ACB Project:

LogBook:

Friday Evening:

- Looking at NGS and variant calling tools, selecting one installing what I should need to manage the project (first time using next generation sequencing algorithms -used mostly data from Ensembl during my project)

- Having a small macbookpro with a saturated memory, getting an amazon EC2 with 16 cores and 32gB of RAM.

- setting up everything and doing the first filtering of the dataset using vcftools

Saturday:

- Finishing understanding the vcf format and starting developing a pipeline to get the dataset from ensembl into a python project (the goal here is to have a fully functional python pipeline)
- preprocessing involved also completing some missing metadata about what is GT and PID, removing all non canonical chromosomes that finally do not exist in the filtered output
- finding a way to filter everything in parallel (LD pruning takes too long)

- Running the second filtering (linkage-disequilibrium pruning)
- using pyvcf and finding good ways to represent the dataset
- starting to run classifier on the data and getting already 90% average accuracy with only a small subsample of the dataset and a random value of PCA (500) and a kNcentroid calssifier on CV.

Sunday:

- finishing writing the class
- debugging the class on the EC2
- writing a small explanation of what I have done
- cleaning the python file
- creating a nice notebook
- adding more interesting functions to explore the dataset and more debugging
- adding more classifier and dim reduction algorithm (max accuracy now of 98.7%)

Monday:

- having exams so not too much time
- adding a saving and plotting function
- writing about what I have learnt
- testing them and sending the final project.

Doing the project:

The beginning of the project was a quite stressful moment, not knowing how long it would take or what problems I would encounter with the dataset. I had to learn quite a few things and in particular vcf data-format and how to manage it. Working with NGS data is not too complex but it took me a few hours to understand the caveats of the format and the tool (vcftools) using the literature and developer's blogs. Once this was done, together with a first classification trial, I set out to read a bit of the literature to gain insights about the best classifiers to label genetic ancestry from SNPs. training a K-nearest-neighbors algorithm turned out to yield a great accuracy from the get-go. after managing the many problems with the EC2 and the command line filtering from python I was able to have fun with many clustering algorithms and versions of PCA, looking to increase the accuracy.

The project

Inferring the genetic ancestry of individuals using a labeled training set and principal components analysis is something that any associate computational biologist at the Broad Institute should be familiar with before taking on more complex projects.

This project was done in the course of the second weekend of May 2018 by Jérémie KALFON, using python, vcftools, sci-kit learn and an Amazon compute server.
This project builds on much work and ideas. Its goal is to find the genetic ancestry of a given individual given a dataset of SNPs from NGS data. The subgoal is really to reduce the number of SNPs, selecting or transforming to get only a subset of highly predictive features that we will be able to feed to our classifier. Much has been tried such as :

- Selecting a subset of tag SNPs with the minimum size and highest informativeness value calculated from an informativeness measure (which evaluates how well a single SNP or a set of SNPs predict another single SNP or another set of SNPs within the neighborhoods).
- Selecting the informative SNPs with the maximum prediction accuracy, which is obtained from a prediction accuracy measure evaluating how well the value of an SNP is predicted by the values of only two closest tag SNPs.
- Selecting informative SNPs by removing redundant features. Redundancy was measured by feature similarity between two features, i.e., the linkage disequilibrium (LD)

Tools also exist such as:

- STRUCTURE which is using prior knowledge about the distribution of alleles and ancestry and use it in a Bayesian model implemented using MCMCs and Gibbs sampling.
- Frappe, ADMIXTURE EIGENSTRAT/ smartpca, ANCESTRYMAP and so on ...

Some of them are now able to infer local ancestry (on given parts of the genome) and might be a first step toward understanding the correlation of individual ancestry proportions with disease risk or treatment response at a genetic level.

In this project PCA, KPCA, SPCA, LDA have been tried and seem all to work well, selecting interesting features. LDA especially seems to be a great fit for reducing the dimension to a minimum (for SVM and Gaussian Mixtures). As usual, it has been found in the experiments that PCA and KNN are extremely robust. Even without filtering the SNPs or using only information about whether or not these are phased, we achieve accuracy no less than 88% with a sufficiently high number of features (>400). The importance of filtering the data is brought to light by the SVM algorithm is really bad when the data has not been linkage disequilibrium pruned (<20% ~ random). It could be interesting to try to LD prune the data more and see if we obtain better classification results. But SVM works extremely well with the low number of high-quality features (achieving 96% accuracy with 100 features) in addition, svm allows you to predict probabilities for each labels which is always interesting when dealing with such algorithms. Finally, a surprise, thinking it was not possible to reduce the features less than 20 I have found out that the prediction accuracy of Gaussian Mixture (in case of high quality features) is very high. Gaussian Classification does 98% accuracy using only features, put against what is used in the litterature and other programs (60 - 500) this is a great feat but is probably an overfitting due to the small amount of individuals and labels. Only could it be inferred with more individuals than what I have.

Many other things could be tried but due to the informality of the project, won't. Some ideas can be found in the ReadMe file

The beginning of the project was a quite stressful moment, not knowing how long it would take or what problems I would encounter with the dataset. I had to learn quite a few things and in particular vcf data-format and how to manage it. Working with NGS data is not too complex but it took me a few hours to understand the caveats of the format and the tool (vcftools) using the literature and developer's blogs. Once this was done, together with a first classification trial, I set out to read a bit of the literature to gain insights about the best classifiers to label genetic ancestry from SNPs. training a K-nearest-neighbors algorithm turned out to yield a great accuracy from the get-go. after managing the many problems with the EC2 and the command line filtering from python I was able to have fun with many clustering algorithms and versions of PCA, looking to increase the accuracy.

ressources

Yushi Liu, Toru Nyunoya, Shuguang Leng, Steven A Belinsky, Yohannes Tesfaigzi, Shannon Bruse, " Softwares and methods for estimating genetic ancestry in human populations", Human Genomics 2013, 7:1

Nina Zhou, Lipo Wang, "Effective selection of informative SNPs and classification on the HapMap genotype data", BMC Bioinformatics 2007, 8:484

[stackoverflow, docspython] for almost all my needs in development

[biostars, seqanswers, bioconductors, gatkforums, wikigenes, wikipedia, internationalgenome] for all the problem specific questions.

Used an AWS amazon EC2 compute Largex2 version deeplearning linux Debian

---

Inferring the genetic ancestry of individuals using a labeled training set and principal components analysis

in (PCA) each lower-dimensional component is orthogonal to (i.e., independent of) the others. PCA has been extensively used in the medical and population genetics community as a tool to visualize and summarize the genetic ancestry of individuals and is regularly used in our computational pipelines to infer and classify samples of unknown ancestry. In this use case, we map the high-dimensional space of genotypes in a call set into a lower-dimensional space of principal components. For a given individual with millions of genotypes, we can reduce the amount of data that goes into the ancestry classification problem into a handful of principal components; this data is then fed into a classification algorithm of our choosing that runs on labeled training data to generate a model that can be used to predict the classification of unlabeled samples.
Your assignment recapitulates this common analysis problem: directly inferring the ancestry of samples with missing ancestry labels. You are provided with a file containing the genotypes of each individual in a cohort of samples with mixed labeling status: some samples have known (labeled) ancestries, and others do not. Your task is to generate the following:

A set of principal component values for the call set (i.e., a table containing the PC values for each sample for each principal component)
A final classification or ancestry label assigned to each sample that is missing a label
A visualization of the distribution of PC values for each sample in the call set, along with the labeled and predicted ancestry classifications

A few general remarks:
You may use whatever computational tools you wish. To demonstrate competence in the lab's core languages, please submit your work in Python, UNIX/bash, R, and/or Matlab. Please document and comment all code, Make sure you explain your choice of classification algorithm and any parameters chosen (if applicable).
that went into the generation of the final deliverables. When you are finished with the project, please upload the results and the code you wrote to generate the results on our GitHub repo: https://github.com/macarthur-lab/hiring. When you submit your work, please push to a new branch of your naming, as the master branch is protected.

A few hints:
Some packages you may find useful in completing the assignment: vcftools, plink, eigensoft (smartpca)

Successful, clean PCA on human genetic data will require filtering data to high-quality variants that are linkage disequilibrium (LD)-pruned. In general, we like to run PCA on high-callrate, bi-allelic, common (allele frequency > 0.01) variants that are pruned to r^2<0.1; but you are welcome to run PCA on whichever set of variants you find work best for you. You will also need to normalize your genotypes after filtering to high-quality variants.