sle_notebook_B_cells_T0_LN_celltypes

October 13, 2021

[1]:
```
# preparing the dataset based on the SLEDAI score and LDA

import pandas as pd
import os
import re
import copy
import time
import numpy as np
import pickle
folder ='B_cells'
trt = 'T0'
```

[2]:
```
folder_suffix='notebook_result_LN_'+ '1630368126'#str(curr_time)
nfold=5
if trt=='T0':
    marker_file='surface_markers.csv'
else:
    marker_file='signal_markers.csv'
```

[3]:
```
OUTDIR = '../SLE_long_results/'+folder+'_'+trt+'_'+folder_suffix
```

[4]:
```
#patients=pickle.load(open(OUTDIR+'/model/patients.p','rb'))
indv_samples=["328-T0-A-Bcells.npy",
"311-T0-A-Bcells.npy",
"323-T0-A-Bcells.npy",
"305-T0-A-Bcells.npy",
"302-T0-A-Bcells.npy",
"309-T0-A-Bcells.npy",
"326-T0-A-Bcells.npy",
"316-T0-A-Bcells.npy",
"317-T0-A-Bcells.npy",
"330-T0-A-Bcells.npy",
"321-T0-A-Bcells.npy",
"306-T0-A-Bcells.npy",
"310-T0-A-Bcells.npy",
"329-T0-A-Bcells.npy",
"327-T0-A-Bcells.npy",
"336-T0-A-Bcells.npy",
```

```
"319-T0-A-Bcells.npy",
"325-T0-A-Bcells.npy",
"303-T0-A-Bcells.npy",
"308-T0-A-Bcells.npy",
"322-T0-A-Bcells.npy",
"307-T0-A-Bcells.npy"]

sample=[]
for sm in range(len(indv_samples)):
        sample.append(np.load('../Bcells_event_T0_A/'+indv_samples[sm]))
#print(sample)

cellids=[]
indv_cellids=["328-T0-A-cellids.npy",
"311-T0-A-cellids.npy",
"323-T0-A-cellids.npy",
"305-T0-A-cellids.npy",
"302-T0-A-cellids.npy",
"309-T0-A-cellids.npy",
"326-T0-A-cellids.npy",
"316-T0-A-cellids.npy",
"317-T0-A-cellids.npy",
"330-T0-A-cellids.npy",
"321-T0-A-cellids.npy",
"306-T0-A-cellids.npy",
"310-T0-A-cellids.npy",
"329-T0-A-cellids.npy",
"327-T0-A-cellids.npy",
"336-T0-A-cellids.npy",
"319-T0-A-cellids.npy",
"325-T0-A-cellids.npy",
"303-T0-A-cellids.npy",
"308-T0-A-cellids.npy",
"322-T0-A-cellids.npy",
"307-T0-A-cellids.npy"]

cellids=[]
for cids in range(len(indv_cellids)):
    cellids.append(np.load('../Bcells_event_T0_A/'+indv_cellids[cids]))
#print(cellids)

sample_dict={"317-T0-A-Bcells.npy": 1,
"325-T0-A-Bcells.npy": 1,
"327-T0-A-Bcells.npy": 1,
"329-T0-A-Bcells.npy": 1,
"323-T0-A-Bcells.npy": 1,
"330-T0-A-Bcells.npy": 1,
```

```
"336-T0-A-Bcells.npy": 1,
"307-T0-A-Bcells.npy": 0,
"306-T0-A-Bcells.npy": 0,
"308-T0-A-Bcells.npy": 0,
"316-T0-A-Bcells.npy": 0,
"302-T0-A-Bcells.npy": 0,
"319-T0-A-Bcells.npy": 0,
"321-T0-A-Bcells.npy": 0,
"322-T0-A-Bcells.npy": 0,
"326-T0-A-Bcells.npy": 0,
"328-T0-A-Bcells.npy": 0,
"303-T0-A-Bcells.npy": 0,
"305-T0-A-Bcells.npy": 0,
"309-T0-A-Bcells.npy": 0,
"310-T0-A-Bcells.npy": 0,
"311-T0-A-Bcells.npy": 0}
```

[5]:
```
print(len(np.hstack(cellids)), np.vstack(sample).shape)
phenotypes=[]
for sm in range(len(indv_samples)):
    phenotypes.append(sample_dict[indv_samples[sm]])
print(phenotypes)
```

```
536912 (536912, 24)
[0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0]
```

[6]:
```
phenotypes=pickle.load(open(OUTDIR+'/model/phenotypes.p','rb'))
results=pickle.load(open(OUTDIR+'/model/results.p','rb'))
marker_csv = '../metadata/'+marker_file
marker_info = np.array(pd.read_csv(marker_csv, sep=',').columns)
markers = list(marker_info)
```

[7]:
```
from plotting import plot_results
print('generating the plots...')
percentile=85
filter_idx,selected_cells=plot_results(results, sample,cellids, phenotypes,
                        markers, OUTDIR, percentile, filter_response_thres=0,
                        filter_diff_thres=0.
 ↪2,regression=True,group_a='LN(0)',group_b='LN(1)')
```

```
generating the plots…
number of measured markers 24
keep_idx [0]
performing z-transformation
umap for subset of the data…
shape (x_for_umap,x) 100000 536912
UMAP is computed
marker lables ['89Y_CD45', '113In_CD66', '115In_CD15', '141Pr_CD21',
```

```
'142Nd_CD8', '143Nd_CD123', '144Nd_CD3', '146Nd_IgD', '149Sm_CD7',
'151Eu_IgM_EQ', '152Sm_CD11c', '153Eu_CD45RA_EQ', '154Sm_CD14', '155Gd_CD27',
'161Dy_CD1c', '162Dy_PD1', '163Dy_CD19', '165Ho_CD16_EQ', '167Er_HLADR',
'168Er_CD56', '172Yb_CD38', '173Yb_CXCR5', '174Yb_CD4', '209Bi_CD11b']
percentile 85
box plots and histograms are being generated
plotting marker distribution
marker distribution is generated …
box plots and histograms generated
```

[8]:
```
from collections import Counter
for idx in filter_idx:

    print(Counter(selected_cells[idx]))
```

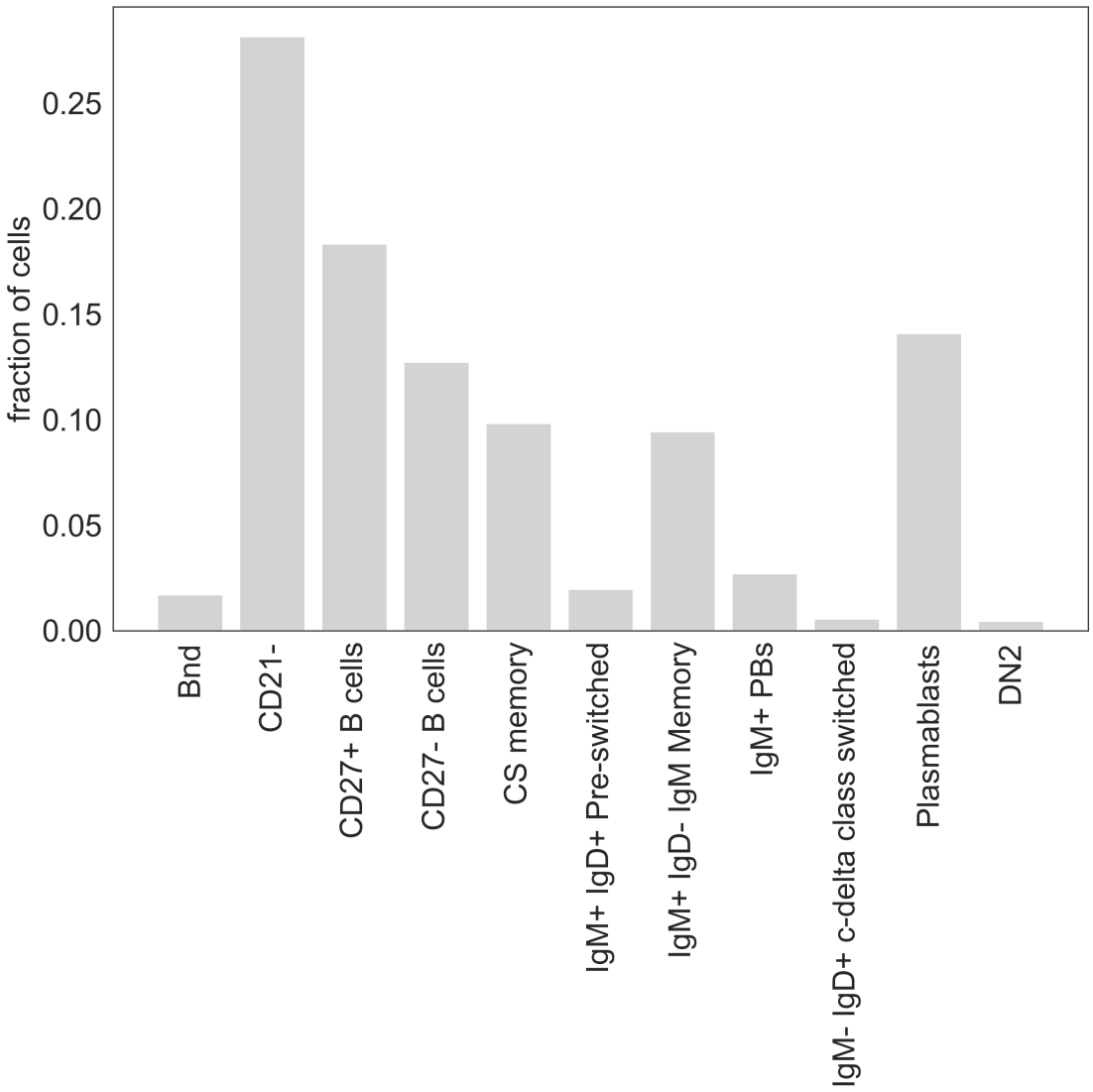
```
Counter({'CD21-': 22678, 'CD27+ B cells': 14770, 'Plasmablasts': 11339, 'CD27- B
cells': 10255, 'CS memory': 7918, 'IgM+ IgD- IgM Memory': 7607, 'IgM+ PBs':
2184, 'IgM+ IgD+ Pre-switched': 1585, 'Bnd': 1379, 'IgM- IgD+ c-delta class
switched': 453, 'DN2': 367})
```

[9]:
```
data=Counter(selected_cells[idx])

# Get the Keys and store them in a list
labels = list(data.keys())

# Get the Values and store them in a list
values = list(data.values())
import matplotlib.pyplot as plt
#%matplotlib
factor=1.0/sum(values)
for k in data:
    data[k] = data[k]*factor
plt.bar(*zip(*data.items()),color='lightgrey')
plt.xticks(fontsize=20,rotation=90)
plt.yticks(fontsize=20)
plt.ylabel('fraction of cells',fontsize=20)
plt.xlabel("")
plt.tight_layout()
from IPython.display import Image
fig_path = '../SLE_long_results/'+folder+'_'+trt+'_'+ folder_suffix
plt.savefig(fig_path+'/'+'selected_celltypes.png', dpi=300)
Image(fig_path+'/'+'selected_celltypes.png', width=600, height=350)
```

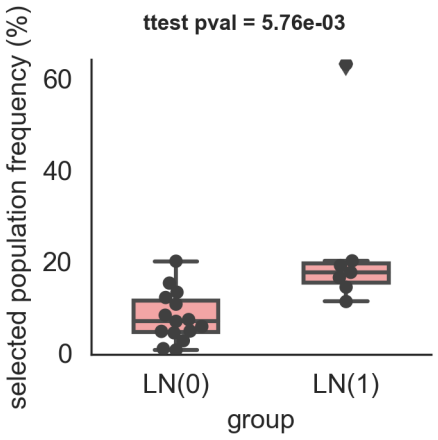
[9]:



[10]:
```
print('selected population frequencies for filter 0')
Image(fig_path+'/'+'selected_population_frequencies_filter_0.png',width=600,␣
↪height=350)
```

```
selected population frequencies for filter 0
```

[10]:



[11]:
```
print('selected population distribution for filter 0')
Image(fig_path+'/'+'selected_population_distribution_filter_0.png')
```

```
selected population distribution for filter 0
```

[11]: