# Detection of epistasis patterns in genetic data

## Methods used for this project

In this project two methods were implemented to search for epistasy patterns in genetic data.

- SMMB-ACO for Multiple Markov Blankets by Ant Colony Optimization.
- VNS for Variable Neighborhood Search.

## Initialisation of project

In order to compile the whole project a makefile is provided at project's root. This makefile will call each method's makefile and produce both executables. This makefile, can recreate directory structure if needed.

## Preparation of prerequisite

In makefiles please change BOOST_FOLDER value with path of the installed boost library on current workstation.
Moreover, please check if current g++ version is compatible with C++11 functionalities (version >= 5.0). Please also check boost library's version. Authors cannot guaranty compatibility with other boost version even if boost version seems to be compatible between them. For this method version 1.61.0 was employed.

Some directories need to be created. Aslo few other thing like preparation of python environment are needed. A quick "install" can be done using:

```
./installation_of_project.sh
```

This script will create some needed directories, install conda environment and purge compiled file to recreate executables

## Compilation instruction

To recreate directory structure for both methods, please call this line at the root of the project:

```
make install
```

To compile both methods, please call this lines at the root of the project:

```
make
```

If a recompilation is needed please use to purge all compiled files:

```
make clean
```

# Generate a naive data set

This project is provided with a tool to generate some naive data set. This tool fit with a logistic regression and produce a data set of given sizes with a given number of causal SNPs. This operation is performed by simu_naive.py.

## Prerequisite, generation of right virtual environment

These steps are done if installation_of_project.sh was used.

Some packages are required to run this tool. List of them is provided with needed version in environment.yml or environment.txt file. In order to generate a virtual environment where simu_naive.py can be executed please follow next steps.

For this step conda is needed, please follow instructions from anaconda documentation to get it installed. If conda is installed please execute this line to generate an environment called projet_c:

```
conda env create -f ./environment_to_execute_python/environment.yml
```

If virtualenv if prefered and is not installed. Please use following lines:

```
pip install virtualenv
virtualenv -p /usr/bin/python3.6 projet_c
source projet_c/bin/activate
pip install -r environment.txt
```

```
deactivate
```

Following this, activate this environment and execute simu_naive with arguments:

```
source activate projet_c
./simu_naive.py -p simu_naive -f 1 -v 28 -pa 2000 -o toy_dataset -c 2000 -s 2
source deactivate
```

## Run simu_naive.py

It is possible to call this tool using a default call using

```
./launch_simu_naive_toy_example.py
```

In this launcher is also added explications about arguments used in call. It is also possible to display help using

```
simu_naive.py --help
```

## Pre-generated naive dataset

Project is provided with pre-generated naive dataset. They can be find in data_simulated_from_class. This directory gather some archives, one of them contain files that need to be parsed. In order to automatize those steps it is possible to run ./initialization_of_simulated_datas_from_repository.sh. This script can be run by the installation script.

# Parameters

It is possible to tweaks different parameters for both methods. To change parameters of SMMB ACO method use parameters.txt localized in smmb_aco/parameters. This file is commented allowing an easy setup for particular needs. If some tweaks are needed for VNS method, please modify vns/parameters/parameters.txt. Please note that a particular section has been included in associated project report about some parameters and their effects.

# Expected output

On each data set constituted by one genotype file and his associated phenotype file epistasis pattern is searched using both methods. Here is an example of expected output file :

```
# Result from vns
# Pattern || occurences || chi2-score || p-value || unreliable case
{M0P7,M0P8} || 3 || 87.8235 || 2e-16 || 0
{N9,M0P8} || 1 || 49.3762 || 5.96046e-08 || 0
# Execution time : 156 milliseconds
```

Each method can be evaluated using the eval_simu.py script, it uses the method as many times as it is asked on each data file in the directory given as an argument. A file summarizing the results is produced, it includes the counting of the type of results (true positive: TP, false positive: FP, false negative: FN) as well as the precision calculations, f-measurement, etc.

## Run eval_simu.py

This script allows to execute a given number of times the methods on files contained in a given folder. In order to execute this script with a default configuration please call it with following line:

```
./launch_evaluation.py
```

This launcher will perform each methods a given number of times on some files generated using gametes. If a particular configuration is needed, please check call used in launcher or display help using:

```
./evaluation/eval_simu.py --help
```

It will produce a file of this type :

```
Filename,TP,FP,FN,recall,precision,f_measure,power,average time per run
Naif_21_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.6516388654708862
Naif_71_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.676003575325012
Naif_29_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.807805299758911
Naif_55_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.7822365760803223
Naif_70_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.8581265211105347
Naif_47_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.7423101663589478
Naif_48_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.5195796489715576
```

```
Naif_18_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.580695629119873
Naif_83_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.2409913539886475
Naif_38_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.10369074344635
Naif_7_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.7983251810073853
Naif_90_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.890117645263672
Naif_95_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.696593999862671
Naif_54_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,3.9864935874938965
Naif_69_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,5.381306171417236
Naif_43_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.6655861139297485
Naif_10_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.1970778703689575
Naif_33_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.428283214569092
Naif_41_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.373726487159729
Naif_64_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.405292868614197
Naif_50_Genotype.txt,2,0,0,1.0,1.0,0.5,1.0,4.1667115688323975
### Total evaluation time : 169.9076943397522
```

# Packages and library used in this project

- BOOST a peer reviewed c++ collection of library.
- NumpY a python library to handle with matrix.
- Anaconda a software to handle with python virtual environment.

# Authors

Tanguy Lallemand -
Jonathan Cruard