Parkinson’s Disease Progression Prediction: Report

Department of Computer Science

Illinois Institute of Technology

April 29, 2023

Authors

Javier Echavarren A20521368

Victor Bóveda A20521372

Carlos Castillo A20520708

**Abstract**

[WIP]

# Introduction

Our project involves competing in a Kaggle competition about Parkinson disease evolution. As a competing group in this competition, our primary objective is to predict MDS-UPDR scores that measure the progression of Parkinson's disease in patients. To achieve this, we will work together to develop a machine learning model trained on data containing protein and peptide levels in subjects with Parkinson's disease and normal age-matched control subjects.

# Data

The competition data consists of three files:

* train\_peptides.csv: Contains mass spectrometry data at the peptide level, including visit ID, visit month, patient ID, UniProt ID, peptide sequence, and peptide abundance.
* train\_proteins.csv: Provides aggregated protein expression frequencies, with details such as visit ID, visit month, patient ID, UniProt ID, and normalized protein expression (NPX).
* train\_clinical\_data.csv: Includes clinical data like visit ID, visit month, patient ID, UPDRS scores for different parts (1-4), and information on whether the patient was on medication during the UPDRS assessment.

# Exploratory data analysis

The EDA was composed of the following steps:

* Check the training datasets: We firstly looked to every train dataset. This way we can figured out how the data was stored and the dtypes of the features. We discovered that the data in train\_proteins and train\_peptides datasets were stored in a long format style.
* Modify the tables styles: In order to change from long to wide style, we pivot the train\_proteins and train\_peptides tables. After this step we obtained two tables with every peptide and protein as a feature.
* Merge: After obtaining the pivot tables, we merged them to the train\_clinical data frame. We used a left join so we would not miss any data from the train\_clinical dataset.
* Fill missing data: Once we had all the data in one data frame, we had to deal with the missing data. Due to we are working with time series, we thought that the best idea would be to use a linear regression model in order to fill the NA values.
* Scale the data: Finally, we had the final train data frame with all the data. Now we scaled the data so it would be easier and quicker to create a model. We use a standard scaler function from sklearn to perform this step.

# References