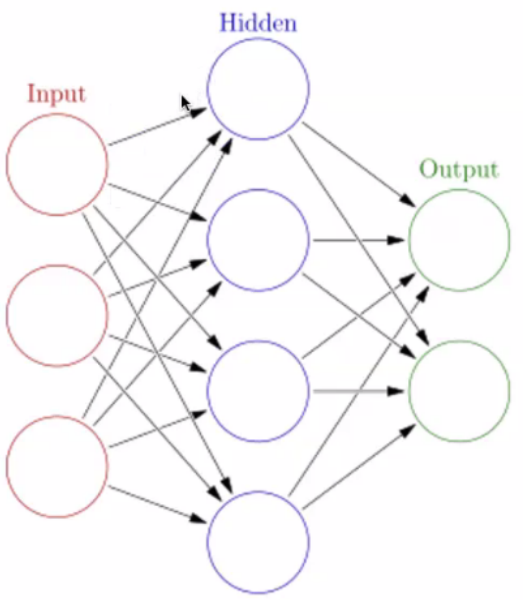
Neural Networks: Auto Encoders

TOPICS	
DATE	@May 25, 2022
MODULE	
PROF	Calogero
STATUS	☐

Autoencoders are a kind of **neural networks** (systems that get a solution even if they do not know the optimal path to it). They are characterized by an **input**, a **hidden network layer** which could be complex and the **output**. The network is trained with a dataset containing the info you would like to extract and another that does not. You want to optimize the learning to identify the majority of the events from the data you know. By modifying the hidden layers you can obtain other solutions.



Their main limitation is to have 2 datasets, one containing the info you want to extract and one not. You need a lot of data to feed to NN → single cell experiments are good.

Learning

Is based on the weight you are assigning to each of the edges connecting the nodes.

On the basis of the output you get you back correct the weights: this is the learning. We are trying to reduce the error rate on our known dataset.

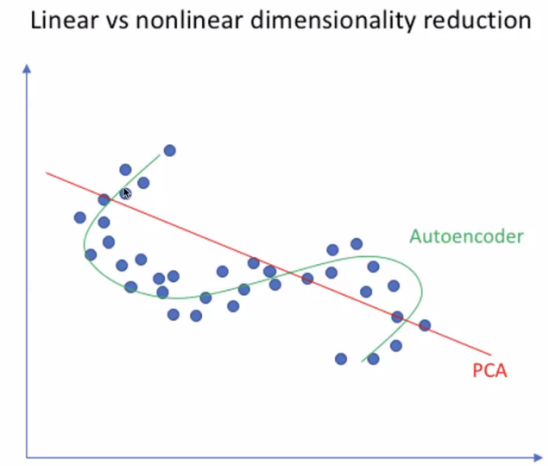
If the NN error gets stuck to a error rate which is not sufficiently low we have to try and modify the neural network.

- Learning is the adaptation of the network to better handle a task by considering sample observations.
- Learning involves adjusting the weights (and optional thresholds) of the network to improve the accuracy of the result.
- This is done by minimizing the observed errors.
- Learning is complete when examining additional observations does not usefully reduce the error rate.
  - Even after learning, the error rate typically does not reach 0.
  - If after learning, the error rate is too high, the network typically must be redesigned.
  - Practically this is done by defining a cost function that is evaluated periodically during learning.
    - As long as its output continues to decline, learning continues.
    - The cost is frequently defined as a statistic whose value can only be approximated.
    - The outputs are actually numbers, so when the error is low, the difference between the output (almost certainly a cat) and the correct answer (cat) is small.
- Learning attempts to reduce the total of the differences across the observations.

Autoencoders (AE) are used to perform sound compression: mp3 and mp4 std for sound and video are based on AE. The input are the data to be compressed, they are compressed in a smaller dimension and then decompressed to give you the information. Compressed information will be more or less accurate depending on the ability of the NN.

A critical issue in NN for scRNA seq is the **training dataset:** even in the case of AE, the AE need training before being applied to a new datasets. Criticalities are related to different experimental setting between the training set and the other set.

The **activation function** calculates weights for the edges connecting the input and output with the hidden network.



Linear AF→ compression similar to PCA (not good)

Non-linear AF is better: it will allow us to extract more information: can grab non-linear info from our dataset.

A typical AF should have 2 main characteristics: it should be non-linear and should be able to be differentiable.

ReLU = rectified linear unit returns 0 is the input is ≤ 0. Whenever the input values are > 0, it returns that specific inputed value.

Function:

$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases}$$

or (another way to write the ReLU function is...)

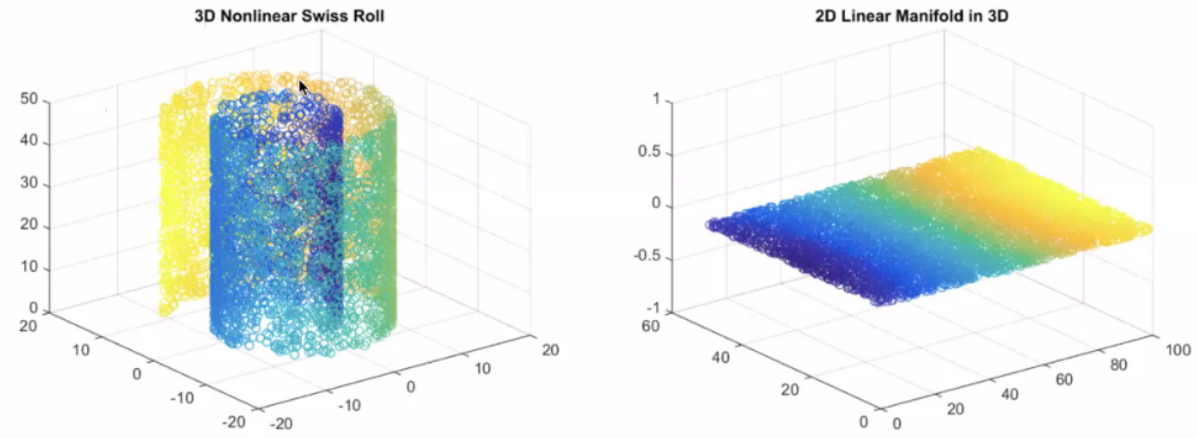
$$f(x) = \max(x, 0)$$

Derivative:

$$f'(x) = \begin{cases} 0 & for \quad x < 0 \\ 1 & for \quad x \geq 0 \end{cases}$$

The hidden layer in which we are compressing the data is representing the info in a low dimensional space without loosing much of it. mp3 are sufficiently good so that a normal ear cannot distinguish it. The compressed version is used to have the sound in a smaller space

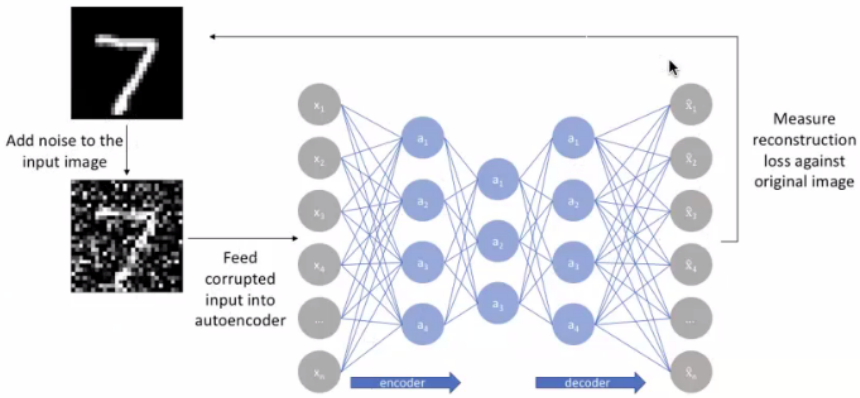
For higher dimensional data, autoencoders are capable of learning a complex representation of the data (manifold) which can be used to describe observations in a lower dimensionality and correspondingly decoded into the original input space.



The spiral on the left has color gradient changing over the length. The roll itself is less important and can be visualized simply in a smaller dimensional space keeping the color gradient.

In general, the aim of the AE is to find out a balance between the size of the input layer and the size of the inner compressed layer: compression won't be sufficiently similar if the inner layer is too small and won't be useful if it is bigger. It must be useful in compression but not too much otherwise we loose to much quality.

AE are used to **denoise scRNA data**:



- One of the most commonly used is a denoising autoencoder:
  - With this approach, **our model isn't able to simply develop a mapping which memorizes the training data because our input and target output are no longer the same.**
  - Rather, the model learns a vector field for mapping the input data towards a lower-dimensional manifold, which describes the high density region where the input data concentrates; if this manifold accurately describes the natural data, we've effectively "canceled out" the added noise.

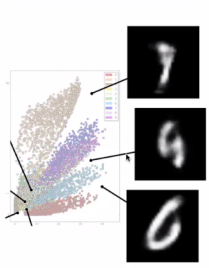
We remove the non-necessary info: the system learns to see the important part of the picture (we input many many of these images and it keeps only the part which is conserved let's say)

sparse AE have a very large inner layer but not all the nodes are used in the training. Anytime you do the training only a subset of the nodes are fired: the ones that are always fired are used to build the input data in a denoised version.

AE have still some limitations

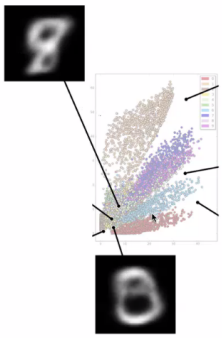
- gaps in the latent space → case of AE for cleaning up images of hand-written numbers: in this hidden space, some picture formats are actually ending in an area of the hidden representation where there are no data. This kind of pictures cannot be extrapolated and are lost → not all the characters are picked up from this network.

This diagram shows us the location of different labeled numbers within the latent space. We can see that the latent space contains gaps, and we do not know what characters in these spaces may look like. This is equivalent to having a lack of data in a supervised learning problem, as our network has not been trained for these circumstances of the latent space.



- separability in the latent space → there is a point of overlapping which does not allow discrimination between 9 and 8 here in the image. This causes errors in the assignment of the characters.

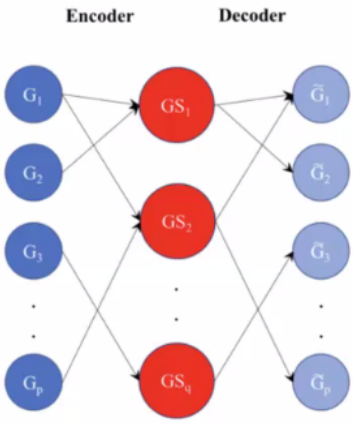
This diagram shows us the location of different labeled numbers within the latent space. Another issue is the separability of the spaces, several of the numbers are well separated in the above figure, but there are also regions where the labeled is randomly interspersed, making it difficult to separate the unique features of characters (in this case the numbers 0–9).



Here at MBC, following the idea of another group of research investigating AE

Each hidden layers get info from all the input layers, so you have no idea of the significant ones. Sparsely connected AE have the hidden layer not fully connected but only connected on known biological features (in SC we have gene information: info must control more genes, such as association kinase-target). Transcriptional info is used to extrapolate info related to the upper layer modulators.

- The encoding/decoding functions are only one layer deep and these layers are sparse (not fully-connected like standard autoencoders), with connections based on known biological relationships.
- Each encoded node represents a gene set and only receives inputs from gene nodes included in the set.
- These new methods can recover known biology.



Another example is Input = genes regulated by TF. Hidden nodes = TF.

All the info used are experimentally validated

https://bioinfo.uth.edu/kmd/

https://mirtarbase.cuhk.edu.cn/~miRTarBase/miRTarBase_2022/php/index.php

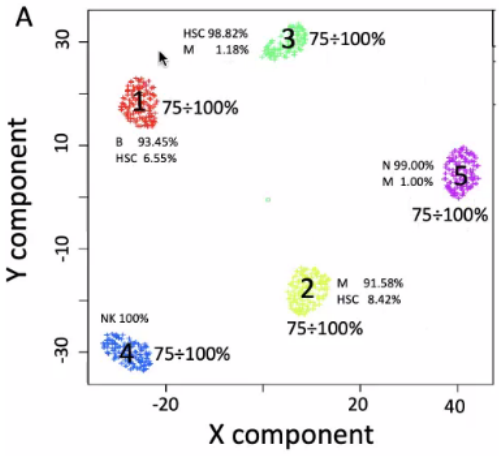
https://www.grnpedia.org/trrust/

http://www.cuilab.cn/transmir

TransmiR v2.0 database

One adv of this representation is that miRNA cannot be detected by scRNA analysis. However they control genes which can b detected in such a way → we can get info about the miRNA by the expression of their target genes in a reversed way via NN.

Test dataset with different kinds of elements: stem cells (3), cancer cells 1, naive T cells (5), monocites (2), NK (4). This is the clustering if we use all the genes. *Are we able to reproduce this clustering only using TF information?*

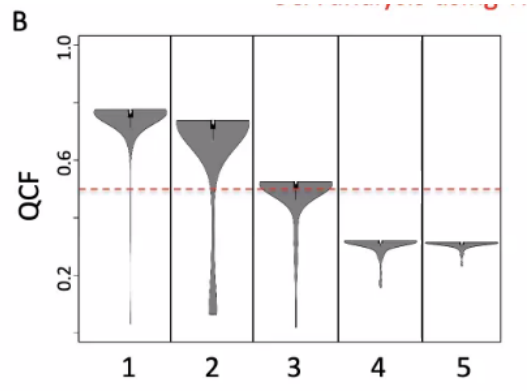


Feature matrix comes from repeating the process and check if the AE is able to reproduce the data of

I know the structure with all the genes, then we do the compression using only TF and we see the output: if the structure is similar, we should be able to reconstruct the cluster

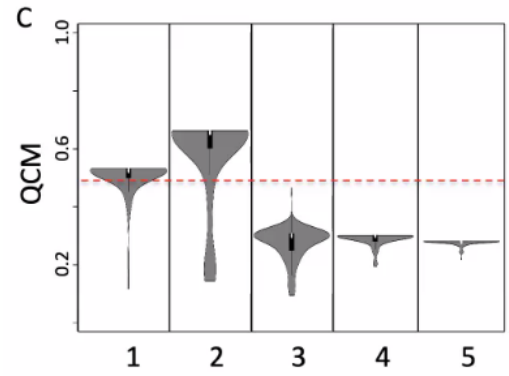
Each time we train the AE, this training starts from a different part of the data.

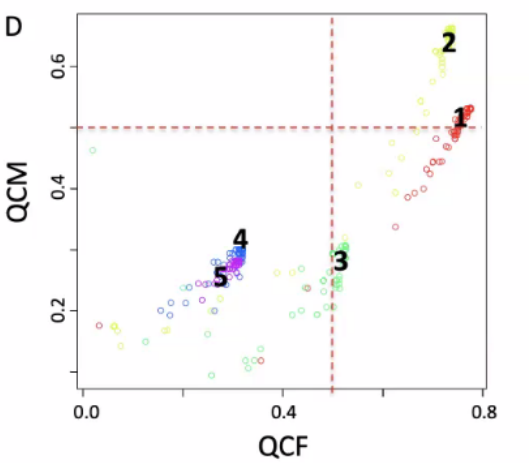
QCF measures the n of times thart during the training the clusters are reconstructed good: 1 2 and 3 borderline are reconstructed → 3 out of 5 clusters are reconstructed by this means.



The other 2 are not because the info at the level of TF is not enough to put cells together in that cluster.

Model quality is another control used: how many times in the training the reconstructed models are similar to eachother. Only if >50% trainings we get QCM and QCF > 0.5, these are clusters that can be reconstructed

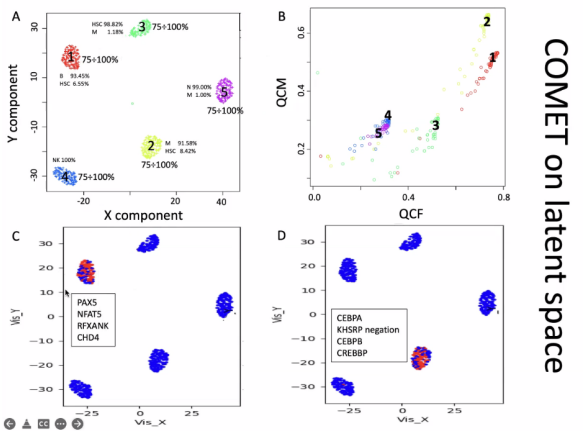




Only 2 clusters were reconstructed in the end considering both QCM and QCF.

The adv of this approach is that once data are compressed (here for cluster 1 and 2 that are passing the thresholds for both model similarity and similarity to input cluster), they can be used to run COMET allowing to depict specific genes that characterize that cluster.

Doing this compression we know that TF are driving force for cluster 1 nd 2, but with comet we can have a specific signature for each cluster.

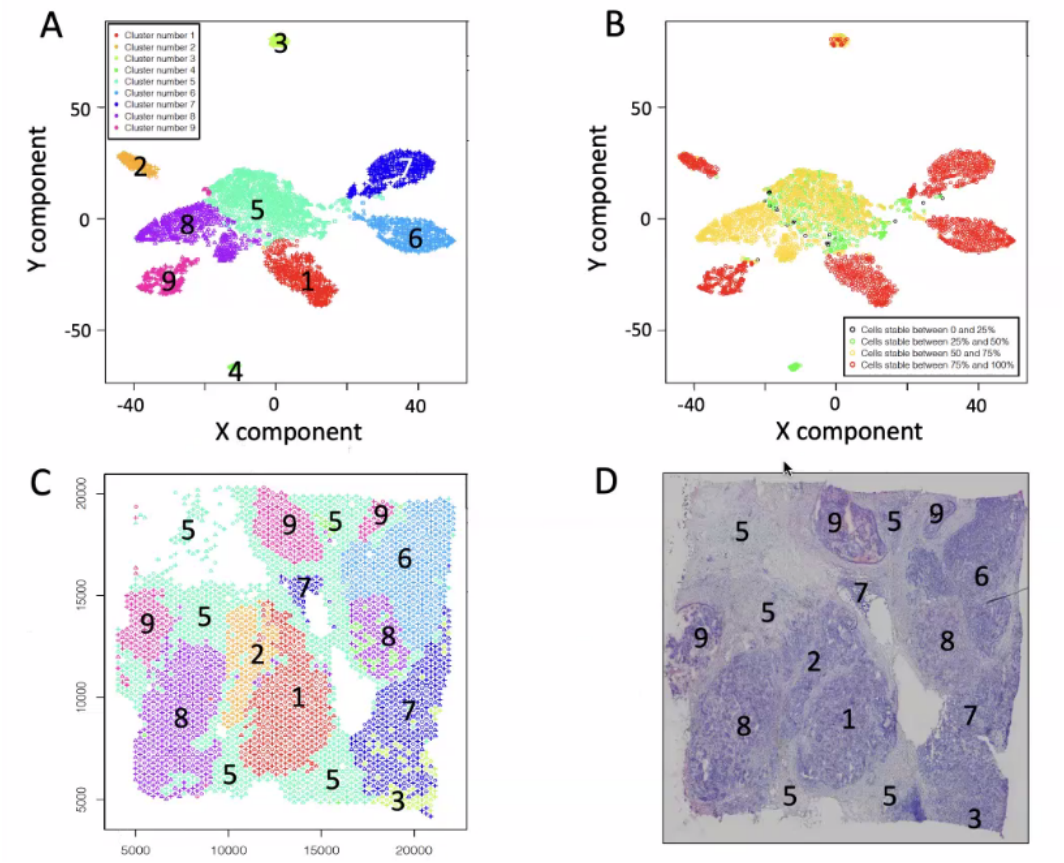


COMET on latent space

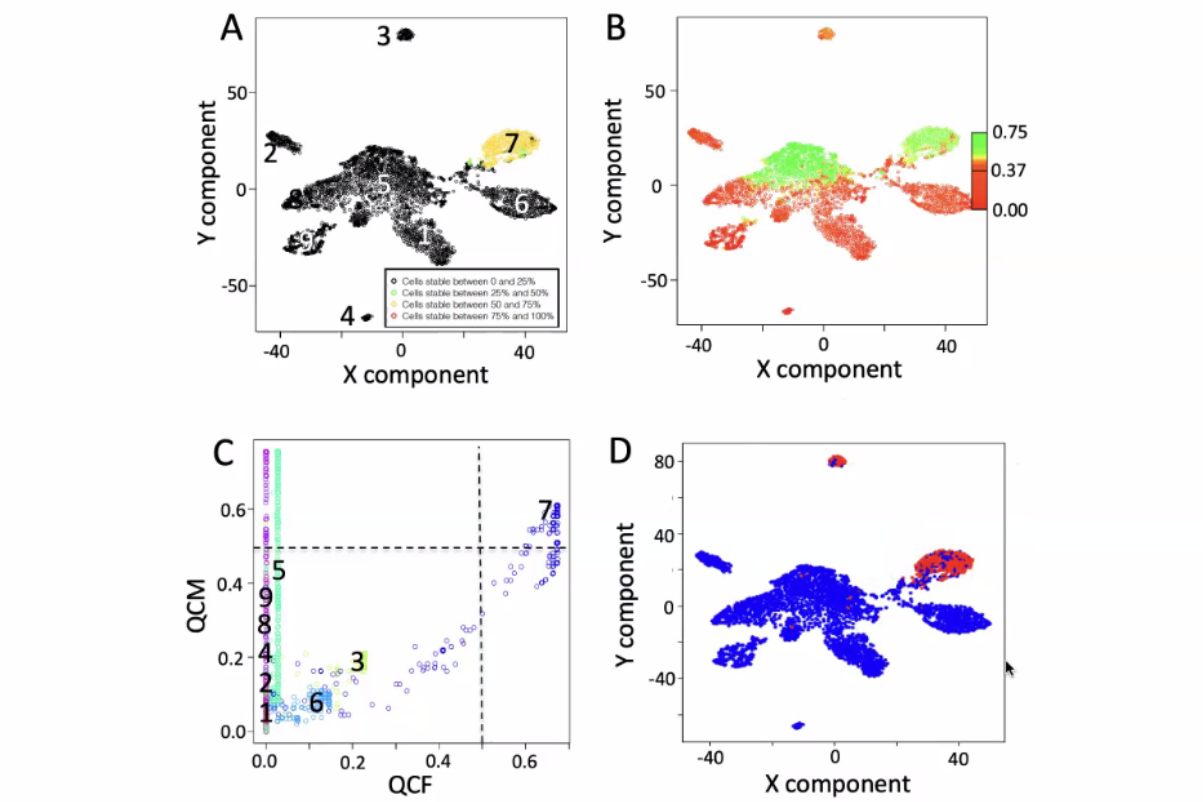
COMET

- **The four markers detected for B-cell very well recapitulate some of the main key elements of B-cells.**
  - The transcription factor PAX5 is essential for commitment of lymphoid progenitors to the B-lymphocyte lineage. PAX5 fulfils a dual role by repressing B lineage 'inappropriate' genes and simultaneously activating B lineage-specific genes.
  - NFAT5 is essential for optimal antibody productivity.
  - RFXANK is involved in activation of MHC-II genes. MHC-II molecules are largely restricted to thymic epithelial cells and professional antigen-presenting cells, including dendritic cells, macrophages, and B-cells. CHD4 is essential for early B-cell development.
- **Also, in the case of the top ranked 4 markers for cluster 2 (CEBPA, KHSRP negation, CEBPB, CREBBP, there is a very good relation with Monocyte functionalities.**
  - CEBPB is required for generation of granulocyte/monocyte progenitors. The transcription factor CCAAT/enhancer-binding protein β (CEBPB) is highly expressed in monocytes/macrophages and is a critical factor for Ly6C-monocyte survival.
  - The downregulated expression of the KH-Type Splicing Regulatory Protein (KSRP) during monocytopoiesis and up-regulated expression during granulopoiesis suggests that KSRP has divergent roles during monocytic and granulocytic differentiation.
  - CREB induces an anti-apoptotic survival signal in monocytes and macrophages
  - CREBBP specifically binds to the active phosphorylated form of CREB.

More complex dataset (tumor dataset generated using special transcriptomics picking TF profiles from a slide sample from which we can also have spatial information of the subsets of cells) were used as input. The clusters can also be superimposed to the H&E staining (bottom right picture).



All these elements unless the 5, are different subsets of the same tumors charachterized by a different transcriptional profile. This structure can be inferred using TF and miRNA. TF info only could only reconstruct cluster 7 and partially for cluster 5 for QCM, not for QCF. Taken together (C) only cluster 7 passes the threshold, because the structure of cluster 5 is not reproducible.

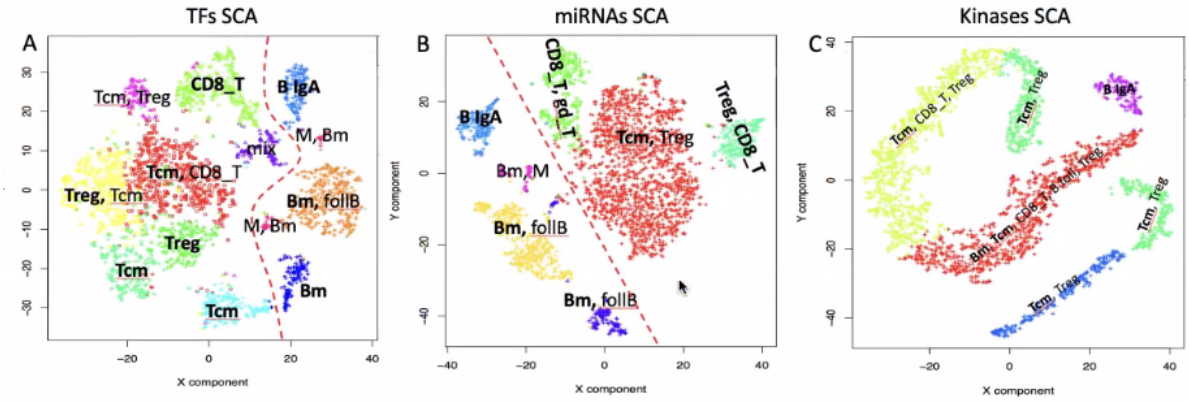


Information contents extracted using as hidden layer the TFs. A) QCF. B QCM. C) Only cluster 7 show a QCF and a QCF greater than 0.5. D) SOX5 detected as top ranked genes specific for cluster 7, using as input for COMET the latent space frequency table.

Cluster 5 is represented by stromal cells present within the tumor. They might represent different amount of infiltration that when converting into TF, there is a subset that is maintained together and a part not. The cluster done might mask some inner clusters.

Instead of making clusters, they used AE to generate a new set of data clustered on the basis of ??

They took Colon Immune Atlas (immune cells are similar and very likely to get separated from eachother). CIA is based on gut immune cells from 5 different healthy donors. B and T cells are sort of mixed in the clustering representation. If we transform the representation in TF, we get a separation among B and T cell response. They are not anymore together.



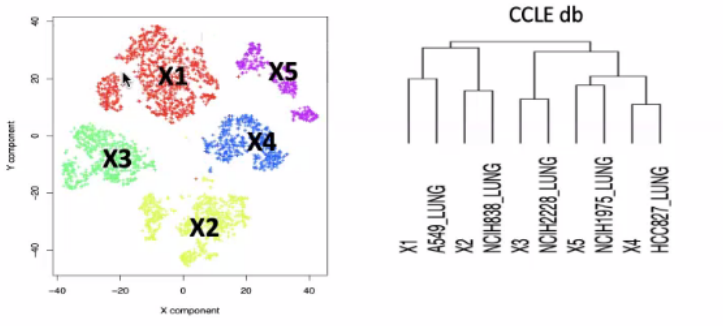
Kinases is a bit tricky and they did not get good representation → kinases are too far from transcription data to be modelled with them.

Other paper

3 adenocarcinoma cell lines and other 5 are mixed and analyzed with 10X genomics trying to understand

Epigenetic of different tumor cell lines, there will be different areas on/off in the different cell lines.

Kariotype of healthy donors will be all equal in Giemsa coloring: different areas of different darkness (more AT = darker, more GC = lighter).

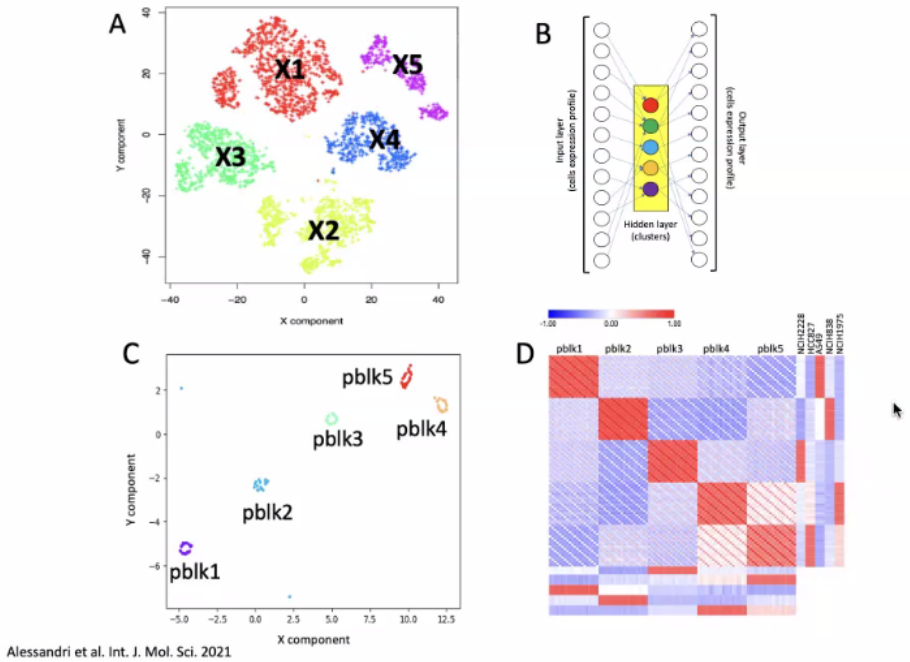


Cluster 1 and 5 have some levels of heterogeneity. Comparing sc with bulk we can assign each specific cluster to its cell line. Representing data as cytoband, picture changes slightly: we can select each block of cells and investigate the changes in chromosomes affecting that specific kind of cell.

Cytobands provide info on spatial organization of chromatin in oncological setting. IC data (long distance interactions) can be substituted to cytobands to get info on structural loops telling about areas of open or closed chromatin. We are adding chromatin structure info to the transcriptional one.

One of the criticalities of single cells is that we have many genes that are not depicted because of too low expression. If we put together all thedata we get a **pseudo bulk experiment** in which we grouyp the expression of cells from a single cluster. No replications → This can not be used to do diff expr analysis.

We can apply this sparsely connected AE to transform the expression data on pseudobylk on the basis of their belonging to a cluyster. We can repeat the analysis many times and obtain **pseudoreplicates** that resemble a std bulk RNA seq on which we can do std diff expr analysis.



Pseudobulk correlate much with real replicates coming from public databases.

If the model works too smoothly, the samples are too much similar and diff expr will have too low noise and have some low-noise-derived artifacts.

Comparing the experiment on 3 to the 5 cell lines, we see that 2 indep experiments

Pearson correlation similarity can associate cluster 3 of the 5cell lines pseudobulk to the cluster 3 of the one done on 3 cell lines.

Each cell type can be assigned using specific ab on single cell(?) min 1:29:29

Conclusions

sparsely connected AE allow us to predict hidden relationships among the genes. These rel can be of different nature (same localization, TF-genes, kinases-targets etc)

They can extract some functional networks peculiar of the clusters we are looking at