# **ELF\_Cytokine\_Analysis PCA, Apr 23, 2024. R (v.4.3.2)**

#--------------------------------------------------------------------------

#ELF\_Cytokine Principal Component Analysis

# Apr 23, 2024 R (v.4.3.2) PCA 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

#--------------------------------------------------------------------------

#-**[R Code #1]**-------------------------------------------------

install.packages("FactoMineR")

install.packages("factoextra")

library(ggplot2)

library(GGally)

library(FactoMineR)

library(factoextra)

library(tidyverse)

#-[END]-----------------------------------------------------------

#-**[R Code #2]**-------------------------------------------------

data1 <- read\_csv("pMOL\_ELF4\_r25\_p27\_20240301\_PCA2.csv", show\_col\_types = FALSE)

data2 <-read\_csv("pMOL\_Plasma4\_r25\_p27\_20240301\_PCA2.csv", show\_col\_types = FALSE)

#-[END]-----------------------------------------------------------

#-**[R Code #3]**-------------------------------------------------

pca\_d1 <- PCA(data1[,9:29], graph =FALSE)

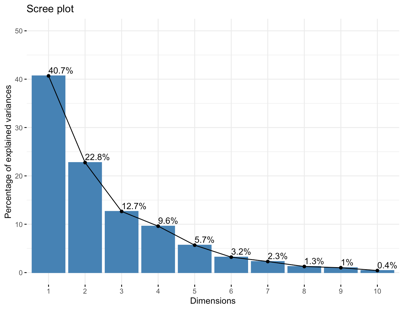
pca\_d2 <- PCA(data2[,9:22], graph =FALSE)

#-[END]-----------------------------------------------------------

#-**[R Code #4]**-------------------------------------------------

fviz\_screeplot(pca\_d1, addlabels =TRUE, ylim =c(0, 50))

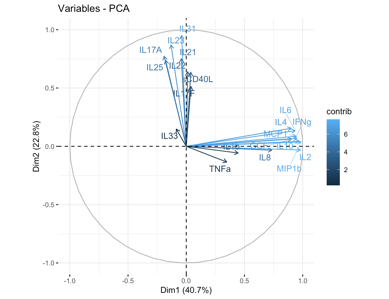
#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #5]**-------------------------------------------------

fviz\_pca\_var(pca\_d1,axes= c(1,2), col.var ="contrib", repel =TRUE)

#-[END]-----------------------------------------------------------

**[out]**

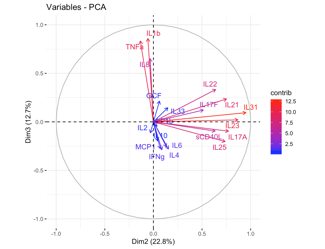
#-**[R Code #6]**-------------------------------------------------

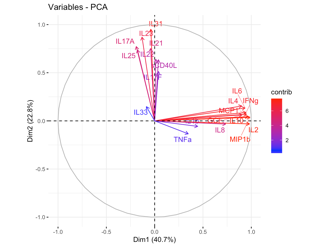
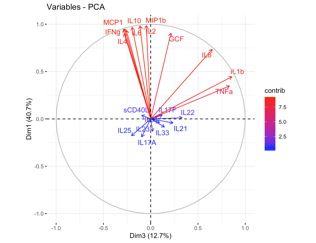
fviz\_pca\_var(pca\_d1, axes= c(1,2), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

fviz\_pca\_var(pca\_d1, axes= c(2,3), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

fviz\_pca\_var(pca\_d1, axes= c(3,1), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

#-[END]-----------------------------------------------------------

**[out]**



#-**[R Code #7]**-------------------------------------------------

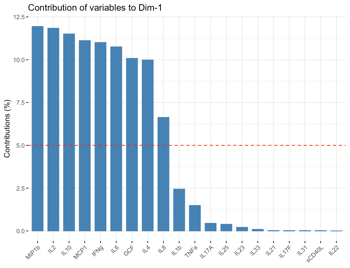
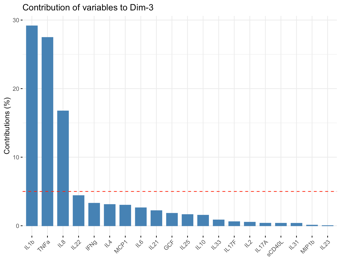
fviz\_contrib(pca\_d1, choice = "var", axes = 1)

fviz\_contrib(pca\_d1, choice = "var", axes = 2)

fviz\_contrib(pca\_d1, choice = "var", axes = 3)

#-[END]-----------------------------------------------------------

**[out]**



#-**[R Code #8]**-------------------------------------------------

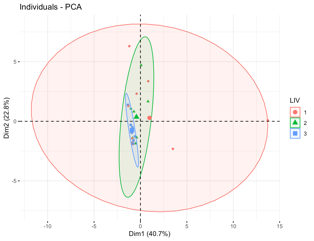
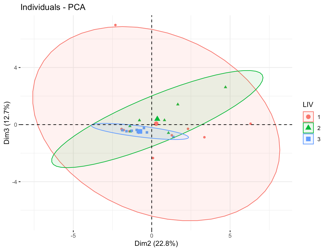
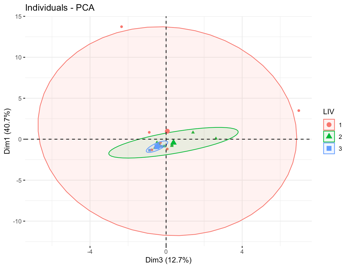
fviz\_pca\_ind(pca\_d1, geom.ind="point", axes= c(1,2), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

fviz\_pca\_ind(pca\_d1, geom.ind="point", axes= c(2,3), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

fviz\_pca\_ind(pca\_d1, geom.ind="point", axes= c(3,1), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

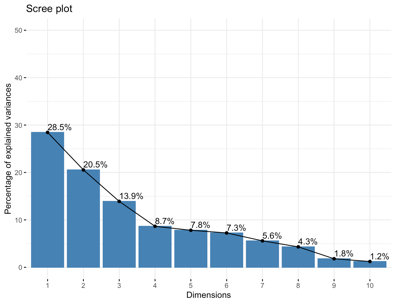
#-[END]-----------------------------------------------------------

**[out]**



#-**[R Code #9]**-------------------------------------------------

fviz\_screeplot(pca\_d2, addlabels =TRUE, ylim =c(0, 50))

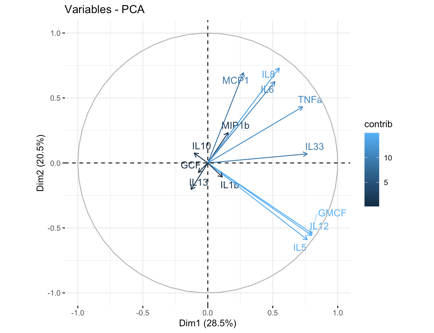
#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #10]**-------------------------------------------------

fviz\_pca\_var(pca\_d2,axes= c(1,2), col.var ="contrib", repel =TRUE)

#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #11]**-------------------------------------------------

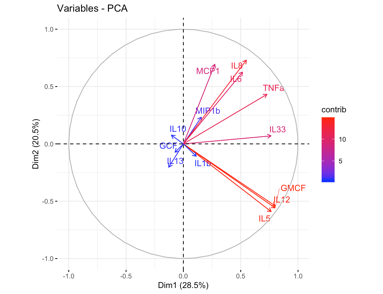
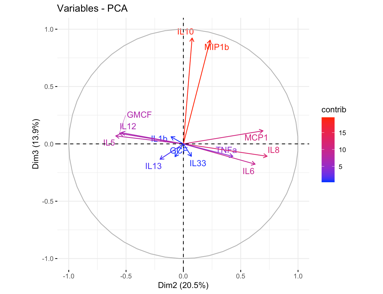
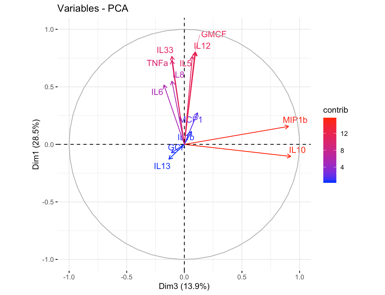
fviz\_pca\_var(pca\_d2, axes= c(1,2), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

fviz\_pca\_var(pca\_d2, axes= c(2,3), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

fviz\_pca\_var(pca\_d2, axes= c(3,1), col.var ="contrib", repel =TRUE) + scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=0.6)

#-[END]-----------------------------------------------------------

**[out]**



#-**[R Code #12]**-------------------------------------------------

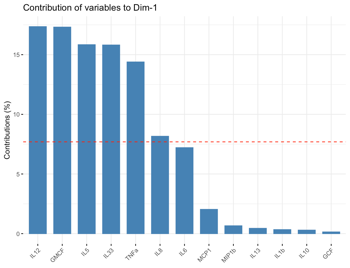
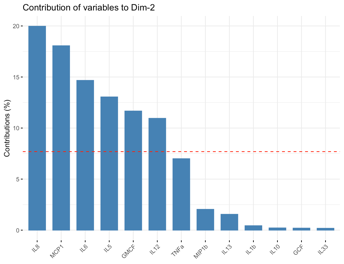
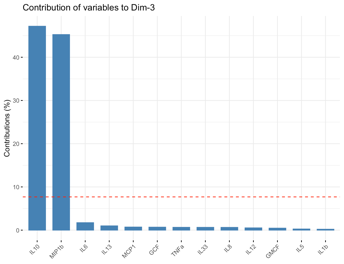
fviz\_contrib(pca\_d2, choice = "var", axes = 1)

fviz\_contrib(pca\_d2, choice = "var", axes = 2)

fviz\_contrib(pca\_d2, choice = "var", axes = 3)

#-[END]-----------------------------------------------------------

**[out]**



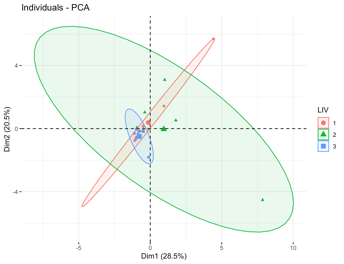
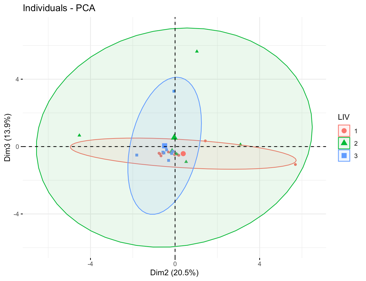
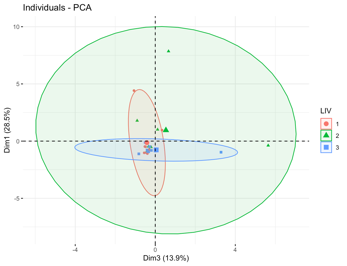
#-**[R Code #13]**-------------------------------------------------

fviz\_pca\_ind(pca\_d2, geom.ind="point", axes= c(1,2), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

fviz\_pca\_ind(pca\_d2, geom.ind="point", axes= c(2,3), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

fviz\_pca\_ind(pca\_d2, geom.ind="point", axes= c(3,1), habillage =as.factor(data1$liv\_group), legend.title = "LIV", repel=TRUE, addEllipses =TRUE)

#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #14]**-------------------------------------------------

library(corrplot)

corrplot 0.92 loaded

#-[END]-----------------------------------------------------------

#-**[R Code #15]**-------------------------------------------------

var1 <- get\_pca\_var(pca\_d1)

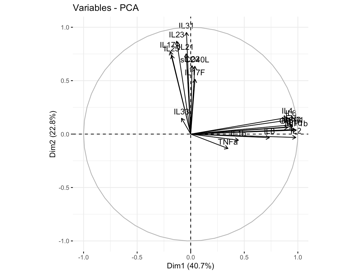
#-[END]-----------------------------------------------------------

#-**[R Code #16]**-------------------------------------------------

pca\_d1 %>% fviz\_pca\_var(col.var1 = "black")

#-[END]-----------------------------------------------------------

**[out]**



#-**[R Code #17]**-------------------------------------------------

var1$cos2 %>% round(4)

#-[END]-----------------------------------------------------------

**[out]**

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

IL6 0.8750 0.0179 0.0661 0.0051 0.0056

IFNg 0.8956 0.0075 0.0829 0.0030 0.0001

MCP1 0.9046 0.0049 0.0759 0.0017 0.0022

MIP1b 0.9718 0.0014 0.0022 0.0014 0.0160

IL10 0.9362 0.0019 0.0388 0.0025 0.0150

IL2 0.9634 0.0010 0.0129 0.0003 0.0138

IL4 0.8124 0.0240 0.0783 0.0001 0.0448

GCF 0.8201 0.0039 0.0458 0.0048 0.0001

IL8 0.5395 0.0011 0.4235 0.0185 0.0035

IL1b 0.1986 0.0033 0.7375 0.0483 0.0035

TNFa 0.1210 0.0181 0.6948 0.1233 0.0061

IL31 0.0016 0.9055 0.0089 0.0072 0.0380

IL17A 0.0362 0.5964 0.0092 0.1284 0.0575

IL23 0.0171 0.7562 0.0004 0.0307 0.1268

IL25 0.0315 0.5440 0.0412 0.2675 0.0037

sCD40L 0.0015 0.4033 0.0091 0.1935 0.1780

IL21 0.0018 0.5657 0.0556 0.1136 0.1781

IL22 0.0001 0.4101 0.1111 0.3238 0.0056

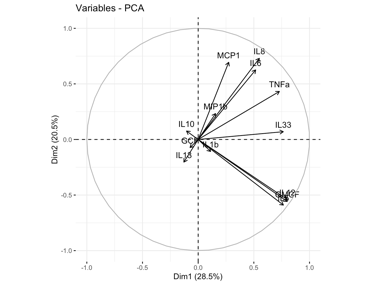
IL17F 0.0016 0.2636 0.0148 0.1539 0.4360

IL33 0.0073 0.0220 0.0213 0.4929 0.0001

#-**[R Code #18]**-------------------------------------------------

var1$cos2 %>% corrplot(is.corr = FALSE)

#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #19]**-------------------------------------------------

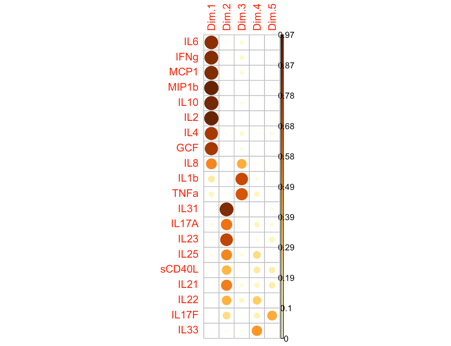
var2 <- get\_pca\_var(pca\_d2)

#-[END]-----------------------------------------------------------

#-**[R Code #20]**-------------------------------------------------

pca\_d2 %>% fviz\_pca\_var(col.var2 = "black")

#-[END]-----------------------------------------------------------

**[out]**

#-**[R Code #21]**-------------------------------------------------

var2$cos2 %>% round(4)

#-[END]-----------------------------------------------------------

**[out]**

Dim.1 Dim.2 Dim.3 Dim.4 Dim.5

IL6 0.2667 0.3918 0.0313 0.0090 0.0021

MCP1 0.0753 0.4825 0.0130 0.0241 0.1022

MIP1b 0.0244 0.0545 0.8181 0.0060 0.0012

IL10 0.0110 0.0058 0.8529 0.0001 0.0012

IL5 0.5859 0.3485 0.0048 0.0231 0.0173

IL12 0.6420 0.2926 0.0096 0.0080 0.0055

GMCF 0.6405 0.3116 0.0084 0.0013 0.0081

GCF 0.0053 0.0053 0.0125 0.3630 0.0791

IL8 0.3020 0.5335 0.0116 0.0112 0.0341

IL1b 0.0126 0.0116 0.0039 0.4438 0.1748

TNFa 0.5324 0.1869 0.0119 0.0027 0.0258

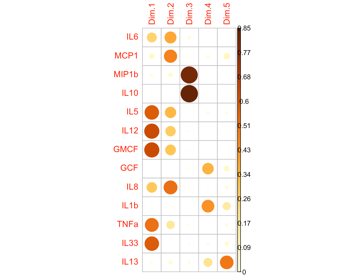
IL33 0.5850 0.0049 0.0118 0.0211 0.0501

IL13 0.0165 0.0415 0.0180 0.2116 0.5142

#-**[R Code #22]**-------------------------------------------------

var2$cos2 %>% corrplot(is.corr = FALSE)

#-[END]-----------------------------------------------------------

**[out]**

**ELF\_Cytokine\_統計解析 PCA, Apr 23, 2024 Python 3.18**

#-**[Python Code #1]**-------------------------------------------------

**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 #2]**-------------------------------------------------

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 #3]**-------------------------------------------------

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 #4]**-------------------------------------------------

ELF **=** data\_ELF.to\_numpy()

plasma**=** data\_plasma.to\_numpy()

LIV\_group **=** data\_LIV.to\_numpy()

#-[END]-----------------------------------------------------------------

#-**[Python Code #5]**-------------------------------------------------

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 #6]**-------------------------------------------------

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 #7]**-------------------------------------------------

**from** pca **import** pca

# Initialize

model **=** pca(normalize **= True**)

#-[END]-----------------------------------------------------------------

#-**[Python Code #8]**-------------------------------------------------

from pca import pca

# Initialize

model2 **=** pca(normalize **= True**)

#-[END]-----------------------------------------------------------------

#-**[Python Code #9]**-------------------------------------------------

# 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 #10]**-------------------------------------------------

# 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 #11]**-------------------------------------------------

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)

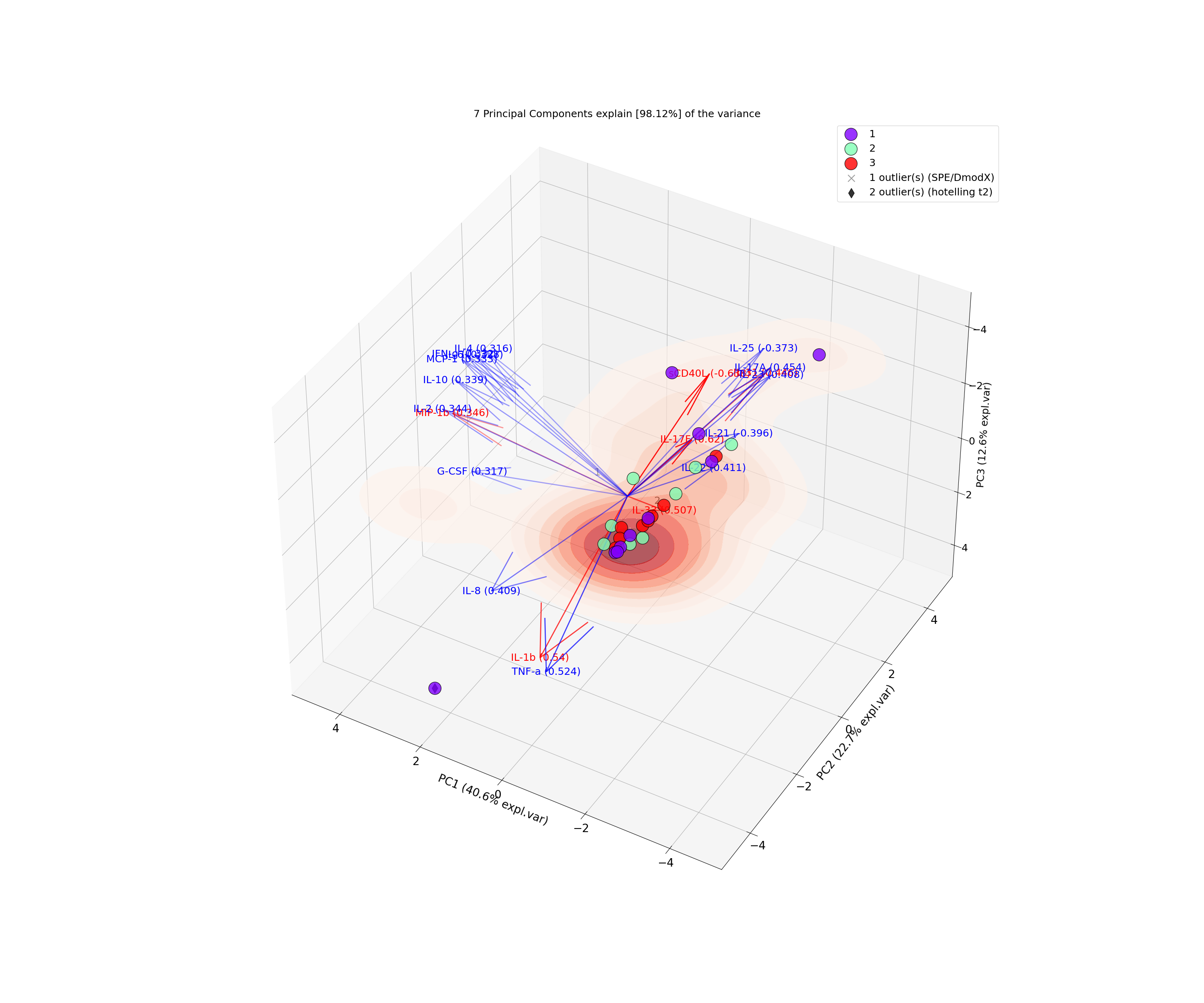
plt.savefig('Fig1\_3D\_ELF.svg')

plt.savefig('Fig1\_3D\_ELF.png')

plt.show()

#-[END]-----------------------------------------------------------------

**[out]**



#-**[Python Code #12]**-------------------------------------------------

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)

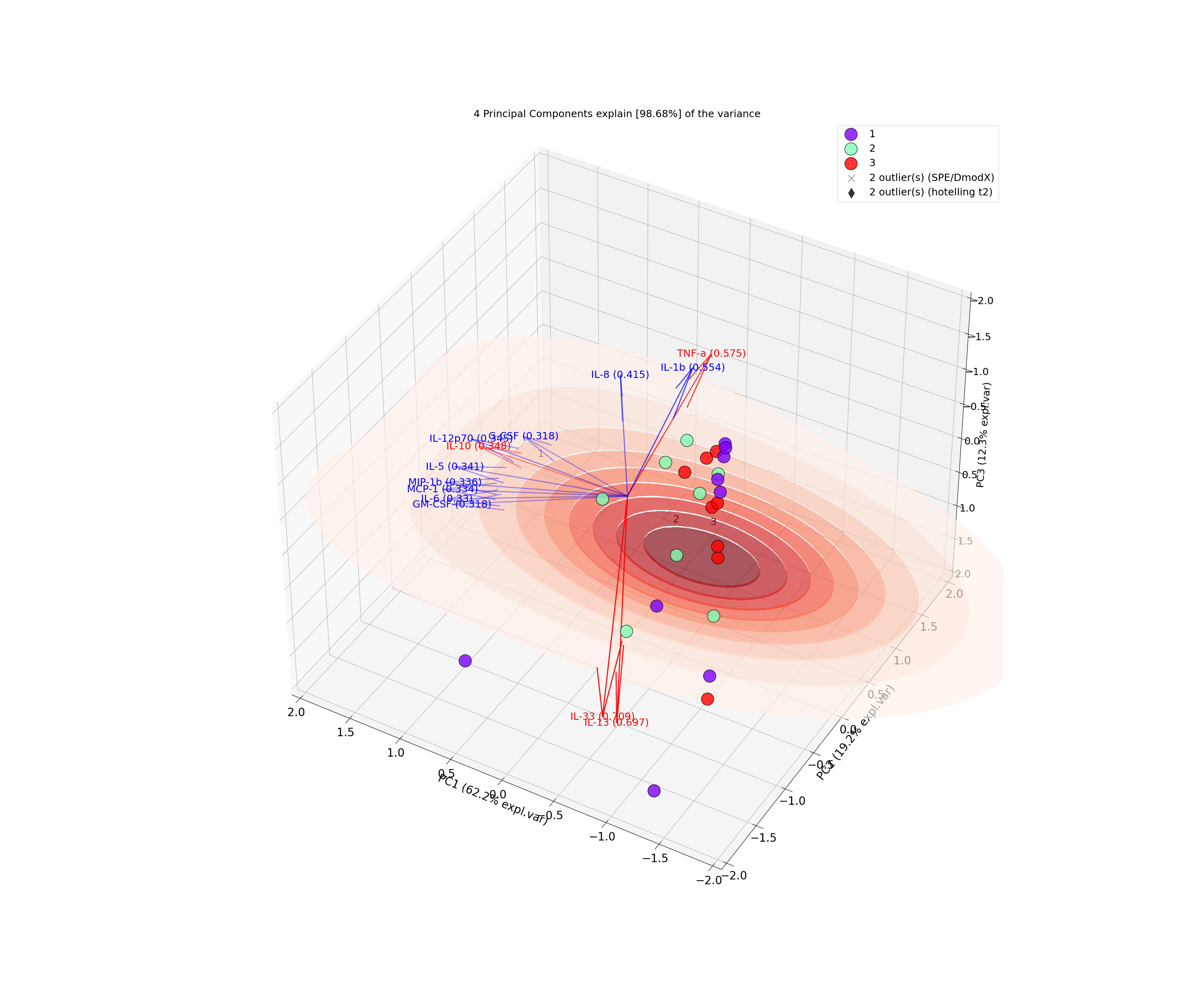
plt.savefig('Fig2\_3D\_plasma.svg')

plt.savefig('Fig2\_3D\_plasma.png')

plt.show()

#-[END]-----------------------------------------------------------------

**[Out]**



#-**[Python Code #13]**-------------------------------------------------

# 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)

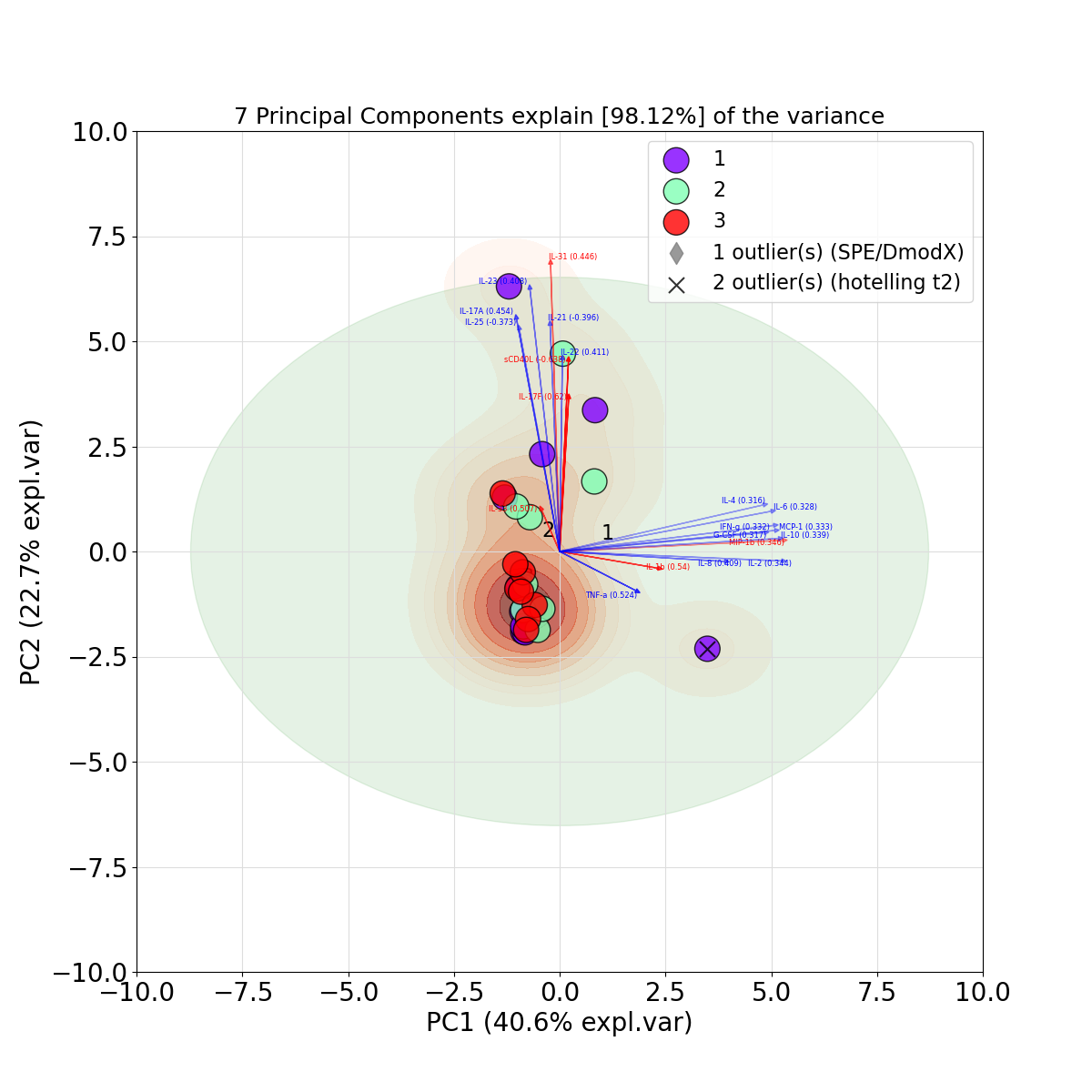
plt.savefig('Fig3\_ELF\_DIM1&2.svg')

plt.savefig('Fig3\_ELF\_DIM1&2.png')

plt.show()

#-[END]-----------------------------------------------------------------

**[Out]**



#-**[Python Code #14]**-------------------------------------------------

# 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)

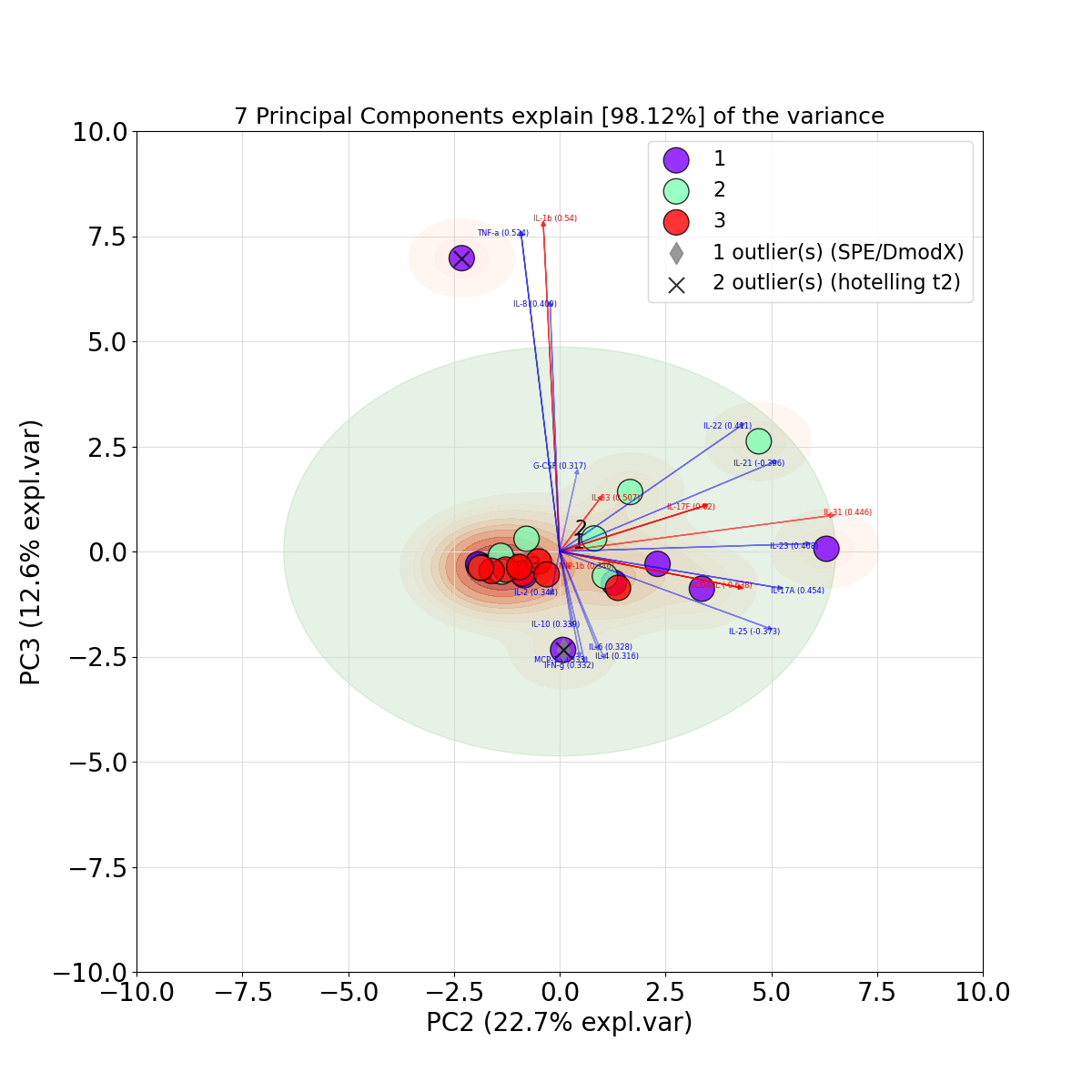
plt.savefig('Fig4\_ELF\_DIM2&3.svg')

plt.savefig('Fig4\_ELF\_DIM2&3.png')

plt.show()

#-[END]-----------------------------------------------------------------

**[Out]**



#-**[Python Code #15]**-------------------------------------------------

# 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)

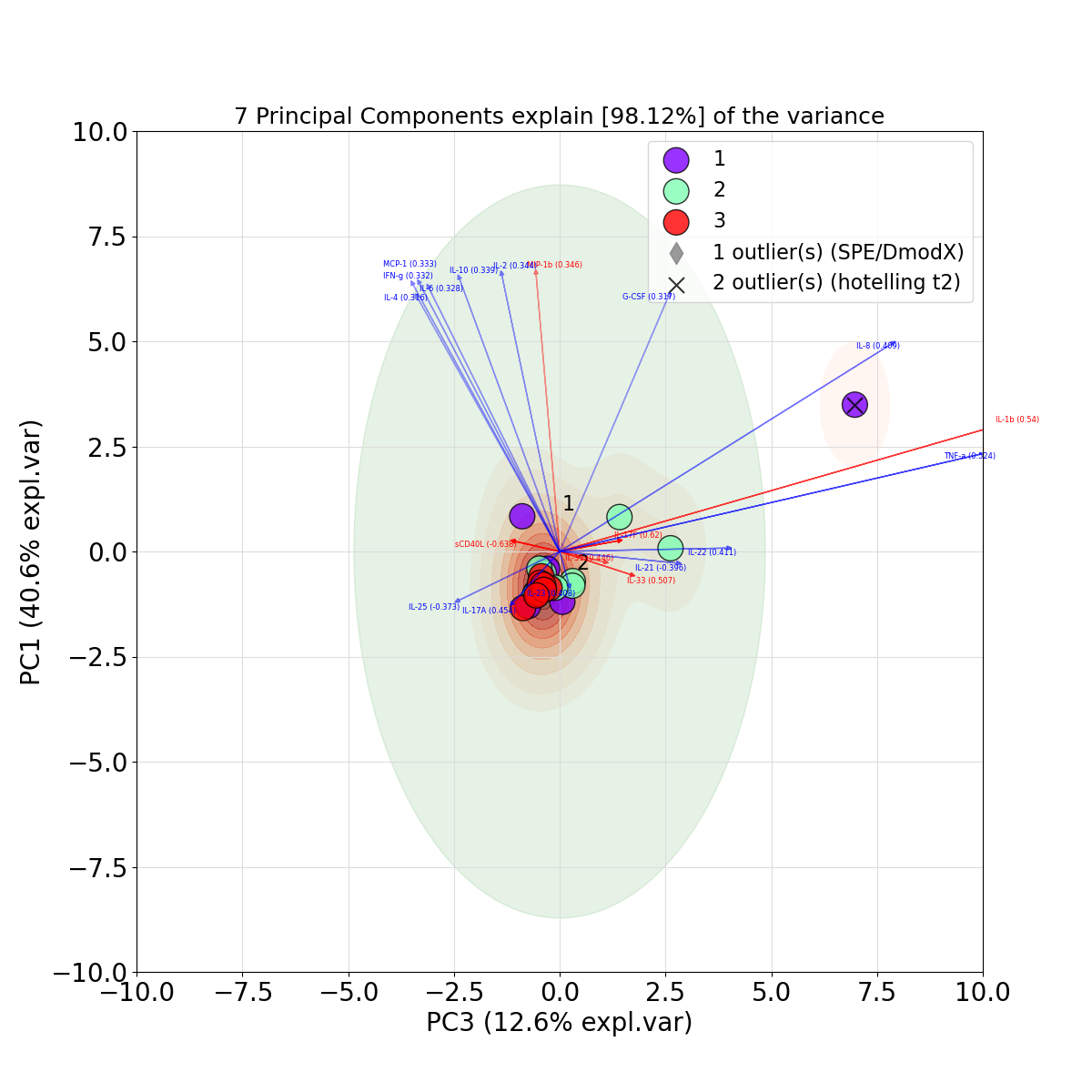
plt.savefig('Fig5\_ELF\_DIM3&1.svg')

plt.savefig('Fig5\_ELF\_DIM3&1.png')

plt.show()

#-[END]-----------------------------------------------------------------

**[Out]**



#-**[Python Code #16]**-------------------------------------------------

# 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)

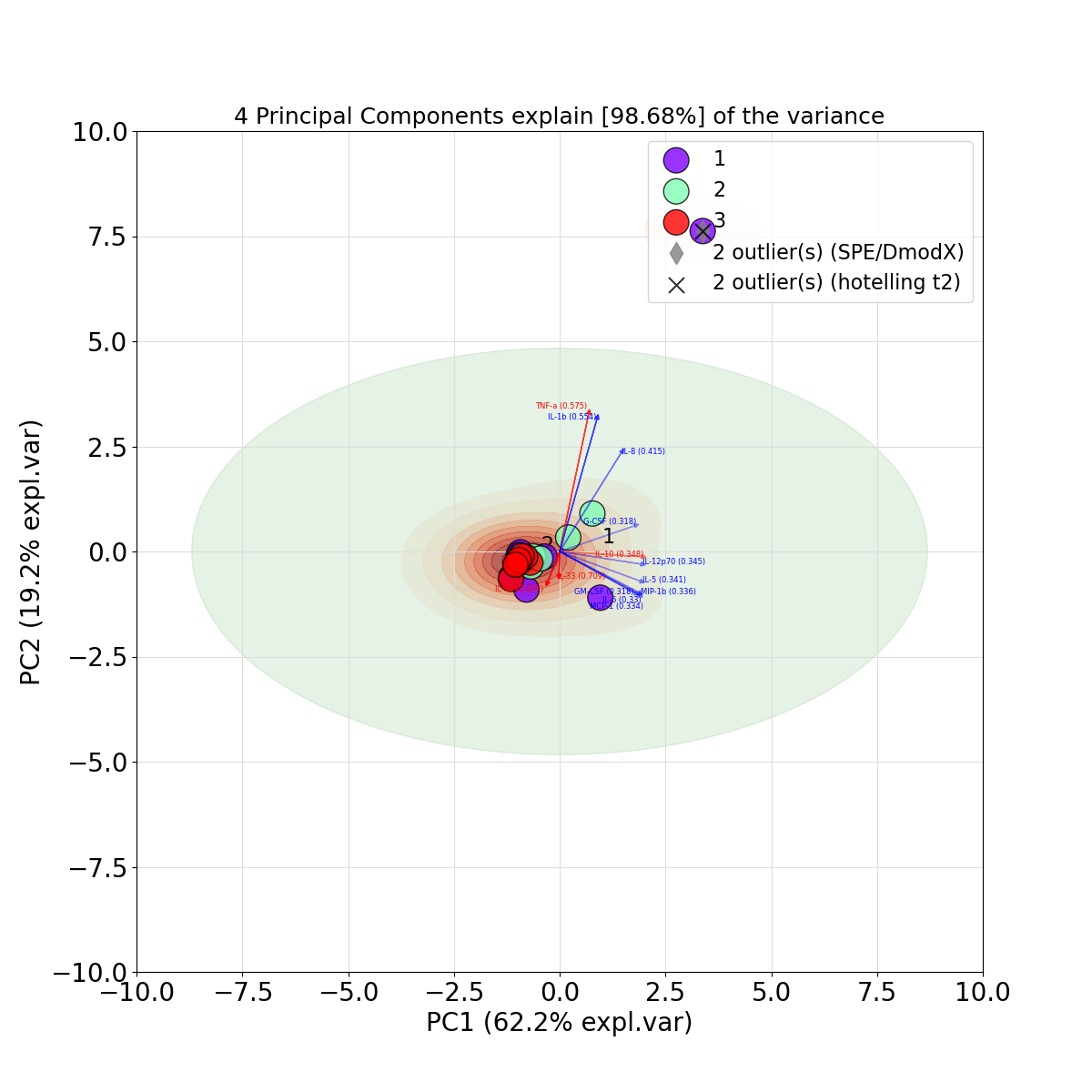
plt.savefig('Fig6\_plasma\_DIM1&2.svg')

plt.savefig('Fig6\_plasma\_DIM1&2.png')

plt.show()

#-**[END]**------------------------------------------------------------

**[Out]**



#-**[Python Code #17]**-------------------------------------------------

# 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)

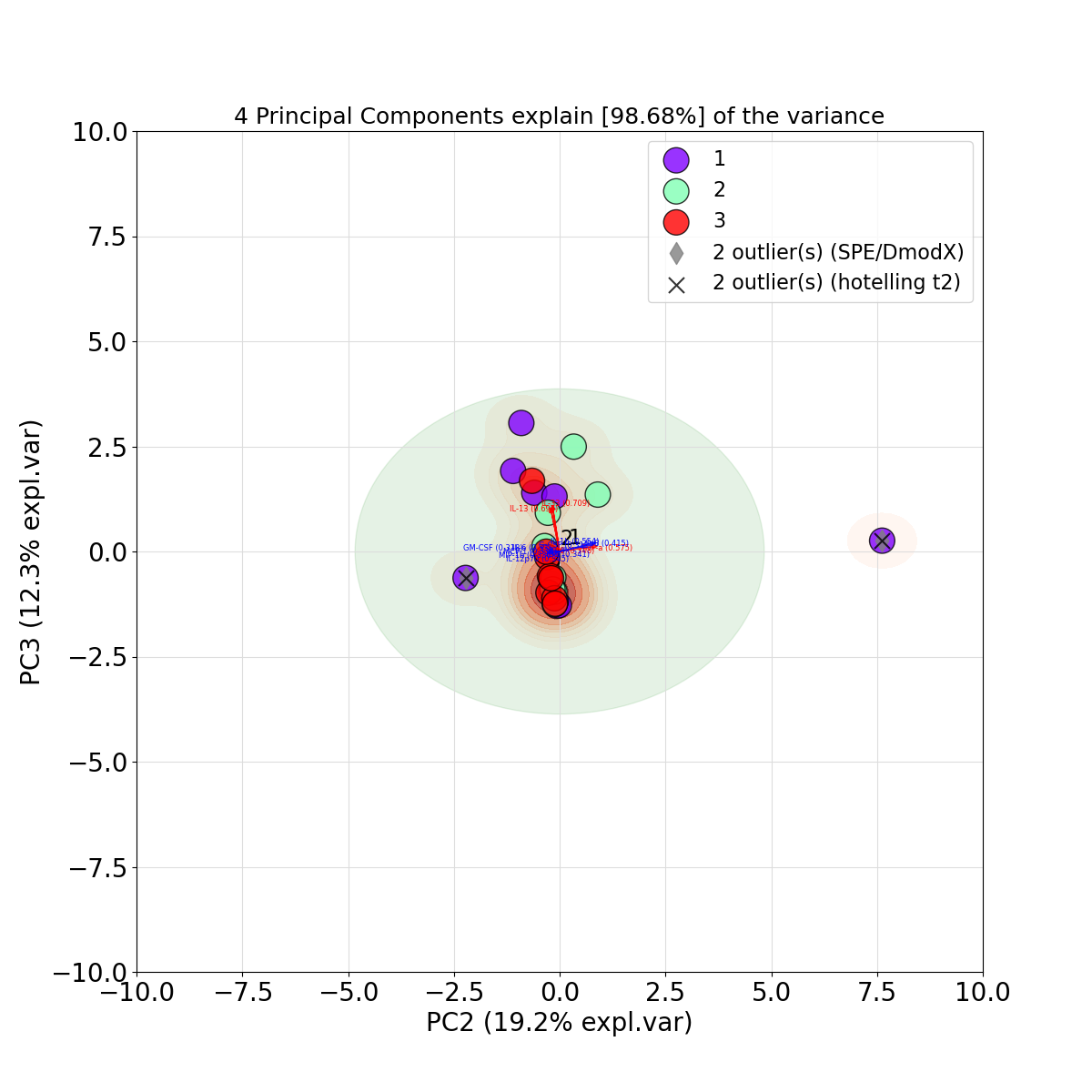
plt.savefig('Fig7\_plasma\_DIM2&3.svg')

plt.savefig('Fig7\_plasma\_DIM2&3.png')

plt.show()

#-**[END]**------------------------------------------------------------

**[Out]**



#-**[Python Code #18]**-------------------------------------------------

# 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]**

