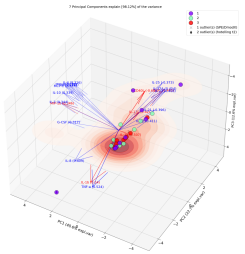
ELF_Cytokine_Analysis PCA, Apr 23, 2024 Python 3.18

```
#-------------------------------------------------------------------------
# ELF_Cytokine_Stats Analysis. PCA
# Apr 23, 2024 Python 3.18 by Teiji Sawa, MD, PhD
# Kyoto Prefectural University of Medicine, Japan
# For the analyses of the following publication:
# Authors: Sazuki Sudo, others & Teiji Sawa.
# Title: Case study observational research: inflammatory
# cytokines in the bronchial epithelial lining fluid of COVID-19
# patients with acute hypoxemic respiratory failure.
# Journal: Critical Care volume 28: 134, 2024.
# DOI: doi:10.1186/s13054-024-04921-3
#-------------------------------------------------------------------------

#-[Python Code #21]-----------------------------------------------------
from sklearn.preprocessing import StandardScaler
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
#pip install pca
#-[END]-----------------------------------------------------------------

#-[Python Code #22]-----------------------------------------------------
data_ELF_all = pd.read_csv('Data5_ELF_PCA_20240426_selected.csv')
data_plasma_all = pd.read_csv('Data6_Plasma_PCA_20240426_selected.csv')
#-[END]-----------------------------------------------------------------

#-[Python Code #23]-----------------------------------------------------
data_all_ELF = data_all[data_all.source=='elf']
data_all_plasma = data_all[data_all.source=='plasma']

data_ELF = data_all_ELF.iloc[:,3:23]
data_plasma = data_all_plasma.iloc[:,3:16]
data_LIV = data_all.iloc[0:27,2]
#-[END]-----------------------------------------------------------------

#-[Python Code #24]-----------------------------------------------------
ELF = data_ELF.to_numpy()
plasma= data_plasma.to_numpy()
LIV_group = data_LIV.to_numpy()
#-[END]-----------------------------------------------------------------

#-[Python Code #25]-----------------------------------------------------
scaler1 = StandardScaler(with_mean=True, with_std=True)
scaler2 = StandardScaler(with_mean=True, with_std=True)
ELF_std = scaler1.fit_transform(ELF)
plasma_std = scaler2.fit_transform(plasma)
#-[END]-----------------------------------------------------------------

#-[Python Code #26]-----------------------------------------------------
Cytokines_ELF= ['IL-6','IFN-g','MCP-1','MIP-1b','IL-10','IL-2','IL-5','IL-12p70','IL-7',\
                'GM-CSF','IL-4','G-CSF','IL-8','IL-1b','TNF-a','IL-31','IL-17A', \
                'IL-23','IL-25','sCD40L','IL-21','IL-22','IL-17F','IL-33','IL-13']

cytokines_plasma= ['IL-6','MCP-1','MIP-1b','IL-10',\
                'IL-5','IL-12p70','GM-CSF','G-CSF','IL-8','IL-1b','TNF-a', \
                'IL-33','IL-13' ]
#-[END]-----------------------------------------------------------------

#-[Python Code #27]-----------------------------------------------------
from pca import pca
# Initialize
model = pca(normalize = True)
#-[END]-----------------------------------------------------------------

#-[Python Code #28]-----------------------------------------------------
from pca import pca
# Initialize
model2 = pca(normalize = True)
#-[END]-----------------------------------------------------------------

#-[Python Code #29]-----------------------------------------------------
# Fit transform and include the column labels and row labels
results_ELF = model.fit_transform(ELF_std, col_labels=cytokines_ELF, \
                  row_labels=LIV_group)
#-[END]-----------------------------------------------------------------
```

```
#-[Python Code #30]-----------------------------------------------
# Fit transform and include the column labels and row labels
results_plasma = model2.fit_transform(plasma_std, col_labels=cytokines_plasma, \
                    row_labels=LIV_group)
#-[END]------------------------------------------------------------------
```

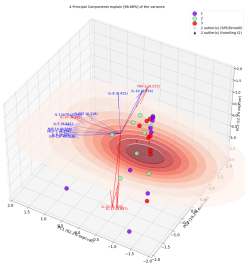
```
#-[Python Code #31]-----------------------------------------------
fig, ax = model.plot()
fig, ax = model.biplot3d(SPE=True,
                                HT2=True, arrowdict={'scale_factor': 3, \
                                'color_strong': 'red', 'color_weak': 'blue'}, s=500,
                                cmap="rainbow", density= True)
ax.set_xlim(-5, 5)
ax.set_ylim(-5, 5)
ax.set_zlim(-5, 5)
ax.legend(loc=1)
plt.savefig('Fig1_3D_ELF.svg')
plt.savefig('Fig1_3D_ELF.png')
plt.show()
#-[END]------------------------------------------------------------------
```

[out]



```
#-[Python Code #32]-----------------------------------------------
fig, ax = model2.biplot3d(SPE=True,
                                HT2=True, arrowdict={'scale_factor': 3, \
                                'color_strong': 'red', 'color_weak': 'blue'}, s=500,
                                cmap="rainbow", density= True)
ax.set_xlim(-5, 5)
ax.set_ylim(-5, 5)
ax.set_zlim(-5, 5)
ax.legend(loc=1)
plt.savefig('Fig2_3D_plasma.svg')
plt.savefig('Fig2_3D_plasma.png')
plt.show()
#-[END]------------------------------------------------------------------
```

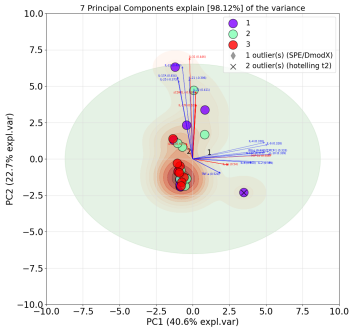
[Out]



```
#-[Python Code #33]-----------------------------------------------
# All available markers
markers = np.array(['o:blue', 'o:orange', 'o:red'])
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=ELF_std.shape[0])
# Draw markers
marker = markers
# Set alpha
```

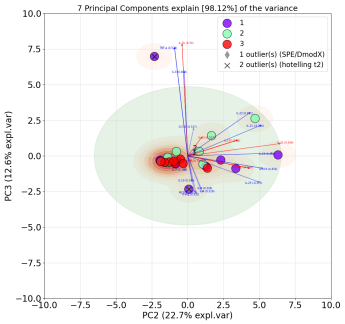
```
alpha = np.random.rand(1, ELF_std.shape[0])[0][random_integers]

fig, ax = model.biplot(PC= [0, 1],
                       SPE= True,
                       HT2= True,
                       s=400,
                       cmap="rainbow",
                       marker=marker,
                       n_feat=27,
                       fontsize=16,
                       fontweight='normal',
                       arrowdict={'fontsize': 6, 'scale_factor': 3, \
                                  'color_strong': 'r', 'color_weak': 'b'},
                       density= True,
                       density_on_top=False,
                       figsize= (12,12)
                       )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig3_ELF_DIM1&2.svg')
plt.savefig('Fig3_ELF_DIM1&2.png')
plt.show()
#-[END]-----------------------------------------------------------------------
```

[Out]



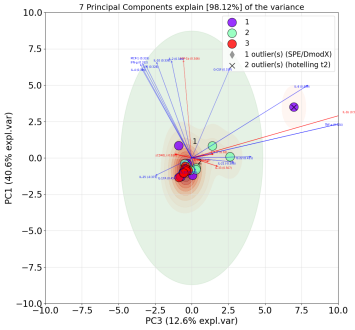

```
#-[Python Code #34]-----------------------------------------------
# All available markers
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=ELF_std.shape[0])
# Draw markers
marker = markers
# Set alpha
alpha = np.random.rand(1, ELF_std.shape[0])[0][random_integers]

fig, ax = model.biplot(PC= [1, 2],
                       SPE= True,
                       HT2= True,
                       s=400,
                       cmap="rainbow",
                       marker=marker,
                       n_feat=27,
                       fontsize=16,
                       fontweight='normal',
                       arrowdict={'fontsize': 6, 'scale_factor': 3, \
                                  'color_strong': 'r', 'color_weak': 'b'},
                       density= True,
                       density_on_top=False,
                       figsize= (12,12)
                       )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig4_ELF_DIM2&3.svg')
plt.savefig('Fig4_ELF_DIM2&3.png')
plt.show()
#-[END]-----------------------------------------------------------------------
```

[Out]




```
#-[Python Code #35]----------------------------------------------
# All available markers
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=ELF_std.shape[0])
# Draw markers
marker = markers
# Set alpha
alpha = np.random.rand(1, ELF_std.shape[0])[0][random_integers]

fig, ax = model.biplot(PC= [2, 0],
                       SPE= True,
                       HT2= True,
                       s=400,
                       cmap="rainbow",
                       marker=marker,
                       n_feat=27,
                       fontsize=16,
                       fontweight='normal',
                       arrowdict={'fontsize': 6, 'scale_factor': 3, \
                                  'color_strong': 'r', 'color_weak': 'b'},
                       density= True,
                       density_on_top=False,
                       figsize= (12,12)
                       )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig5_ELF_DIM3&1.svg')
plt.savefig('Fig5_ELF_DIM3&1.png')
plt.show()
#-[END]-----------------------------------------------------------------
```

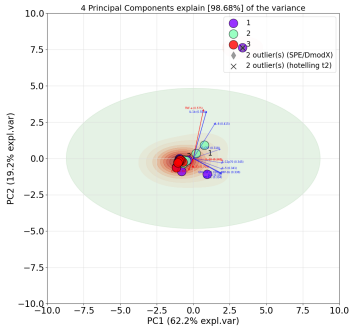
[Out]




```
#-[Python Code #36]----------------------------------------------
# All available markers
markers = np.array(['o:blue', 'o:orange', 'o:red'])
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=plasma_std.shape[0])
# Draw markers
marker = markers
# Set alpha
```

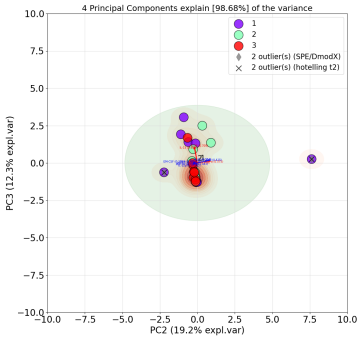
```
alpha = np.random.rand(1, plasma_std.shape[0])[0][random_integers]

fig, ax = model2.biplot(PC= [0, 1],
                        SPE= True,
                        HT2= True,
                        s=400,
                        cmap="rainbow",
                        marker=marker,
                        n_feat=27,
                        fontsize=16,
                        fontweight='normal',
                        arrowdict={'fontsize': 6, 'scale_factor': 1, \
                                   'color_strong': 'r', 'color_weak': 'b'},
                        density= True,
                        density_on_top=False,
                        figsize= (12,12)
                        )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig6_plasma_DIM1&2.svg')
plt.savefig('Fig6_plasma_DIM1&2.png')
plt.show()
#-[END]-----------------------------------------------------------------
```

[Out]




```
#-[Python Code #37]-----------------------------------------------
# All available markers
markers = np.array(['o:blue', 'o:orange', 'o:red'])
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=plasma_std.shape[0])
# Draw markers
marker = markers
# Set alpha
alpha = np.random.rand(1, plasma_std.shape[0])[0][random_integers]

fig, ax = model2.biplot(PC= [1, 2],
                        SPE= True,
                        HT2= True,
                        s=400,
                        cmap="rainbow",
                        marker=marker,
                        n_feat=27,
                        fontsize=16,
                        fontweight='normal',
                        arrowdict={'fontsize': 6, 'scale_factor': 1, \
                                   'color_strong': 'r', 'color_weak': 'b'},
                        density= True,
                        density_on_top=False,
                        figsize= (12,12)
                        )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig7_plasma_DIM2&3.svg')
plt.savefig('Fig7_plasma_DIM2&3.png')
plt.show()
#-[END]-----------------------------------------------------------------
```

[Out]




```
#-[Python Code #38]-------------------------------------------------
# All available markers
markers = np.array(['o:blue', 'o:orange', 'o:red'])
# Generate random integers
random_integers = np.random.randint(0, len(markers), size=plasma_std.shape[0])
# Draw markers
marker = markers
# Set alpha
alpha = np.random.rand(1, plasma_std.shape[0])[0][random_integers]

fig, ax = model2.biplot(PC= [2, 0],
                        SPE= True,
                        HT2= True,
                        s=400,
                        cmap="rainbow",
                        marker=marker,
                        n_feat=27,
                        fontsize=16,
                        fontweight='normal',
                        arrowdict={'fontsize': 6, 'scale_factor': 1, \
                                   'color_strong': 'r', 'color_weak': 'b'},
                        density= True,
                        density_on_top=False,
                        figsize= (12,12)
                        )
ax.set_xlim(-8, 8)
ax.set_ylim(-8, 8)

ax.legend(loc=1)
plt.savefig('Fig8_plasma_DIM3&1.svg')
plt.savefig('Fig8_plasma_DIM3&1.png')
plt.show()
#-[END]--------------------------------------------------------------
```

[Out]