GENOMICS-FIND YOURSELF APP

APP V1.0

TABLE OF CONTENTS

1. INTRODUCTION

The Find Yourself App is inspired by the 1000 Genomes Project which studied the human genetic variation among 1000 unidentified individuals across the world. The goal of that initiative was to use the next generation sequencing technologies to provide a resource of genetic variants like SNPs (single nucleotide polymorphisms) to study the common human diseases that are associated with their genes.

In this project, our app demonstrates a visualization tool that helps in visualizing the high-dimensional genetic variants (SNPs) data into a low-dimension space using PCA (principal component analysis). It displays the interactive clusters of individual groups from across the world. Additionally, it also showcases the genome dependency wheel to further explain the relationship between individuals and their genetic variants.

To make this data visualization tool more powerful and actionable as a real-world application, we wanted to explore the behavior of new datapoints (or "new human samples") in our dataset. To explain how these new data points will be classified, we demonstrate an end-to-end machine-learning data pipeline. This pipeline performs data pre-processing, data aggregation, one-hot encoding, dimensionality reduction, model training, and final testing on the new data points.

2. PROBLEM STATEMENT

The genomics research collaborative initiative wants to accomplish the scientific goal of understanding the patterns of genetic variants like SNPs (single nucleotide polymorphisms) among human populations across the world. The genetic data is complex and high-dimensional and there is a need to visualize the data in a simplified manner.

3. FIND YOURSELF APP PROTOTYPE

The Find Yourself App displays an interactive dashboard of genomic samples data visualization in 3-Dimensional mapping. It also shows the dependency between the samples and their genetic SNPs(or variants). It shows 5 main population groups of individuals from America, Africa, Europe, East Asia and South Asia regions of the world

The app is engineered to render as a web application as seen in Figure 1. below. The app can also be visualized using a mobile device or tablet.

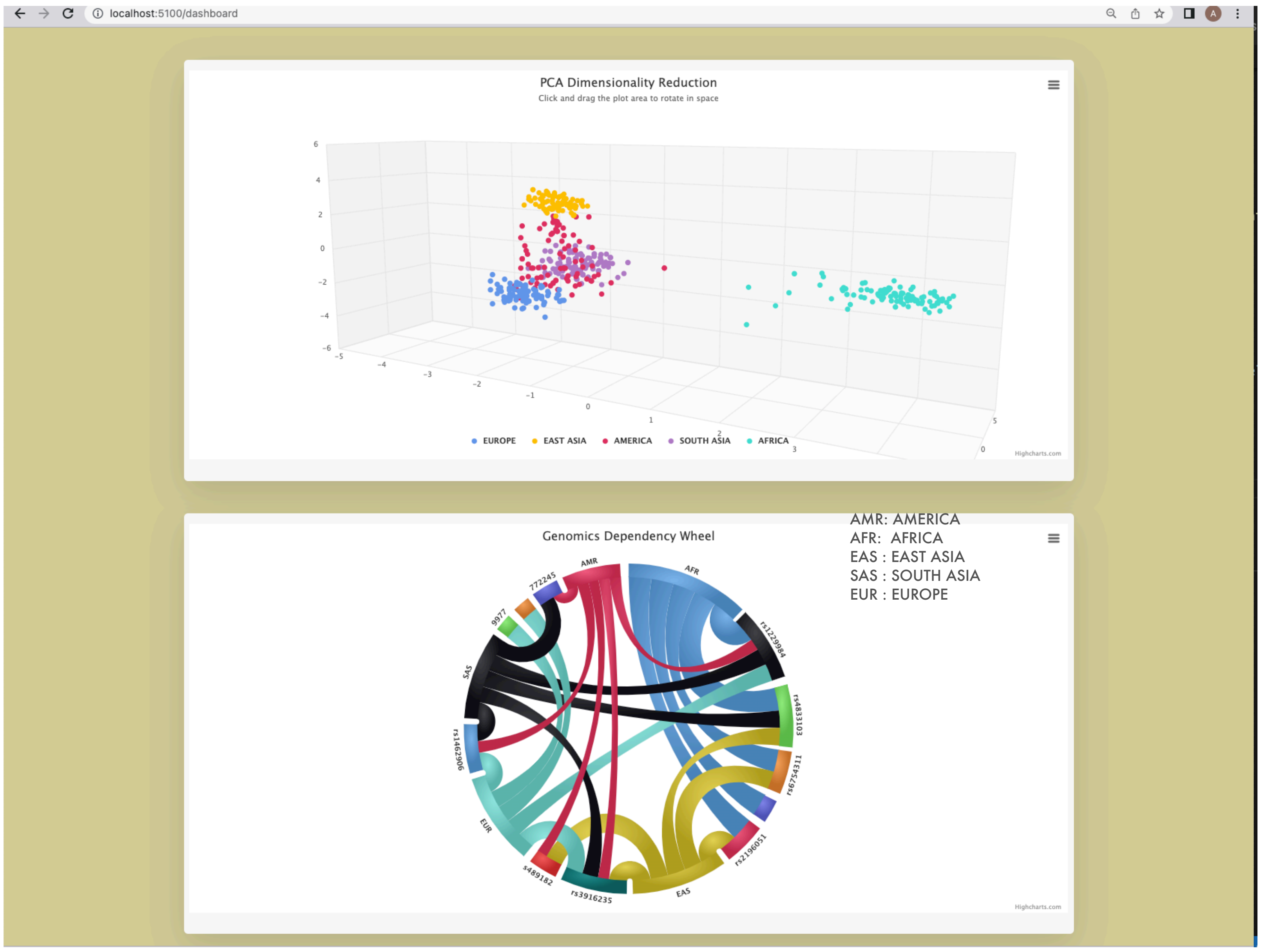

Figure 1. Find Yourself App

4. METHODOLOGY

4.1 High Level Design Architecture

The figure 2 shows the high level design architecture diagram for the find yourself web application. The following steps are performed in this workflow

1. IGSR Data Pull

The IGSR genomics data is pulled from the FTP Server data portal [1],[4]. The data contains the 1000 genome samples in ".panel" format and SNPs data in the ".vcf" format.

2. Raw Data Storage

The data is pushed into a raw data storage system. Ideally, in production, we will use a instructed data storage like Azure Blob Storage

3. Machine Learning Data Pipeline

The ML data pipeline is responsible for the the following services- data preprocessing, one-hot encoding, dimensionality reduction, model classification

4. Data Aggregation

The filtered and transformed structured data is then ingress into the postgreSQL database to store the clean genome data and it's transformation results

5. Visualizing Data in the Web Application

The application frontend is a javascript based web application with python flask as backend server side. The data refreshes using the postgreSQL

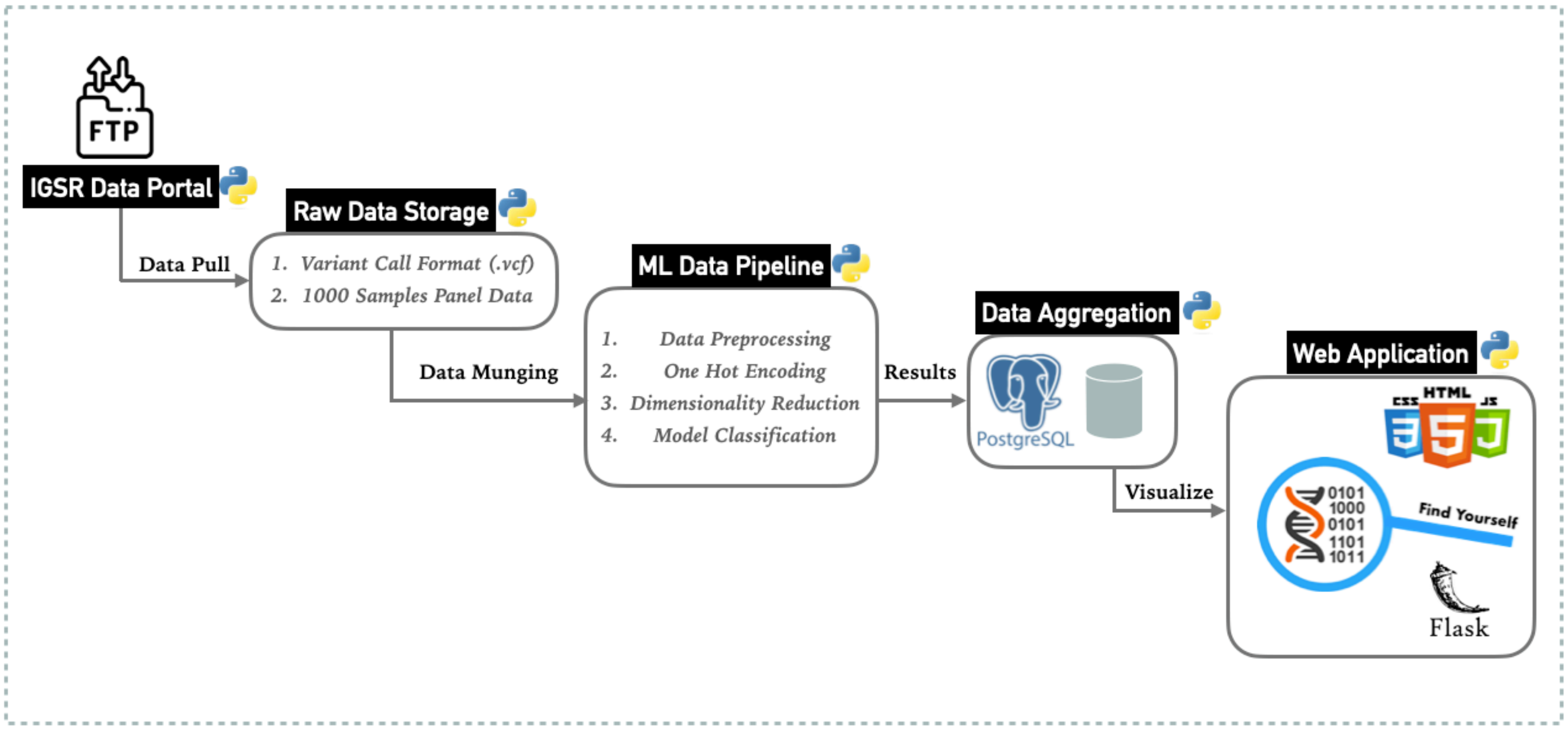

Figure 2. High Level Design Architecture

4.2 Software Design Architecture

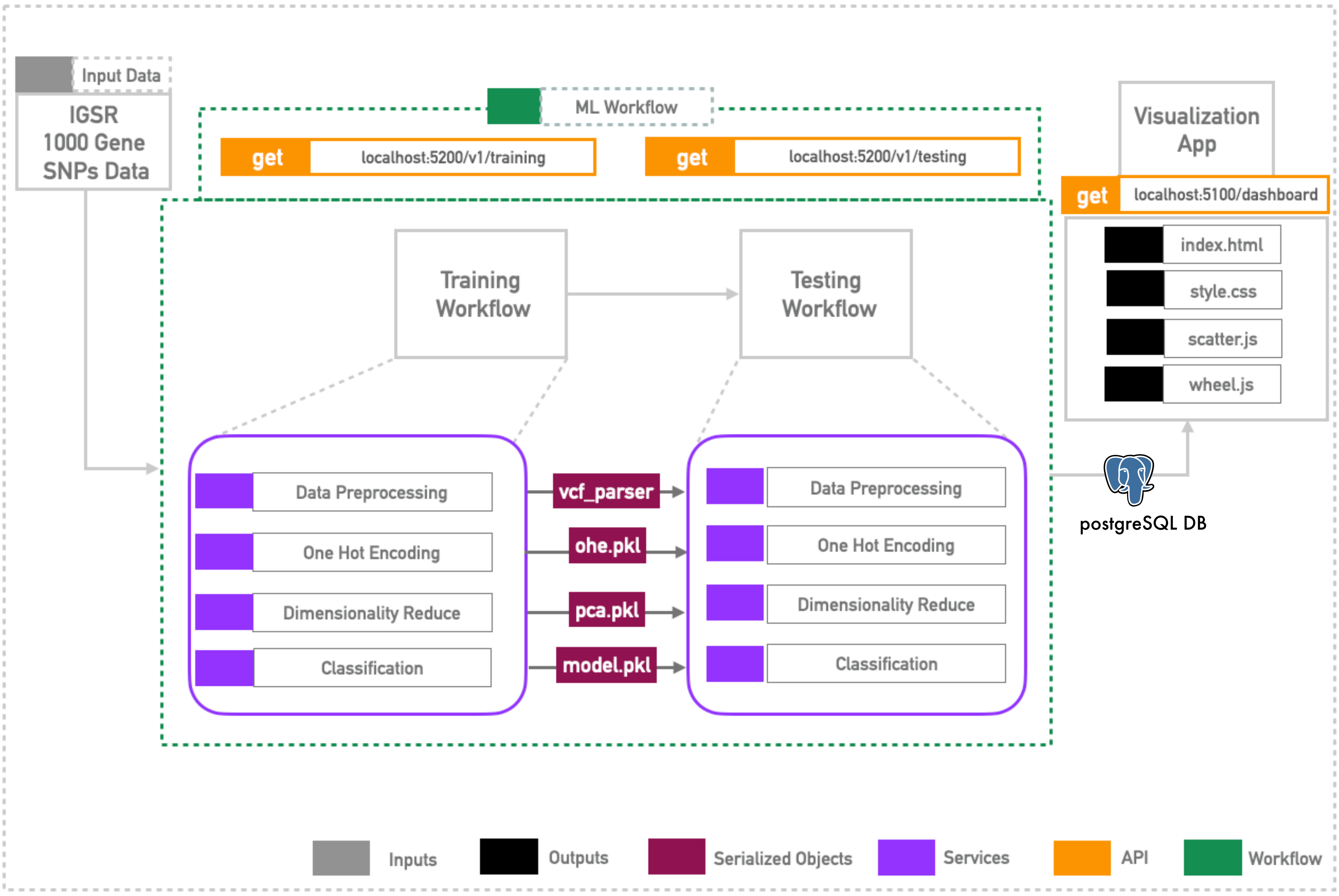

Figure 3. Software Design Architecture

The figure 3 demonstrates the software design architecture.

- The software design pattern is built on the philosophy of modularization and packaging. The software is designed with a collection of modules and services to achieve code reusability and extensibility

- The ML data pipeline workflow for training executes as an api with a GET request on localhost:5200/v1/training. This api is broken down into smaller services that produce serialized objects which will be used for testing api. The ML data pipeline workflow for testing executes as an api GET request on localhost:5200/v1/testing. This api uses the serialized objects from the training workflow to perform the ML tasks

- The data from training and testing is flushed into the postgreSQL database

- The web application performs data refresh by pulling the data from the postgreSQL db

4.3 Object Oriented Design

- The software application using an Object Oriented Design where all the methods are developed using Classes and Function objects which helps in the reusability of the code
- There is a "__init__.py" file in every directory and sub-directory to transform the directory into a python package so that cross-module communication is possible
- The services package has dedicated services to perform an isolated task which makes it easy to scale it across other applications
- The project modules uses object oriented programming concepts like inheritance and polymorphism

4.4 FrontEnd App Architecture Design

- The front-end of the app is created using Javascript, HTML and CSS and it renders an interactive visualization dashboard
- The backend is a server-side rendering flask based application that runs on a WSGI (Web Server Gateway Interface) server

4.5 Limitations

- The database is currently locally hosted, so the database operations will require a local postgreSQL database
- The app is currently hosted locally. To share it will internal and external stakeholders, it will be required to host this application on a virtual machine or deploy as docker image on the virtual machine

5. FUTURE WORK

The future motivation is to generalize the workflows and make them scalable. The future steps are as follows.

- Generalizing the dimensionality reductional module to add support for t-SNE and U-map
- Deploy the workflows as containerized applications using docker to make it scalable
- Deploy the postgreSQL database on a virtual machine
- Deploy the unstructured Azure Blob Storage service
- Generalize the frontend application to add multiple dashboards

6. REFERENCES

[1] https://www.internationalgenome.org/data

[2] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature* **491**, 56–65 (2012). https://doi.org/10.1038/nature11632

[3] https://www.kaggle.com/datasets/kevinarvai/ancestry-informative-snps

[4] http://ftp.1000genomes.ebi.ac.uk/

7. APPENDIX

1. GitHub Project URL : https://github.com/akshaydarora/genomics