Multi-Modal Surgery Pipeline with TOTALVI

In [ ]:
```
import os
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)

import scanpy as sc
import anndata
import torch
import scarches as sca
import matplotlib.pyplot as plt
import numpy as np
import scvi as scv
import pandas as pd

sc.settings.set_figure_params(dpi=200, frameon=False)
sc.set_figure_params(dpi=200)
sc.set_figure_params(figsize=(4, 4))
torch.set_printoptions(precision=3, sci_mode=False, edgeitems=7)
```

```
WARNING:root:In order to use the mouse gastrulation seqFISH datsets, please
install squidpy (see https://github.com/scverse/squidpy).
WARNING:root:In order to use sagenet models, please install pytorch geometri
c (see https://pytorch-geometric.readthedocs.io) and
 captum (see https://github.com/pytorch/captum).
WARNING:root:multigrate is not installed. To use multigrate models, please i
nstall it first using "pip install multigrate".
```

Data loading and preprocessing

In [ ]:
```
condition_key = 'orig.ident'
cell_type_key = 'seurat_clusters'
target_conditions = [3228]

adata_all = sc.read('/Users/evelynschmidt/Protein_folzconversion_fig5.h5ad')
adata = adata_all.raw.to_adata()
```

In [ ]:
```
adata_3228 = adata[adata.obs['orig.ident'].isin([3228])].copy()
adata_3228.obs["batch"] = "3228"
adata_730 = adata[adata.obs['orig.ident'].isin([730])].copy()
adata_730.obs["batch"] = "730"
adata_451 = adata[adata.obs['orig.ident'].isin([451])].copy()
adata_451.obs["batch"] = "451"
```

In [ ]:
```
# create the reference
adata_ref = anndata.concat([adata_730,adata_451])

# separate the query
adata_query = adata_3228
```

```
# put matrix of zeros for protein expression (considered missing)
pro_exp = adata_ref.obsm["protein_expression"]
data = np.zeros((adata_query.n_obs, pro_exp.shape[1]))
adata_query.obsm["protein_expression"] = pd.DataFrame(columns=pro_exp.columr
```

In [ ]:
```
adata_query.obsm["protein_expression"]
```

Out[ ]:

	CD38ADT	CD314ADT	HLA-DRADT	CD62LADT	CD45ROADT
3228_AAAGTAGAGCTACCGC-1	0.0	0.0	0.0	0.0	0.0
3228_AAATGCCCAGATGGGT-1	0.0	0.0	0.0	0.0	0.0
3228_AAGACCTCATTCGACA-1	0.0	0.0	0.0	0.0	0.0
3228_ACACTGATCAGGTTCA-1	0.0	0.0	0.0	0.0	0.0
3228_ACGAGGAGTTACGCGC-1	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...
3228_TTTGTCATCTTGTCAT-1	0.0	0.0	0.0	0.0	0.0
3228_ACTGAGTGTTACGACT-1	0.0	0.0	0.0	0.0	0.0
3228_CAGAATCAGCACCGTC-1	0.0	0.0	0.0	0.0	0.0
3228_GCGCGATTCACGGTTA-1	0.0	0.0	0.0	0.0	0.0
3228_TCGGGACCACGCATCG-1	0.0	0.0	0.0	0.0	0.0

7814 rows × 28 columns

In [ ]:
```
adata_full = anndata.concat([adata_ref, adata_query])

sc.pp.highly_variable_genes(
    adata_full,
    n_top_genes=4000,
    flavor="seurat_v3",
    batch_key="batch",
    subset=True,
)

adata_ref = adata_full[np.logical_or(adata_full.obs.batch == "451", adata_fu
adata_query = adata_full[adata_full.obs.batch == "3228"].copy()
```

In [ ]:
```
adata_full
```

Out[ ]:
```
AnnData object with n_obs × n_vars = 20046 × 4000
    obs: 'orig.ident', 'nCount_RNA', 'nFeature_RNA', 'species', 'nCount_AD
T', 'nFeature_ADT', 'nCount_HTO', 'nFeature_HTO', 'percent.mt', 'HTO_maxI
D', 'HTO_secondID', 'HTO_margin', 'HTO_classification', 'HTO_classificatio
n.global', 'hash.ID', 'HTO_classification_species', 'Cell_Types', 'S.Scor
e', 'G2M.Score', 'Phase', 'RNA_snn_res.0.2', 'seurat_clusters', 'unfilt_clu
sters', 'RNA.weight', 'ADT_denoised_iso_quant.weight', 'batch'
    var: 'highly_variable', 'highly_variable_rank', 'means', 'variances',
'variances_norm', 'highly_variable_nbatches'
    uns: 'hvg'
    obsm: 'protein_expression'
```

In [ ]:
```
adata_ref
```

Out[ ]:
```
AnnData object with n_obs × n_vars = 12232 × 4000
    obs: 'orig.ident', 'nCount_RNA', 'nFeature_RNA', 'species', 'nCount_AD
T', 'nFeature_ADT', 'nCount_HTO', 'nFeature_HTO', 'percent.mt', 'HTO_maxI
D', 'HTO_secondID', 'HTO_margin', 'HTO_classification', 'HTO_classificatio
n.global', 'hash.ID', 'HTO_classification_species', 'Cell_Types', 'S.Scor
e', 'G2M.Score', 'Phase', 'RNA_snn_res.0.2', 'seurat_clusters', 'unfilt_clu
sters', 'RNA.weight', 'ADT_denoised_iso_quant.weight', 'batch'
    var: 'highly_variable', 'highly_variable_rank', 'means', 'variances',
'variances_norm', 'highly_variable_nbatches'
    uns: 'hvg'
    obsm: 'protein_expression'
```

In [ ]:
```
adata_ref.X
```

Out[ ]:
```
array([[  0.,   0.,   0.,   0.,   0.,   0.,   0., ...,  43.,  19.,   5.,
          8.,   7.,  14.,   0.],
       [  0.,   0.,   1.,   0.,   0.,   0.,   0., ...,  21.,   8.,   1.,
          4.,   1.,   7.,   0.],
       [  0.,   0.,   3.,   0.,   0.,   0.,   0., ...,  70.,  45.,   6.,
         42.,  12.,   6.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   1., ...,  22.,  13.,   1.,
          6.,   5.,   2.,   0.],
       [  0.,   0.,   1.,   0.,   0.,   0.,   0., ...,  18.,  14.,   4.,
         14.,  11.,   5.,   0.],
       [  0.,   0.,   1.,   0.,   0.,   0.,   0., ...,  11.,  10.,   1.,
          9.,   3.,   0.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   1.,   0., ...,  23.,  13.,   2.,
          9.,   3.,   3.,   0.],
       ...,
       [  0.,   0.,   0.,   0.,   0.,   0.,   0., ...,  60.,   7.,  16.,
         12.,   2.,   2.,   0.],
       [  0.,   0.,   1.,   0.,   0.,   0.,   0., ..., 106.,  15.,  22.,
         60.,   6.,   1.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0., ..., 125.,   9.,  15.,
         43.,   6.,   1.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0., ...,  33.,   6.,   4.,
         12.,   7.,   1.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0., ..., 260.,  55.,  46.,
        106.,  12.,   5.,   0.],
       [  0.,   0.,   0.,   0.,   0.,   0.,   0., ..., 101.,  13.,  22.,
         44.,   7.,   1.,   0.],
       [  0.,   0.,   3.,   0.,   0.,   0.,   0., ...,  56.,  11.,  18.,
         17.,   4.,   3.,   0.]], dtype=float32)
```

In [ ]:
```
adata_query
```

Out[ ]:
```
AnnData object with n_obs × n_vars = 7814 × 4000
    obs: 'orig.ident', 'nCount_RNA', 'nFeature_RNA', 'species', 'nCount_AD
T', 'nFeature_ADT', 'nCount_HTO', 'nFeature_HTO', 'percent.mt', 'HTO_maxI
D', 'HTO_secondID', 'HTO_margin', 'HTO_classification', 'HTO_classificatio
n.global', 'hash.ID', 'HTO_classification_species', 'Cell_Types', 'S.Scor
e', 'G2M.Score', 'Phase', 'RNA_snn_res.0.2', 'seurat_clusters', 'unfilt_clu
sters', 'RNA.weight', 'ADT_denoised_iso_quant.weight', 'batch'
    var: 'highly_variable', 'highly_variable_rank', 'means', 'variances',
'variances_norm', 'highly_variable_nbatches'
    uns: 'hvg'
    obsm: 'protein_expression'
```

Create TOTALVI model and train it on CITE-seq reference dataset

In [ ]:
```
sca.models.TOTALVI.setup_anndata(adata_ref, batch_key="batch", protein_expre
```

```
INFO     Using column names from columns of adata.obsm['protein_expression']
INFO     Found batches with missing protein expression
```

Train model

script:
/Volumes/mgriffit/Active/griffithlab/gc2596/e.schmidt/fig4_foltz/conversion/TOTALVI/Foltz_t

In [ ]:
```
vae_ref = sca.models.TOTALVI.load("/Volumes/mgriffit/Active/griffithlab/gc25
```

```
INFO     File
         /Volumes/mgriffit/Active/griffithlab/gc2596/e.schmidt/fig4_foltz/co
nversion/saved_model_subsetting/model.p
         t already downloaded
INFO     Found batches with missing protein expression
INFO     Computing empirical prior initialization for protein background.
```

Save Latent representation and visualize RNA data

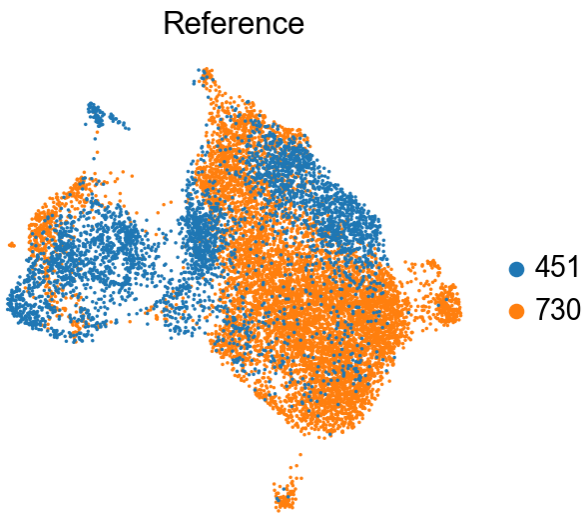
In [ ]:
```
adata_ref.obsm["X_totalVI"] = vae_ref.get_latent_representation()
sc.pp.neighbors(adata_ref, use_rep="X_totalVI")
sc.tl.umap(adata_ref, min_dist=0.4)
```

In [ ]:
```
adata_ref.obsm["X_totalVI"]
```

Out[ ]:
```
array([[ 0.01,  1.04, -0.42, -0.04, -0.19, -0.06, -0.16, ..., -0.77,
         0.18,  0.34, -1.57,  0.65,  0.04,  0.49],
       [-0.9 ,  0.82,  0.54, -0.05, -0.16, -0.11,  0.04, ..., -1.06,
         0.02, -0.79,  0.69,  0.6 ,  0.05, -0.09],
       [ 0.34, -0.34, -0.08, -0.04,  0.96, -0.1 ,  0.78, ...,  1.27,
        -0.54, -0.57,  0.56,  1.09,  0.06,  0.21],
       [ 0.63, -1.11, -0.9 , -0.04,  0.41, -0.07,  0.41, ..., -0.67,
         0.07, -0.99,  0.43,  0.35,  0.06, -0.51],
       [-0.32,  1.54,  0.29, -0.02,  0.81, -0.09, -0.97, ...,  0.67,
        -0.49,  0.67, -0.46, -0.83,  0.06, -0.71],
       [ 1.08,  0.05, -0.54, -0.06, -0.33, -0.08, -0.15, ..., -0.13,
        -0.12, -0.28,  1.27,  0.49,  0.05, -0.21],
       [-0.26, -0.68, -0.82, -0.04,  0.18, -0.04,  0.09, ..., -0.16,
         0.55, -0.63,  0.09,  0.24,  0.06, -0.4 ],
       ...,
       [ 0.64,  0.73,  0.26, -0.04,  0.45, -0.04,  0.57, ..., -1.1 ,
         1.97,  0.68,  0.01,  0.19,  0.07, -1.28],
       [-1.73,  0.38, -0.59, -0.02,  0.55, -0.01,  1.74, ...,  1.56,
         1.55, -0.06,  0.32, -0.02,  0.06, -1.11],
       [-0.49,  0.38,  0.42, -0.03,  0.2 , -0.13,  0.85, ...,  0.27,
        -0.97, -1.22,  0.47, -1.06,  0.06, -0.29],
       [ 0.21,  0.21, -0.9 , -0.06, -0.24, -0.04,  1.25, ..., -1.06,
         1.92,  0.45,  0.27,  0.06,  0.09, -0.83],
       [-1.85,  0.71,  1.56, -0.04,  1.33, -0.1 ,  1.46, ...,  0.96,
        -0.13, -0.17, -1.33,  0.39,  0.07,  1.44],
       [-0.29, -0.6 ,  0.43, -0.04,  1.77, -0.14,  0.76, ...,  0.83,
        -0.38, -0.69, -0.95,  0.75,  0.09,  1.44],
       [ 0.96, -0.48,  0.3 , -0.05, -0.66, -0.05,  0.53, ...,  1.2 ,
        -0.16,  1.04,  0.12,  0.63,  0.03,  0.72]], dtype=float32)
```

In [ ]:
```
sc.pl.umap(
    adata_ref,
    color=["batch"],
    frameon=False,
```

```
    ncols=1,
    title="Reference",
    save="_reference_test1.png"
)
```



Perform surgery on reference model and train on query dataset without protein data

Train Query Model

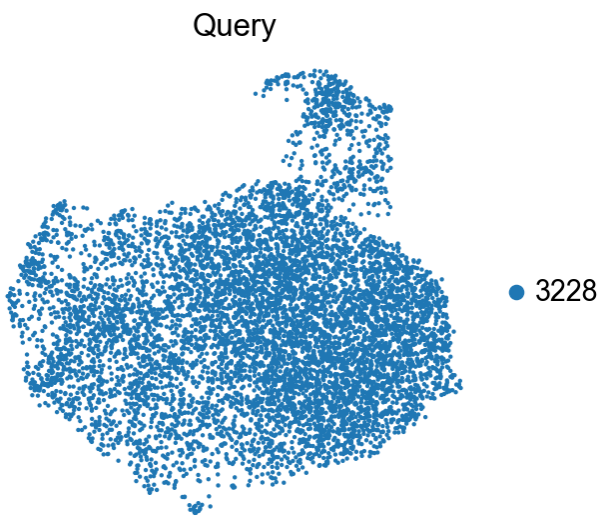
script:

/Volumes/mgriffit/Active/griffithlab/gc2596/e.schmidt/fig4_foltz/conversion/TOTALVI/Foltz_t

In [ ]:
```
vae_q = sca.models.TOTALVI.load("/Volumes/mgriffit/Active/griffithlab/gc2596
```

```
INFO     File /Volumes/mgriffit/Active/griffithlab/gc2596/e.schmidt/fig4_fol
tz/conversion/query_model/model.pt
         already downloaded
INFO     Found batches with missing protein expression
INFO     Computing empirical prior initialization for protein background.
```

In [ ]:
```
sc.pl.umap(
    adata_query,
    color=["batch"],
    frameon=False,
    ncols=1,
    title="Query",
    save="_query.png"
)
```

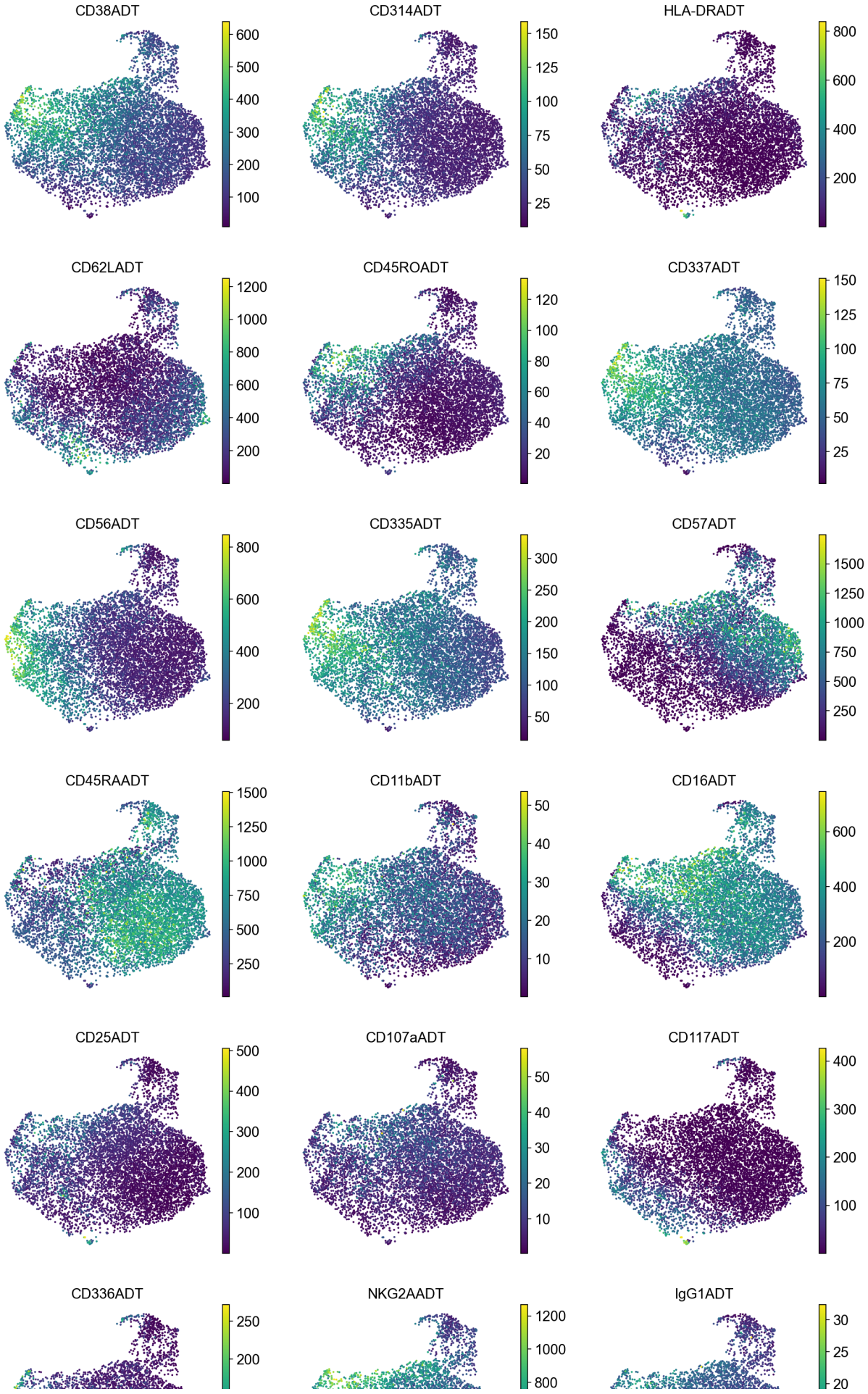


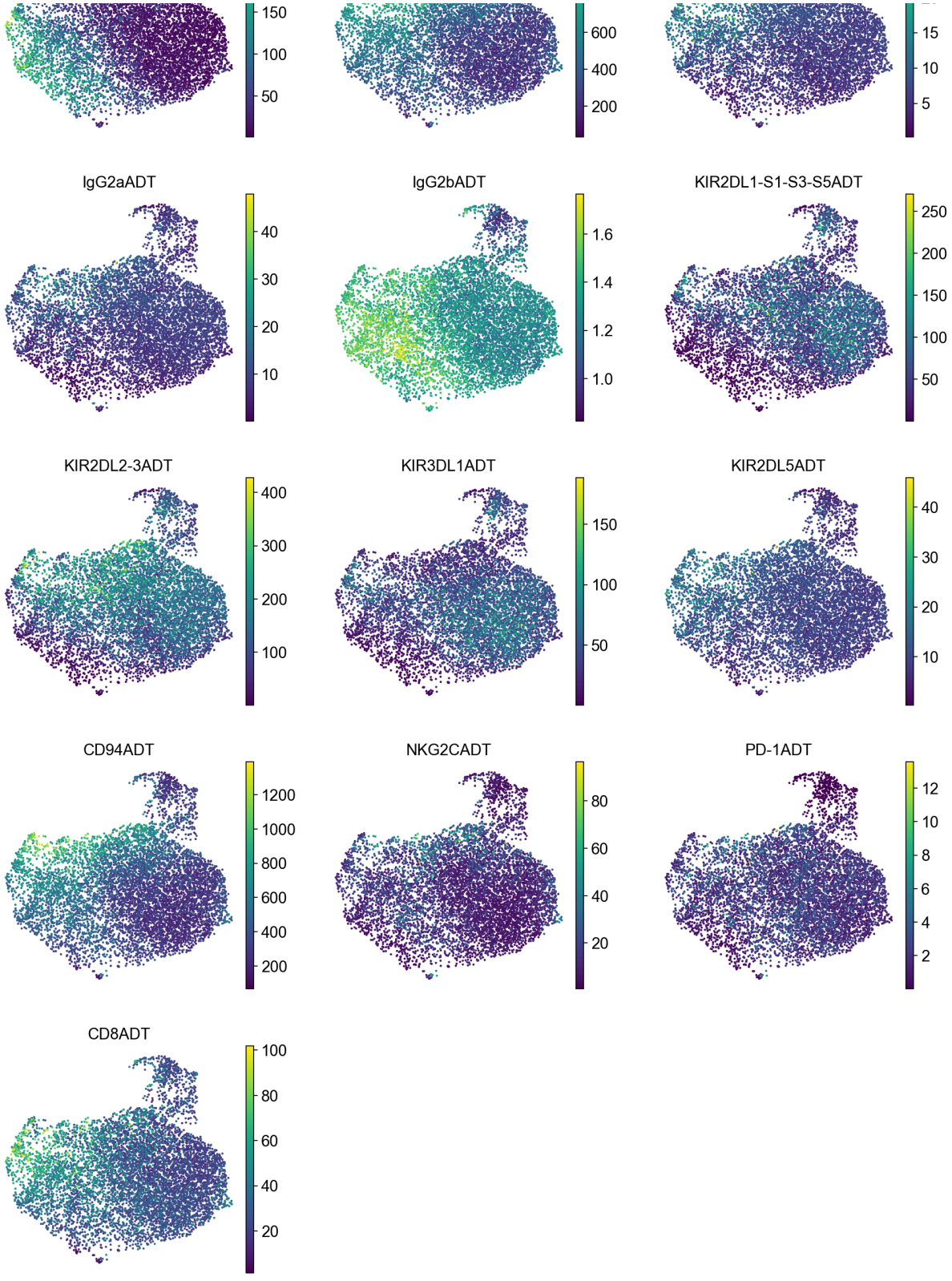
In [ ]:
```
adata_query.obsm["X_totalVI"] = vae_q.get_latent_representation()
sc.pp.neighbors(adata_query, use_rep="X_totalVI")
sc.tl.umap(adata_query, min_dist=0.4)
```

In [ ]:
```
_, imputed_proteins = vae_q.get_normalized_expression(
    adata_query,
    n_samples=25,
    return_mean=True,
    transform_batch=["730", "451", "3228"],
)
```

In [ ]:
```
adata_query.obs = pd.concat([adata_query.obs, imputed_proteins], axis=1)

sc.pl.umap(
    adata_query,
    color=imputed_proteins.columns,
    frameon=False,
    ncols=3,
)
```


CD38ADT
600
500
400
300
200
100
CD314ADT
150
125
100
75
50
25
HLA-DRADT
800
600
400
200
CD62LADT
1200
1000
800
600
400
200
CD45ROADT
120
100
80
60
40
20
CD337ADT
150
125
100
75
50
25
CD56ADT
800
600
400
200
CD335ADT
300
250
200
150
100
50
CD57ADT
1500
1250
1000
750
500
250
CD45RAADT
1500
1250
1000
750
500
250
CD11bADT
50
40
30
20
10
CD16ADT
600
400
200
CD25ADT
500
400
300
200
100
CD107aADT
50
40
30
20
10
CD117ADT
400
300
200
100
CD336ADT
250
200
NKG2AADT
1200
1000
800
IgG1ADT
30
25
20



Get latent representation of reference + query dataset and compute UMAP

In [ ]:
```
adata_full_new = adata_query.concatenate(adata_ref, batch_key="none")
```

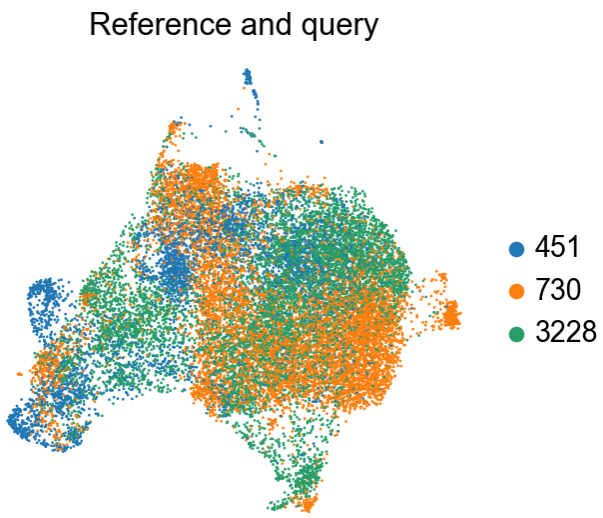
In [ ]:
```
adata_full_new.obsm["X_totalVI"] = vae_q.get_latent_representation(adata_ful
sc.pp.neighbors(adata_full_new, use_rep="X_totalVI")
sc.tl.umap(adata_full_new, min_dist=0.3)
```

```
INFO     Input AnnData not setup with scvi-tools. attempting to transfer Ann
Data setup
INFO     Found batches with missing protein expression
```

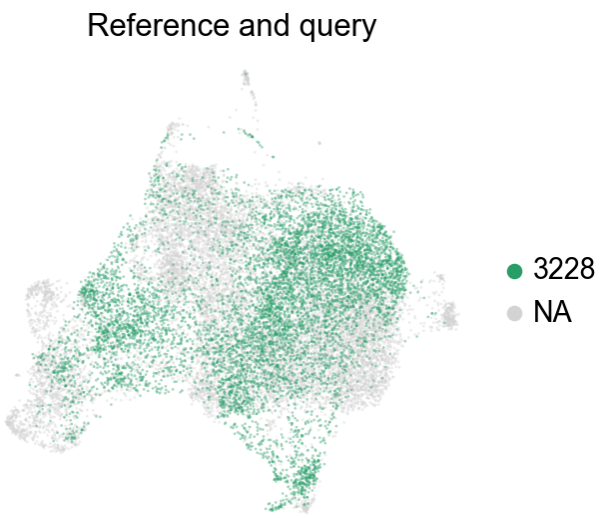
In [ ]:
```
_, imputed_proteins_all = vae_q.get_normalized_expression(
    adata_full_new,
    n_samples=25,
    return_mean=True,
    transform_batch=["730", "451", "3228"],
)

for i, p in enumerate(imputed_proteins_all.columns):
    adata_full_new.obs[p] = imputed_proteins_all[p].to_numpy().copy()
```

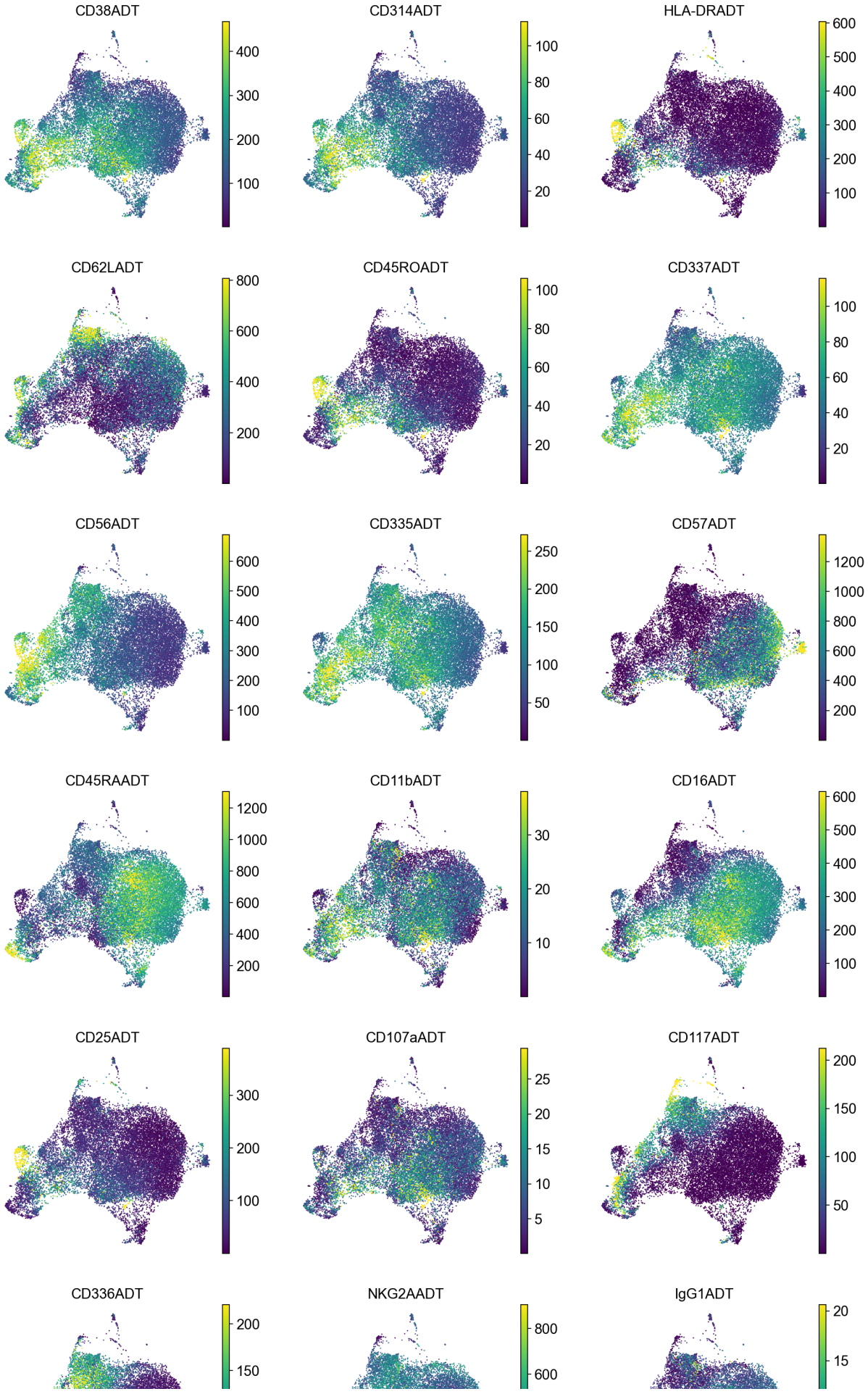
In [ ]:
```
perm_inds = np.random.permutation(np.arange(adata_full_new.n_obs))
sc.pl.umap(
    adata_full_new[perm_inds],
    color=["batch"],
    frameon=False,
    ncols=1,
    title="Reference and query"
)
```

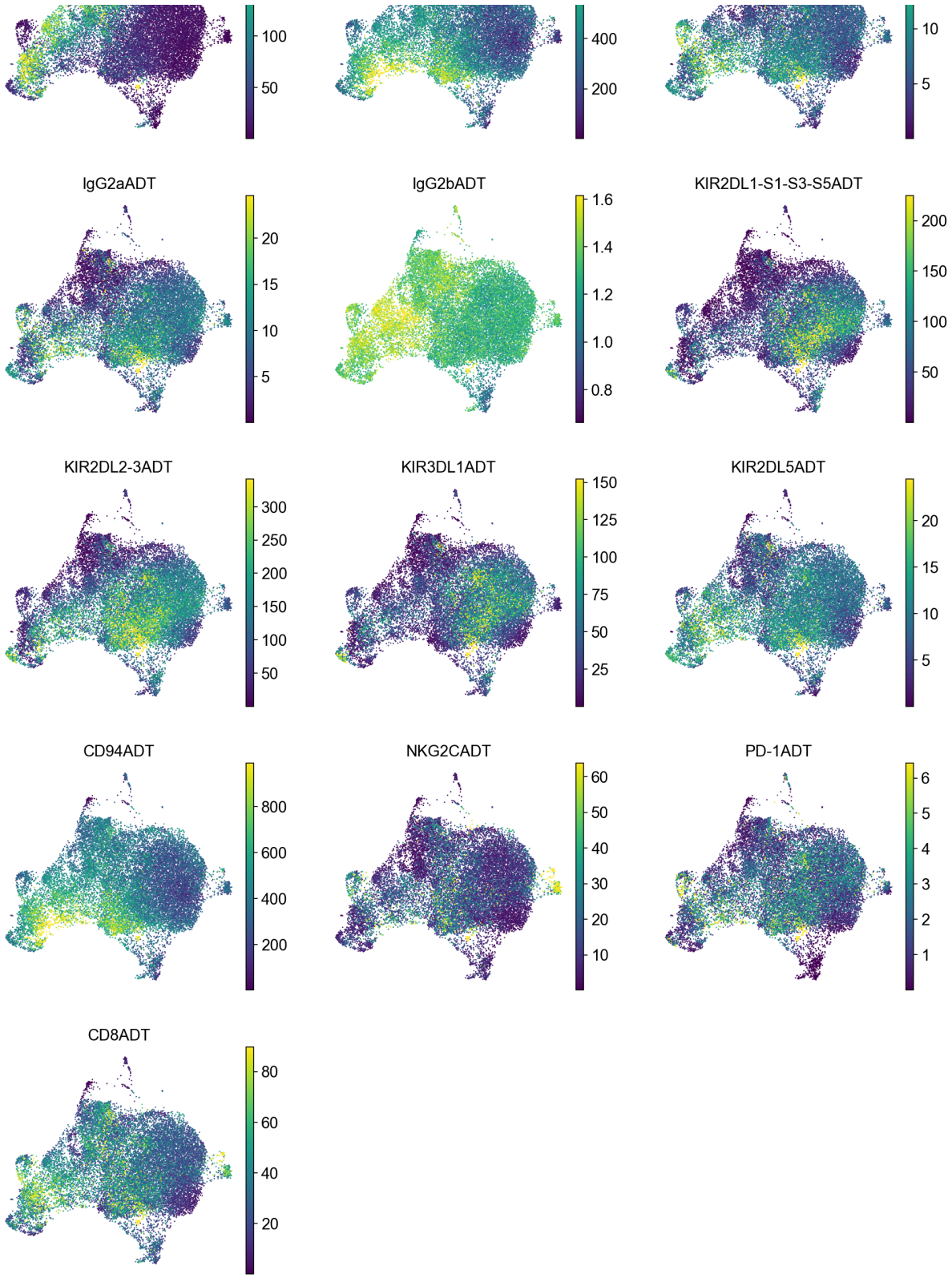


In [ ]:
```
ax = sc.pl.umap(
    adata_full_new,
    color="batch",
    groups=["3228"],
    frameon=False,
    ncols=1,
    title="Reference and query",
    alpha=0.4
)
```



In [ ]:
```
sc.pl.umap(
    adata_full_new,
    color=imputed_proteins_all.columns,
    frameon=False,
    ncols=3,
    vmax="p99"
)
```


CD38ADT
CD314ADT
HLA-DRADT
CD62LADT
CD45ROADT
CD337ADT
CD56ADT
CD335ADT
CD57ADT
CD45RAADT
CD11bADT
CD16ADT
CD25ADT
CD107aADT
CD117ADT
CD336ADT
NKG2AADT
IgG1ADT


IgG2aADT
IgG2bADT
KIR2DL1-S1-S3-S5ADT
KIR2DL2-3ADT
KIR3DL1ADT
KIR2DL5ADT
CD94ADT
NKG2CADT
PD-1ADT
CD8ADT