FeatureCloud

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare

Deliverable D5.2
Implementation and Integration of Layman-friendly interfaces

Work Package WP5
Unsupervised Federated Machine Learning

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078. Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

Document information

Grant Agreement Number: 826078			**Acronym: FeatureCloud**	
Full title	Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare			
Topic	Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures			
Funding scheme	RIA - Research and Innovation action			
Start Date	1 January 2019	**Duration**	60 months	
Project URL	https://featurecloud.eu/			
EU Project Officer	Christos MARAMIS, Health and Digital Executive Agency (HaDEA) - Established by the European Commission, Unit HaDEA.A.3 – Health Research			
Project Coordinator	Jan BAUMBACH, UNIVERSITY OF HAMBURG (UHAM)			
Deliverable	D5.2 Implementation and Integration of Layman-friendly interfaces			
Work Package	WP5 Unsupervised Federated Machine Learning			
Date of Delivery	**Contractual**	30/06/2022 (M42)	**Actual**	30/06/2022 (M42)
Nature	Demonstrator	**Dissemination Level**	Public	
Lead Beneficiary	4 SDU			
Responsible Author(s)	Richard Röttger			
	Tobias Frisch, Anne Hartebrodt			
Keywords	Clustering; Principal Component Analysis; k-means; Expectation Maximization; Gaussian Mixture Model; Visualization			

Table of Content

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

1 Objectives of the deliverable based on the Description of Action (DoA)

The objective was to develop a FeatureCloud application able to visualize and present clustering results in an interactive manner.

Task 4: Visualization & presentation of the clustering results (SDU, UM, GND)
Layman-friendly interfaces for the tools will facilitate the adoption and acceptance of the published software. Here, SDU will develop feasible ways to represent the results of federated clustering which also will account for visualizing the distribution over the various participating FeatureCloud instances. Particularly, for visualizing the results of the multi-OMICS network-based clustering, sophisticated visualization methods will be implemented allowing for the inspection of the extracted networks as well as of the stratification at the same time. Also, feasible and interpretable cluster quality measures will be implemented including an automated improvement suggestion based on the guidelines developed in task 3. The acceptance of the system with potential future users will be evaluated with UM to best reflect the future clinical practice. The visualization is prototyped by SDU and fully developed by GND.

- **M31** Implementation of federated pre-processing methods
- **M32** Implementation of federated clustering methods
- **M33** Implementation of federated cluster evaluation methods & completion of the clustering pipeline
- **M35** Laymen-friendly visualization of clustering results and easy-to-use front-end for federated clustering

2 Executive Summary

Visualization of results for easy access is an essential part of unsupervised machine learning. Within the FeatureCloud project the main challenge is the distributed nature of the data that does not allow for visualization of raw sample information across contributing institutions. To unleash the full potential of unsupervised approaches, a comprehensive visualization is necessary to allow experts in machine learning to access the results in order to evaluate their quality and potentially correct for existing issues such as outliers or confounding factors. At the same time the user-interface needs to be easy to understand also for users not familiar with software development such as medical doctors. In this deliverable we therefore present our user-friendly cluster visualization app, developed together with our partner GND. We have performed an analysis of requirements and then carefully designed a FeatureCloud app that is flexible, easy to use and still provides the necessary insight into the data. In the following deliverable, we present the UI based on an artificial dataset to highlight its full potential and then apply it in the context of a full clustering workflow to show the interaction between our developed apps.

3 Introduction (Challenge)

In unsupervised machine learning we are aiming to learn patterns and extract information from unlabeled data. Therefore, the correct grouping of the data points is generally unknown and the quality of the results can often only be accessed by proper visualization of the clustering results. Within the FeatureCloud project, visualization comes with additional challenges introduced by the distributed nature of the samples. As to this point only very few apps developed within the FeatureCloud project have utilized any form of visualization, we had to decide whether we want to

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

include plotting functionality within the FeatureCloud controller or if our clustering visualization should be an independent application developed within the framework. We have decided on the latter as this allowed for much more flexibility especially with respect to interactivity. In an initial process we tried to derive the most important requirements with respect to a user-interface.

User-Friendly:

Although usability should always be a key aspect in bio-medical software applications we often experience that especially, people that are not familiar with software development such as medical doctors are struggling. Additionally, challenges arise within the FeatureCloud project, as our visualizing app is not only used by non-computer scientists but might also be of interest for other app developers. While the former requires an intuitive, self-explanatory UI, the latter might require extensive documentation allowing them to utilize the functionality of our visualization.

Visualization:

For the visualization we require a wide variety of plot types to be present:

1. Scatterplot
2. Density Plot
3. Heatmap
4. Line & bar plot
5. Pie-chart
6. Histogram
7. Scree Plot

The plot types we selected are widely used for the visualization of clustering results and should be intuitively understandable.

Interactive:

In order to maximize the usability of our visualizing app we require an interactive UI that can change plots based on user-input. This is connected to the “human in the loop” approach that allows users to interact with the training algorithm. In particular, the selection of subsets of clusters/samples based on confounding factors or visual appearance of data points in the graphs is an important requirement. When working with clustering we always have the risk of outliers or batch effects that might render results unusable. Additionally, we want to allow the user to re-run clustering approaches based on the results and on the selected sample subset. Thereby, existing clusters might be re-clustered into more distinct groups of patients. The general goal that is connected with this is that the workflow integrated in the FeatureCloud backend will be able to continue with a subset of data points or features allowing to re-run other apps.

Confounders:

Confounding factors such as age, sex or ethnicity can have a significant influence on the result of any unsupervised learning approach. Therefore, we want our user-interface to be able to visualize the distribution of confounders not only by cluster, but also by contributing side. Here, the user should be able to select certain confounding factors that are of interest and importance for a specific set of clusters.

Flexibility:

Last but not least, the app requires a certain flexibility that allows future clustering applications to utilize its functionality. Therefore, the input format for the app, should be flexible enough, to also

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

account for different types of parameters, and should be able to also deal with multiple runs necessary for parameters optimizations. It should allow the user to export plots as well as the specific plot settings to increase reproducibility in medical applications. Finally, flexibility includes that the app is able to deal with large datasets. This might not be a problem with respect to the amount of samples, but the amount of confounding factors can of course be quite significant.

Privacy:

A key aspect that is part of the FeatureCloud project is obviously the aspect of data privacy. This includes especially, that the raw data present at a client cannot be exchanged with other clients. Consequently, we have to find a way of exchanging the density of data points across clients in order to allow for the visualization of point distribution in a scatterplot.

4 Methodology

As stated previously, we have decided to develop a standalone FeatureCloud app that will be able to be used within a workflow. This approach ensures maximum flexibility and will make the app more independent from the clustering apps that we have implemented as it only relies on the correct input structure. The app is implemented in python, which is consistent with the language used in the FeatureCloud ecosystem. In Table 1, we have listed the set of packages our app is relying on. We have decided to utilize Dash and Plotly [https://dash.plotly.com/datatable/reference], since those are available as python packages and offer a wide range of functionality and flexibility.

4.1 Input Files

In Figure 2 we can see the file structure within the input folder that is required by the visualizing app. However, the exact location of the files can be altered using the configuration file.

4.1.1 Configuration

The configuration file is generally used in FeatureCloud applications to transmit state/runtime information to the respective app that either runs as standalone or within a workflow. The path to all the files that are specified within this section is required.

```
fc-cluster-visualization-app:
    data-dir: 'data/exampleData'
    local-data-path: 'data/exampleData/localData.csv'
    distance-matrix-path: 'data/exampleData/distanceMatrix.csv'
    confounding-meta-path: 'data/exampleData/confoundingData.meta'
    confounding-data-path: 'data/exampleData/confoundingData.csv'
    variance-explained-path: 'data/exampleData/varianceExplained.csv'
    k-values-clustering-result-dir: 'data/exampleData/results'
    k-values-clustering-file-name: 'clustering.csv'
    k-values-silhouette-file-name: 'silhouette.csv'
    # all files downloaded from the browser will end up here too
    download-dir: 'data/exampleData/downloads'
```

Figure 1: An example for the configuration file

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

4.1.2 Result folder

The results folder contains the results of the clustering, where each sub-folder represents a clustering with a distinct set of parameters. Multiple parameters should be separated by the character "_" and the folder name, hence the set of parameters needs to be unique. This means we do not allow different results for a clustering algorithm that are based on the same parameters.

```
data
└──results
|   └──K2
|       |   clustering.csv
|       |   silhouette.csv
|   └──K3
|       |   clustering.csv
|       |   silhouette.csv
|   └──...
|   └──K<n>
|       |   clustering.csv
|       |   silhouette.csv
|   confoundingData.csv
|   confoundingData.meta
|   localData.csv
|   distanceMatrix.csv
|   varianceExplained.csv
```

Figure 2: Overview of input files and the required structure

4.1.3 Confounding Data

The visualization app is able to differentiate between three different types of confounding data, namely continuous, discrete and ordinal types. The specific type needs to be specified in the meta file and the specific values should be present in the "confounding-data-path" file. The separation into those groups allow us to visualize the confounders in different ways.

4.1.4 Local Data

This file contains the sample information that should be visualized. This can either be raw data that is present only within the local client, or it could be the projections of a PCA. The latter one has the advantage that we'll be able to visualize all data points within a scatter plot that will then look very much like the original aggregated data set. The projections of the PCA, on the other hand, leak private information across the clients. The PCA implementation of our group however allows us to compute more private sample projections mitigating the privacy risk (Hartebrodt *et al.*, 2021) (Hartebrodt and Röttger, 2022).

4.1.5 Distance Matrix

For clustering algorithms that produce a distance matrix as output, we can accept a csv file containing a distance matrix and visualize this matrix in a heatmap. For the majority of algorithms this will probably be the distance matrix that is only based on the local samples.

4.1.6 Variance explained

When using the PCA we can here store the variance explained by the projections in order to produce a scree plot.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

4.2 Workflow

In order to come closer to a real-world application that comprises not only the visualization app but also the previously developed apps (Deliverable 5.1), we have built a workflow (Figure 3).

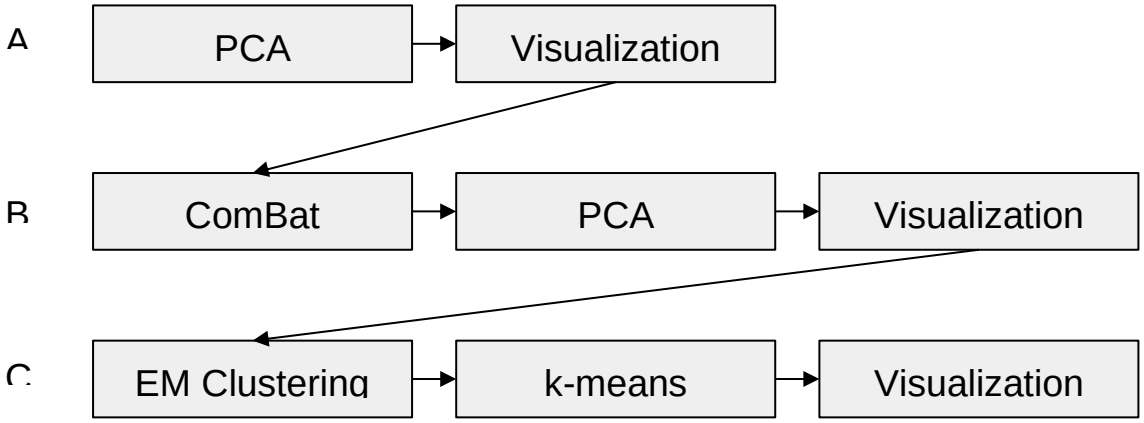

Figure 3: Visualization of the workflow chosen for the real-world data set with the three steps towards a reasonable clustering.

This workflow consists of three major steps:

A) Visualization of the overall dataset
B) Correction of batch effects followed by visualization as confirmation
C) The actual clustering pipeline and the comparison of the results.

For the workflow, the app's output was synchronized to match the corresponding input of the following app. The dataset we used consists of three GEO datasets (GSE129508 (Ligibel *et al.*, 2019), GSE149276 (Park *et al.*, 2020) , GSE58135 (Varley *et al.*, 2014). Overall, the combined dataset consists of 132 samples and 35238 genes. As each cohort has been "independently collected and sequenced at three different laboratories" (Zolotareva *et al.*, 2021), the data are suffering from a quite strong batch effect. Those effects are very common in biomedical data and will strongly affect the result of any unsupervised algorithm. Hence the pipeline will A) visualize the PCs of the raw data in order to confirm the batch effect, B) correct the batch effects and confirm this by visualization and finally C) cluster the samples based on our clustering approaches.

5 Results

To show the full potential of our app, and also its applicability, usefulness and potential for real world applications, we have divided the results into two subsections. In subsection 5.1, we show the full potential and all features of the app using an artificial dataset. Afterwards (subsection 5.2) we demonstrate its utility based on a real-world dataset.

5.1 Visualizing App

Here we used artificial example data in order to show the full potential of the visualization app.
The app developed in cooperation with GND can be split into four major components that are present in the frontend as tabs (see Figure 4).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

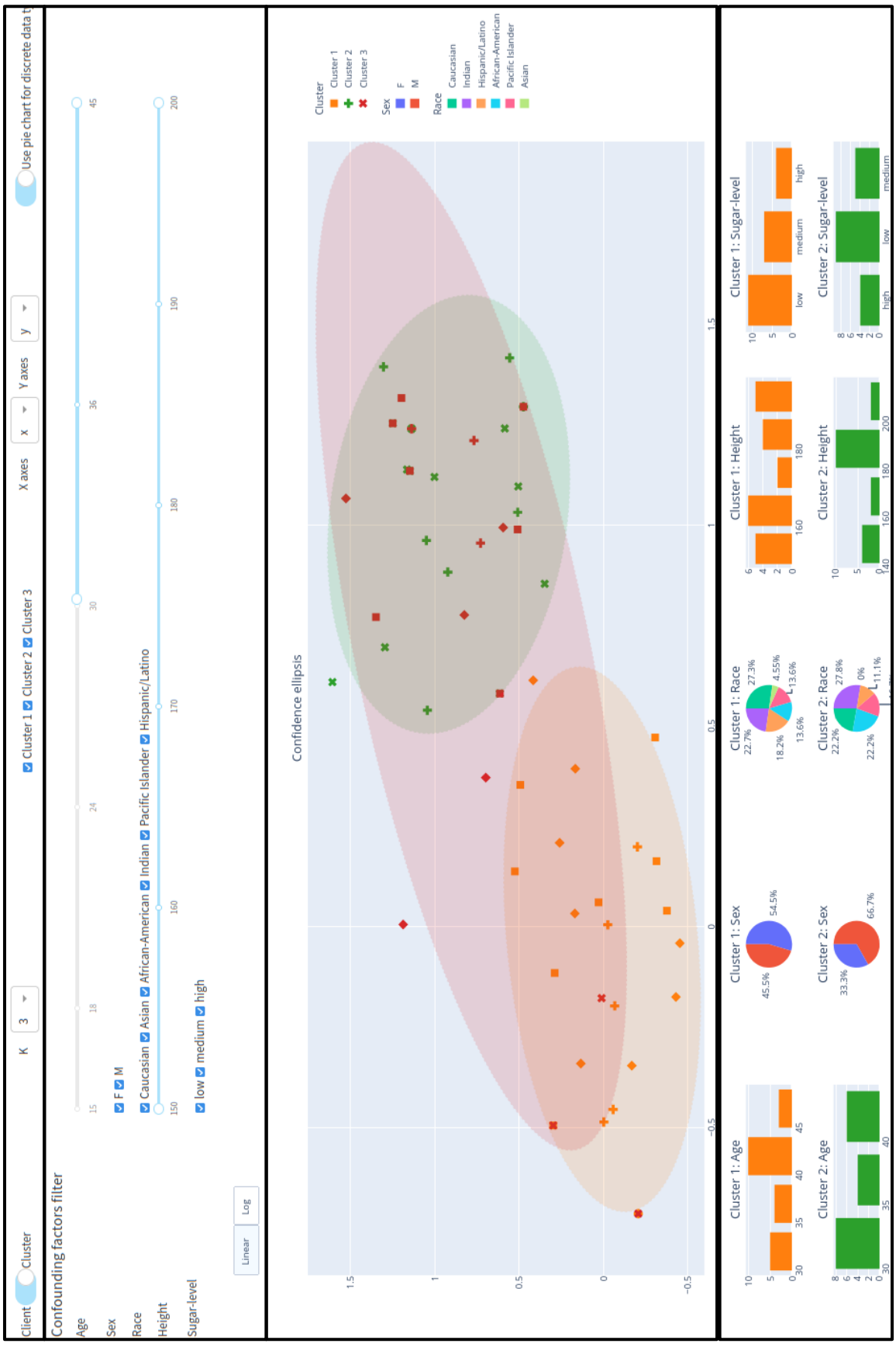

Figure 4: Initial screen of the cluster visualization app showing the different tabs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

The first and major screen that visualizes a significant amount of information is called “Confounders”. It can be divided into four areas (A-D; Figure 4), that are either dedicated to parameter selection (A, B) or visualization (C, D). The high-level settings are present in area A. Here, in the upper left corner we can see the slider that allows us to switch between the cluster results and the local data. When set to “client”, the visualization app treats the data present at each client as a cluster and shows the distribution of confounders that are present at each local side. This gives a great overview on the actual distribution of samples across contributors and allows to identify sets that should be excluded due to their skewed raw data. The next drop-down menu allows for the selection of the parameter set for the specific clustering algorithm. The parameters available are extracted out of the folder names, as described in subsection 4.1.2. The IU also allows here to de-select the clusters that should be included within the plots. This feature significantly increases the usability in case of many but small clusters. The next drop-down menus connect the axis of the scatterplot in area C with the corresponding columns in the “localData” file (subsection 4.1.4). The visualization app itself is highly flexible and does not specify the amount of

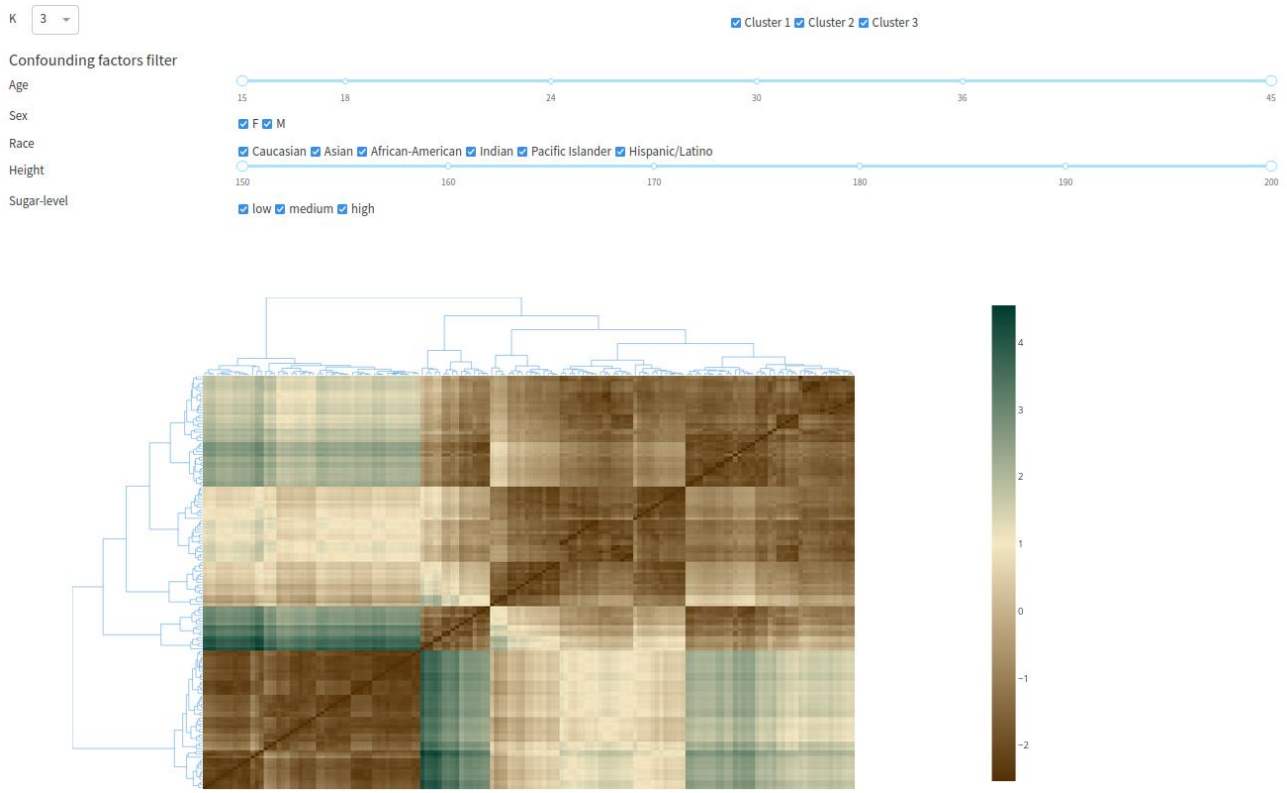

Figure 5: The heatmap shows pairwise distances. The samples can be filtered by confounding factors and cluster ID.

rows and columns present in the local data file. Of course, this file might be limited to the data available at the local site. In this case, the amount of columns within “localData” represent the amount of dimensions and the rows represent the number of samples. In our second use-case however, the localData might contain the projections or pseudo-projections of a PCA performed across all participants, in which case the columns represent the principal components ordered by their impact on the variance. The user then has the chance to compare all possible combinations of PCs to see if the points are separated. The final option in the upper right corner allows the user to switch the plot type for discrete confounding factors in area D. This option has proven to be useful since different plot types are easier to read depending on the amount of confounders and the amount of levels for a confounding factor. Below the general cluster settings, we have placed a dynamic area that allows for selection of confounders (area B). For each confounder, we are allowed to select whether it

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

should be visualized or not, either with tick boxes (discrete type) or with sliders (continuous type). The plots in areas C and D are then re-drawn for user selection.

The most prominent, and probably most helpful visualization is the scatter plot in area C. It is also an interactive plot that generally allows to show additional information, when hovering over data points with the mouse and the values can be log transformed if necessary. Additionally, clusters and confounding factors can be selected to customize the plot and the user is able to export a png file. One of the most important features here is also the selection of either individual- or groups of data points with the mouse. Selected data points can either be of interest for the next step or could be outliers that should be excluded from the downstream pipeline. The list of selected (local) data points can also be exported as a list. Below the scatter plot, in area D, we visualize the distribution of the confounding factors. As a compromise, only five different confounding factors can be shown at the same time. However, confounders are of high interest, as it might be that the clusters are not revealing any new information, but rather differentiate groups of samples that are mostly defined by them.

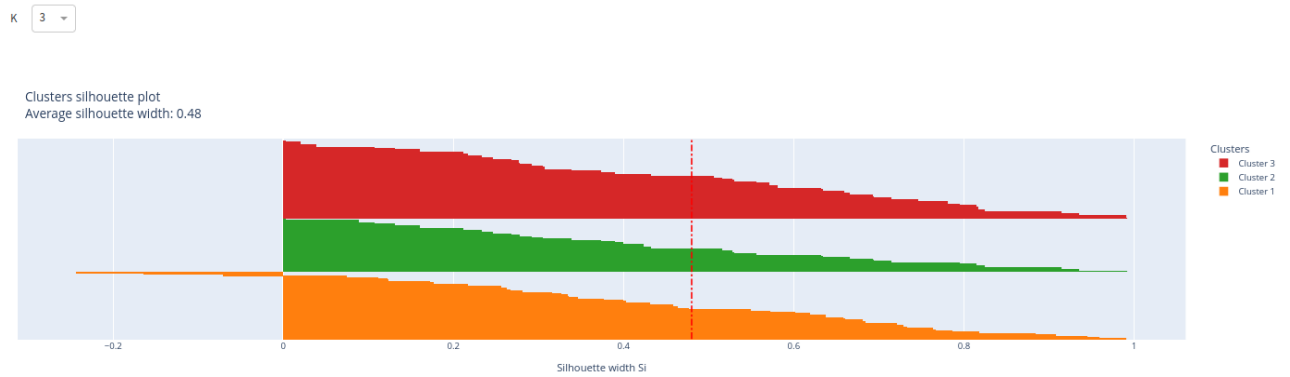

Figure 6: Visualization of clustering quality based on the simplified silhouette score.

In Figure 5, we show the second tab called “Distances”, where distance or similarity measures can be visualized. Computation of a pairwise distance between federated samples is at this point an unsolved problem, but the pairwise distance between local samples is of course an option. The visualizer automatically applies a hierarchical clustering to the heatmap, and draws the resulting dendrogram. We also allow to choose subsets of samples based on the confounding factors and excluded samples will be removed from the heatmap.
In the clustering quality tab, we show the simplified silhouette score for each datapoint in a bar plot. The red line indicated the average silhouette score of the clustering. Finally, the scree plot can be utilized to show the variance covered by the principal components of the underlying PCA.
It is worth noticing that the majority of results within the input folder, and thereby the corresponding plots are optional. If the respective files are missing, the application will warn the user but still produce the remaining plots.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

5.2 Workflow application

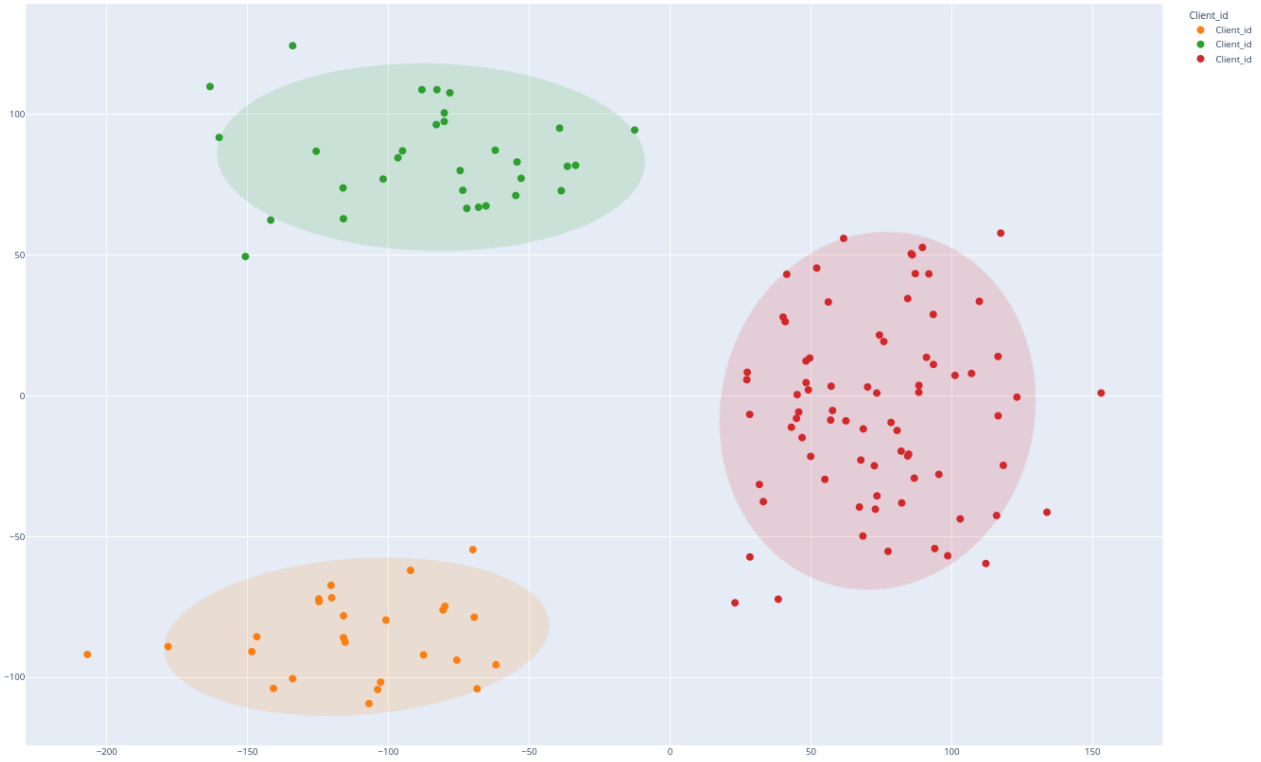

Figure 7: PCA Visualization of the dataset where colors represent the data present at the local contributors

For this deliverable, we wanted to apply the visualization app in a complete clustering workflow to show its usability in a real-world setting. We have therefore the three datasets as described in Section 4.2 as they follow the same library preparation protocol but have been sequenced independently in three different facilities. Consequently, we have used three clients, to simulate three participating institutions, where each client can only access its own sample set. The idea behind the workflow shown in Figure 3 was to A) get an overview over the dataset, B) perform necessary corrections and C) to cluster the samples to see if the resulting grouping is somewhat meaningful to the breast cancer samples. In Figure 7, we show the scatter plot visualizing the first two principal components based on the federated PCA. Clearly, the samples are grouped by contribution site due to very strong batch effects introduced by the sequencing in different institutions. We consequently concluded that we need to apply the ComBat empirical batch effect correction method to subtract the unwanted biases.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

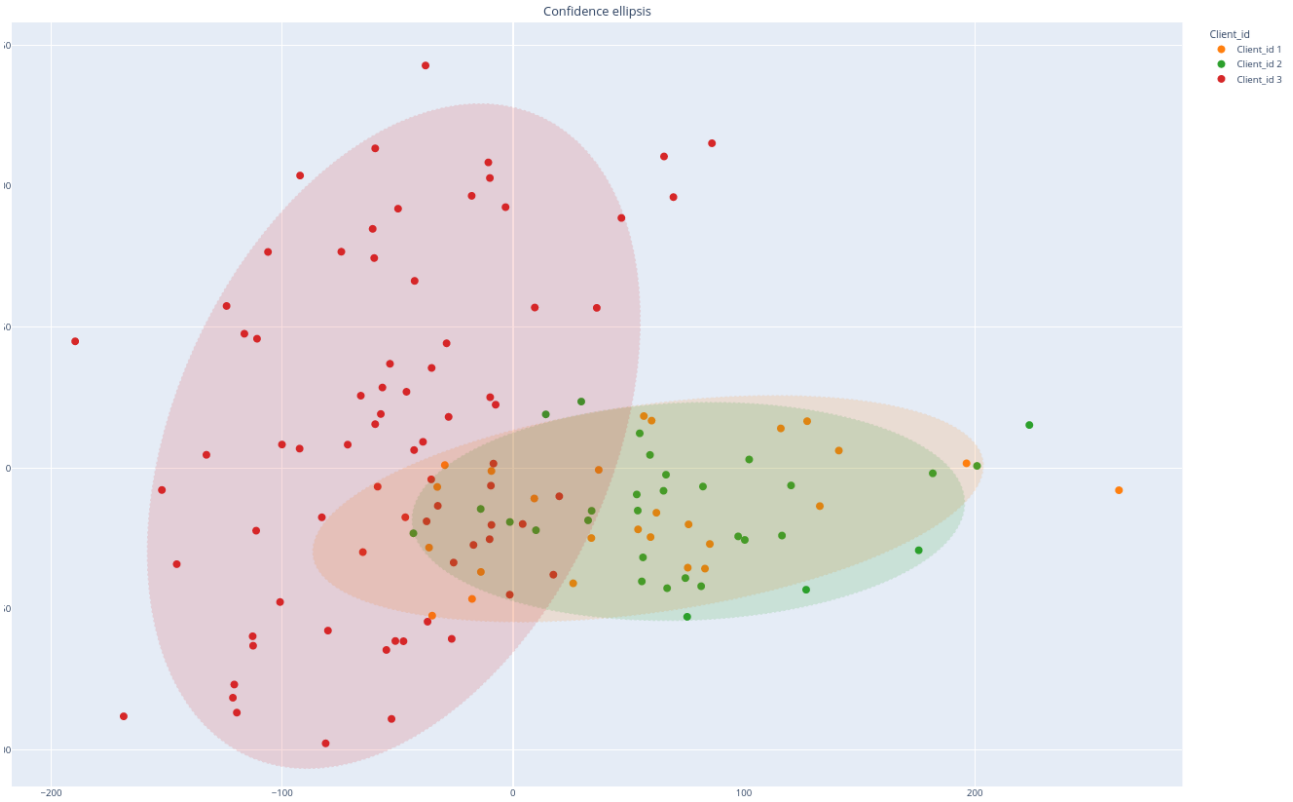

Figure 8: Data distribution across clients after batch effect correction (PC1 vs. PC2)

Afterwards, we have re-run the PCA app and visualized the first two principal components of our new dataset (see Figure 8). It can be seen that the batch effect correction was quite successful, as the local samples from client one and client two are now merged and do not cluster anymore by contribution side. The batch effects separating dataset three from the rest have also been significantly shrunk and the remaining differences are most likely actual differences between the samples. We then proceeded to run a federated K-means clustering with different K parameters on our dataset to see if we can find groups of samples.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Figure 9: Federated Clustering with K-Means based on the batch effect corrected data.

In Figure 9, the results of the federated K mean, for K equals 3, are shown. As expected, the samples are quite difficult to separate into groups, but we have added the gold standard, namely the true classification of the samples as confounders to visualize their distribution within the clustering. Breast cancer can be separated into basal- and luminal-like types, where the latter can be further divided into Luminal subtypes A and B. The luminal- and basal-like breast cancer subtypes can be almost perfectly separated in cluster two and three, while cluster one shows a mixture of both indicating that, although different subtypes, the samples show highly similar expression patterns. In Figure 10, we also show the simplified silhouette score that clearly indicates that all three clusters are in close proximity to each other.

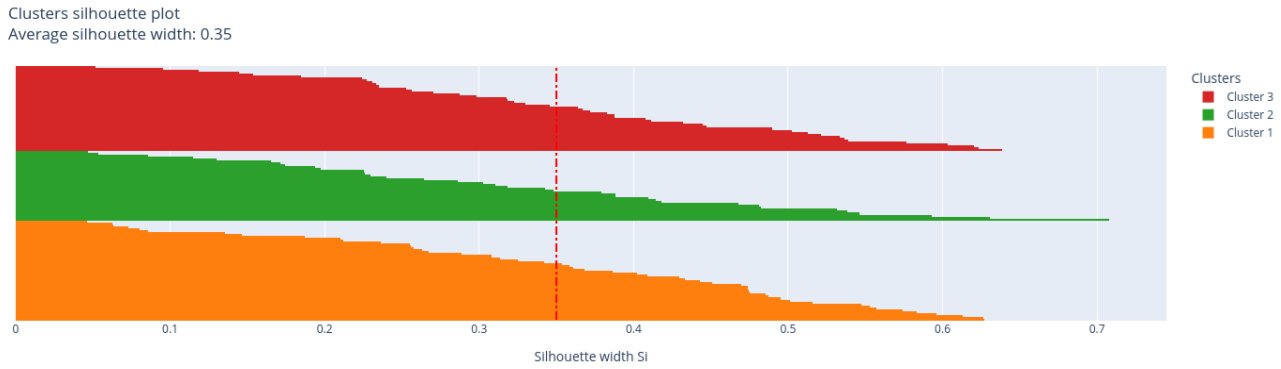

Figure 10: Simplified silhouette score of federated K-Means based on the batch-effect corrected data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

6 Open issues

Throughout the development process of the visualization app and the workflow for the analysis of the real-world data, we have extensively utilized the FeatureCloud platform, the pip package and the extensive testing functionality. We had a constant interaction with our colleagues from UHAM who were quick in addressing our issues and bug reports. The workflow system and the transition of data from one app to another was, as expected, an important step in the development process. Here, especially, non-linear and circular workflows are suffering from very limited support in the FeatureCloud infrastructure. Here, we have already communicated our ideas and suggestions with our colleagues from GND and UHAM and further development to address this has already been initiated.

7 Deviations (if applicable)

None.

8 Conclusion

In this deliverable, we have presented a FeatureCloud application able to load and visualize clustering results. Our development was characterized by close collaboration between SDU and GND, accompanied by close support from UHAM. We were able to show that the FeatureCloud platform at its current state of development is able to support a federated clustering workflow. Although we have shown the principal functionality of our apps in previous deliverables, we have chosen to utilize a dataset much closer to the world datasets. The cluster visualization app is available online and has been equipped with extensive documentation. We have been anxious to develop a user-friendly interface that allows for intuitive interaction with the user. The selection of (groups of) data points based on either visual inspection or confounders is highly useful and will, in the future, allow for circular workflows that are able to re-compute results based on user input. Therefore, our visualization app will serve as a prototype and drive further development towards “human in the loop” interfaces, which also supports explainability and causability (Holzinger and Müller, 2021) (Pfeifer *et al.*, 2022). We have implemented a variety of plot types to visualize as many aspects of a clustering as possible, even though they might not be used by our density-based approaches.

The visualization is often based on the projections or pseudo-projections of our PCA approach, which gives us the ability to visualize important aspects of the dataset across clients, while avoiding the exchange of raw information. Based on that, each contributor can individually remove samples that might represent outliers and would potentially render results unusable. We have demonstrated all features of our app and have applied it on real world data to prove its usability.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

9 References

Hartebrodt, A., Nasirigerdeh, R., Blumenthal, D.B. and Röttger, R. (2021), "Federated Principal Component Analysis for Genome-Wide Association Studies", *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1090–1095.

Hartebrodt, A. and Röttger, R. (2022), "Federated horizontally partitioned principal component analysis for biomedical applications", *Bioinformatics Advances*, Oxford Academic, Vol. 2 No. 1, p. vbac026.

Holzinger, A. and Müller, H. (2021), "Toward Human–AI Interfaces to Support Explainability and Causability in Medical AI", *Computer*, Vol. 54 No. 10, pp. 78–86.

Ligibel, J.A., Dillon, D., Giobbie-Hurder, A., McTiernan, A., Frank, E., Cornwell, M., Pun, M., *et al.* (2019), "Impact of a Pre-Operative Exercise Intervention on Breast Cancer Proliferation and Gene Expression: Results from the Pre-Operative Health and Body (PreHAB) Study", *Clinical Cancer Research: An Official Journal of the American Association for Cancer Research*, Vol. 25 No. 17, pp. 5398–5406.

Park, S., Lee, E., Park, S., Lee, S., Nam, S.J., Kim, S.W., Lee, J.E., *et al.* (2020), "Clinical Characteristics and Exploratory Genomic Analyses of Germline BRCA1 or BRCA2 Mutations in Breast Cancer", *Molecular Cancer Research: MCR*, Vol. 18 No. 9, pp. 1315–1325.

Pfeifer, B., Secic, A., Saranti, A. and Holzinger, A. (2022), "GNN-SubNet: disease subnetwork detection with explainable Graph Neural Networks", *bioRxiv*, 12 January.

Varley, K.E., Gertz, J., Roberts, B.S., Davis, N.S., Bowling, K.M., Kirby, M.K., Nesmith, A.S., *et al.* (2014), "Recurrent read-through fusion transcripts in breast cancer", *Breast Cancer Research and Treatment*, Vol. 146 No. 2, pp. 287–297.

Zolotareva, O., Nasirigerdeh, R., Matschinske, J., Torkzadehmahani, R., Bakhtiari, M., Frisch, T., Späth, J., *et al.* (2021), "Flimma: a federated and privacy-aware tool for differential gene expression analysis", *Genome Biology*, Vol. 22 No. 1, p. 338.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

10 Table of acronyms and definitions

concentris	concentris research management GmbH
GND	Gnome Design SRL
MS	Milestone
MUG	Medizinische Universitaet Graz
Patients	In this deliverable, we use the term “patients” for all research participants In FeatureCloud, we will focus on patients, as this is already the most vulnerable case scenario and this is where most primary data is available to us. Admittedly, some research subjects participate in clinical trials but not as patients but as healthy individuals, usually on a voluntary basis and are therefore not dependent on the physicians who care for them. Thus, to increase readability, we simply refer to them as “patients”.
RI	Research Institute AG & Co. KG
SBA	SBA Research Gemeinnutzige GmbH
SDU	Syddansk Universitet
TUM	Technische Universitaet Muenchen
UHAM	University of Hamburg
UM	Universiteit Maastricht
UMR	Philipps Universitaet Marburg
WP	Work package

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

11 Other supporting documents / figures / tables (if applicable)

Package	Version
attrs	21.4.0
biopython	1.79
Brotli	1.0.9
certifi	2021.10.8
charset-normalizer	2.0.12
click	8.0.4
colour	0.1.5
cycler	0.11.0
dash	2.2.0
dash-bio	1.0.1
dash-bootstrap-components	1.0.3
dash-core-components	2.0.0
dash-daq	0.5.0
dash-html-components	2.0.0
dash-table	5.0.0
Flask	2.0.3
Flask-Compress	1.11
fonttools	4.30.0
GEOparse	2.0.3
idna	3.3
itsdangerous	2.1.0
Jinja2	3.0.3
joblib	1.1.0
jsonschema	4.4.0
kiwisolver	1.3.2
MarkupSafe	2.1.0
matplotlib	3.5.1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

numpy	1.22.2
packaging	21.3
pandas	1.4.1
ParmEd	3.4.3
periodictable	1.6.0
Pillow	9.0.1
plotly	5.6.0
pyparsing	3.0.7
pyrsistent	0.18.1
python-dateutil	2.8.2
pytz	2021.3
requests	2.27.1
scikit-learn	1.0.2
scipy	1.8.0
six	1.16.0
tenacity	8.0.1
threadpoolctl	3.1.0
tqdm	4.64.0
urllib3	1.26.9
Werkzeug	2.0.3
yellowbrick	1.4

Table 1: Full list of required packages for the visualization app.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.