In [ ]:

```
#Repo here: https://github.com/NUPulmonary/scarches-covid-reference
```

In [1]:

```
import scarches as sca
import scanpy as sc
import matplotlib as mpl
import numpy as np
import pandas as pd
import os
import sankey
import matplotlib.pyplot as plt
import sc_utils
import sklearn.metrics
import tensorflow as tf
```

```
C:\Users\[redacted]Anaconda\lib\site-packages\anndata\_core\anndata.py:21:
FutureWarning: pandas.core.index is deprecated and will be removed in a fu
ture version.  The public classes are available in the top-level namespac
e.
  from pandas.core.index import RangeIndex
Using TensorFlow backend.
```

In [2]:

```
data = sc.read("C:/Users[redacted]ondon/PostDoc_Kings/Meta a
nalysis of RNAseq SGC data/scArches/renthal_naive.h5ad")
```

Check of metadata:

In [3]:

```
data.obs[["sex"]]
```

Out[3]:

```
                                      sex
2020_01_18_naive_8.1_bcGVMV             1
2020_01_18_naive_8.1_bcICQA             1
2020_01_18_naive_8.1_bcFBAG             1
2020_01_18_naive_8.1_bcFXMT             1
2020_01_18_naive_8.1_bcGLCH             1
                        ...           ...
2018_08_23_nai_mrgprd_1.3_bcCMWO        1
2018_08_23_nai_mrgprd_1.3_bcCYUJ        1
2018_08_23_nai_mrgprd_1.3_bcCDJD        1
2018_08_23_nai_mrgprd_1.3_bcGPTH        1
2018_08_23_nai_mrgprd_1.3_bcDARW        1

16859 rows × 1 columns
```

Something is wrong with the metadata so I import it again:

In [4]:

```
pd_obs = pd.read_csv("C:/User[redacted] London/PostDoc_Kings/
Meta analysis of RNAseq SGC data/scArches/renthal_naive_metadata_scArches.csv")
```

In [5]:

```
data.obs = pd_obs
```

In [6]:

```
data.obs[["sex"]]
```

Out[6]:

	sex
0	male
1	male
2	male
3	male
4	male
...	...
16854	male
16855	male
16856	male
16857	male
16858	male

16859 rows × 1 columns

In [7]:

```
sc.pp.log1p(data)
```

In [8]:

```
sc.pp.highly_variable_genes(data, n_top_genes=4000, batch_key="subtype")
```

```
... storing 'orig.ident' as categorical
... storing 'sample_name' as categorical
... storing 'sex' as categorical
... storing 'injury' as categorical
... storing 'strain' as categorical
... storing 'injured' as categorical
... storing 'cellID' as categorical
... storing 'subtype' as categorical
... storing 'class' as categorical
... storing 'biorep' as categorical
```

In [9]:

```
adata = data[:, data.var.highly_variable]
```

In [10]:

```
network = sca.models.scArches(task_name='Renthal_naive_highly_var',
                              x_dimension=adata.shape[1],
                              z_dimension=10,
                              architecture=[128, 128],
                              gene_names=adata.var_names.tolist(),
                              conditions=adata.obs["orig.ident"].unique().tolist(),
                              alpha=0.001,
                              loss_fn='sse',
                              model_path="C:/[redacted]L
ondon/PostDoc_Kings/Meta analysis of RNAseq SGC data/scArches/",
                              )
```

```
WARNING:tensorflow:From C:\Users[redacted]ib\site-packages\keras
\backend\tensorflow_backend.py:174: The name tf.get_default_session is dep
recated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From C:\Users[redacted]b\site-packages\keras
\backend\tensorflow_backend.py:181: The name tf.ConfigProto is deprecated.
Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From C:\Users[redacted]b\site-packages\keras
\backend\tensorflow_backend.py:186: The name tf.Session is deprecated. Ple
ase use tf.compat.v1.Session instead.

WARNING:tensorflow:From C:\Users[redacted]b\site-packages\keras
\backend\tensorflow_backend.py:190: The name tf.global_variables is deprec
ated. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From C:\User[redacted]ib\site-packages\keras
\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated.
Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Use[redacted]lib\site-packages\keras
\backend\tensorflow_backend.py:4185: The name tf.truncated_normal is depre
cated. Please use tf.random.truncated_normal instead.

WARNING:tensorflow:From C:\Us[redacted]\lib\site-packages\keras
\backend\tensorflow_backend.py:74: The name tf.get_default_graph is deprec
ated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Us[redacted]\lib\site-packages\keras
\backend\tensorflow_backend.py:133: The name tf.placeholder_with_default i
s deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From C:\Use[redacted]lib\site-packages\keras
\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.pyth
on.ops.nn_ops) with keep_prob is deprecated and will be removed in a futur
e version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1
- keep_prob`.
WARNING:tensorflow:From C:\[redacted]da\lib\site-packages\keras
\backend\tensorflow_backend.py:4115: The name tf.random_normal is deprecat
ed. Please use tf.random.normal instead.

scArches' network has been successfully constructed!
WARNING:tensorflow:From C:\Use[redacted]lib\site-packages\keras
\optimizers.py:790: The name tf.train.Optimizer is deprecated. Please use
tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users[redacted]b\site-packages\scarc
hes\models\_losses.py:46: The name tf.variable_scope is deprecated. Please
use tf.compat.v1.variable_scope instead.

WARNING:tensorflow:From C:\User[redacted]ib\site-packages\scarc
hes\models\_losses.py:46: The name tf.AUTO_REUSE is deprecated. Please use
tf.compat.v1.AUTO_REUSE instead.

WARNING:tensorflow:From C:\Users[redacted]b\site-packages\scarc
hes\models\_utils.py:84: The name tf.is_nan is deprecated. Please use tf.m
ath.is_nan instead.

WARNING:tensorflow:From C:\User[redacted]ib\site-packages\scarc
```

```
hes\models\_utils.py:84: where (from tensorflow.python.ops.array_ops) is d
eprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
scArches' network has been successfully compiled!
```

In [11]:

```
network.train(adata,
              condition_key="orig.ident",
              n_epochs=200,
              batch_size=128,
              save=True,
              retrain=False)
```

```
WARNING:tensorflow:From C:\Users[redacted]\lib\site-packages\keras
\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is
deprecated. Please use tf.compat.v1.is_variable_initialized instead.

WARNING:tensorflow:From C:\Use[redacted]da\lib\site-packages\keras
\backend\tensorflow_backend.py:206: The name tf.variables_initializer is d
eprecated. Please use tf.compat.v1.variables_initializer instead.

scArches' network has been successfully compiled!
cvae's weights has been successfully restored!
```

In [12]:

```
latent_adata = network.get_latent(adata, "orig.ident")
```

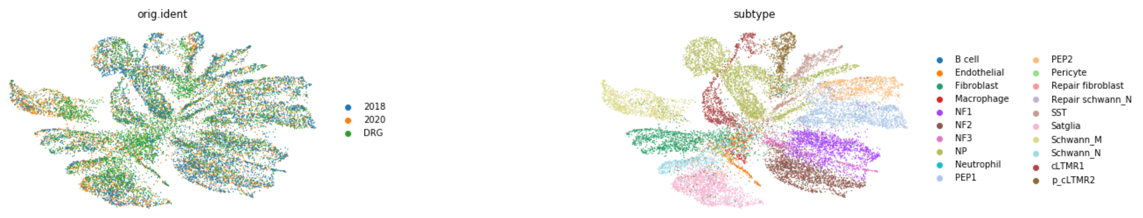
```
Transforming to str index.
```

In [13]:

```
sc.pp.neighbors(latent_adata)
sc.tl.umap(latent_adata)
```

In [14]:

```
sc.pl.umap(latent_adata, color=["orig.ident", "subtype"],
           frameon=False, wspace=0.6)
```



In [39]:

```
data_avra = sc.read("C:/Users/[redacted] King's College London/PostDoc_Kings/M
eta analysis of RNAseq SGC data/scArches/avraham_int_scArches.h5ad")
```

In [42]:

```
avra_obs = pd.read_csv("C:/Users/[redacted] King's College London/PostDoc_King
s/Meta analysis of RNAseq SGC data/Avraham et al/avraham_int_metadata_scArches.csv")
```

[redacted]

In [43]:

```
data_avra.obs = avra_obs
```

In [44]:

```
data_avra.obs["orig.ident_2"]="avra"
```

In [45]:

```
data_avra.obs[["subtype"]]
```

Out[45]:

	subtype
0	Macrophage
1	Macrophage
2	Satglia
3	Smooth muscle
4	Mesenchymal
...	...
6206	Satglia
6207	Macrophage
6208	Fibroblast
6209	Satglia
6210	Macrophage

6211 rows × 1 columns

To make sure I have the same genes in the query dataset (data_avra) as in the reference dataset I join them with outer join and seperate them again based on "orig.ident_2"

In [46]:

```
query_referenxe_adata = adata.concatenate(data_avra, join="outer")
```

```
Transforming to str index.
Transforming to str index.
```

In [47]:

```
query_referenxe_adata.obs[["orig.ident_2"]]
```

Out[47]:

	orig.ident_2
0-0	NaN
1-0	NaN
2-0	NaN
3-0	NaN
4-0	NaN
...	...
6206-1	avra
6207-1	avra
6208-1	avra
6209-1	avra
6210-1	avra

23070 rows × 1 columns

In [48]:

```
query = query_referenxe_adata[query_referenxe_adata.obs["orig.ident_2"] == "avra", : ]
```

Next I make sure that the query data only have the 4000 reference genes that was used to make the model with the reference dataset

In [49]:

```
hvg_names = data.var_names[data.var.highly_variable]
```

In [50]:

```
query2 = query[:, query.var_names.isin(hvg_names)]
```

In [51]:

```
query2
```

Out[51]:

```
View of AnnData object with n_obs × n_vars = 6211 × 4000
    obs: 'UMAP1', 'UMAP2', 'Unnamed: 0', 'batch', 'biorep', 'cellID', 'cel
ltype.stim', 'celltype_condition', 'class', 'condition', 'injured', 'injur
y', 'integrated_snn_res.0.4', 'library_ID', 'library_date', 'nCount_RNA',
'nFeature_RNA', 'nGene', 'nUMI', 'orig.ident', 'orig.ident_2', 'percent.mi
to', 'percent.mt', 'realtime', 'res.1', 'sample_name', 'seurat_clusters',
'sex', 'strain', 'subtype', 'tSNE_1', 'tSNE_2'
    var: 'features-0', 'highly_variable-0', 'means-0', 'dispersions-0', 'd
ispersions_norm-0', 'highly_variable_nbatches-0', 'highly_variable_interse
ction-0', 'features-1', 'integrated_features-1'
```

In [52]:

```
new_network = sca.operate(network,
                          new_task_name="query_avraham_2",
                          new_conditions=[])
```

```
scArches' network has been successfully constructed!
scArches' network has been successfully compiled!
scArches' network has been successfully compiled!
```

In [53]:

```
new_network.train(query2,
                  train_size=0.8,
                  condition_key="orig.ident",
                  n_epochs=50,
                  batch_size=512,
                  save=True,
                  retrain=False)
```

```
WARNING: unique_labels is not subset of the given encoder
WARNING: unique_labels is not subset of the given encoder
scArches' network has been successfully compiled!
cvae's weights has been successfully restored!
```

In [54]:

```
n_neighbors = 10
threhsold = 0.7
```

In [55]:

```
train_latent = new_network.get_latent(adata, "orig.ident")
valid_latent = new_network.get_latent(query2, "orig.ident")
```

```
Transforming to str index.

WARNING: unique_labels is not subset of the given encoder
```

In [56]:

```
sca.ann.weighted_knn(train_latent,
                     valid_latent,
                     label_key="subtype",
                     n_neighbors=n_neighbors,
                     threshold=threhsold,)
```

```
Weighted KNN with n_neighbors = 10 and threshold = 0.7 ... finished!
Number of correctly classified samples: 2278
Number of misclassified samples: 2211
Number of samples classified as unknown: 1722
```

In [57]:

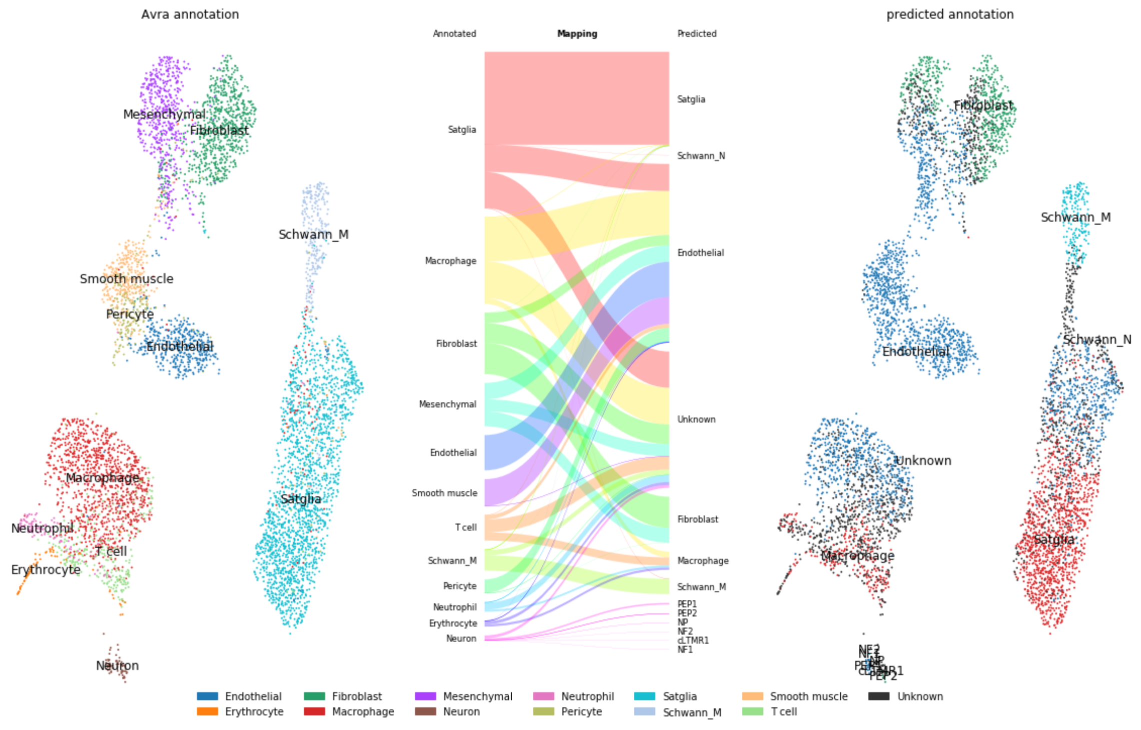
```
sc.pp.neighbors(valid_latent)
sc.tl.umap(valid_latent)
```

In [59]:

```
fig, axes = plt.subplots(ncols=3, figsize=(16, 10), gridspec_kw={
    "width_ratios": [1, 1, 1],
    "wspace": 0
})
sc.pl.umap(valid_latent,
           color="subtype",
           frameon=False,
           size=15,
           legend_loc="on data",
           title="Avra annotation",
           ax=axes[0],
           show=False,
           legend_fontweight="normal",
           legend_fontsize=12)

cluster_colors = pd.Series(valid_latent.uns["subtype_colors"])
cluster_colors.index = valid_latent.obs.subtype.cat.categories
sankey.sankey(valid_latent.obs.subtype,
              valid_latent.obs.pred_subtype,
              title="Mapping",
              title_left="Annotated",
              title_right="Predicted",
              ax=axes[1]);
cluster_colors["Unknown"] = "#333333"
cluster_colors = cluster_colors.sort_index()
pred_cluster_colors = cluster_colors.loc[cluster_colors.index.isin(valid_latent.obs.pre
d_subtype.unique())]
sc.pl.umap(valid_latent,
           color="pred_subtype",
           frameon=False,
           size=15,
           legend_loc="on data",
           title="predicted annotation",
           ax=axes[2],
           palette=list(pred_cluster_colors),
           show=False,
           legend_fontweight="normal",
           legend_fontsize=12)
handles = []
for i in cluster_colors.index:
    handles.append(mpl.patches.Patch(color=cluster_colors[i], label=i))
fig.legend(handles=handles, loc="lower center", frameon=False, ncol=cluster_colors.size
// 2 + 1)
fig.tight_layout()
p = axes[1].get_position()
p.y0 += 0.05
axes[1].set_position(p)
```

```
WARNING: Length of palette colors is smaller than the number of categories
(palette length: 6, categories length: 13. Some categories will have the s
ame color.
```



The figure generated with the above code is a bit confusing because the Satglia on the lefthand side is blue while the Satglia on the righthand side are red.