# Federated Patient Stratification with Autoencoders

## Description

Patient stratification is essential in personalized medicine, enabling clinicians to categorize patients based on molecular profiles to predict disease progression and treatment responses. Mass spectrometry (MS)-based proteomics provides a comprehensive overview of protein expressions. However, data sharing between institutions is often restricted due to privacy concerns and regulatory constraints (e.g., GDPR, HIPAA).

Federated Learning (FL) presents a solution by allowing collaborative model training without the need to share raw data. This project aims to develop a federated patient stratification tool using proteomics data, implemented via the FeatureCloud platform. This approach will facilitate collaborative research while maintaining privacy and compliance with regulatory standards.

FeatureCloud is a platform where data scientists can provide workflows or even implement their federated algorithms in the form of apps. Data holders can use a simple frontend to use these workflows and federated algorithms on their data in collaboration.

[Here](https://featurecloud.ai/developers), you can find information about FeatureCloud as a platform for app developers (which you will be in this project. And [Here](https://featurecloud.ai/assets/developer_documentation/index.html), you can find the documentation.

**Unsupervised Approach: Federated Autoencoders**

This project will focus on the unsupervised stratification of patients using Autoencoders. Autoencoders can be utilized to reduce the dimensionality of complex proteomics data, thereby enabling the identification of underlying patient subgroups that may not be apparent from clinical data alone. This data-driven stratification has the potential to reveal novel biological insights. As a clustering algorithm, you can choose k-means, already implemented as FC app (<https://featurecloud.ai/app/fc-federated-kmeans>).

## Required Tasks to pass

1. Run the central version of the method on all data together.
2. Create a prototype for the App (with lists or any other local structure for simulating client-server communications). It is essential for testing the implementation and validating the federated approach in a simpler and faster environment. You can use FedAvg for aggregating local model updates.
3. Run pure Fed-k-means clustering (<https://featurecloud.ai/app/fc-federated-kmeans>) on simulated data and compare the results with VAE.
4. You must be able to present the App or the prototype. So, for the presentation, you should be able to demo with some data.

## Additional Tasks

1. Test the central approach on the real-world data.
2. Create a FeatureCloud App based on the prototype.
3. Performance evaluation: compare federated model performance with centralized model performance. For example, using silhouette score or other metrics.
4. Hyperparameter tuning to achieve better performance.
5. Comparison with Federated unsupervised random forest for privacy-preserving patient stratification (<https://doi.org/10.1093/bioinformatics/btae382>).

## Implementation details

1. Input Data:

* The app should support .csv files, where columns represent samples (i.e., patients), and rows represent features (i.e., proteins).
* The config file should specify essential information such as the separator used in the CSV file and the method (supervised or unsupervised). You can add other relevant settings, such as normalization methods (for example, log-transformation flag).

1. Algorithm:

* The app should support at least one algorithm, e.g., supervised (random forest) or unsupervised approach (k-means clustering or autoencoders).

1. Output

* Samples-Cluster Table:

The output should consist of a samples-cluster table, where each participating client only receives information about its own samples, mapped to the stratified clusters or classification categories. The output should maintain data confidentiality and avoid sharing any information about other participating institutions.

## Datasets

<https://drive.google.com/drive/folders/1DH4axgfDkEonjYVRfhovOShjPIJ6ZilR?usp=sharing>

The folder includes three simulated datasets (more can be generated if needed). Also, you can test the central approach on real datasets (in the same folder):

* JHU cohort <https://pdc.cancer.gov/pdc/study/PDC000110>
* PNNL cohort <https://pdc.cancer.gov/pdc/study/PDC000118>

## References

* You can try models that were developed for scRNAseq data:

A comparison of deep learning-based pre-processing and clustering approaches for single-cell RNA sequencing data. Briefings in Bioinformatics, Volume 23, Issue 1, January 2022, bbab345, <https://doi.org/10.1093/bib/bbab345>

And models and evaluation methods from this benchmark:

Leng, D., Zheng, L., Wen, Y. et al. A benchmark study of deep learning-based multi-omics data fusion methods for cancer. Genome Biol 23, 171 (2022). <https://doi.org/10.1186/s13059-022-02739-2>

* Federated AE and VAE — may not be suitable for this specific task, but may give an impression of how to implement it in a federated manner in general:
* M. Polato, "Federated Variational Autoencoder for Collaborative Filtering," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533358.
* An autoencoder-based confederated clustering leveraging a robust model fusion strategy for federated unsupervised learning <https://doi.org/10.1016/j.inffus.2024.102751>
* FREPD: A Robust Federated Learning Framework on Variational Autoencoder <https://www.techscience.com/csse/v39n3/44055>