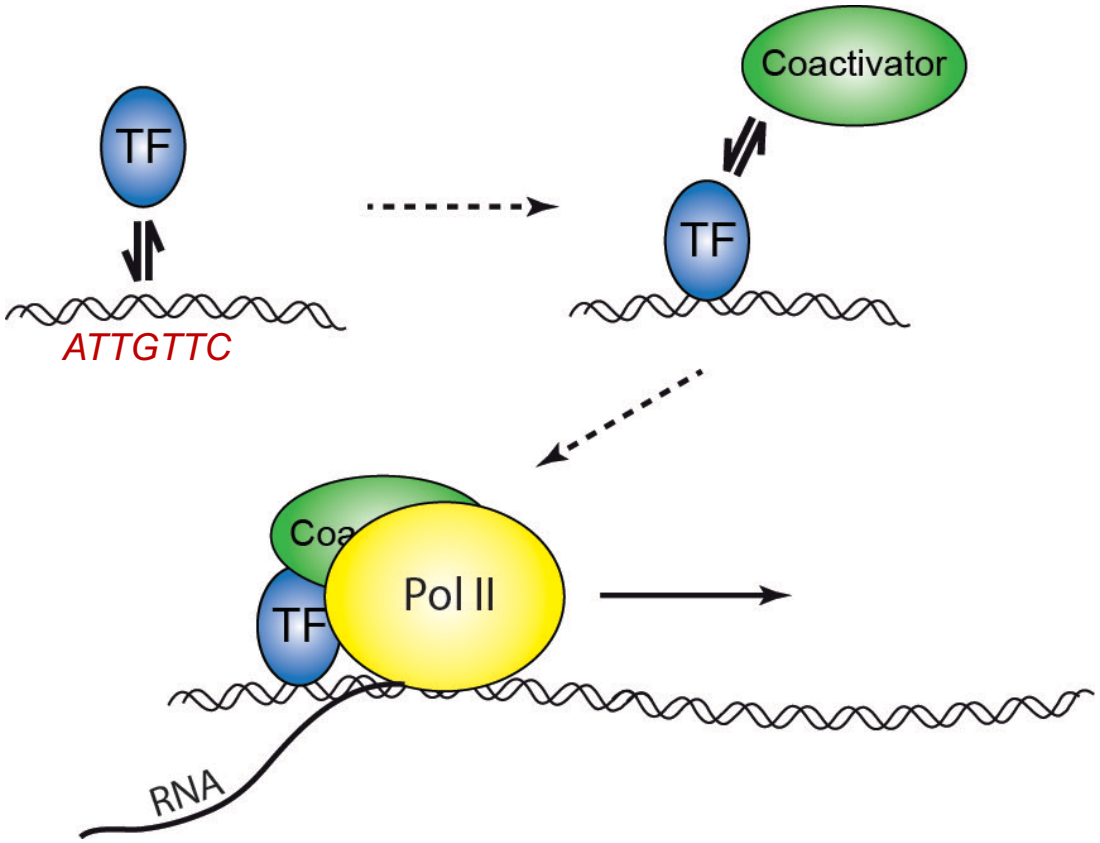
Transcription Factor Analysis with pySCENIC

Single Cell Transcriptomics in Python

Alex Lederer

What is a transcription factor?

- A protein that binds to regulatory elements at specific DNA sequence motifs to activate transcription of a set of downstream genes.
- Important for coordinated regulation of related genes for a common biological program.

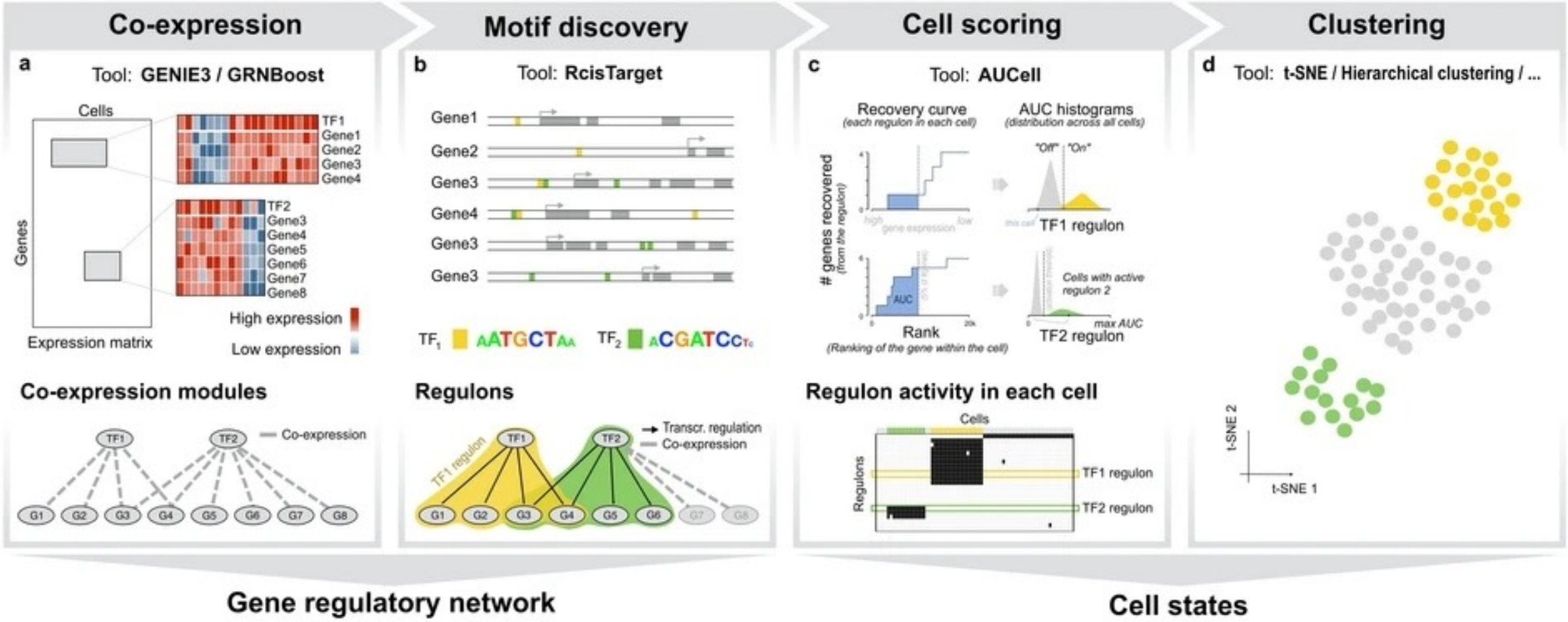


Can we elucidate which TF regulatory programs are active in a cell type or cluster using co-expression of TFs and their target genes?

From: https://www.uni-ulm.de/nawi/nawi-biophys/forschung/forschung-gebhardt/in-cellula/

SCENIC: Single-Cell rEgulatory Network Inference and Clustering

https://scenic.aertslab.org/; pySCENIC: https://pyscenic.readthedocs.io/en/latest/tutorial.html



Aibar et al. Nat Methods (2017)
https://doi.org/10.1038/nmeth.4463

STEP 1: Gene regulatory network inference, and generation of co-expression modules

GEne Network Inference with Ensemble of trees

Input:

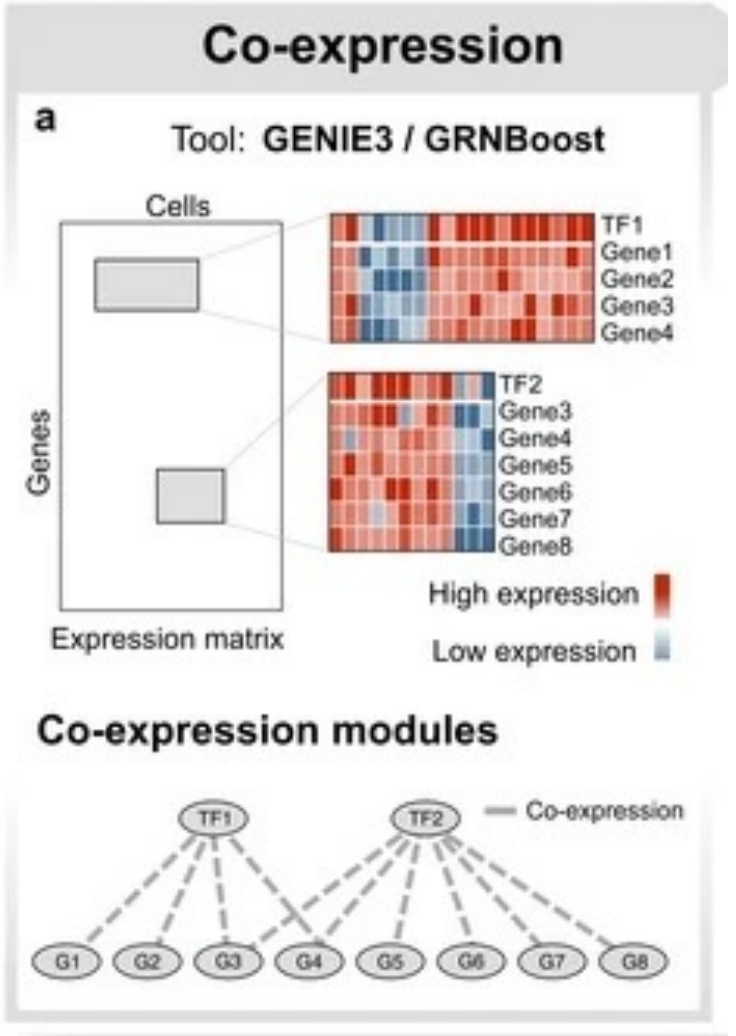
- scRNA-seq data (after processing with scanpy)
- A text file (allTFs_hg38.txt) containing gene symbols for all transcription factors (TFs)

Output:

- An adjacency table of TF and targets that are potentially in a shared gene regulatory network
- *Only based on co-expression* of the TF and target gene; may include false positives or indirect targets

```
adjacencies.head()
```

	TF	target	importance
0	CEBPD	VCAN	33.587173
1	ZEB2	LTB	33.086506
2	KLF4	VCAN	29.961844
3	CEBPD	SRGN	29.306472
4	MEF2C	HLA-DRA	28.519784



https://github.com/aertslab/GENIE3

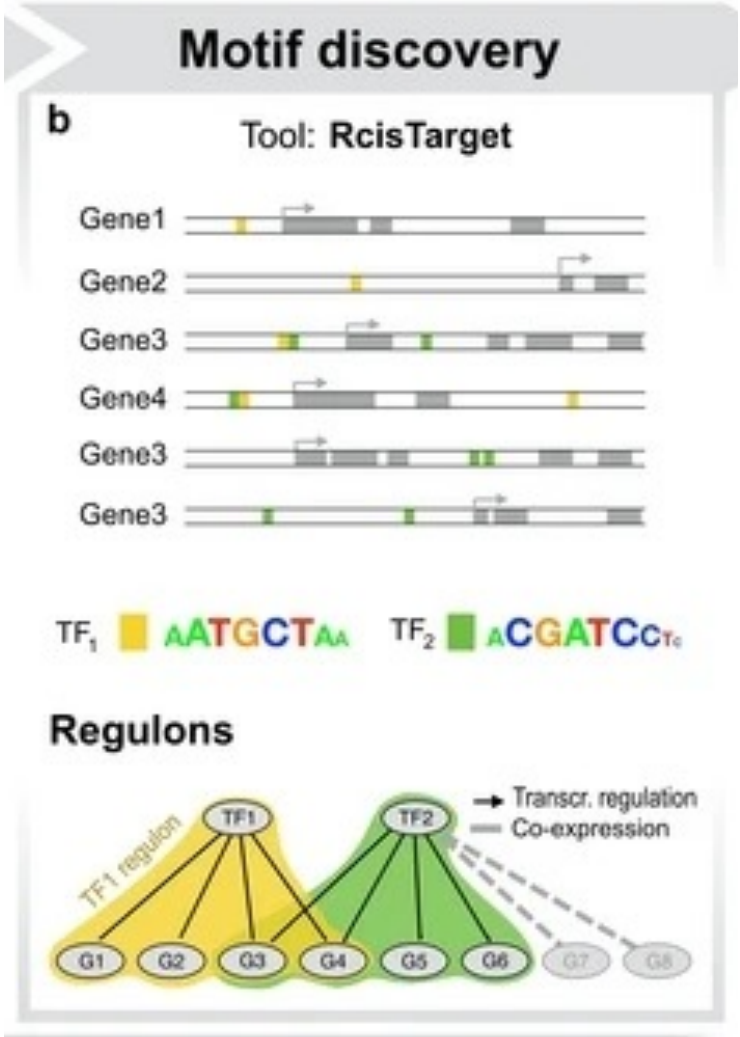
STEP 2: Regulon prediction aka cisTarget

Input:

- Adjacency table
- List of transcription start site (TSS) annotations (hg38__refseq-r80__10kb_up_and_down_tss.mc9nr.genes_vs_motifs.rankings.feather)

Output:

- An adjacency table of TF and target gene pairs with appropriate TF binding motifs at the TSS of the target gene.
- A single TF and its target genes is defined as a **regulon.**



https://github.com/aertslab/RcisTarget

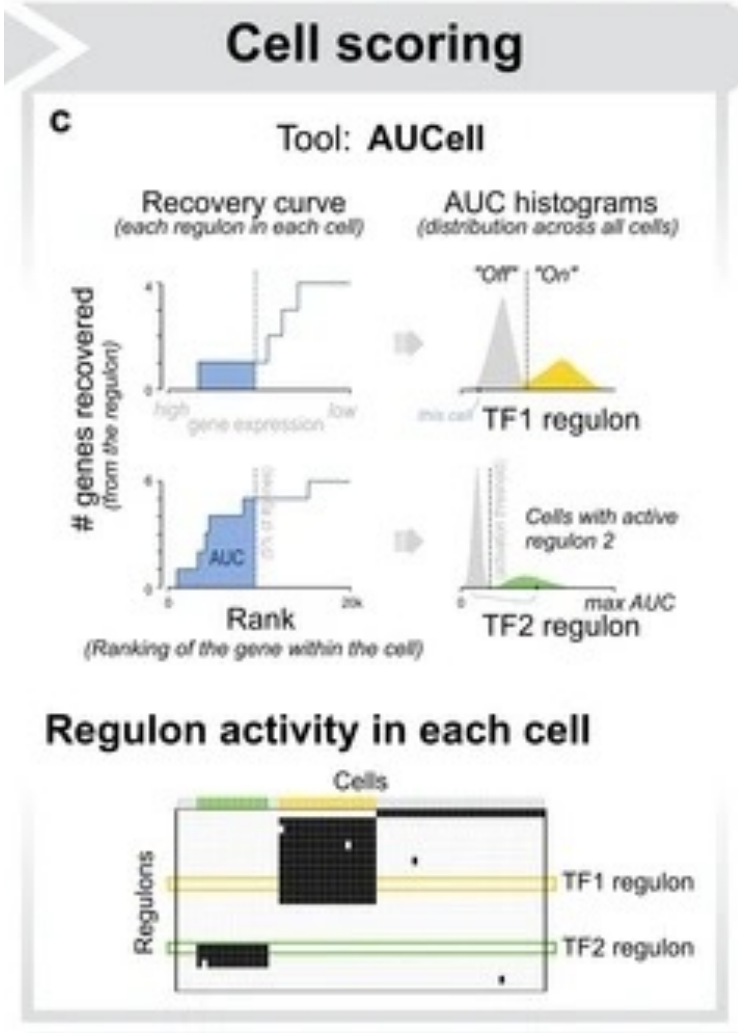
STEP 3: Cellular enrichment (aka AUCell)

Input:

- List of TF regulons

Output:

- A data matrix of regulon scores for all single cells in the dataset.
- AUC matrix is of shape = n_cells x n_regulons



https://github.com/aertslab/RcisTarget

STEP 3: Cellular enrichment (aka AUCell)

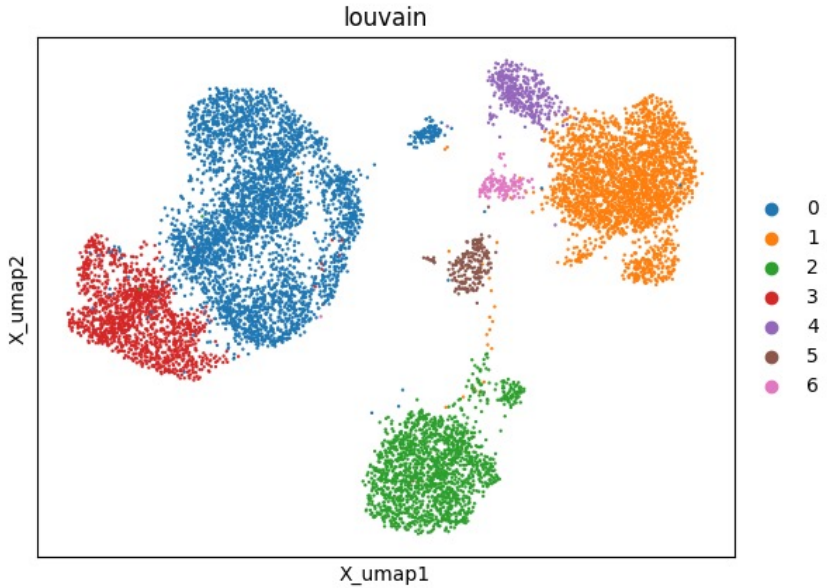
Input:

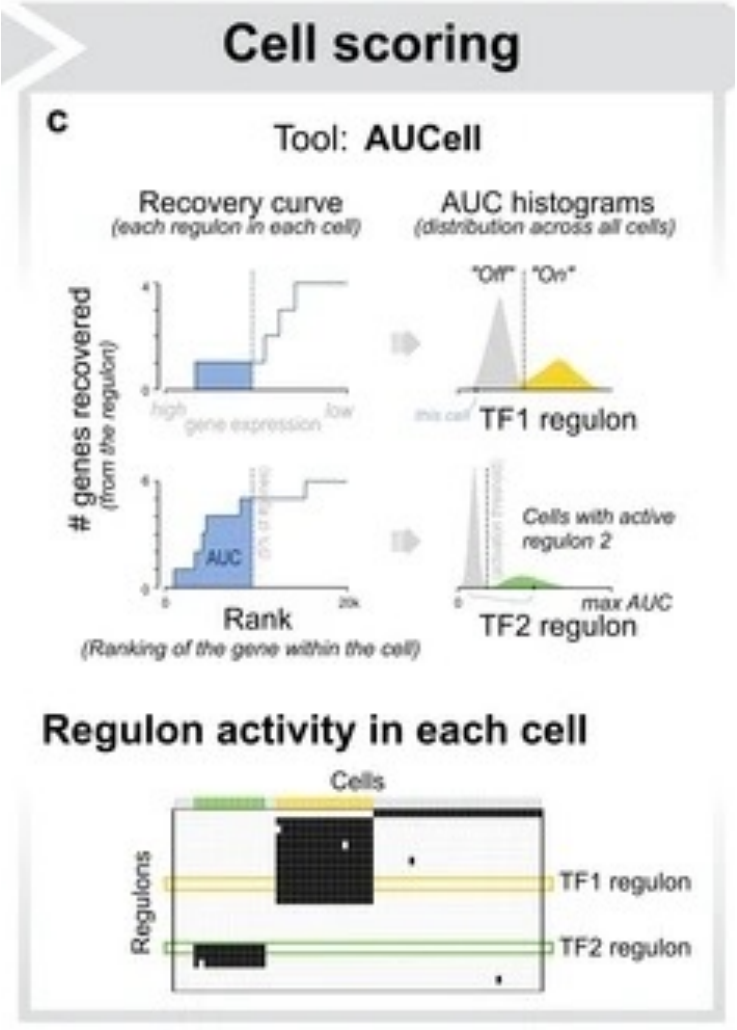
- List of TF regulons

Output:

- A data matrix of regulon scores for all single cells in the dataset.
- AUC matrix is of shape = n_cells x n_regulons

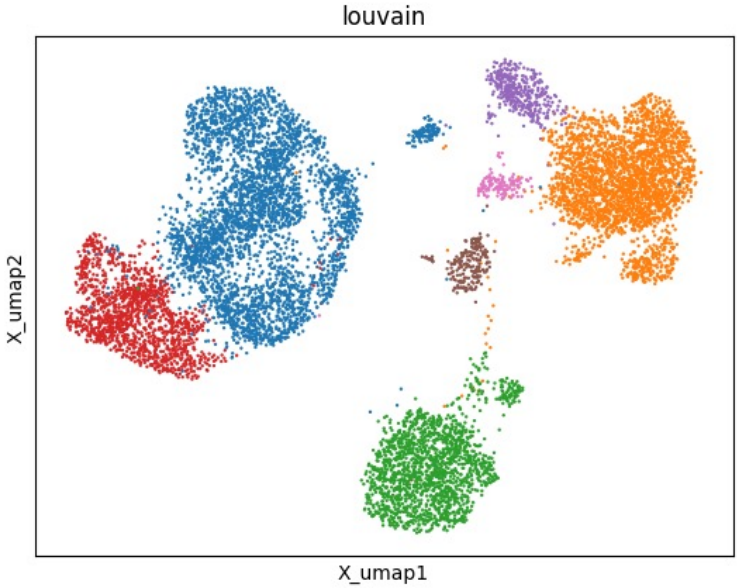
A UMAP can be computed using the AUC matrix instead of the counts data

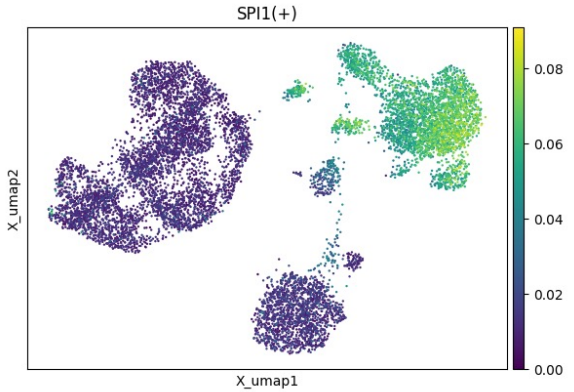


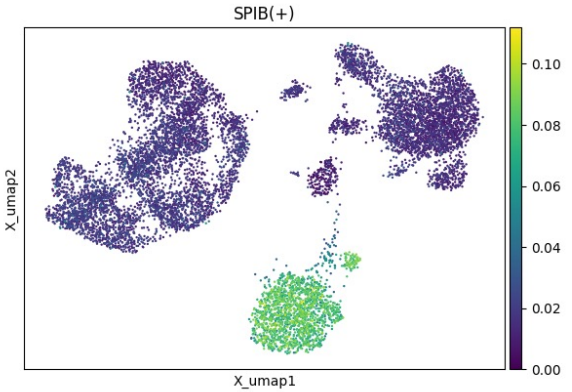


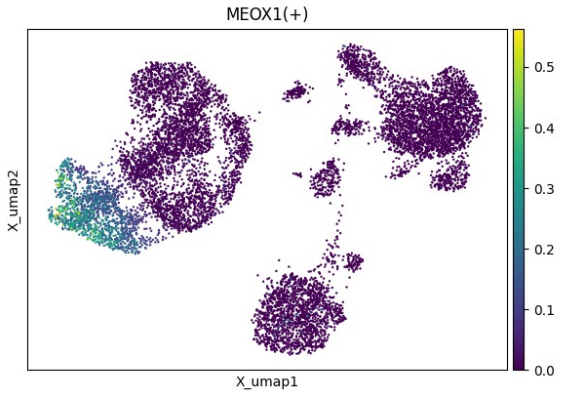
Visualization of “marker regulons” for cell clusters

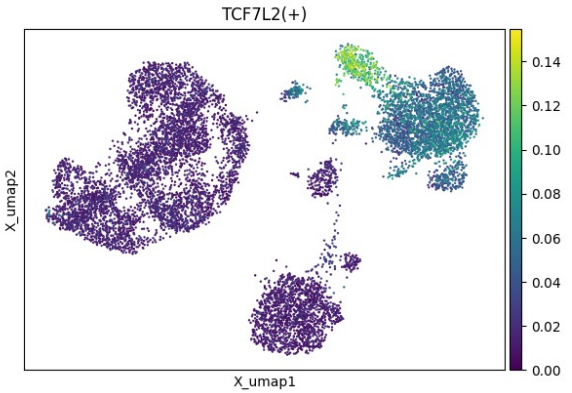
A UMAP can be computed using the AUC matrix instead of the counts data












```
marker_genes = pd.DataFrame(ad_auc_mtx.uns["rank_genes_groups"]["names"])
marker_genes.columns = ["Cluster" + str(x) for x in range(0, len(ad_auc_mtx.obs["louva
marker_genes.head()
```

	Cluster0	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
0	PRDM1(+)	SPI1(+)	SPIB(+)	MEOX1(+)	TCF7L2(+)	TAL1(+)	CEBPB(+)
1	EOMES(+)	NFE2(+)	BCL11A(+)	LEF1(+)	SPI1(+)	GATA1(+)	SPI1(+)
2	TBX21(+)	CEBPD(+)	PAX5(+)	TCF7(+)	CEBPB(+)	MAFG(+)	NFE2(+)
3	RUNX3(+)	CEBPB(+)	IRF8(+)	KLF12(+)	MAFB(+)	NFE2(+)	GATA3(+)
4	KLF12(+)	MAFB(+)	IRF4(+)	BACH2(+)	E2F1(+)	E2F1(+)	CEBPD(+)