Real dataset tutorial – PCA plot

Goal of this real dataset tutorial

In this tutorial using real dataset, we want to see if we can answer the following biological question: Can we see the pattern of clustering based on different proportions of barley grain fed in the cows?

Dataset description

"Metabolite concentrations of 39 rumen samples measured by proton NMR from dairy cows fed with different proportions of barley grain (Ametaj BN, et al.). Group label - 0, 15, 30, or 45 - indicating the percentage of grain in diet."

Step by step installation and running

Let's get started! First, we need to install the package.
Steps: 1. Git clone or download the github folder;
2. Open the terminal, and go to this folder;
3. Enter
`pip install dist/metabolomics_analysis_tools-0.1.0.tar.gz` to install the package locally;

Then, we can import functions we will use for this demo from the package metabolomics_analysis_tools@import_

```
import metabolomics_analysis_tools.data_preprocessing.data_reading as dr
import metabolomics_analysis_tools.data_preprocessing.normalization as dn
import metabolomics_analysis_tools.stats_analyses.analyses as sa
import metabolomics_analysis_tools.data_preprocessing.data_check as dc
import warnings
warnings.filterwarnings('ignore')
```

1. Then we can use the data_reading module to read in the data, by default it will read in the data from the resources/test_dataset folder in the package.
   We can also use the data_reading module to read in the data from a custom path, by passing the path as an argument to the read_data_file function

(file_path='path/to/file.csv').
The read_data_file function will return a pandas dataframe.

```
df=dr.read_data_file()
df.head()
```

```
data read successfully
the shape of the dataframe is:  (77, 65)
```

	Patient ID	Muscle loss	1,6-Anhydro-beta-D-glucose	1-Methylnicotinamide	2-Aminobutyrate
0	PIF_178	cachexic	40.85	65.37	18.73
1	PIF_087	cachexic	62.18	340.36	24.29
2	PIF_090	cachexic	270.43	64.72	12.18
3	NETL_005_V1	cachexic	154.47	52.98	172.43
4	PIF_115	cachexic	22.2	73.7	15.64

We can use the data_check module to check if the data is normally distributed.
The normal_dist_check function will return true if the data is normally distributed, false otherwise

```
dc.normal_dist_check(df)
```

```
True
```

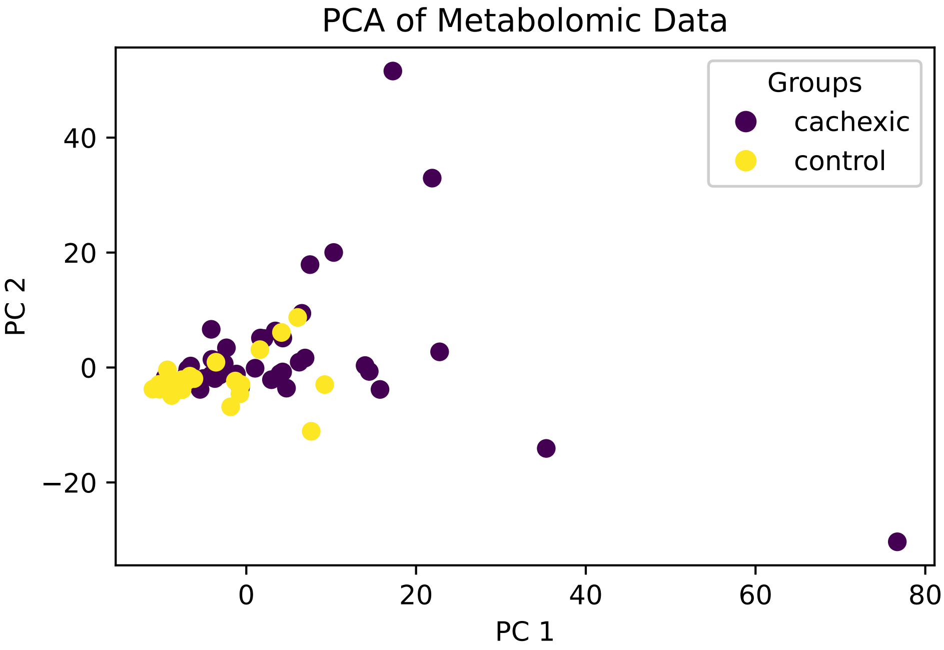
2. Next we can use the normalization module to normalize the data, here we will use the median normalization method normalized_data=dn.normalize_by_median(df).
   We can have a look at the first 5 rows of the normalized data normalized_data.head().

```
normalized_data=dn.normalize_by_median(df)
normalized_data.head()
```

	Patient ID	Muscle loss	1,6-Anhydro-beta-D-glucose	1-Methylnicotinamide	2-Aminobutyrate
0	PIF_178	cachexic	0.895833	1.786066	1.78551
1	PIF_087	cachexic	1.363596	9.299454	2.315539
2	PIF_090	cachexic	5.930482	1.768306	1.161106
3	NETL_005_V1	cachexic	3.3875	1.447541	16.43756
4	PIF_115	cachexic	0.486842	2.013661	1.490944

3. We can use the analyses module to perform statistical analyses on the data.
   Here we will first perform a PCA analysis on the data to see if there are any patterns in the data.
   The PCA_analysis function will return a pandas dataframe containing the principal components principal_components=sa.PCA_analysis(normalized_data).

```
principal_components=sa.PCA_analysis(normalized_data)
```



- Interpretation of the PCA results:
  From the PCA results, we can see it is hard to distinguish the groups of cows bsaed on their different feeding plan (indicated by colors). However, the first two PCs explained most of the variance in the data. Therefore, even though the clustering didn't give us a good pattern for different groups of data, I might assume that it was due to limited number of samples. If we have more samples, we might be able to see a better pattern.