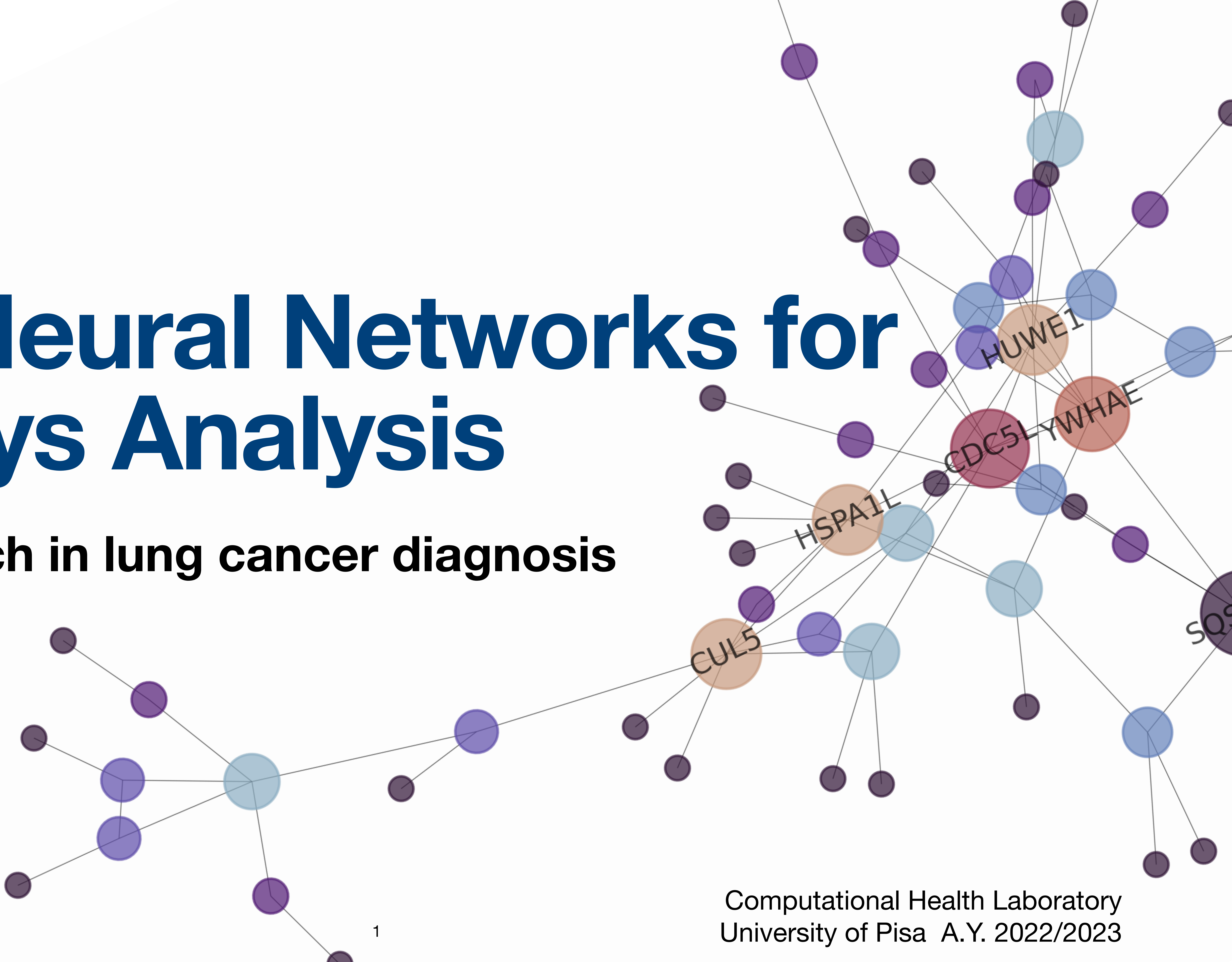
Graph Neural Networks for Pathways Analysis

A novel approach in lung cancer diagnosis

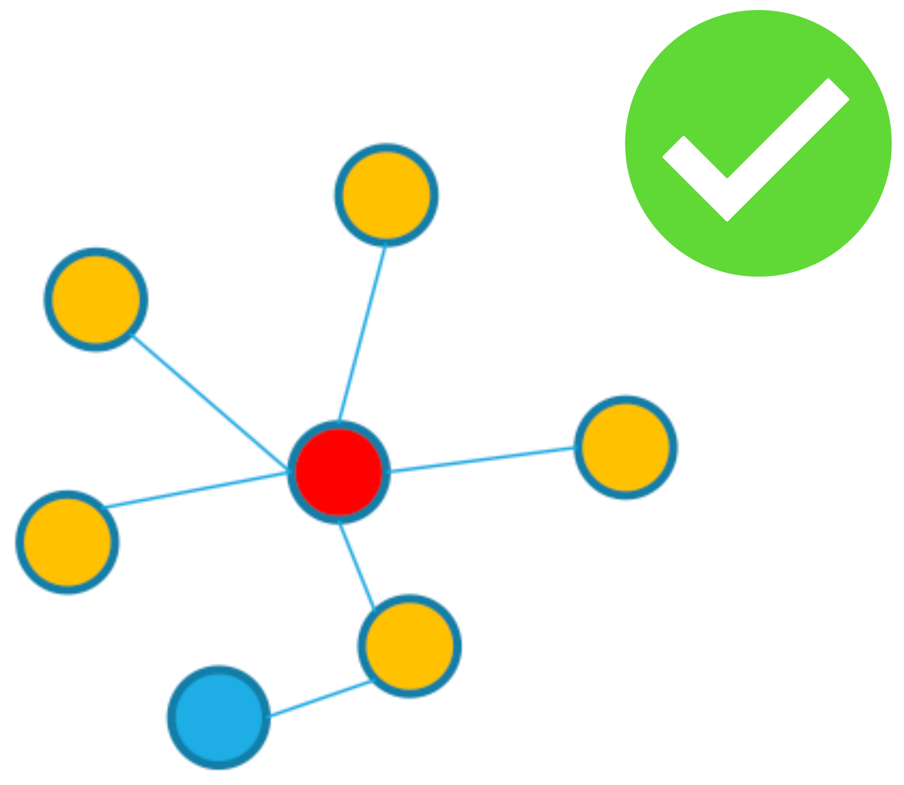


Luca Miglior
Leonardo Stoppani

Computational Health Laboratory
University of Pisa A.Y. 2022/2023

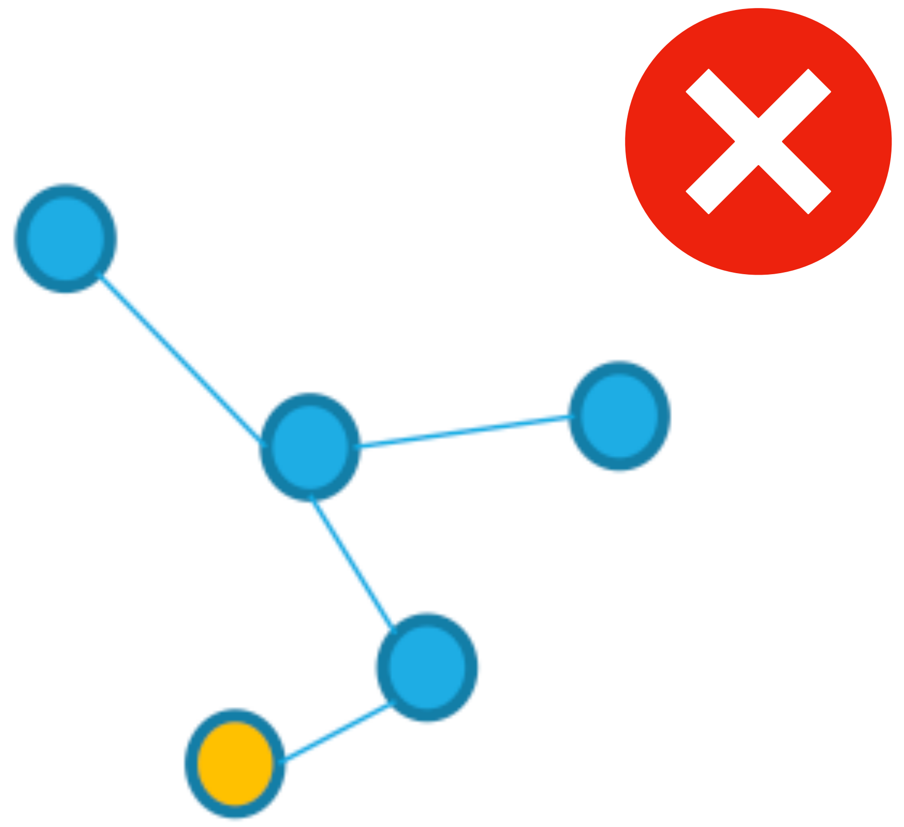
Main project idea

Lung cancer is a world leading cause of death, early diagnosis is fundamental for a benign prognosis



Interesting result in classification with high-throughput microarray data and biomarkers

New idea: exploit the a structured space of biomarkers, introducing protein to protein interactions information



Graphs and GNNs perfectly suit this work

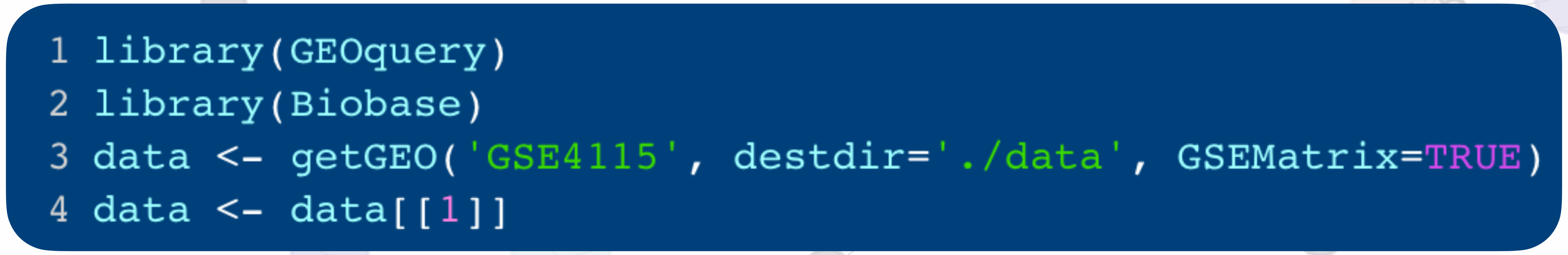
Dataset structure and project setting

Dataset we used is publicly available on the GeoQuery Database.

It consists in 192 samples of Affymetrix HG-U133A microarray data

22,215 genes expression levels taken from airways epithelial cells

Also phenotype data for each sample (gender, age, and other medical assessments)



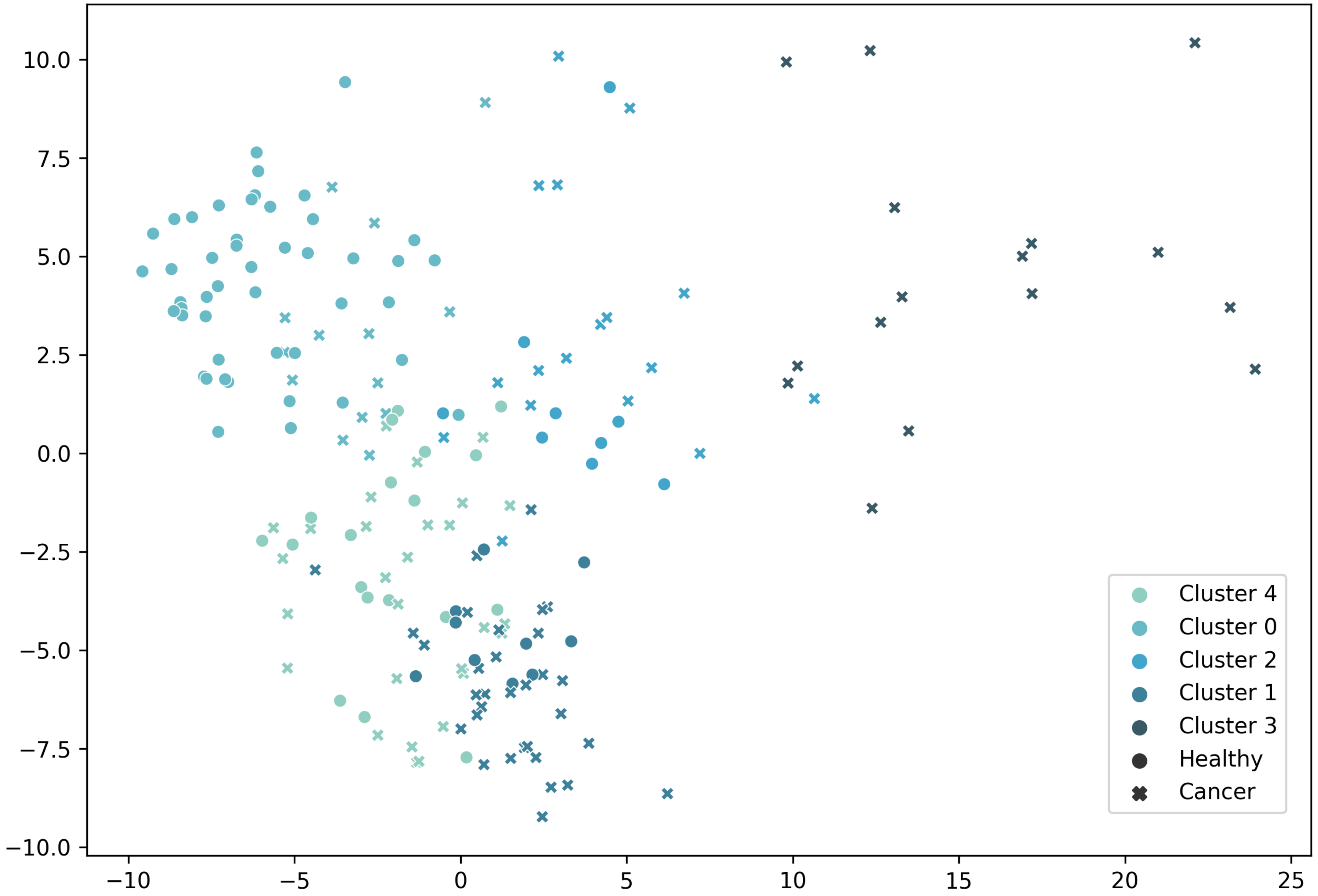
```r
library(GEOquery)
library(Biobase)
data <- getGEO('GSE4115', destdir='./data', GSEMatrix=TRUE)
data <- data[[1]]
```

Biomarkers Identification

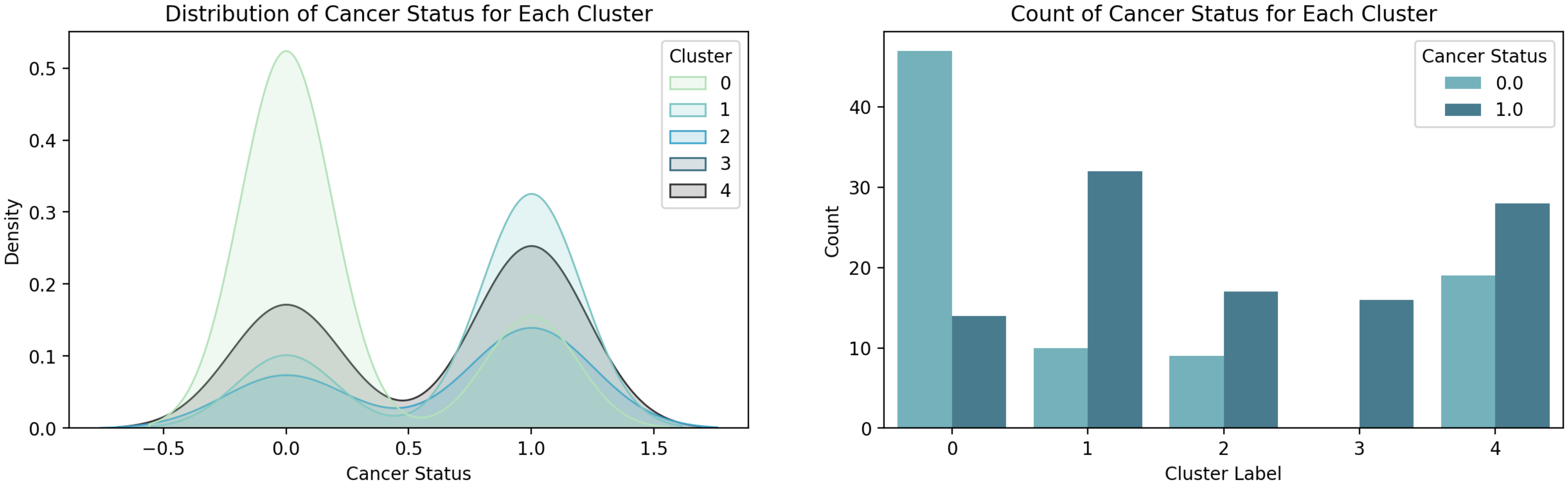
Dimensionality reduction and relevant biomarkers identification was performed processing the dataset with LIMMA R package

Differential expression analysis identified 141 biomarkers (P=0.01)

Data were then exported for further processing with Python



Unsupervised Learning



Unsupervised learning techniques (clustering) showed that in fact our biomarker’s quality is good and discriminant towards both classes.

Graph building

We developed a 2 steps algorithm:

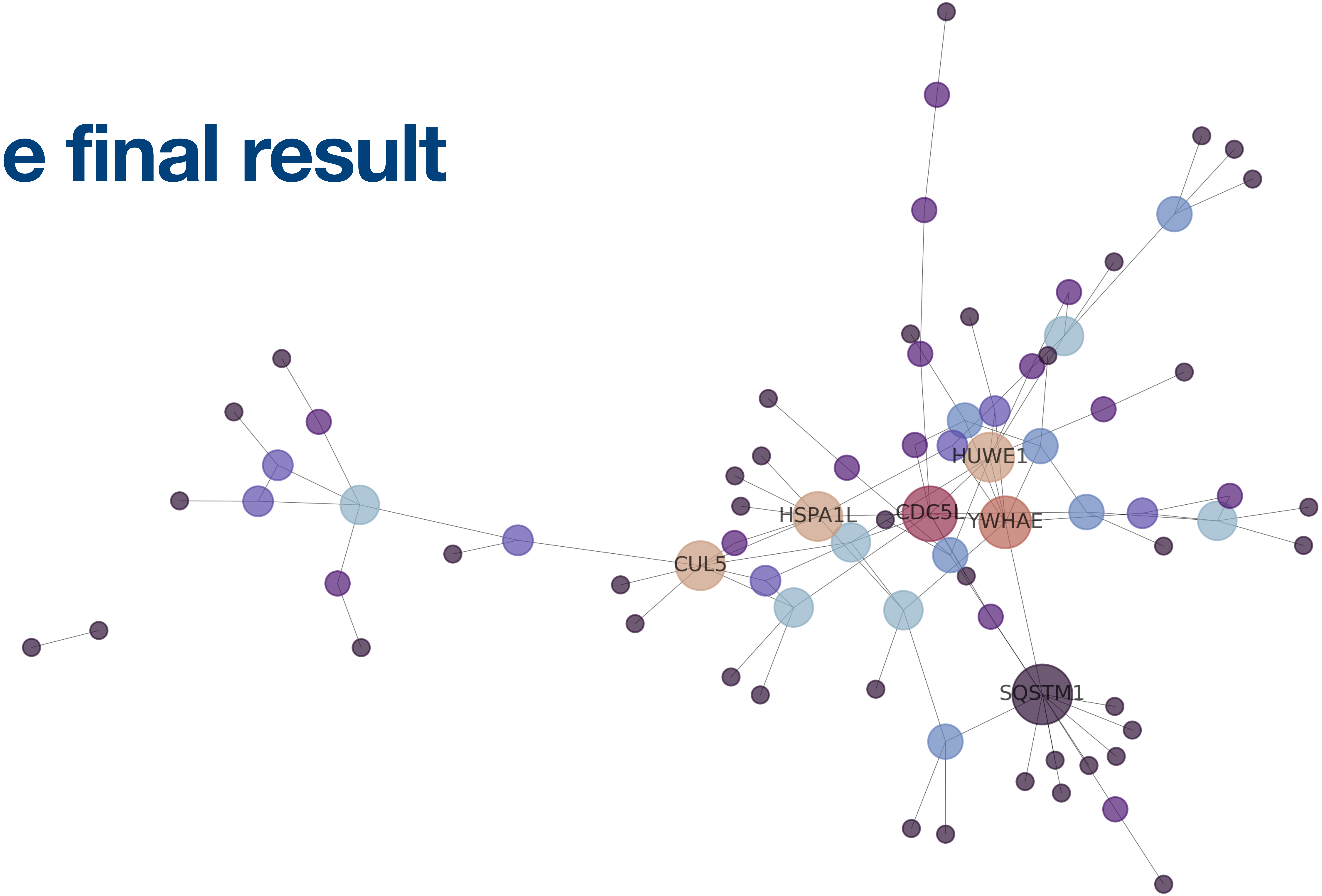
1. For each biomarker query BioGrid and find interactions

2. Then, build the adjacency matrix, from the obtained results

This matrix represents an unique graph topology for each patient of the identified biomarkers

We then filled node labels with gene expression levels

The final result



Model Architecture

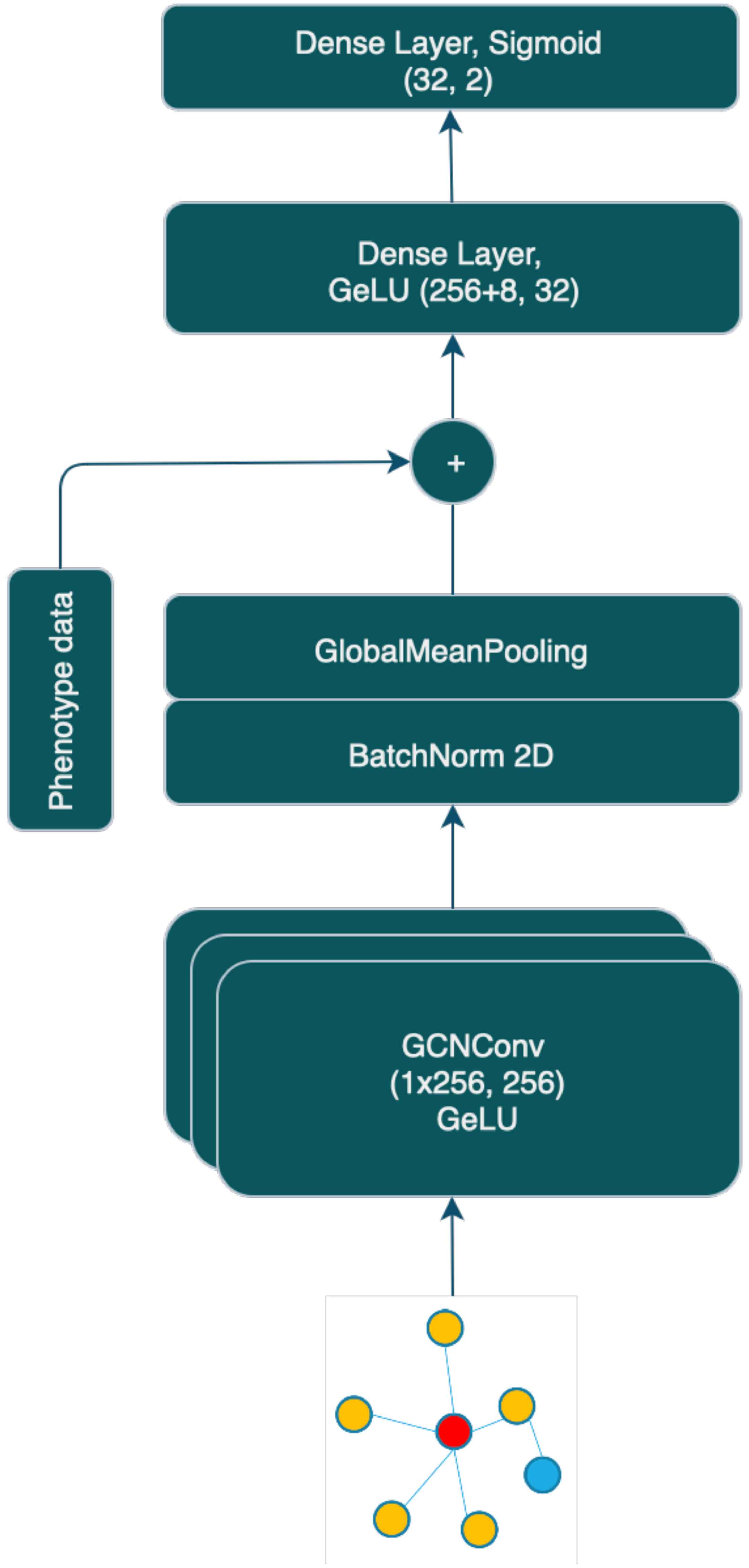
Model is composed by a single and shallow, Message Passing, Graph Convolutional Layer (GCN)

GCN generates an embedding for each graph

Result is then Batch-Normalized and pooling is performed to “flatten” the resulting embedding

The embedding is now concatenated with patient’s phenotype data

Standard MLP performs the final classification



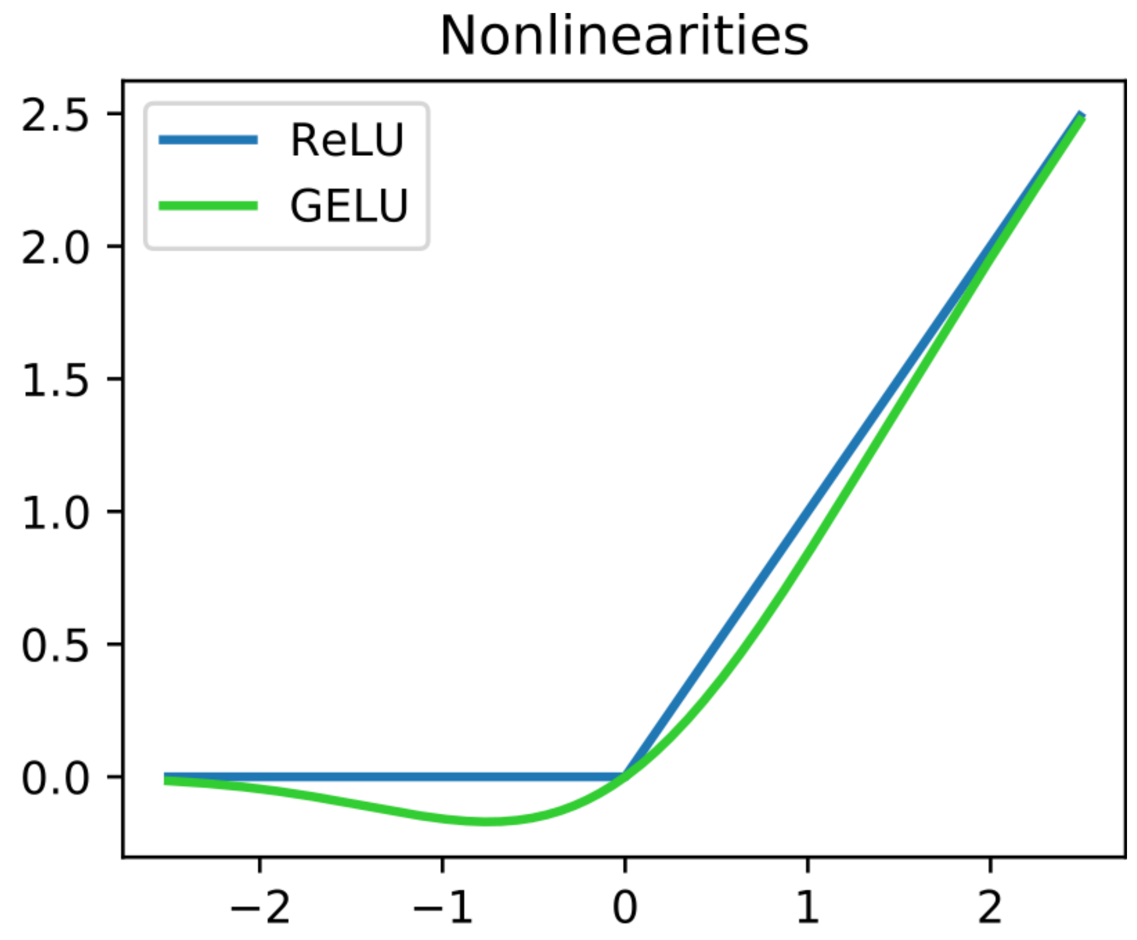
Model Training

The model was developed in Python, exploiting PyTorch and PyTorch Geometric framework

Some tech informations: 200 training epochs, Adam optimiser and MSE loss function; L2 Regularisation

Our neural network was quite shallow and fully differentiable.

Shallowness and differentiability avoided node smoothing and helped in improving accuracy



PyG

PyTorch

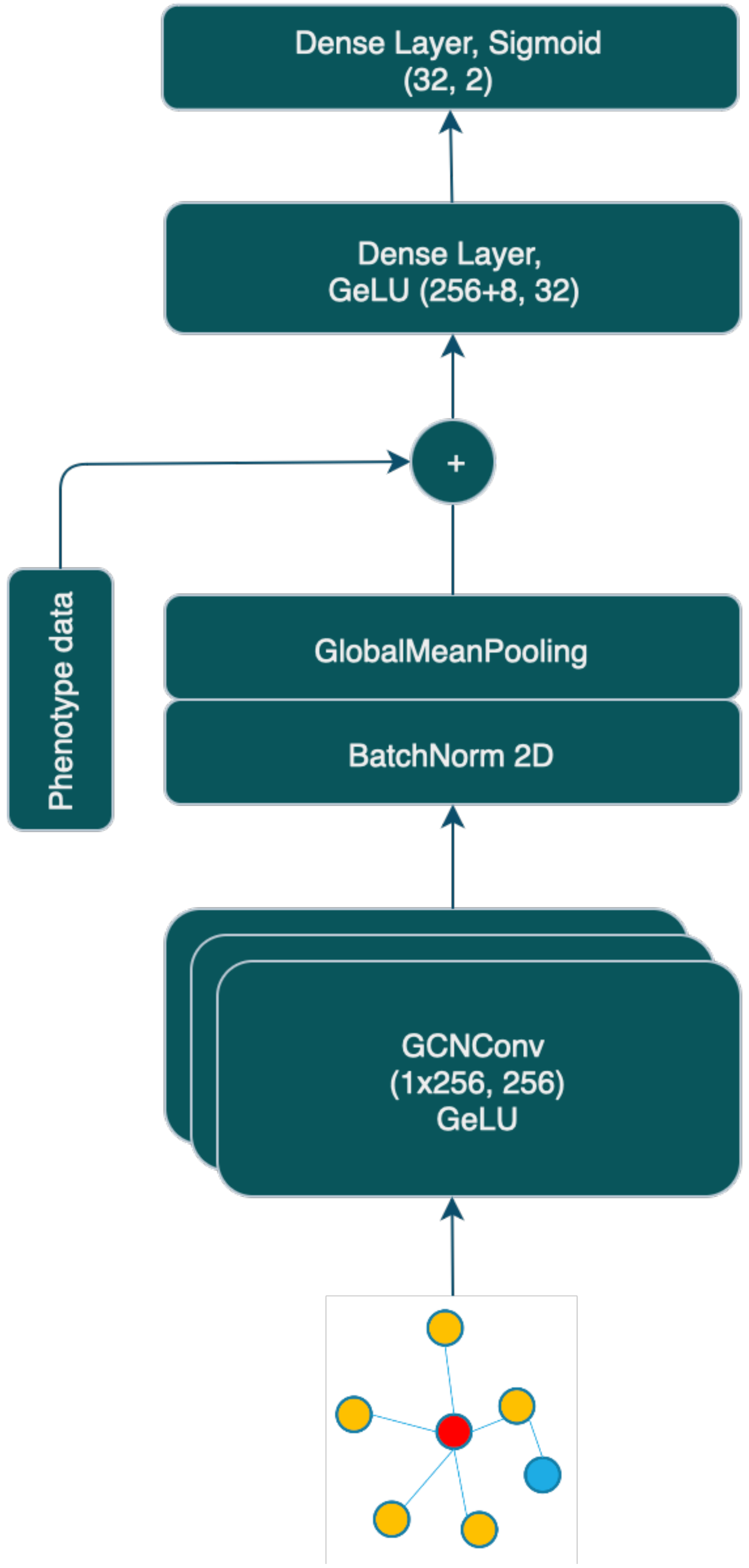
Model Selection

Model selection was performed with standard hold-out technique.

Dataset divided into Dev Set (70% of data) and Test Set (30%). Balanced classes.

First, we trained on 70% of our DS, then validated performance on the remaining 30% VS

We finally tested the model on the hold-out 30% test split, getting around 93.2% accuracy

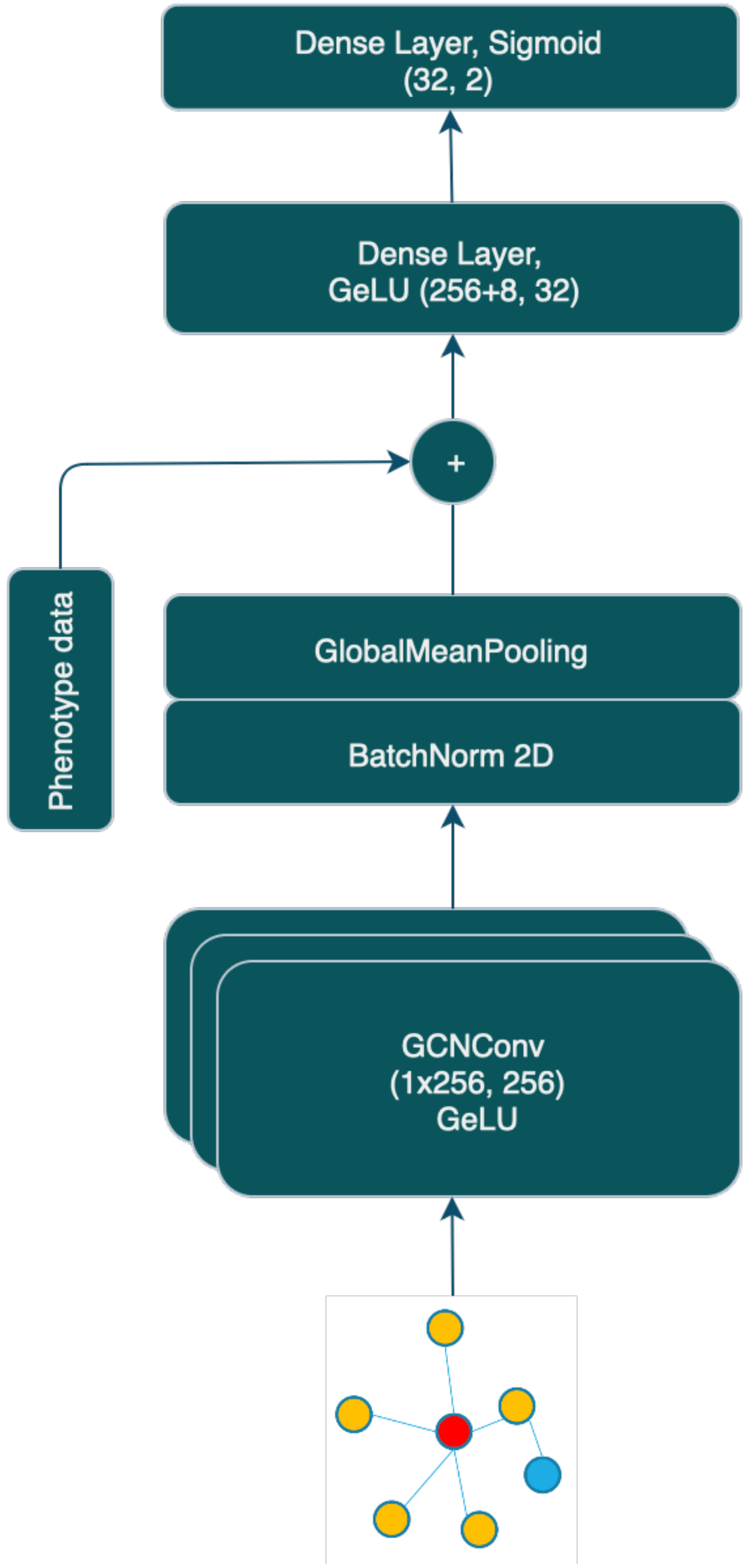

Dense Layer, Sigmoid (32, 2)
Dense Layer, GeLU (256+8, 32)
+
Phenotype data
GlobalMeanPooling
BatchNorm 2D
GCNConv (1x256, 256) GeLU

Risk Assessment

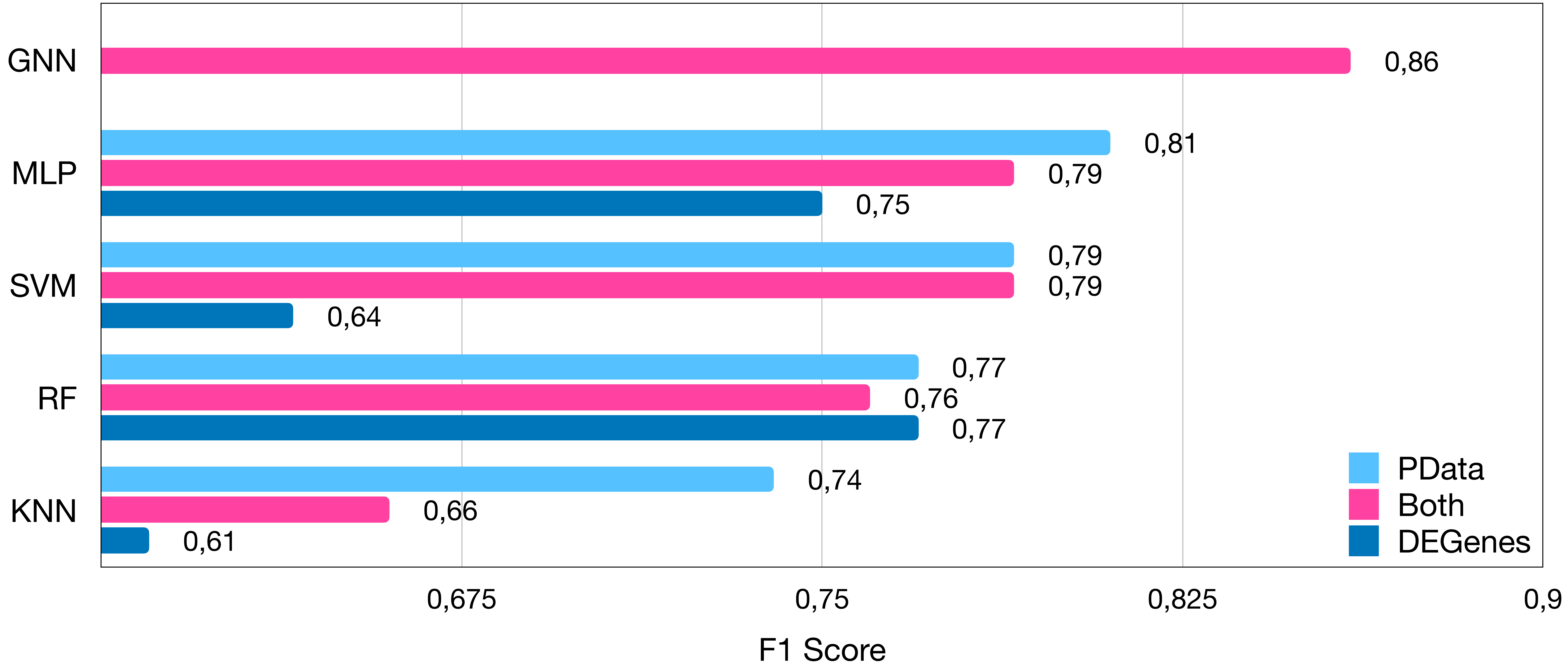
After model selection, we performed risk assessment by 3-Fold cross validation.

We re-trained on a 70% split of the whole dataset, then tested on a 30% balanced and independent test set for each fold.

	Train F1 score	Test F1 Score
Fold 1	91.1%	93.2%
Fold 2	92.2%	81.3%
Fold 3	90.6%	85.0%
		Average: 86.3%



Final Results



Luca Miglior
Leonardo Stoppani

Computational Health Laboratory
University of Pisa A.Y. 2022/2023

Conclusions

Results are promising: adding structural information to data lead in a significant increase in model's performance.

There are still some challenges to overcome:

- Assess actual model generalisation capability on other datasets
- Test this approach for other diseases