Final_Plots-Copy1

September 16, 2021

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

[50]:
```
input_path = 'Z:/V11/vM1_Ref_Frame/FINAL/'
cm = 1/2.54
```

1 Clustering

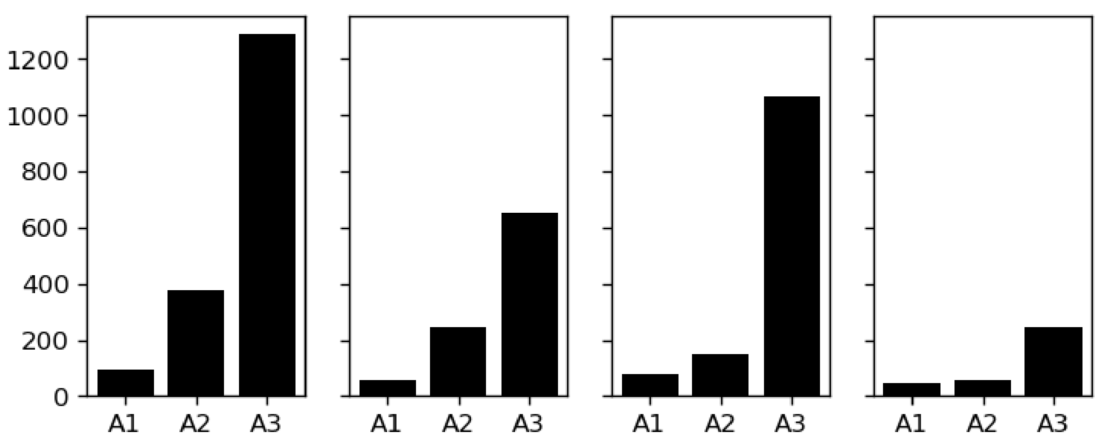
1.1 Absolute rabies counts and cov

[71]:
```
# the total number of rabies labelled neurons varies a lot across 3 animals
df_rabies_counts = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/
 ↪Clustering/'+'landmarks_counts.csv',index_col=0)
#mg48_count = df_rabies_counts['MG48_lhs_vM1']
mg_48_count_total = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_lhs_vM1','MG48_rhs_vM1','MG48_lhs_vS1','MG48_rhs_vS1']].
 ↪values[0].sum()
mg_48_count_total_lhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_lhs_vM1','MG48_lhs_vS1']].values[0].sum()
mg_48_count_total_rhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_rhs_vM1','MG48_rhs_vS1']].values[0].sum()

mg_49_count_total = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG49_lhs_vM1','MG49_rhs_vM1','MG49_lhs_vS1','MG49_rhs_vS1']].
 ↪values[0].sum()
mg_49_count_total_lhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG49_lhs_vM1','MG49_lhs_vS1']].values[0].sum()
mg_49_count_total_rhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG49_rhs_vM1','MG49_rhs_vS1']].values[0].sum()

mg_50_count_total = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG50_lhs_vM1','MG50_rhs_vM1','MG50_lhs_vS1','MG50_rhs_vS1']].
 ↪values[0].sum()
```

```
mg_50_count_total_lhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG50_lhs_vM1','MG50_lhs_vS1']].values[0].sum()
mg_50_count_total_rhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG50_rhs_vM1','MG50_rhs_vS1']].values[0].sum()

totals = np.array([mg_48_count_total,mg_49_count_total,mg_50_count_total])
totals_lhs = np.
 ↪array([mg_48_count_total_lhs,mg_49_count_total_lhs,mg_50_count_total_lhs,])
totals_rhs = np.
 ↪array([mg_48_count_total_rhs,mg_49_count_total_rhs,mg_50_count_total_rhs])

totals.mean(),totals.std(),totals.std()/totals.mean()*100
```

[71]: (1452.0, 1295.5155987739656, 89.2228373811271)

[72]:
```
fig, axes = plt.subplots(nrows=1,ncols=4,␣
 ↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
axes[0].bar(['A1','A2','A3'],df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']].
 ↪values[0],color=(0,0,0))
axes[1].bar(['A1','A2','A3'],df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']].
 ↪values[0],color=(0,0,0))
axes[2].bar(['A1','A2','A3'],df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']].
 ↪values[0],color=(0,0,0))
axes[3].bar(['A1','A2','A3'],df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']].
 ↪values[0],color=(0,0,0))
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/Clustering/total_counts.
 ↪eps')
```



[73]:
```
df_rabies_counts[df_rabies_counts.
↪index==0][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']]
```

[73]:
```
        MG48_lhs_vM1  MG49_lhs_vM1  MG50_lhs_vM1
Cutoff
0                 96           376          1292
```

[74]:
```
vm1_lhs = df_rabies_counts[df_rabies_counts.
↪index==0][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']]
vm1_lhs.mean(axis=1),vm1_lhs.std(axis=1),vm1_lhs.std(axis=1)/vm1_lhs.
↪mean(axis=1),
```

[74]:
```
(Cutoff
 0    588.0
 dtype: float64,
 Cutoff
 0    625.549359
 dtype: float64,
 Cutoff
 0    1.063859
 dtype: float64)
```

[75]:
```
df_rabies_counts[df_rabies_counts.
↪index==0][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']]
```

[75]:
```
        MG48_rhs_vM1  MG49_rhs_vM1  MG50_rhs_vM1
Cutoff
0                 58           245           653
```

[76]:
```
vm1_lhs = df_rabies_counts[df_rabies_counts.
↪index==0][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']]
vm1_lhs.mean(axis=1),vm1_lhs.std(axis=1),vm1_lhs.std(axis=1)/vm1_lhs.
↪mean(axis=1),
```

[76]:
```
(Cutoff
 0    318.666667
 dtype: float64,
 Cutoff
 0    304.263592
 dtype: float64,
 Cutoff
 0    0.954802
 dtype: float64)
```

[77]:
```
df_rabies_counts[df_rabies_counts.
↪index==0][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']]
```

[77]:
```
        MG48_lhs_vS1  MG49_lhs_vS1  MG50_lhs_vS1
Cutoff
0                 76           149          1066
```

[78]:
```
vm1_lhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']]
vm1_lhs.mean(axis=1),vm1_lhs.std(axis=1),vm1_lhs.std(axis=1)/vm1_lhs.
 ↪mean(axis=1),
```

[78]:
```
(Cutoff
 0    430.333333
 dtype: float64,
 Cutoff
 0    551.712183
 dtype: float64,
 Cutoff
 0    1.282058
 dtype: float64)
```

[79]:
```
df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']]
```

[79]:
```
        MG48_rhs_vS1  MG49_rhs_vS1  MG50_rhs_vS1
Cutoff
0                 43            57           245
```

[80]:
```
vm1_lhs = df_rabies_counts[df_rabies_counts.
 ↪index==0][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']]
vm1_lhs.mean(axis=1),vm1_lhs.std(axis=1),vm1_lhs.std(axis=1)/vm1_lhs.
 ↪mean(axis=1),
```

[80]:
```
(Cutoff
 0    115.0
 dtype: float64,
 Cutoff
 0    112.800709
 dtype: float64,
 Cutoff
 0    0.980876
 dtype: float64)
```

[81]:
```
np.array([1.063859,0.954802,1.282058,0.980876]).mean()
```

[81]:
```
1.07039875
```

1.2 Clustered rabies volumes and cov

[82]:
```
df_surf_vols_amira = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/
↪Clustering/surf_volumes_from_amira.csv',\
                                 sep=';',thousands='.',index_col=0)
mg_48_volume_total = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vM1','MG48_rhs_vM1','MG48_lhs_vS1','MG48_rhs_vS1']].
↪values[0].sum()
mg_48_volume_total_lhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vM1','MG48_lhs_vS1']].values[0].sum()
mg_48_volume_total_rhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_rhs_vM1','MG48_rhs_vS1']].values[0].sum()

mg_49_volume_total = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG49_lhs_vM1','MG49_rhs_vM1','MG49_lhs_vS1','MG49_rhs_vS1']].
↪values[0].sum()
mg_49_volume_total_lhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG49_lhs_vM1','MG49_lhs_vS1']].values[0].sum()
mg_49_volume_total_rhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG49_rhs_vM1','MG49_rhs_vS1']].values[0].sum()

mg_50_volume_total = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG50_lhs_vM1','MG50_rhs_vM1','MG50_lhs_vS1','MG50_rhs_vS1']].
↪values[0].sum()
mg_50_volume_total_lhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG50_lhs_vM1','MG50_lhs_vS1']].values[0].sum()
mg_50_volume_total_rhs = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG50_rhs_vM1','MG50_rhs_vS1']].values[0].sum()

totals = np.array([mg_48_volume_total,mg_49_volume_total,mg_50_volume_total])
totals_lhs = np.
↪array([mg_48_volume_total_lhs,mg_49_volume_total_lhs,mg_50_volume_total_lhs])
totals_rhs = np.
↪array([mg_48_volume_total_rhs,mg_49_volume_total_rhs,mg_50_volume_total_rhs])

totals.mean(),totals.std(),totals.std()/totals.mean()*100
```

[82]: (29918853888.0, 6588226373.641311, 22.02031668159504)

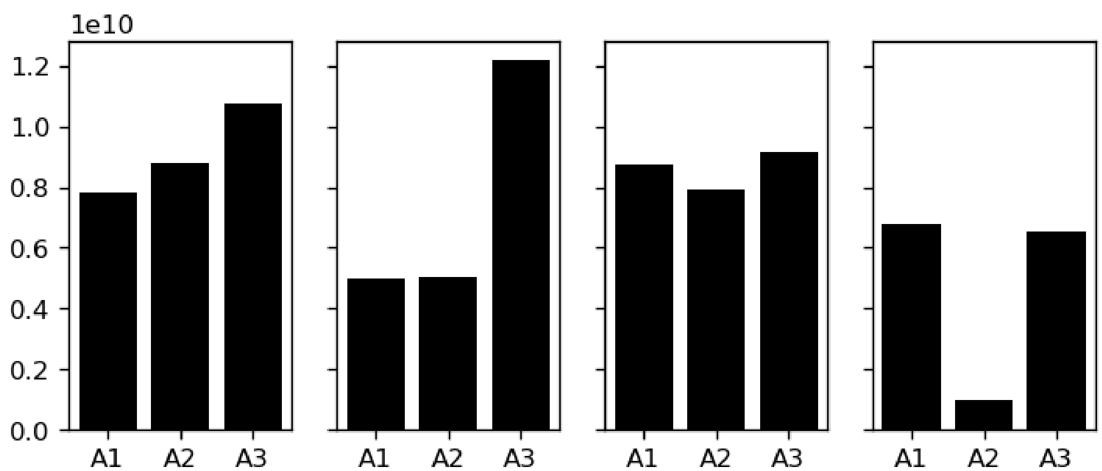
[83]:
```
totals_lhs.mean(),totals_lhs.std(),totals_lhs.std()/totals_lhs.mean()*100
```

[83]: (17738112000.0, 1535953136.6981075, 8.659056480746695)

[84]:
```
totals_rhs.mean(),totals_rhs.std(),totals_rhs.std()/totals_rhs.mean()*100
```

[84]: (12180741888.0, 5207223830.138087, 42.749644299318454)

[85]:
```
fig, axes = plt.subplots(nrows=1,ncols=4,␣
↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
axes[0].bar(['A1','A2','A3'],df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']].
↪values[0],color=(0,0,0))
axes[1].bar(['A1','A2','A3'],df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']].
↪values[0],color=(0,0,0))
axes[2].bar(['A1','A2','A3'],df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']].
↪values[0],color=(0,0,0))
axes[3].bar(['A1','A2','A3'],df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']].
↪values[0],color=(0,0,0))
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/Clustering/total_volumes.
↪eps')
```



[86]:
```
df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']]
```

[86]:

	MG48_lhs_vM1	MG49_lhs_vM1	MG50_lhs_vM1
Cutoff			
10	7803322368	8817117184	10736518144

[87]:
```
bla = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_lhs_vM1','MG49_lhs_vM1','MG50_lhs_vM1']]
bla.mean(axis=1),bla.std(axis=1),bla.std(axis=1)/bla.mean(axis=1),
```

[87]:
```
(Cutoff
 10    9.118986e+09
```

```
 dtype: float64,
 Cutoff
 10    1.489716e+09
 dtype: float64,
 Cutoff
 10    0.163364
 dtype: float64)
```

[88]:
```
df_surf_vols_amira[df_surf_vols_amira.
 ↪index==10][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']]
```

[88]:
```
        MG48_rhs_vM1  MG49_rhs_vM1  MG50_rhs_vM1
Cutoff
10        5000535552    5035372032   12209556480
```

[89]:
```
bla = df_surf_vols_amira[df_surf_vols_amira.
 ↪index==10][['MG48_rhs_vM1','MG49_rhs_vM1','MG50_rhs_vM1']]
bla.mean(axis=1),bla.std(axis=1),bla.std(axis=1)/bla.mean(axis=1),
```

[89]:
```
(Cutoff
 10    7.415155e+09
 dtype: float64,
 Cutoff
 10    4.152110e+09
 dtype: float64,
 Cutoff
 10    0.559949
 dtype: float64)
```

[90]:
```
df_surf_vols_amira[df_surf_vols_amira.
 ↪index==10][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']]
```

[90]:
```
        MG48_lhs_vS1  MG49_lhs_vS1  MG50_lhs_vS1
Cutoff
10        8765255680    7920558080    9171564544
```

[91]:
```
bla = df_surf_vols_amira[df_surf_vols_amira.
 ↪index==10][['MG48_lhs_vS1','MG49_lhs_vS1','MG50_lhs_vS1']]
bla.mean(axis=1),bla.std(axis=1),bla.std(axis=1)/bla.mean(axis=1),
```

[91]:
```
(Cutoff
 10    8.619126e+09
 dtype: float64,
 Cutoff
 10    6.381768e+08
 dtype: float64,
 Cutoff
```

```
10    0.074042
dtype: float64)
```

[92]:
```
df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']]
```

[92]:
```
        MG48_rhs_vS1  MG49_rhs_vS1  MG50_rhs_vS1
Cutoff
10        6777475072     978758912    6540527616
```

[93]:
```
bla = df_surf_vols_amira[df_surf_vols_amira.
↪index==10][['MG48_rhs_vS1','MG49_rhs_vS1','MG50_rhs_vS1']]
bla.mean(axis=1),bla.std(axis=1),bla.std(axis=1)/bla.mean(axis=1),
```

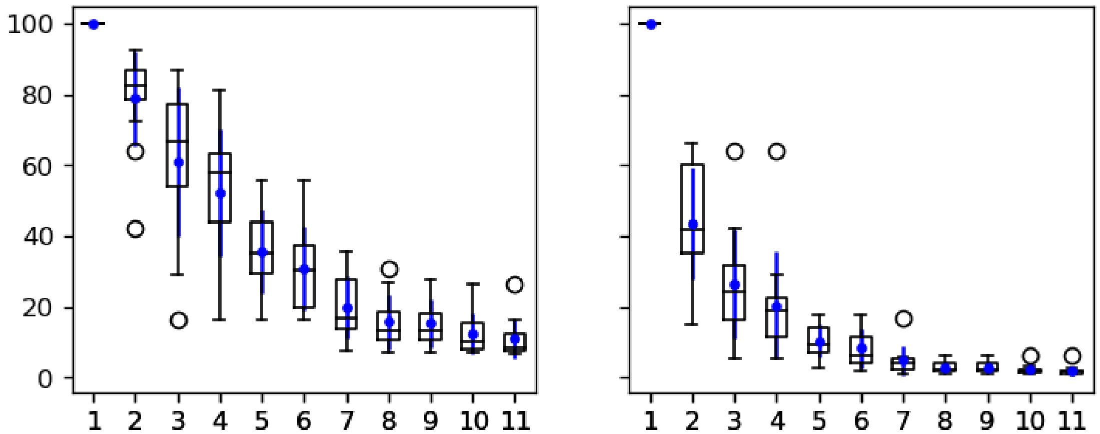
[93]:
```
(Cutoff
 10    4.765587e+09
 dtype: float64,
 Cutoff
 10    3.281629e+09
 dtype: float64,
 Cutoff
 10    0.68861
 dtype: float64)
```

1.3 Effect of density cutoff thersholds on clustering

[94]:
```
df_rabies_counts = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/
↪Clustering/'+'landmarks_counts.csv',index_col=0)
df_surf_areas = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/Clustering/
↪'+'surf_areas.csv',index_col=0)
df_surf_volumes_per = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/
↪Clustering/'+'surf_volumes_from_amira_per.csv',index_col=0)
```

[95]:
```
df_counts = df_rabies_counts.filter(regex='per').transpose()
df_surfs = df_surf_areas.filter(regex='per').transpose()
fig, axes = plt.subplots(nrows=1,ncols=2,␣
↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(df_counts[df_counts.columns[::-1]])
eb2 = axes[0].errorbar(x=[1,2,3,4,5,6,7,8,9,10,11],y=np.
↪array(df_counts[df_counts.columns[::-1]].transpose()).mean(axis=1),yerr=np.
↪array(df_counts[df_counts.columns[::-1]].transpose()).std(axis=1),\
             color='blue',marker='.',fmt='.')
b2 = axes[1].boxplot(df_surfs[df_surfs.columns[::-1]])
eb2 = axes[1].errorbar(x=[1,2,3,4,5,6,7,8,9,10,11],y=np.array(df_surfs[df_surfs.
↪columns[::-1]].transpose()).mean(axis=1),yerr=np.array(df_surfs[df_surfs.
↪columns[::-1]].transpose()).std(axis=1),\
             color='blue',marker='.',fmt='.')
```

```
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')

fig.savefig(input_path + 'Fig_1/Rabies_Distribution/Clustering/density_cutoffs.
↪eps')
```



[96]:
```
df_counts[df_counts.columns[::-1]].mean(),df_counts[df_counts.columns[::-1]].
↪std(),df_counts[df_counts.columns[::-1]].std()/df_counts[df_counts.columns[::
↪-1]].mean()
```

[96]:
```
(Cutoff
 0      100.000000
 10      78.972736
 20      61.044467
 30      52.317847
 40      35.613992
 50      30.848289
 60      20.047840
 70      15.788484
 80      15.332909
 90      12.629794
 100     11.061912
 dtype: float64,
 Cutoff
 0        0.000000
```

```
10      13.959851
20      21.806309
30      18.581325
40      12.173787
50      12.421827
60       9.206613
70       7.951789
80       7.135830
90       5.902816
100      5.772488
dtype: float64,
Cutoff
0       0.000000
10      0.176768
20      0.357220
30      0.355162
40      0.341826
50      0.402675
60      0.459232
70      0.503645
80      0.465393
90      0.467372
100     0.521835
dtype: float64)
```

[97]:
```
df_surfs[df_surfs.columns[::-1]].mean(),df_surfs[df_surfs.columns[::-1]].
↪std(),df_surfs[df_surfs.columns[::-1]].std()/df_surfs[df_surfs.columns[::
↪-1]].mean()
```

[97]:
```
(Cutoff
 0       100.000000
 10       43.652848
 20       26.505614
 30       20.498617
 40       10.416689
 50        8.327781
 60        4.837567
 70        3.024383
 80        2.952427
 90        2.244578
 100       2.007696
 dtype: float64,
 Cutoff
 0        0.000000
 10      16.398099
 20      16.109189
 30      15.614610
```

```
40      4.843768
50      5.562739
60      4.200960
70      1.617775
80      1.555020
90      1.361272
100     1.413926
dtype: float64,
Cutoff
0      0.000000
10     0.375648
20     0.607765
30     0.761740
40     0.465001
50     0.667974
60     0.868403
70     0.534911
80     0.526692
90     0.606471
100    0.704253
dtype: float64)
```

2 Rabies Distribution in Atlas

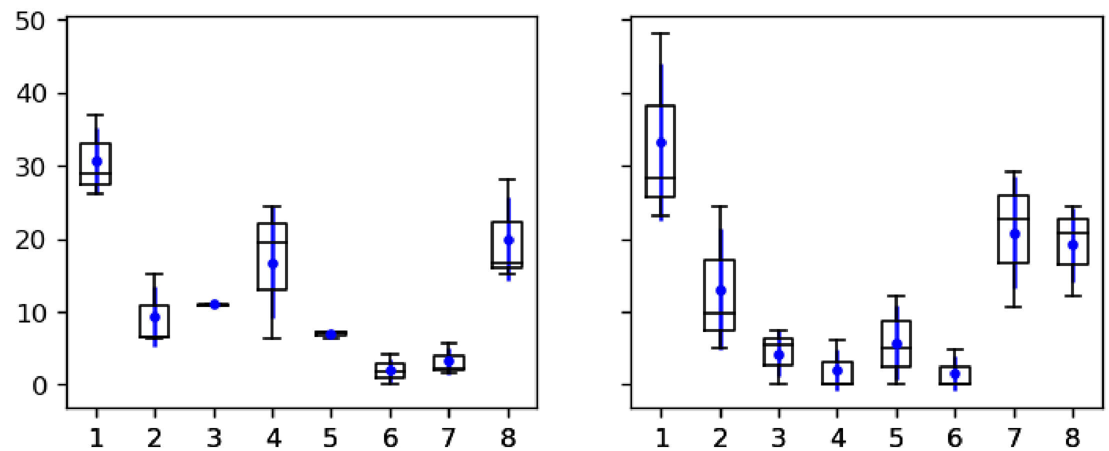
[98]:
```
df_atlas = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/Atlas/
↪'+'Atlas_Counts_per.csv',index_col=0)
df_atlas_clustered = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/
↪Atlas_Clustered/'+'Atlas_Counts_per.csv',index_col=0)
df = df_atlas_clustered
```

[99]:
```
df_atlas_clustered.index
```

[99]:
```
Index(['M1', 'M2', 'S1_ULp', 'SBF', 'S2', 'S1_fl', 'Insular', 'Others'],
dtype='object', name='Atlas_Region')
```

[100]:
```
df_lhs = df[['lhs_MG48','lhs_MG49','lhs_MG50']]
df_rhs = df[['rhs_MG48','rhs_MG49','rhs_MG50']]
#df_lhs.mean(axis=1),df_lhs.std(axis=1),df_rhs.mean(axis=1),df_rhs.std(axis=1)

#df = pd.DataFrame()
#df = df.append(df_lhs)
#df = df.append(df_rhs)
#df_lhs.transpose().plot(kind='box', stacked=True)
#df_rhs.transpose().plot(kind='box', stacked=True)
#axes.boxplot(df_lhs.transpose(),)
#axes.boxplot(df_rhs.transpose())
```

```
#axes.set_edgecolor('red')
fig, axes = plt.subplots(nrows=1,ncols=2,␣
↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
#axes.errorbar(df_lhs.index.values,df_lhs.transpose().mean(axis=0),yerr=df_lhs.
↪transpose().std(axis=0))
#axes.errorbar(df_rhs.index.values,df_rhs.transpose().mean(axis=0),yerr=df_rhs.
↪transpose().std(axis=0))
b1 = axes[0].boxplot(df_lhs.transpose())
eb1 = axes[0].errorbar(x=[1,2,3,4,5,6,7,8],y=np.array(df_lhs).
↪mean(axis=1),yerr=np.array(df_lhs).std(axis=1),\
             color='blue',marker='.',fmt='.')
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')

b2 = axes[1].boxplot(df_rhs.transpose())
eb2 = axes[1].errorbar(x=[1,2,3,4,5,6,7,8],y=np.array(df_rhs).
↪mean(axis=1),yerr=np.array(df_rhs).std(axis=1),\
             color='blue',marker='.',fmt='.')
plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')

fig.savefig(input_path + 'Fig_1/Rabies_Distribution/Atlas_Clustered/
↪atlas_counts.eps')
```



[101]:
```
df_lhs.mean(axis=1),df_lhs.std(axis=1),df_lhs.std(axis=1)/df_lhs.mean(axis=1)
```

[101]:
```
(Atlas_Region
 M1          30.680840
 M2           9.421677
 S1_ULp      10.981062
 SBF         16.776272
 S2           6.984500
 S1_fl        1.936321
 Insular      3.236422
 Others      19.982906
 dtype: float64,
 Atlas_Region
 M1          5.633135
 M2          5.020520
 S1_ULp      0.169554
 SBF         9.290977
 S2          0.469766
 S1_fl       2.059179
 Insular     2.252403
 Others      6.977257
 dtype: float64,
 Atlas_Region
 M1          0.183604
 M2          0.532869
 S1_ULp      0.015441
 SBF         0.553817
 S2          0.067258
 S1_fl       1.063449
 Insular     0.695955
 Others      0.349161
 dtype: float64)
```

[102]:
```
df_rhs.mean(axis=1),df_rhs.std(axis=1),df_rhs.std(axis=1)/df_rhs.mean(axis=1)
```

[102]:
```
(Atlas_Region
 M1          33.224430
 M2          13.061121
 S1_ULp       4.231139
 SBF          2.077323
 S2           5.767550
 S1_fl        1.612903
 Insular     20.864665
 Others      19.160868
 dtype: float64,
 Atlas_Region
 M1          13.159230
 M2          10.152341
 S1_ULp       3.790582
```

```
SBF          3.482277
S2           6.124294
S1_fl        2.793630
Insular      9.460580
Others       6.298316
dtype: float64,
Atlas_Region
M1          0.396071
M2          0.777295
S1_ULp      0.895877
SBF         1.676329
S2          1.061854
S1_fl       1.732051
Insular     0.453426
Others      0.328707
dtype: float64)
```

3 Variability

[103]:
```
df = pd.read_csv(input_path + 'Fig_1/Rabies_Distribution/Variability/
↪'+'variability.csv',index_col=0)
```

[104]:
```
df
```

[104]:
```
      Hem   BR  rabies_surf_area     surf_area  rabies_surf_vol      surf_vol  \
Exp
MG48  lhs  vM1      3.864979e+07  5.949542e+07     9.998786e+09  2.933011e+10
MG49  lhs  vM1      3.100946e+07  5.446138e+07     8.817117e+09  2.770998e+10
MG50  lhs  vM1      4.487086e+07  6.644812e+07     1.073652e+10  3.424272e+10
MG48  rhs  vM1      4.027432e+07  5.095105e+07     8.319941e+09  2.491589e+10
MG49  rhs  vM1      2.457457e+07  4.002189e+07     8.473321e+09  1.525190e+10
MG50  rhs  vM1      4.538989e+07  7.190225e+07     1.220956e+10  3.769797e+10
MG48  lhs  vS1      3.517157e+07  7.991888e+07     8.765256e+09  2.227071e+10
MG49  lhs  vS1      3.440360e+07  5.075986e+07     7.920558e+09  2.250900e+10
MG50  lhs  vS1      3.654979e+07  6.439866e+07     1.155127e+10  3.034522e+10
MG48  rhs  vS1      3.642832e+07  5.875139e+07     5.389144e+09  2.025440e+10
MG49  rhs  vS1      7.360863e+06  8.591115e+06     9.787589e+08  1.421633e+09
MG50  rhs  vS1      2.953847e+07  6.121407e+07     6.540527e+09  2.753162e+10

      pia_surf_area  wm_surf_area  pca0_extent  pca1_extent  …  \
Exp                                                          …
MG48   3.405203e+07  7.648197e+06  5495.218316  2438.709082  …
MG49   3.153497e+07  6.590995e+06  4363.242266  2817.266806  …
MG50   3.756157e+07  1.037292e+07  5190.418578  3059.023762  …
MG48   2.965448e+07  4.680838e+06  4566.534440  2312.278507  …
MG49   1.924605e+07  4.796520e+06  3853.662749  1645.224640  …
```

```
MG50   4.467729e+07  1.128978e+07  4991.162955  2963.189153  …
MG48   3.345650e+07  3.569468e+07  4670.086117  3047.370760  …
MG49   1.796299e+07  9.505442e+06  3733.429637  3224.832867  …
MG50   2.672434e+07  1.409410e+07  4993.667943  3630.501278  …
MG48   3.147497e+07  1.746175e+07  3855.213838  2663.134437  …
MG49   1.872575e+06  6.059152e+05  1631.573866  1442.449528  …
MG50   2.593566e+07  1.260094e+07  5237.160116  3096.969710  …

      pca0_angle_Z_axis  pca1_angle_X_axis  pca1_angle_Y_axis  \
Exp
MG48         130.450097          13.544005          76.455995
MG49         124.009155           2.757566          87.242434
MG50         125.523610           8.952160          81.047840
MG48         117.988712         170.214887          80.214887
MG49          94.926912         175.890868          85.890868
MG50         109.686205         166.053442          76.053442
MG48         125.021209          59.618737          30.381263
MG49         117.787791          56.662826          33.337174
MG50         130.446944          52.095473          37.904527
MG48         110.233981          86.120410           3.879590
MG49         121.481232         101.552473          11.552473
MG50          74.921986         110.241371          20.241371

      pca1_angle_Z_axis  pca2_angle_X_axis  pca2_angle_Y_axis  \
Exp
MG48         170.874873          95.607085           5.607085
MG49         168.582760          88.529888           1.470112
MG50           6.103461         102.546421          12.546421
MG48           8.778636          85.023098           4.976902
MG49          30.609510          87.485591           2.514409
MG50          21.252799          90.253644           0.253644
MG48          32.448471         160.362035          70.362035
MG49          17.305465         160.358336          70.358336
MG50          34.382812         166.764675          76.764675
MG48          18.424177           5.416217          84.583783
MG49           7.722050           0.068894          89.931106
MG50         166.212065          20.395848          69.604152

      pca2_angle_Z_axis  Dist_From_Bregma_X  Dist_From_Bregma_Y  \
Exp
MG48          41.621342         2108.854392         3243.377530
MG49          34.622195         2231.130414         3088.905952
MG50          35.552018         2210.779601         2989.982653
MG48          28.937604         2382.954455         3379.329850
MG49           2.115601         2285.709061         3883.483915
MG50          21.421980         2627.218475         2788.760230
MG48         160.158370         4810.509570        -1019.393555
```

```
MG49        138.652532       4607.222303       -290.536024
MG50        153.548737       4615.292784       -860.982242
MG48        153.978686       5369.017905      -1163.679269
MG49        136.819901       5290.379033      -1274.961181
MG50          1.236915       5196.967151       -685.370806

      Dist_From_Bregma_Z
Exp
MG48          2762.964212
MG49          2745.502601
MG50          2677.345192
MG48          2839.165783
MG49          3213.812551
MG50          2462.813891
MG48          3584.832360
MG49          3648.417585
MG50          3333.408306
MG48          5010.493602
MG49          6123.180491
MG50          5292.105781

[12 rows x 23 columns]
```

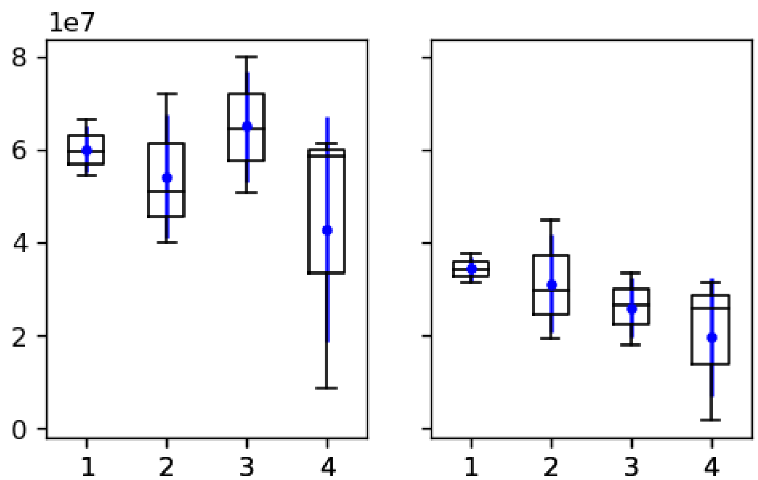
[105]:

```
# surf areas and volumes
df_lhs_vm1 = df[((df['Hem']=='lhs') & (df['BR']=='vM1'))]
df_rhs_vm1 = df[((df['Hem']=='rhs') & (df['BR']=='vM1'))]
df_lhs_vs1 = df[((df['Hem']=='lhs') & (df['BR']=='vS1'))]
df_rhs_vs1 = df[((df['Hem']=='rhs') & (df['BR']=='vS1'))]
```

[106]:

```
surf_areas = [df_lhs_vm1['surf_area'].values,df_rhs_vm1['surf_area'].
 ↪values,df_lhs_vs1['surf_area'].values,df_rhs_vs1['surf_area'].values]
pia_areas = [df_lhs_vm1['pia_surf_area'].values,df_rhs_vm1['pia_surf_area'].
 ↪values,df_lhs_vs1['pia_surf_area'].values,df_rhs_vs1['pia_surf_area'].values]
wm_areas = [df_lhs_vm1['wm_surf_area'].values,df_rhs_vm1['wm_surf_area'].
 ↪values,df_lhs_vs1['wm_surf_area'].values,df_rhs_vs1['wm_surf_area'].values]



fig, axes = plt.subplots(nrows=1,ncols=2,␣
 ↪sharex=True,sharey=True,figsize=(12*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(surf_areas)
eb1 = axes[0].errorbar(x=[1,2,3,4],y=np.array(surf_areas).mean(axis=1),yerr=np.
 ↪array(surf_areas).std(axis=1),\
             color='blue',marker='.',fmt='.')
b2 = axes[1].boxplot(pia_areas)
eb2 = axes[1].errorbar(x=[1,2,3,4],y=np.array(pia_areas).mean(axis=1),yerr=np.
 ↪array(pia_areas).std(axis=1),\
             color='blue',marker='.',fmt='.')
```

```
#b3 = axes[2].boxplot(wm_areas)
#eb3 = axes[2].errorbar(x=[1,2,3,4],y=np.array(wm_areas).mean(axis=1),yerr=np.
↪array(wm_areas).std(axis=1),\
#              color='blue',marker='.',fmt='.')
#b4 = axes[3].boxplot(surf_vols)

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
#plt.setp(b3['boxes'], color='black')
#plt.setp(b3['whiskers'], color='black')
#plt.setp(b3['medians'], color='black')
#plt.setp(b3['caps'], color='black')

fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/surf_areas.eps')
```



[136]:
```
pia_areas
```

[136]:
```
[array([34052026.93275076, 31534973.0764531 , 37561573.11399295]),
 array([29654479.14723352, 19246054.16929449, 44677287.09167311]),
 array([33456498.92602595, 17962988.68771008, 26724336.97520298]),
 array([31474974.9543282 ,  1872575.08959188, 25935659.73541536])]
```

[107]:
```
np.array(surf_areas).mean(axis=1),np.array(surf_areas).std(axis=1),np.
↪array(surf_areas).std(axis=1)/np.array(surf_areas).mean(axis=1)
```

[107]:
```
(array([60134974.32372921, 54291731.92702731, 65025796.55837191,
        42852192.23701403]),
 array([ 4914419.41713892, 13227734.75623022, 11912375.74921831,
        24247092.5934162 ]),
 array([0.08172315, 0.24364179, 0.18319461, 0.56583086]))
```

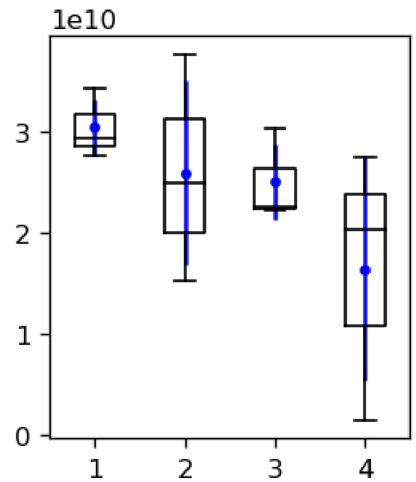
[108]:
```
np.array(pia_areas).mean(axis=1),np.array(pia_areas).std(axis=1),np.
 ↪array(pia_areas).std(axis=1)/np.array(pia_areas).mean(axis=1)
```

[108]:
```
(array([34382857.70773227, 31192606.80273371, 26047941.52964634,
        19761069.92644515]),
 array([ 2471445.42786537, 10439070.18533514,  6343256.15517651,
        12849635.24565877]),
 array([0.07188016, 0.33466489, 0.24352236, 0.65024998]))
```

[109]:
```
np.array(wm_areas).mean(axis=1),np.array(wm_areas).std(axis=1),np.
 ↪array(wm_areas).std(axis=1)/np.array(wm_areas).mean(axis=1)
```

[109]:
```
(array([ 8204036.0609121 ,  6922380.77612065, 19764739.79579503,
        10222870.92297374]),
 array([ 1593203.8704809 ,  3088582.06154907, 11418876.31084508,
         7083843.11825806]),
 array([0.19419757, 0.44617338, 0.57773977, 0.69294068]))
```

[110]:
```
surf_vols = [df_lhs_vm1['surf_vol'].values,df_rhs_vm1['surf_vol'].
 ↪values,df_lhs_vs1['surf_vol'].values,df_rhs_vs1['surf_vol'].values]

fig, axes = plt.subplots(nrows=1,ncols=1,␣
 ↪sharex=True,sharey=True,figsize=(6*cm,7*cm),dpi=120)
b1 = axes.boxplot(surf_vols)
#print(b1)
eb = axes.errorbar(x=[1,2,3,4],y=np.array(surf_vols).mean(axis=1),yerr=np.
 ↪array(surf_vols).std(axis=1),\
             color='blue',marker='.',fmt='.')
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/surf_vols.eps')
```

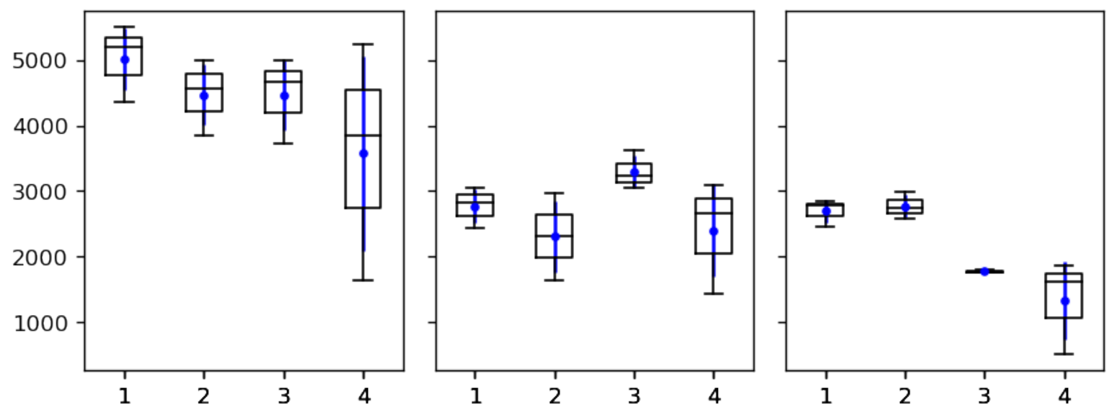


[111]:
```
np.array(surf_vols).mean(axis=1),np.array(surf_vols).std(axis=1),np.
↪array(surf_vols).std(axis=1)/np.array(surf_vols).mean(axis=1)
```

[111]:
```
(array([3.04276071e+10, 2.59552539e+10, 2.50416434e+10, 1.64025540e+10]),
 array([2.77759451e+09, 9.19299494e+09, 3.75145557e+09, 1.10018323e+10]),
 array([0.09128534, 0.35418628, 0.14980868, 0.67073898]))
```

[112]:
```
pca0_extent = [df_lhs_vm1['pca0_extent'].values,df_rhs_vm1['pca0_extent'].
↪values,df_lhs_vs1['pca0_extent'].values,df_rhs_vs1['pca0_extent'].values]
pca1_extent = [df_lhs_vm1['pca1_extent'].values,df_rhs_vm1['pca1_extent'].
↪values,df_lhs_vs1['pca1_extent'].values,df_rhs_vs1['pca1_extent'].values]
pca2_extent = [df_lhs_vm1['pca2_extent'].values,df_rhs_vm1['pca2_extent'].
↪values,df_lhs_vs1['pca2_extent'].values,df_rhs_vs1['pca2_extent'].values]

fig, axes = plt.subplots(nrows=1,ncols=3,␣
↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(pca0_extent)
eb1 = axes[0].errorbar(x=[1,2,3,4],y=np.array(pca0_extent).mean(axis=1),yerr=np.
↪array(pca0_extent).std(axis=1),\
             color='blue',marker='.',fmt='.')
b2 = axes[1].boxplot(pca1_extent)
eb2 = axes[1].errorbar(x=[1,2,3,4],y=np.array(pca1_extent).mean(axis=1),yerr=np.
↪array(pca1_extent).std(axis=1),\
             color='blue',marker='.',fmt='.')
b3 = axes[2].boxplot(pca2_extent)
eb3 = axes[2].errorbar(x=[1,2,3,4],y=np.array(pca2_extent).mean(axis=1),yerr=np.
↪array(pca2_extent).std(axis=1),\
             color='blue',marker='.',fmt='.')
```

```
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
plt.setp(b3['boxes'], color='black')
plt.setp(b3['whiskers'], color='black')
plt.setp(b3['medians'], color='black')
plt.setp(b3['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/pca_extents.
 ↪eps')
```



[113]:
```
np.array(pca0_extent).mean(axis=1),np.array(pca0_extent).std(axis=1),np.
 ↪array(pca0_extent).std(axis=1)/np.array(pca0_extent).mean(axis=1)
```

[113]:
```
(array([5016.2930534 , 4470.45338165, 4465.72789911, 3574.64927344]),
 array([ 478.24834437,  469.32601095,  534.39801517, 1485.28345369]),
 array([0.095339  , 0.10498399, 0.1196665 , 0.41550467]))
```

[114]:
```
np.array(pca1_extent).mean(axis=1),np.array(pca1_extent).std(axis=1),np.
 ↪array(pca1_extent).std(axis=1)/np.array(pca1_extent).mean(axis=1)
```

[114]:
```
(array([2771.6665499 , 2306.89743336, 3300.90163484, 2400.85122542]),
 array([255.28691462, 538.07021295, 244.06302915, 700.4539579 ]),
 array([0.09210593, 0.2332441 , 0.07393829, 0.29175234]))
```

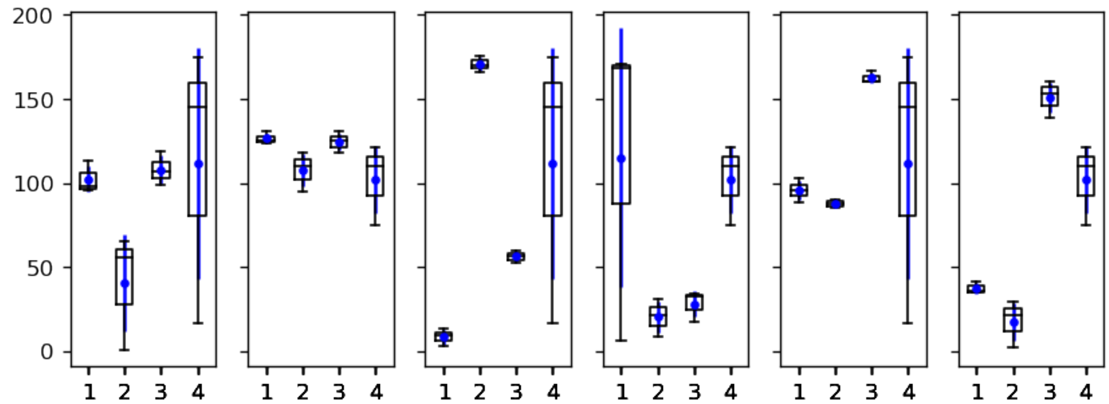
[115]:
```
np.array(pca2_extent).mean(axis=1),np.array(pca2_extent).std(axis=1),np.
 ↪array(pca2_extent).std(axis=1)/np.array(pca2_extent).mean(axis=1)
```

[115]:
```
(array([2692.52605639, 2772.80077258, 1771.99340483, 1329.11796089]),
 array([172.5738613 , 169.59636646,  26.36089975, 588.59135003]),
 array([0.06409366, 0.06116428, 0.01487641, 0.44284358]))
```

[116]:
```
pca0_angle_X_axis = [df_lhs_vm1['pca0_angle_X_axis'].
 ↪values,df_rhs_vm1['pca0_angle_X_axis'].
 ↪values,df_lhs_vs1['pca0_angle_X_axis'].
 ↪values,df_rhs_vs1['pca0_angle_X_axis'].values]
#pca0_angle_Y_axis = [df_lhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_lhs_vs1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Y_axis'].values]
pca0_angle_Z_axis = [df_lhs_vm1['pca0_angle_Z_axis'].
 ↪values,df_rhs_vm1['pca0_angle_Z_axis'].
 ↪values,df_lhs_vs1['pca0_angle_Z_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Z_axis'].values]
pca1_angle_X_axis = [df_lhs_vm1['pca1_angle_X_axis'].
 ↪values,df_rhs_vm1['pca1_angle_X_axis'].
 ↪values,df_lhs_vs1['pca1_angle_X_axis'].
 ↪values,df_rhs_vs1['pca0_angle_X_axis'].values]
#pca0_angle_Y_axis = [df_lhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_lhs_vs1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Y_axis'].values]
pca1_angle_Z_axis = [df_lhs_vm1['pca1_angle_Z_axis'].
 ↪values,df_rhs_vm1['pca1_angle_Z_axis'].
 ↪values,df_lhs_vs1['pca1_angle_Z_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Z_axis'].values]
pca2_angle_X_axis = [df_lhs_vm1['pca2_angle_X_axis'].
 ↪values,df_rhs_vm1['pca2_angle_X_axis'].
 ↪values,df_lhs_vs1['pca2_angle_X_axis'].
 ↪values,df_rhs_vs1['pca0_angle_X_axis'].values]
#pca0_angle_Y_axis = [df_lhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vm1['pca0_angle_Y_axis'].
 ↪values,df_lhs_vs1['pca0_angle_Y_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Y_axis'].values]
pca2_angle_Z_axis = [df_lhs_vm1['pca2_angle_Z_axis'].
 ↪values,df_rhs_vm1['pca2_angle_Z_axis'].
 ↪values,df_lhs_vs1['pca2_angle_Z_axis'].
 ↪values,df_rhs_vs1['pca0_angle_Z_axis'].values]

fig, axes = plt.subplots(nrows=1,ncols=6,␣
 ↪sharex=True,sharey=True,figsize=(18*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(pca0_angle_X_axis)
eb1 =axes[0].errorbar(x=[1,2,3,4],y=np.array(pca0_angle_X_axis).
 ↪mean(axis=1),yerr=np.array(pca0_angle_X_axis).std(axis=1),\
```

```
             color='blue',marker='.',fmt='.')
b2 = axes[1].boxplot(pca0_angle_Z_axis)
eb2 =axes[1].errorbar(x=[1,2,3,4],y=np.array(pca0_angle_Z_axis).
 ↪mean(axis=1),yerr=np.array(pca0_angle_Z_axis).std(axis=1),\
             color='blue',marker='.',fmt='.')
b3 = axes[2].boxplot(pca1_angle_X_axis)
eb3 = axes[2].errorbar(x=[1,2,3,4],y=np.array(pca1_angle_X_axis).
 ↪mean(axis=1),yerr=np.array(pca1_angle_X_axis).std(axis=1),\
             color='blue',marker='.',fmt='.')
b4 = axes[3].boxplot(pca1_angle_Z_axis)
eb4 = axes[3].errorbar(x=[1,2,3,4],y=np.array(pca1_angle_Z_axis).
 ↪mean(axis=1),yerr=np.array(pca1_angle_Z_axis).std(axis=1),\
             color='blue',marker='.',fmt='.')
b5 = axes[4].boxplot(pca2_angle_X_axis)
eb5 = axes[4].errorbar(x=[1,2,3,4],y=np.array(pca2_angle_X_axis).
 ↪mean(axis=1),yerr=np.array(pca2_angle_X_axis).std(axis=1),\
             color='blue',marker='.',fmt='.')
b6 = axes[5].boxplot(pca2_angle_Z_axis)
eb6 = axes[5].errorbar(x=[1,2,3,4],y=np.array(pca2_angle_Z_axis).
 ↪mean(axis=1),yerr=np.array(pca2_angle_Z_axis).std(axis=1),\
             color='blue',marker='.',fmt='.')

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
plt.setp(b3['boxes'], color='black')
plt.setp(b3['whiskers'], color='black')
plt.setp(b3['medians'], color='black')
plt.setp(b3['caps'], color='black')
plt.setp(b4['boxes'], color='black')
plt.setp(b4['whiskers'], color='black')
plt.setp(b4['medians'], color='black')
plt.setp(b4['caps'], color='black')
plt.setp(b5['boxes'], color='black')
plt.setp(b5['whiskers'], color='black')
plt.setp(b5['medians'], color='black')
plt.setp(b5['caps'], color='black')
plt.setp(b6['boxes'], color='black')
plt.setp(b6['whiskers'], color='black')
plt.setp(b6['medians'], color='black')
plt.setp(b6['caps'], color='black')
plt.tight_layout()
```

```python
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/
↪pca_orientations.eps')
```



[117]:
```python
np.array(pca0_angle_X_axis).mean(axis=1),np.array(pca0_angle_X_axis).
↪std(axis=1),np.array(pca0_angle_X_axis).std(axis=1)/np.
↪array(pca0_angle_X_axis).mean(axis=1)
```

[117]:
```
(array([102.29198451,  40.35601563, 108.01072233, 111.84864211]),
 array([ 7.84102708, 28.60588025,  8.03867268, 68.73721322]),
 array([0.07665339, 0.70883807, 0.07442477, 0.61455563]))
```

[118]:
```python
np.array(pca0_angle_Z_axis).mean(axis=1),np.array(pca0_angle_Z_axis).
↪std(axis=1),np.array(pca0_angle_Z_axis).std(axis=1)/np.
↪array(pca0_angle_Z_axis).mean(axis=1)
```

[118]:
```
(array([126.66095436, 107.53394327, 124.41864802, 102.21239963]),
 array([ 2.74973905,  9.53714962,  5.18561173, 19.83599734]),
 array([0.02170945, 0.08868967, 0.04167873, 0.19406645]))
```

[119]:
```python
np.array(pca1_angle_X_axis).mean(axis=1),np.array(pca1_angle_X_axis).
↪std(axis=1),np.array(pca1_angle_X_axis).std(axis=1)/np.
↪array(pca1_angle_X_axis).mean(axis=1)
```

[119]:
```
(array([  8.41791036, 170.71973246,  56.12567879, 111.84864211]),
 array([ 4.41971974,  4.03194657,  3.09475575, 68.73721322]),
 array([0.52503763, 0.02361734, 0.05513975, 0.61455563]))
```

[120]:
```python
np.array(pca1_angle_Z_axis).mean(axis=1),np.array(pca1_angle_Z_axis).
↪std(axis=1),np.array(pca1_angle_Z_axis).std(axis=1)/np.
↪array(pca1_angle_Z_axis).mean(axis=1)
```

[120]: (array([115.18703146, 20.21364853, 28.04558306, 102.21239963]),
 array([77.13940821, 8.94265571, 7.63535702, 19.83599734]),
 array([0.66968831, 0.44240681, 0.27224811, 0.19406645]))

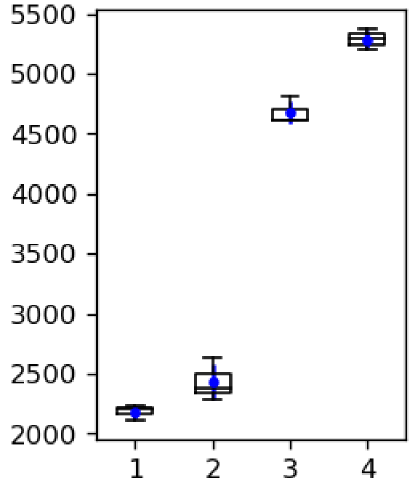
[121]:
```
np.array(pca2_angle_X_axis).mean(axis=1),np.array(pca2_angle_X_axis).
↪std(axis=1),np.array(pca2_angle_X_axis).std(axis=1)/np.
↪array(pca2_angle_X_axis).mean(axis=1)
```

[121]: (array([95.56113108, 87.58744438, 162.49501569, 111.84864211]),
 array([5.72231793, 2.13657571, 3.01910558, 68.73721322]),
 array([0.05988123, 0.02439363, 0.01857968, 0.61455563]))

[122]:
```
np.array(pca2_angle_Z_axis).mean(axis=1),np.array(pca2_angle_Z_axis).
↪std(axis=1),np.array(pca2_angle_Z_axis).std(axis=1)/np.
↪array(pca2_angle_Z_axis).mean(axis=1)
```

[122]: (array([37.26518465, 17.49172831, 150.78654658, 102.21239963]),
 array([3.10357   , 11.29720117, 8.99435146, 19.83599734]),
 array([0.08328337, 0.64585963, 0.05964956, 0.19406645]))

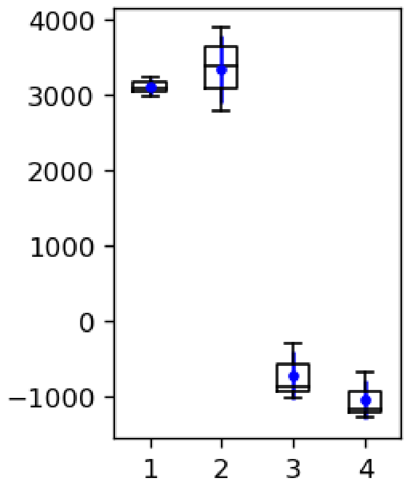
[130]:
```
bregma_dist_X = [df_lhs_vm1['Dist_From_Bregma_X'].
↪values,df_rhs_vm1['Dist_From_Bregma_X'].
↪values,df_lhs_vs1['Dist_From_Bregma_X'].
↪values,df_rhs_vs1['Dist_From_Bregma_X'].values]

fig, axes = plt.subplots(nrows=1,ncols=1,␣
↪sharex=True,sharey=True,figsize=(6*cm,7*cm),dpi=120)
b1 = axes.boxplot(bregma_dist_X)
eb1 =axes.errorbar(x=[1,2,3,4],y=np.array(bregma_dist_X).mean(axis=1),yerr=np.
↪array(bregma_dist_X).std(axis=1),\
             color='blue',marker='.',fmt='.')

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/bregma_dist_X.
↪eps')
```



[131]:
```
np.array(bregma_dist_X).mean(axis=1),np.array(bregma_dist_X).std(axis=1),np.
↪array(bregma_dist_X).std(axis=1)/np.array(bregma_dist_X).mean(axis=1)
```

[131]:
```
(array([2183.58813574, 2431.96066365, 4677.67488569, 5285.45469656]),
 array([ 53.49385142, 143.66251302,  93.98607431,  70.32568205]),
 array([0.02449814, 0.05907271, 0.02009248, 0.01330551]))
```

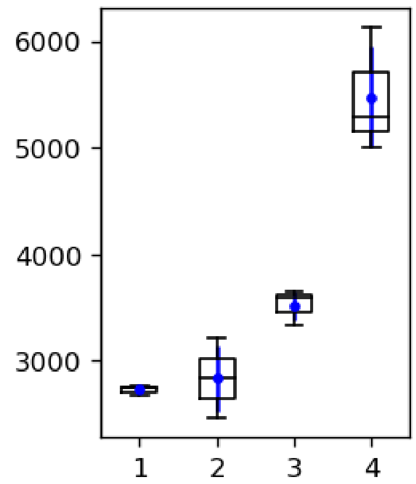
[132]:
```
bregma_dist_Y = [df_lhs_vm1['Dist_From_Bregma_Y'].
↪values,df_rhs_vm1['Dist_From_Bregma_Y'].
↪values,df_lhs_vs1['Dist_From_Bregma_Y'].
↪values,df_rhs_vs1['Dist_From_Bregma_Y'].values]

fig, axes = plt.subplots(nrows=1,ncols=1,␣
↪sharex=True,sharey=True,figsize=(6*cm,7*cm),dpi=120)
b1 = axes.boxplot(bregma_dist_Y)
eb1 =axes.errorbar(x=[1,2,3,4],y=np.array(bregma_dist_Y).mean(axis=1),yerr=np.
↪array(bregma_dist_Y).std(axis=1),\
             color='blue',marker='.',fmt='.')
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/bregma_dist_Y.
↪eps')
```



[133]:
```
np.array(bregma_dist_Y).mean(axis=1),np.array(bregma_dist_Y).std(axis=1),np.
↪array(bregma_dist_Y).std(axis=1)/np.array(bregma_dist_Y).mean(axis=1)
```

[133]:
```
(array([ 3107.42204459,  3350.52466476,  -723.63727363, -1041.33708555]),
 array([104.27327944, 447.38297608, 313.00272124, 255.77321977]),
 array([ 0.0335562 ,  0.13352624, -0.43254091, -0.24562   ]))
```

[134]:
```
bregma_dist_Z = [df_lhs_vm1['Dist_From_Bregma_Z'].
↪values,df_rhs_vm1['Dist_From_Bregma_Z'].
↪values,df_lhs_vs1['Dist_From_Bregma_Z'].
↪values,df_rhs_vs1['Dist_From_Bregma_Z'].values]

fig, axes = plt.subplots(nrows=1,ncols=1,␣
↪sharex=True,sharey=True,figsize=(6*cm,7*cm),dpi=120)
b1 = axes.boxplot(bregma_dist_Z)
eb1 =axes.errorbar(x=[1,2,3,4],y=np.array(bregma_dist_Z).mean(axis=1),yerr=np.
↪array(bregma_dist_Z).std(axis=1),\
             color='blue',marker='.',fmt='.')
plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_1/Rabies_Distribution/variability/bregma_dist_Z.
↪eps')
```



[135]:
```
np.array(bregma_dist_Z).mean(axis=1),np.array(bregma_dist_Z).std(axis=1),np.
↪array(bregma_dist_Z).std(axis=1)/np.array(bregma_dist_Z).mean(axis=1)
```

[135]:
```
(array([2728.60400171, 2838.59740857, 3522.2194169 , 5475.25995822]),
 array([ 36.93982638, 306.59418259, 136.00979609, 472.35376431]),
 array([0.01353799, 0.10800904, 0.0386148 , 0.08627056]))
```

[129]:
```
# Comparing location with others
```

4 Layer Borders

4.1 Comparison of layer borders with others

4.1.1 vS1

[24]:
```
# vS1
df = pd.DataFrame()
df['Layers'] = ['L1','L2/3','L4','L5','L6']

df['Mean'] = [4,26,40,66,100]
df['STD'] = [0.8,2,2,1,0]

df.to_csv(input_path+'Fig_2/vS1_Layer_Borders_Meyer.csv')
```

[25]:
```
df_meyer = pd.read_csv(input_path+'Fig_2/vS1_Layer_Borders_Meyer.
↪csv',index_col=0)
```

[26]:
```
df_meyer['Mean'].values
```

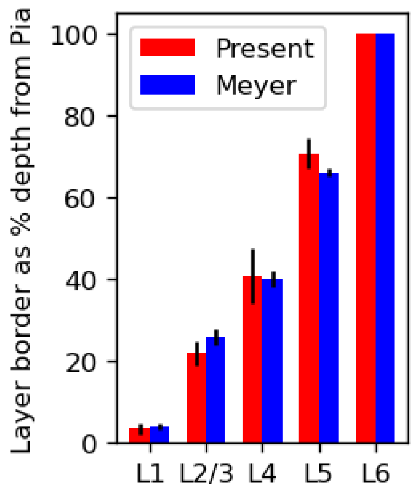
[26]:
```
array([  4,  26,  40,  66, 100], dtype=int64)
```

[18]:
```
df_mine = pd.read_csv(input_path+'Final_Data/Layer_Borders_V2/vS1_Borders.
↪csv',index_col=0)
df_mine.mean()
```

[18]:
```
L1_Border            -75.000000
L2_3_Border         -470.833333
L4_Border           -879.166667
L5_Border          -1520.833333
L1_Border_Per          0.034884
L2_3_Border_Per        0.218992
L4_Border_Per          0.408915
L5_Border_Per          0.707364
dtype: float64
```

[34]:
```
N=5
nd = np.arange(N)
width = 0.35

fig = plt.figure(figsize=(6*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind, [df_mine.mean()['L1_Border_Per']*100,df_mine.
↪mean()['L2_3_Border_Per']*100,df_mine.mean()['L4_Border_Per']*100,\
                     df_mine.mean()['L5_Border_Per']*100,100], width,␣
↪color='red', \
                yerr=[df_mine.std()['L1_Border_Per']*100,df_mine.
↪std()['L2_3_Border_Per']*100,df_mine.std()['L4_Border_Per']*100,\
                     df_mine.std()['L5_Border_Per']*100,0])

rects2 = ax.bar(ind+width, df_meyer['Mean'].values, width, color='blue',␣
↪yerr=df_meyer['STD'].values)

# add some
ax.set_ylabel('Layer border as % depth from Pia')
#ax.set_title('BF Layer border comparison')
ax.set_xticks(ind + width / 2)
ax.set_xticklabels( ('L1', 'L2/3', 'L4', 'L5', 'L6') )

ax.legend( (rects1[0], rects2[0]), ('Present', 'Meyer') )
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/vS1_Border_Comparison.eps')
```

```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```



4.1.2 vM1

[35]:
```
# vS1
df = pd.DataFrame()
df['Layers'] = ['L1','L2/3','LA','LB','L6']

df['Mean'] = [10,16,34,67,100]
df['STD'] = [1,1,2,0,0]

df.to_csv(input_path+'Fig_2/vM1_Layer_Borders_Shepherd.csv')
```

[39]:
```
df_hooks = pd.read_csv(input_path+'Fig_2/vM1_Layer_Borders_Shepherd.
↪csv',index_col=0)
```

[40]:
```
df_hooks
```

[40]:

	Layers	Mean	STD
0	L1	10	1
1	L2/3	16	1
2	LA	34	2
3	LB	67	0
4	L6	100	0

[41]:
```
df_mine = pd.read_csv(input_path+'Final_Data/Layer_Borders/vM1_Borders.
↪csv',index_col=0)
df_mine.mean()
```

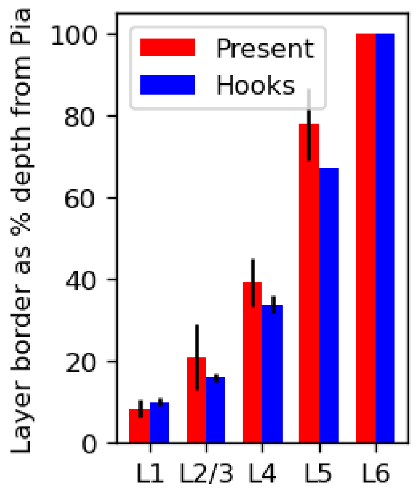
[41]:
```
L1_Border          -222.093023
L2_3_Border        -544.186047
L5A_Border        -1017.571059
```

```
L5B_Border        -2025.968992
L1_Border_Per         0.085420
L2_3_Border_Per       0.209302
L5A_Border_Per        0.391373
L5B_Border_Per        0.779219
dtype: float64
```

[42]:

```
N=5
nd = np.arange(N)
width = 0.35

fig = plt.figure(figsize=(6*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind, [df_mine.mean()['L1_Border_Per']*100,df_mine.
 ↪mean()['L2_3_Border_Per']*100,df_mine.mean()['L5A_Border_Per']*100,\
                     df_mine.mean()['L5B_Border_Per']*100,100], width,␣
 ↪color='red', \
                yerr=[df_mine.std()['L1_Border_Per']*100,df_mine.
 ↪std()['L2_3_Border_Per']*100,df_mine.std()['L5A_Border_Per']*100,\
                     df_mine.std()['L5B_Border_Per']*100,0])

rects2 = ax.bar(ind+width, df_hooks['Mean'].values, width, color='blue',␣
 ↪yerr=df_hooks['STD'].values)

# add some
ax.set_ylabel('Layer border as % depth from Pia')
#ax.set_title('BF Layer border comparison')
ax.set_xticks(ind + width / 2)
ax.set_xticklabels( ('L1', 'L2/3', 'L4', 'L5', 'L6') )

ax.legend( (rects1[0], rects2[0]), ('Present', 'Hooks') )
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/vM1_Border_Comparison.eps')
```

```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```



5 vM1 Laminar organization

[48]:
```
# show layer border variability
df = pd.read_csv(input_path+'Final_Data/Layer_Borders/vM1_Borders.
↪csv',index_col=0)
df.columns
```

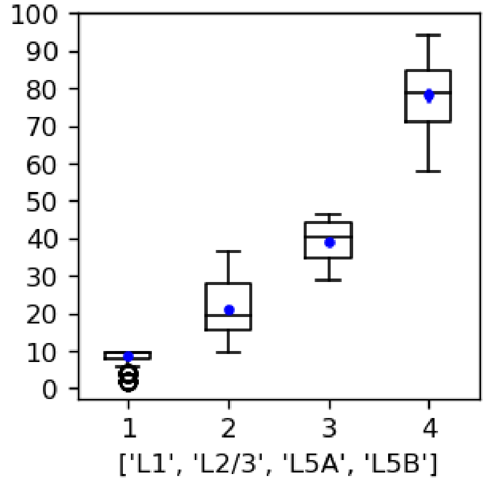
[48]:
```
Index(['Exp_Name', 'L1_Border', 'L2_3_Border', 'L5A_Border', 'L5B_Border',
       'L1_Border_Per', 'L2_3_Border_Per', 'L5A_Border_Per', 'L5B_Border_Per'],
      dtype='object')
```

[121]:
```
mg48_lhs_df = df[df['Exp_Name']=='MG48_lhs']
mg48_rhs_df = df[df['Exp_Name']=='MG48_rhs']
mg49_lhs_df = df[df['Exp_Name']=='MG49_lhs']
mg49_rhs_df = df[df['Exp_Name']=='MG49_rhs']
mg50_lhs_df = df[df['Exp_Name']=='MG50_lhs']
mg50_rhs_df = df[df['Exp_Name']=='MG50_rhs']
```

[122]:
```
l1_means = [mg48_lhs_df['L1_Border_Per'].mean(),mg48_rhs_df['L1_Border_Per'].
↪mean(),mg49_lhs_df['L1_Border_Per'].mean(),\
            mg49_rhs_df['L1_Border_Per'].mean(),mg50_lhs_df['L1_Border_Per'].
↪mean(),mg50_rhs_df['L1_Border_Per'].mean()]

l2_3_means = [mg48_lhs_df['L2_3_Border_Per'].
↪mean(),mg48_rhs_df['L2_3_Border_Per'].mean(),mg49_lhs_df['L2_3_Border_Per'].
↪mean(),\
```

```
            mg49_rhs_df['L2_3_Border_Per'].
↪mean(),mg50_lhs_df['L2_3_Border_Per'].mean(),mg50_rhs_df['L2_3_Border_Per'].
↪mean()]

l5A_means = [mg48_lhs_df['L5A_Border_Per'].mean(),mg48_rhs_df['L5A_Border_Per'].
↪mean(),mg49_lhs_df['L5A_Border_Per'].mean(),\
            mg49_rhs_df['L5A_Border_Per'].mean(),mg50_lhs_df['L5A_Border_Per'].
↪mean(),mg50_rhs_df['L5A_Border_Per'].mean()]

l5B_means = [mg48_lhs_df['L5B_Border_Per'].mean(),mg48_rhs_df['L5B_Border_Per'].
↪mean(),mg49_lhs_df['L5B_Border_Per'].mean(),\
            mg49_rhs_df['L5B_Border_Per'].mean(),mg50_lhs_df['L5B_Border_Per'].
↪mean(),mg50_rhs_df['L5B_Border_Per'].mean()]
```

[133]:
```
per_layer_borders_df = df[['L1_Border_Per', 'L2_3_Border_Per',␣
↪'L5A_Border_Per', 'L5B_Border_Per']]*100
fig, axes = plt.subplots(nrows=1,ncols=1,␣
↪sharex=True,sharey=True,figsize=(7*cm,7*cm),dpi=120)
b1 = axes.boxplot(per_layer_borders_df)

eb1 =axes.errorbar(x=[1,2,3,4],y=[np.array(l1_means).mean()*100,np.
↪array(l2_3_means).mean()*100,np.array(l5A_means).mean()*100,np.
↪array(l5B_means).mean()*100],\
                   yerr=[np.array(l1_means).std()*100,np.array(l2_3_means).
↪std()*100,np.array(l5A_means).std()*100,np.array(l5B_means).std()*100],\
             color='blue',marker='.',fmt='.')

plt.yticks(np.arange(0,110,10))
plt.xlabel(['L1','L2/3','L5A','L5B'])

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/laminar_variability.eps')
```



[126]:
```
np.array(l1_means).mean(),np.array(l1_means).std()
```

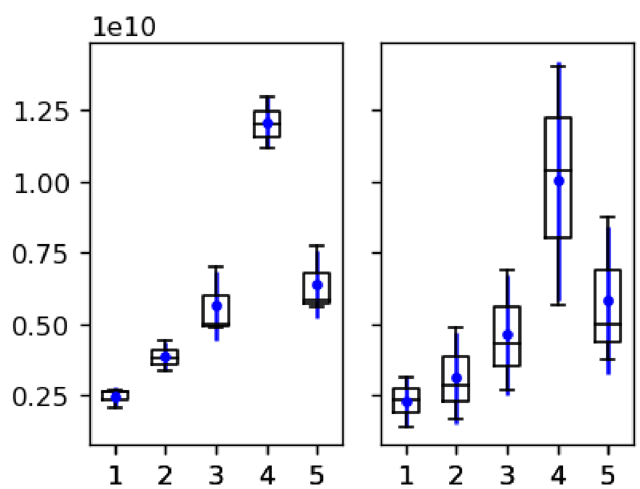
[126]: (0.08623863020839255, 0.011960957728402168)

5.1 3d counts and densities

[47]:
```
df_2d_counts = pd.read_csv(input_path+'Fig_2/Layerwise_2D_Cell_Counts.csv')
df_3d_counts = pd.read_csv(input_path+'Fig_2/Layerwise_3D_Cell_Counts.csv')
df_3d_densities = pd.read_csv(input_path+'Fig_2/Layerwise_3D_Densities.csv')
df_layerwise_volumes = pd.read_csv(input_path+'Fig_2/Layer_Volumes.csv')

df_3d_counts_lhs = df_3d_counts[df_3d_counts['Hem']=='lhs']
df_3d_counts_rhs = df_3d_counts[df_3d_counts['Hem']=='rhs']

df_3d_densities_lhs = df_3d_densities[df_3d_densities['Hem']=='lhs']
df_3d_densities_rhs = df_3d_densities[df_3d_densities['Hem']=='rhs']

df_layerwise_volumes_lhs =␣
↪df_layerwise_volumes[(df_layerwise_volumes['Exp']=='MG48_lhs') |␣
↪(df_layerwise_volumes['Exp']=='MG49_lhs') |␣
↪(df_layerwise_volumes['Exp']=='MG50_lhs')]
df_layerwise_volumes_rhs =␣
↪df_layerwise_volumes[(df_layerwise_volumes['Exp']=='MG48_rhs') |␣
↪(df_layerwise_volumes['Exp']=='MG49_rhs') |␣
↪(df_layerwise_volumes['Exp']=='MG50_rhs')]
```

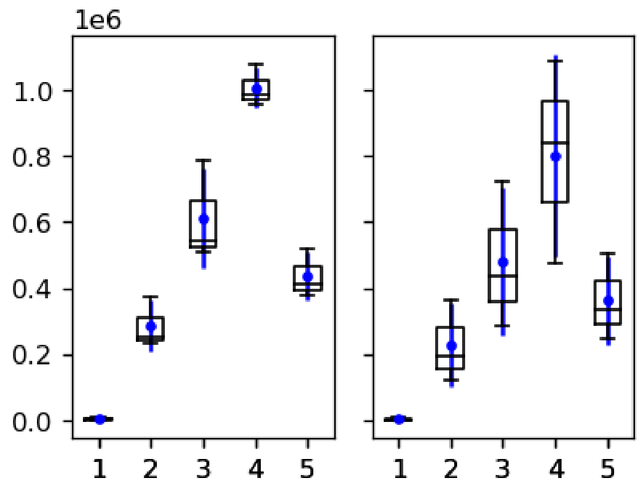
[48]:
```
fig, axes = plt.subplots(nrows=1,ncols=2,␣
↪sharex=True,sharey=True,figsize=(9*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(df_layerwise_volumes_lhs[['L1','L2_3','L5A','L5B','L6']])
```

```
eb1 =axes[0].
 ↪errorbar(x=[1,2,3,4,5],y=df_layerwise_volumes_lhs[['L1','L2_3','L5A','L5B','L6']].
 ↪mean(),\
                    ␣
 ↪yerr=df_layerwise_volumes_lhs[['L1','L2_3','L5A','L5B','L6']].std(),\
             color='blue',marker='.',fmt='.')

b2 = axes[1].boxplot(df_layerwise_volumes_rhs[['L1','L2_3','L5A','L5B','L6']])
eb2 =axes[1].
 ↪errorbar(x=[1,2,3,4,5],y=df_layerwise_volumes_rhs[['L1','L2_3','L5A','L5B','L6']].
 ↪mean(),\
                    ␣
 ↪yerr=df_layerwise_volumes_rhs[['L1','L2_3','L5A','L5B','L6']].std(),\
             color='blue',marker='.',fmt='.')

#plt.yticks(np.arange(0,110,10))
#plt.xlabel(['L1','L2/3','L5A','L5B'])

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()

plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/layerwise_volumes.eps')
```



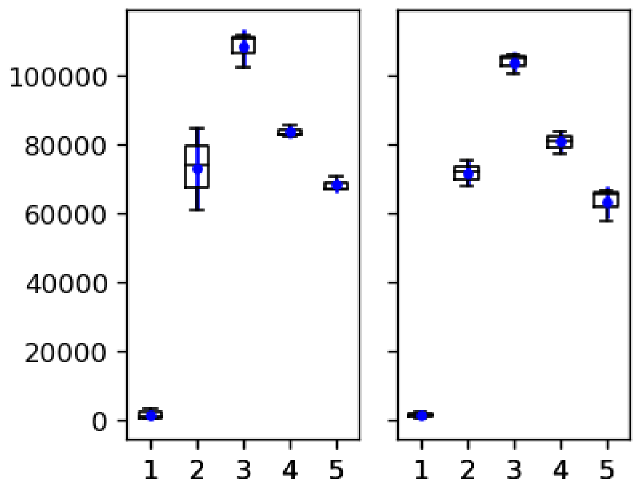
[49]:

```
fig, axes = plt.subplots(nrows=1,ncols=2,␣
↪sharex=True,sharey=True,figsize=(9*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(df_3d_counts_lhs[['L1','L2_3','L5A','L5B','L6']])
eb1 =axes[0].
↪errorbar(x=[1,2,3,4,5],y=df_3d_counts_lhs[['L1','L2_3','L5A','L5B','L6']].
↪mean(),\
                   yerr=df_3d_counts_lhs[['L1','L2_3','L5A','L5B','L6']].std(),\
             color='blue',marker='.',fmt='.')

b2 = axes[1].boxplot(df_3d_counts_rhs[['L1','L2_3','L5A','L5B','L6']])
eb2 =axes[1].
↪errorbar(x=[1,2,3,4,5],y=df_3d_counts_rhs[['L1','L2_3','L5A','L5B','L6']].
↪mean(),\
                   yerr=df_3d_counts_rhs[['L1','L2_3','L5A','L5B','L6']].std(),\
             color='blue',marker='.',fmt='.')

#plt.yticks(np.arange(0,110,10))
#plt.xlabel(['L1','L2/3','L5A','L5B'])

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()

plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/layerwise_3d_cellcounts.eps')
```



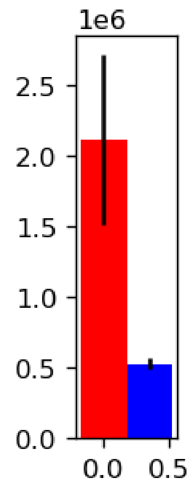
[50]:
```
fig, axes = plt.subplots(nrows=1,ncols=2,␣
↪sharex=True,sharey=True,figsize=(9*cm,7*cm),dpi=120)
b1 = axes[0].boxplot(df_3d_densities_lhs[['L1','L2_3','L5A','L5B','L6']])
eb1 =axes[0].
↪errorbar(x=[1,2,3,4,5],y=df_3d_densities_lhs[['L1','L2_3','L5A','L5B','L6']].
↪mean(),\
                   yerr=df_3d_densities_lhs[['L1','L2_3','L5A','L5B','L6']].
↪std(),\
             color='blue',marker='.',fmt='.')

b2 = axes[1].boxplot(df_3d_densities_rhs[['L1','L2_3','L5A','L5B','L6']])
eb2 =axes[1].
↪errorbar(x=[1,2,3,4,5],y=df_3d_densities_rhs[['L1','L2_3','L5A','L5B','L6']].
↪mean(),\
                   yerr=df_3d_densities_rhs[['L1','L2_3','L5A','L5B','L6']].
↪std(),\
             color='blue',marker='.',fmt='.')

#plt.yticks(np.arange(0,110,10))
#plt.xlabel(['L1','L2/3','L5A','L5B'])

plt.setp(b1['boxes'], color='black')
plt.setp(b1['whiskers'], color='black')
plt.setp(b1['medians'], color='black')
plt.setp(b1['caps'], color='black')
plt.tight_layout()

plt.setp(b2['boxes'], color='black')
plt.setp(b2['whiskers'], color='black')
```

```
plt.setp(b2['medians'], color='black')
plt.setp(b2['caps'], color='black')
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/layerwise_3d_densities.eps')
```

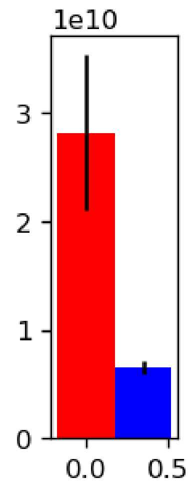


5.2 Comapare counts with vS1 counts

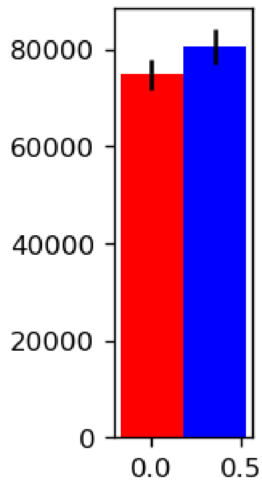
[55]:
```
N=1
ind = np.arange(N)
width = 0.35

fig = plt.figure(figsize=(3*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind, df_3d_counts.sum(axis=1).mean(), width, color='red',␣
↪yerr=df_3d_counts.sum(axis=1).std())

rects2 = ax.bar(ind+width, 529715, width, color='blue', yerr=39104)

# add some
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/3D_Count_Comparison.eps')
```



[61]:
```
volumes = df_volumes[df_volumes['BR']=='vM1']['surf_vol'].values

N=1
ind = np.arange(N)
width = 0.35

fig = plt.figure(figsize=(3*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind, volumes.mean(), width, color='red', yerr=volumes.std())

rects2 = ax.bar(ind+width, 6.6*1e9, width, color='blue', yerr=0.58*1e9)

# add some
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/Volume_Comparison.eps')
```



[64]:
```
densities = []
df_volumes = pd.read_csv(input_path+'Fig_1/Rabies_Distribution/Variability/
 ↪variability.csv')
for exp in ['MG48','MG49','MG50']:
    for hem in ['lhs','rhs']:
        vol = df_volumes[(df_volumes['Exp']==exp) & (df_volumes['Hem']==hem) &␣
 ↪(df_volumes['BR']=='vM1')]['surf_vol'].values[0]
        counts = df_3d_counts[(df_3d_counts['Exp']==exp) &␣
 ↪(df_3d_counts['Hem']==hem) ].sum(axis=1).values[0]
        densities.append(counts*1e9/vol)

N=1
ind = np.arange(N)
width = 0.35

fig = plt.figure(figsize=(4*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind, np.array(densities).mean(), width, color='red', yerr=np.
 ↪array(densities).std())

rects2 = ax.bar(ind+width, 80419, width, color='blue', yerr=3688)

# add some
plt.tight_layout()
fig.savefig(input_path + 'Fig_2/Densities_Comparison.eps')
```



6 Global v/s Local Ref Frame Comparison

[523]:
```
df_global =pd.read_csv('Z:/V11/vM1_Ref_Frame/Outputs/Alignment_Analysis/
↪Global_Alignment_stats.csv')
df_vm1 =pd.read_csv('Z:/V11/vM1_Ref_Frame/Outputs/Alignment_Analysis/vM1_stats.
↪csv')
df_s1bf =pd.read_csv('Z:/V11/vM1_Ref_Frame/Outputs/Alignment_Analysis/
↪S1BF_stats.csv')
```

[524]:
```
df_global.columns
```

[524]:
```
Index(['Unnamed: 0', 'Angular_Error_SEM', 'Angular_Precision_Mean',
       'Angular_Precision_Std', 'BR', 'Barrel_Cortex_Center_Ellipsoid',
       'Barrel_Cortex_Center_Ellipsoid_Norm',
       'Barrel_Cortex_Center_Error_Dists_SEM',
       'Barrel_Cortex_Center_Precision_Mean',
       'Barrel_Cortex_Center_Precision_Std', 'Barrels_Angular_Error_SEM',
       'Barrels_Angular_Precision_Mean', 'Barrels_Angular_Precision_Std',
       'Barrels_Barrel_Center_Precision_Mean',
       'Barrels_Barrel_Center_Precision_Std',
       'Barrels_Barrel_Centers_Error_Ellipsoid_Mean',
       'Barrels_Barrel_Centers_Error_Ellipsoid_Mean_Norm',
       'Barrels_Barrel_Centers_Error_Ellipsoid_Std',
       'Barrels_Barrel_Centers_Error_Ellipsoid_Std_Norm',
       'Barrels_Column_Center_Precision_Mean',
       'Barrels_Column_Center_Precision_Std',
       'Barrels_Column_Error_Ellipsoid_Mean',
       'Barrels_Column_Error_Ellipsoid_Mean_Norm',
       'Barrels_Column_Error_Ellipsoid_Std',
```

```
 'Barrels_Column_Error_Ellipsoid_Std_Norm',
 'Barrels_Pia_Error_Ellipsoid_Mean',
 'Barrels_Pia_Error_Ellipsoid_Mean_Norm',
 'Barrels_Pia_Error_Ellipsoid_Std',
 'Barrels_Pia_Error_Ellipsoid_Std_Norm', 'Barrels_Pia_Precision_Mean',
 'Barrels_Pia_Precision_std', 'Barrels_WM_Error_Ellipsoid_Mean',
 'Barrels_WM_Error_Ellipsoid_Mean_Norm',
 'Barrels_WM_Error_Ellipsoid_Std', 'Barrels_WM_Error_Ellipsoid_Std_Norm',
 'Barrels_WM_Precision_Mean', 'Barrels_WM_Precision_Std', 'Hem',
 'Inner_Alignment_Type', 'Number_Of_Axis_Used', 'Outer_Alignment_Type',
 'Overall_Rabies_landmarks', 'Overall_Volume',
 'Overlap_Rabies_landmarks', 'Overlap_Volume',
 'PCA_0_Angle_Between_Axes_Mean', 'PCA_0_Angle_Between_Axes_Std',
 'PCA_1_Angle_Between_Axes_Mean', 'PCA_1_Angle_Between_Axes_Std',
 'PCA_2_Angle_Between_Axes_Mean', 'PCA_2_Angle_Between_Axes_Std',
 'PCA_Angular_Precision_Mean', 'PCA_Angular_Precision_SEM',
 'PCA_Angular_Precision_Std', 'Pia_Error_Dist_SEM',
 'Pia_Error_Ellipsoid_Mean', 'Pia_Error_Ellipsoid_Mean_Norm',
 'Pia_Error_Ellipsoid_Std', 'Pia_Error_Ellipsoid_Std_Norm',
 'Pia_Precision_Mean', 'Pia_Precision_Std', 'Rabies_Center_Ellipsoid',
 'Rabies_Center_Ellipsoid_Norm', 'Rabies_Center_Error_Dists_SEM',
 'Rabies_Center_Precision_Mean', 'Rabies_Center_Precision_Std',
 'Rabies_Landmarks_Overlap_Percentage', 'Rabies_Surf_Center_Dists_Mean',
 'Rabies_Surf_Center_Dists_Std', 'Rabies_Surf_Center_Ellipsoid',
 'Rabies_Surf_Center_Ellipsoid_Norm',
 'Rabies_Surf_Center_Error_Dists_SEM', 'Surf_Center_Ellipsoid',
 'Surf_Center_Ellipsoid_Norm', 'Surf_Center_Error_Dists_SEM',
 'Surf_Center_Precision_Mean', 'Surf_Center_Precision_Std',
 'Volume_Overlap_Percentage', 'WM_Error_Dist_SEM',
 'WM_Error_Ellipsoid_Mean', 'WM_Error_Ellipsoid_Mean_Norm',
 'WM_Precision_Mean', 'WM_Precision_Std'],
dtype='object')
```

6.1 Center Precisions

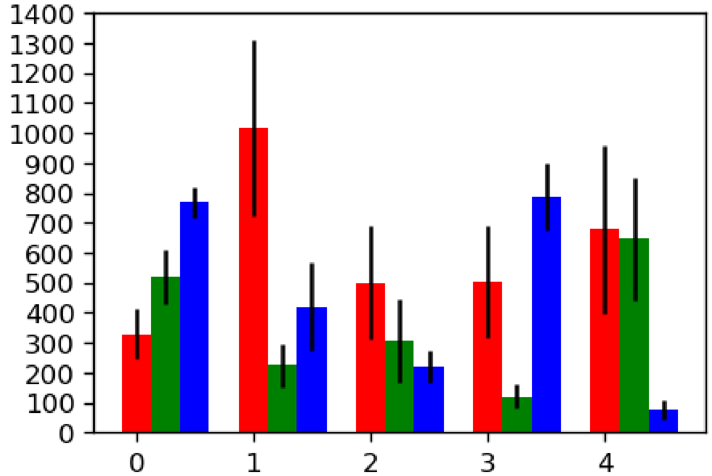
[525]:
```
N=5
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(10*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.
↪bar(ind,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],\
```

```
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().
↪values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],\
                    ],\
                     width, color='red', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0],\
                     ],\
                     )

rects2 = ax.
↪bar(ind+width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Mean',]].mean().
↪values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Mean',]].mean().
↪values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Mean',]].mean().
↪values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vM1')][['Rabies_Surf_Center_Dists_Mean',]].mean().values[0],\
```

```
                         ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Rabies_Surf_Center_Dists_Mean',]].mean().
↪values[0],\
                        ],\
                         width, color='green', \
                   ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Std',]].mean().
↪values[0],\
                         ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Std',]].mean().
↪values[0],\
                         ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Rabies_Surf_Center_Dists_Std',]].mean().
↪values[0],\
                         df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vM1')][['Rabies_Surf_Center_Dists_Std',]].mean().values[0],\
                         ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Rabies_Surf_Center_Dists_Std',]].mean().values[0],\
                         ],\
                         )

rects3 = ax.
↪bar(ind+2*width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['Barrels_Barrel_Center_Precision_Mean',]].
↪mean().values[0],\
                         ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Barrels_Barrel_Center_Precision_Mean',]].mean().
↪values[0],\
                         ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Barrels_Barrel_Center_Precision_Mean',]].
↪mean().values[0],\
                         df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vM1')][['Barrels_Barrel_Center_Precision_Mean',]].mean().
↪values[0],\
                         ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Barrels_Barrel_Center_Precision_Mean',]].mean().
↪values[0],\
                        ],\
```

```
                    width, color='blue', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vS1')][['Barrels_Barrel_Center_Precision_Std',]].mean().
↪values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vS1')][['Barrels_Barrel_Center_Precision_Std',]].mean().
↪values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vS1')][['Barrels_Barrel_Center_Precision_Std',]].mean().
↪values[0],\
                    df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vS1')][['Barrels_Barrel_Center_Precision_Std',]].mean().
↪values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &␣
↪(df_s1bf['BR']=='vS1')][['Barrels_Barrel_Center_Precision_Std',]].mean().
↪values[0],\
                    ],\
                    )

ticks = plt.yticks(np.arange(0,1500,100))
#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#               ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects1 = ax.bar(ind,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='red', \
#               ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
```

```
#                              ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

plt.tight_layout()
fig.savefig(input_path + 'Fig_3/Centers_Precision.eps')
```



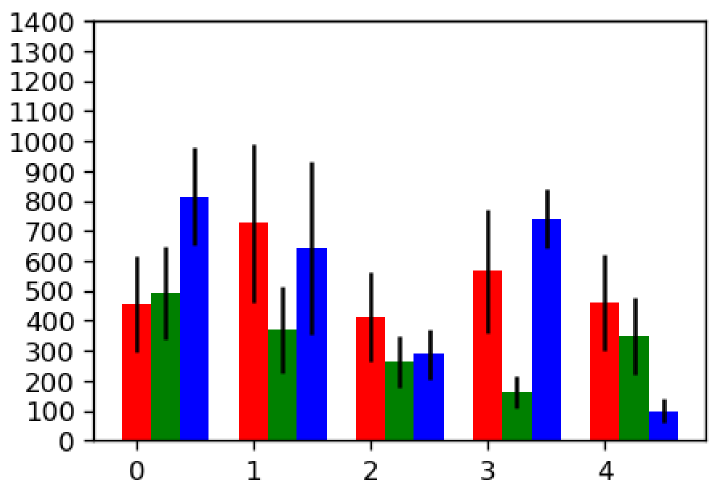
6.2 Pia Precisions

[526]:
```
N=5
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(10*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.
↪bar(ind,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Pia_Precision_Mean',]].mean().values[0],\
                                  ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Pia_Precision_Mean',]].mean().values[0],\
                                  ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['Pia_Precision_Mean',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='WB')][['Pia_Precision_Mean',]].mean().values[0],\
```

```
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='WB')][['Pia_Precision_Mean',]].mean().values[0],\
                   ],\
                    width, color='red', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Pia_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Pia_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['Pia_Precision_Std',]].mean().values[0],\
                    df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='WB')][['Pia_Precision_Std',]].mean().values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='WB')][['Pia_Precision_Std',]].mean().values[0],\
                    ],\
                    )

rects2 = ax.
↪bar(ind+width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Center')␣
↪& (df_global['BR']=='vM1')][['Pia_Precision_Mean',]].values[0][0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Pia_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Pia_Precision_Mean',]].mean().values[0],\
                    df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['Pia_Precision_Mean',]].values[0][0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Pia_Precision_Mean',]].mean().values[0],\
                   ],\
                    width, color='green', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_global['BR']=='vM1')][['Pia_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Pia_Precision_Std',]].mean().values[0],\
```

```
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Pia_Precision_Std',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['Pia_Precision_Std',]].values[0][0],\
                     ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Pia_Precision_Std',]].mean().values[0],\
                     ],\
                     )

rects3 = ax.
↪bar(ind+2*width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]].mean().
↪values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]].mean().
↪values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]].mean().values[0],\
                     ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]].mean().values[0],\
                    ],\
                     width, color='blue', \
                ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vS1')][['Barrels_Pia_Precision_std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vS1')][['Barrels_Pia_Precision_std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vS1')][['Barrels_Pia_Precision_std',]].mean().
↪values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vS1')][['Barrels_Pia_Precision_std',]].mean().values[0],\
                     ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vS1')][['Barrels_Pia_Precision_std',]].mean().values[0],\
                     ],\
                     )
```

```
ticks = plt.yticks(np.arange(0,1500,100))
#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects1 = ax.bar(ind,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='red', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

plt.tight_layout()
fig.savefig(input_path + 'Fig_3/Pia_Precision.eps')
```

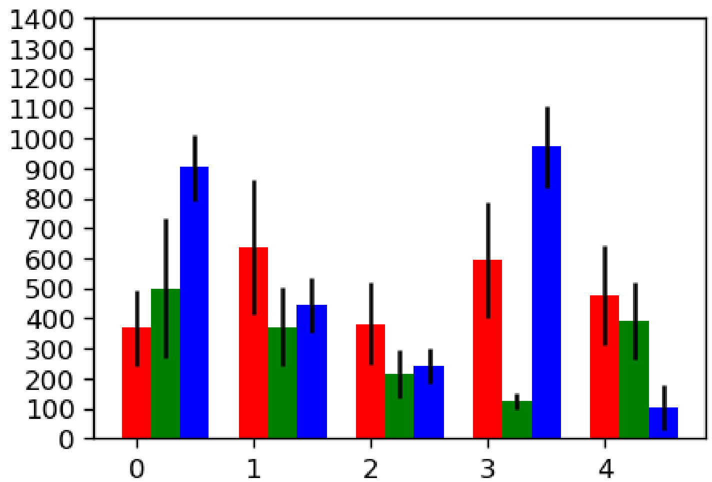


6.3 WM Precisions

[527]:
```
N=5
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(10*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.
↪bar(ind,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['WM_Precision_Mean',]].mean().values[0],\
                       ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['WM_Precision_Mean',]].mean().values[0],\
                       ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['WM_Precision_Mean',]].mean().values[0],\
                        df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='WB')][['WM_Precision_Mean',]].mean().values[0],\
                       ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='WB')][['WM_Precision_Mean',]].mean().values[0],\
                       ],\
                        width, color='red', \
                  ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['WM_Precision_Std',]].mean().values[0],\
                       ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['WM_Precision_Std',]].mean().values[0],\
                       ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['WM_Precision_Std',]].mean().values[0],\
                        df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='WB')][['WM_Precision_Std',]].mean().values[0],\
                       ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='WB')][['WM_Precision_Std',]].mean().values[0],\
                        ],\
                        )

rects2 = ax.
↪bar(ind+width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Center')␣
↪& (df_global['BR']=='vM1')][['WM_Precision_Mean',]].mean().values[0],\
```

```
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['WM_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['WM_Precision_Mean',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['WM_Precision_Mean',]].values[0][0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['WM_Precision_Mean',]].mean().values[0],\
                    ],\
                     width, color='green', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_global['BR']=='vM1')][['WM_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['WM_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['WM_Precision_Std',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['WM_Precision_Std',]].values[0][0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['WM_Precision_Std',]].mean().values[0],\
                     ],\
                     )

rects3 = ax.
↪bar(ind+2*width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['Barrels_WM_Precision_Mean',]].mean().
↪values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Barrels_WM_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Barrels_WM_Precision_Mean',]].mean().
↪values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vM1')][['Barrels_WM_Precision_Mean',]].mean().values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Barrels_WM_Precision_Mean',]].mean().values[0],\
```

```
                    ],\
                     width, color='blue', \
               ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vS1')][['Barrels_WM_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vS1')][['Barrels_WM_Precision_Std',]].mean().values[0],\
                    ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vS1')][['Barrels_WM_Precision_Std',]].mean().values[0],\
                     df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &␣
↪(df_vm1['BR']=='vS1')][['Barrels_WM_Precision_Std',]].mean().values[0],\
                    ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vS1')][['Barrels_WM_Precision_Std',]].mean().values[0],\
                     ],\
                     )

ticks = plt.yticks(np.arange(0,1500,100))
#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#               ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects1 = ax.bar(ind,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='red', \
#               ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#               ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

plt.tight_layout()
fig.savefig(input_path + 'Fig_3/WM_Precision.eps')
```



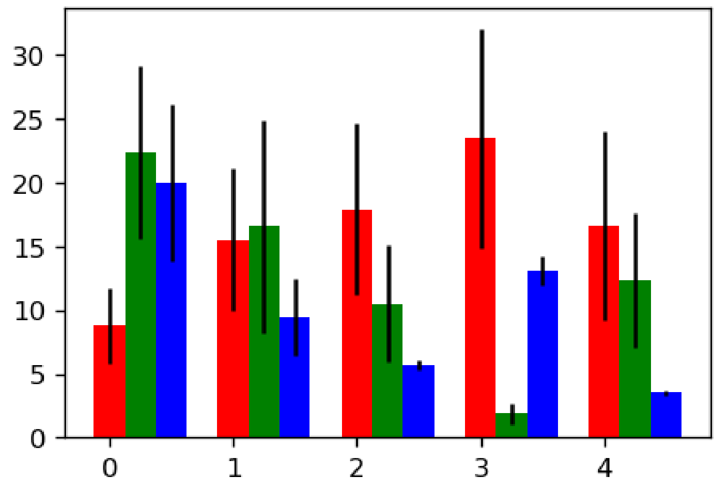
6.4 Angular Precisions

[541]:
```
N=5
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(10*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.
 ↪bar(ind,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
 ↪(df_global['BR']=='WB')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                         ␣
 ↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
 ↪(df_global['BR']=='WB')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                         ␣
 ↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
 ↪& (df_global['BR']=='WB')][['PCA_Angular_Precision_Mean',]].mean().
 ↪values[0],\
                         ␣
 ↪df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Rabies_Surface') &␣
 ↪(df_vm1['BR']=='WB')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                         ␣
 ↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
 ↪(df_s1bf['BR']=='WB')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                         ],\
                          width, color='red', \
```

```
                ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='WB')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Rabies_Surface') &␣
↪(df_vm1['BR']=='WB')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='WB')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                      ],\
                      )

rects2 = ax.
↪bar(ind+width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].mean().
↪values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].mean().
↪values[0],\
                      df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].values[0][0],\
                     ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].mean().values[0],\
                     ],\
                      width, color='green', \
                ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vM1')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                     ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['PCA_Angular_Precision_Std',]].mean().
↪values[0],\
```

```
                    df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_vm1['BR']=='vM1')][['PCA_Angular_Precision_Std',]].values[0][0],\
                   ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['PCA_Angular_Precision_Std',]].mean().values[0],\
                    ],\
                    )


rects3 = ax.
↪bar(ind+2*width,[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface')␣
↪& (df_global['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean().
↪values[0],\
                   ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean().
↪values[0],\
                   ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean().
↪values[0],\
                   ␣
↪df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Rabies_Surface') &␣
↪(df_vm1['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean().
↪values[0],\
                   ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean().
↪values[0],\
                   ],\
                    width, color='blue', \
              ␣
↪yerr=[df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='vS1')][['Barrels_Angular_Precision_Std',]].mean().
↪values[0],\
                   ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='vS1')][['Barrels_Angular_Precision_Std',]].mean().
↪values[0],\
                   ␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Rabies_Surfaces')␣
↪& (df_global['BR']=='vS1')][['Barrels_Angular_Precision_Std',]].mean().
↪values[0],\
                   ␣
↪df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Rabies_Surface') &␣
↪(df_vm1['BR']=='vS1')][['Barrels_Angular_Precision_Std',]].mean().values[0],\
```

```
                          ␣
↪df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &␣
↪(df_s1bf['BR']=='vS1')][['Barrels_Angular_Precision_Std',]].mean().
↪values[0],\
                          ],\
                          )

#ticks = plt.yticks(np.arange(0,1300,100))
#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects1 = ax.bar(ind,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='red', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Surface') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

#rects2 = ax.bar(ind+width,␣
↪df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Mean',]].mean().values[0],␣
↪width, color='blue', \
#                    ␣
↪yerr=df_global[(df_global['Inner_Alignment_Type']=='Using_Local_Centers') &␣
↪(df_global['BR']=='WB')][['Surf_Center_Precision_Std',]].mean().values[0])

plt.tight_layout()
fig.savefig(input_path + 'Fig_3/Angular_Precision.eps')
```



[577]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &
 ↪(df_s1bf['BR']=='vS1')][['Barrels_Angular_Precision_Mean',]]
```

[577]:
```
   Barrels_Angular_Precision_Mean
6                        3.249892
7                        3.805283
```

[571]:
```
(1/55)*38
```

[571]: 0.6909090909090909

[ ]:
```
55
```

[576]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Using_Surface') &
 ↪(df_vm1['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]].mean()
```

[576]:
```
PCA_Angular_Precision_Mean    5.334361
dtype: float64
```

[531]:
```
df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &
 ↪(df_vm1['BR']=='vM1')][['PCA_Angular_Precision_Mean',]].mean().values[0]
```

[531]: 22.36244632510002

[532]:
```
df_vm1.columns
```

[532]:
```
Index(['Unnamed: 0', 'Angular_Error_SEM', 'Angular_Precision_Mean',
       'Angular_Precision_Std', 'BR', 'Barrel_Cortex_Center_Ellipsoid',
       'Barrel_Cortex_Center_Ellipsoid_Norm',
       'Barrel_Cortex_Center_Error_Dists_SEM',
```

```
 'Barrel_Cortex_Center_Precision_Mean',
 'Barrel_Cortex_Center_Precision_Std', 'Barrels_Angular_Error_SEM',
 'Barrels_Angular_Precision_Mean', 'Barrels_Angular_Precision_Std',
 'Barrels_Barrel_Center_Precision_Mean',
 'Barrels_Barrel_Center_Precision_Std',
 'Barrels_Barrel_Centers_Error_Ellipsoid_Mean',
 'Barrels_Barrel_Centers_Error_Ellipsoid_Mean_Norm',
 'Barrels_Barrel_Centers_Error_Ellipsoid_Std',
 'Barrels_Barrel_Centers_Error_Ellipsoid_Std_Norm',
 'Barrels_Column_Center_Precision_Mean',
 'Barrels_Column_Center_Precision_Std',
 'Barrels_Column_Error_Ellipsoid_Mean',
 'Barrels_Column_Error_Ellipsoid_Mean_Norm',
 'Barrels_Column_Error_Ellipsoid_Std',
 'Barrels_Column_Error_Ellipsoid_Std_Norm',
 'Barrels_Pia_Error_Ellipsoid_Mean',
 'Barrels_Pia_Error_Ellipsoid_Mean_Norm',
 'Barrels_Pia_Error_Ellipsoid_Std',
 'Barrels_Pia_Error_Ellipsoid_Std_Norm', 'Barrels_Pia_Precision_Mean',
 'Barrels_Pia_Precision_std', 'Barrels_WM_Error_Ellipsoid_Mean',
 'Barrels_WM_Error_Ellipsoid_Mean_Norm',
 'Barrels_WM_Error_Ellipsoid_Std', 'Barrels_WM_Error_Ellipsoid_Std_Norm',
 'Barrels_WM_Precision_Mean', 'Barrels_WM_Precision_Std', 'Hem',
 'Inner_Alignment_Type', 'Number_Of_Axis_Used', 'Outer_Alignment_Type',
 'Overall_Rabies_landmarks', 'Overall_Volume',
 'Overlap_Rabies_landmarks', 'Overlap_Volume',
 'PCA_0_Angle_Between_Axes_Mean', 'PCA_0_Angle_Between_Axes_Std',
 'PCA_1_Angle_Between_Axes_Mean', 'PCA_1_Angle_Between_Axes_Std',
 'PCA_2_Angle_Between_Axes_Mean', 'PCA_2_Angle_Between_Axes_Std',
 'PCA_Angular_Precision_Mean', 'PCA_Angular_Precision_SEM',
 'PCA_Angular_Precision_Std', 'Pia_Error_Dist_SEM',
 'Pia_Error_Ellipsoid_Mean', 'Pia_Error_Ellipsoid_Mean_Norm',
 'Pia_Error_Ellipsoid_Std', 'Pia_Error_Ellipsoid_Std_Norm',
 'Pia_Precision_Mean', 'Pia_Precision_Std', 'Rabies_Center_Ellipsoid',
 'Rabies_Center_Ellipsoid_Norm', 'Rabies_Center_Error_Dists_SEM',
 'Rabies_Center_Precision_Mean', 'Rabies_Center_Precision_Std',
 'Rabies_Landmarks_Overlap_Percentage', 'Rabies_Surf_Center_Dists_Mean',
 'Rabies_Surf_Center_Dists_Std', 'Rabies_Surf_Center_Ellipsoid',
 'Rabies_Surf_Center_Ellipsoid_Norm',
 'Rabies_Surf_Center_Error_Dists_SEM', 'Surf_Center_Ellipsoid',
 'Surf_Center_Ellipsoid_Norm', 'Surf_Center_Error_Dists_SEM',
 'Surf_Center_Precision_Mean', 'Surf_Center_Precision_Std',
 'Volume_Overlap_Percentage', 'WM_Error_Dist_SEM',
 'WM_Error_Ellipsoid_Mean', 'WM_Error_Ellipsoid_Mean_Norm',
 'WM_Precision_Mean', 'WM_Precision_Std'],
dtype='object')
```

[546]:
```
df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Surface') &
 ↪(df_vm1['BR']=='vM1')][['WM_Error_Ellipsoid_Mean',]]
#[(30.8+59.1)/2,( 54.5+106.7)/2,(61.9+51.3)/2]
```

[546]:
```
                     WM_Error_Ellipsoid_Mean
16      [49.1533137  77.38980311 78.4623976 ]
17   [ 84.96801149 108.21712732 179.83579373]
```

[557]:
```

```

[557]: 1.1636363636363636

[551]:
```
df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &
 ↪(df_vm1['BR']=='vM1')][['Rabies_Surf_Center_Ellipsoid',]]
```

[551]:
```
               Rabies_Surf_Center_Ellipsoid
4  [ 42.27441856 110.01097201  84.38809675]
5     [58.50609879 14.00859704 95.14052509]
```

[535]:
```
df_vm1[(df_vm1['Inner_Alignment_Type']=='Using_Center') &
 ↪(df_vm1['BR']=='vM1')][['Surf_Center_Precision_Mean',]].values[0][0]
```

[535]: 289.59023782377807

[536]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &
 ↪(df_s1bf['BR']=='vM1')][['Barrels_Pia_Precision_Mean',]]
```

[536]:
```
    Barrels_Pia_Precision_Mean
10                  112.001154
11                   87.909474
```

[537]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &
 ↪(df_s1bf['BR']=='vM1')][['Barrels_WM_Precision_Mean']]
```

[537]:
```
    Barrels_WM_Precision_Mean
10                 118.938618
11                  85.799177
```

[538]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Barrels_Alone') &
 ↪(df_s1bf['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]]
#[(30.8+59.1)/2,( 54.5+106.7)/2,(61.9+51.3)/2]
```

[538]:
```
   Barrels_Angular_Precision_Mean
4                        6.085805
5                        3.800505
```

[539]:
```
df_s1bf[(df_s1bf['Inner_Alignment_Type']=='Use_Projections_Alone') &
 ↪(df_s1bf['BR']=='vM1')][['Barrels_Angular_Precision_Mean',]]
```

```
#[(
```

[539]:

```
    Barrels_Angular_Precision_Mean
10                        3.249892
11                        3.805283
```

7 Density Cluster Analysis for Registration

[578]:

```
df = pd.read_csv('Z:/V11/Registration_Final_Coronal_Ref_Frame/Input_Ref_Frame/
 ↪Local_Ref_Frame_View/density_clusters.csv')
```

[580]:

```
df
```

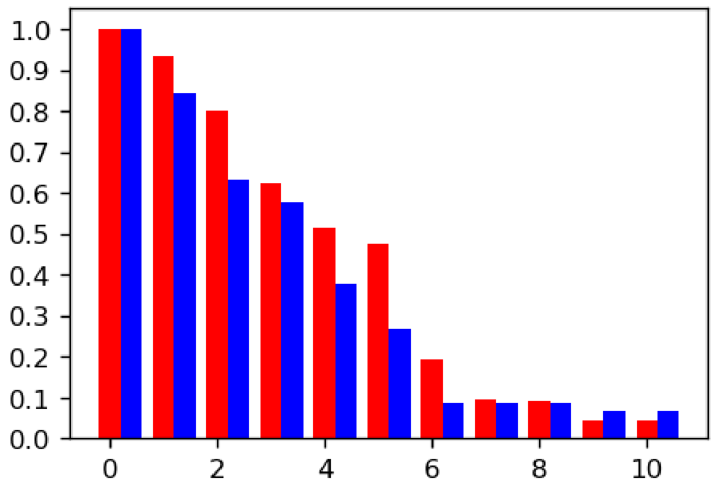
[580]:

```
    Unnamed: 0  dens_cutoff  morph_count  morph_count_per  rabies_count  \
0            0          0.0           90         1.000000          1567
1            0          0.1           76         0.844444          1462
2            0          0.2           57         0.633333          1252
3            0          0.3           52         0.577778           976
4            0          0.4           34         0.377778           804
5            0          0.5           24         0.266667           743
6            0          0.6            8         0.088889           303
7            0          0.7            8         0.088889           148
8            0          0.8            8         0.088889           143
9            0          0.9            6         0.066667            72
10           0          1.0            6         0.066667            72

    rabies_count_per
0           1.000000
1           0.932993
2           0.798979
3           0.622846
4           0.513082
5           0.474154
6           0.193363
7           0.094448
8           0.091257
9           0.045948
10          0.045948
```

[592]:

```
N=11
ind = np.arange(N)
width = 0.4

fig = plt.figure(figsize=(10*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)
```

```
rects1 = ax.bar(ind,df['rabies_count_per'].values,width, color='red',)

rects2 = ax.bar(ind+width,df['morph_count_per'].values,width, color='blue',)

plt.yticks(np.arange(0,1.1,0.1))

plt.tight_layout()
fig.savefig(input_path + 'Fig_4/density_clustering.eps')
```



8 SBF old vs new

[44]:
```
df_new = pd.read_csv('D:/vM1/Fig_2/barrel_variability.csv')
df_old = pd.read_excel('D:/vM1/Fig_2/sbf_measurements.xlsx')
df_new_orientations = pd.read_csv('D:/vM1/Fig_2/barrel_orientations.csv')
```

[21]:
```
df_new['Barrel_Top'].mean(),df_new['Barrel_Top'].std(),df_old['Barrel_Top'].
 ↪mean(),df_old['Barrel_Top'].std()
```

[21]: (645.2281857687066, 209.9525460410454, 508.9583333333333, 35.818964088954296)

[22]:
```
df_new['Barrel_Bottom'].mean(),df_new['Barrel_Bottom'].
 ↪std(),df_old['Barrel_Bottom'].mean(),df_old['Barrel_Bottom'].std()
```

[22]: (613.0181090338428, 125.87513172433093, 856.7916666666666, 46.61123130810825)

[31]:
```
df_new['Barrel_Height'].mean(),df_new['Barrel_Height'].
 ↪std(),df_old['Barrel_Height'].mean(),df_old['Barrel_Height'].std()
```

[31]: (152.35663949166354, 197.38083968394773, 347.75, 17.40377270444044)

[34]:
```
df_new['Column_Height'].mean(),df_new['Column_Height'].
↪std(),df_old['Column_Height'].mean(),df_old['Column_Height'].std()
```

[34]: (1772.2760047070042, 234.6472462002318, 1887.8333333333333, 155.70531438437234)

[38]:
```
df_new['Barrel_Center'].mean(),df_new['Barrel_Center'].std(),\
(df_new['Barrel_Top'].mean()+df_new['Barrel_Bottom'].mean())/
↪2,(df_new['Barrel_Top'].std()+df_new['Barrel_Bottom'].std())/2,
```

[38]: (621.0285499617579, 157.62421794701226, 629.1231474012748, 167.91383888268817)

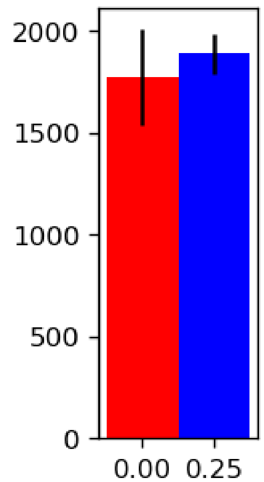
[42]:
```
df_old['Column_Orientation'].mean(),df_old['Column_Orientation_Std'].mean()
```

[42]: (8.375, 4.054166666666666)

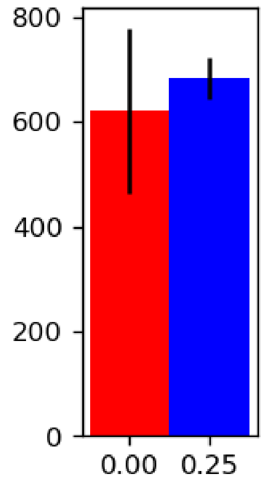
[48]:
```
df_new_orientations['Orientation'].mean(),df_new_orientations['Orientation'].
↪std()
```

[48]: (9.50977985178146, 8.469195617154067)

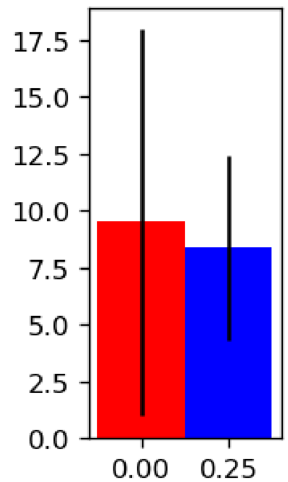
[62]:
```
N=1
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(4*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind,[df_new['Column_Height'].mean(),],\
         width, color='red', \
         yerr=[df_new['Column_Height'].std()],\
         )
rects2 = ax.bar(ind+width,[df_old['Column_Height'].mean(),],\
         width, color='blue', \
         yerr=[df_old['Column_Height_Std'].mean()],\
         )

plt.tight_layout()
fig.savefig('D:/vM1/Fig_2/column_ht_comparison.eps')
```



[64]:
```
N=1
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(4*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind,[df_new['Barrel_Center'].mean(),],\
         width, color='red', \
         yerr=[df_new['Barrel_Center'].std()],\
         )
rects2 = ax.bar(ind+width,[(df_old['Barrel_Top'].mean()+df_old['Barrel_Bottom'].
 ↪mean())/2,],\
         width, color='blue', \
         yerr=[(df_old['Barrel_Top'].std()+df_old['Barrel_Bottom'].std())/2],\
         )

plt.tight_layout()
fig.savefig('D:/vM1/Fig_2/barrel_center_ht_comparison.eps')
```



[65]:
```
N=1
ind = np.arange(N)
width = 0.25

fig = plt.figure(figsize=(4*cm,7*cm),dpi=120)
ax = fig.add_subplot(111)

rects1 = ax.bar(ind,[df_new_orientations['Orientation'].mean(),],\
         width, color='red', \
         yerr=[df_new_orientations['Orientation'].std()],\
         )
rects2 = ax.bar(ind+width,[df_old['Column_Orientation'].mean(),],\
         width, color='blue', \
         yerr=[df_old['Column_Orientation_Std'].mean()],\
         )

plt.tight_layout()
fig.savefig('D:/vM1/Fig_2/barrel_orientation_comparison.eps')
```


17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.00
0.25

[ ]: