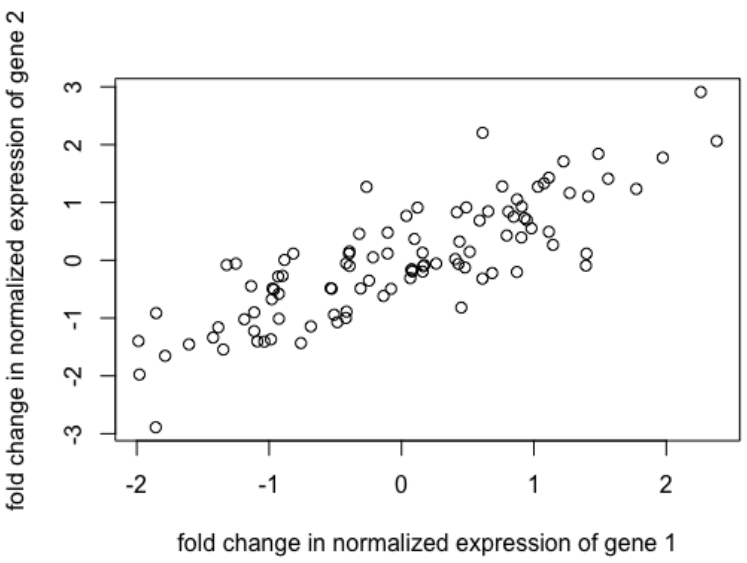
The Problem of High-Dimensional Data

Ideally, we would like to visualize every cell in a 1838-dimensional space (one dimension for each variably expressed gene), and count the number of cell types we have based on patterns of gene expression. However, given that we cannot visualize a 1838-dimensional space, we can use dimensionality reduction techniques to plot all the cells in our dataset in a 2-D space that preserves the distinction between cells.

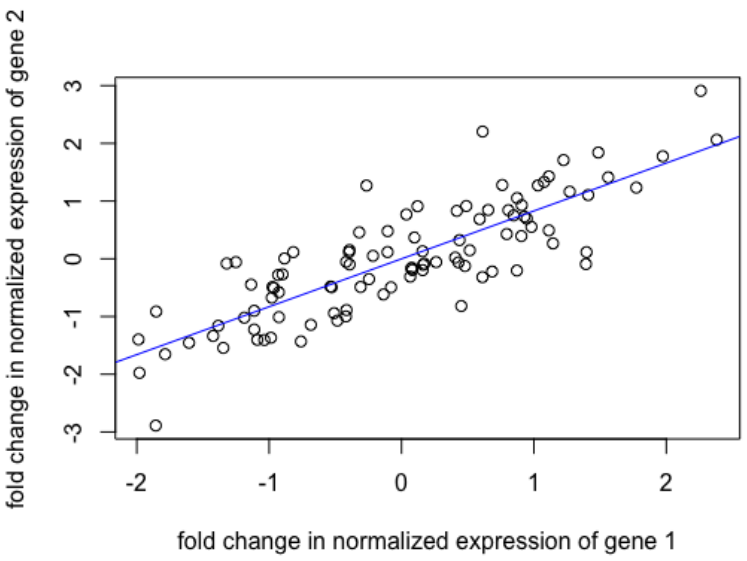
The Principle of PCA

The goal of PCA, or Principal Component Analysis, is to represent our dataset in fewer dimensions that capture most of the variance in our data. For instance, in our gene expression data, we would like to represent each cell as a function of a few such **principal components**, which we will see represent the major axes of variation in our dataset.

Let's imagine a dataset where we have collected data for 100 cells, and 2 genes. Thus, we have 2 dimensions in our data, one for each gene: we can plot the fold change in the normalized expression of gene 1 on the x-axis, and the fold change in the normalized expression of gene 2 on the y-axis. The data look like this, with each point as a cell:



In a previous section, you learned about describing variation in descriptive statistics. In a PCA you want to find a line along which most of the variation in the data is captured. In the following image, the blue line summarizes a new axis on which much of variation in the data is located:



This line is the first principal component of our data! This line is essentially a new axis (similar to the x- and y-axes), such that variation along that axis is as high as possible in our dataset. This is really what PCA does: **this method tries to find the best way to rotate the axes of our data, such that the first axis captures most variation in the data, followed by the second one, and so on.**

Now that we have found the first principal component, we can start looking for the second one. We will now try to find another line, orthogonal to the first principal component that captures most of the remaining variation not captured by the first principal component.

Reducing Dimensionality with PCA

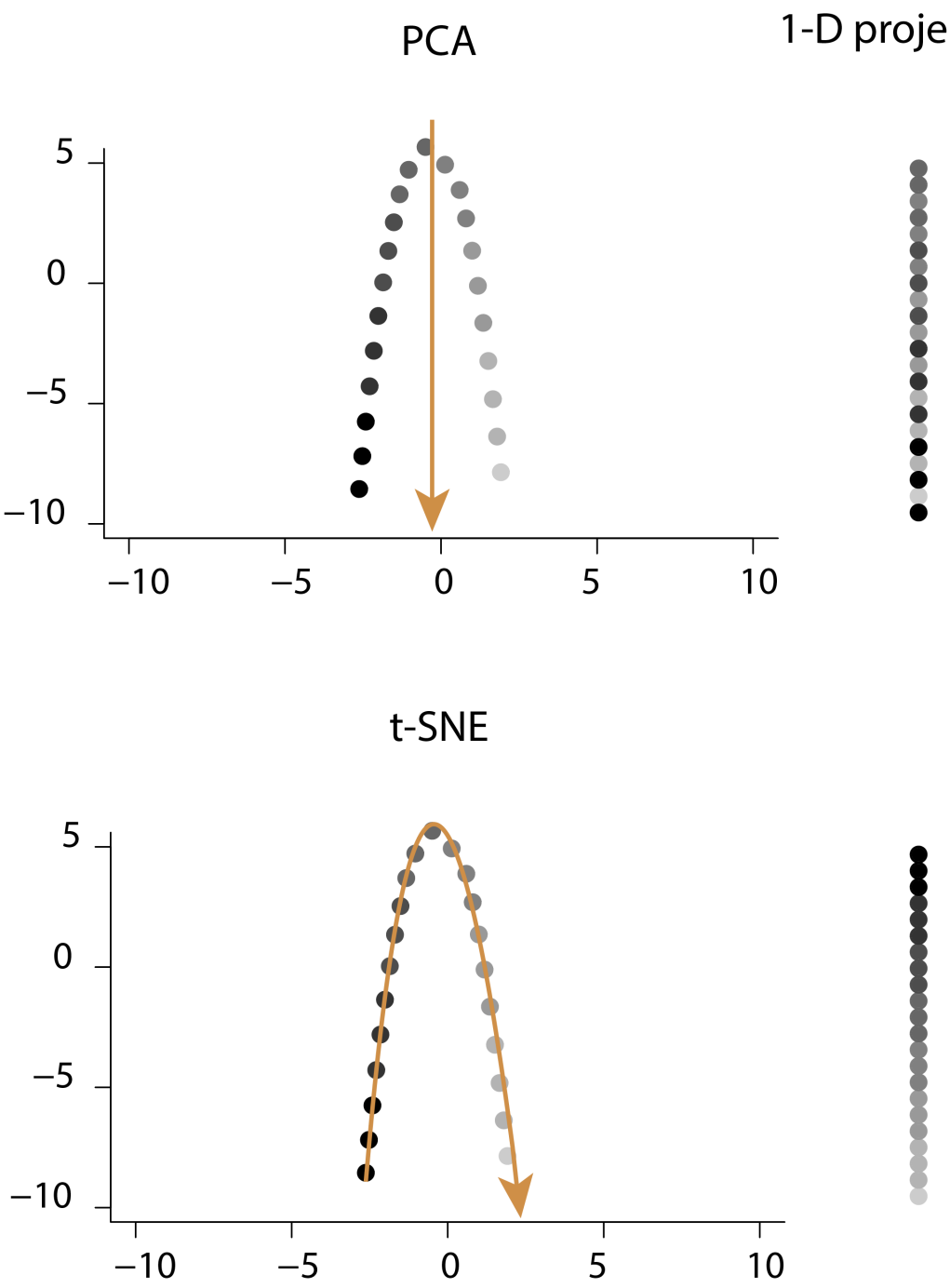
How is this process reducing the dimensionality of our data? We started with a dataset with 2 dimensions, one for each cell, and ended up with a rotated version of it, also with 2 dimensions, which are now the first 2 principal components (PC1 and PC2) for each cell.

The beauty of dimensionality reduction is that we now have a new set of axes, sorted by the amount of variance in the data that they explain. So, if someone asked you to summarize the dataset with 1 dimension per cell, rather than the 2 we had originally, you could report the projection of the data onto the first principal component and do a pretty good job.

In practice, PCA is used for dimensionality reduction of datasets with much higher number of dimensions. For instance, for gene expression data, we often summarize 2000-dimensional matrices with only 20-50 dimensions that explain more than 90% of the variation in the data!

Using t-SNE to Reduce Dimensionality

In addition to PCA, there are other sophisticated methods for dimensionality reduction that work in a non-linear way. One such example that is widely used for single-cell gene expression analysis is t-SNE. t-SNE in addition to being non-linear, also preserves local relationships between points. To illustrate the contrast to PCA, consider the following example where the points that were originally on a 2-D plane are projected on a 1-D line. The points are colored in a gray gradient according to x-axis values to help distinguish between points and emphasize the spatial relationships between points.



The projection from PCA distorts the original spatial relationship between dark and light colored points, whereas t-SNE preserves these relationships.

t-SNE is really good for visualizing relationships in 2 dimensions, but this method may lose some explanatory power when collapsing variation into two dimensions compared to dozens of principal components. Another important note is that t-SNE is not deterministic, so if you run this analysis multiple times, you will get different results each time. So, during the Exercise we will use a combination of PCA and t-SNE.

More Resources on Reducing Dimensionality

- You can watch a video on the premise behind PCA and t-SNE and find more examples of how PCA can reduce dimensionality at MIT OpenCourseware.
- For a more thorough explanation of t-SNE, read this blog post.
- Or if you are interested in the original formulation of the technique, you can read the original paper.
