME and SCR).

Now we can calculate PCA among ground tissue samples for shr mutants. The main reason for calculating PCA is because it is highly complicated to carry out comparisons based on plenty of genes. Therefore, we use PCA or Principal Components Analysis to reduce the dimensionality of the features of our study.

After performing the PCA, in Figure 1 can see how they contribute to the variance of the given data. In our case only the two first components seem to contribute at any value to this variance. Therefore, I will considered them as principal components and continue my study with them.

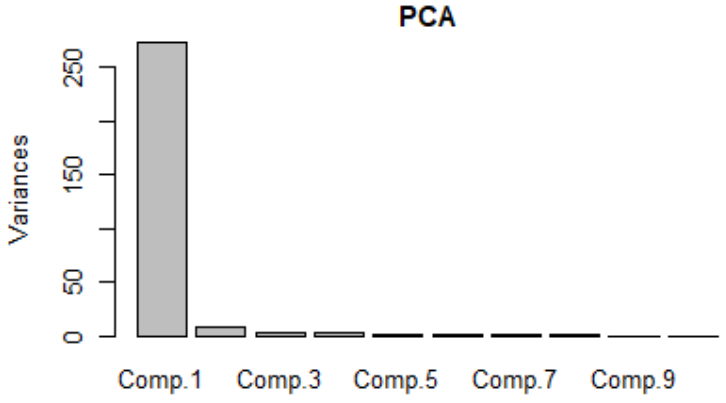

Figure 1: PCA results.

Once PCA is developed, it is interesting to plot the loadings to see which cell lines are closer to others. In Figure 2 it is observable that JKD, SCR and NUC are closer to the wild-type (WT), which means that they were closer than others to a total complementation. On the other hand, MGP, BLJ and IME are closer to the shr mutant (shrJ0571), which means that they are far from a total complementation.

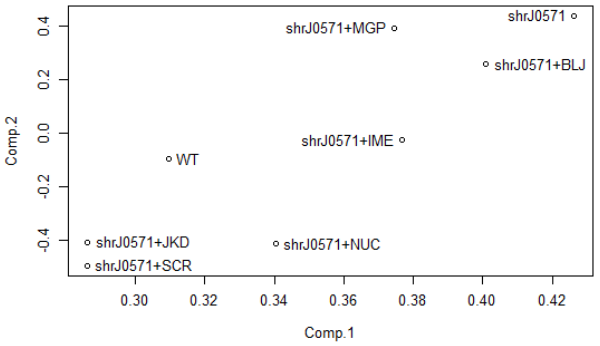

Figure 2: PCA loadings distribution.

Create intermediate transcriptomes between shr mutant and the wild type (J0571) which represent 25%, 50% and 75% of recovery (complementation) in gene expression. For every gene in the transcriptome you need to re-calculate its expression; for instance: gene in transcriptome 25% = 0.25* genei(WT/J0571)+ 0.75* genei(shr).

Recalculate PCA. Plot component variance and loadings.

Do these transcription factors recover the identity of mutant as compared to the wild-type? What role may you establish for these transcription factors?

After developing a new PCA with the created intermediate transcriptomes, I plotted their loadings distribution. There are two facts to highlight. Firstly, transcriptomes created are all inline with the wild-type and the shr mutant, so it is correctly represented. Secondly, as the level of complementation increases, compared with the last plot, genomes appear closer to the wild-type and further from the mutant.

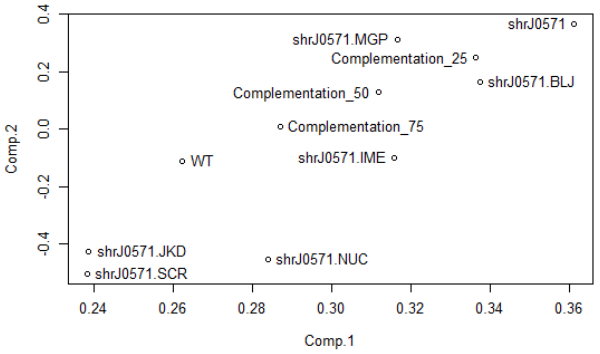

Figure 3: PCA loadings distribution.

Add the transcriptome of cells corresponding to SCR domain and recalculate PCA. Plot variance for components and loadings.

What might you conclude for several of these transcription factors? Firstly, I performed a new PCA including the column SCRDomain of the initial table and, one more time I plotted its loading. However this time, we can see that JKD and SCR appear closer to the SCR domain.

SCR Domine is an evolutionarily conserved protein domain [1]. This approach shown in the plot might mean that they are highly related with this domain or at least, more related than other cell lines.

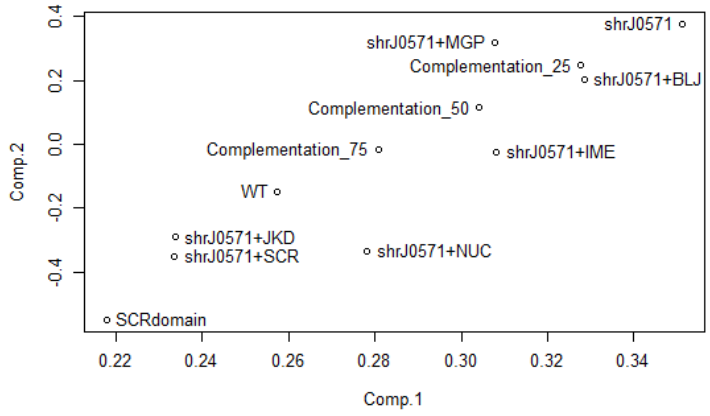

Figure 4: PCA loadings distribution.

Find the most important genes which contribute to observed transcriptomic changes by extracting genes (20) with highest and lowest score values. Investigate expression patterns of these genes across original samples using heatmap (). Remember functions sort(), head() and tail() might be useful.

What do you observe?

As we know, PCA scores are the normalised contribution of each gene. For this reason, I have used the two principal components which expressed more variance; component 1 and 2.

After sorting those columns separately I obtained the genes with higher scores using head() function. Then, I used tail() function to retrieve the 20 genes with lowest scores.

Once I knew which information I would represent, I created a matrix with the expressed genes and represent that matrix with a heatmap. In Figure 5 we can see at the right side the heatmap with the highest expressed genes and at the left side the one with lowest representation.

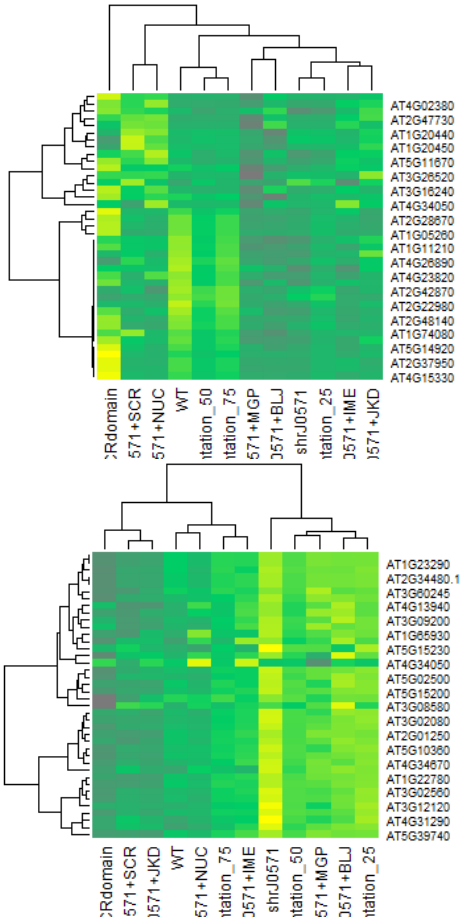

Figure 5: Most important genes contributing to transcriptomics changes

In both heatmaps we can see that the cell lines are ordered based on their similarity (their proximity on the distribution seen in Figure 4). Therefore, this supports the relation between them.

2 Practice 2

Visualize distribution of gene expression values across samples. Use function boxplot (). File = TableS5.csv. What are the samples with more variability?

Box plots, also known as box-and-whisker plots, are a graphical representation that allows you to summarize the main characteristics of the data (position, dispersion, asymmetry, ...) and identify the presence of outliers [2]. Our box plot shows on the x-axis the given samples and on the y-axis the expression values. Focusing on the question, E30 and S18 show a greater variance in their expression (as their dots are more dispersed). This variance means that some genes in the sample are little expressed and others highly expressed.

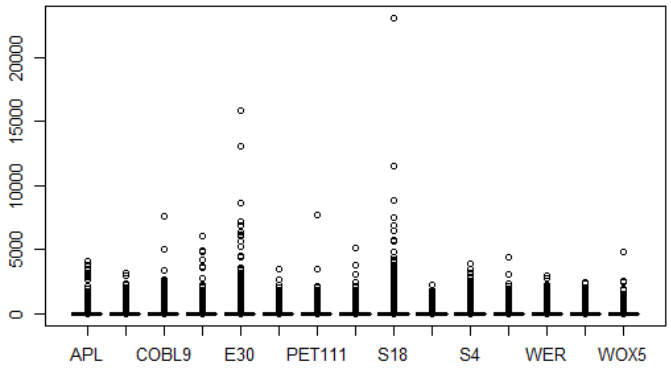

Figure 6: Distribution of gene expression values.

Analyze transcriptome differences for all cell types using PCA. What are the components which account for (or capture) more variance among samples (plot variance associated to each component).

Is there any possible relationship between expression value distribution in boxplots (step1) and PCA representation of loadings for most important components (for instance: PCA1 versus PCA2 and PCA1 versus PCA3)?

The main reason for calculating PCA is because it is highly complicated to carry out comparisons based on plenty of samples. Therefore we use PCA or Principal Components Analysis to reduce the dimensionality of the features of our study.

After performing the PCA, in Figure 7 we can see how they contribute to the variance of the given data. In our case, there are several components which seem to contribute to this variance. For instance, I will consider the 3 first components as principal components and continue my study with them.

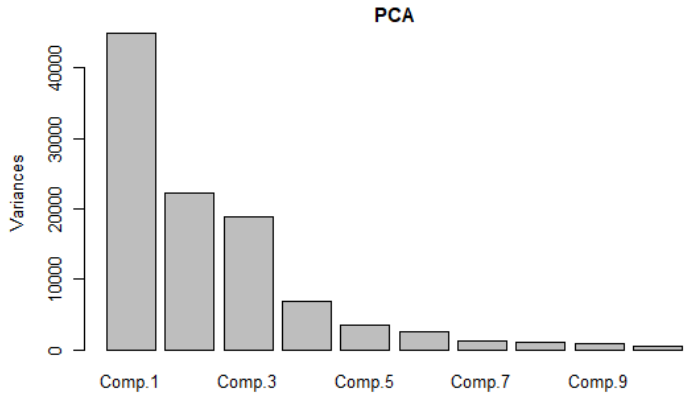

Figure 7: PCA results.

As I were proposed, I have plotted the loadings of the 3 principal components of the data given. First Component 1 versus Component 2 and secondly Component 1 versus Component 3.

By plotting the loadings, we can see which samples are closer to others and therefore, share a similar value of expression. In both plots we can see that samples S18 and E30 are far from the other sample, so their expression (variance) will be different from the ones which are placed together.

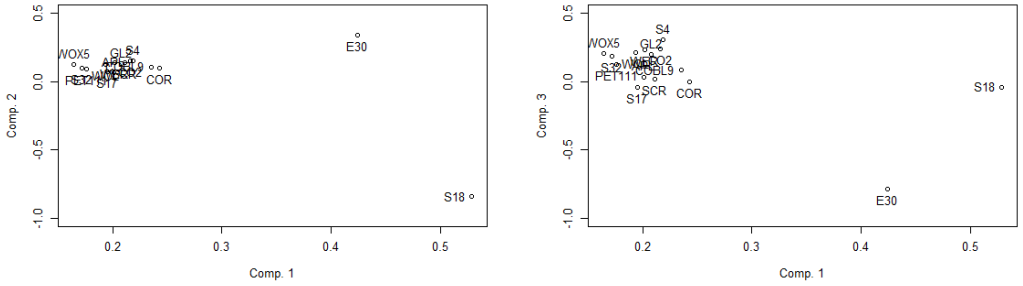

Figure 8: Loading distribution for components 1, 2 and 3.

If we compare these results with the ones obtained with the boxplot, we can affirm that there is a true relationship between them and the results obtained performing a PCA analysis. Both studies revealed that E30 and S18 have a greater variance on their expression and that they differ from the rest of the samples.

Find biomarkers for Stem Cells as those genes only expressed in WOX5 domain (threshold for expression=1) and not expressed in rest of cell types. You may need to apply the logic operator "&". You may also want to use function subset (). Obtain biomarker names and visualize their expression patterns in a heatmap.

For this task I have created a subset using subset() function for filtering the biomarkers as I were asked to. Then I created a heatmap for showing the level of expression. I have used 2 colors: yellow for high expression and green for low.

With the obtained heatmap (See Figure 9) I can prove that the biomarkers I have filtered using the function subset() are only expressed in the WOX5 (yellow cells) and are not expressed (green cells) in other cell lines over the threshold used (1.0).

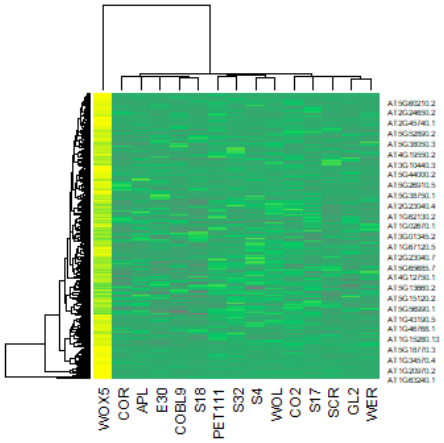

Figure 9: PCA results.

Separate now samples by performing PCA only in genes previously identified as Stem Cell Biomarkers (in step 3). Plot variance percentage for components, loadings and scores. What do you observe? Why?

If we only take into account the selected biomarkers and we plot their loadings, we can show that WOX5 is very separated from the others. This is because we have limited the expression of all the samples (unless WOX5) to be lower than one. Consequently, we can see that all of them are near to 0.0 because they are not able to vary between the limits marked. On the other hand, as we have not set a top limit for WOX5 it is able to vary its expression.

To conclude, WOX5 has a very different value of expression and the remaining samples have similar expression value. This is the main reason of the distribution of the loadings seen in figure 9

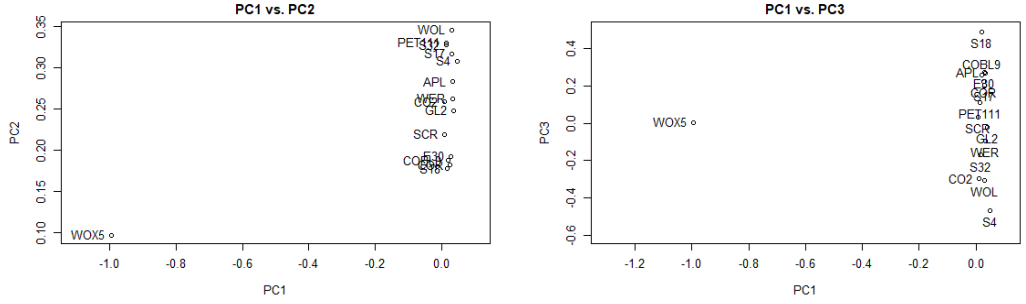

Figure 10: Loading distribution for components 1, 2 and 3.

Plot PCA scores of Stem Cell biomarkers in PCA analysis of step 2. Check first 4 components, for instance, scores in PCA1 versus PCA2 and scores in PCA3 versus PCA4. Make a graph with multiple panels in which you plot variance percentage for components, and scores in first 4 components.

Can biomarkers be easily identified from PCA scores? Speculate about for what cell type genes with highest/lowest scores might be biomarkers.

Scores plots let us see if the selected biomarkers are placed scattered [3]. However, blue dots (selected biomarkers) are grouped together. Although they have plenty of different values for the first PC as we see in Figure 6, they are not separated from the rest of genes which is surprising.

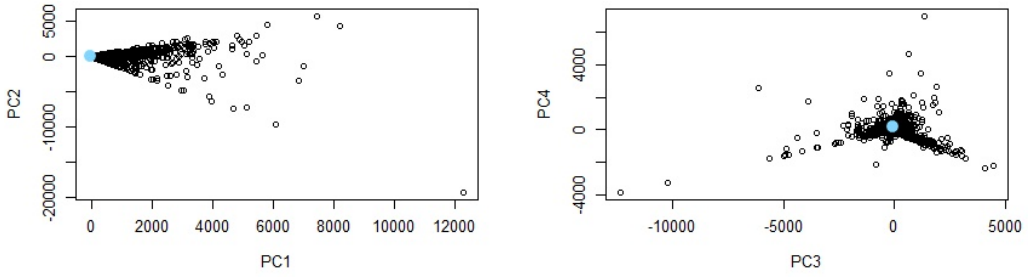

Figure 11: Scores of Components 1 vs 2 and Component 3 vs 4.

3 Practice 3

Using the raw data in file "SCOplanaria.txt" do:

1. **Initialize a Seurat object.**
2. **Perform quality control, normalization, variable features selection and scaling of the data.**
3. **Perform a PCA analysis and obtain a t-SNE clustering plot.**

First step is creating a Seurat Object. For this task I have loaded the given data of "SCOplanaria.txt" and converted it into a sparse matrix. The main reason for using sparse matrix is because I detected that the data contains many "0". Using this kind of matrix allowed us to save memory and reduce the execution time.

Once we create the matrix, it is time to create the Seurat Object. I decided that the number of cells in which a gene has to be expressed for it to be consider is 1 and the number of genes which have to be expressed in a cell for it to be considered is also 1.

Afterwards, I performed a quality control which was automatically calculated during the creation of the object. For visualizing it I used a Violin plot which can be seen in Figure 12

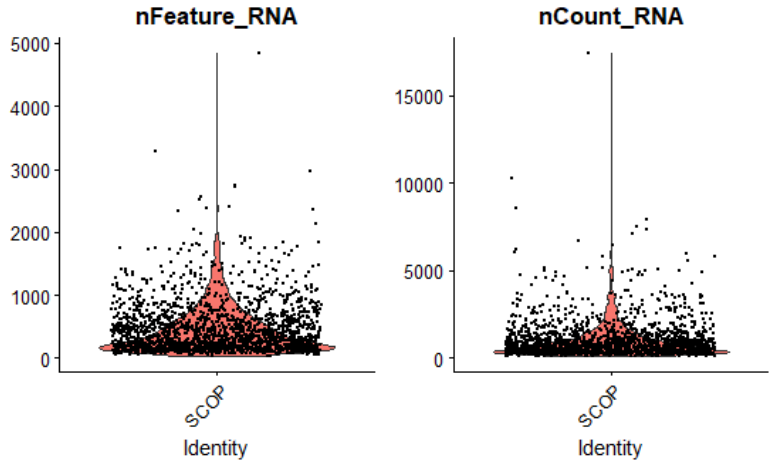

Figure 12: Quality Control metrics visualization.

As data has not been normalized, all the points are on the lower part of the plot. Now that we have performed our initial Cell level QC, we can go ahead and normalize the data.

By default, Seurat implements a global-scaling normalization method "LogNormalize" that normalizes the gene expression measurements for each cell by the total expression, multiplies this by a scale factor (10,000 by default), and log-transforms the result [4].

As we can see in Figure 13 after the normalization, not all the points are grouped on the lower part of the plot as in Figure 12.

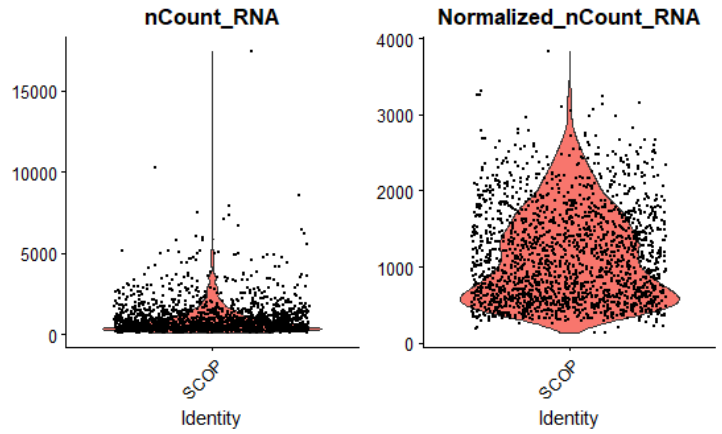

Figure 13: Comparative of n_Counts before and after the normalization process.

Now it is time to perform a feature selection. This task selects those genes whose expression varies the most between cells. In this case, I identified the 1000 most variable genes using FindVariableFeatures() function. Using a LabelPoints plot I could visualize these results:

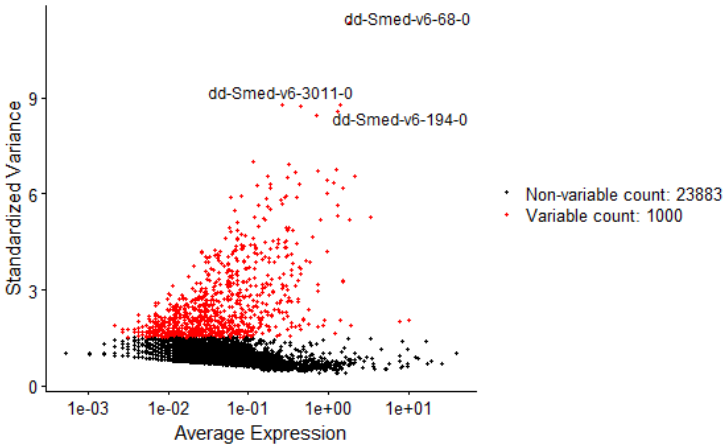

Figure 14: Expression variety of each gene.

Moreover, I label the top 3 genes expressed. They are "dd-Smed-v6-3011-0", "dd-Smed-v6-68-0" and "dd-Smed-v6-194-0".

For scaling I used a linear transformation that makes mean expression and variance across cells.

Finally I performed a PCA analysis using RunPCA() function and only taking into account 1-5 PC. In addition I used a resolution of 0.2. Then I obtained a t-SNE clustering plot using FindClusters() and RunTSNE() functions. Results of this performance can be seen in Figure 20 where I used a DimPlot to visualize them. The result is the distribution of all the data in 7 different clusters which are differentiated by colors.

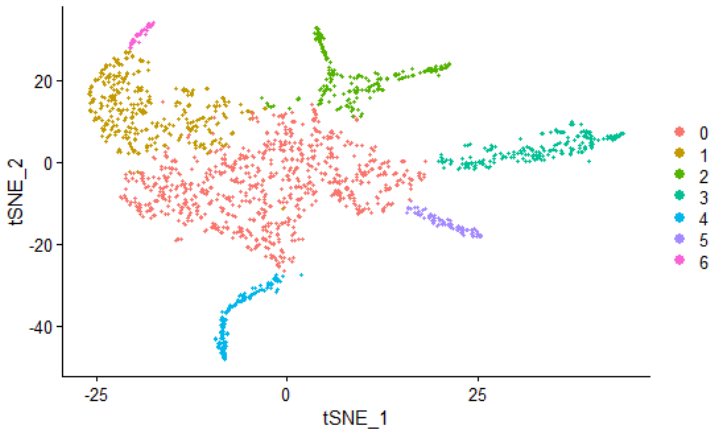

Figure 15: Expression variety of each gene.

Now try using the following parameters:

1. **Keep genes expressed in at least 3 cells and cells with at least 200 genes detected.**
2. **Keep cells with 200 - 2500 number of detected genes**
3. **Use LogNormalize method with a scale factor of 10000.**
4. **Select 3000 variable features**
5. **Regress out variability for number of genes in each cell.**
6. **Use PCs 1-5 and a resolution value of 0.6.**

Comment and explain the difference with previous results.

In this section I performed almost the same steps as for the first Seurat performed.

Firstly I created a new Seurat Object using a sparse Matrix. However, this time the number of cells in which a gene has to be expressed for it to be considered are 3 and the number of genes which have to be expressed in a cell for it to be considered are 200 as the exercises said.

Afterwards I performed a quality control which was automatically calculated during the creation of the object. For visualizing it I used a Violin plot which can be seen in Figure 18 where I compare the features of the first Seurat object and the current one.

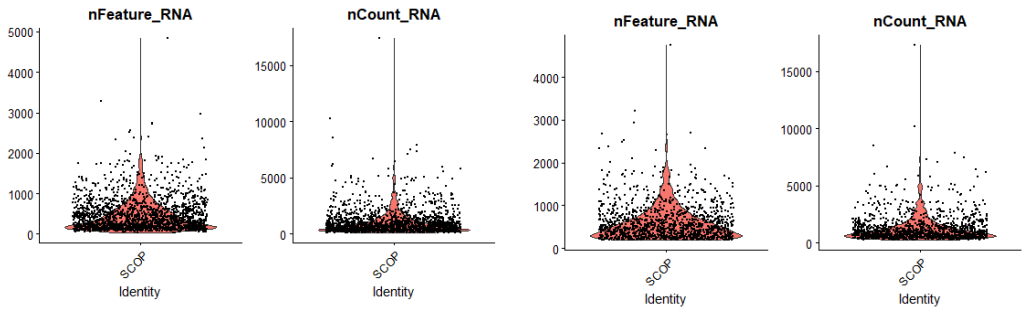

Figure 16: Features from current Seurat Object (left) and Seurat Object form last section (right).

As we can observe in the last figure, there are less number of features and so, less number of n_counts than in the first Seurat Object. This is because the "filter" we have applied for the number of cells and genes is more restrictive.

Next step is selection. Only the cells with a number of genes between 200 and 2500 are chosen. For this task I created a subset of the initial data by filtering nFeature_RNA column.

Once we have all the information we want to study, it is time to perform a normalization but this time using LogNormalization method and a scale factor of 10000.

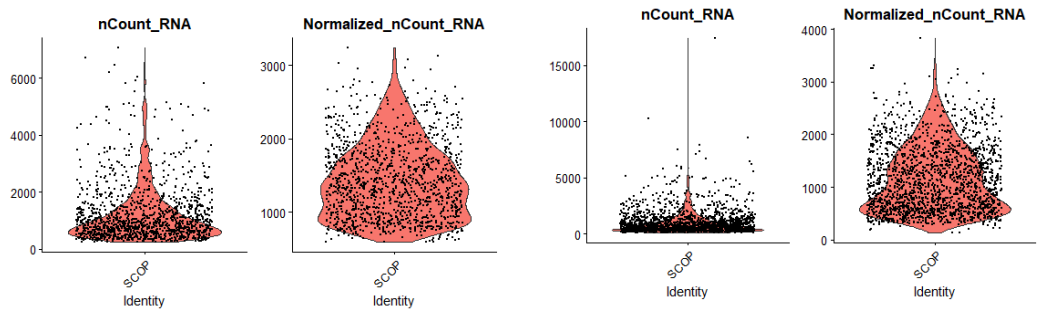

Figure 17: Comparative of n_Counts before and after the normalization process in each Seurat object (current Seurat at left side)

As we can see in the left side of the image, the current Seurat has low its n_counts due to the filter made with the subset. Moreover, comparing with the first Seurat (right side) we have now a more constant distribution. This is because we have eliminated cells with very few and many genes, so variation of the cells are limited.

In order to perform the feature selection by identifying the subset of genes whose expression varies the most between cells. This time I have considered 3000 genes. For this task I used LabelPoints() function and this were the results:

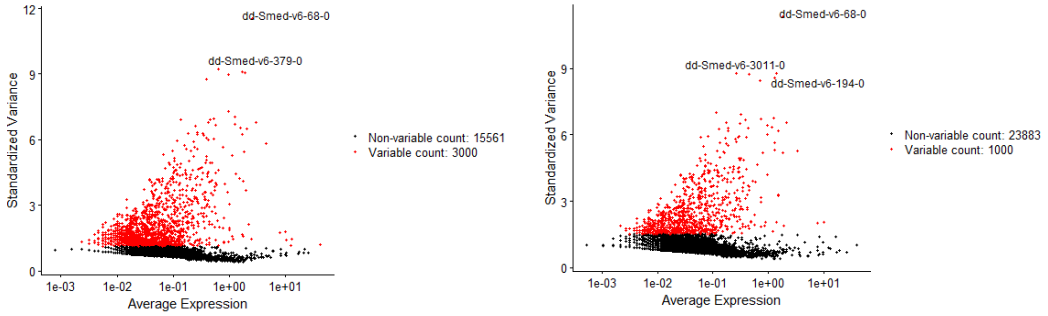

Figure 18: Expression variety of each gene and each Seurat Object (current Seurat at left side).

As I supposed, the total number of genes is higher in the first Seurat Object (right side of the image) than in the current one (left side). It is interesting to see that the most variable genes in both studies are not the same. This might be due to the filtering process.

Finally I performed a PCA using RUNPCA() function. Then I used the firsts 5 PCs and a resolution value of 0.5 for obtaining a t-SNE clustering plot. These are the results:

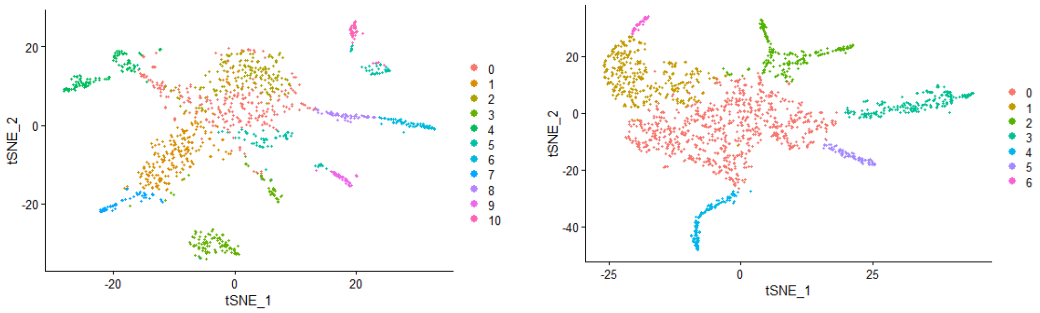

Figure 19: t-SNE clustering plot for each Seurat Object (current Seurat at left side).

Although in both studies I have take into account 5 PCs, in the first Seurat Object (right side) the resolution value was 0.2 and therefore there are less clusters than in the first image (current Seurat) where there are more clusters but they are smaller.

Extract a table of the 5 top biomarkers for each of your clusters. Use these biomarkers to generate a heatmap. Comment the results

Biomarkers are genes that are expressed significantly positive or negative in one cell line [5]. For this task I have used FinAllMarkers() function with arguments "min.pct" (tests only those genes that are detected at a minimum percentage in the all cells) and "logfc.threshold" (Limit testing to genes which show, on average, at least X-fold difference between the two groups of cells.) [6]. Afterwards I grouped the obtained biomarkers by each cluster and only took the top 5 of each one.

Finally, I created an expression heatmap for the remaining biomarkers. On the left side we can see the list of biomarkers. First 5 corresponds to cluster 0, five next correspond to cluster 1 and so on. This is why the higher expression is appreciable as a yellow diagonal.

This heatmap shows how cells from the same cluster share a common expression pattern. It is interesting to see that 4 and 9 clusters share similar expression patterns (yellow color). They are probably corresponding to similar cell lines.

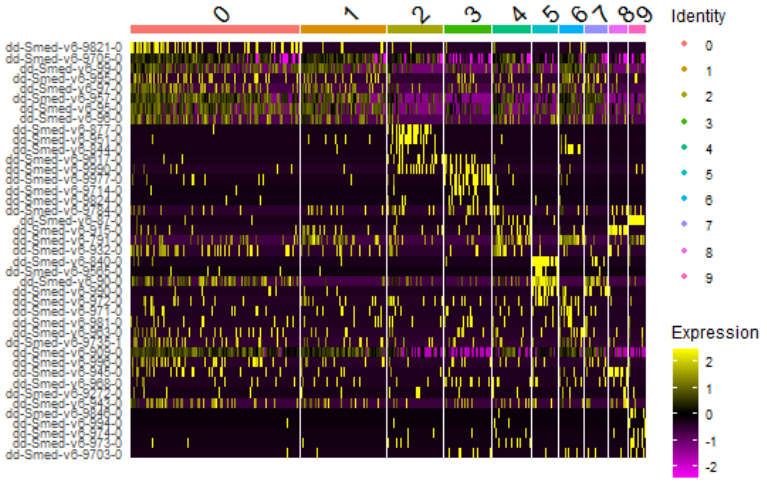

Figure 20: 5 top biomarkers for each of your clusters heatmap.

Here we show a table with biomarkers for several cell identities:

Cluster id	Gene name	Cell identity
0	dd-Smed-v6-61-0	Early epidermal progenitors
1	dd-Smed-v6-2178-0	Late epidermal progenitors
2	dd-Smed-v6-298-0	Epidermis
3	dd-Smed-v6-1410-0	Muscle progenitors
4	dd-Smed-v6-702-0	Muscle body
5	dd-Smed-v6-2548-0	Neural progenitors
6	dd-Smed-v6-9977-0	GABA neurons
7	dd-Smed-v6-48-0	Phagocytes
8	dd-Smed-v6-175-0	Parenchymal cells
9	dd-Smed-v6-1161-1	Pigment

Using VlnPlot() and FeaturePlot() functions, show the expression distribution of these markers along your clusters. With this information, identify and rename your clusters.

Using Violin plot I could show the expression distribution of the 10 markers along the 10 clusters. Here is the result:

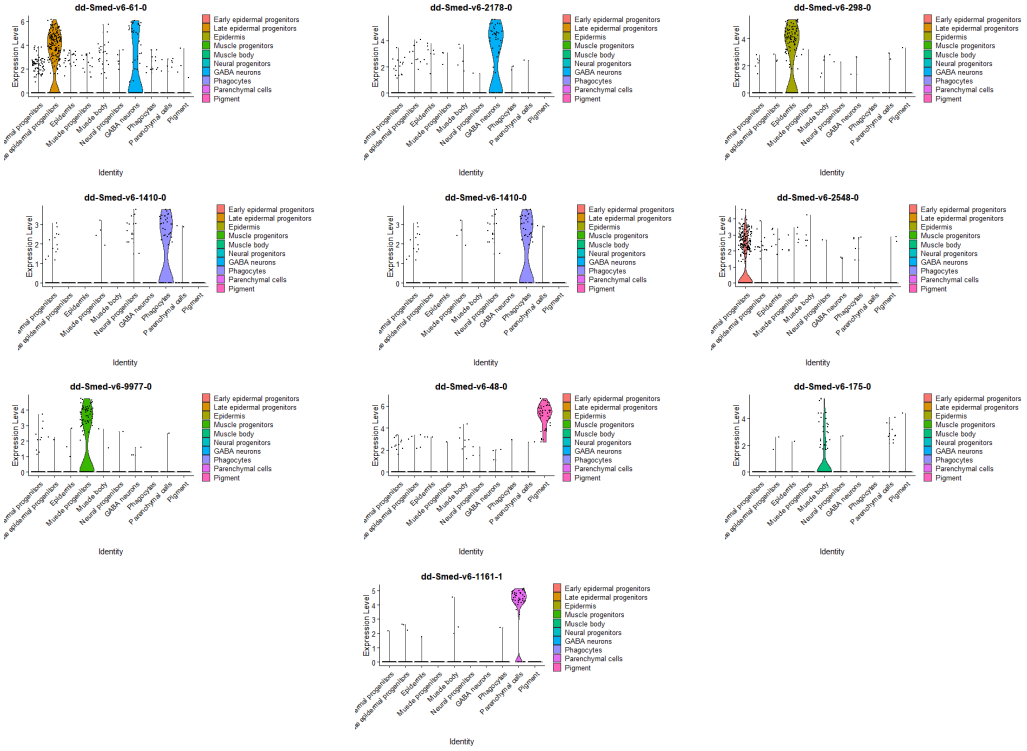

Figure 21: Violin plots of the distribution of the 10 markers along the 10 clusters.

Then, I used FeaturePlot() function to show the distribution of each marker in the DimPlot. Here are the results:

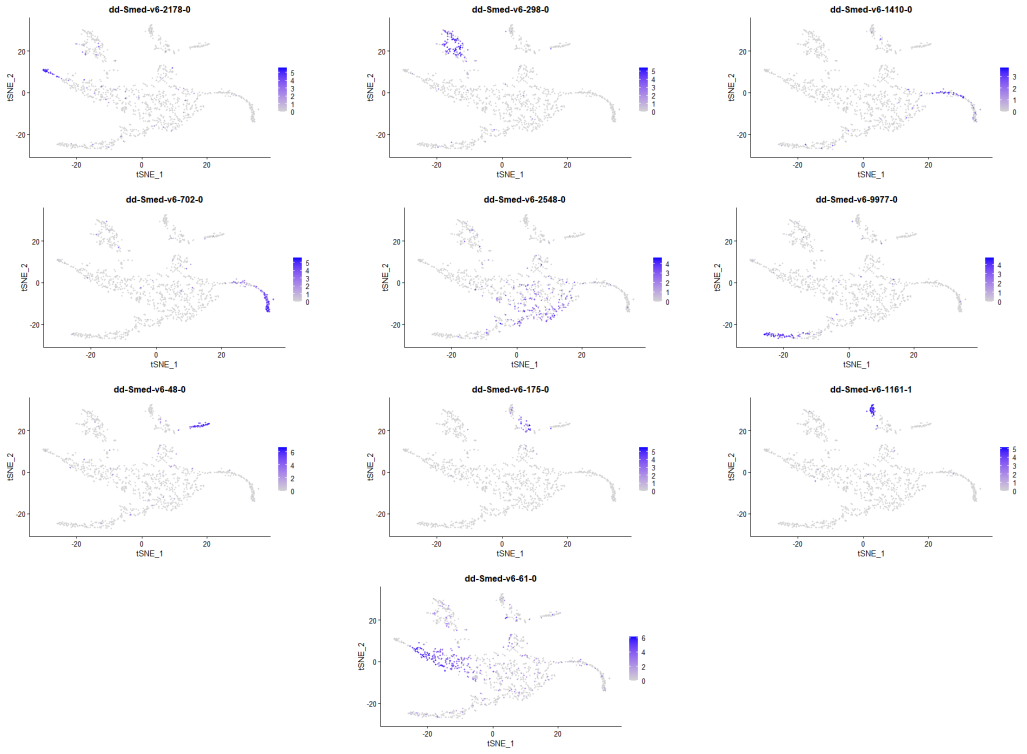

Figure 22: Feature plots of the distribution of the 10 markers along the 10 clusters.

Compare your results with this lineage tree reconstruction of planarian cell types (Plass et al., 2018).

What cell type do you think cluster 0 (the central cluster) is? Can all of the planarian cell types be found in your plot? Why?

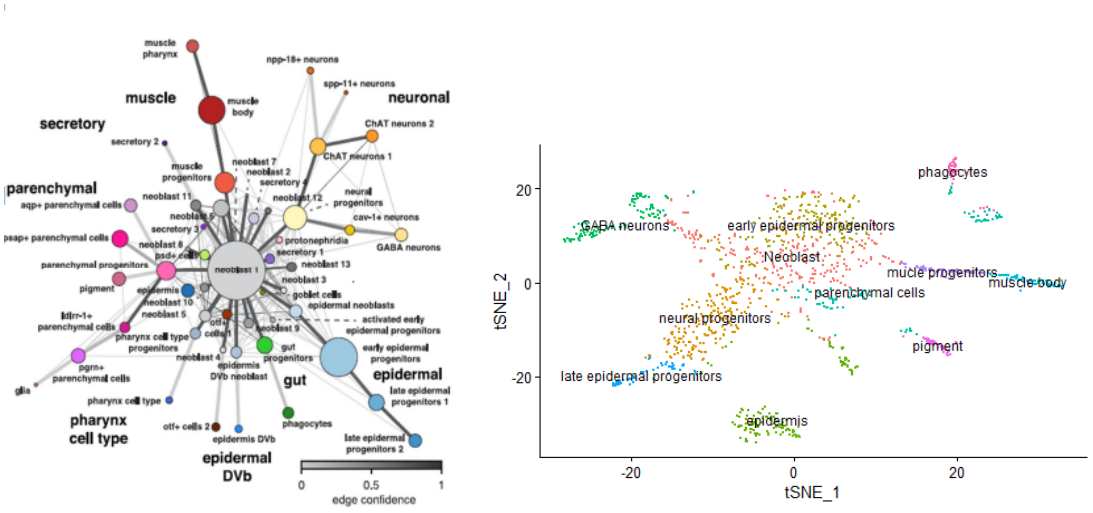

Figure 23: Reconstruction of planarian cell types (Plass et al., 2018) and current DimPlot with renamed clusters

If we compare our DimPlot with the tree reconstruction of planarian cell lines shown in Figure 24, all the cell lines appear. This makes sense because our dataset is a sample of

planaria and, even if we had performed a local biopsy, we expect to find a mix of all cell lines which are not tissue specific. In our case, the central cluster correspond to Neoblast which matches the central node in the planarian cell lines graph.

dd-Smed-v6-1999-0 is a neoblast (stem) marker gene. Show its expression distribution with VlnPlot() and FeaturePlot() and explain the result.

Neoblasts are distributed all over the body and represent almost 30% of all the cells [7]. As they are not tissue specific, we can find its marker genes spread in the transcriptome map as we can see in the Violin plot.

Although we can see in the Feature plot that it is very distributed, it is more abundant in the central cluster as we see also in Figure 24.

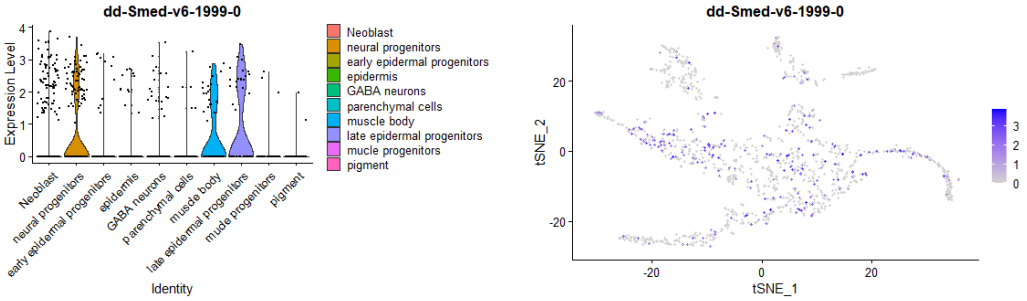

Figure 24: Violin and Feature plot for dd-Smed-v6-1999-0.

4 Annex

Scripts developed for each practices can be consulted in my GitHub repository.

References

[1] InterPro. *Sushi/SCR/CCP domain*. URL: https://www.ebi.ac.uk/interpro/entry/InterPro/IPR000436/.

[2] R Documentation. *Boxplot.matrix function*. URL: https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/boxplot.matrix.

[3] Kevin Dunn. *Interpreting score plots*. URL: https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/interpreting-score-plots-and-loading-plots.

[4] learn.gencore.bio.nyu.edu. *Data normalization and PCA*. URL: https://learn.gencore.bio.nyu.edu/single-cell-rnaseq/seurat-part-3-data-normalization/#:~:text=By%5C%20default%5C%2C%5C%20Seurat%5C%20implements%5C%20a,and%5C%20log%5C%2Dtransforms%5C%20the%5C%20result..

[5] Kyle Strimbu and Jorge A Tavel. "What are biomarkers?" In: *Current Opinion in HIV and AIDS* 5.6 (2010), p. 463.

[6] RDocumentation. *FindAllMarkers: Gene expression markers for all identity classes*. URL: https://www.rdocumentation.org/packages/Seurat/versions/4.1.0/topics/FindAllMarkers.

[7] Wikipedia. *Neoblast*. URL: https://en.wikipedia.org/wiki/Neoblast.