Google Summer of Code 2019
Canadian Center for Computational Genomics

March 28, 2019

Project Info :

- **Project title :** Human history and data visualizationv (HHDV)
- **Project short title :** Dimensionality reduction and visualization
- **URL of project idea page :** C3G HHDV project idea page

Bio of Student :

- I have been taking part in a lot of online and off-line hackathons in the past 2 years.
- Machine hack- predict the beer score, where I was placed 7th.
- Axis bank AI hackathon (National hackathon) where I was part of top 23 teams
- I presented a PoC paper in IEEE CCEM Bangalore 2018 and got 3rd place. The topic was, ”Signature recognition and verification using 1D convolution and HOG descriptors”.
- During this entire journey of machine learning, I always wondered what would let me visualize these high dimensional data and I knew for a fact that techniques like PCA could help me in this, but I never got time to learn them.
- Now due to this project, I am very excited to explore more in this domain and hope to make a good contribution to the open source community by helping them visualize genotype data in a simple language like python.

- Apart from tha,t I am learning more about population structure and Genomics from my biology teacher so as to have a deeper understanding of the domain while visualizing the data.
- I am more into mathematics and hence I am also building my knowledge of abstract algebra and linear algebra. This will help me to compete in LIMIT 2019 (An open book mathematics competition.)
- As a hobby, I play table tennis and I was also a part of playing 11 in the school cricket team. Took part in state-level shotput competition.

Contact Information :

- **Student name :** Shivaraj B H
- **Student postal address :**
  College Address
  Dayananda Sagar University, opposite bus stop, Near kudlu gate, Chikkabegur, Srinivasa Nagar, Hal Layout, Bengaluru, Karnataka 560068

  Home Address
  House no. 75 Shiva Nilaya, Jnana jyothi nagar jnana bharti post, bengaluru - 560056, India
- **Telephone(s) :** +91 (725) 9592595
- **Email(s) :** sbh69840@gmail.com
- **Other communications channels :** Skype_id : live:596832ddabb14976

Student affiliation :

- **Institution :** Dayananda Sagar University
- **Program :** B.Tech in Computer Technology
- **Stage of completion :** 1 year
- **Contact to verify :** Dr. Chandra Vaidyanathan ((91) 8105723020)

Schedule Conflicts :

- No conflicts

Mentors :

- **Mentor name :** Alex Diaz-Papkovich
- I was done with my selection test on 17th of March 2019, then I mailed Dr. Simon Gravel. I was later redirected to my mentor, Alex Diaz-Papkovich, where I shared a brief intro about me and the reason I want to be a part of this project. I had a bunch of queries that were resolved by the mentor and I was satisfied with all the replies I received. I bugged him a lot of times asking various doubts regarding the project and my implementation approach, all the doubts were cleared. I thoroughly understood what they wanted out of the project and used those tips to write the proposal.

Coding plan and methods :

- **Language : Python 3**
- **External python libraries (might use) :**
  - scikit-allel
  - zarr
  - scanpy
  - matplotlib
  - seaborn
  - numpy
  - sklearn
  - umap-learn
  - MulticoreTSNE
  - others...
- **I will be dividing the project into six steps :**
  - **Step 1 :** Fast genotype data (mostly VCF) loading and randomizing data.
  - **Step 2 :** Write code to combine different datasets and test the results.
  - **Step 3 :** Write code to visualize the individual or combined dataset, with different functions for each of these approaches :

    - PCA
    - UMAP
    - PCs-UMAP (taking PCs as UMAP input)
    - t-SNE
    - PCs-t-SNE (taking PCs as t-SNE input)
    - p-value boxplot (between population group and phenotype) for any phenotype data (example: age-adjusted height) present in the dataset
  - **Step 4 :** Write code to consider all the components that have PCs as input to plot a range of PC values to compare the effect of taking a different number of PCs to plot the output.
  - **Step 5 :** Put all of these small bits of code and create a python library.
  - **Step 6 :** Testing the implementation on HRS, UKBB, 1KGP datasets.
  - **I have kept a few additional days in between each evaluation phase, a week before the final week and also a week at the beginning of the coding period for any unfortunate situations.**
    If nothing as such happens, which I hope, I will be dedicating this time to think of auto-UMAP (to select optimum minimum distance and nearest neighbor parameters.)
  - At the end of every week of coding period, I will commit the code (even in between the week) and keep my mentor updated.

Timeline :

- **April 9 - May 6 18:00 (Before announcement):**
  - Learn more about population structure.
  - Search for different open source VCF format genotype datasets online.
  - Try my selection test on UKBB and HRS dataset.
  - Combine HRS, UKBB and visualize the results.
- **May 6 18:00 UTC - May 26 (Community bonding) :**
  - Get to know different projects in C3G and their respective students.

  - Learn their requirements and re-structure my plan accordingly.
  - Ask them for help if I don't understand anything and also help them if they don't get anything.
  - Interact more with my mentor and come up with new and innovative approach.
  - ask the mentors and students if they would need any additional visualization feature that would help them in their project, in that sense I will be solving few problems that users of this software would be facing.
  - update my mentor regarding the things I will be picking up.

- **May 27- Jun 2 (Coding officially begins) :**
  - Beginning Step 1.
  - Collect the data (including HRS, UKBB, 1KGP) and create a proper list of these datasets with their respective features and what characteristics I can look at using each dataset.
  - Go through the documentation of Zarr library and look for an efficient way to convert the VCF format to Zarr readable format(Not because VCF is bad, but Zarr loads data in a fraction of seconds, conversion is what takes time).
  - commit the code and update my mentor.

- **Jun 3- Jun 9 :**
  - Give 2-3 more days, if necessary, to complete the code for the Step 1 and then begin Step 2 in the second week.
  - Beginning Step 2.
  - Firstly, I will write the code using numpy to combine 2 or more given datasets, considering the fact that their genotype data structure is similar and also combine their characteristic data file (example population code, ethnic background) according to the unique sample ID.
  - commit the code and update my mentor.

- **Jun 10- Jun 16 :**
  - Combining Genotype data is fine but the problem lies at combining their population phenotypic data as one file may contain and another may not so the best approach would be to not combine this file and label them separately and also access them separately while coloring the plot, but I would not give up here, I will take some time from the third week and rethink.
  - Later this week I will code the functionality out.
  - commit the code and update my mentor.
- **Jun 17- Jun 23 :**
  - Beginning Step 3.
  - Start implementing the first three of the six visualizations as mentioned in the above section. PCA, UMAP, and PCs-UMAP.
  - commit the code and update my mentor.
- **Jun 24 - Jun 28 (Phase 1 evaluation) :**
  - Work on Phase 1 evaluation and submit it before deadline.
  - I will be using some time from here in case I was unable to finish any of my previous works.
- **Jun 29 - Jul 7 (Work period begins):**
  - Continuation of fourth week.
  - Start implementing the next three of the six visualizations as mentioned in the above section. t-SNE, PCs-t-SNE and p-value boxplot.
  - commit the code and update my mentor.
- **Jul 8 - Jul 14 :**
  - Beginning Step 4.
  - Write code to accept a range of a different number of PCs and then return a plot comprising of these plots.

  - commit the code and update my mentor.

- **Jul 15 - Jul 21 :**
  - Go through all the visualizations once and check for faults.
  - Many a time when trying to code quickly we miss out on some details related to the plots and also we make it very clumsy. To avoid that I have decided to give one full week to make every plot more readable and beautiful.
  - commit the code and update my mentor.

- **Jul 22 - Jul 26 (Phase 2 evaluation) :**
  - Work on Phase 2 evaluation and submit it before deadline.
  - I will be using some time from here in case I was unable to finish any of my previous works.

- **Jul 27 - Aug 5 (Work period begins):**
  - Beginning Step 5.
  - During this week I will be putting in all the bits of code I have created into a formal code and create a python library out of it.
  - commit the code and update my mentor.

- **Aug 6 - Aug 11 :**
  - In this week I will create an IPython notebook demo to use all the functions in the library, for the ease of users.
  - This is a part of Step 5.
  - commit the code and update my mentor.

- **Aug 12 - Aug 18 :**
  - Beginning Step 6.
  - In this week I will test the IPython notebook with HRS, UKBB, 1KGP datasets and others which I will look for in the first week.

  - commit the code and update my mentor.

- **Aug 19 - Aug 26 (Final week) :**
  - During this week I will make sure that my documentation is perfect and I provide both Tex file and the PDF as open-source files so that anyone would be able to edit and update, on accepting the request.
  - Submit the final code to my mentor.
  - Submit the final evaluation.

- **Aug 26 - Sep 2 (Mentor submit student evaluation) :**
  - Just hope I be a good student.

- **Sep 3 (Results announced) :**
  - Wait for the result and meanwhile think of a GPU implementation of these visualizations.

Management of coding project :

- Write the initial coding steps in the IPython notebook, which makes it easier to look through while writing the main library.
- Keep committing the code after completion of every task, at max once a week.
- Keep my mentor updated about the progress at least once in two days.
- Store all the versions of libraries used so that other users can replicate the conditions and implement the code.

Test :

- The selection test given for this project was to create a python implementation to dimensionally reduce the given data and make a 2D visualization with output as a log file.
- Before going through this paper, I decided to understand the challenge myself. The terms seemed new and hence I decided to ask my biology teacher and then also did research on the internet to understand what exactly I am supposed to do.

- I started implementing the solution on an IPython notebook.
- Then I wrote the code to create a log file that had the information that was asked to print out.
- This completed my selection test, this is the link to the notebook, it can be re-run on google colab, will function without any errors.
- **The final results of PCA and UMAP respectively are :**

Figure 1: PCA colored using population code. execution time = 10.8783 sec

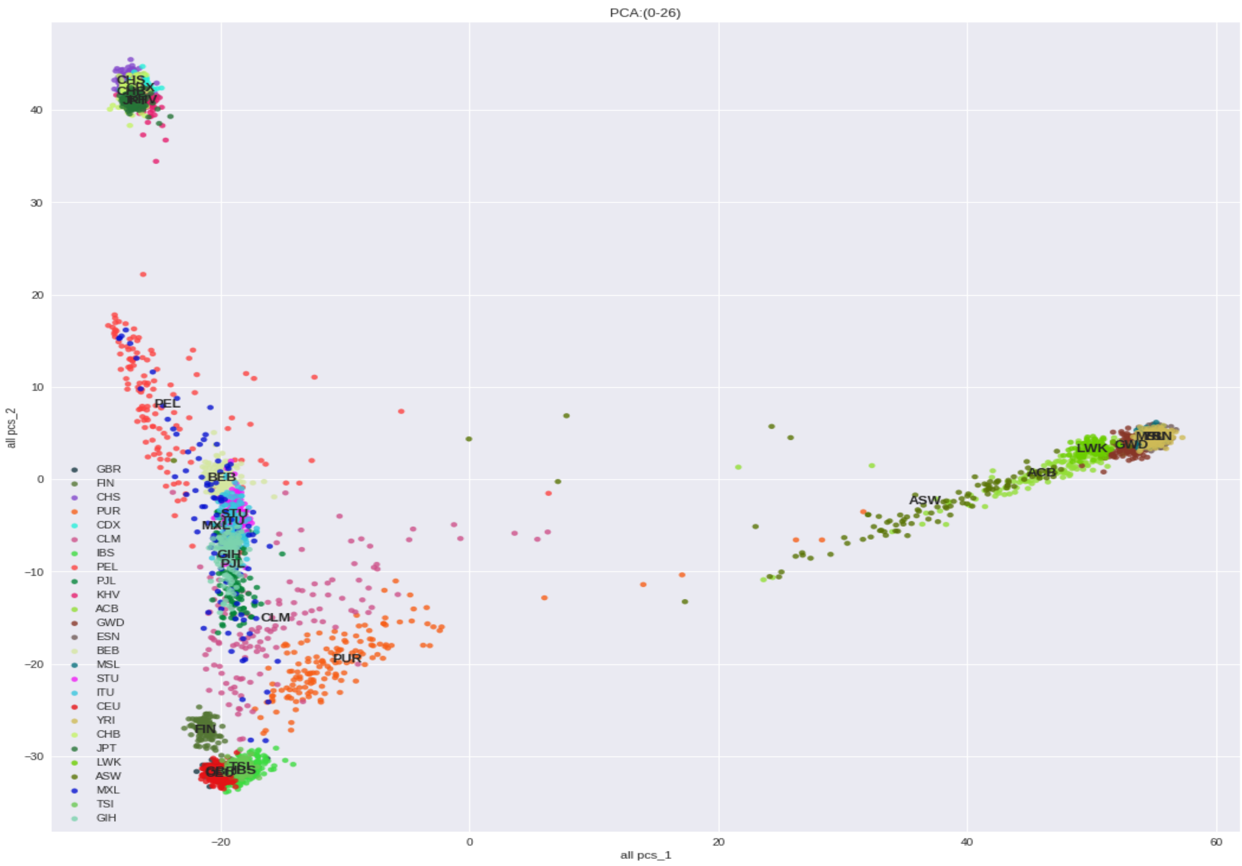
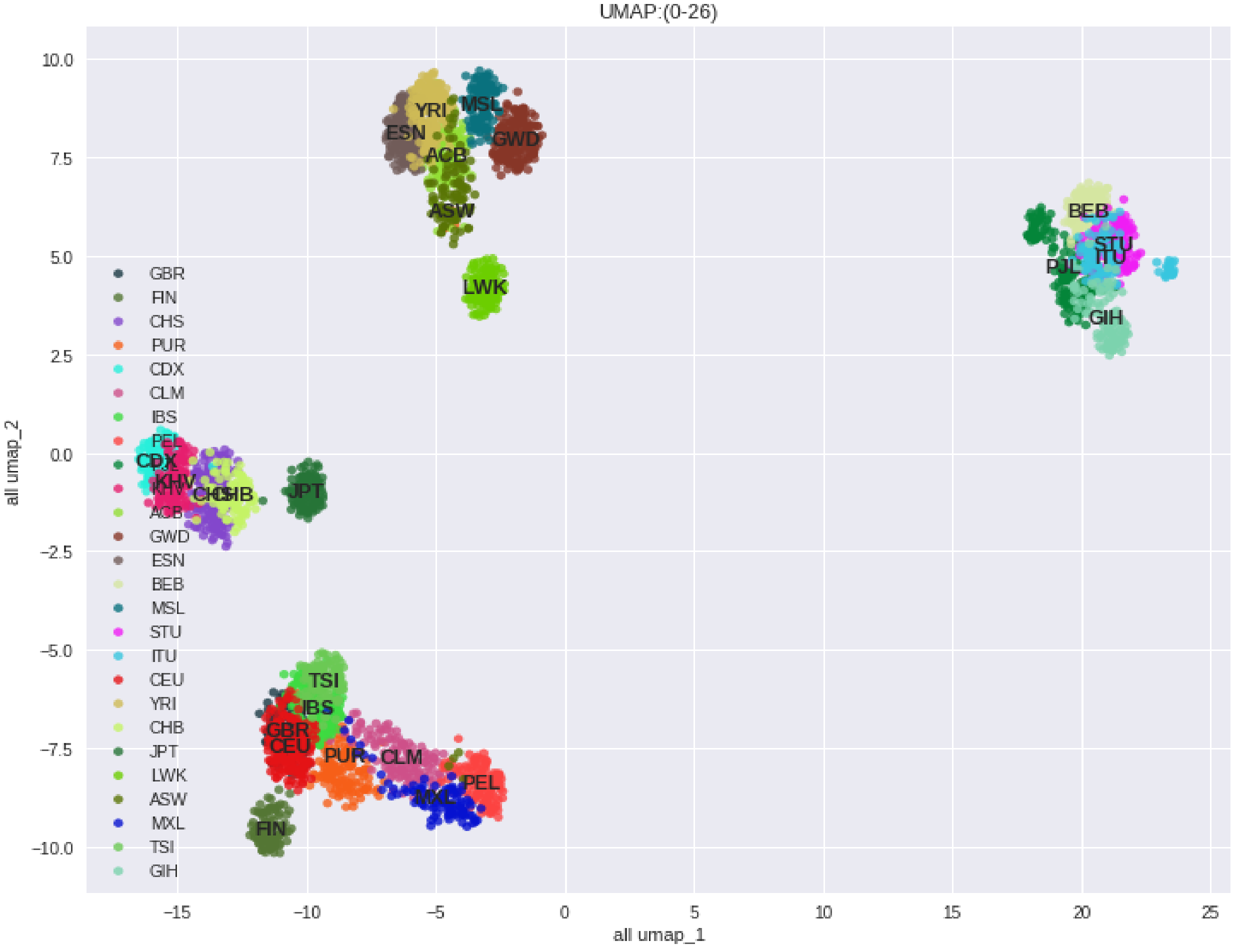


Figure 2: UMAP colored using population code. execution time = 0.0020 sec


UMAP:(0-26)
all umap_2
all umap_1
YRI
MSL
ESN
GWD
ACB
ASW
LWK
BEB
STU
ITU
PJL
GIH
CDX
KHV
CHS
CHB
JPT
TSI
IBS
GBR
CEU
PUR
CLM
PEL
MXL
FIN
● GBR
● FIN
● CHS
● PUR
● CDX
● CLM
● IBS
● PEL
● PJL
● KHV
● ACB
● GWD
● ESN
● BEB
● MSL
● STU
● ITU
● CEU
● YRI
● CHB
● JPT
● LWK
● ASW
● MXL
● TSI
● GIH