Report: Transcriptomics and Single Cell Genomics

Pedro Bueso-Inchausti Garcia

2020-1-13

Contents

1 Background

1.1 PCA (linear dimensionality reduction)

When information is collected from a data sample, the most frequent is to take as many variables as possible. However, we need to be aware that some of them might be related or measure the same aspect from different points of views. If that is the case, it might be necessary to reduce the number of variables.

The Principal Component Analysis (PCA) [1] consists on the transformation of the set of original variables into another set, the Principal Components (PCs), obtained as a linear combination of the originals. The new variables, in the same number as the originals, retain all the variability. However, most of the PCs explain such a small variability that can be ignored, while a few PCs can be considered without a significant loss of information. PCs are independent between each other, so they explain the maximum possible residual variability that has not been explained by previous ones.

Before performing the PCA, there are a couple of things that need to be considered. The first thing is whether the original variables are correlated; only if they are does PCA makes sense. The second thing is whether the original variables are heterogeneous -expressed in different units of measure- or homogeneous -expressed in the same units of measure-. In the first case, it would be necessary to use the correlation matrix, which normalises all the variables; in the second case, using the covariance matrix, which implies no loss of information, would be more appropiate.

Once the PCA has been performed, there are some metrics that should be examined. The eigenvalues are the variance explained by each PC. The eigenvectors are the linear combinations that define each PC. The loadings are the eigenvector coefficients; this is, the degree in which each original variable influences a PC. The scores are the original instances put into the PCs units. All these metrics can be visualized for better understanding. The sediment graph shows the eigenvalues for each PC; although there are other methods, it is quite common for determining the how many to use (note that a high number of PCs can explain a greater proportion of the total variability while a low number allows for greater simplicity in the representation). The loading plot shows, for the first two PCs, the loadings of each variable; the variables furthest from 0 in the horizontal axis are the ones giving meaning to the first PC; the variables furthest from 0 in the vertical axis are the ones giving meaning to the second PC; the variables close to the centre or displaced in both axis do not help in the interpretation of the PCs. The score plot shows, for the first two PCs, the scores of each instance; this representation allows describing, in a simple way, the multidimensional dataset.

Although PCA is an interesting technique when dealing with complex datasets, it has some drawbacks. The first is the partial loss of information. The second is that, while the original variables have real meaning, the PCs generally lack such meaning; therefore, the interpretability of the results is worst.

1.2 t-SNE and UMAP (nonlinear dimensionality reduction)

Nonlinear dimensionality reduction techniques allow embedding high-dimensional data in a low-dimensional space, which is appropriate for visualization. This is possible because they convert similar high-dimensional instances into nearby points and dissimilar high-dimensional instances into distant points. How is this done? The mathematics are a bit advance but the intuition behind is simple. They first built a high-dimensional representation of the data that establishes which instances are connected; they then optimize a low-dimensional representation to be as structurally similar as possible.

T-distributed Stochastic Neighbor Embedding (t-SNE) [2] was the original technique. Its performance, however, suffers with large datasets. Uniform Manifold Approximation and Projection (UMAP) [3] is a new technique that is faster, scaling well in terms of size and dimensionality, and allows preserving not only the local structure of the data (whether two instances belong to the same cluster) but also its global structure (whether two clusters are more similar between them than other two). One big problem for both techniques is that using them correctly can be challenging as clusters are influenced by the chosen parametrization.

2 Practice 1: PCA for studying genetic expression on cell lines

2.1 Objective

In this practice, we will study the genetic expression of different cell lines. We have information for a wild-type, a mutant and complemented lines (mutants to which we add a transcription factor). By adding these TFs, we expect to complement the loss of genetic expression in the mutant; some TFs will achieve a total complementation while others will undercomplement or overcomplement the wild-type expression. Some complemented lines might even develop new functionalities. PCA will be used to explore such hypothesis.

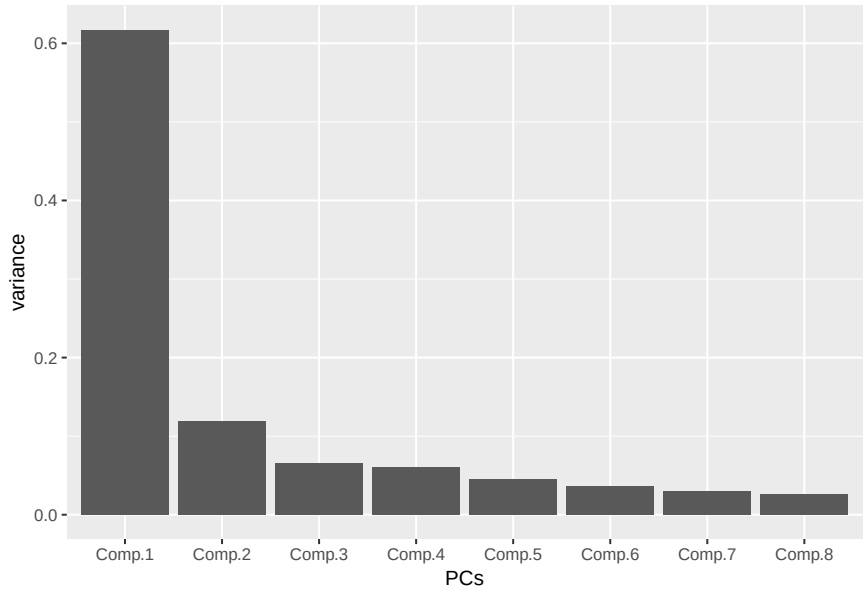
2.2 Data preparation

The first thing we do is to read the table with the gene expression and keep the subsets of interest: first_subset (includes genes, mutant, wild-type and complemented lines BLJ, JKD, MGP, NUC, IME, SCR), second_subset (adds the artificially created transcriptomes with 25% wt + 75% mut, 50% wt + 50% mut, 75% wt + 25% mut) and third_subset (adds the SCR domain).

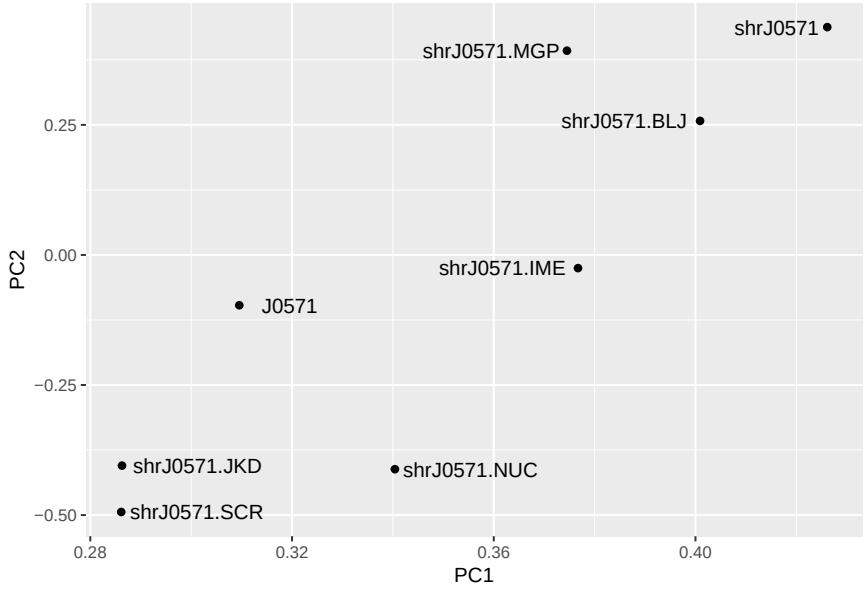
2.3 Analyze transcriptome differences between cell lines

The relationships between cell lines are calculated in terms of disimilarity. As it is very difficult to perform comparisons based on thousands of genes, we need to do a dimensionality reduction. This is achieved through PCA, which transforms thousands of correlated genes into a few uncorrelated PCs.

By plotting the eigenvalues associated to each PC, we can see how they contribute to the variance of the original set. In this case, there are few PCs that contribute to most of the variance. From now on, we will be working with the first 2, that accumulate almost 90% of it.

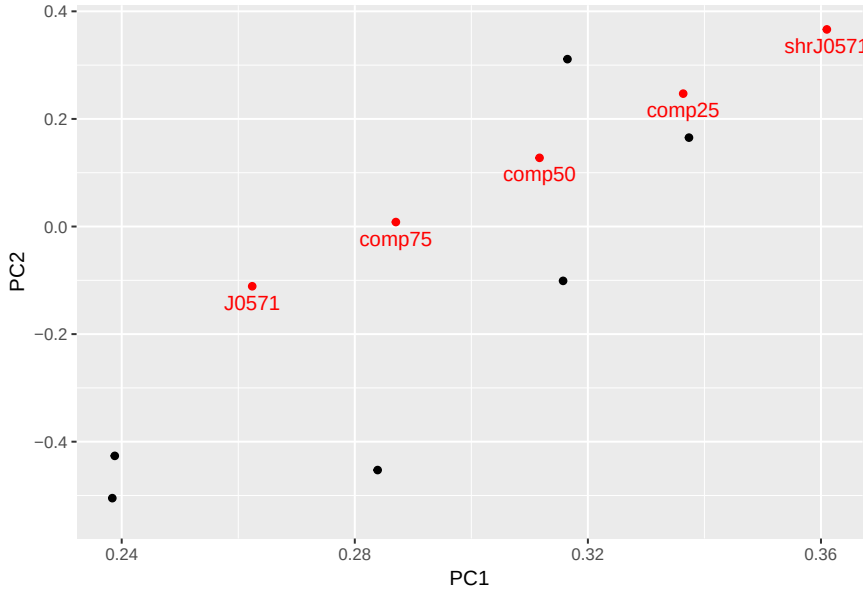


By plotting the loadings, we can see which cell lines are closer to others (we focuse primaly on the first PC, this is, the X axis). The complemented lines JKD, SCR and NUC are closer to the wild-type, which means that they were closer than others to a total complementation.



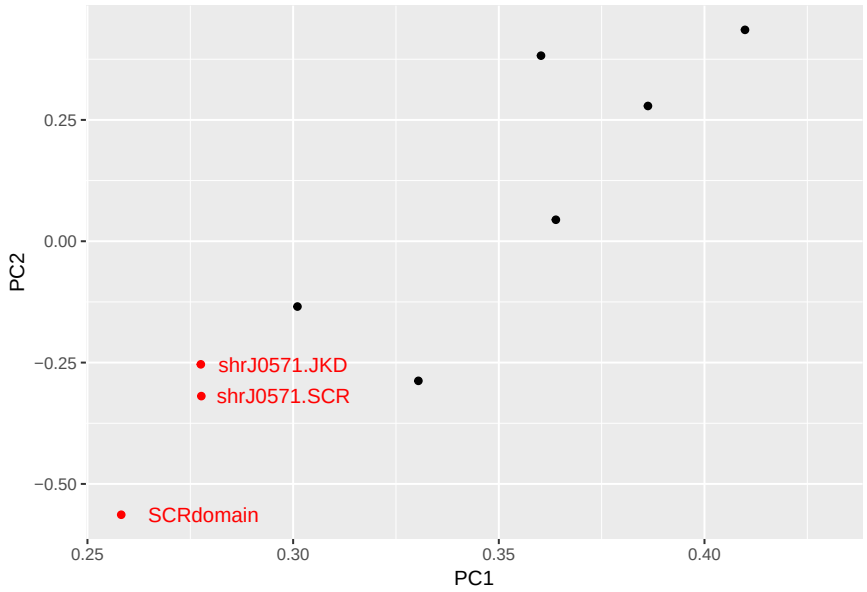
2.4 Repeat but consider artificial genomes

As the level of complementation increases, the artificial genomes appear closer to the wild-type and further from the mutant. It is interesting to see how such changes happen in a completely linear way.



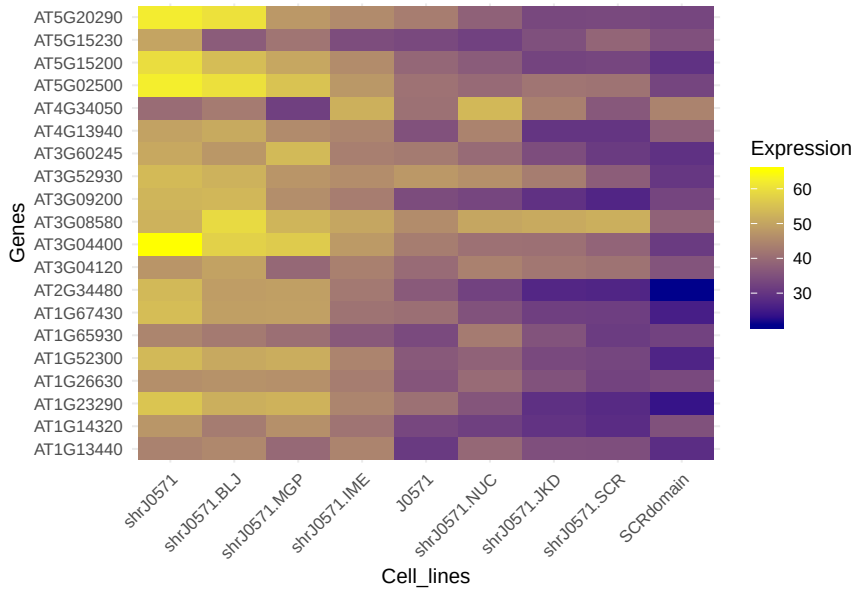
2.5 Repeat but consider SCR domain

The complemented lines JKD and SCR appear closer to the SCR domain, which means that they include extra information related to stem cells. This will, therefore, correspond to a case where the complementation results in acquisition of new functions that were not present in the wild-type.



2.6 Find the genes contributing to these transcriptomic changes

The normalised contribution of every gene is given by the PCA scores. We focuse only on the first PC, as it is the one which explain most of the variance. By sorting that column, we can retrieve the the genes with higher scores (both positive and negative). We create a matrix with the expression of these genes in each of the cell lines and represent that matrix with a heatmap. In the heatmap including the positive genes (the only one shown), we can see tendencies which are coherent with our previous analysis.



3 Practice 2: PCA for detecting biomarkers on different cell lines

3.1 Objective

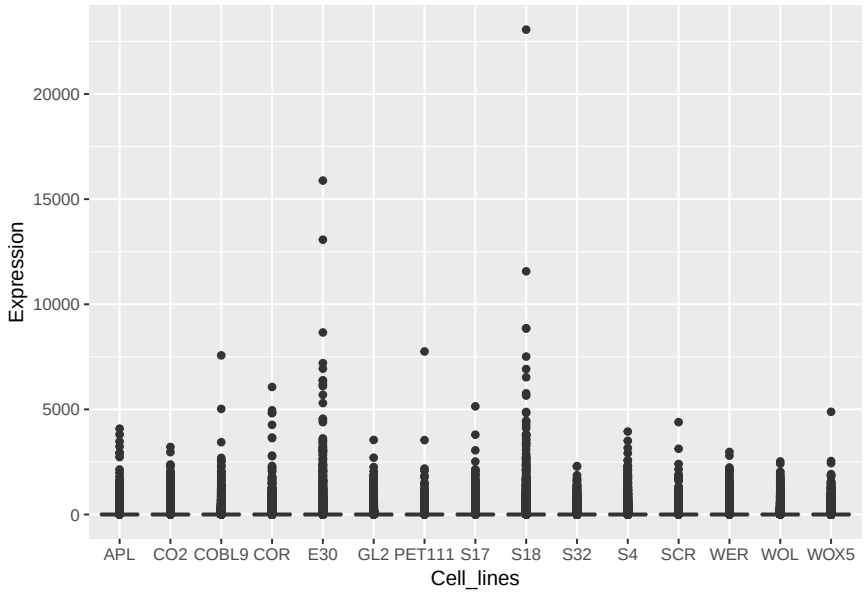
In this practice, we will study the genetic expression of different cell lines. In particular, we want to detect cell lines with a variable expression as well as to segregate some cell lines from others based on biomarkers detection. To do so, we will be using PCA.

3.2 Data preparation

The first thing we do is to read the table with the gene expression and keep the subsets of interest: first_subset (includes all the genes) and second_subset (includes the biomarkers for stem cells; these are genes which are expressed in WOX5 domain and not expressed in rest of cell lines).

3.3 Visualize distribution of gene expression across cell lines

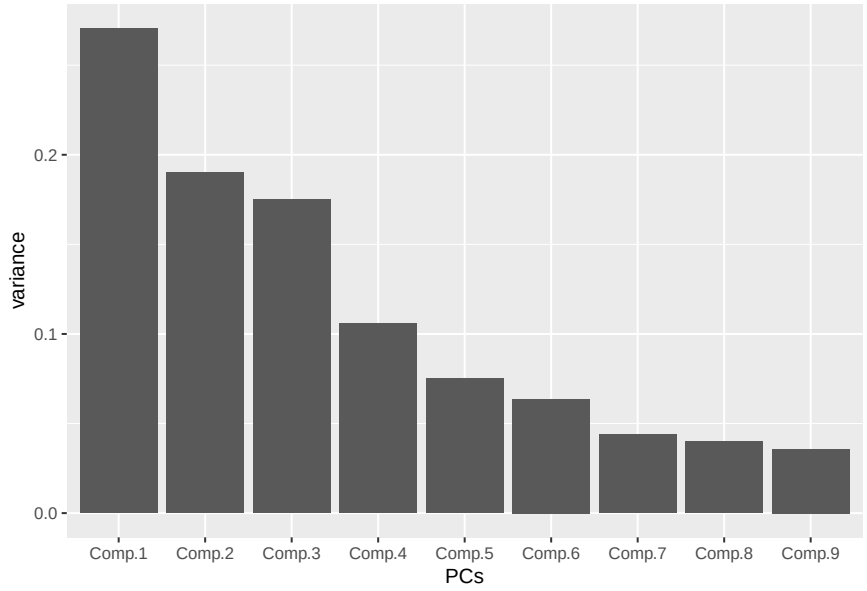
We use the boxplot as a way to show distributions. We can see that both E30 and S18 show a greater variance in their expression. They are very specialised cell lines, which means that some genes express little while others express a lot. This could explain their high variance.



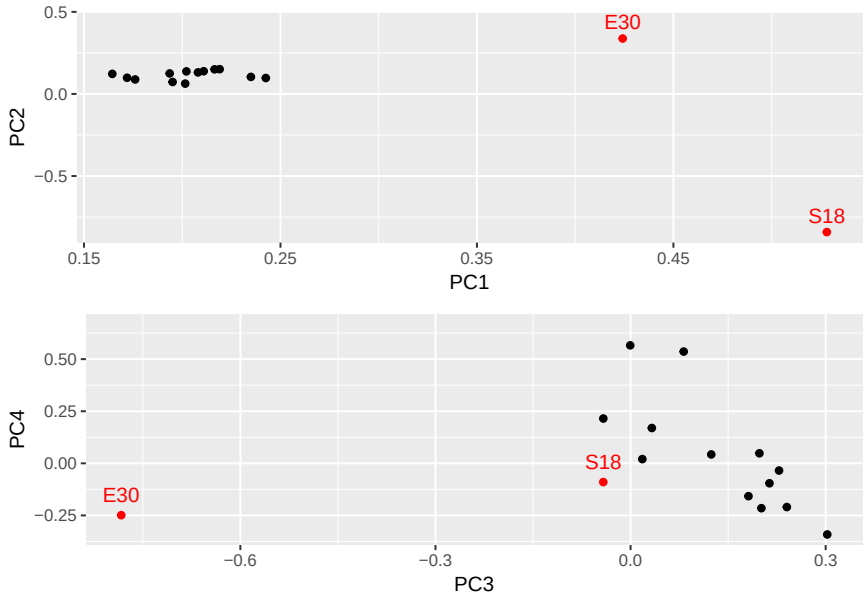
3.4 Analyze transcriptome differences between cell lines

The relationships between cell lines are calculated in terms of disimilarity. As it is very difficult to perform comparisons based on thousands of genes, we need to perform a dimensionality reduction. This is achieved through PCA, which transforms thousands of correlated genes into a few uncorrelated PCs.

By plotting the eigenvalues associated to each PC, we can see how they contribute to the variance of the original set. In this case, there are quite a lot of PCs that contribute to the variance. From now on, we will be working with the first 4, that accumulate almost 70% of it.

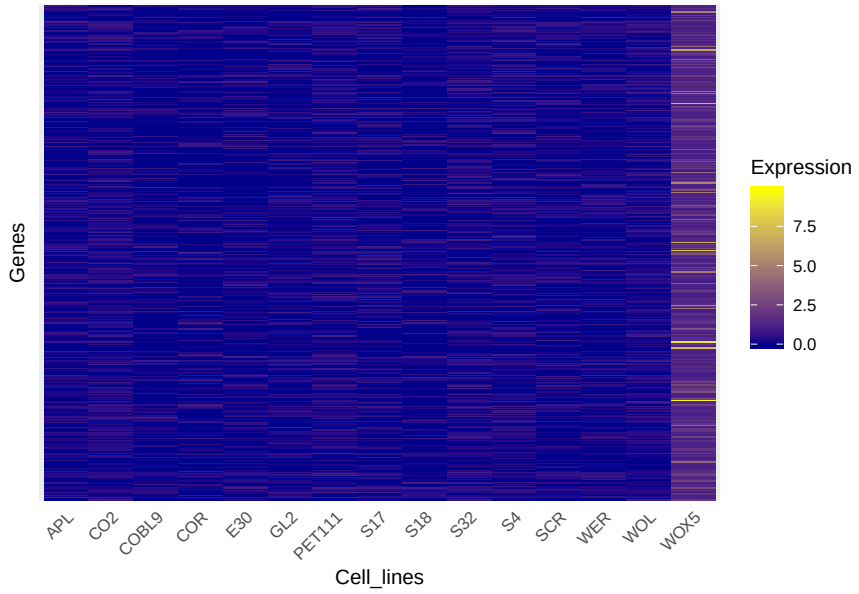


By plotting the loadings, we can see which cell lines are closer to others. The cell lines E30 and S18 are clearly segregated from the others, which means that their gene expression is very different. Note that we should give more importance to the first plot, as it results from the combination of the first two PCs.



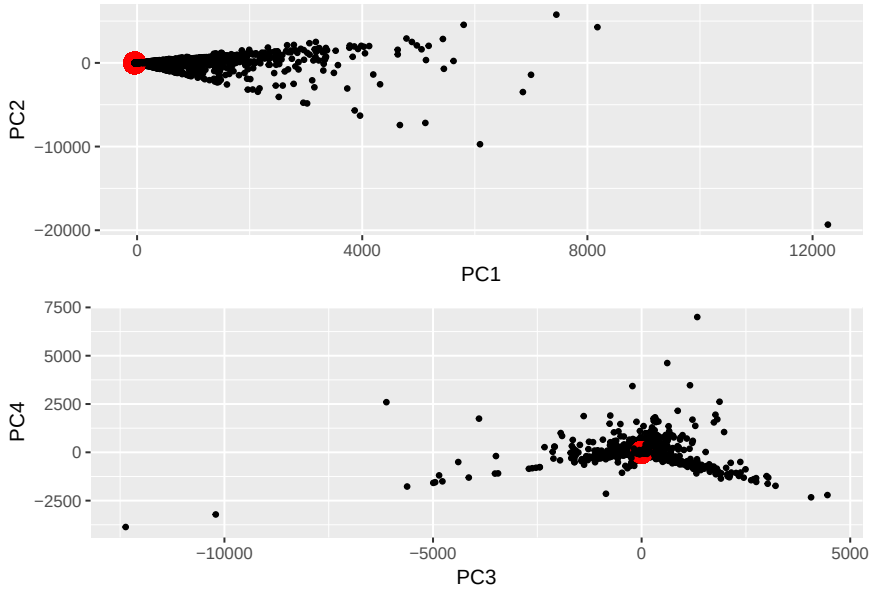
3.5 Find biomarkers for stem cells

By plotting the heatmap, we can verify that the expression of stem cell biomarkers is higher in WOX5 than in any other cell line. The biomarkers were filtered as genes which are expressed in WOX5 and not expressed in other cell lines; therefore, this is the result one would expect.



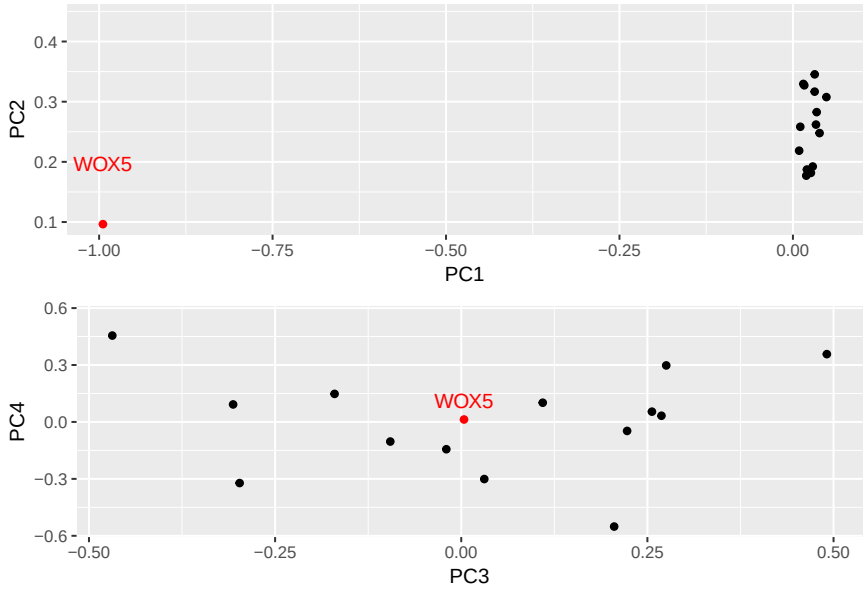
3.6 Visualize biomarkers in the PCA scores

By plotting the scores, we can see that all the selected biomarkers group together. However, there is no clear segregation from the rest of the genes.



3.7 Separate cell lines based on biomarkers

If we just consider the selected biomarkers, plotting the loadings shows how WOX5 is clearly separated from the other cell lines. Note that we should give more importance to the first plot, as it results from the combination of the first two PCs.



4 Practice 3: t-SNE for building transcriptome atlas map

4.1 Objective

In this practice, we will study the genetic expression of single-cells coming from a planaria sample. The main goal is to build a cell line transcriptome atlas map. To do so, we will use t-SNE and UMAP.

4.2 Data preparation

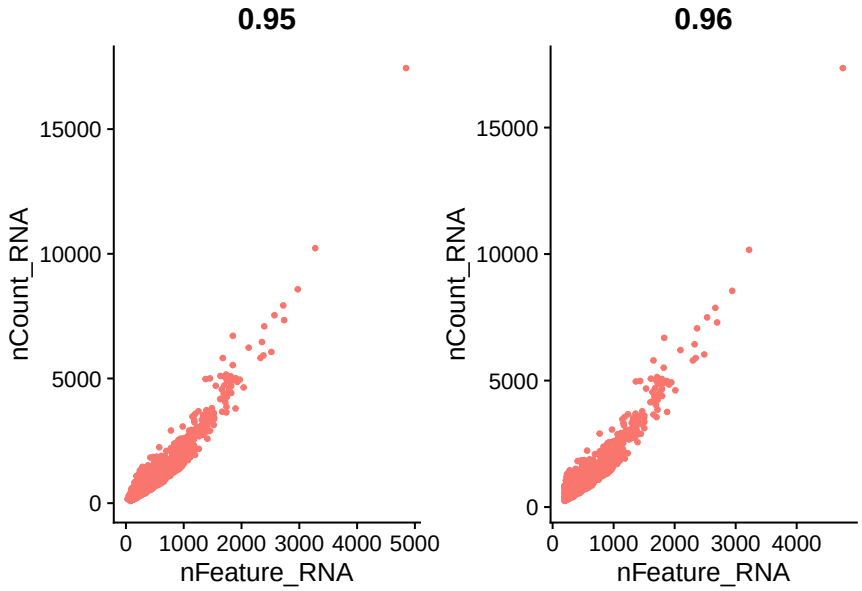
The first thing we do is to read the data and convert it into a sparse matrix (which result in significant memory and speed savings when many cells are zeros). From such matrix, we create a seurat object, container for both data and analysis (we create two objects, according to what is specified in excercises 1 and 2). Their differences refer to the number of cells in which a gene has to be expressed for it to be considered (1 in the first object vs 3 in the second) and to the number of genes which have to be expressed in a cell for it to be considered (1 in the first object vs 200 in the second).

4.3 Quality control, normalization, features selection and scaling

We explore the data with metrics. As expected, the number of genes and cells in the second object is lower.

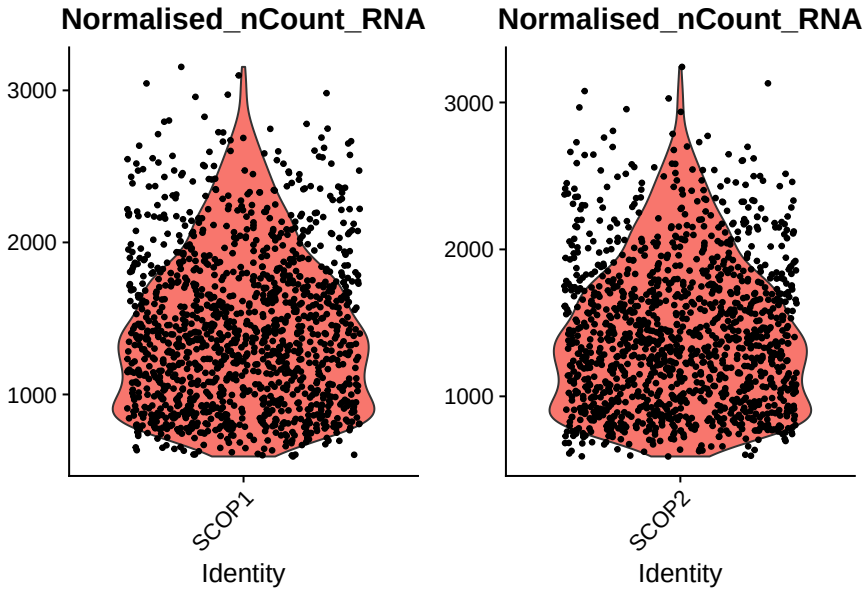
	seurat1	seurat2
Total number of genes	24883.000	18561.000
Total number of cells	1870.000	1301.000
Average number of genes per cell	459.648	595.586
Average number of reads per cell	970.954	1248.086

There is a clear correlation between number of genes (nFeature_RNA) and reads (nCount_RNA) per cell.

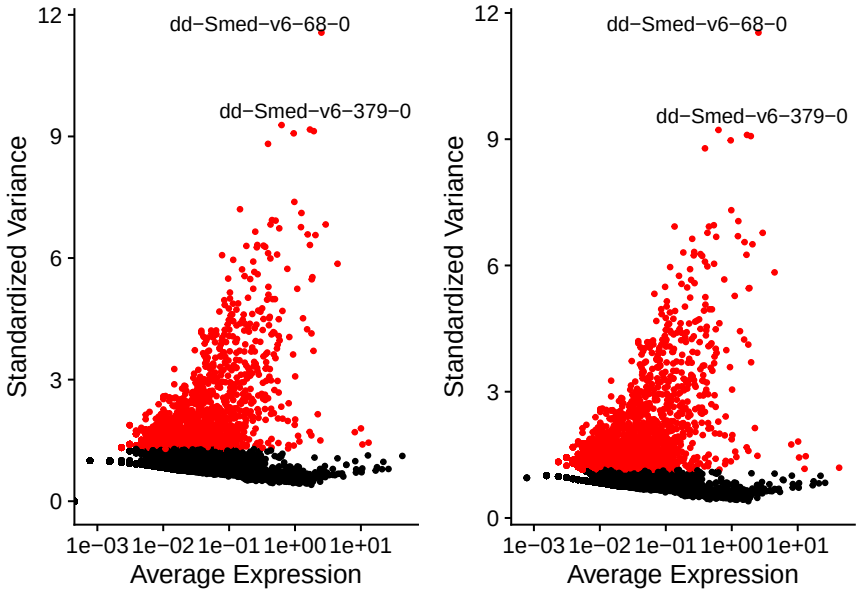


We perform a quality control. Low quality cells or empty droplets often have very few genes; cell doublets or multiplets often have to many genes. Thus, we keep those cells with not to few and not to many genes. It looks reasonable to filter cells with a number of genes between 200 and 2,500.

We perform the normalization, using the LogNormalization method. This normalizes the gene expression for each cell by the total expression, multiplies by a scale factor and log-transforms the result. If we compare the violin plots pre-normalization (not shown) and post-normalization, the normalization of the data is made evident (not all the point are grouped on the lower part of the plot).



We perform the feature selection by identifying the subset of genes whose expression varies the most from cell to cell. Focusing on these genes helps to highlight biological signals in single-cell datasets. We consider 2000 genes for the first seurat object and 3000 genes for the second seurat object (in red) and the 2 most variable appear labelled.



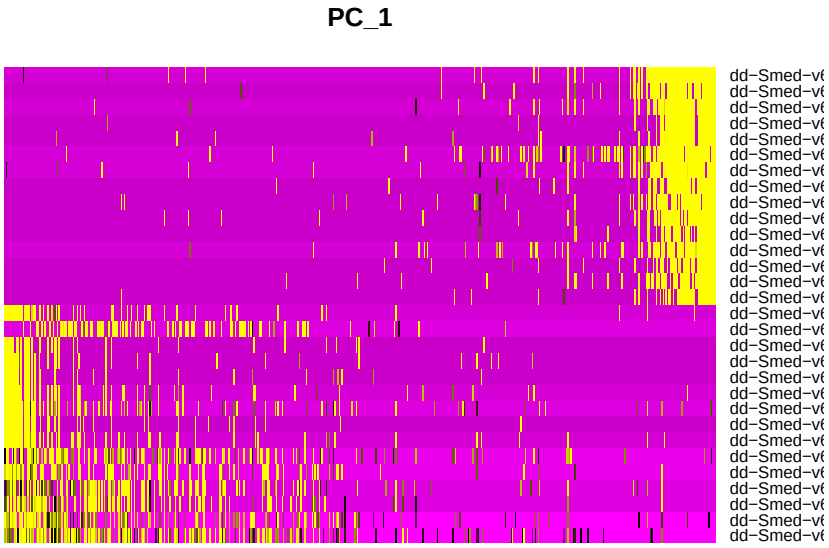
We perform the scaling by applying a linear transformation that makes mean expression and variance across cells 0 and 1. This is a way of regressing out variability (without specifying the source of variation removed).

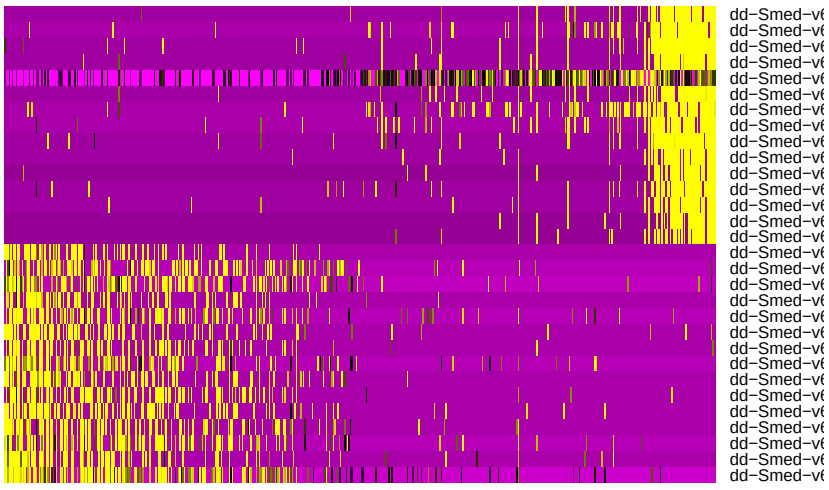
4.4 PCA, t-SNE and UMAP

We perform a linear dimension reduction with PCA.

Seurat clusters the cells based on PCA scores, with each PC representing a metagene that combines information across correlated gene sets. To identify the dimensionality of the dataset, there are two alternative methods. The JackStrawPlot function provides a comparison of the distribution of p-values for each PC with a uniform distribution; significant PCs show a strong enrichment of genes with low p-values. The ElbowPlot provides a ranking of PCs based on the percentage of variance explained by each one. Although we don't show this plots, they were examined and 10 PCs look like a good representation of the original dataset.

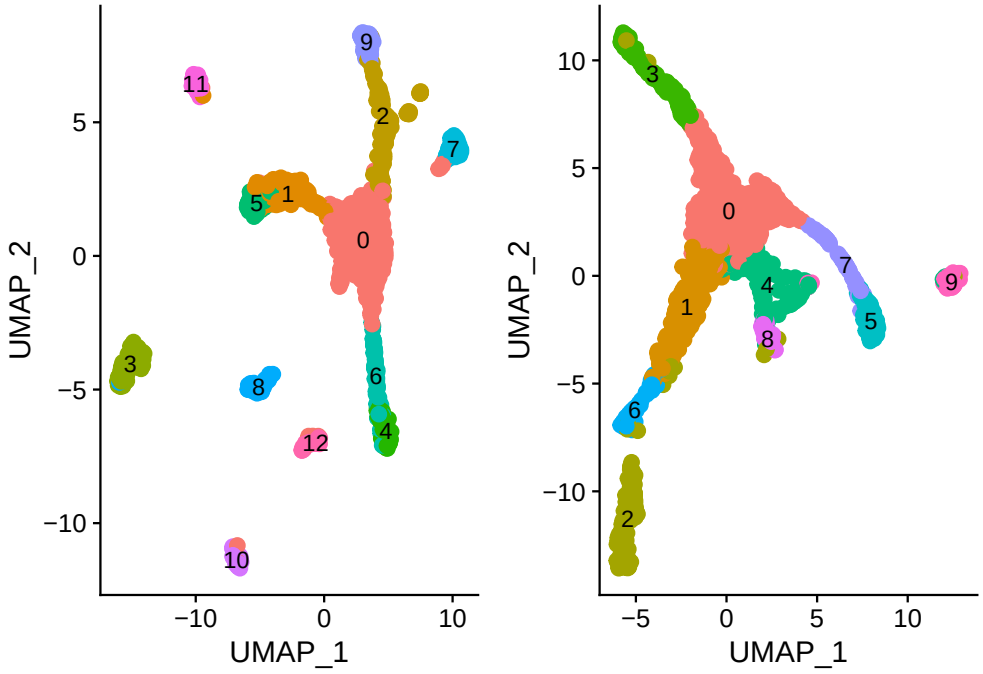
We visualize the PCA results with a heatmap, where cells and genes are ordered according to their scores. This allows for easy exploration of the primary sources of heterogeneity in a dataset. It can be useful when trying to decide which PCs to include in further downstream analyses. Although we just show the results for the first PC, 10 have been examined.





We perform a clustering analysis based on the previously identified significant PCs. This implies constructing a K-nearest neighbor graph and then clustering the cells through modularity optimization techniques. For the first object, we use 10 PCs and a resolution of 0.5; for the second object, we use 5 PCs and a resolution of 0.6.

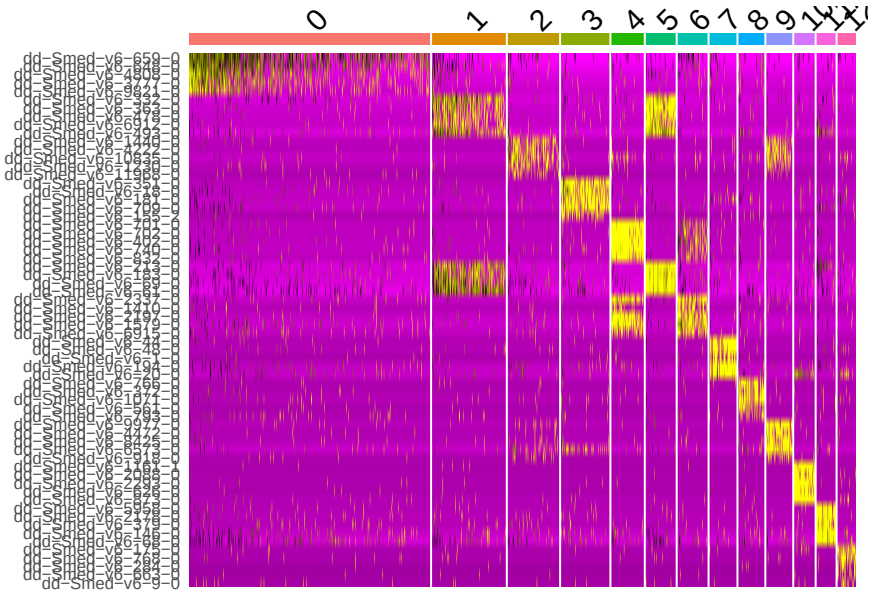
We use UMAP and t-SNE to visualize the data in low-dimensional graphs. Only UMAP results are shown. We chose this technique because it appears to give better results. In the preparation of the seurat objects, we set different initial conditions (beings more restrictive in the second object) and we did a different feature selection (being more restrictive in the first objects).In the cluster search, the only differences between both seurat objects was the number of PCs and the resolution. Increasing the resolution can lead to having larger clusters (the second object has less clusters of a bigger size). Decreasing the number of PCs used can lead to overlapping between the clusters (the second object has more overlapping between clusters than the first one). We will continue our analysis considering only the first seurat object.



4.5 Extract top biomarkers for each cluster

Biomarkers are genes that are expressed significantlly more (positive) or less (negative) in one cell line. We search for biomarkers comparing every cluster to the rest. The argument "min.pct" tests only those genes that are detected at a minimum percentage in the all cells; the argument "logfc.threshold" tests only those genes showing a minimum difference between cells lines. This table shows the first five biomarkers for each of the clusters previously identified.

We generate an expression heatmap for this biomarkers. This allows us to see how cells belonging to the same cluster share a common expression pattern. It is interesting to see that some clusters (1-5, 2-9 and 4-6) share expression patterns. They are probably corresponding to related cell lines.



4.6 Rename clusters based on expression distribution of markers

This is the given list of markers and the cell identities they are associated to.

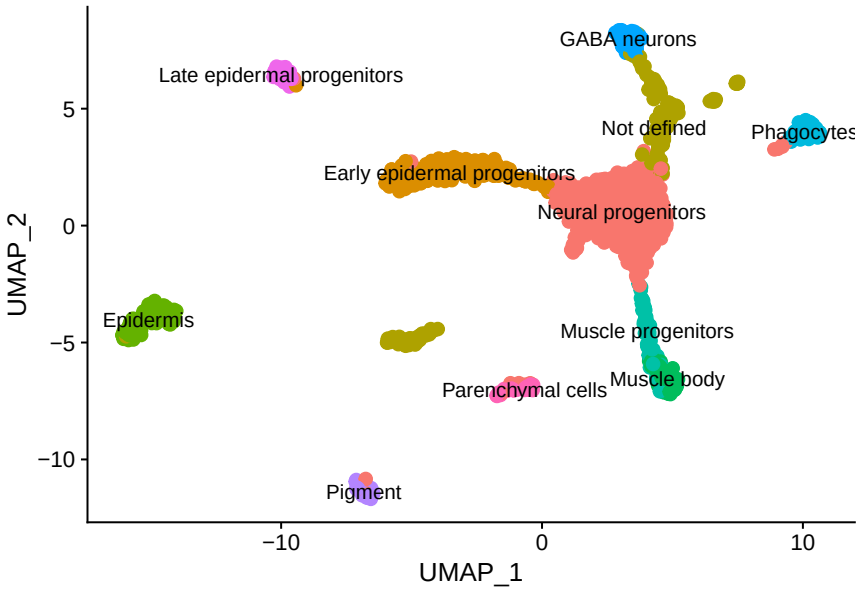
markers	cell_lines
dd-Smed-v6-61-0	Early epidermal progenitors
dd-Smed-v6-2178-0	Late epidermal progenitors
dd-Smed-v6-298-0	Epidermis
dd-Smed-v6-1410-0	Muscle progenitors
dd-Smed-v6-702-0	Muscle body
dd-Smed-v6-2548-0	Neural progenitors
dd-Smed-v6-9977-0	GABA neurons
dd-Smed-v6-48-0	Phagocytes
dd-Smed-v6-175-0	Parenchymal cells
dd-Smed-v6-1161-1	Pigment

We can create the correspondance between cluster and cell identities with two methodologies: Method 1 (M1) uses VlnPlot and FeaturePlot, that trace the gene expression across clusters visually. Method 2 (M2) searches for the genes among the biomarkers already retrieved in previous steps.

Both methodologies give as the same information. Now that we have the correspondencies, we can see how clusters 1 and 5 refer to the same cell line (Early epidermal progenitors) and clusters 4 and 6 refer to very similar cell lines (Muscle body and Muscle progenitors). The thing with clusters 2 and 9 is that cluster 2 has not been defined; we can assume it will be a similar cell line to GABA neurons, which is what cluster 9 stands for.

clusters	identities_M1	identities_M2
0	Neural progenitors	Neural progenitors
1	Early epidermal progenitors	Early epidermal progenitors
2	Not defined	Not defined
3	Epidermis	Epidermis
4	Muscle body	Muscle body
5	Early epidermal progenitors	Early epidermal progenitors
6	Muscle progenitors	Muscle progenitors
7	Phagocytes	Phagocytes
8	Not defined	Not defined
9	GABA neurons	GABA neurons
10	Pigment	Pigment
11	Late epidermal progenitors	Late epidermal progenitors
12	Parenchymal cells	Parenchymal cells

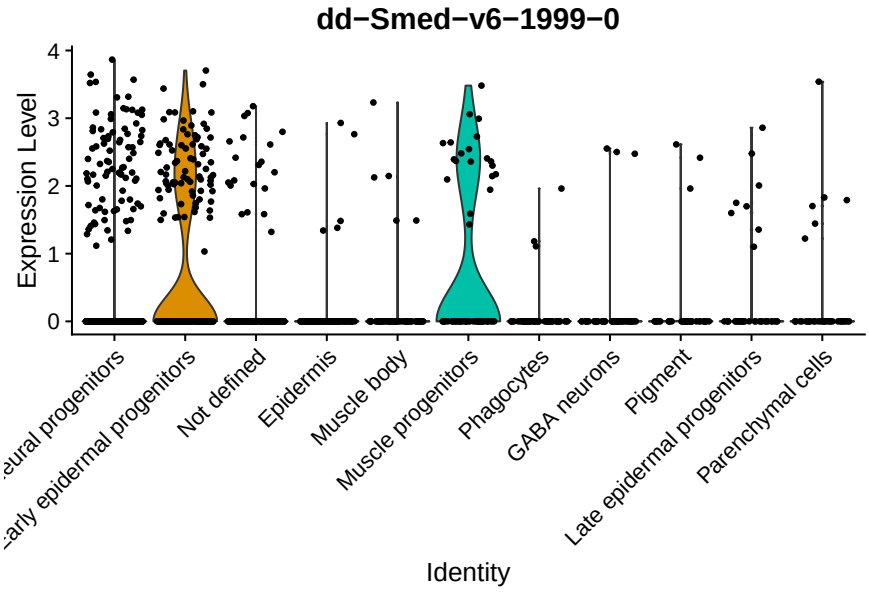
After including the cell identities names, we use UMAP and t-SNE to visualize the data in low-dimensional graphs. Only UMAP results are shown in the report.

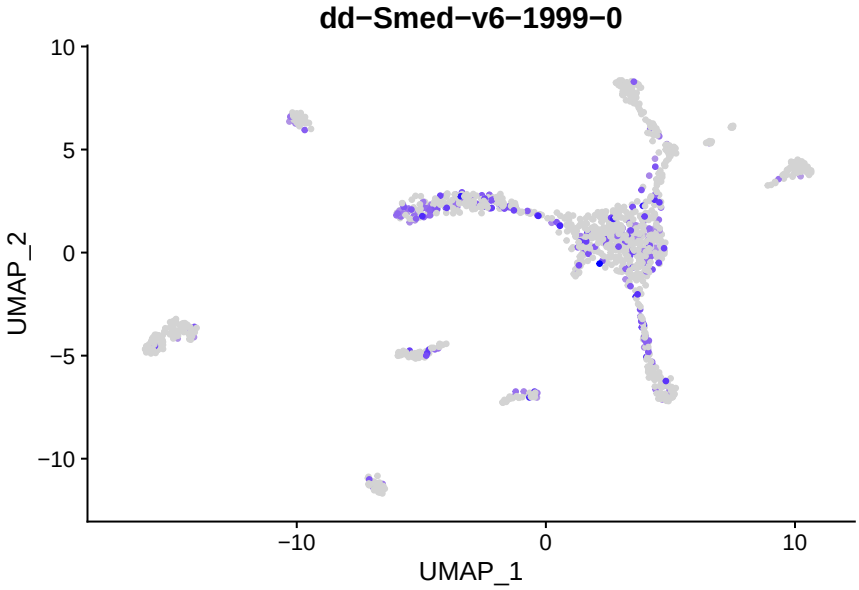


If we compare it with the tree reconstruction of planarian cell lines shown in [4], it seems like most of the general cell lines appear. This makes sense because our dataset is a sample of planaria and, even if we had took a local biopsy, it is expected to find a mixture of cell lines which are not tissue specific. Our central cluster includes neural progenitors; in the linage tree reconstruction, however, it is neoblast that appears. The next exercise will prove that such cell line also exists in our dataset, having an important role in the central cluster.

4.7 Show the distribution of a neoblast marker gene

Neoblasts are distributed all over the body and represent between 25-30% of all the cells. This explains why we can find one of its marker genes spread across the transcriptome atlas map (though it is more abundant in the central cluster).





References

[1] "RPubs - Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE", Online, Available: https://rpubs.com/Joaquin_AR/287787, Accessed: 28-Dec-2019.

[2] "t-distributed stochastic neighbor embedding - Wikipedia", Online, Available: https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding, Accessed: 28-Dec-2019.

[3] "Understanding UMAP", Online, Available: https://pair-code.github.io/understanding-umap/, Accessed: 28-Dec-2019.

[4] M. Plass et al., "Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics", Science, vol. 360, no. 6391, May 2018.