Transcriptomic III: Single-cell RNA-sequencing

Introduction

In the last decade the relevance of transcriptomic studies was hugely increased by technological progress which allowed us to measure the transcriptome of an individual cell. This is especially significant since most of the biological samples we are interested in (such as normal or diseased human tissues, or tumors) are composed of cells of several different types. Gene expression as measured by classic ("bulk") transcriptomic assays is thus an average over all the cell types present in the sample, weighted by their relative abundance. Therefore, when we compare transcriptomic samples, for example by class comparison, the differences we see are the combined result of differences in gene regulation between the two samples, and differences in cell type composition. Disentangling these two effects in bulk transcriptomic assays is a difficult problem, which is partially solved by *deconvolution methods*: Single-cell transcriptomics provides a direct solution.

More generally, single-cell transcriptomics allows us to determine all the cell types present in a given biological samples, and to study the transformation of one cell type into a different one, for example in the process of differentiation. Thus single-cell transcriptomics has become a crucial tool in the study of development. Understanding cell-type composition, and its dynamics, is also of fundamental importance in cancer biology: First, to understand the complex interactions between cancer cells and their microenvironment, composed of a variety of non-cancerous cells; and second, because single-cell transcriptomics allows us to study how tumor evolve, in particular to gain resistance to therapy.

Therefore the most basic task in the analysis of single-cell transcriptomic assays is the identification of all the cell types present in a sample. This is typically done by a combination of *dimensional reduction* and clustering algorithms. Dimensional reduction will be introduced in this chapter. The clustering algorithms introduced in chapter 3 can be and are used also for single-cell data, but specific, *graph-based clustering algorithms* are used more often, and will also be introduced in this chapter. The biological interpretation of the clusters relies on *marker analysis*, which is based on class comparison applied to cell types. Finally, we will briefly introduce *trajectory inference* methods, used to reconstruct how the dynamics of gene regulation guides the transformation of one cell type into a different one, for example in the process of differentiation, or in cancer evolution.

Dimensional reduction

Many types of data in modern biology have *high dimensionality*, meaning that a data point is described by a large number of values, as previously discussed in the context of transcriptomics in chapter 2: For example a sample in a human transcriptomic assay in which we measure the expression of all protein-coding genes can be thought of as a point in a ~20,000-dimensional space. This high-dimensionality creates a series of problems in the analysis of these data, the most obvious being how to represent the data in a way that can be understood and interpreted by our brain.

> **Dimensional reduction**
>
> Dimensional reduction allows representing high-dimensional data in a space of lower dimensionality, while preserving as much as possible the meaningful properties of the original data, and is used to allow data visualization and to simplify data analysis

We will first introduce a linear method for dimensional reduction, principal component analysis, and then briefly discuss newer, non-linear methods (tSNE and UMAP) that are often used in the analysis of single-cell transcriptomic data.

Principal component analysis

The classical method for performing dimensional reduction is *principal component analysis* (PCA).

> **Principal component analysis**
>
> Principal component analysis (PCA) performs a change of coordinate system in the original D-dimensional data, from the original coordinates to new coordinates (the *principal components*) that are *linear combinations* of the original ones, and such that:
>
> (1) The first principal component accounts for the maximum amount of variance that can be accounted for using a single coordinate
> (2) The second principal component is orthogonal to the first one and accounts for the maximum amount of residual variance (i.e. not accounted for by the first PC) that can be accounted for using a single coordinate orthogonal to the first PC
> (3) and so on until the D-th PC

Let's consider an example with $D = 2$ and $N = 100$ data points, shown in Fig. 1A. Looking at the data, we see that while they are indeed 2-dimensional, they tend to lie approximately on a straight line of slope ~ 0.5. So if we used a coordinate system in which one axis had

slope ~ 0.5 and the other axis was orthogonal to it, most of the variance would be accounted for by the first coordinate. This is precisely what PCA does. Figure 1B shows the new axes in red and blue (first and second PC). At this point we can show the data points in the new coordinate system (Fig. 1C), where in parentheses you see the fraction of variance explained by the two PCs.

Note that the total number of PCs is equal to the original dimensionality $D = 2$ of the data, and together the two PCs explain 100% of the variance. If we use a number of PCs equal to the original dimensionality of the data, we are simply changing the coordinate system, thus explaining 100% of the original variance without actually achieving any dimensional reduction. However when the original D is greater than 2, we can use the first 2 PCs to display our data on a plane, knowing that the coordinates we use capture as much variance as it is possible to capture using only 2 coordinates. In this sense PCA is the best possible dimensional reduction that can be achieved through a linear change of coordinates.

An example

As an example of the application of PCA to transcriptomics, consider the gene expression data for kidney cancer obtained by the TCGA project. These are (bulk) RNA-seq data of 1,020 tumor samples, in which the expression of 20,531 genes was measured. Thus each sample is described by 20,531 expression values, i.e. by a point in a 20,531-dimensional space. We can use PCA to represent the samples in a 2-dimensional space (Fig. 2A): the first two PCs explain, respectively, 10.6% and 8.15% of the variance.

Thus the 1020 samples have been represented in a plane. We can determine if such representation carries biologically useful information by noting that the samples are actually classified by TCGA into three subtypes of kidney cancer: kidney chromophobe (KICH), kidney renal clear cell carcinoma (KIRC), and kidney renal papillary cell carcinoma (KIRP). Coloring the dots according to this classification we see that tumors of the same type tend indeed to be close in the two-dimensional PC space. Therefore the first two PCs indeed capture biologically relevant variance.

Non-linear dimensional reduction

PCA is a linear method (in that the PCs are linear combinations of the original coordinates). Sometimes, however, most of the variance of the data is concentrated along a few dimensions, but in a non-linear way. Look for example at the data points shown in Fig. 3A: Most of the variance is indeed concentrated along a one-dimensional line, which however is not a straight line. Therefore, as shown in Fig. 3B, PCA cannot capture the fact that the data lie on such a line, and the data in the PC axes look very much like the original ones (Fig. 3C). There is indeed one "dimension" that explains most of the variance, except it is a curve that cannot be expressed as a linear combination of x and y.

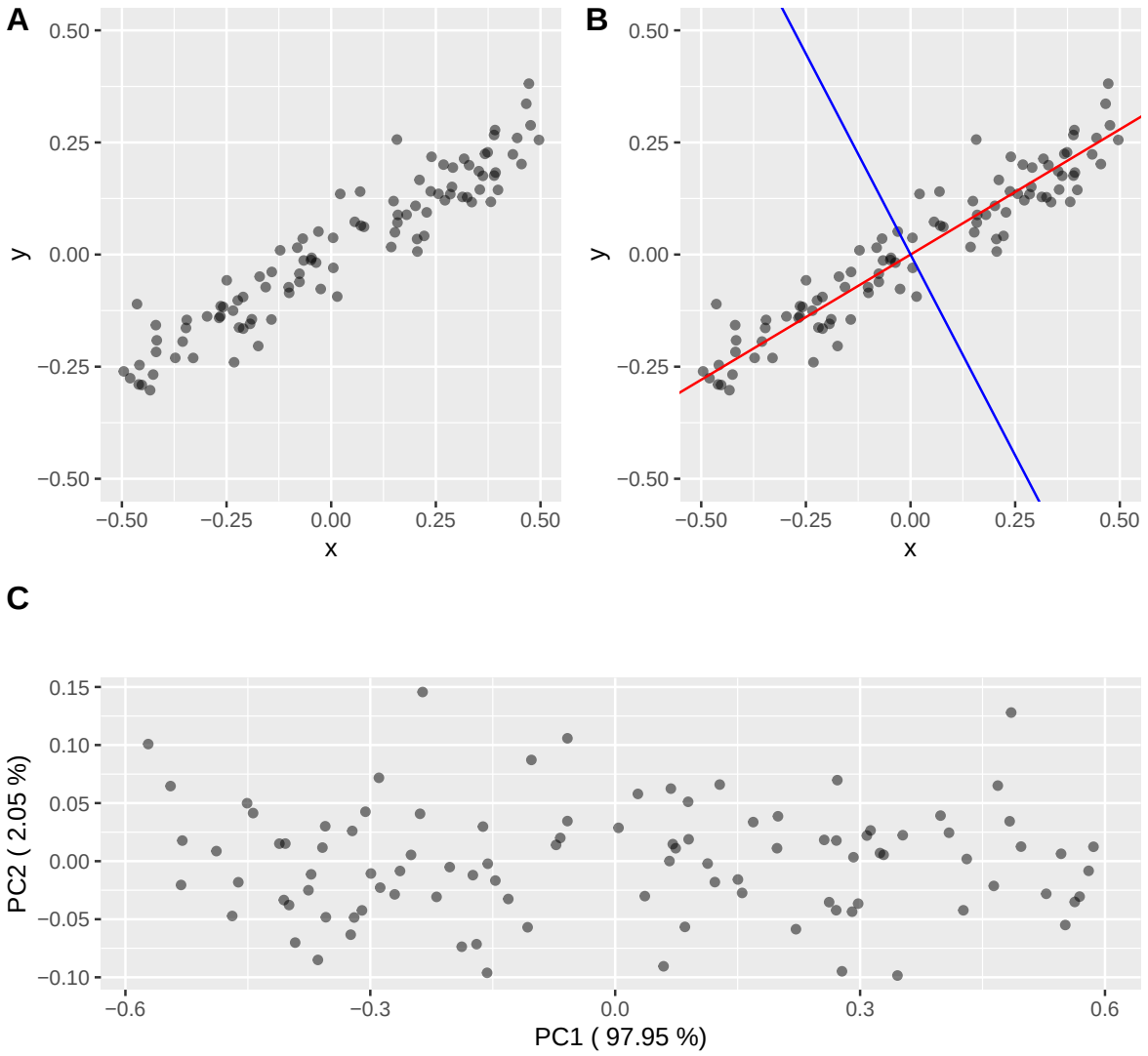

Figure 1: A: A 2-dimensional dataset with 100 data points. The points tend to lie approximately on a straight line of slope ~ 0.5. B: The new axes found by principal component analysis (first principal component in red, second in blue). C: The data shown in the new axes, with the precentage of variance explained by each component. Since here we are not actually performing dimensional reduction, as the original data were 2-dimensional, there are only 2 PCs, and together they explain 100% of the variance

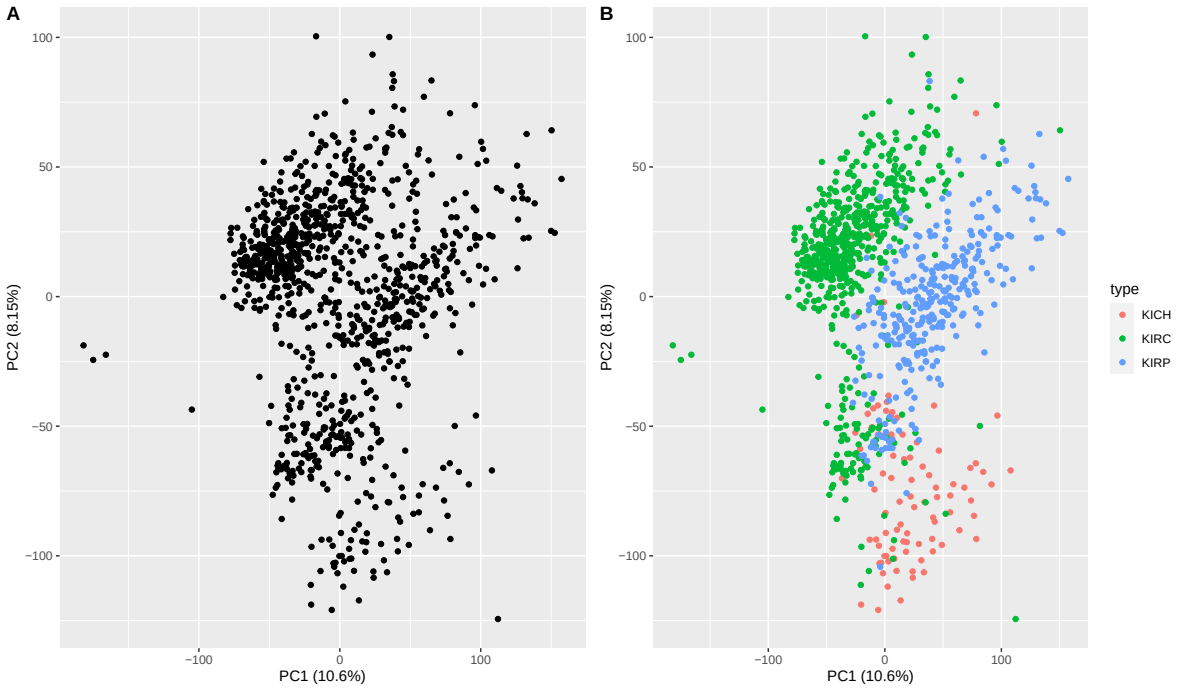

Figure 2: A: Dimensional reduction by PCA of the transcriptomes of 1,020 kidney cancer patients from the TCGA project. B: The samples have been colored by subtype of kidney cancer: The fact that colors are fairly well separated demonstrates that the first 2 PCs indeed capture much of the biologically relevant variance

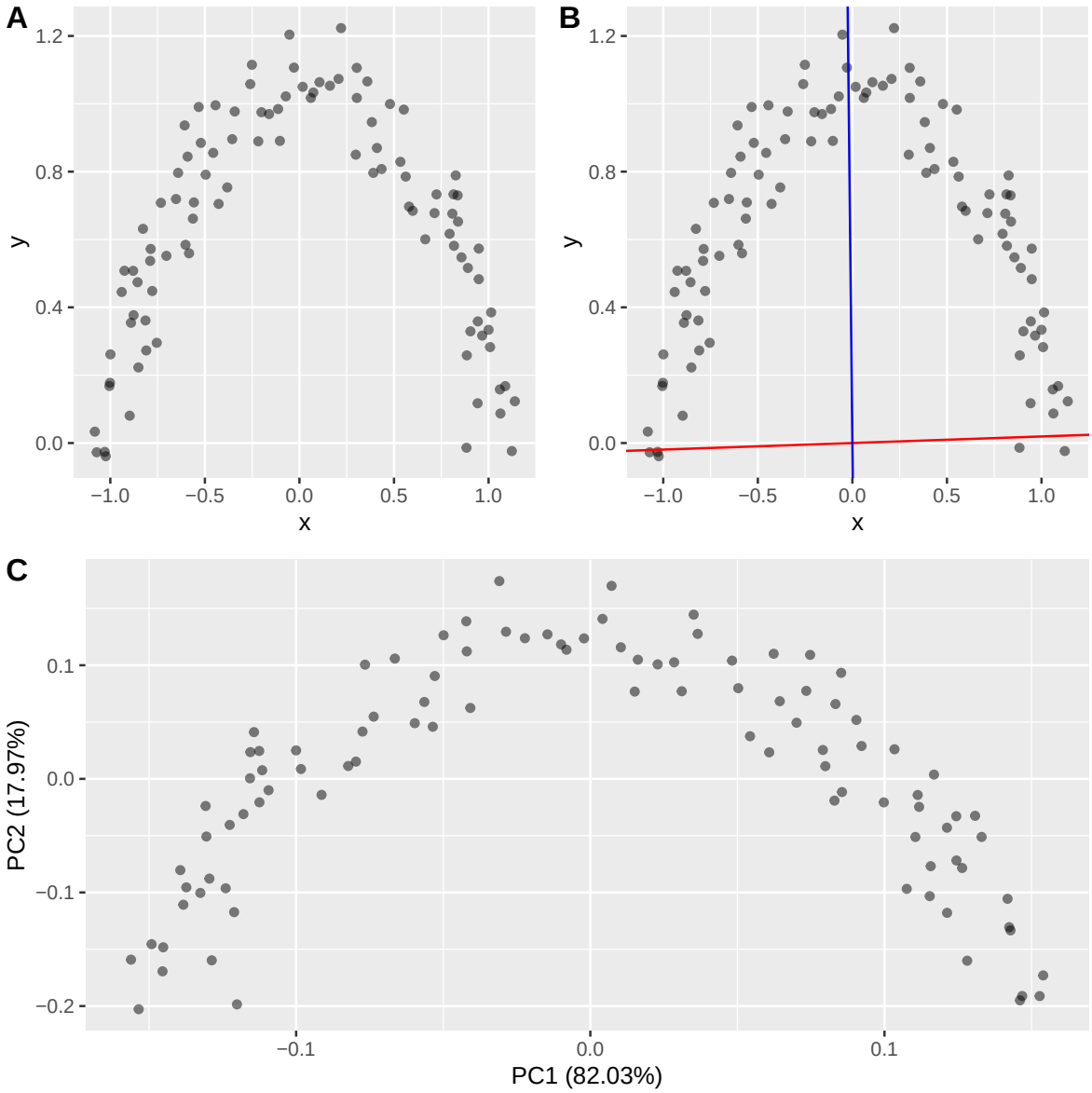

Figure 3: A: A 2-dimensional dataset with 100 data points. The points tend to lie approximately on a line, which however is not a straight line. B: The new axes found by principal component analysis (first principal component in red, second in blue). C: The data shown in the new axes look very much like the original ones: As a linear method, PCA cannot capture the fact that the data lie on a curve

Non-linear methods of dimensional reduction have been developed to deal with these cases (isomaps, locally linear embedding, Laplacian eigenmaps, and many others). In single-cell transcriptomics, two methods are very popular, namely *t-distributed stochastic neighbor embedding* (tSNE) and *uniform manifold approximation and projection* (UMAP). A thorough mathematical explanation of these methods is outside of the scope of these lectures, and we will limit ourselves to a description of the basic principles behind them.

tSNE was introduced in [@van_der_maaten_2008], and like many of the methods described in these lectures, did not originate in a biological context, but was presented as a general method to visualize high-dimensional data, with examples taken from image analysis. Given two points x_i and x_j in the original space, a similarity p_{ij} is defined based on their Euclidean distance. Importantly, p_{ij} decays exponentially with the square of the Euclidean distance between i and j, so that p_{ij} is significantly greater than zero only for pairs that are very close in the original space. Each point x_i in the original space is then mapped into a point y_i in the dimensionally reduced space, and a similarity q_{ij} is defined in this space. The points y_i are chosen so as to maximize the concordance between p_{ij} and q_{ij}[1].

Two main differences between tSNE and PCA (besides the non-linear character of tSNE) should be noted:

(1) In tSNE, the dimensions of the dimensionally reduced space are equivalent to each other, as opposed to PCA in which the first PC is "more important" than the following ones (explains more variance)

(2) While PCA is better at representing the large-scale features of the data (i.e. at placing highly dissimilar points far apart in the dimensionally reduced space), tSNE is better at smaller scales, that is at placing highly similar points close together in the dimensionally reduced space

When applying tSNE on the same kidney cancer data that we used to illustrate PCA we obtain the two-dimensional representation shown in Fig. 4. As advertised, tSNE separates the three types of kidney cancer more clearly than PCA (and suggests the existence of a separate group of tumors including samples classified in all the three groups, a potentially interesting lead).

Uniform Manifold Approximation and Projection (UMAP) [@mcinnes_2020] is another method for non-linear dimensional reduction, very popular in the field of single-cell transcriptomics, although it is again a very general method applicable to any high-dimensional dataset. It is based on Riemannian geometry and algebraic topology, and thus the theory behind it is even more outside of the scope of these lectures than that of tSNE. We will just mention that the two properties that distinguish tSNE from PCA, listed above, equally apply to UMAP.

[1] More precisely, this description applies also to tSNE's predecessor, calls SNE. While in SNE also q_{ij}, like p_{ij}, decays exponentially with the distance between y_i and y_j, in tSNE q_{ij} depends on the distance through a Student's t distribution, a modification that turns out to produce better low-dimensional visualization (see the original paper [@van_der_maaten_2008])

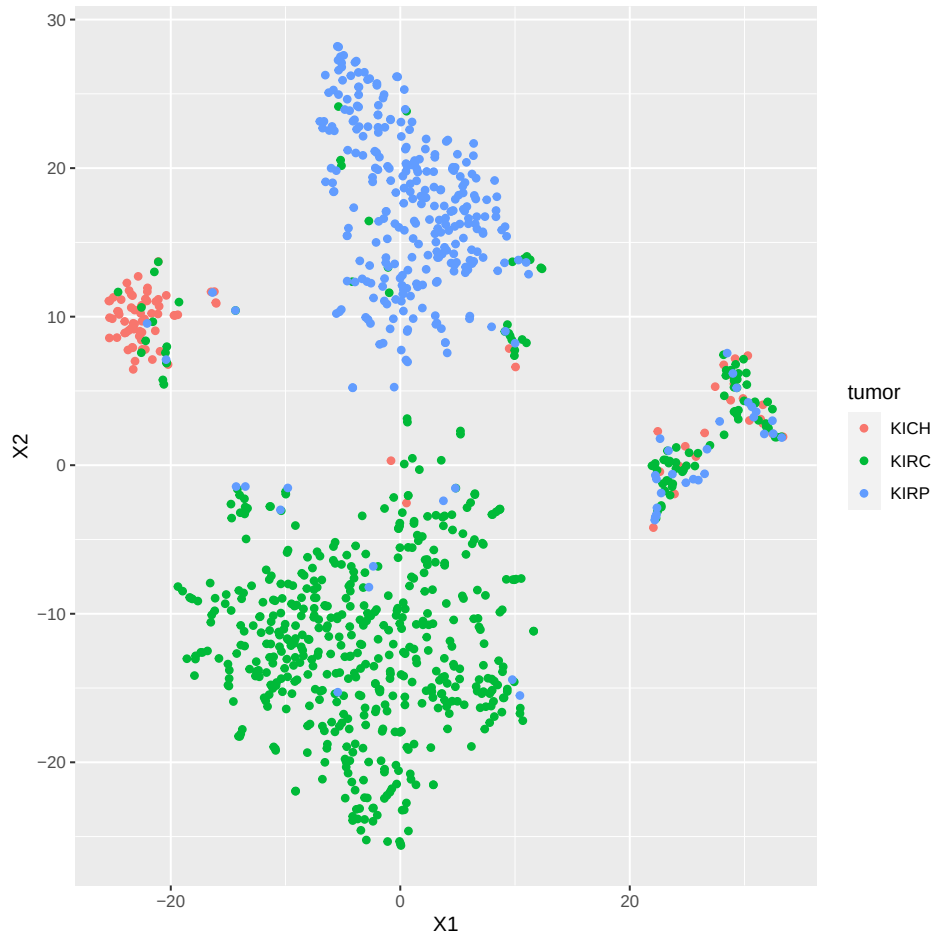

Figure 4: Two-dimensional representation of the kidney cancer samples from the TCGA using tSNE achieves excellent separation of the three known subtypes, although some samples from all three groups get grouped together in what appears to be a fourth subtype

The analysis of single-cell gene expression data

We will now describe the main steps of the typical analysis of single-cell transcriptomic data, which heavily rely on the dimensional reduction techniques described above. The main goals of the analysis are:

(1) to identify clusters of cells sharing similar transcriptomes and visualize them in reduced dimension. These clusters are interpreted as the cell types present in the sample

(2) to map the clusters/cell types into biologically known cell types by analyzing *markers*, i.e. genes specifically expressed by the cells in each cluster

Graph-based clustering

While the clustering algorithms that we have seen in chapter 3 can be used for single-cell expression data, a different approach is often taken, for example by the popular analysis suite Seurat [@hao_2021], where clustering follows the creation of a graph whose nodes are the cells and whose edges join cells with similar transcriptomes.

> **K-nearest neighbor (KNN) graph**
>
> A *K-nearest-neighbor* (KNN) graph of the cells measured in a single-cell transcriptomic assay is built by
>
> (1) computing a suitable distance between all pairs of cells. A possible choice is the Euclidean distance between cells after dimensional reduction obtained by principal component analysis
>
> (2) placing an edge from each cell to each of its K nearest neighbors. Note that this is a directed graph, since the fact that cell a is among the K nearest neighbors of cell b does not imply that b is within the K nearest neighbors of a

Note that the dimensional reduction used before computing inter-cell distances is *not* the one used for visualization, as it typically reduces the space to $D = 10$ rather than $D = 2$, and uses PCA rather than non-linear methods.

For example, consider the 20 cells represented in Fig. 5A after dimensional reduction (we will use 2 dimensions to allow graphical representation, keep in mind that a higher D is used in practice). The KNN graph with $K = 1$ is a directed graph in which each cell is joined to its nearest neighbor, shown in Fig. 5B, while for higher K we join each cell with its K nearest neighbors (Fig. 5C,D).

A graph naturally produces a clustering of its nodes through the concept of *community*:

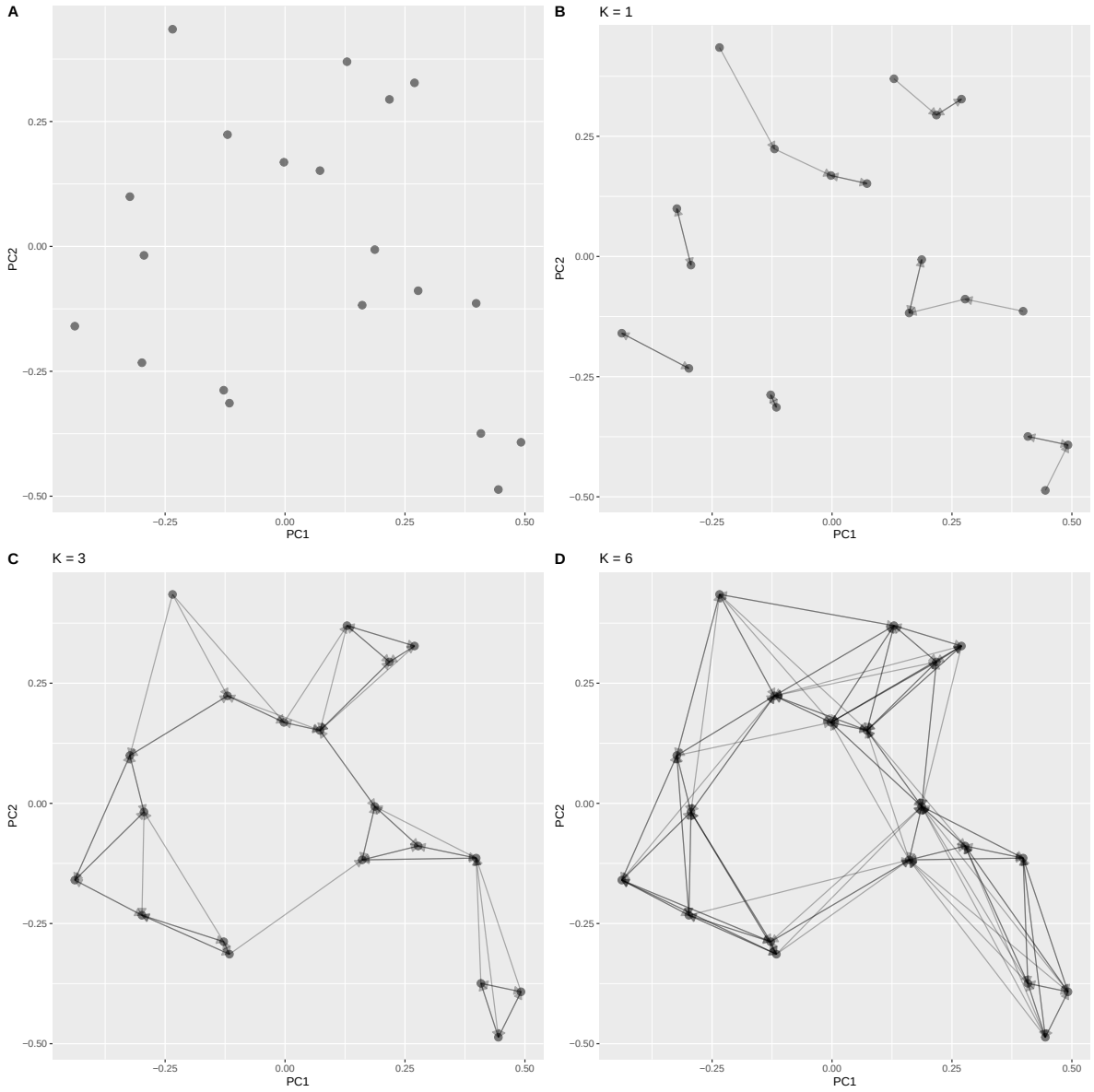

Figure 5: A: Twenty cells are shown after dimensional reduction to $D = 2$. B: For $K = 1$, the KNN graph joins each cell to its nearest neighbor (based on Euclidean distance in the dimensionally reduced space). C, D: For $K = 3, 6$, the KNN graph joins each cell to its three or six nearest neighbors. The graphs are directed, since the relationship 'being among the K nearest neighbor of' is not symmetrical

Community

A *community* in a graph is a set of nodes that are connected to each other much more than they are connected to nodes outside the community

A mathematically precise definition uses the concept of *modularity*:

Modularity

Given a partition of the nodes of the graph into clusters, the *modularity* of the partition is the difference between the fraction of the edges that connect nodes within the same cluster and the same fraction expected from a random partition

A high modularity implies that the partition of the nodes reflects the true community structure of the graph. Thus clustering will be performed by looking for the partition of the nodes that maximizes modularity.

Let us start with the KNN graph of our 20 cells, with $K = 6$, and a random partition of the cells into 3 clusters, represented as colors in Fig. 6A. To simplify the discussion we will disregard the directed character of the KNN graph. Since we used a random partition, we expect the modularity to be close to zero, and indeed the edges do not show any special tendency to join nodes of the same cluster: The modularity of this partition is -0.0786. The Louvain algorithm heuristically finds the partition with maximal modularity. In our case the resulting clustering is shown in Fig. 6B, which, of course, has much greater modularity (equal to 0.405) than the random partition.

An example with real data

A typical workflow for the analysis of single-cell RNA-seq data starts by partitioning the cells into clusters using graph-based clustering and representing them, usually colored by cluster, in a plane, using non-linear dimensional reduction. For example, using a dataset of 2,700 peripheral blood mononuclear cells (PBMCs)[2] with Seurat we obtain, using tSNE, the two-dimensional representation shown in Fig. 7A. Clustering divides the cells into nine clusters, shown as colors in Fig. 7B. As expected, the cells belonging to each cluster are close together in the tSNE representation, although the clustering was based on a different dimensional reduction (PCA with $D = 10$).

Each cluster thus represents cells sharing a similar transcriptome. The next step is to interpret biologically these clusters in terms of cell types, and to map these cell types into known ones: For example, in PBMCs we expect to find both lymphocytes and monocytes, and to recognize these cell types in the clusters produced by the analysis. This is done through the identification

[2]This dataset is used as an example in the Seurat tutorials, and is freely available from 10X Genomics here

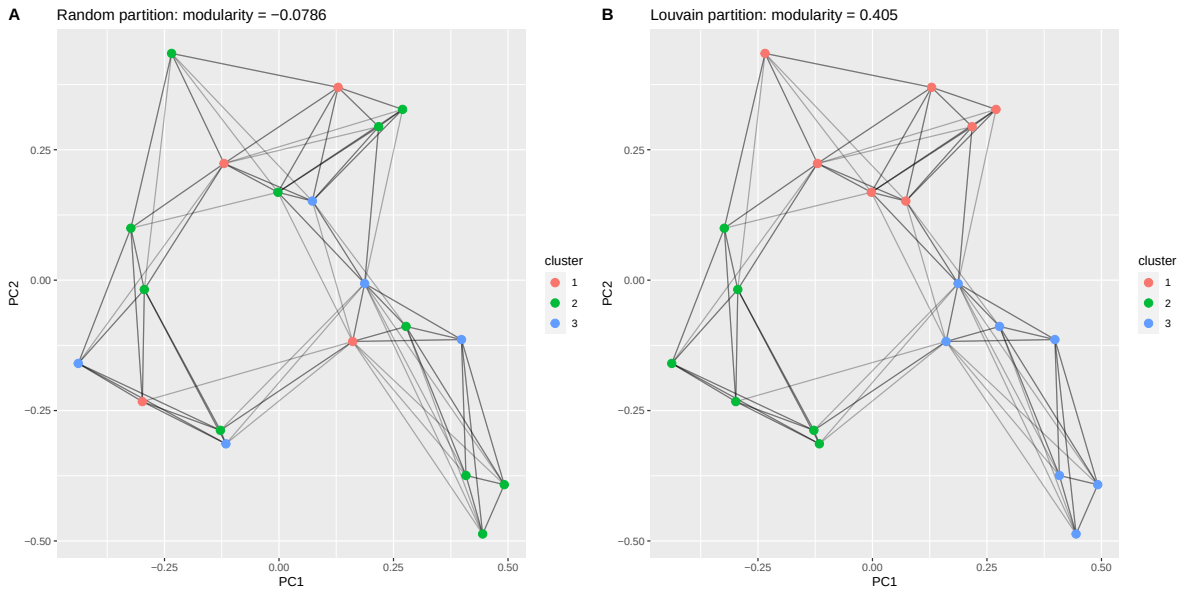

Figure 6: A: The cells are randomly divided into three clusters: The resulting partition has low modularity, i.e. cells are not preferentially connected to cells within the same cluster. B: The Louvain algorithm finds (heuristically) a partition that maximizes modularity

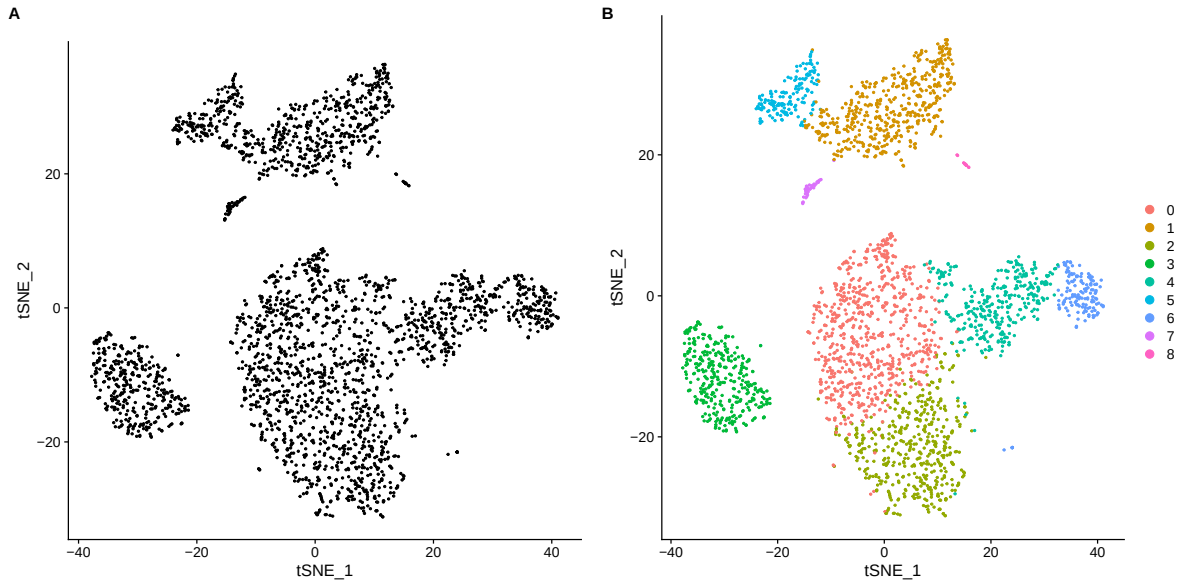

Figure 7: A: Two-dimensional representation of 2,700 peripheral blood mononuclear cells obtained by tSNE. B: clustering analysis divided the cells into 9 clusters, shown as colors

of *markers*, as explained below. However, first a few notes about some aspects of the analysis that were glossed over in our simplified treatment:

(1) While the data produced by single-cell RNA-seq are, just like those of bulk RNA-seq, integer counts (number of reads from each cell mapping to each gene), the KNN clustering procedure does not work directly on these counts, but is preceded by filtering of the cells based on various quality parameters, and normalization of the expression values followed by logarithmic transformation

(2) Seurat actually refines the KNN graph prior to clustering by assigning to each edge joining two cells a weight depending of how many neighbors the two cells share. This graph is called a shared nearest neighbor (SNN) graph and is used for clustering through modularity maximization instead of the original KNN graph.

Cluster markers

Interpreting the clusters of cells in terms of known or novel cell types is done by first identifying the *markers* of each cluster.

> **Marker**
>
> A *marker* of a cluster of cells is a gene that is differentially expressed when comparing its expression in the cells belonging to the cluster to that in all other cells

Both *positive* markers (overexpressed in the cluster of interest) and *negative* ones (underexpressed) are typically extracted. The identification of the clusters is thus a problem of *class comparison*, as discussed in chapter 2, in which the two classes to be compared are the cells belonging to the cluster of interest and all other cells. Since clusters typically contain hundreds of cells, non-parametric hypothesis testing methods can be used instead of the t-test described in chapter 2. Non-parametric tests have the advantage of not relying on specific assumptions on the distribution of the random variables being measured, such as the assumption of normality underlying the t-test. Usually they are less powerful than parametric tests, and thus require large number of replicate measurements. The Wilcoxon test is the most commonly used non-parametric replacement of the t-test, and can be used to find cluster markers.

For example, the genes *CD79A* and *MS4A1* are found to be the most significant markers of cluster 3, as shown in Fig. 8. Both these genes are known to be specifically expressed by B cells: Therefore we can use these results to posit that cluster 3 represents indeed B cells. By analyzing the list of cluster markers for genes known to be specifically expressed by known cell types we can thus propose a biological interpretation for each cluster.

A popular way of displaying graphically the correspondence between clusters and cell types is to show together the clustering and the expression of a gene specifically expressed by a cell type of interest, as in Fig. 9, which shows in a visually striking way that the cells classified

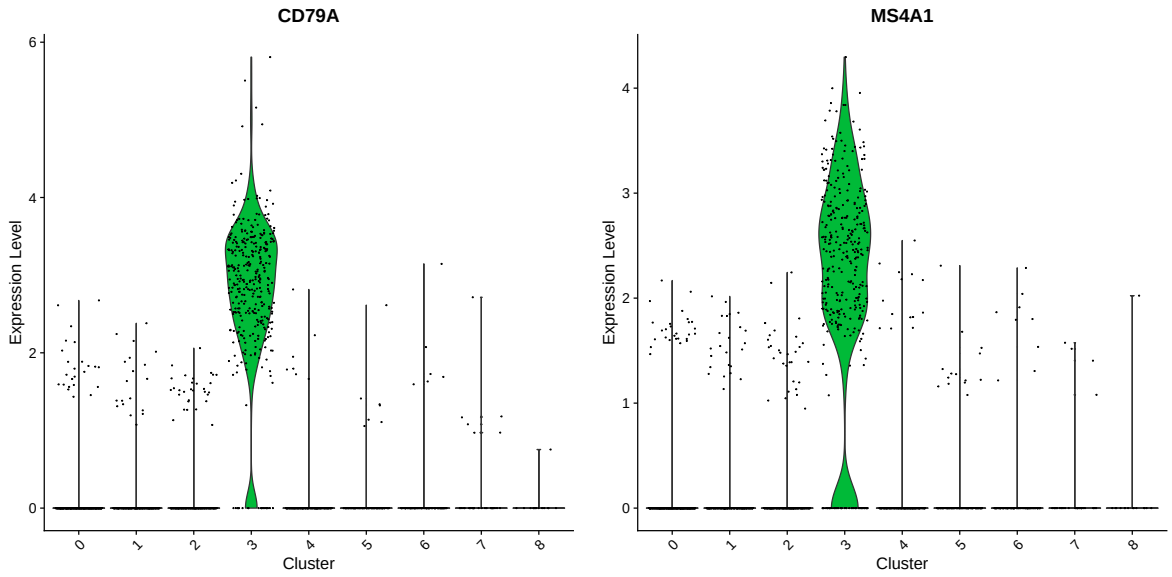

Figure 8: Expression of $CD79A$ and $MS4A1$ in the nine clusters. These two genes are markers of cluster 3. Since both are known to be specifically expressed by B cells, we can confidently interpret cluster 3 as mainly containing B cells

into cluster 3 are precisely those expressing two genes known to be specifically expressed by B cells.

Note that for each cluster, we define as markers all genes for which the Wilcoxon test shows significant differential expression between the cluster and all other cells. This P-value should be considered with some caution, as it is based on the questionable assumption that the individual cells represent *independent* measurements of the expression of the gene. Moreover, in some cases, the top markers by P-value are actually expressed in all the clusters, which makes them less useful for interpreting a cluster as a cell type. For example, the top marker of cluster 0 is RPS6, with a Wilcoxon P-value of $5.43\ 10^{-142}$, whose expression is shown in Fig. 10. Indeed RPS6 is a ribosomal protein, expressed by all the cells, but at higher levels in cluster 0. Thus, for purposes of interpretation, the top markers in terms of P-value are not necessarily the most useful ones.

Pseudo-bulk analysis

The identification of cluster markers is an example of class comparison applied to single-cell data. Just as in the case of bulk RNA-seq, class comparison can be used to answer many other biological questions: For example, one could be interested in genes that are differentially expressed between a diseased and a healthy condition in a given cell type (for example, we might be interested in the differences in gene expression of excitatory neurons in schizophrenia

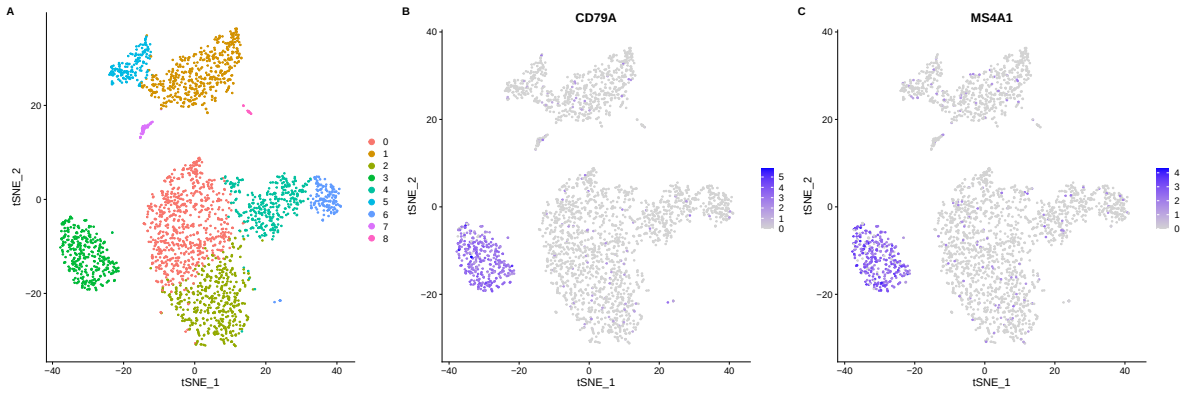

Figure 9: The tSNE representation of the cells is shown, in panel A, with colors corresponding to the clusters, and in panels B and C with a color scale representing the expression of *CD79A* and *MS4A1*, respectively. This representation makes it apparent that cluster 3 (bottom left in panel A) is strongly enriched in cells expressing these two genes, and can thus be interpreted as mainly containing B cells

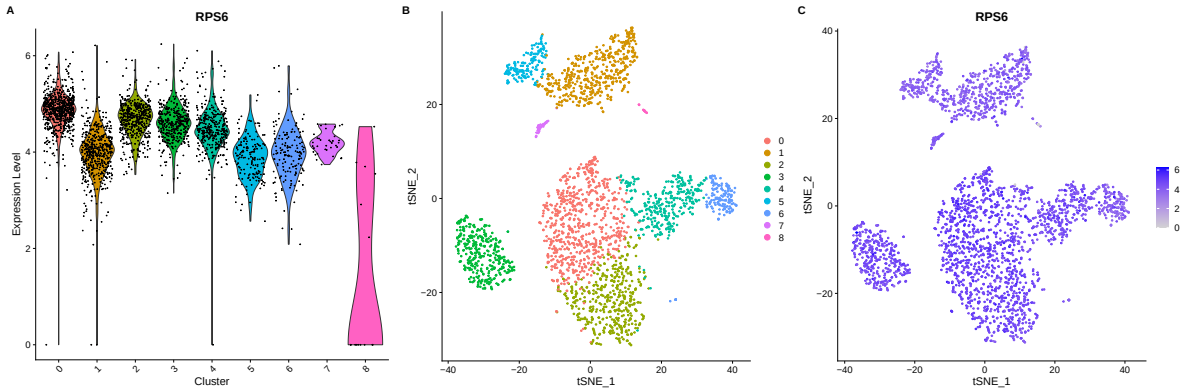

Figure 10: RPS6 is the most significant marker of cluster 0 according to the Wilcoxon test. However, as shown by both the violin plot (A) on the left and the tSNE representation (B, C), it is not a satisfactory marker, as it is robustly expressed in all clusters, although at higher levels in cluster 0

patients compared with healthy controls). To solve this problem, one could adopt an approach similar to that described for marker analysis, and perform a Wilcoxon test (or a t-test) for each gene, comparing its expression in the diseased vs healthy cells, and limiting the analysis to the cells previously classified into the cell type of interest. However, precisely as in marker analysis, this procedure is likely to produce many false positives because it is based on the hypothesis of independence of gene expression in the individual cells.

The problem can be solved by resorting instead to *pseudo-bulk* analysis:

> **Marker**
>
> In *pseudo-bulk* analysis the NGS reads assigned to all cells of a given cell type or cluster in a specific biological replicate are aggregated. Differential expression analysis is then performed with the same methods used in class comparison for bulk RNA-sequencing data

Thus, in our example we would need single-cell RNA-seq data for the (post-mortem) brains of several schizophrenia patients and several healthy controls. After identifying, in each patient, the cell cluster(s) recognizable as made of excitatory neurons, we would create pseudo-bulk expression profiles of these cells for each subject, and then find the genes differentially expressed between patients and controls. Thus the subjects, and not the cells, are used as replicates for statistical analysis - see e.g. [@ruzicka_2024]. Also marker detection can be carried out in this way, as long as biological replicates are available.

Trajectory inference

The clusters of cells that we built and used so far are static in nature: There is no notion of a temporal ordering of the clusters. This might seem natural, as most single-cell transcriptomic assays indeed capture and analyse cells that are simultaneously present in a biological sample. However, suppose the sample we are studying is undergoing some dynamical process, such as development, in which precursor cell types differentiate into progressively more specialized ones. This process will not be perfectly synchronous, so that at any given time during the process cells from several differentiation stages will be simultaneously present, and of course understanding which cell types "come first" in the process is crucial if we want to use single-cell transcriptomics to study development. Ideally, we would like to organize all cells into *trajectories* with a well defined temporal orientation. These trajectories could be linear (from a precursor cell type to a terminally differentiated one through a series of intermediate steps), but also more complex: When a single precursor can generate multiple types of differentiated cells, we expect the trajectory to take the shape of a tree, very similar to those we used in phylogenetic analysis in chapter 1[3].

[3] In phylogenetic analysis the fact that the trajectory must be a tree is obvious since two species cannot converge back after having diverged. For cells this is much less clear cut: First, there are dynamical biological processes, such as the cell cycle, in which the trajectory is, obviously, cyclical, and thus cannot be

Many algorithms have been developed to infer cell trajectories from single-cell transcriptomics. They can be broadly classified into two classes: network-based methods and biology-based ones[4]. The former typically build a network of cells (conceptually similar, and sometimes identical, to the KNN graphs we used for clustering) and use methods of graph theory to reconstruct the trajectory. The latter are based on specific biological assumption on how cell types evolve in time. We will briefly describe one method for each of these classes, our choice being based on the opportunities they present to introduce new useful concepts rather than any judgement on their performance.

Monocle: a graph-based trajectory inference method

Also graph-based trajectory inference methods are based on some assumptions about the way cell type mutate into each other during biological processes. The first, quite natural assumption is that changes in the transcriptome are gradual, so that cells that derive directly from each other are more similar in their transcriptome that cells that are separated by many intermediate steps. The second is an assumption about the topology of the trajectories. For simplicity, with the caveats mentioned above, we will consider trajectories described by a tree.

As an example, we will outline the procedure implemented by Monocle, one of the most used graph-based trajectory inference tools. The first step taken by Monocle is dimensional reduction, similar to what is usually done before KNN clustering, but performed using a non-linear dimensional reduction algorithm (the specific algorithm used changed with the successive versions of Monocle). The following steps require the introduction of a few new concepts of graph theory:

> **Weighted graph**
>
> A *weighted graph* is a graph in which a real (usually non-negative) number, the *weight*, is associated to each edge

The unweighted graphs we have considered so far can be considered as a special case of weighted ones in which every edge has the same weight.

> **Complete graph**
>
> A *complete graph* is a graph in which every node is connected to every other node by an edge

Clearly, complete graphs are informative only when they are weighted. Finally,

described by a tree. Also in development, cases have been described of transcriptionally distinct progenitors generating differentiated cells that are indistinguishable

[4] Although also network-based methods are based on some biological assumptions

Spanning tree and minimum spanning tree

A *spanning tree* of a connected graph is a subset of the edges that forms a tree and connects all the nodes. The *minimum spanning tree* (MST) of a weighted connected graph is the spanning tree with the minimum possible sum of edge weights

After dimensional reduction, Monocle creates a complete weighted graph with the cells as nodes and their Euclidean distances as weights, then finds the MST of this graph. The MST can have or not have branches. For example, for the 10 cells shown in Fig. 11A (after dimensional reduction), the MST (Fig. 11B) has no branches, while for the cells in Fig. 11C we obtain a branched MST (Fig. 11D).

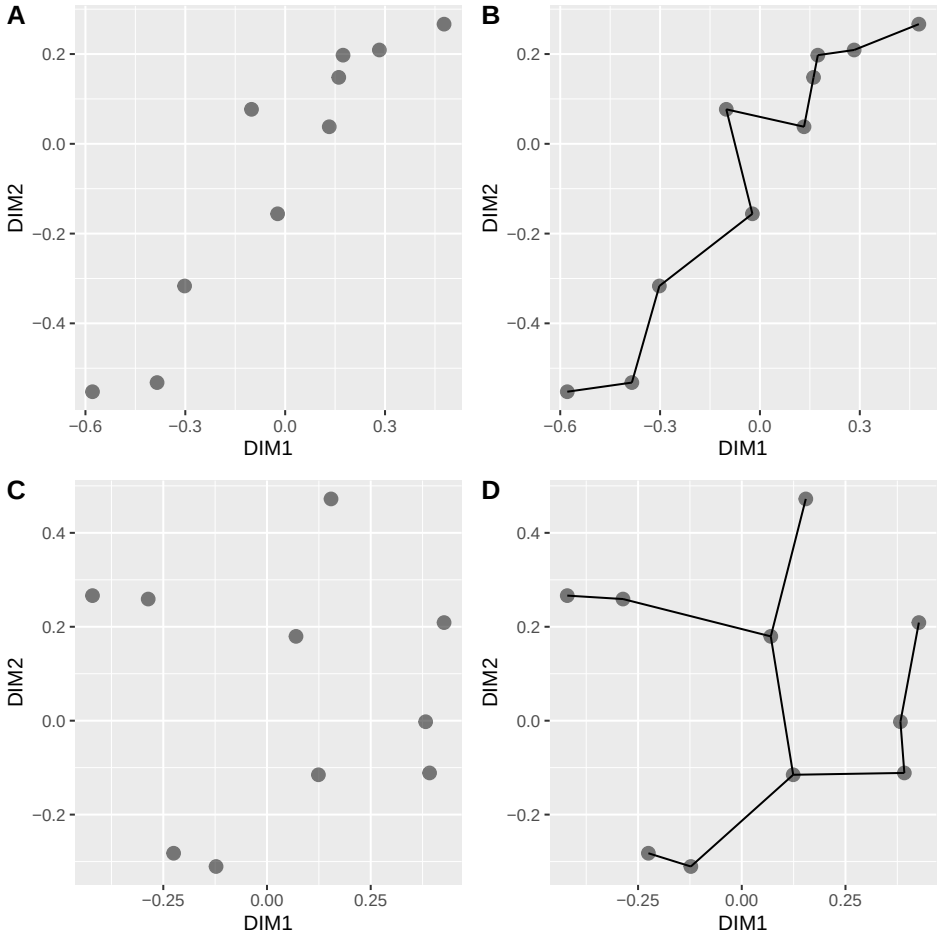

Figure 11: Minimum spanning trees of complete weigthed graphs in which each edge is weighted by the Euclidean distance between two cells. A,B: In this case the MST has no branches. C,D: an example of brached MST

Eventually, the MST will describe the trajectory: To this end, we have to assign a direction in time to each edge. Before we do that, note that the choice of the MST as the trajectory corresponds precisely to the assumption of gradual change of the transcriptome: Biologically, the MST represent the trajectory that minimizes the transcriptomic differences that are adjacent in time.

In the case of an unbranched MST, Monocle interprets the MST as describing a linear trajectory from one end of the MST to the other, and *arbitrarily* chooses one end as corresponding to the beginning of the process and the other as the end. Each cell is assigned a *pseudotime* which increases when going from the beginning to the end. The investigator has to use external, biological knowledge to decide whether the beginning and the end have been chosen correctly or they have to be reversed (this can be done for example by checking the expression of known pluripotency genes, which mark the beginning of the process)[5].

The case of a branched MST is dealt with in a more complex way, first by distinguishing between branches due to noise in the data and those of true biological significance. Cells belonging to the latter branches are then assigned a pseudotime value, and hence temporally ordered, with a technically complex procedure which we will not describe in detail.

CytoTrace: a biology-based trajectory inference method

CytoTrace can be considered a biology-based trajectory inference method because it introduces a new biological principle in the inference. This principle was established by analyzing a set of scRNA-seq datasets in which the differentiation trajectories were experimentally known, and looking for markers that significantly correlated with differentiation states. A marker is defined here as any quantity that (1) can be assigned to each cell using the scRNA-seq data and (2) significantly correlates with the known differentiation state. Remarkably, a quite simple marker, namely the number of genes detectably expressed by cells (*gene count* in the following) showed strong correlation with differentiation state. Specifically, it was found that cell gene counts *decrease* as cells progress along their differentiation trajectory.

CytoTrace starts by computing the gene count for each cell. According to the principle stated above, this could be directly used as a marker of differentiation state. However, it turns out that better performance can be obtained by (1) identifying the genes whose expression, in the specific dataset, most strongly correlates with gene counts (*top genes*) and (2) using as marker the *gene count signature* (GCS), defined as the average expression of the top genes in each cell[6]. The GCS of each cell is thus used to derive its pseudotime.

[5]The RNA velocity method described below allows establishing the direction of the trajectory in a way that is independent of the specific process under study

[6]More precisely, the GCS undergoes a smoothing process before being used as a marker

RNA velocity

Finally, we will discuss a method to infer trajectories based on a basic biological fact, namely that mature, spliced mRNA derives from its unspliced form, which contains intronic sequence. Consider a cell undergoing a change in time of its transcriptome, for example because it is transitioning from a progenitor to a differentiated state. Since the unspliced RNA is produced before the spliced version, the balance of spliced vs unspliced RNA of each gene can tell us something about the *future* transcriptome of the cell.

Luckily, it turns out that even if single-cell RNA-sequencing protocol enrich for polyadenylated RNA, the resulting reads still contain a sizable portion of intronic sequence, so that it is actually possible to assess separately the abundance of unspliced and spliced RNA for each (multi-exonic) gene. Different software packages implement this principle in slightly different ways (corresponding to different assumptions about the differentiation states of the cells present in the sample), but they all produce as their main output an evaluation of the *time derivative* of the expression of each gene in each cell, that is of how the transcriptome of a given cell is likely to look like in the near future. Roughly speaking, if the future transcriptome of cell A looks like the present transcriptome of cell B, we can conclude that A needs to placed before B in our trajectory reconstruction.

An example

As an example, we will consider the data collected in [@hermann_2018], who produced, in particular, single-cell RNA-sequencing of ~2,000 mouse spermatogonic cells. Figure 12 shows the results of applying the three methods we have described above to this dataset. Each method assigns a pseudotime value to each cell, while RNA velocity also shows the direction of the time derivative of the gene expression of the cell. It is reassuring that the pseudotime assignments of the three methods are strongly correlated to each other (correlation coefficients all > 0.9), considering that they are based on very different principles. However, this is a rather "simple" situation with cells placed along a single, unbranched differentiation trajectory: The concordance between the methods might not be as high when more complex contexts are examined.

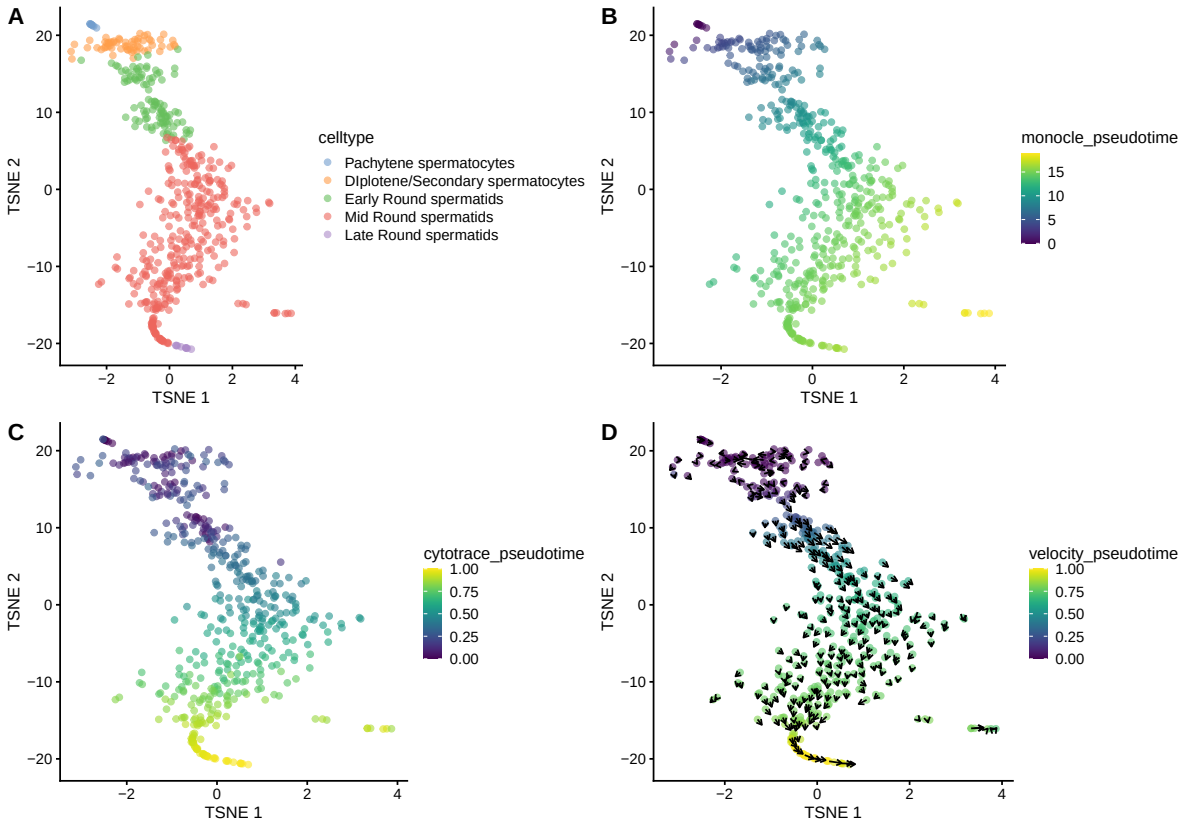

Figure 12: A: tSNE dimensional reduction of the trasncriptome of mouse spermatogonic cells. The cells are colored by cluster, and the clusters are interpreted based on the markers they express. B, C, D: The same cells colored by pseudotime as assessed by Monocle (B), CytoTrace (C), and RNA velocity (D). The arrows in D show the direction of the time derivative of the transcriptome of each cell