

Single cell analysis of conventional chondrosarcomas

Léa ROGUE

28-03-2025

This script analyze single cell RNA-seq data (GSE184118) by selecting conventional chondrosarcomas. This permit us to confirm the presence of CHM type 1 on tumoral cell surfaces because at least it as necessary in the project

Table of contents

- Single cell analysis of conventional chondrosarcomas
 - Load packages
 - Process data
 - I. Integration
 - II. Expression of interest CTAs
 - 1) CTAs that impact survival
 - a- Cluster 1
 - b- Cluster 2
 - c- Cluster 3
 - 2) CTAs a bit expressed in normal tissues but expressed in chondrosarcoma
 - 3) CTAs of interest
 - III. Expression of genes per patients
 - 1) Colored by patients
 - 2) MHC 1 genes by samples
 - 3) CTAs that impact survival
 - a- Cluster 1
 - b- Cluster 2
 - c- Cluster 3
 - 4) Low expressed CTAs in normal tissues and expressed in chondrosarcoma
 - 5) Selected CTA of interest

Load packages

```
In [1]: import scanpy as sc  
import matplotlib as mpl  
import matplotlib.pyplot as plt
```

```
import anndata as ad
sc.settings.verbosity = 0
```

```
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_csv from `anndata` is deprecated.
Import anndata.io.read_csv instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_excel from `anndata` is deprecate
d. Import anndata.io.read_excel instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_hdf from `anndata` is deprecated.
Import anndata.io.read_hdf instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_loom from `anndata` is deprecated.
Import anndata.io.read_loom instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_mtx from `anndata` is deprecated.
Import anndata.io.read_mtx instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_text from `anndata` is deprecated.
Import anndata.io.read_text instead.
    warnings.warn(msg, FutureWarning)
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.1
1_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\ut
ils.py:429: FutureWarning: Importing read_umi_tools from `anndata` is deprec
ated. Import anndata.io.read_umi_tools instead.
    warnings.warn(msg, FutureWarning)
```

Process data

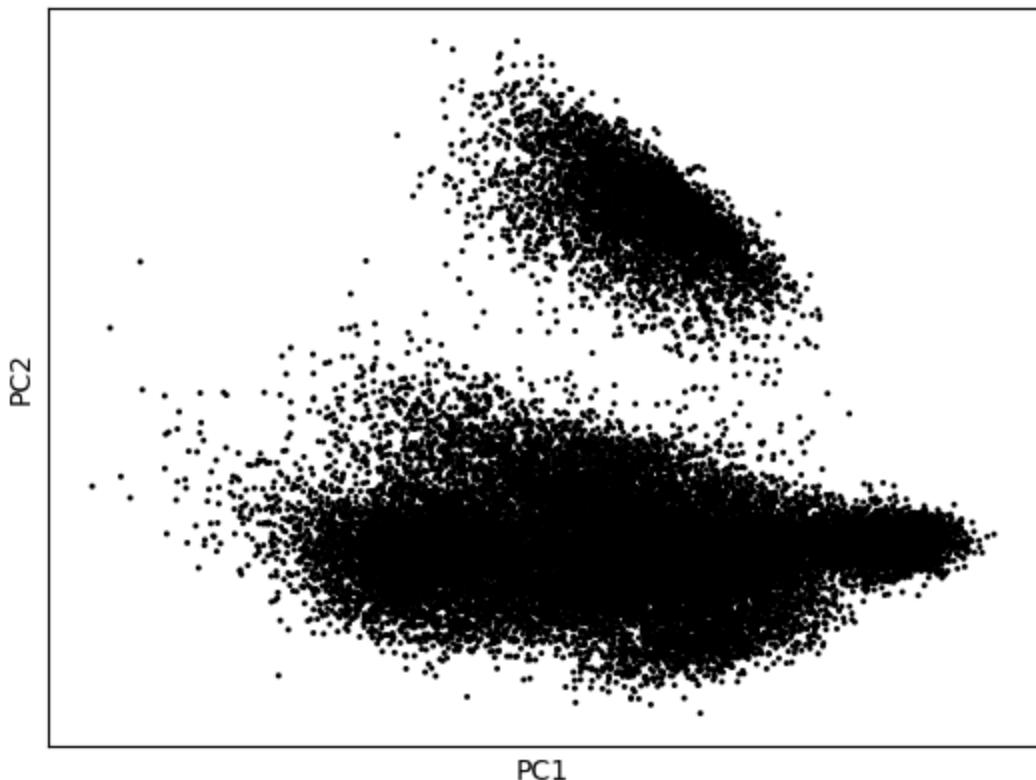
The object has been processed with a tool created during my M1 intership that we have adapted for this project. The samples were filtered with 1000 min genes per cell, 3 cells per genes and 5% max of mitochondrial genes. After, these sample filtered were merged.

```
In [2]: # Read data
adata = ad.read_h5ad('../results/sc_results/chondro_conv_ben_analysis/Object
adata
```

```
C:\Users\learogue\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\anndata\core\anndata.py:1756: UserWarning: Observation names are not unique. To make them unique, call `obs_names_make_unique`.  
    utils.warn_names_duplicates("obs")
```

```
Out[2]: AnnData object with n_obs × n_vars = 30434 × 23175  
        obs: 'n_genes_by_counts', 'total_counts', 'total_counts_MT', 'pct_counts_MT', 'dataset', 'n_genes'
```

```
In [3]: # Normalize the data  
sc.pp.normalize_total(adata, target_sum = 1e4)  
  
# Log-transform the data  
sc.pp.log1p(adata)  
  
# Store the raw data  
adata.raw = adata  
  
# Scale the data  
sc.pp.scale(adata)  
  
# Run PCA  
sc.tl.pca(adata)  
  
# Plot PCA  
sc.pl.pca(adata, size = 18, na_color = 'black')
```



```
In [4]: resolution = 0.1 # resolution for clustering, more high = more clusters, less neighbors = 15 # number of neighbors for compute the neighborhood graph n_pcs = 20 # number of principal components analysis
```

```

# Compute the neighborhood graph
sc.pp.neighbors(adata, n_neighbors = n_neighbors)

# Cluster the cells using the leiden algorithm
sc.tl.leiden(adata, resolution = resolution)

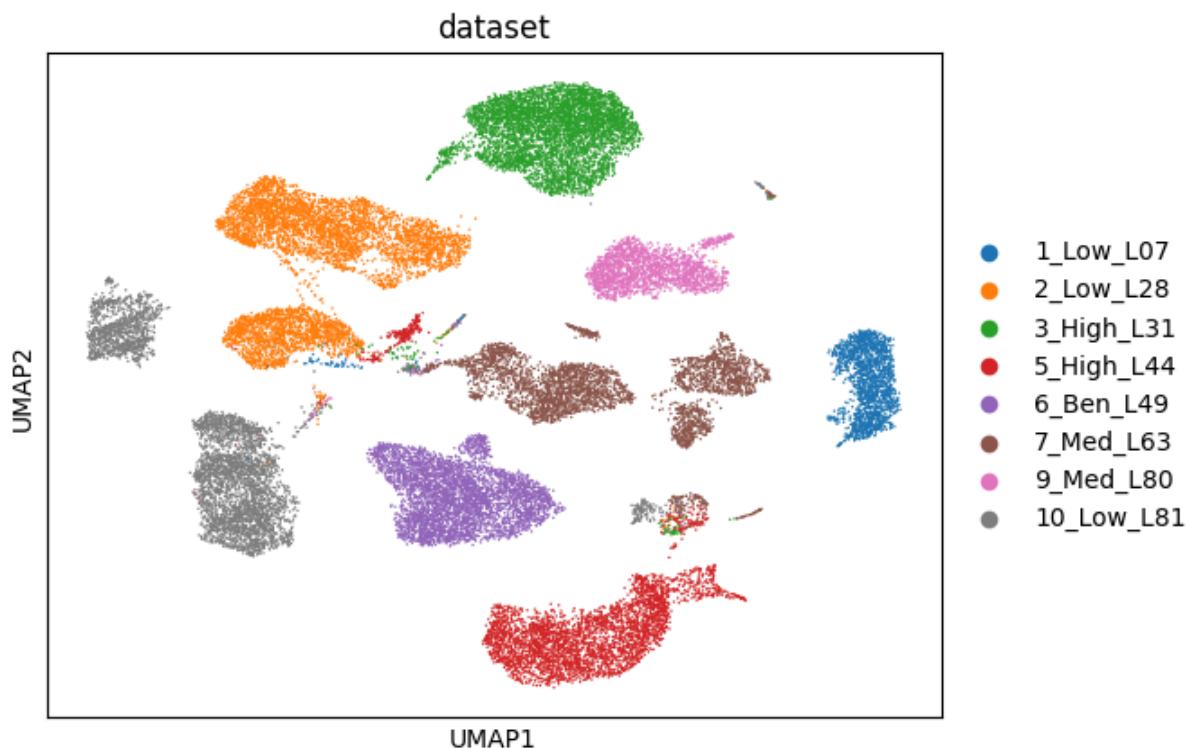
# Compute UMAP
sc.tl.umap(adata)

# Plot the UMAP datasets Louvain plot
sc.pl.umap(adata, color = 'dataset')

```

C:\Users\learogue\AppData\Local\Temp\ipykernel_5696\2542342129.py:9: FutureWarning: In the future, the default backend for leiden will be igraph instead of leidenalg.

To achieve the future defaults please pass: flavor="igraph" and n_iteration_s=2. directed must also be False to work with igraph's implementation.
sc.tl.leiden(adata, resolution = resolution)



Here, we see that samples are not mixed because of batch effects, so we use harmony to integrate data and delete batch effects.

I. Integration

```

In [5]: # Integrate data using Harmony
sc.external.pp.harmony_integrate(adata, 'dataset', theta = 2.5, max_iter_har

# Save the Harmony-corrected PCA results
adata.obsm['X_pca'] = adata.obsm['X_pca_harmony']

```

```
2025-04-14 11:08:17,648 - harmonypy - INFO - Computing initial centroids with sklearn.KMeans...
2025-04-14 11:08:20,399 - harmonypy - INFO - sklearn.KMeans initialization complete.
2025-04-14 11:08:20,563 - harmonypy - INFO - Iteration 1 of 10
2025-04-14 11:08:30,943 - harmonypy - INFO - Iteration 2 of 10
2025-04-14 11:08:41,273 - harmonypy - INFO - Iteration 3 of 10
2025-04-14 11:08:52,192 - harmonypy - INFO - Iteration 4 of 10
2025-04-14 11:09:02,742 - harmonypy - INFO - Iteration 5 of 10
2025-04-14 11:09:13,243 - harmonypy - INFO - Iteration 6 of 10
2025-04-14 11:09:22,010 - harmonypy - INFO - Iteration 7 of 10
2025-04-14 11:09:29,803 - harmonypy - INFO - Converged after 7 iterations
```

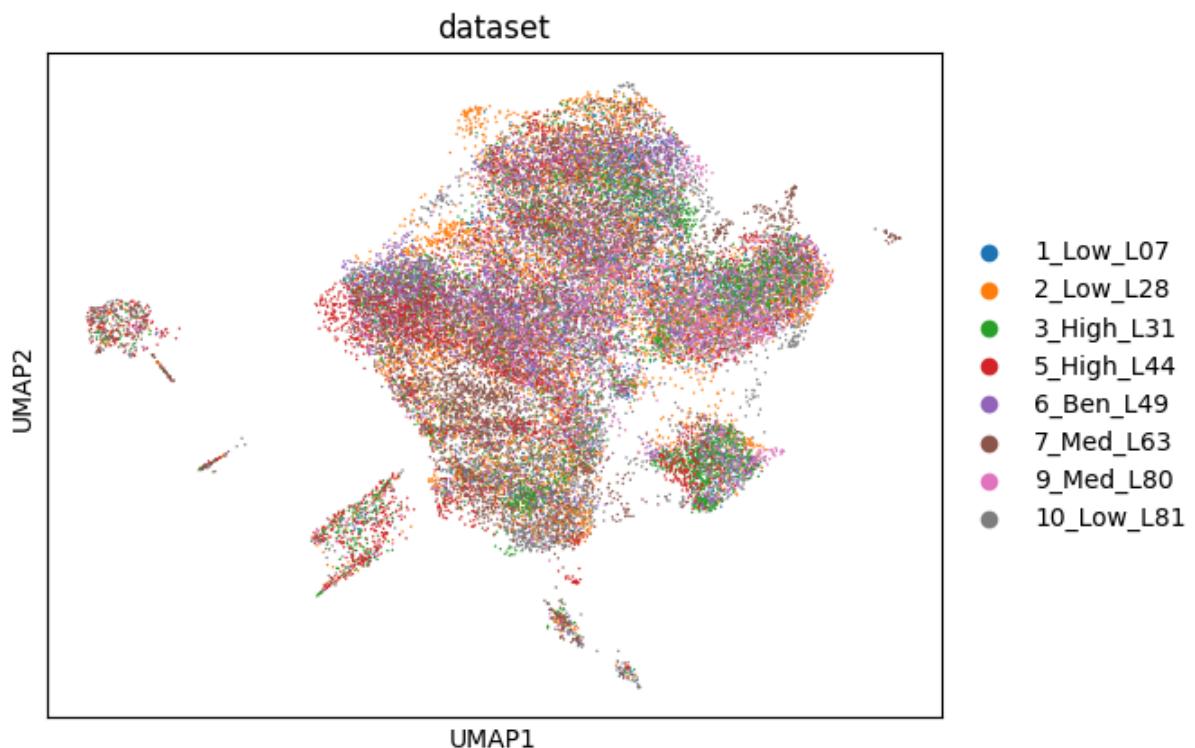
```
In [6]: resolution = 0.3 # resolution for clustering, more high = more clusters, lower = less
n_neighbors = 15 # number of neighbors for compute the neighborhood graph,
n_pcs = 20 # number of principal components analysis

# Compute the neighborhood graph
sc.pp.neighbors(adata, n_neighbors = n_neighbors)

# Cluster the cells using the leiden algorithm
sc.tl.leiden(adata, resolution = resolution)

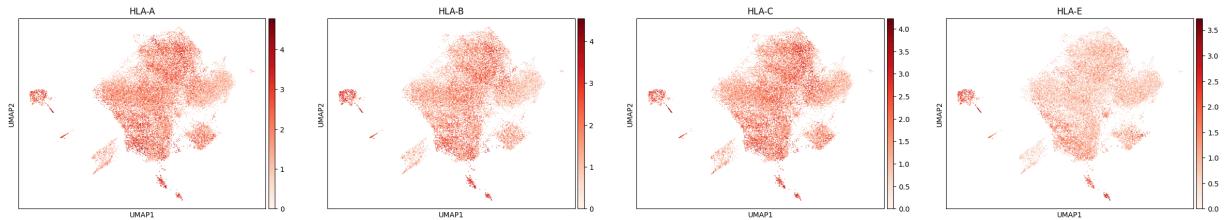
# Compute UMAP
sc.tl.umap(adata)

# Plot the UMAP datasets
sc.pl.umap(adata, color = 'dataset')
```



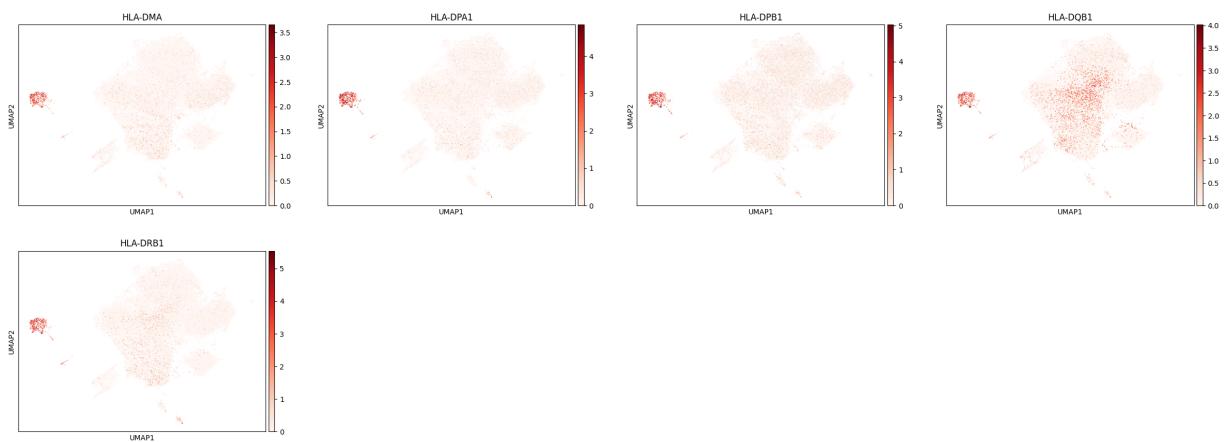
We see that all samples are now mixed so now we can determine cell types of each groups and genes of interest.

```
In [7]: # MHC I
sc.pl.umap(adata, color = ['HLA-A', 'HLA-B', 'HLA-C', 'HLA-E'], legend_loc =
```



These UMAPs show that all cells expressed HLA type 1 genes. So we can deduce that cancer cells expressed HLA type 1 and potentially present peptides.

```
In [8]: # MHC II
sc.pl.umap(adata, color = ['HLA-DMA', 'HLA-DPA1', 'HLA-DPB1', 'HLA-DQB1', 'H
```

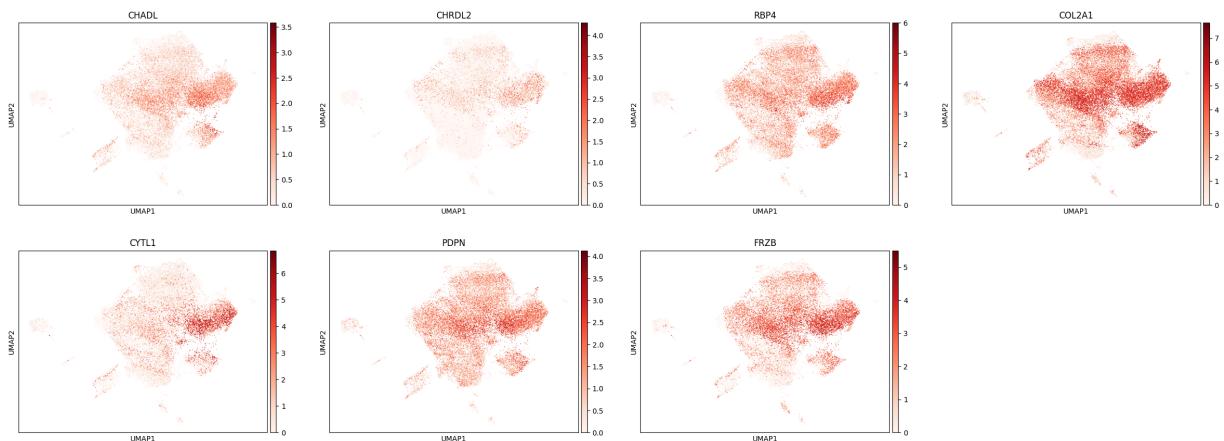


This shows the expression of HLA type 2 and it expressed in the left cluster, so we can suppose that this cluster contains immune cells. The next genes are from the data article, they use these genes to determine their clusters.

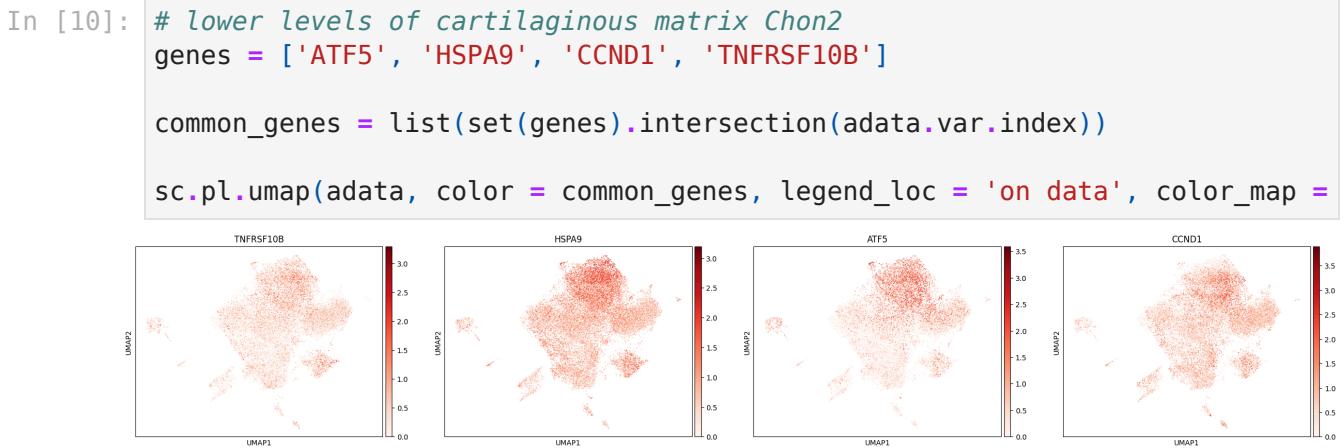
```
In [9]: # Suggesting represents well differentiated neoplastic chondrocytes Chon1
genes = ['COL2A1', 'PDPN', 'CHADL', 'CYTL1', 'FRZB', 'CHRDL2', 'RBP4']

common_genes = list(set(genes).intersection(adata.var.index))

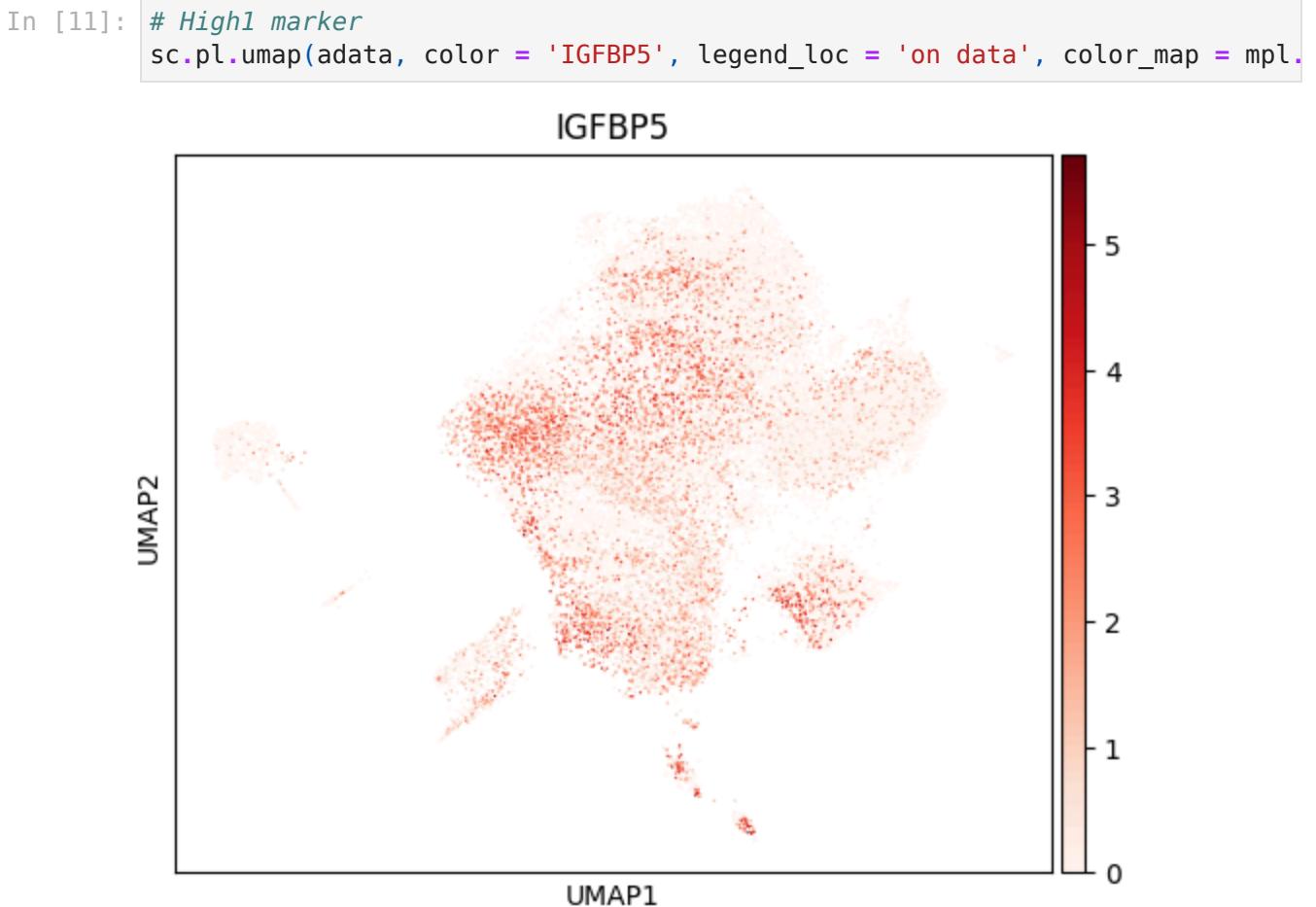
sc.pl.umap(adata, color = common_genes, legend_loc = 'on data', color_map =
```



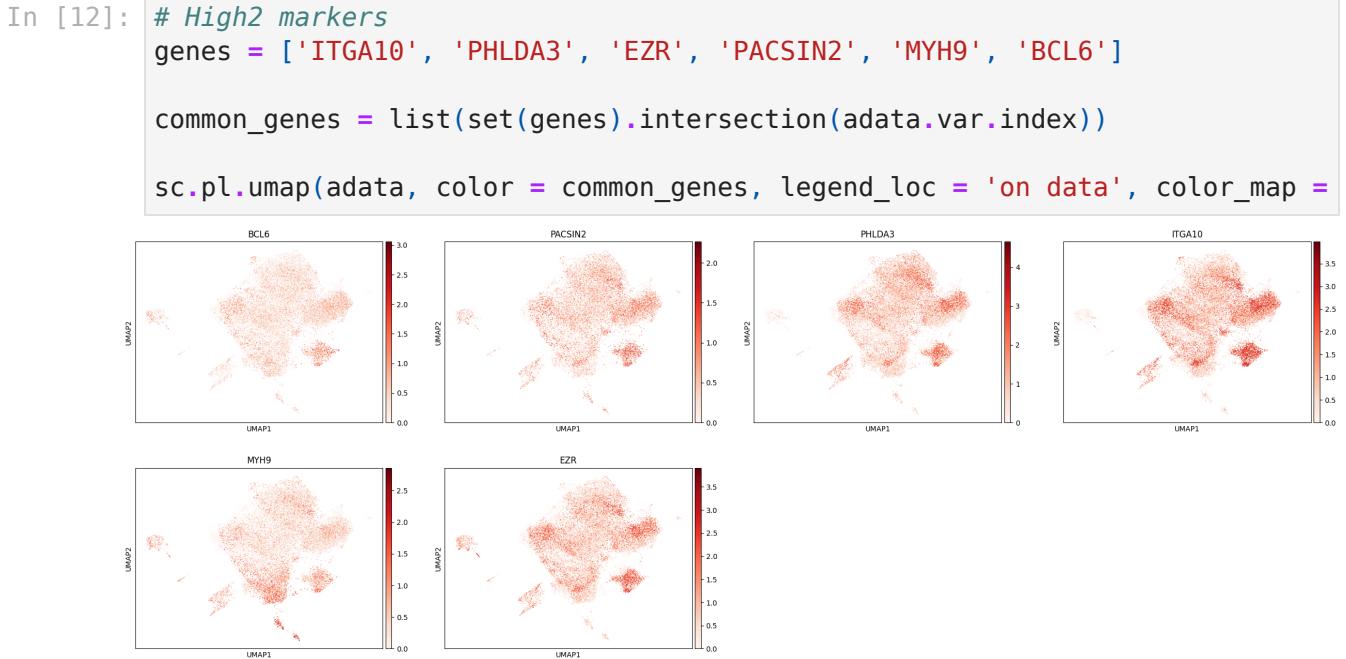
So, we see that chondrocytes are present in the big cluster, more in the middle and right.



With these markers, we can deduce that there is also chondrocytes in the big cluster on the top.



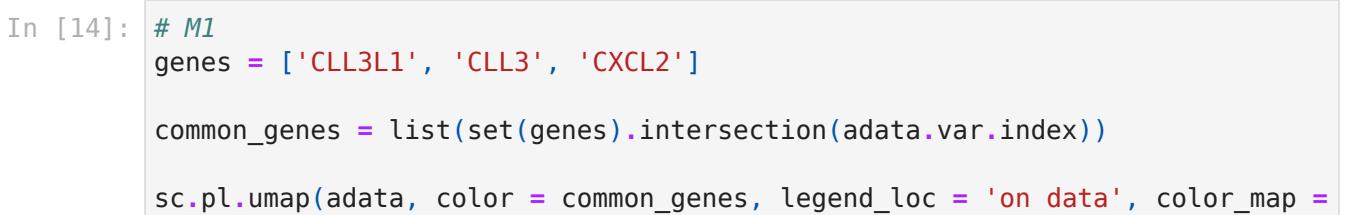
High1 markers are not very clear except IGFBP5, we see a diffuse expression, maybe more in the middle.

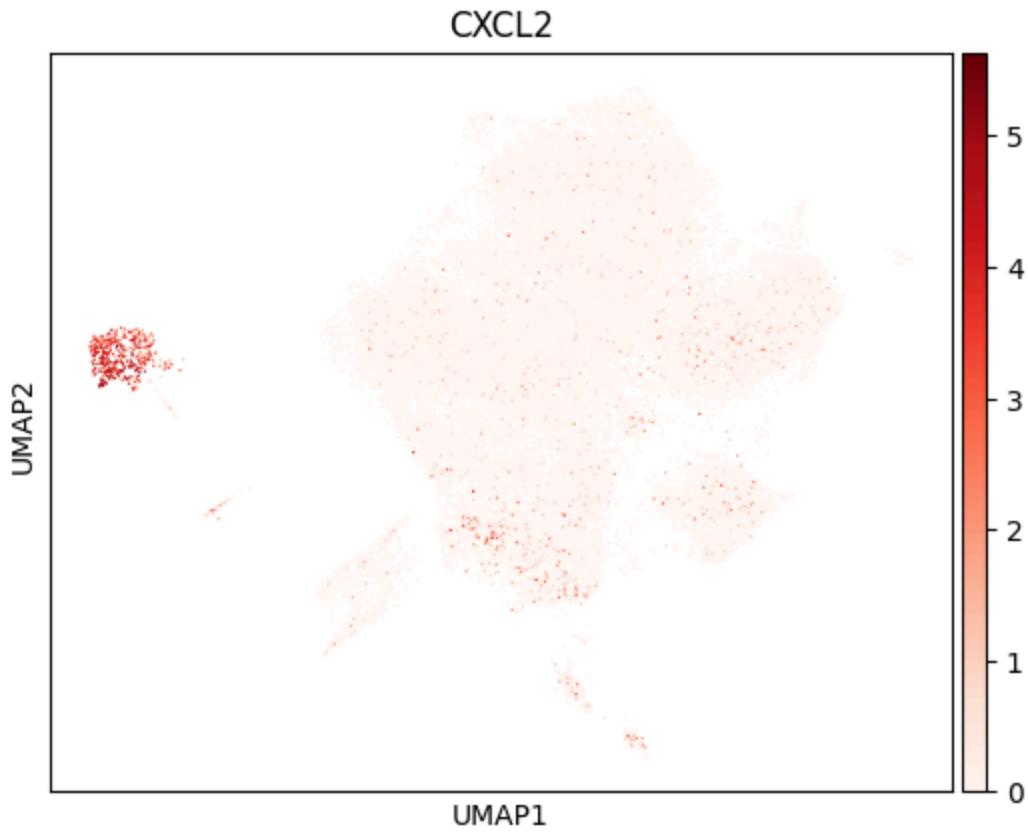


It is not very clear also, we can deduce that cluster on the bottom right and on the top of it correspond to High2 markers so may be contains tumoral cells.



Here, we can confirm that the lef cluster is immune cells.

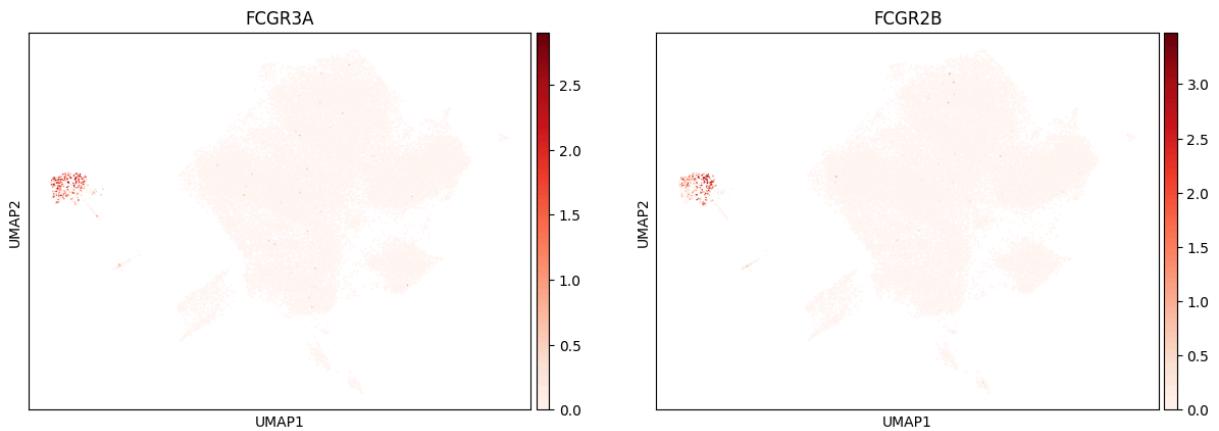




```
In [15]: # M2
genes = ['FCGR2B', 'FCGR3A']

common_genes = list(set(genes).intersection(adata.var.index))

sc.pl.umap(adata, color = common_genes, legend_loc = 'on data', color_map =
```

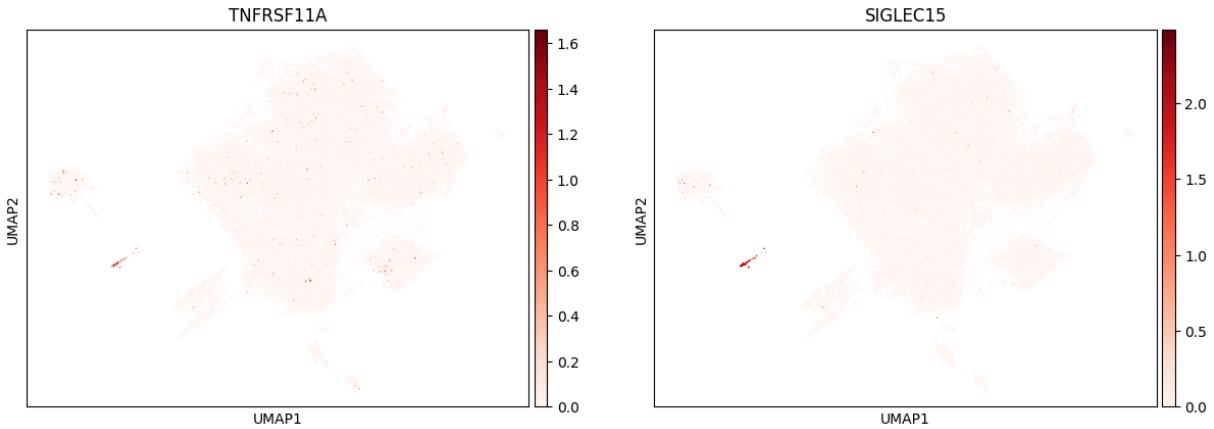


We see that M1 and M2 are mixed in the immune cells cluster.

```
In [16]: # Osteoclasts
genes = ['TNFRSF11A', 'SIGLEC15']

common_genes = list(set(genes).intersection(adata.var.index))

sc.pl.umap(adata, color = common_genes, legend_loc = 'on data', color_map =
```

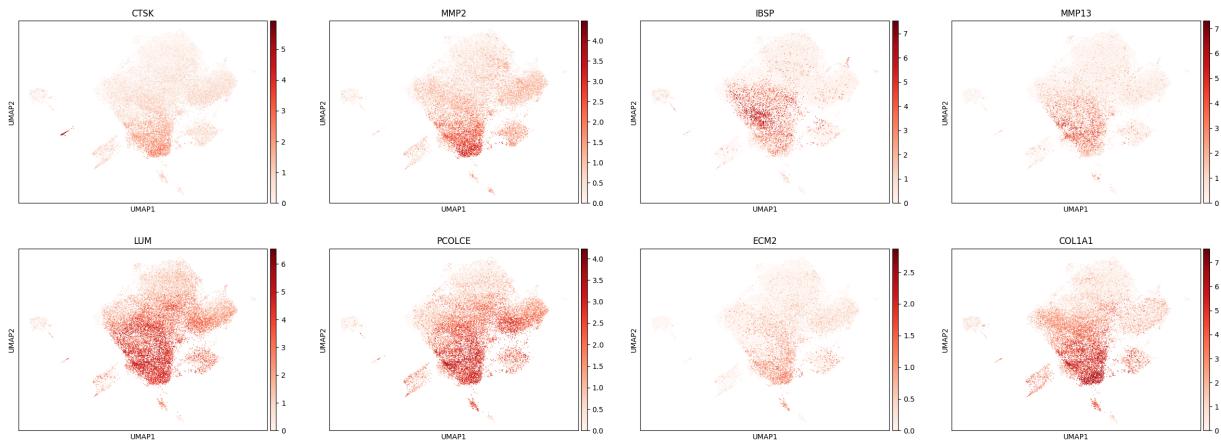


Finally the little cluster like a line is osteoclasts that are cells which degrade bones.

```
In [17]: # Stromal markers
genes = ['COL1A1', 'LUM', 'PCOLCE', 'MMP13', 'CTSK', 'MMP2', 'ECM2', 'IBSP']

common_genes = list(set(genes).intersection(adata.var.index))

sc.pl.umap(adata, color = common_genes, legend_loc = 'on data', color_map =
```

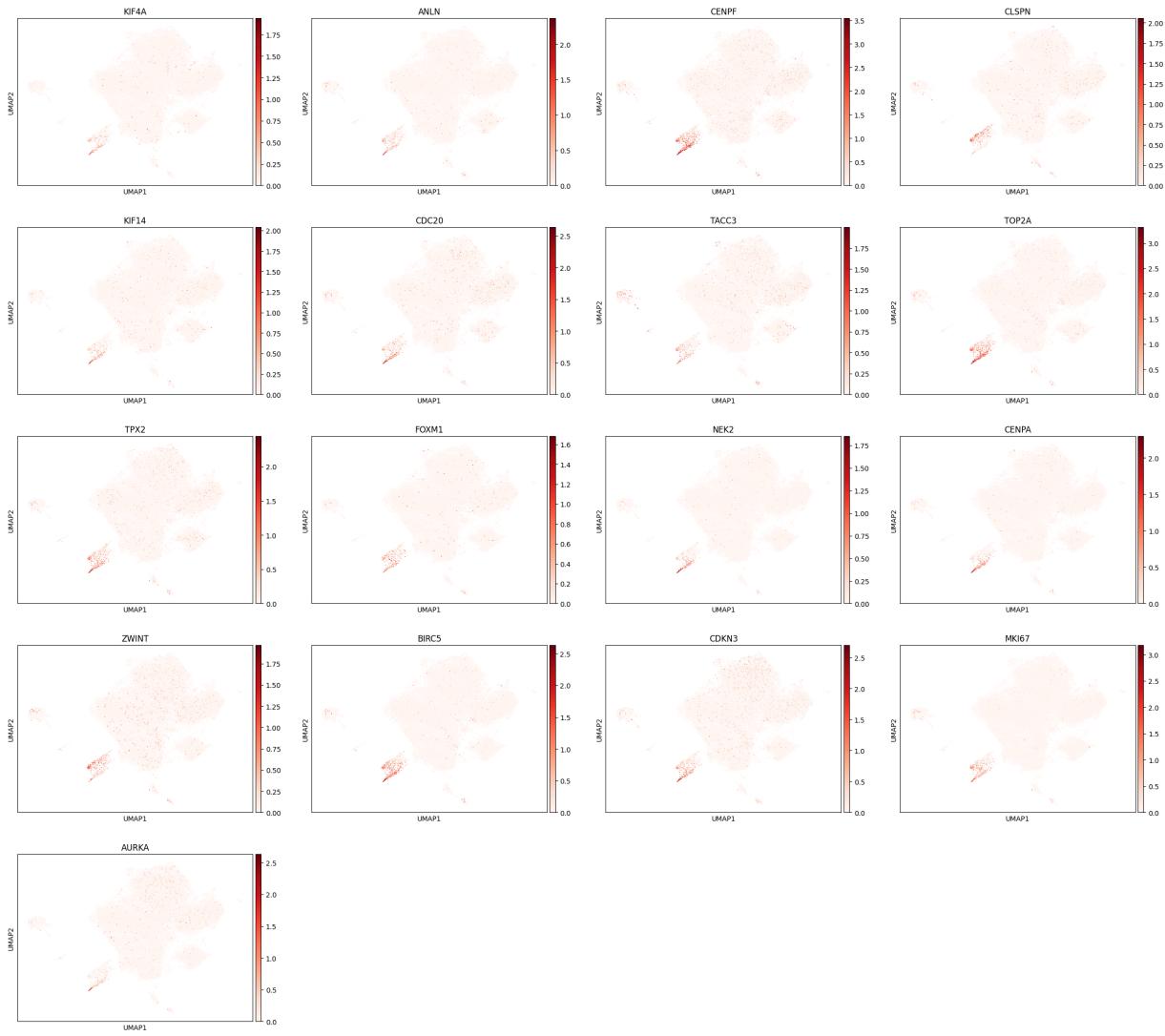


Here, we can deduce that the middle bottom cluster is stromal cells.

```
In [18]: # Prolif markers
genes = ['TOP2A', 'BIRC5', 'MKI67', 'AURKA', 'TPX2', 'KIF4A', 'CLSPN', 'CDKN1A',
        'ANLN', 'TACC3', 'FOXM1', 'CENPA', 'CDC20', 'NEK2', 'CENPF', 'KIF14']

common_genes = list(set(genes).intersection(adata.var.index))

sc.pl.umap(adata, color = common_genes, legend_loc = 'on data', color_map =
```



Here, we see that the bottom left cluster are cells in proliferation.

II. Expression of interest CTAs

This section permit us to if CTAs are expressed and see were there are expressed.

1) CTAs that impact survival

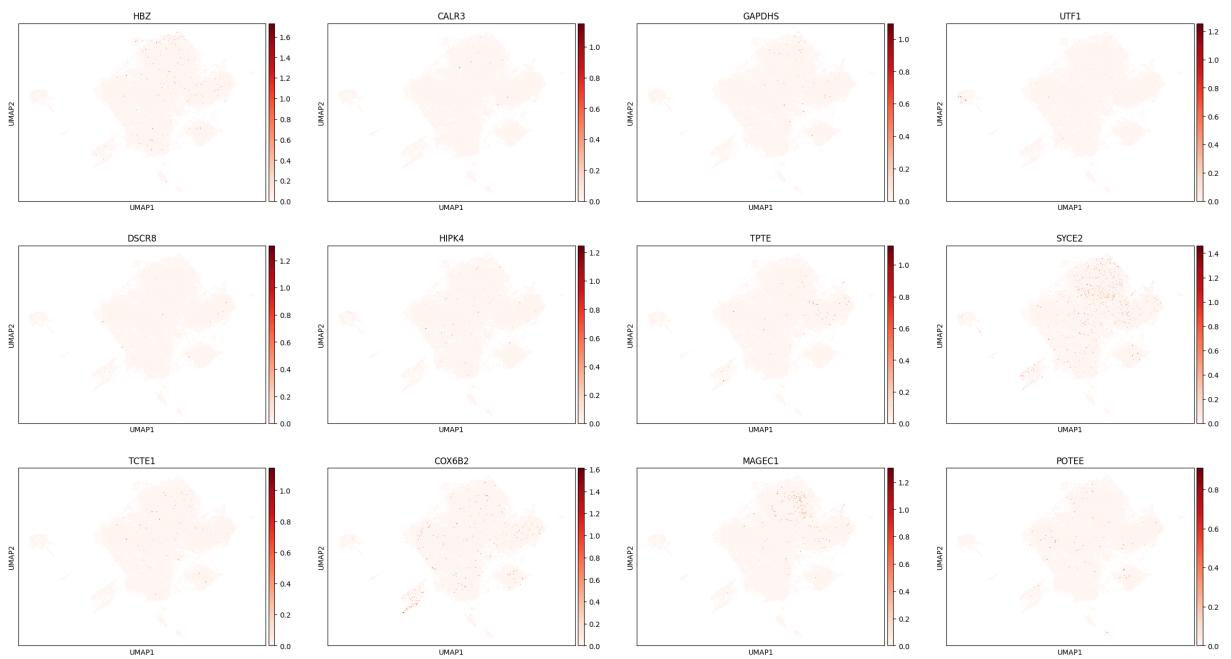
Here is the CTAs that impact survival in conventional chondrosarcomas with the clustering from heatmap of script 5 which analyze expression of genes in chondrosarcoma.

a- Cluster 1

```
In [19]: # Markers
# Cluster 1 conv
genes_clust_1 = ['HMGB4', 'CALR3', 'UTF1', 'HBZ', 'HES3', 'CCDC63', 'TSPY2',
                 'TCTE1', 'KRT72', 'CTAG2', 'SPACA5', 'COX6B2', 'HIPK4', 'GAPDHS',
```

```
'SCAND3', 'TPTE', 'SYCE2', 'DSCR8', 'MAGEC1', 'POTEE', 'POTEG', 'ARGFX']

genes_clust_1 = list(set(genes_clust_1).intersection(adata.var.index))
sc.pl.umap(adata, color = genes_clust_1 , color_map = plt.cm.Reds)
```

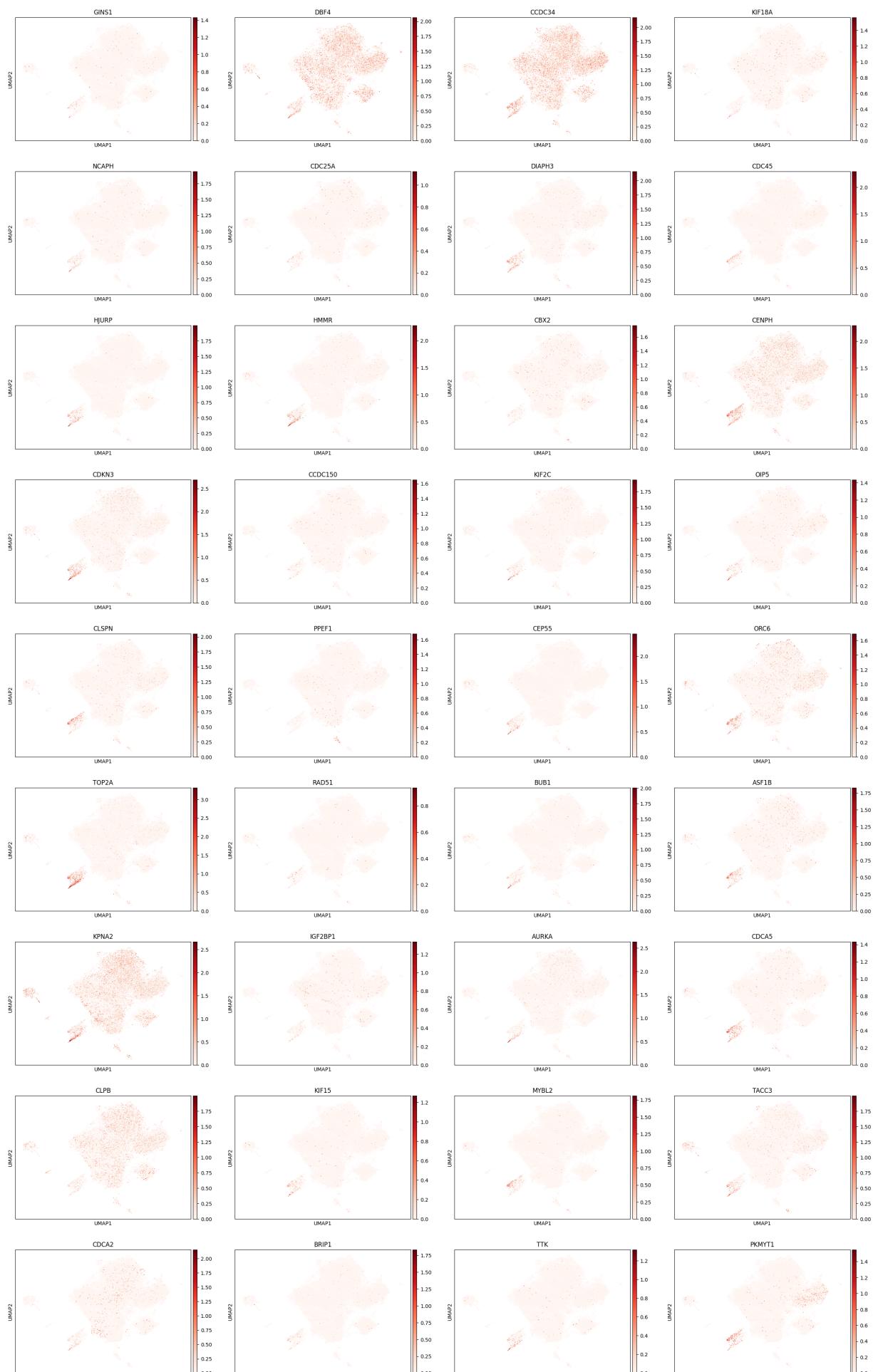


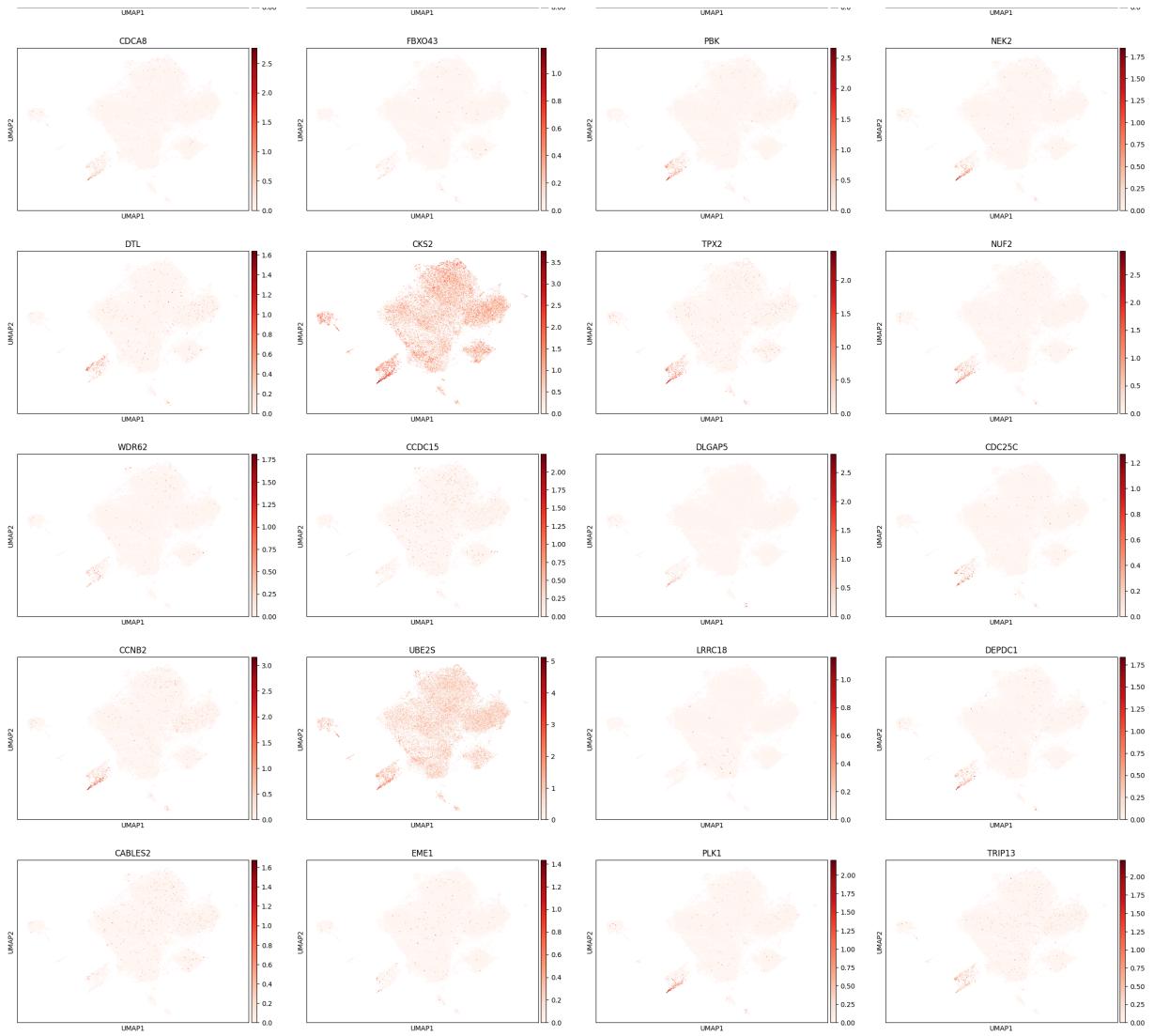
We see that the some CTA are expressed in some cells of the big cluster that probably contains chondrocytes, stromal cells and tumoral cells. We cannot conclude on the choice of one marker. Moreover, sc is not very sensitive so it is not surprising to see this.

b- Cluster 2

```
In [20]: # Markers
# Cluster 2 conv
genes_clust_2 = ['CLSPN', 'PPEF1', 'CABLES2', 'LRRC18', 'CCDC150', 'FBX043',
'CCDC34', 'OIP5', 'CCDC15', 'ORC6', 'EME1', 'CDC25A', 'CENPH',
'ASF1B', 'MYBL2', 'GINS1', 'TRIP13', 'BRIP1', 'DTL', 'RAD51',
'CDC5', 'CDC45', 'PKMYT1', 'IGF2BP1', 'CBX2', 'CLPB', 'UBE2S',
'AURKA', 'CKS2', 'DBF4', 'KPNA2', 'CEP55', 'TACC3', 'KIF15',
'CCNB2', 'TOP2A', 'TPX2', 'CDCA8', 'CDKN3', 'HMMR', 'KIF2C',
'DLGAP5', 'KIF18A', 'PLK1', 'NCAPH', 'NEK2', 'DEPDC1', 'CDCA2',
'BUB1', 'PBK', 'HJURP', 'TTK', 'NUF2', 'WDR62', 'CDC25C', 'DIAPH3']

genes_clust_2 = list(set(genes_clust_2).intersection(adata.var.index))
sc.pl.umap(adata, color = genes_clust_2 , color_map = plt.cm.Reds)
```



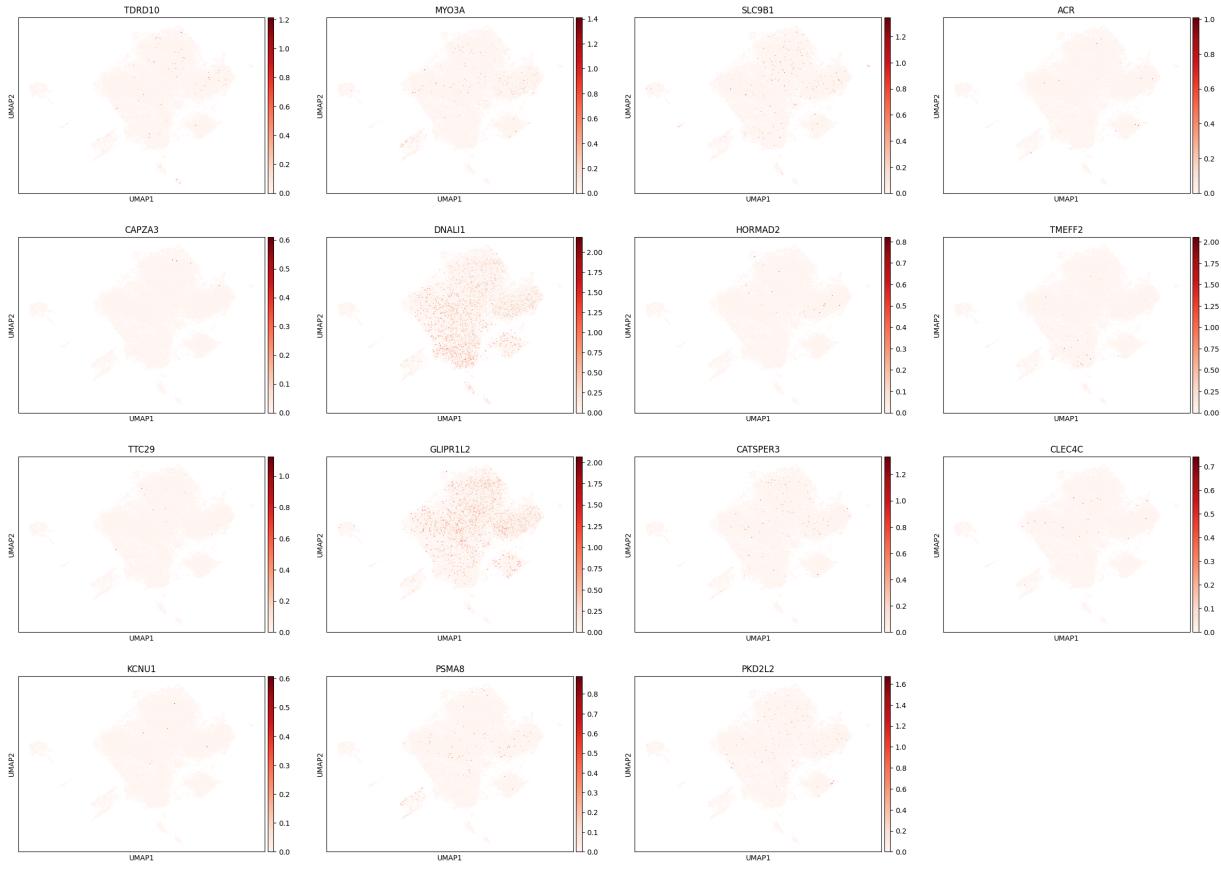


Thanks to literature, we observe that genes from cluster 2 may be proliferation genes and this fact is confirmed here. Most of the genes are expressed in the proliferation cluster.

c- Cluster 3

```
In [21]: # Markers
# Cluster 3 conv
genes_clust_3 = ['ACR', 'DNALI1', 'GK2', 'TDRD10', 'PKD2L2', 'CATSPER3', 'TTI',
                 'BOLL', 'DEFB126', 'TMEFF2', 'MYO3A', 'HORMAD2', 'PRSS37', 'SLC06A1',
                 'GLIPR1L2', 'CLEC4C', 'FSHR', 'ASB17', 'OR13C3', 'SLC9B1', 'SLC9C1',
                 'GLIPR1L1', 'TMC05A', 'ADAM7', 'SEPTIN14', 'PSMA8', 'OR56A3', 'TMEM202',
                 'KCNU1', 'CAPZA3']

genes_clust_3 = list(set(genes_clust_3).intersection(adata.var.index))
sc.pl.umap(adata, color = genes_clust_3 , color_map = plt.cm.Reds)
```



Like cluster 1, we have not a CTA very expressed, but some of it are expressed in the big cluster.

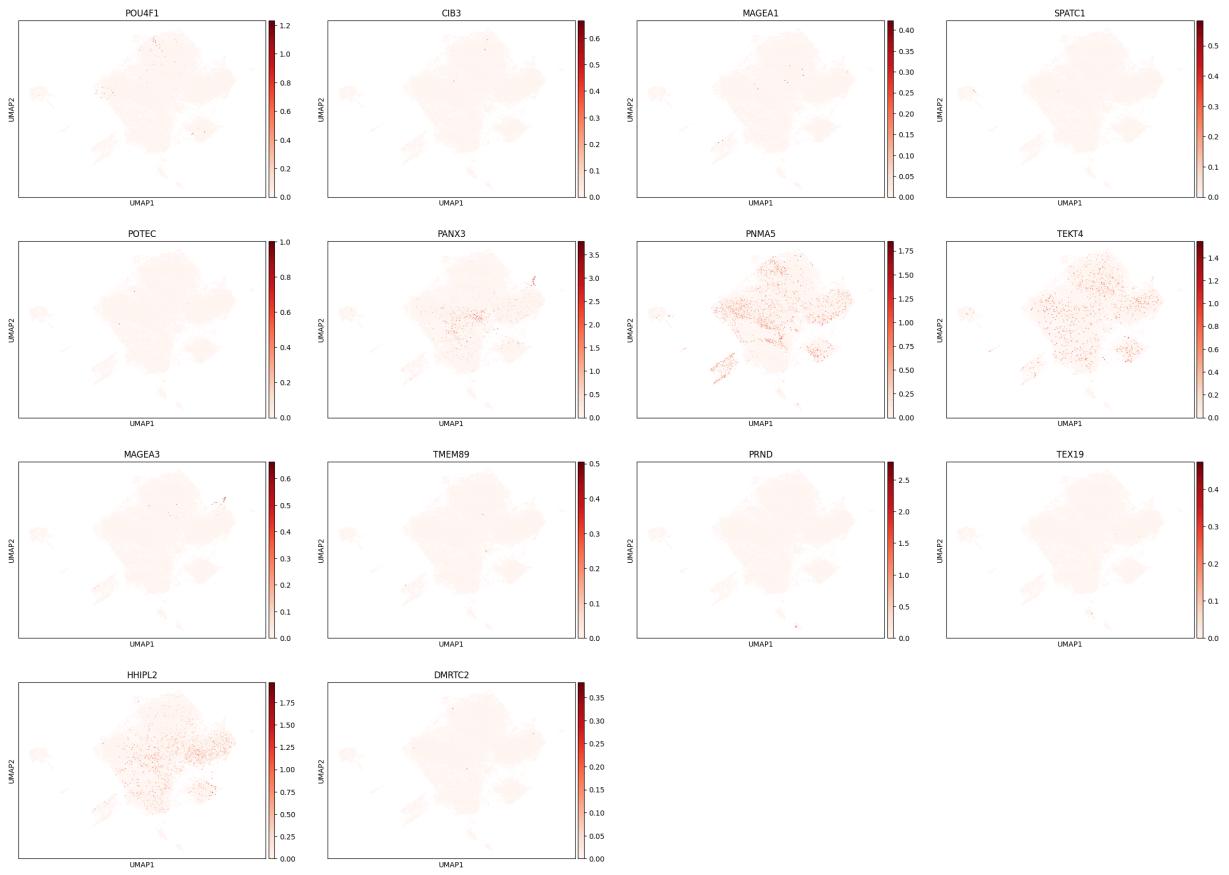
2) CTAs a bit expressed in normal tissues but expressed in chondrosarcoma

In this part, we observe CTA that are not very expressed in normal tissues and expressed in chondrosarcoma thanks to the scatter plot from script 8.

```
In [22]: # Markers of specific chondrosarcoma genes
genes_scatter = ['ZNF705D', 'MAGEB6', 'MAGEB16', 'CNGA2', 'POU4F1',
                 'DMRTC2', 'VCX2', 'CSH2', 'TMEM89', 'PRND', 'PNMA5',
                 'CT45A1', 'PANX3', 'HHIPL2', 'SPATC1', 'TEKT4', 'MAGEA1',
                 'POTEC', 'CT47B1', 'MTNR1B', 'FTHL17', 'C2orf78', 'CIB3',
                 'CCDC166', 'TEX19', 'DMBX1', 'C20orf141', 'PRDM14', 'BIRC8',
                 'MAGEA3', 'BSX']

genes_scatter = list(set(genes_scatter).intersection(adata.var.index))

sc.pl.umap(adata, color = genes_scatter, legend_loc = 'on data', color_map =
```



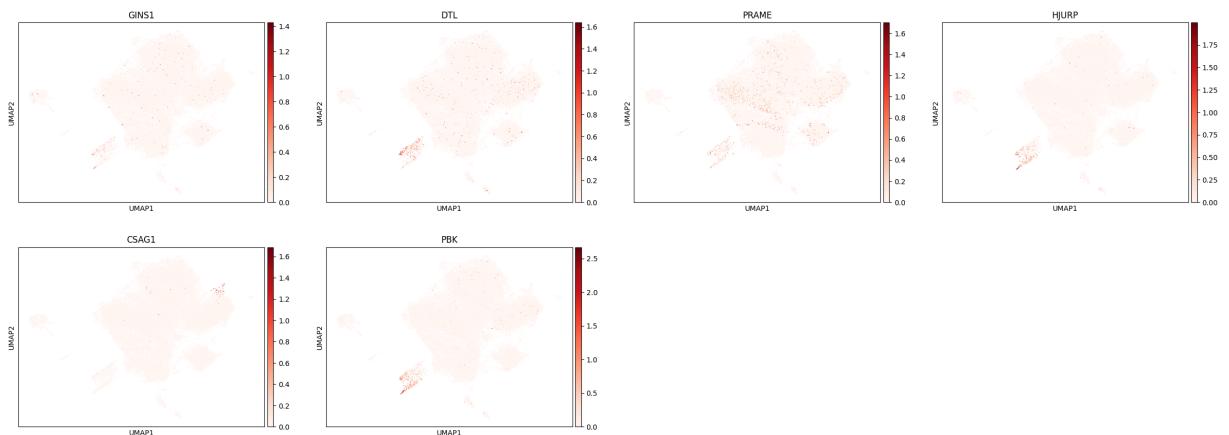
PNMA5, TEKT4, HHPL2 and PANX3 seems to be the most interested CTAs in this dataset.

3) CTAs of interest

We interested us in PRAME because it is a known CTA.

```
In [23]: genes = ['DTL', 'HJURP', 'GINS1', 'PBK', 'CTAG2', 'CT45A1', 'CSAG1', 'PRAME']
genes = list(set(genes).intersection(adata.var.index))

sc.pl.umap(adata, color = genes, legend_loc = 'on data', color_map = mpl.cm.
```

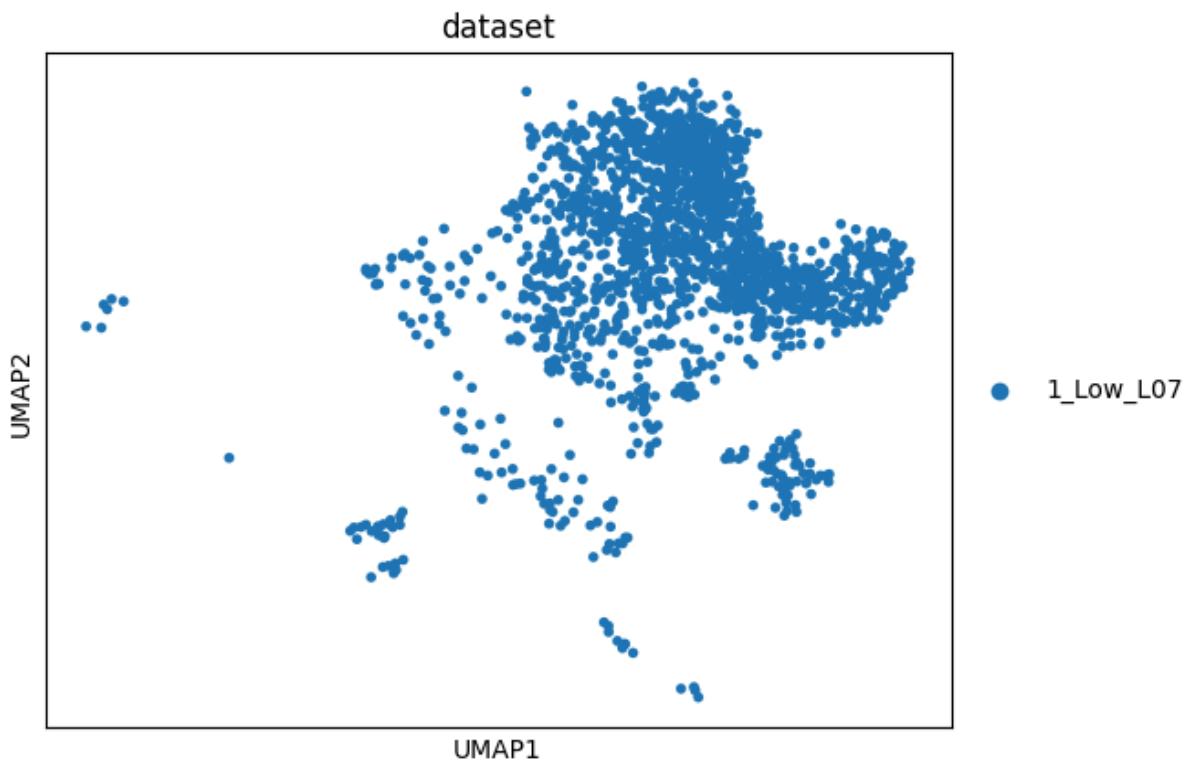


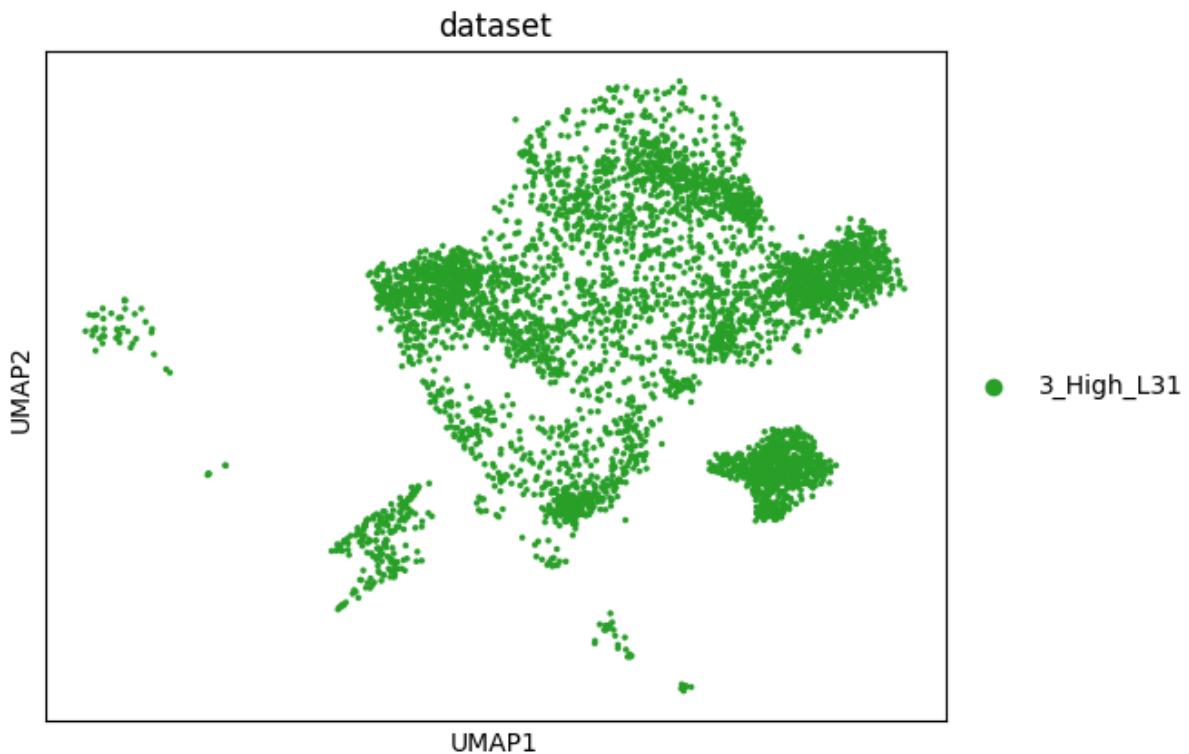
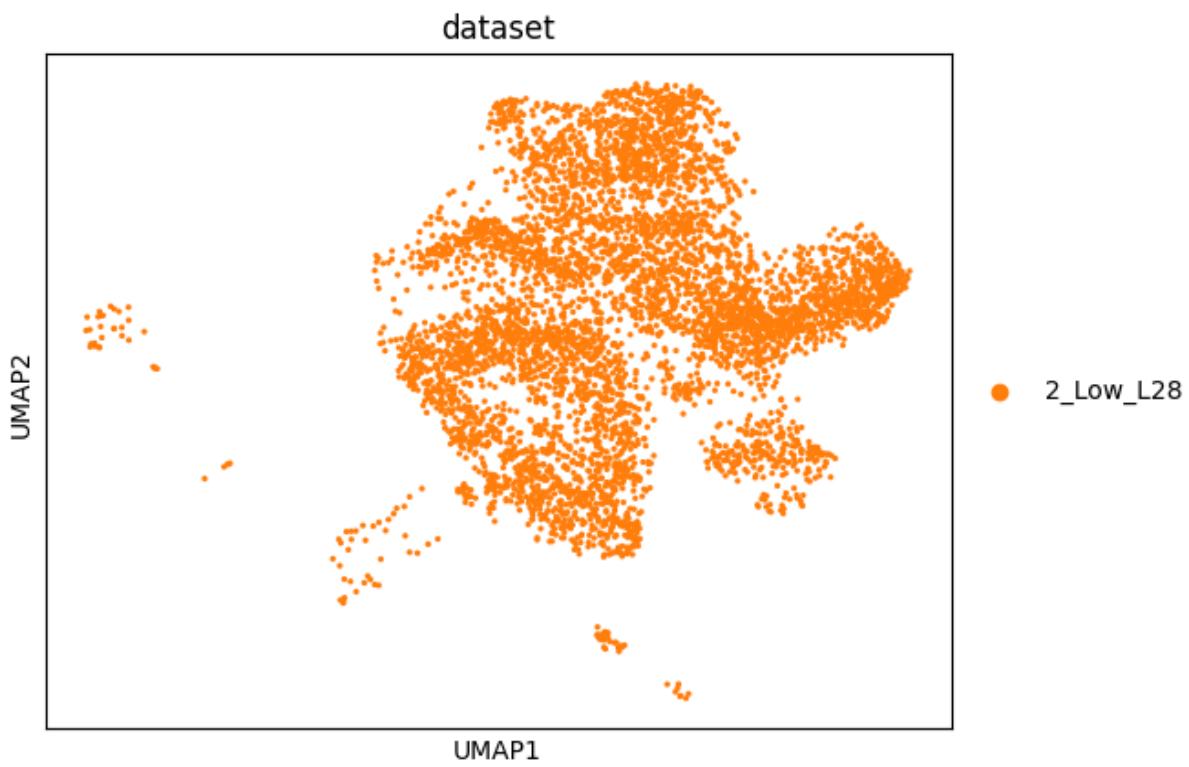
III. Expression of genes per patients

This section is about the expression of genes per patients to see if CTAs are more expressed in high grade or not and also the expression of MHC.

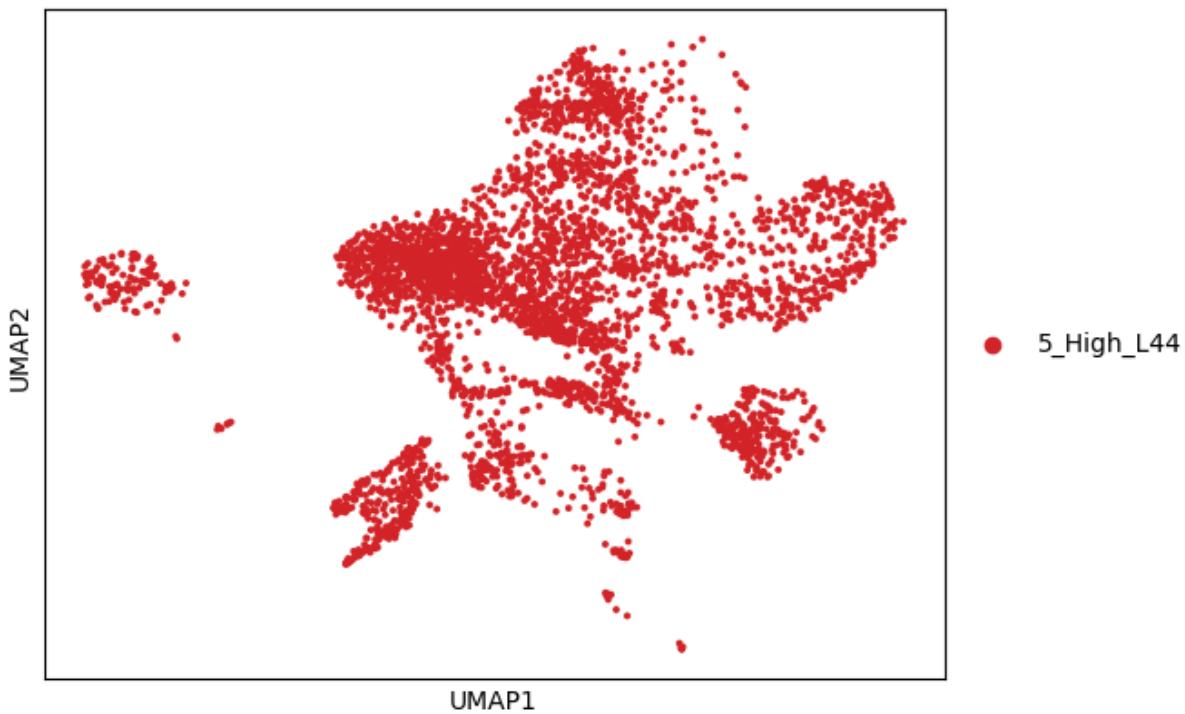
1) Colored by patients

```
In [24]: l = ['1_Low_L07', '2_Low_L28', '3_High_L31', '5_High_L44', '6_Ben_L49', '7_M  
for dataset in l:  
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]  
    sc.pl.umap(adata_filtered, color = 'dataset', color_map = plt.cm.Reds)
```

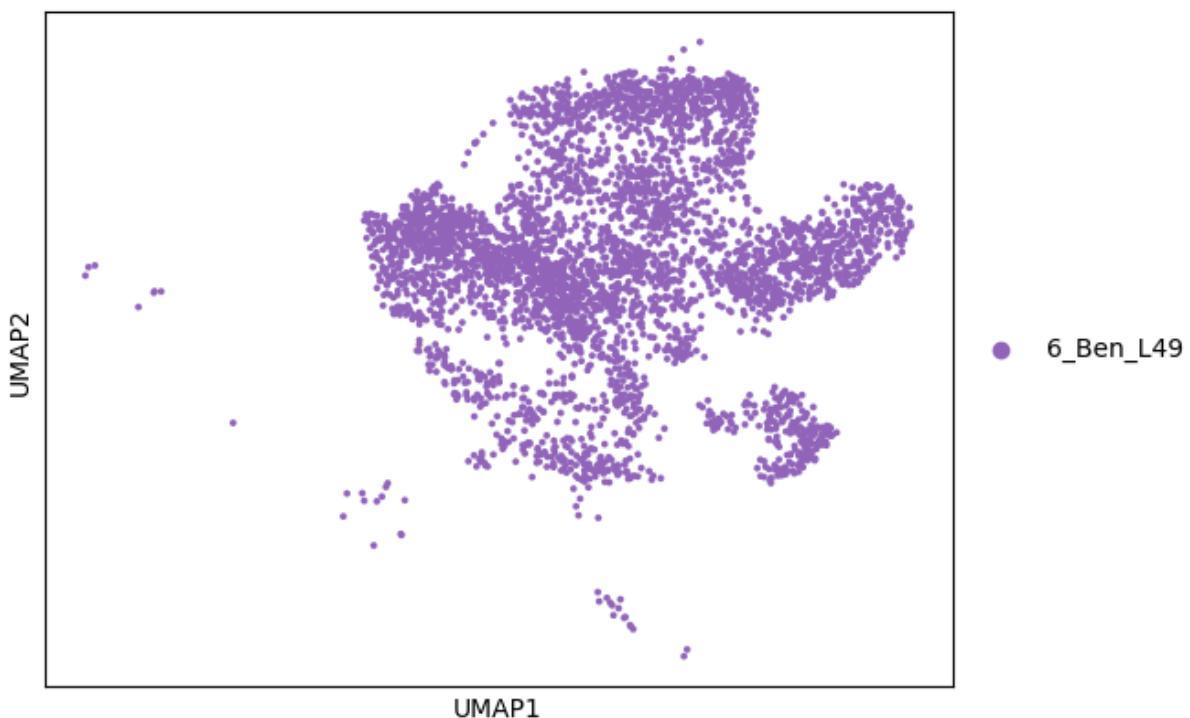


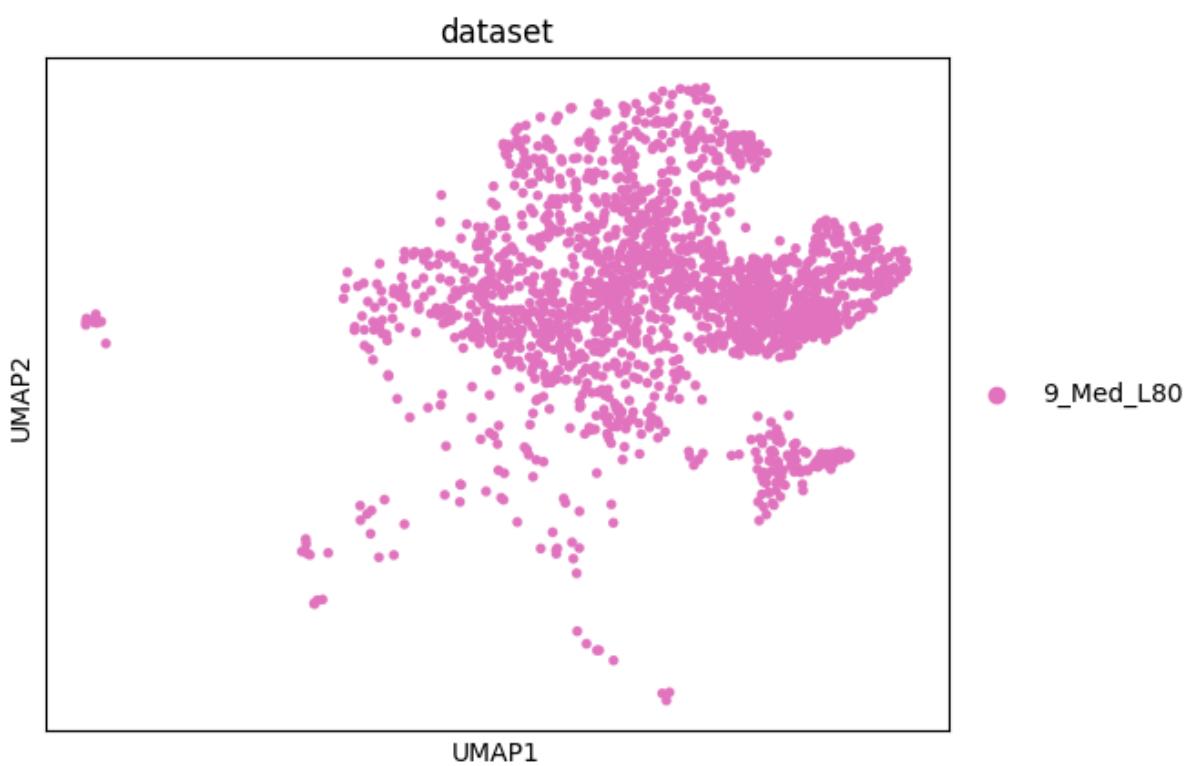
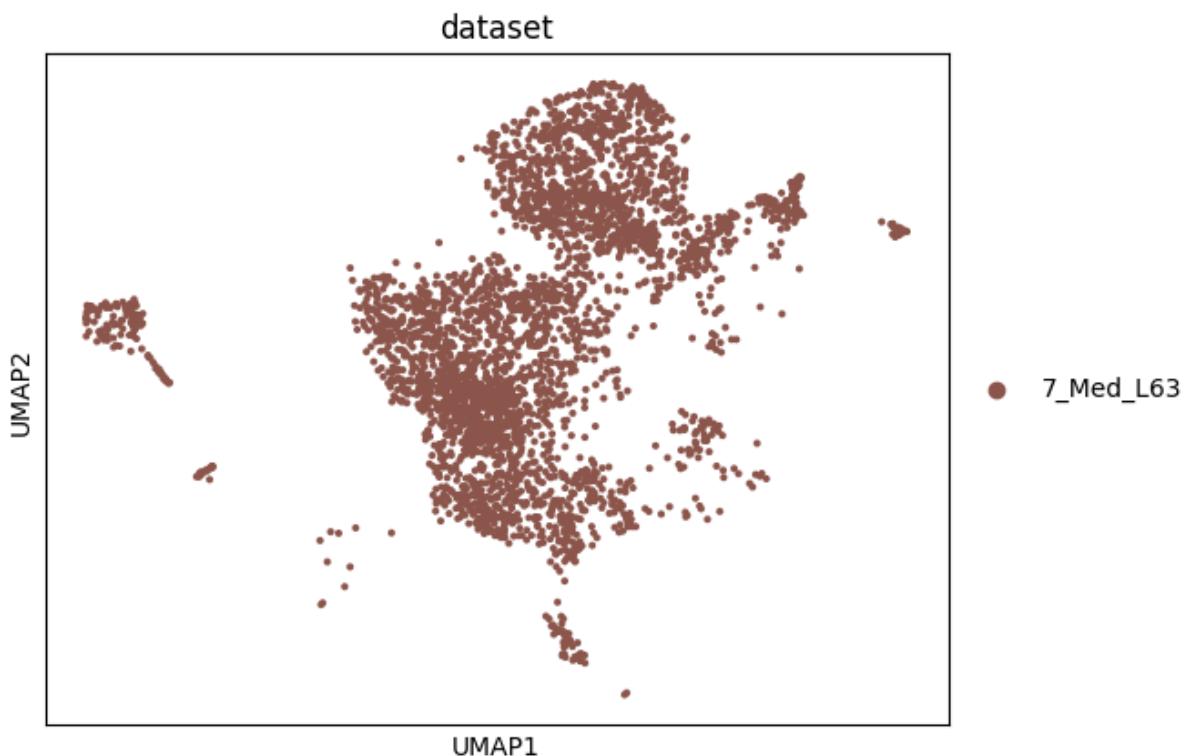


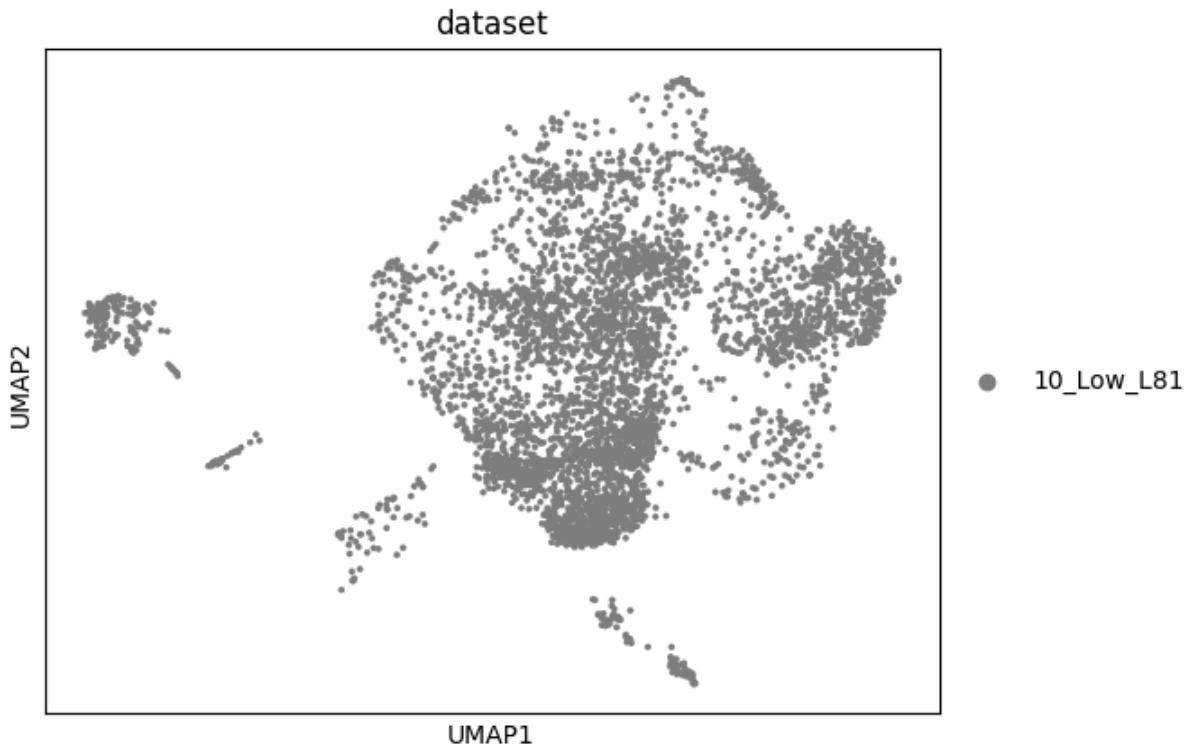
dataset



dataset

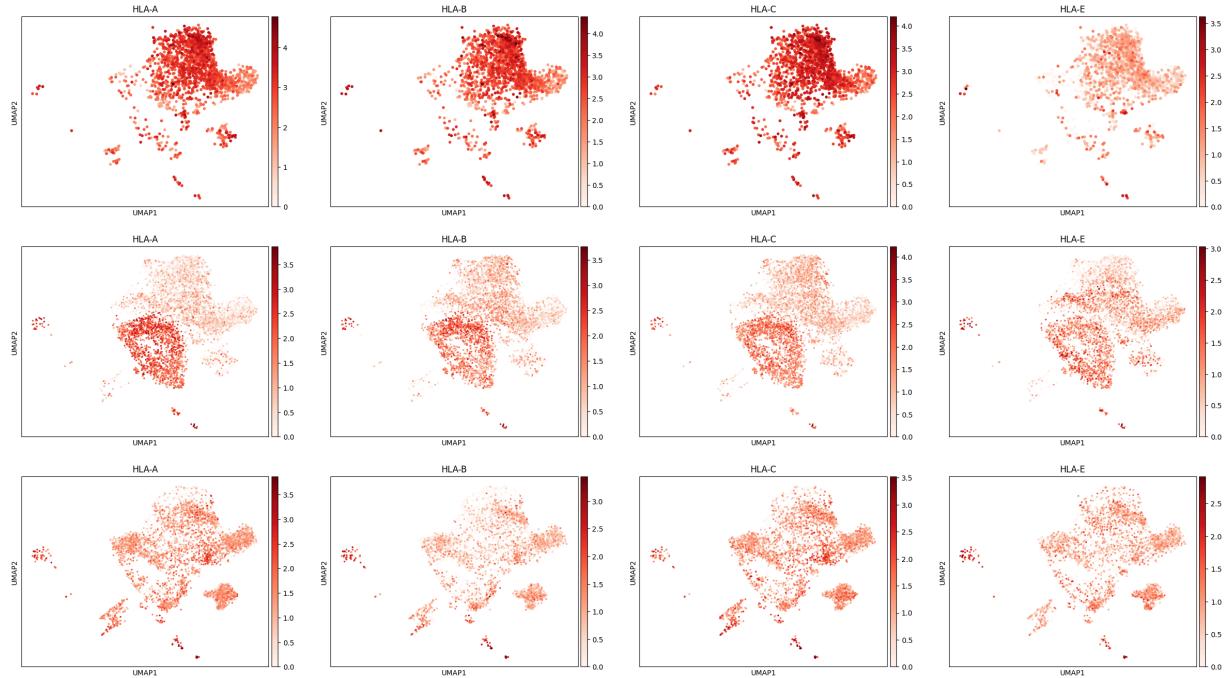


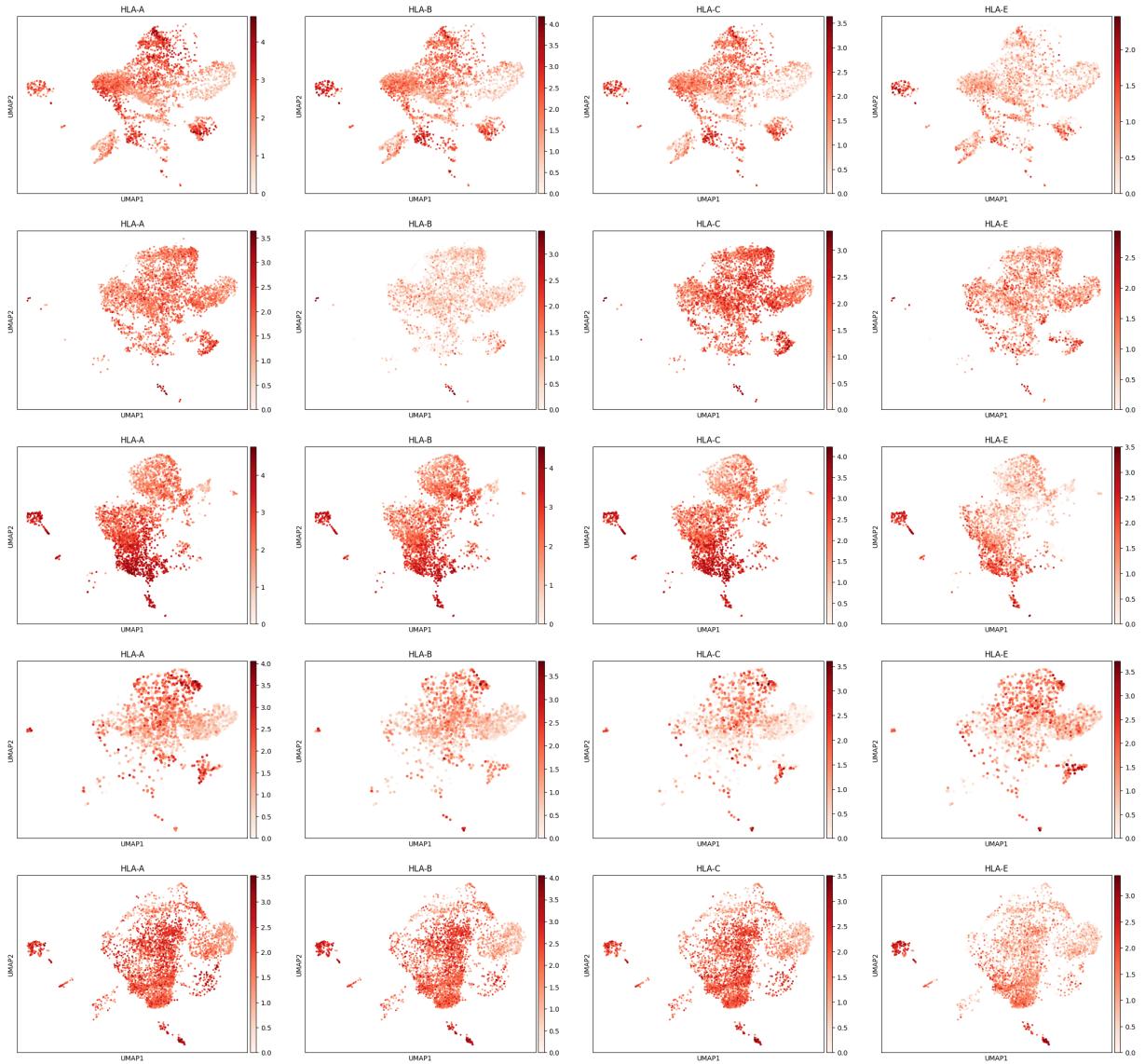




2) MHC 1 genes by samples

```
In [25]: for dataset in l:
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]
    sc.pl.umap(adata_filtered, color = ['HLA-A', 'HLA-B', 'HLA-C', 'HLA-E'])
```



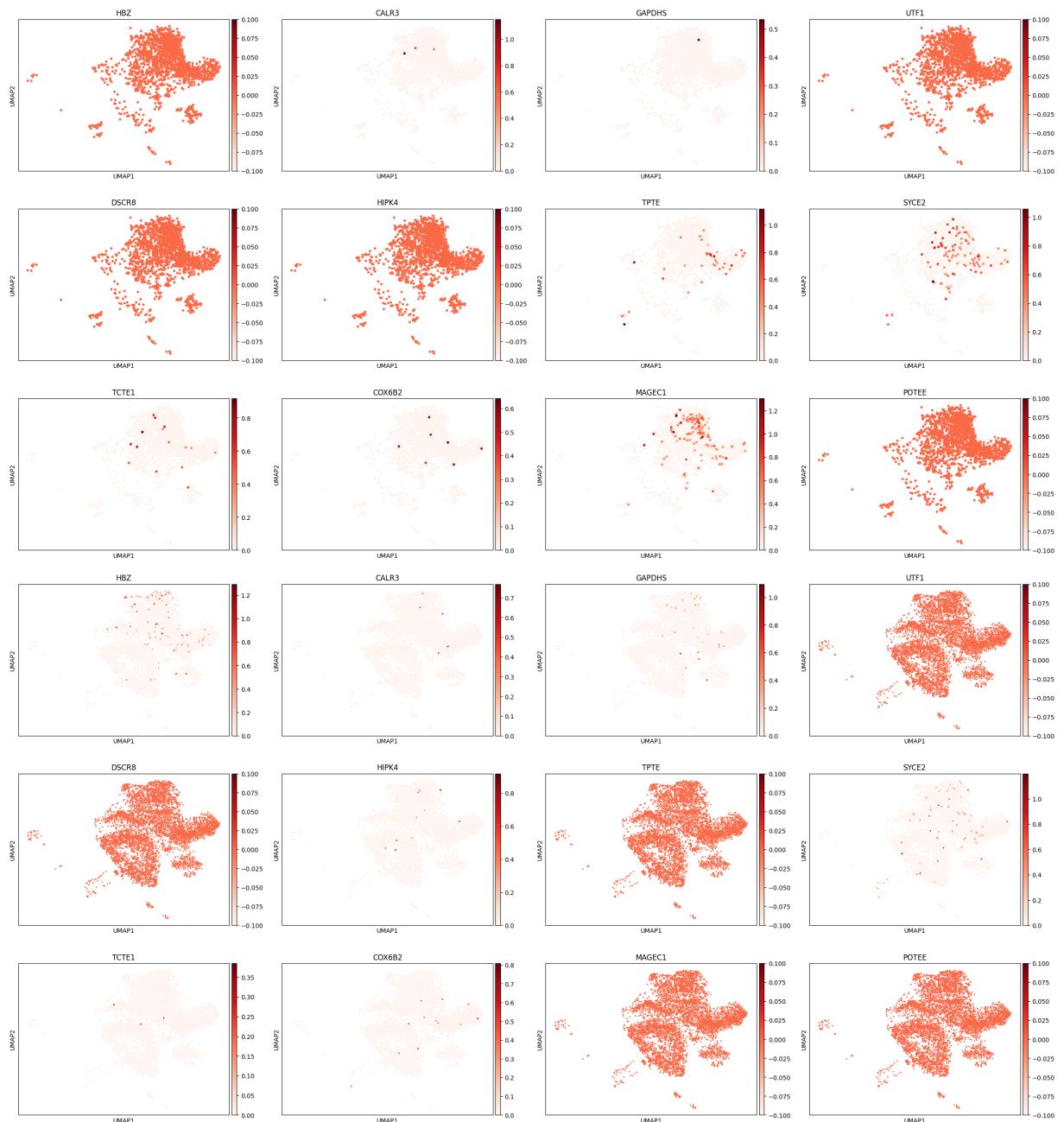


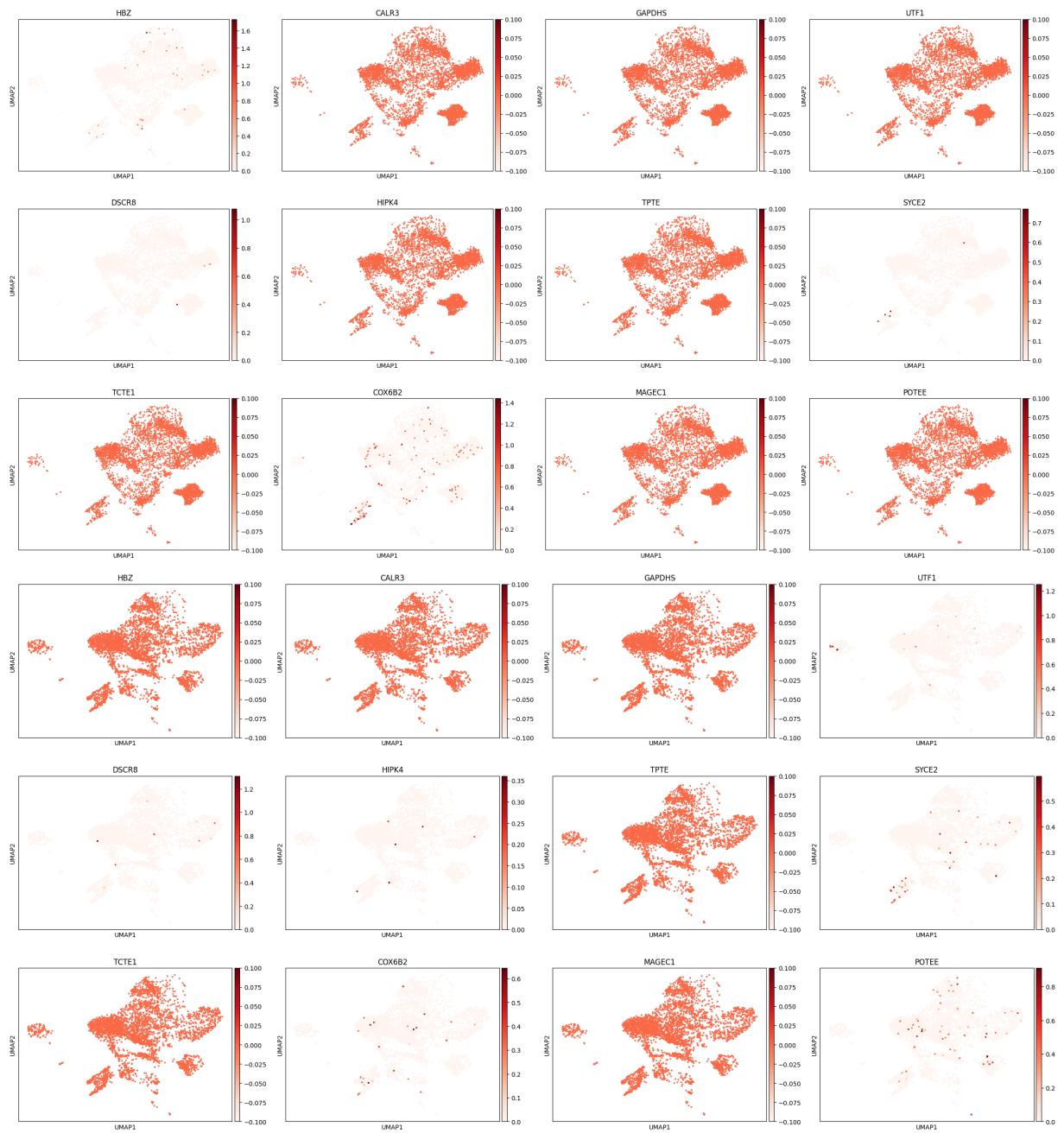
We see that MHC 1 is expressed everywhere more or less for some samples. We can deduce that tumoral cells expressed MHC 1 in chondrosarcoma, so a peptide is presented and could be targeted.

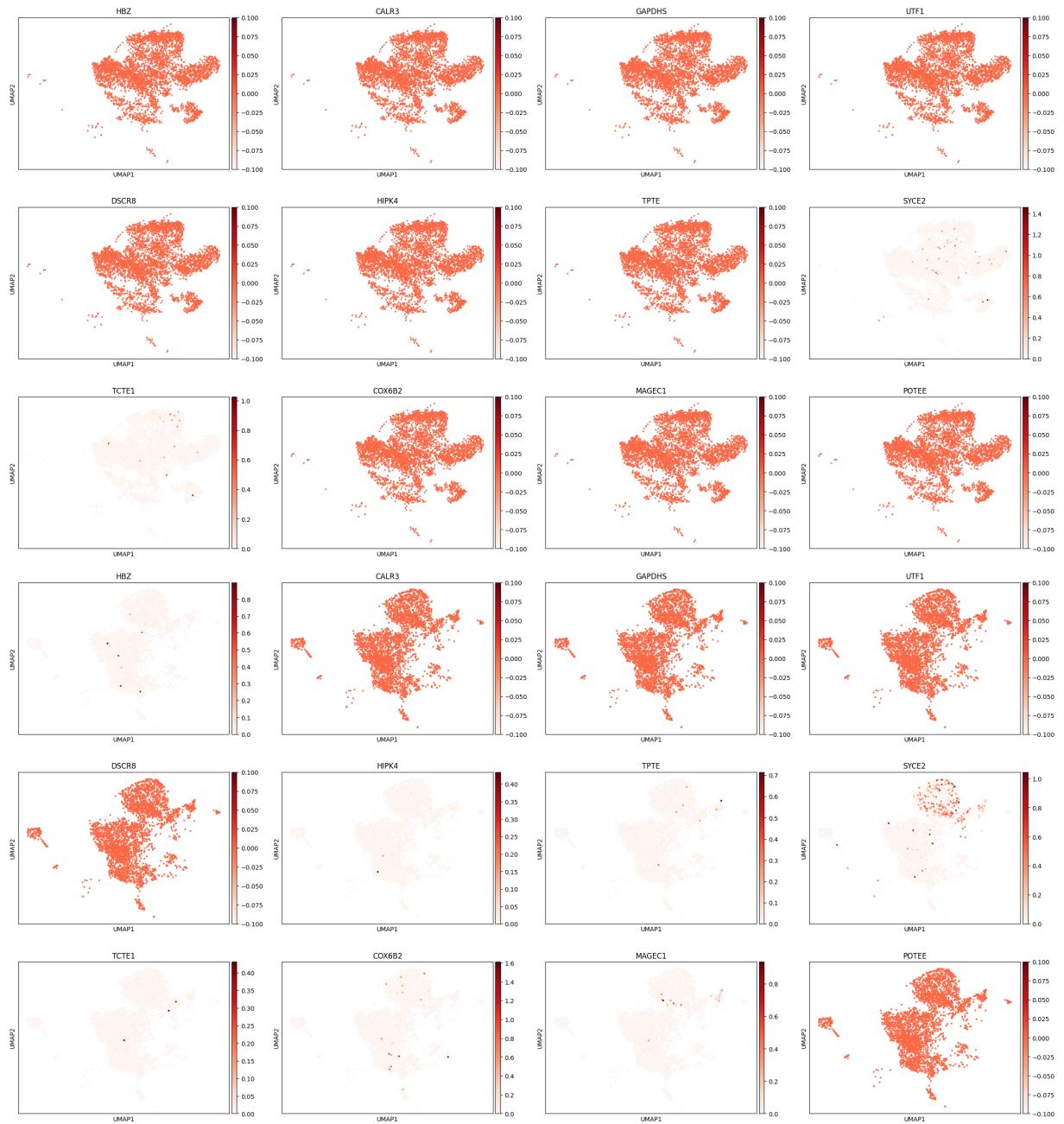
3) CTAs that impact survival

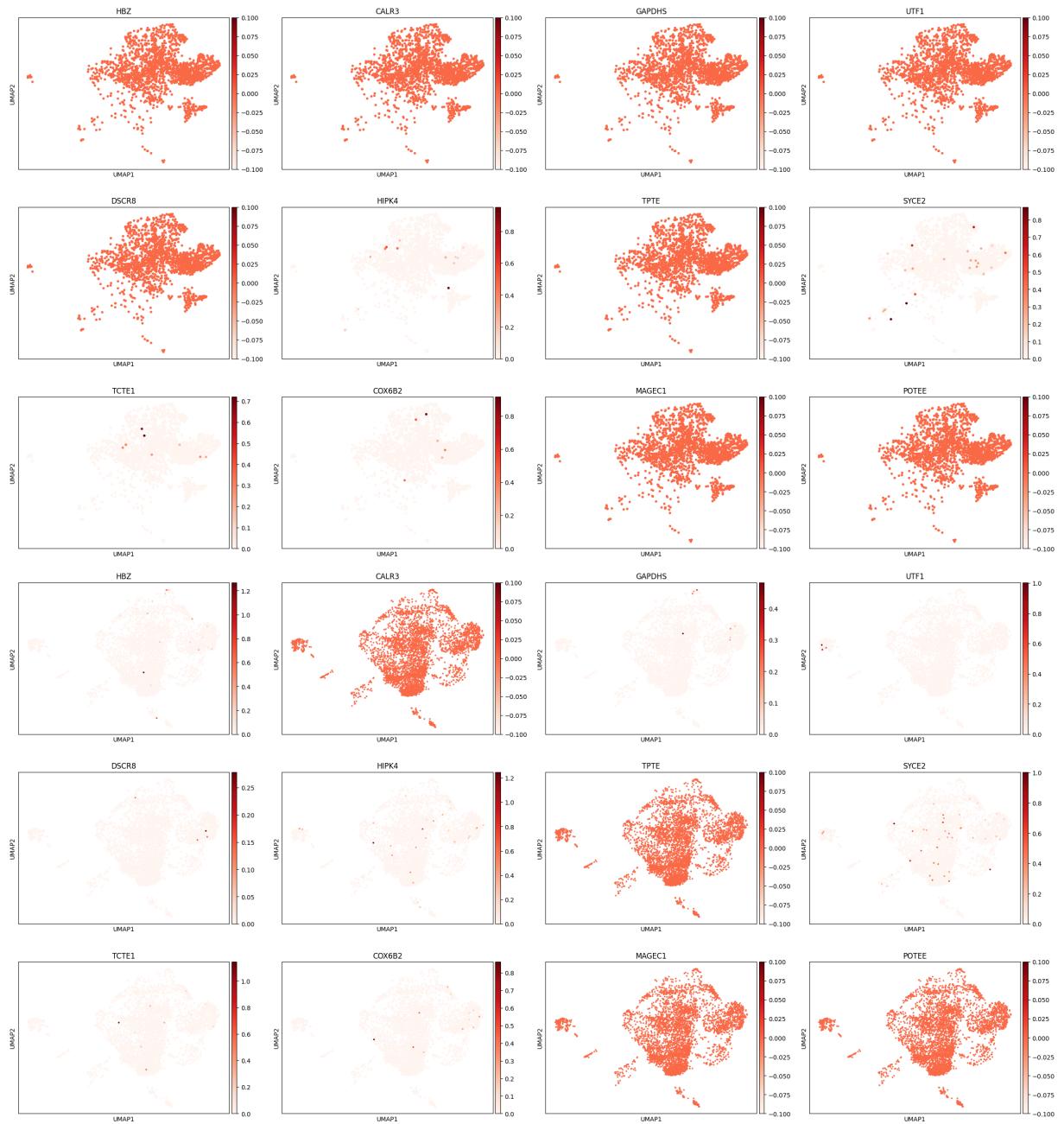
a- Cluster 1

```
In [26]: # Markers
# Cluster 1 conv
for dataset in l:
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]
    sc.pl.umap(adata_filtered, color = genes_clust_1 , color_map = plt.cm.RdYlBu_r)
```





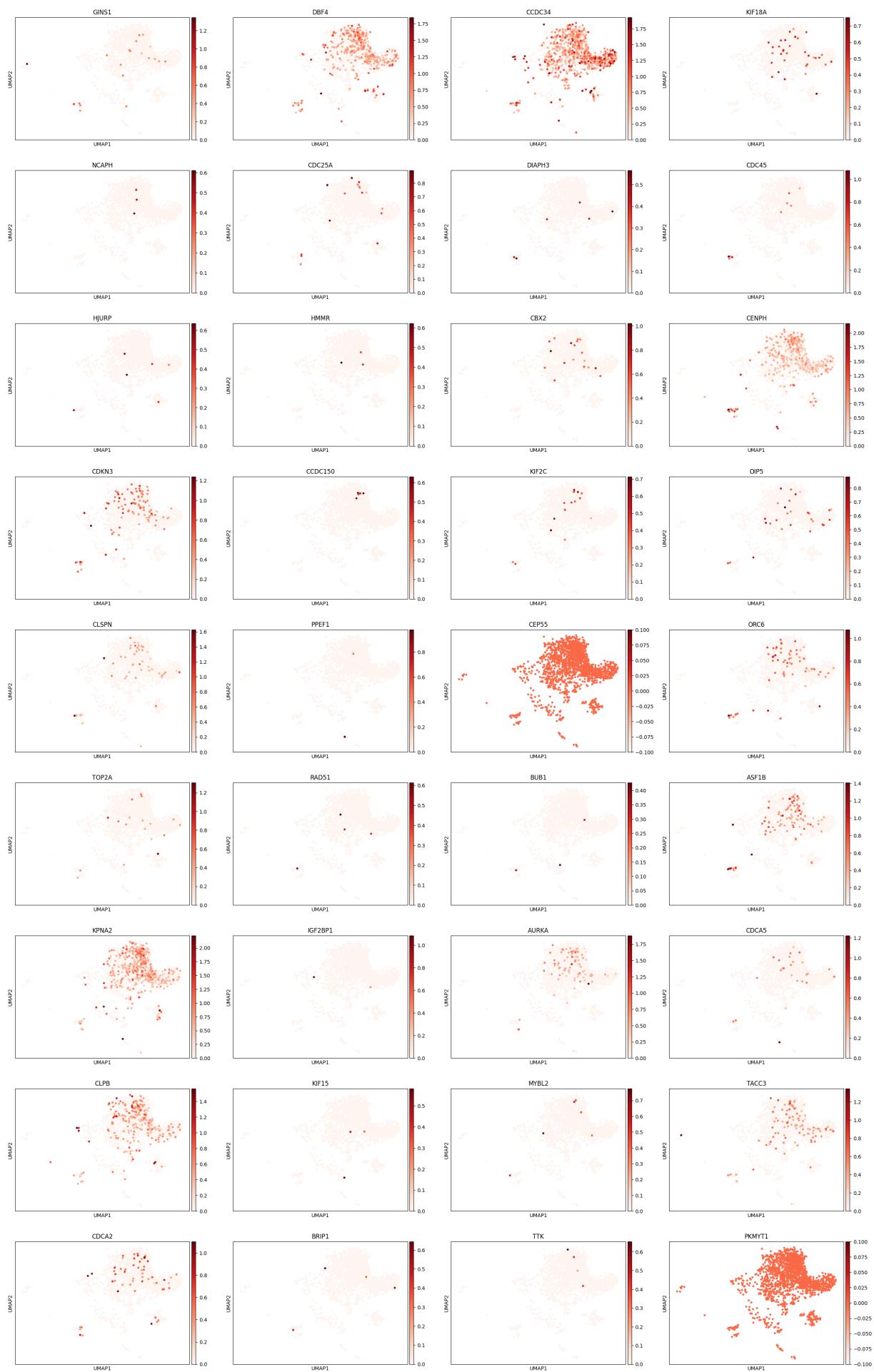


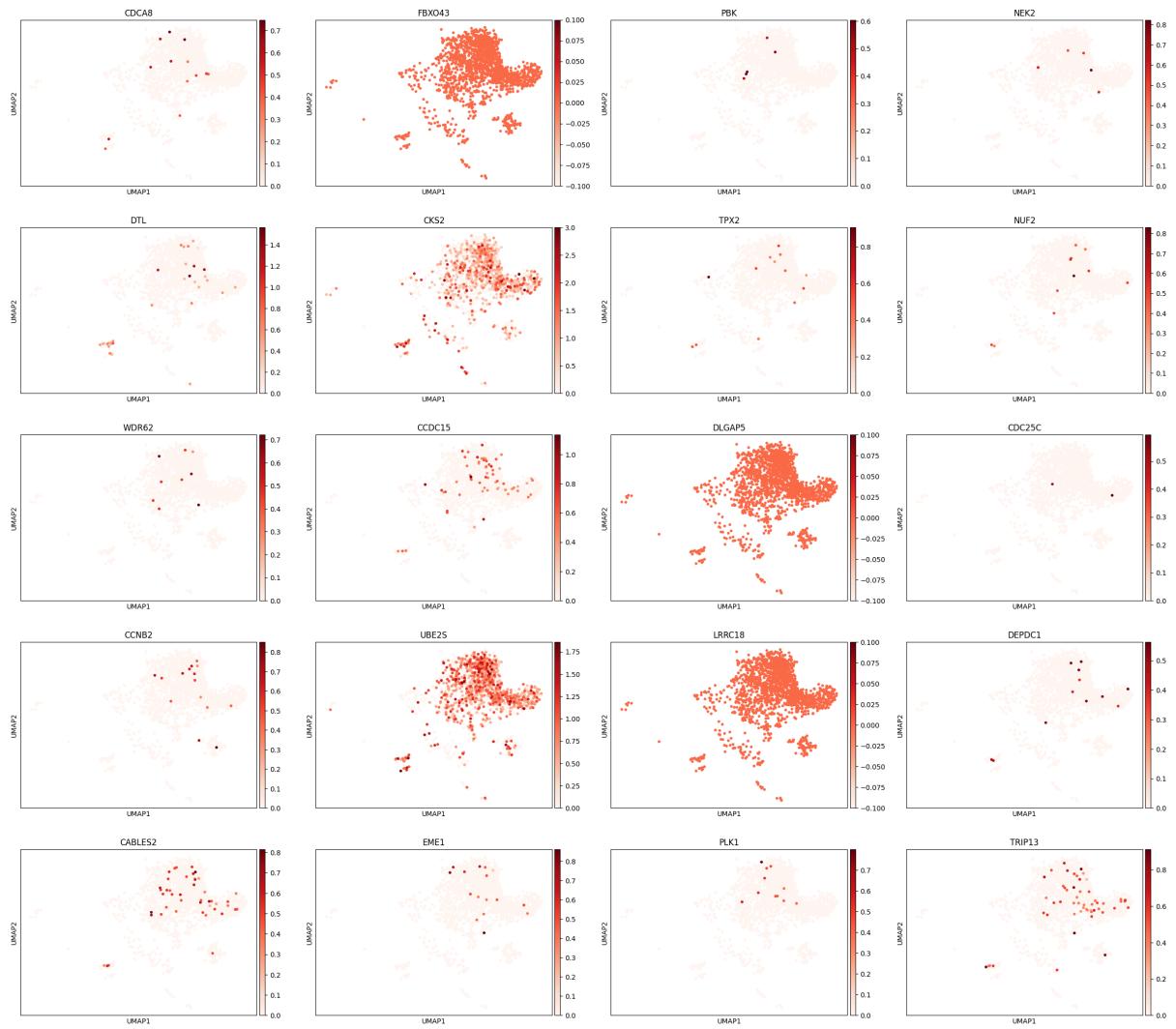


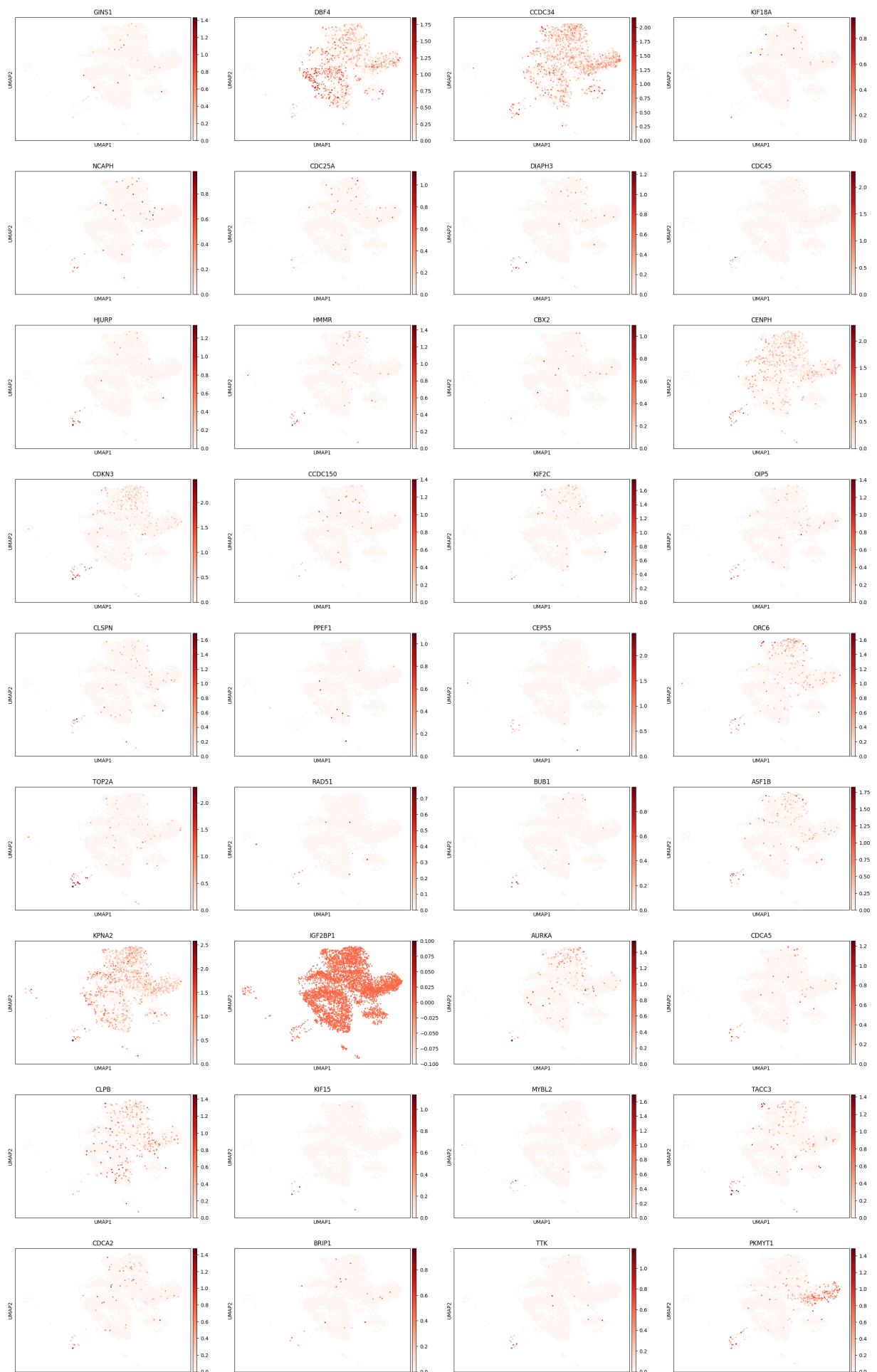
We can better see the expression of CTA per samples.

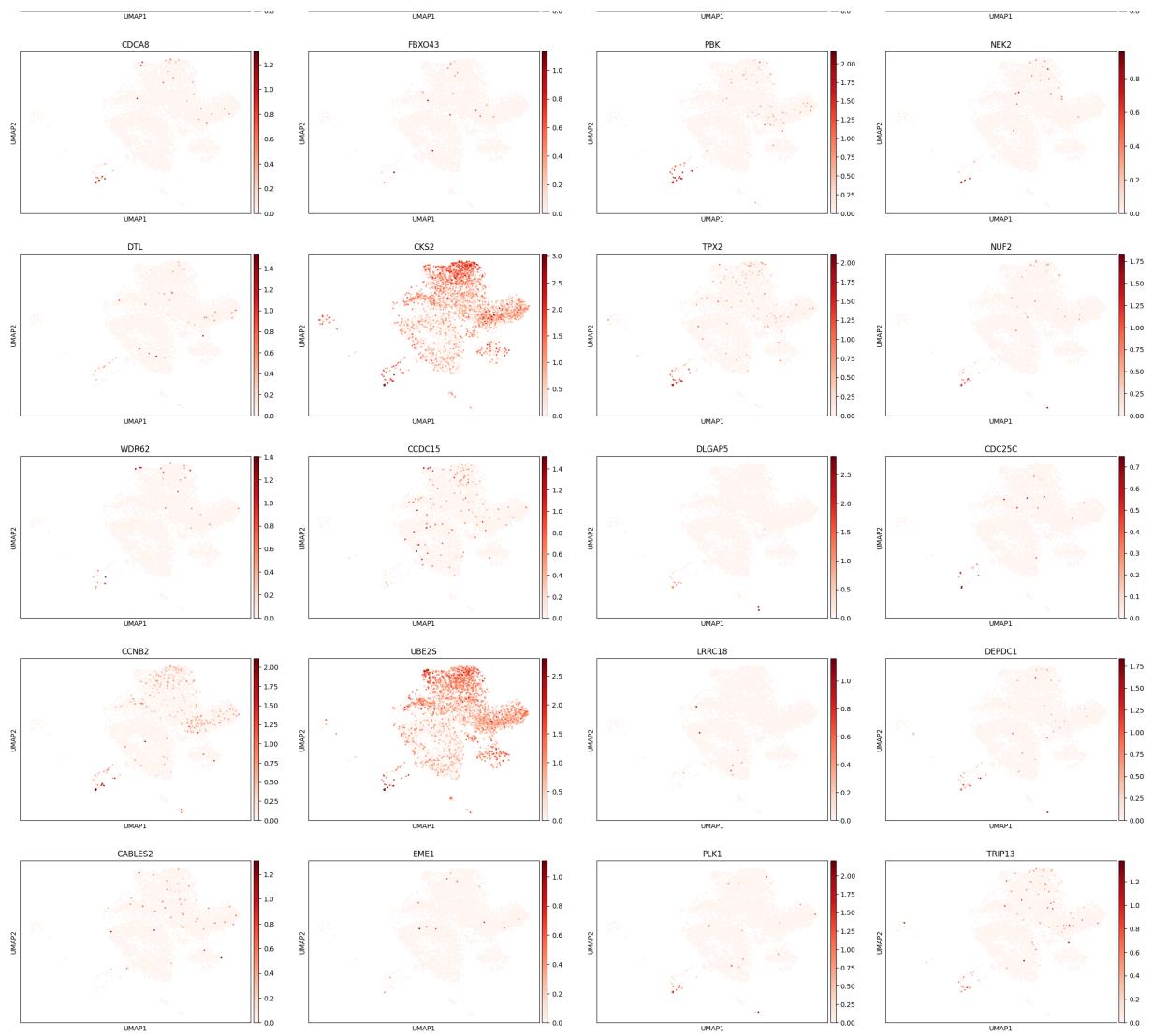
b- Cluster 2

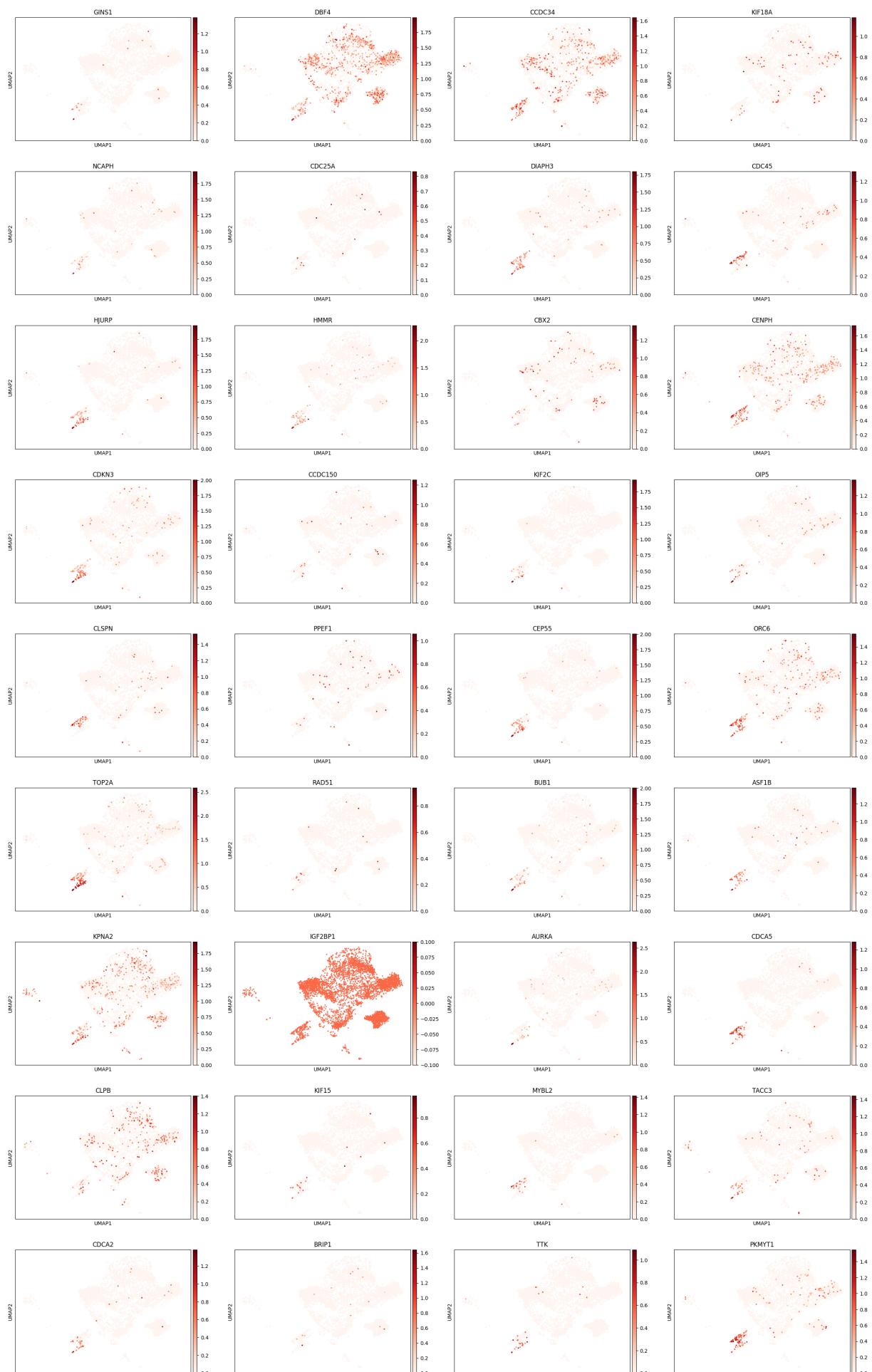
```
In [27]: # Markers cluster 2
for dataset in l:
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]
    sc.pl.umap(adata_filtered, color = genes_clust_2 , color_map = plt.cm.RdYlBu_r)
```

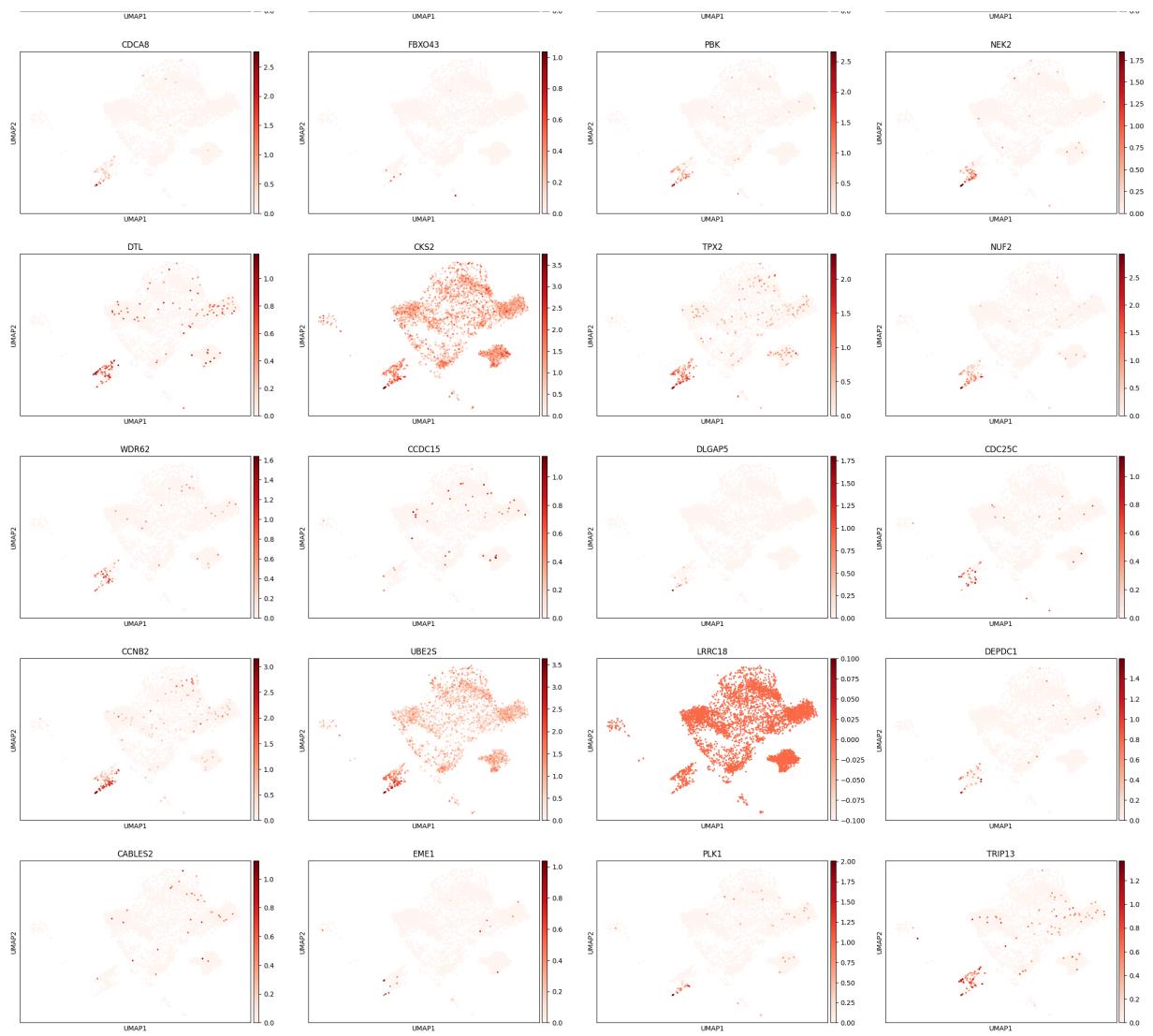


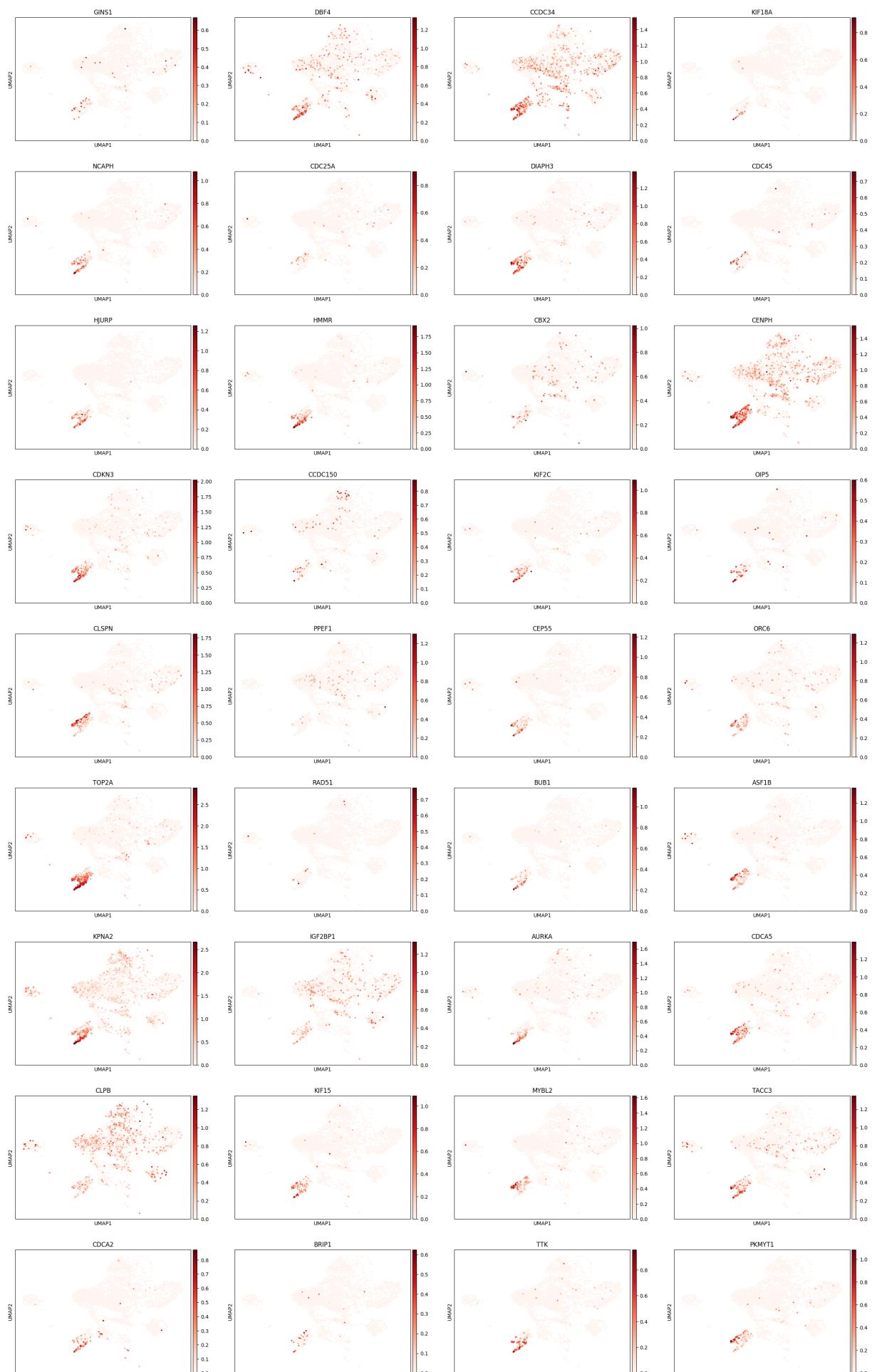


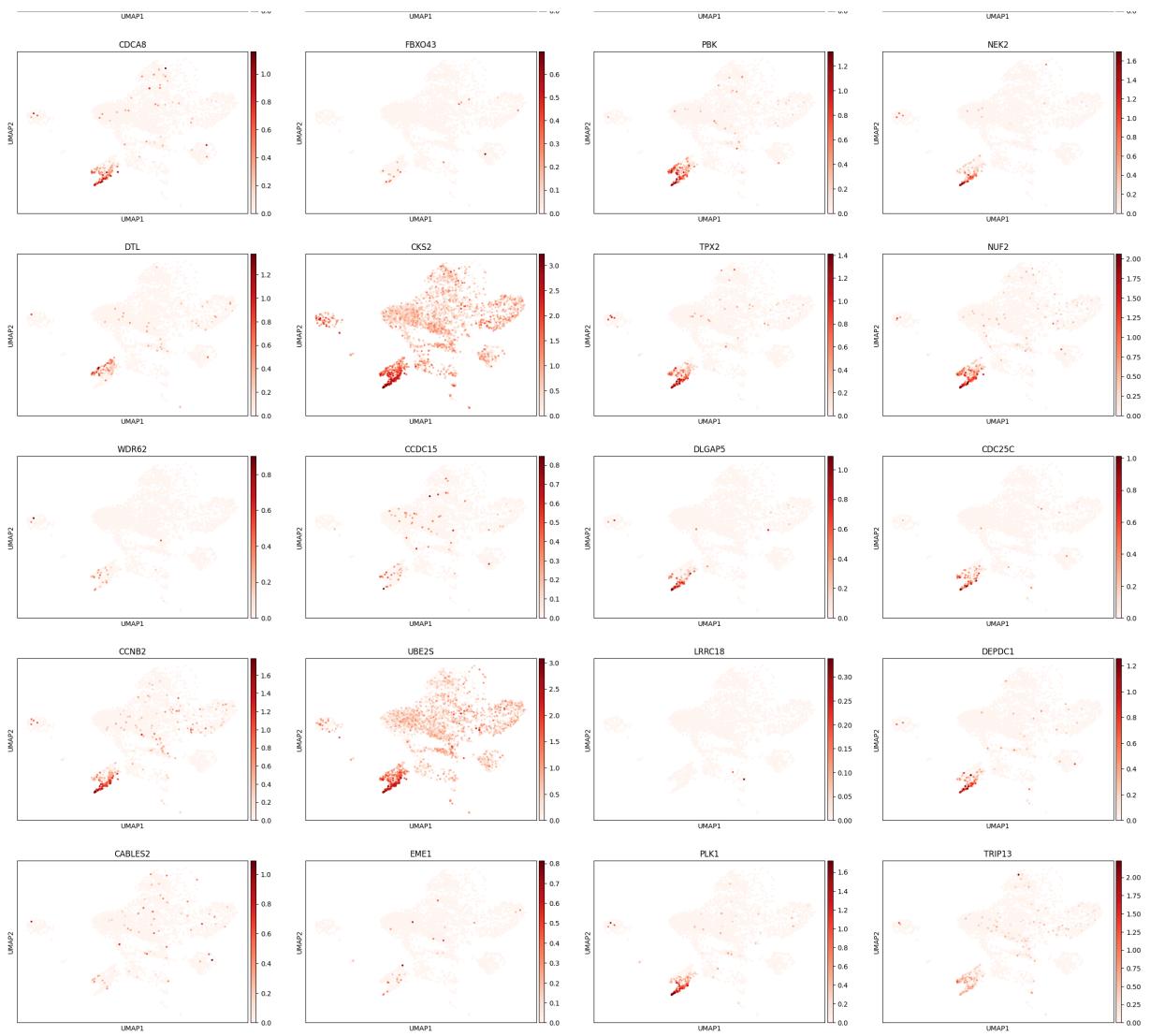


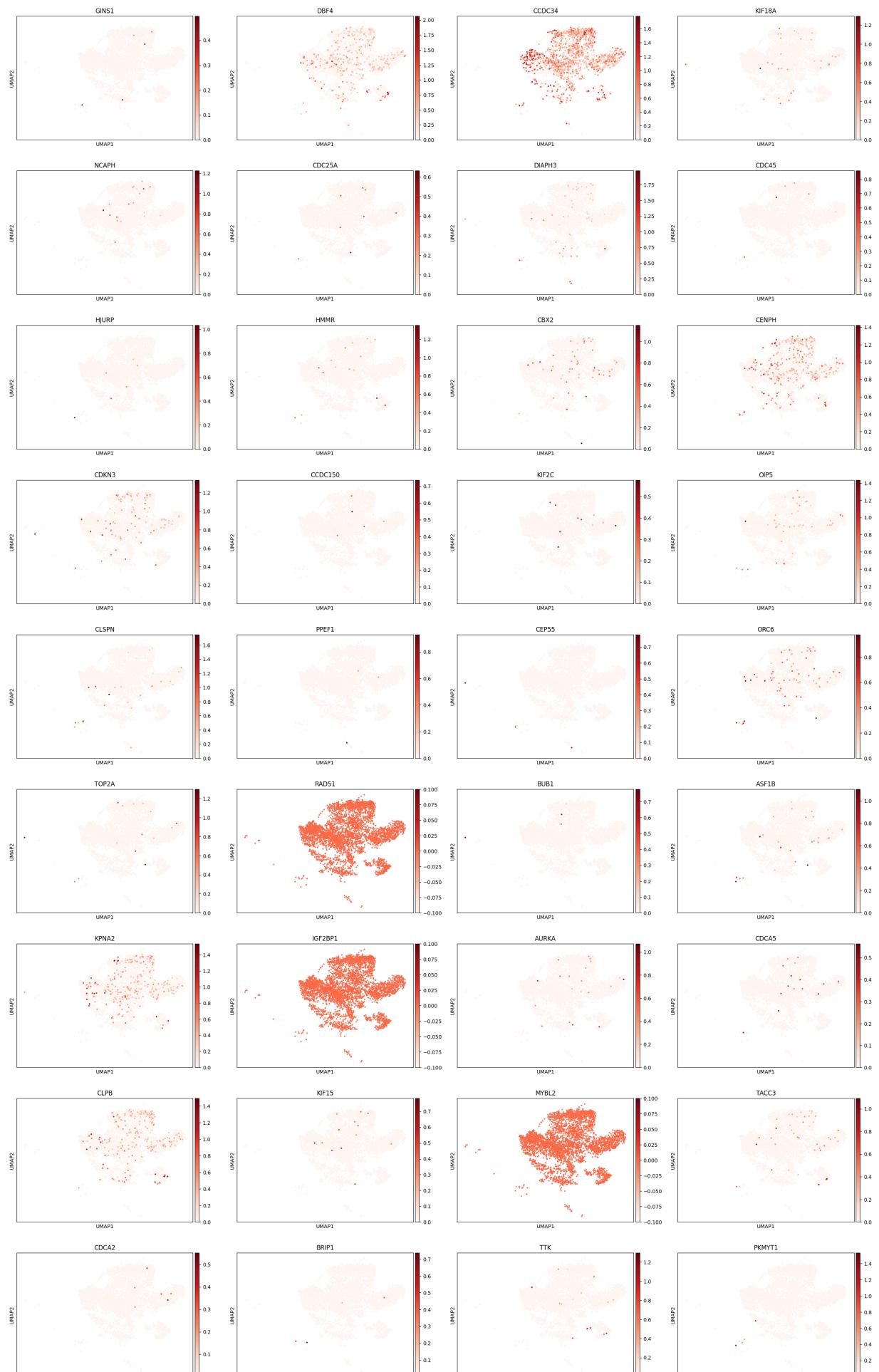


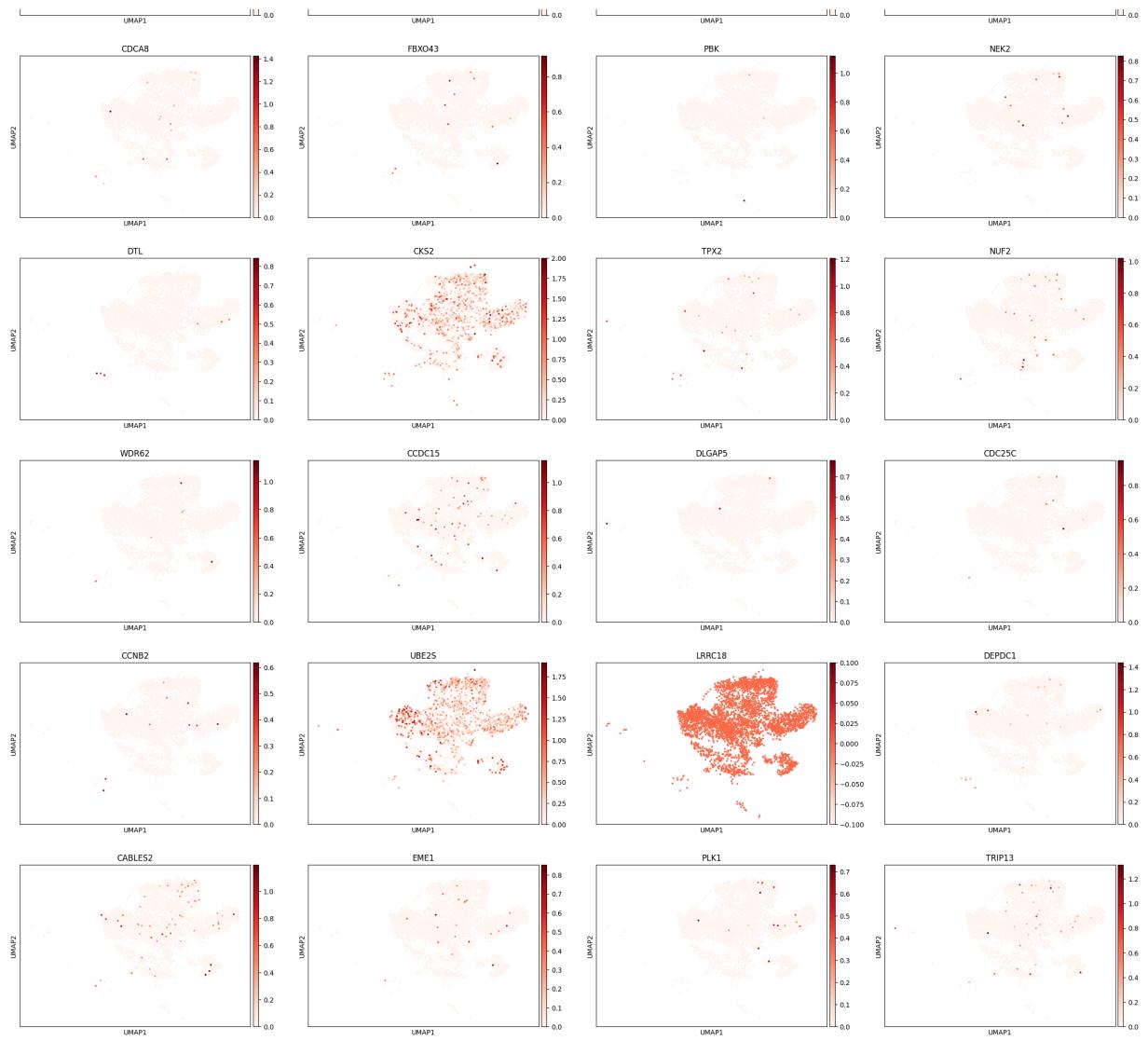


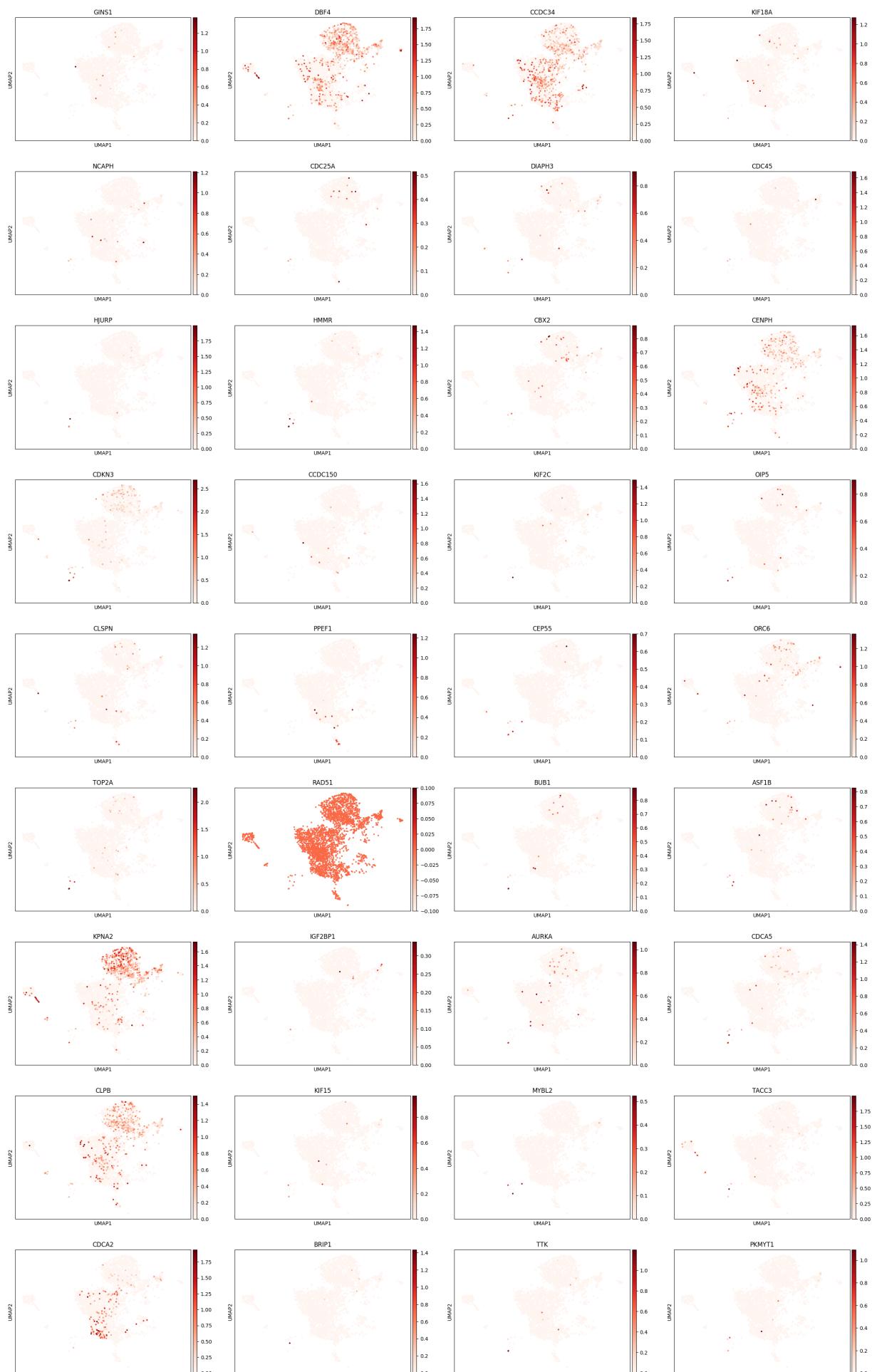


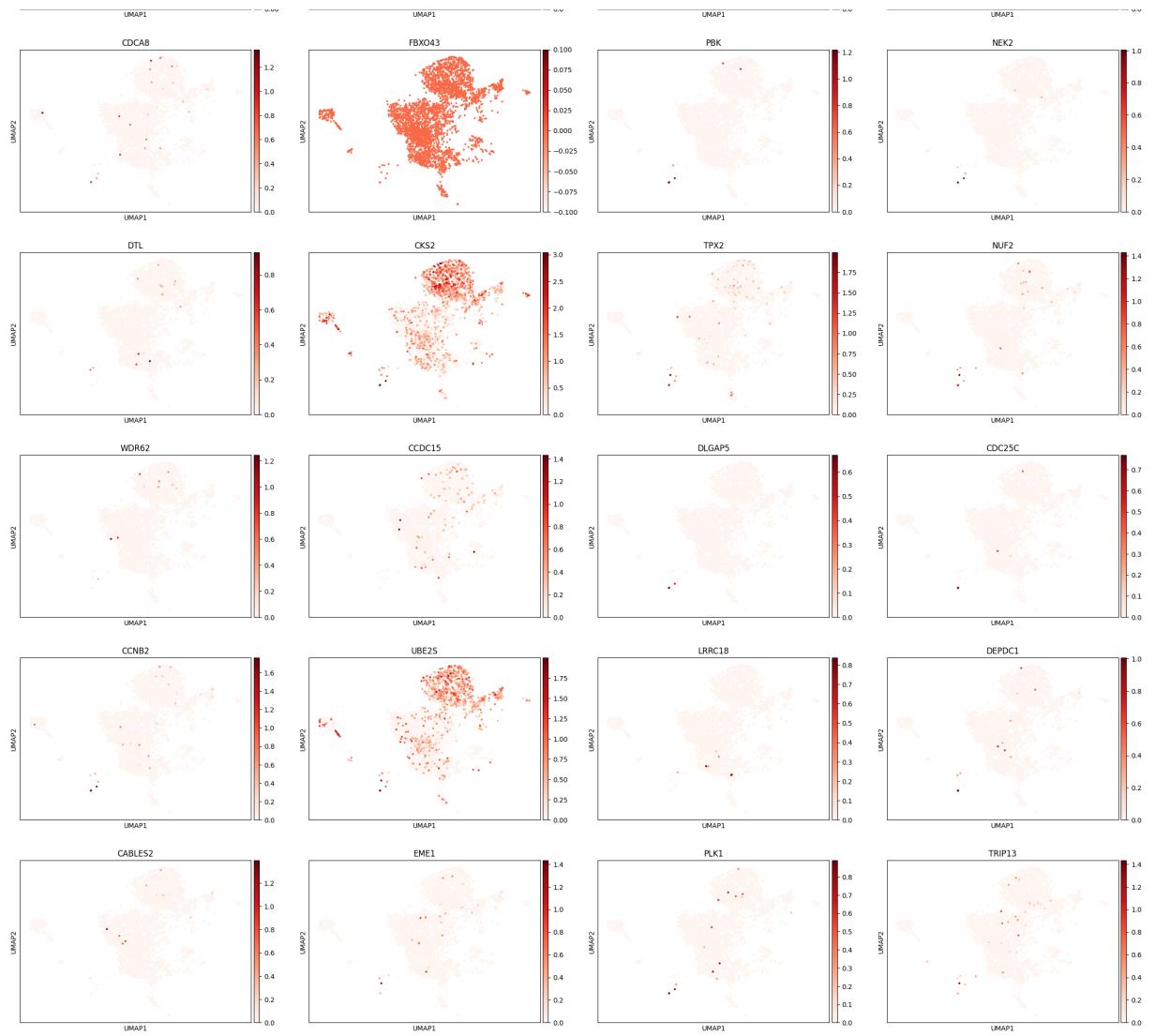


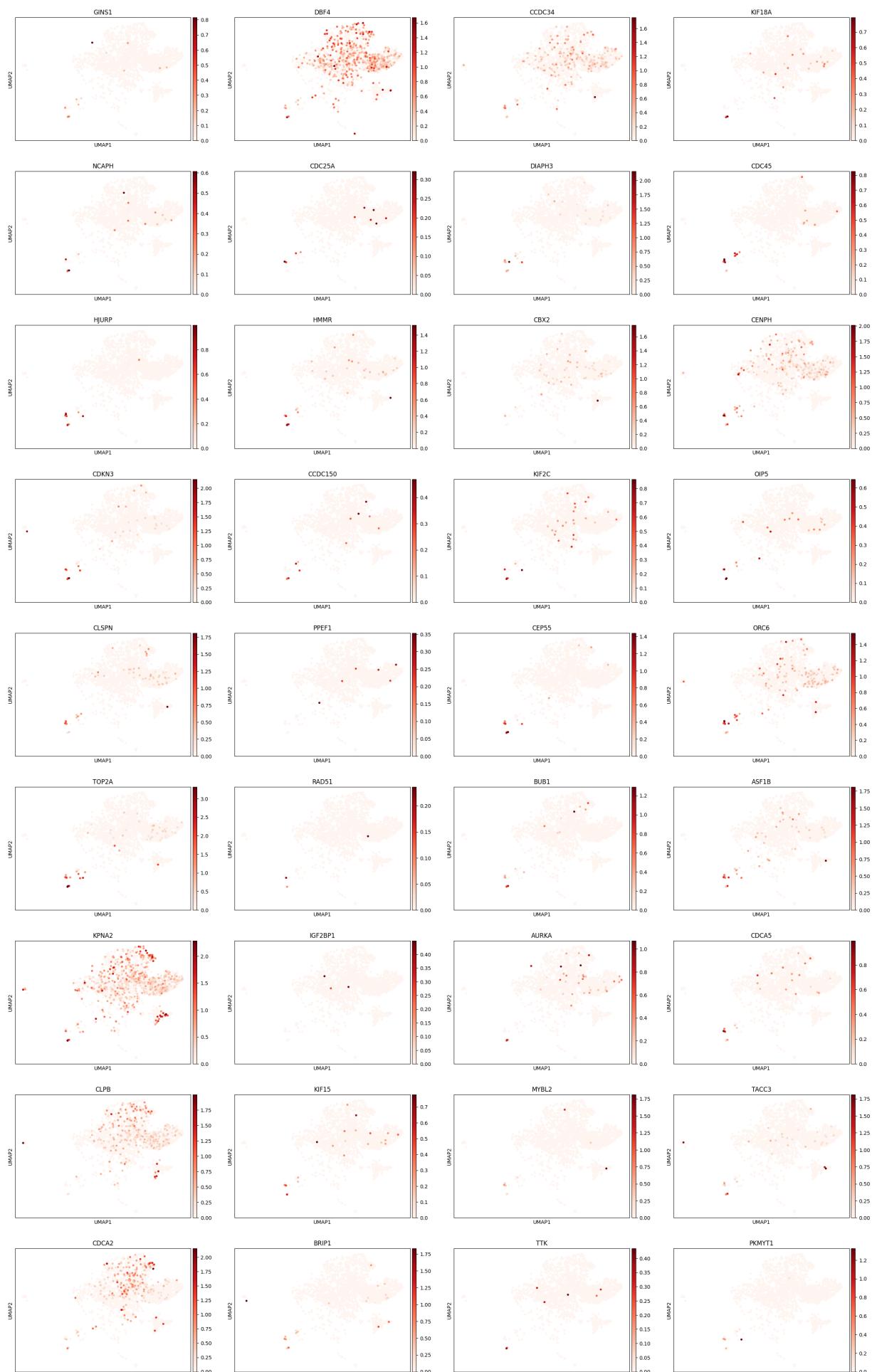


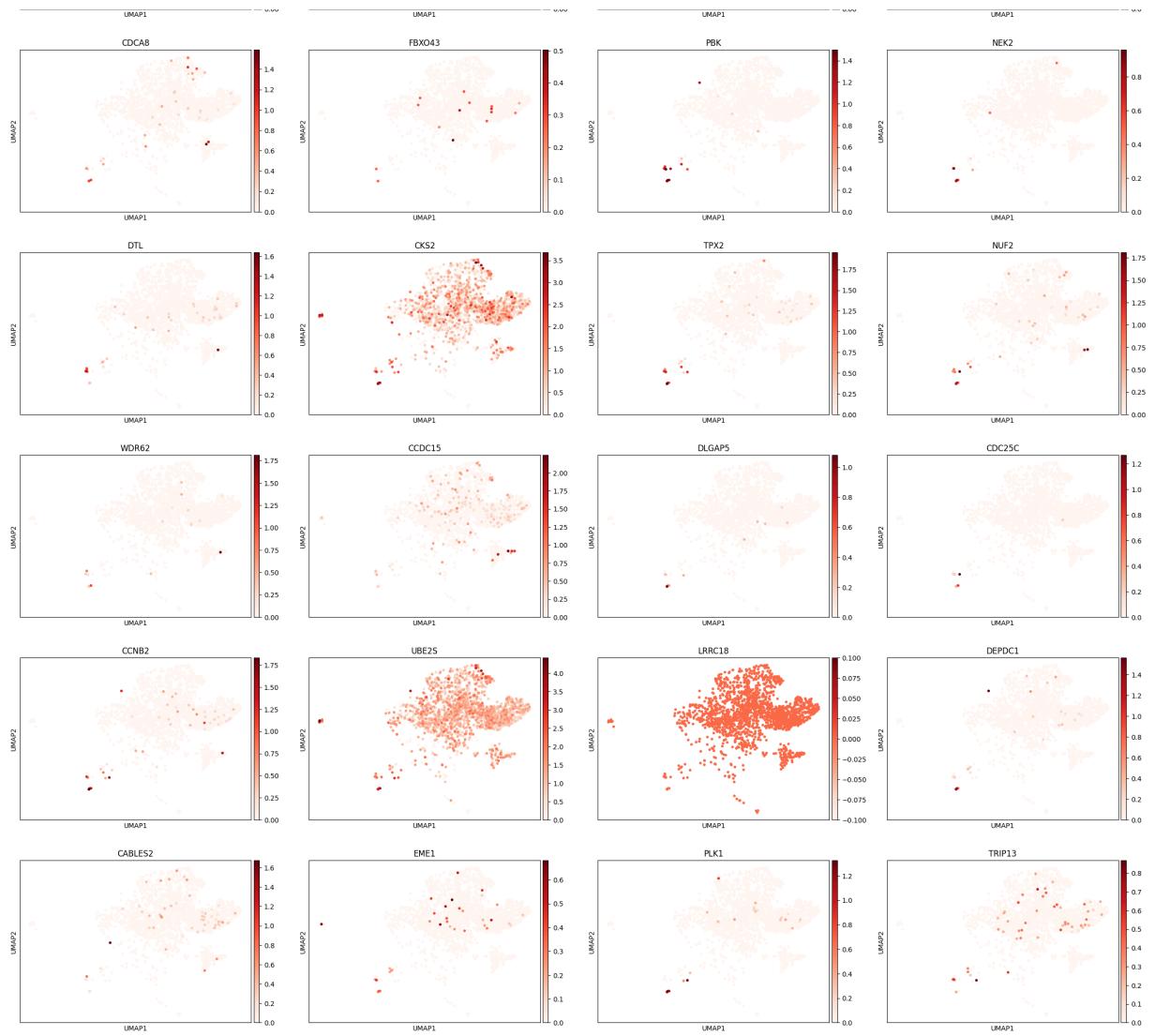


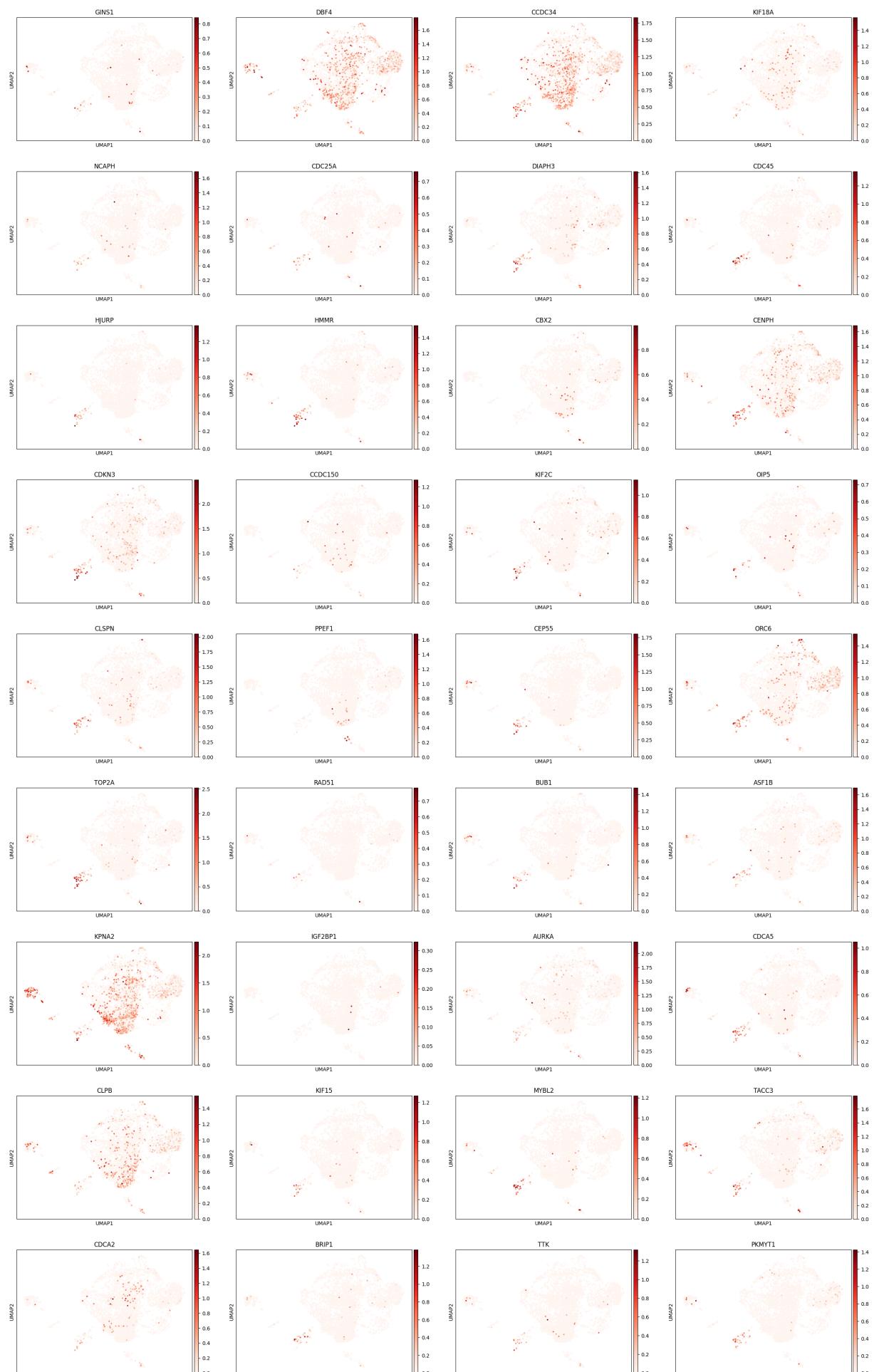


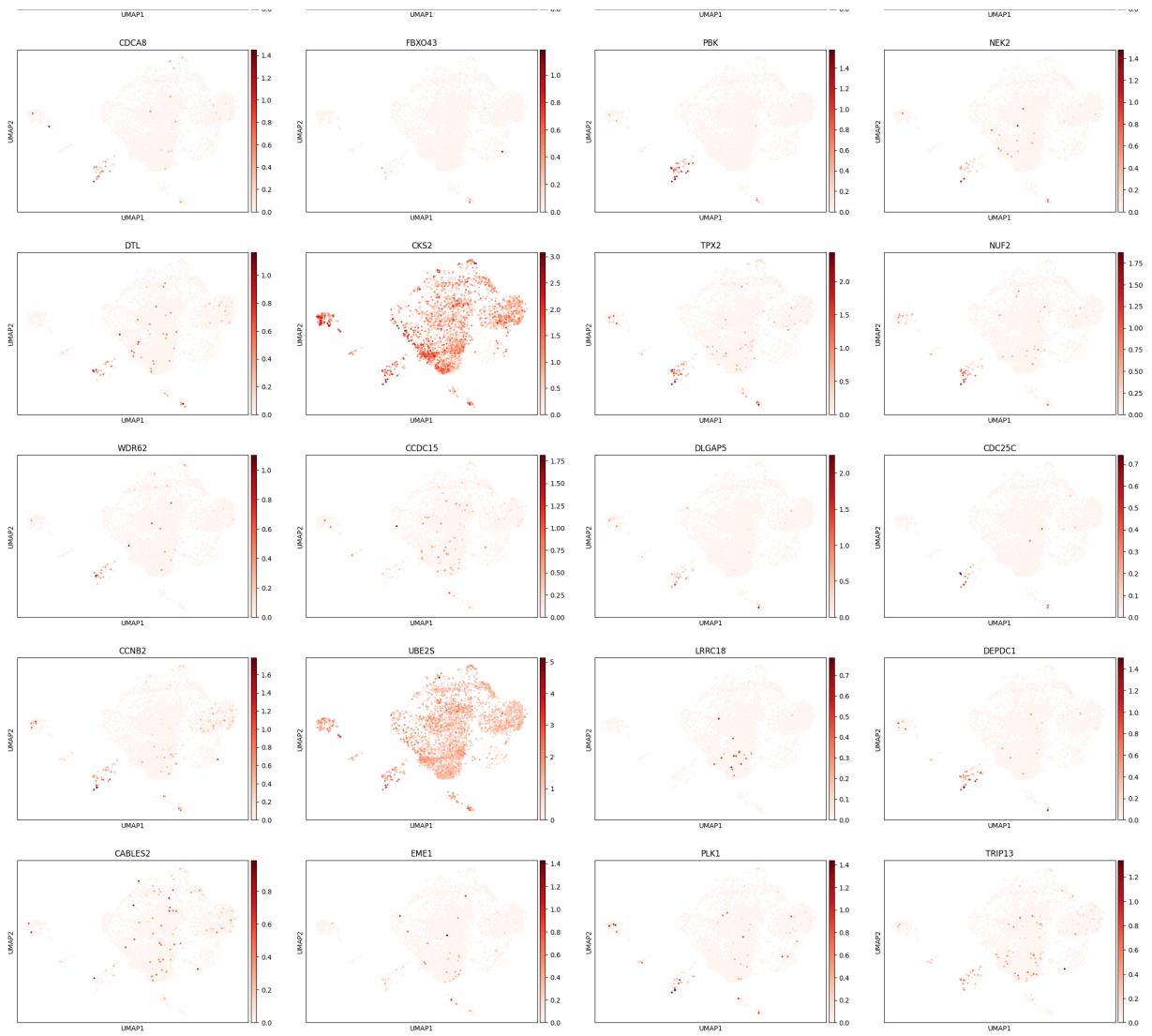








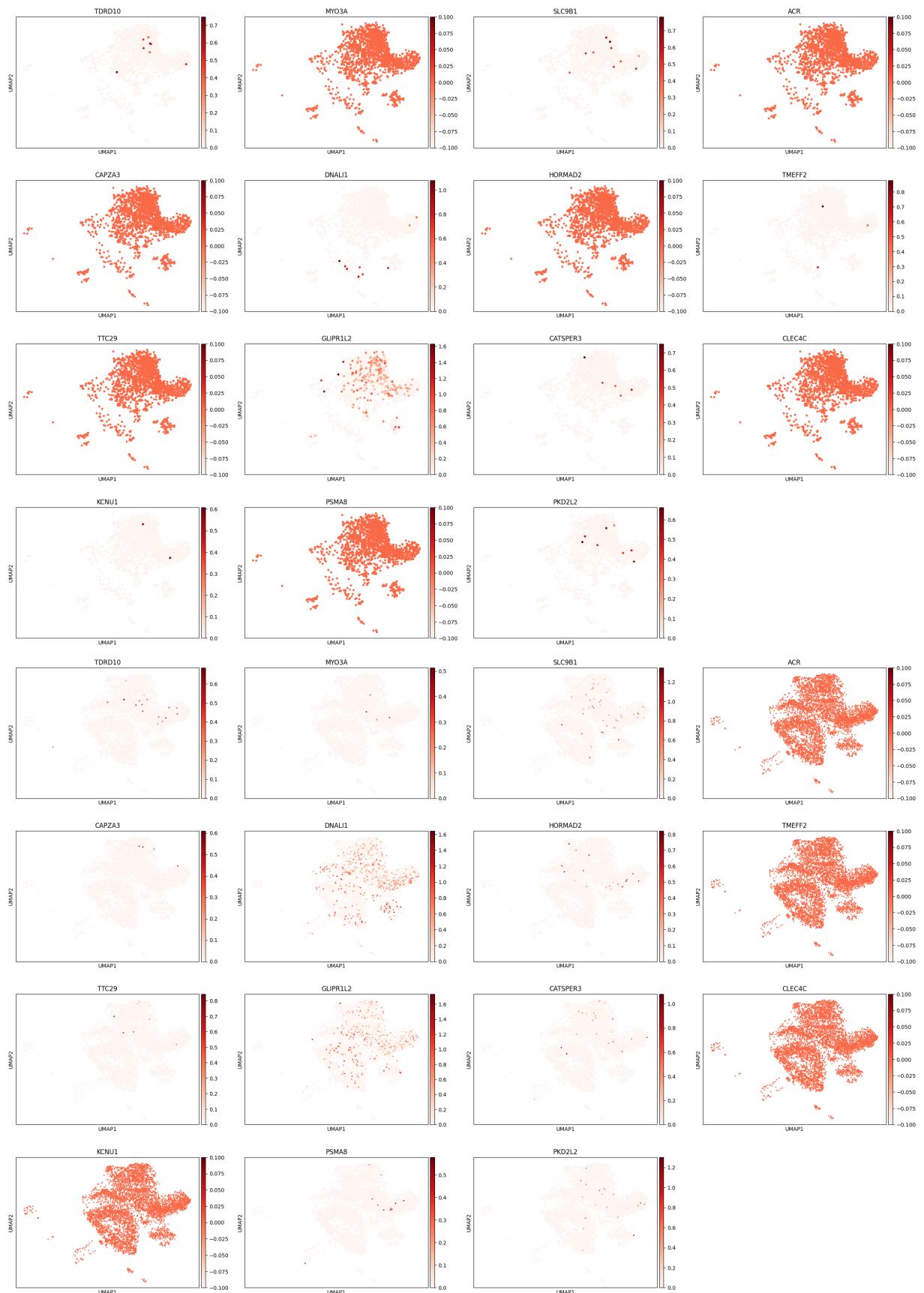


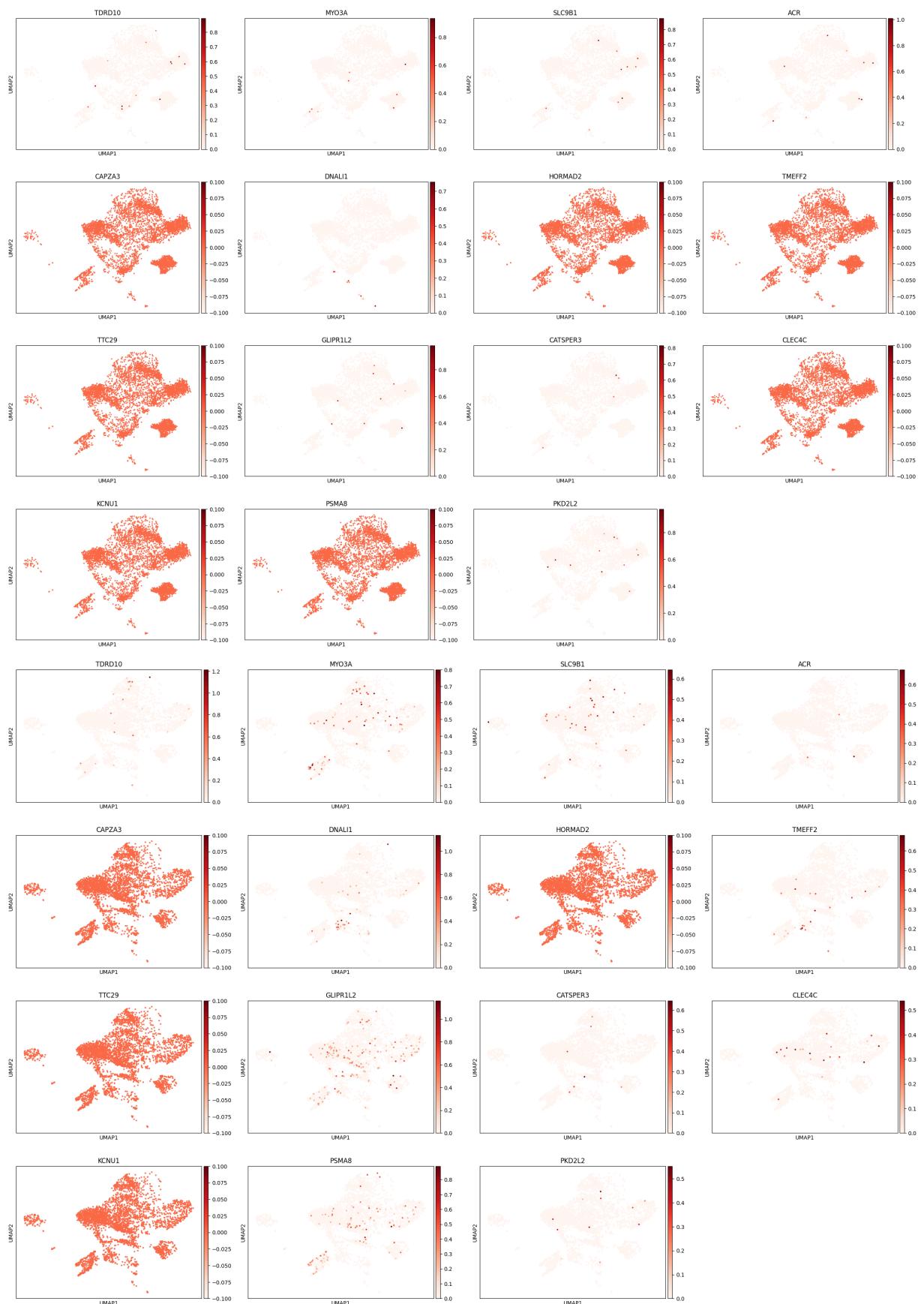


It seems that high grade patient have cells that expressed more CTA than low grade.

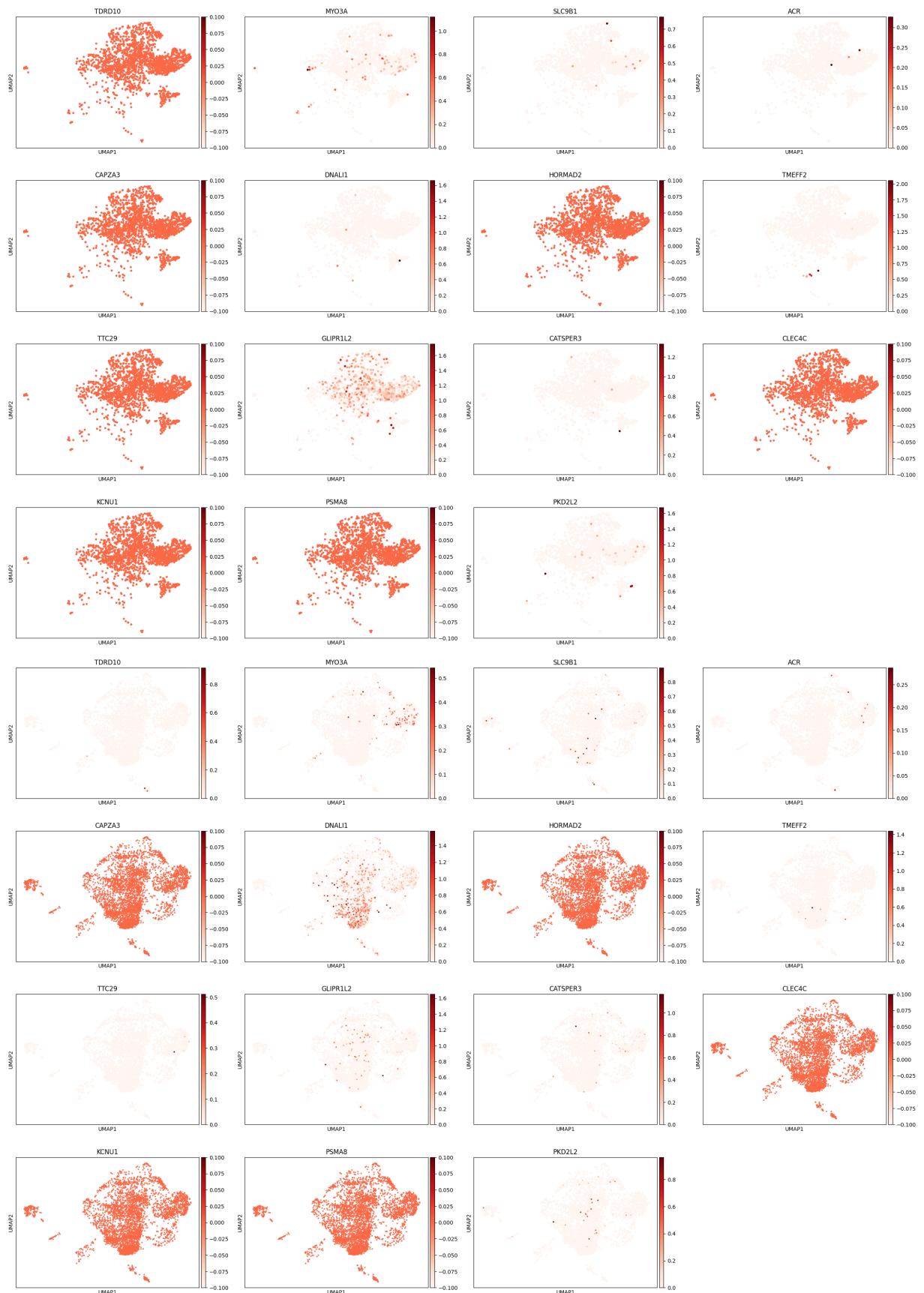
c- Cluster 3

```
In [28]: # Markers cluster 3
for dataset in l:
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]
    sc.pl.umap(adata_filtered, color = genes_clust_3 , color_map = plt.cm.R
```



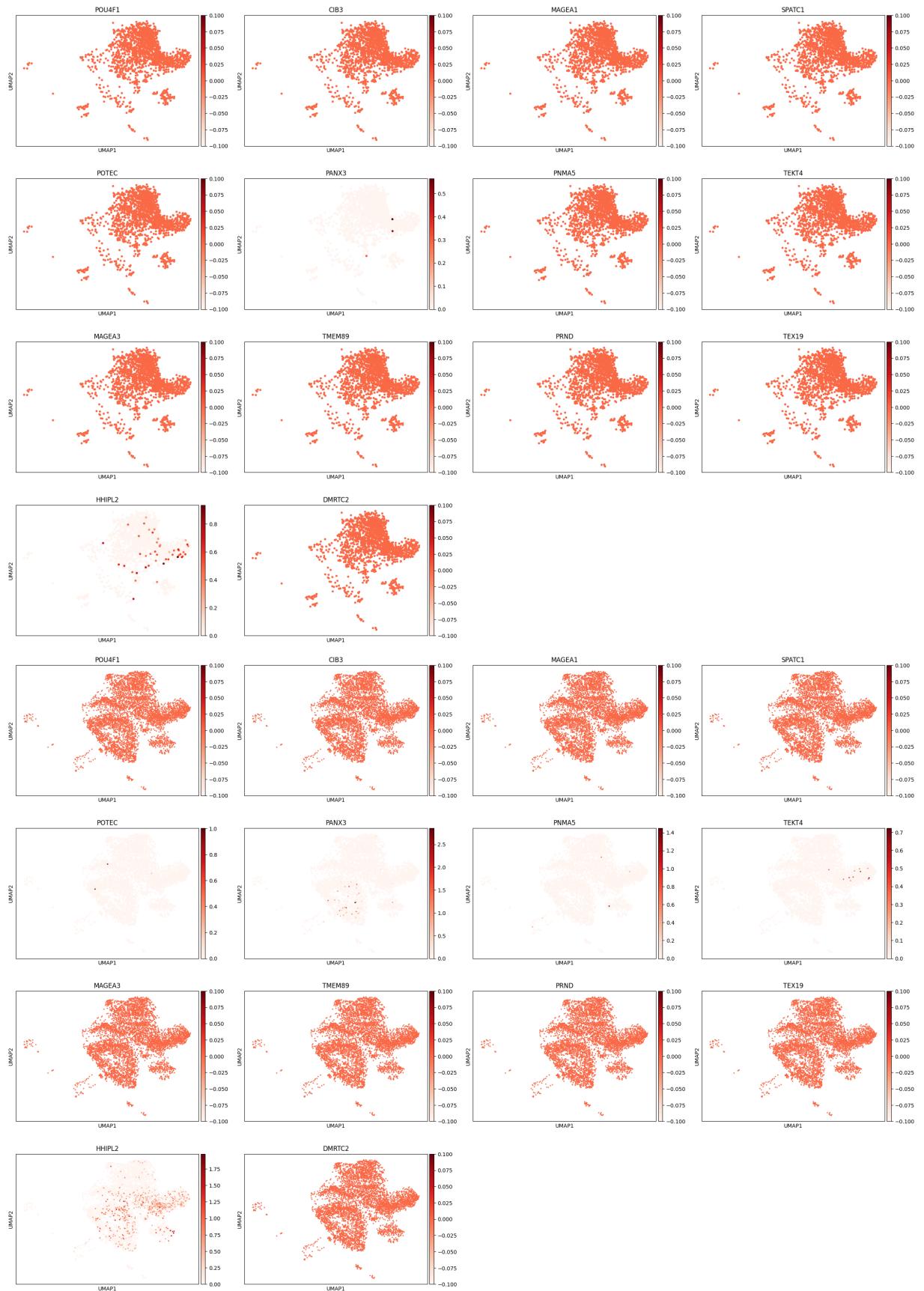


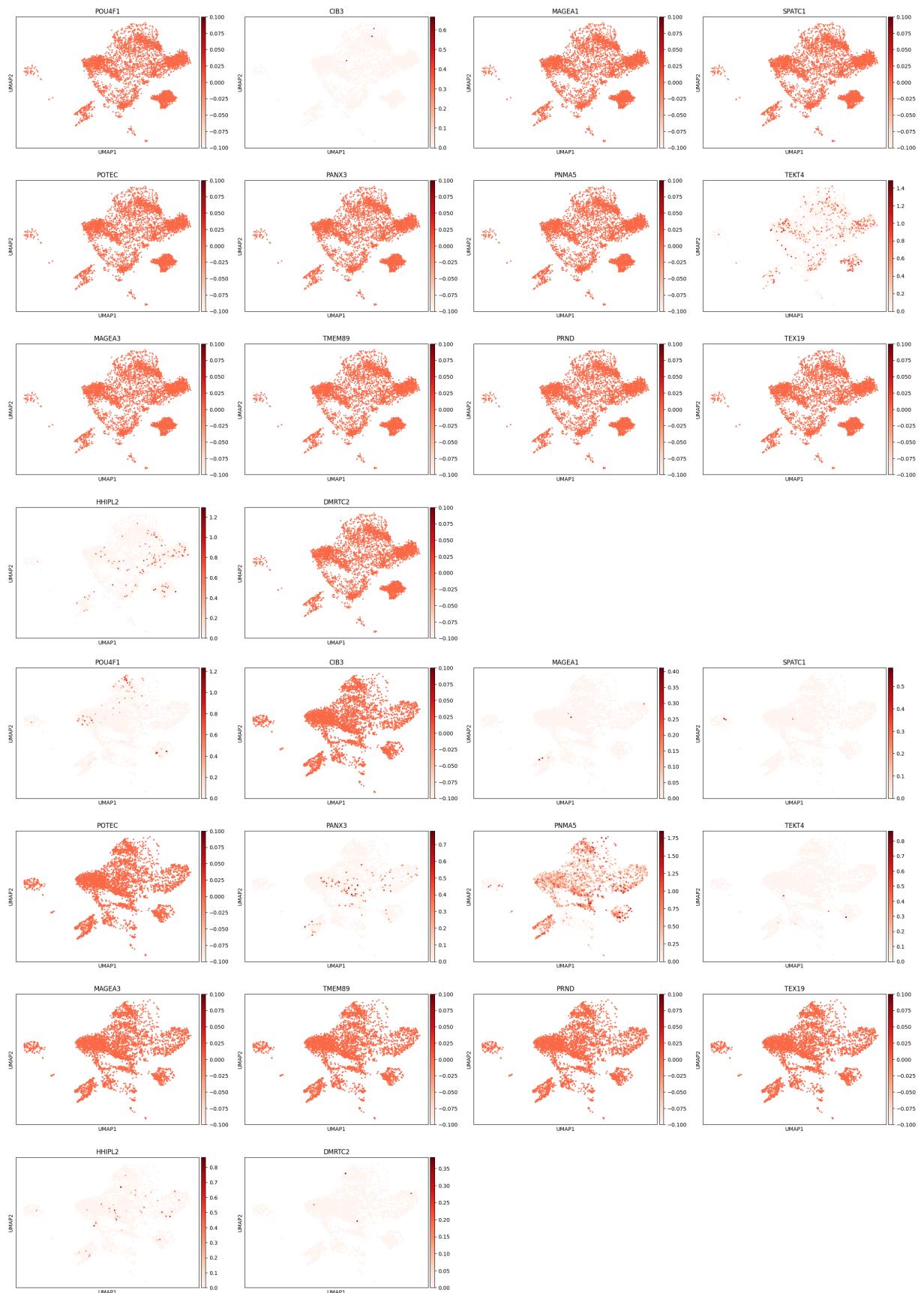




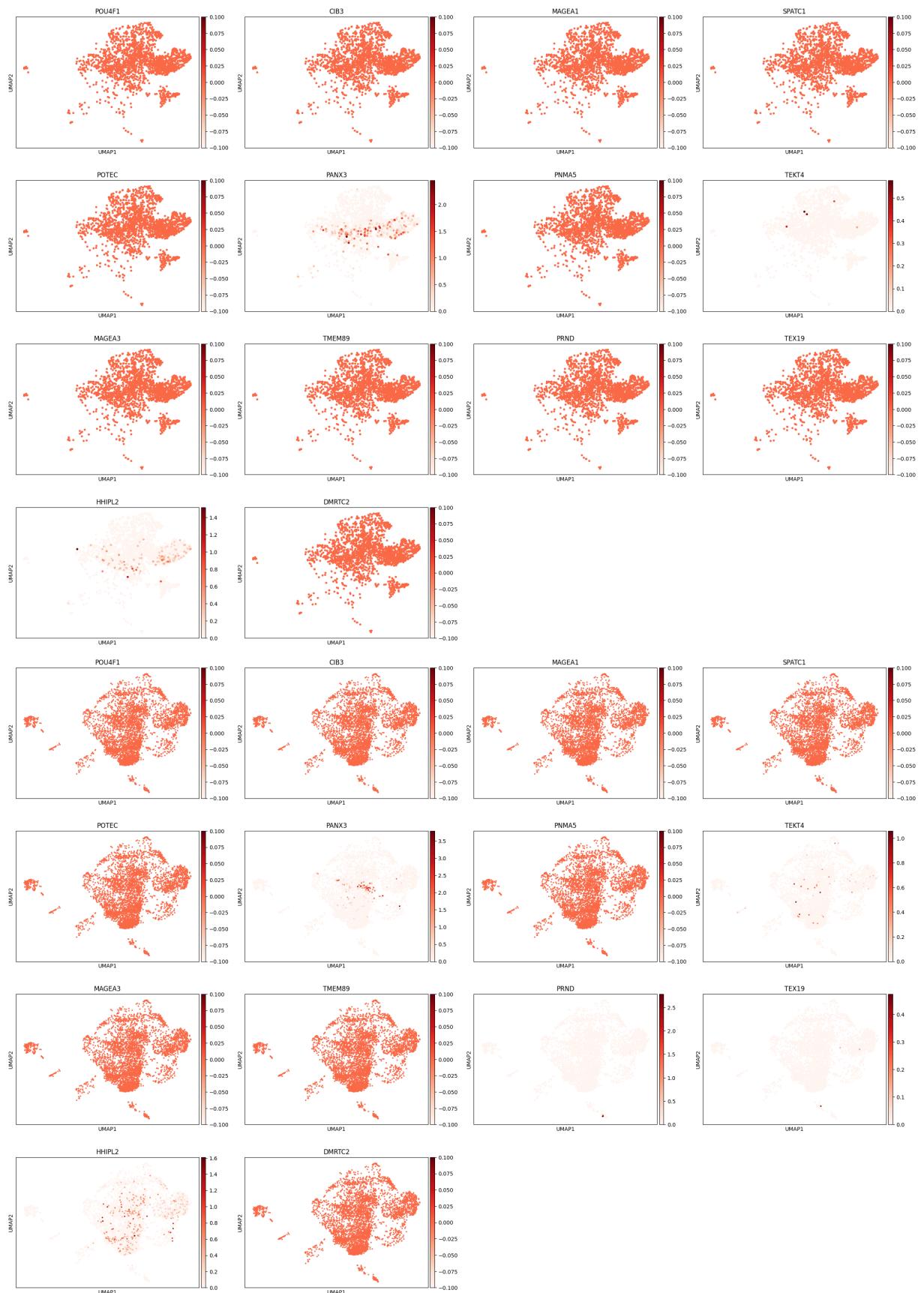
4) Low expressed CTAs in normal tissues and expressed in chondrosarcoma

```
In [29]: for dataset in l:
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]
    sc.pl.umap(adata_filtered, color = genes_scatter, color_map = plt.cm.Reds)
```





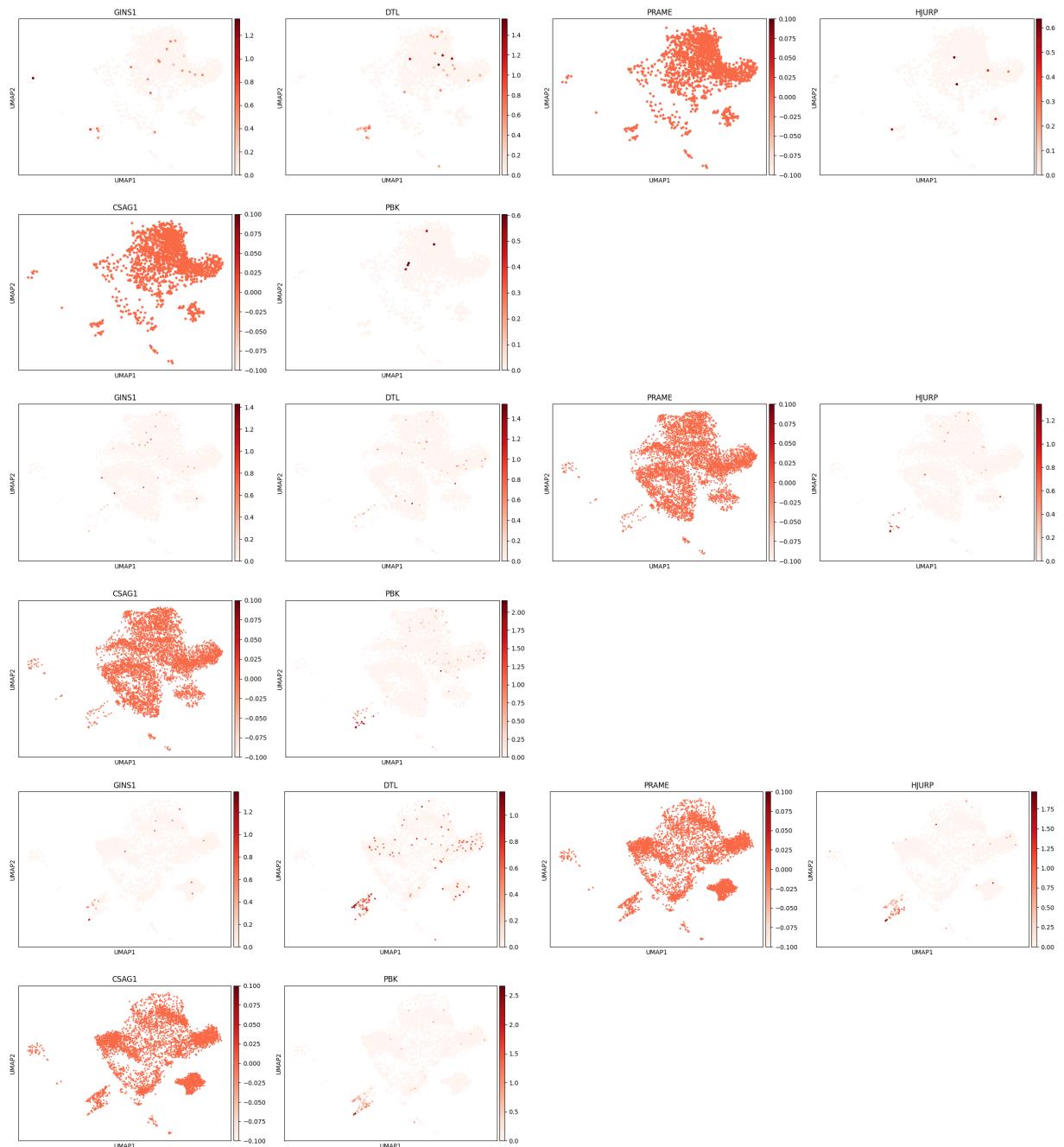


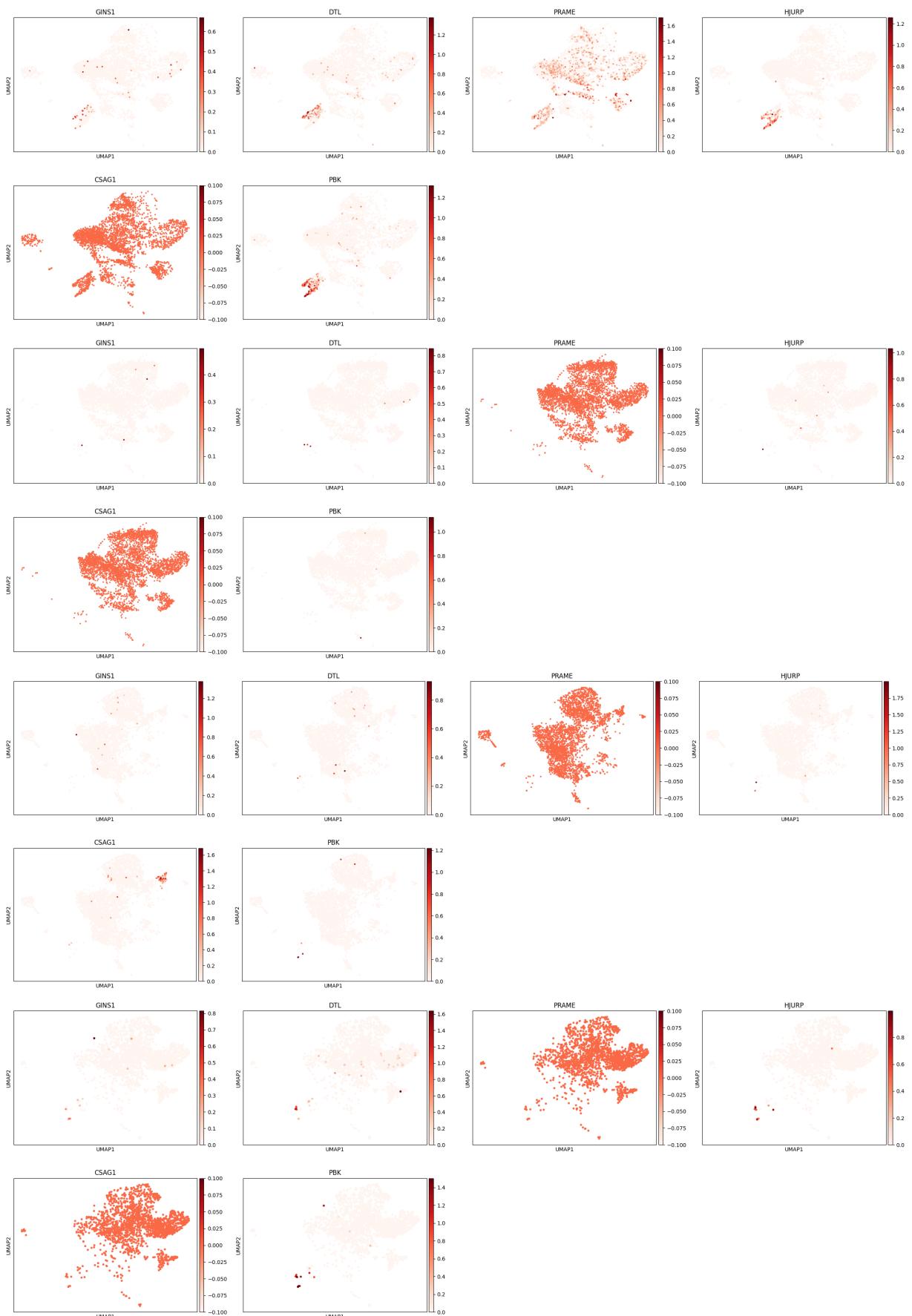


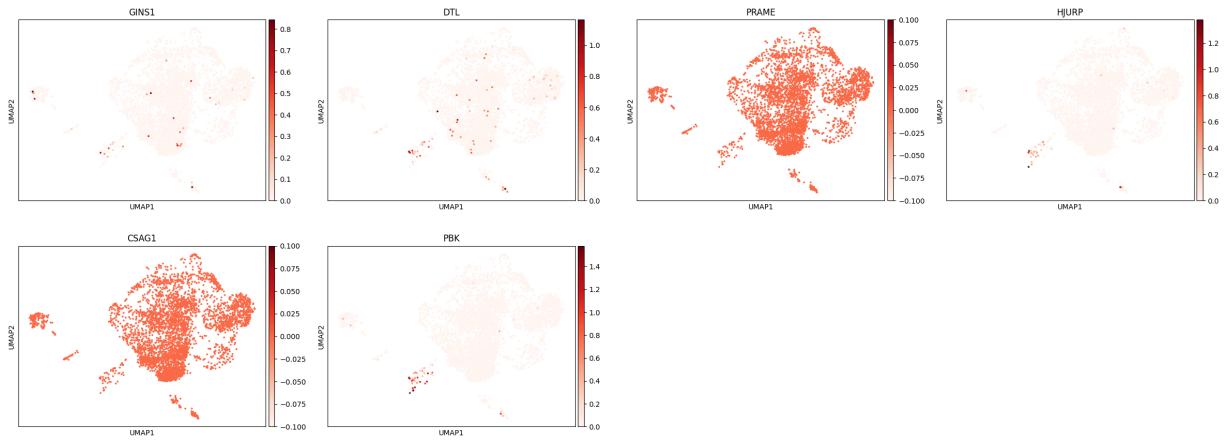
With this, PANX3 and TEKT4 seem to be the most expressed CTAs.

5) Selected CTA of interest

```
In [30]: for dataset in l:  
    adata_filtered = adata[adata.obs['dataset'] == dataset, :]  
    common_genes = list(set(['DTL', 'HJURP', 'GINS1', 'PBK', 'CT45A1', 'CSAG1'])  
    sc.pl.umap(adata_filtered, color = common_genes, color_map = plt.cm.Reds)
```







This notebook was converted with convert.ploomber.io