**Documentation for Automatic FF Training &Prediction**

**Step 1: Keep all Your sam File inside sam Folder**

(“ff/sam/”

Run Code bam\_rl\_read.py

(This Code will Automatically Convert sam File to srl and rgc and move Converted File to separate Folders)

Note: Converting sam file may took long time according to Your PC Speed

RL files will Moved inside SRL Folder and RGC Files will Moved to RGC Folder

**Let’s Train Our Model**

Users Must Put Separate Bin info files in RL and Read folders with their Corresponding files srl and rgc

If files name is in Bin info and file not in RGC/SRL Folder program cant run

After all Condition Before Running Code Python dependency pandas , Numpy , sicit need to be install

And also R must be installed and RScript must be in Path of Your System

Also R dependencies/libraries glmnet, Matrix, MASS, doParallel , methods

################################################################

Pre - Cautions:

Folder Name, Path, bininfo file format must be the same

(Folder Path is ./ 3 folder samfile testing training and python source code)

Inside samfile samfiles must keep

Inside training there are two folders Read and RL

Read folder keeps all read files .csv format read\_bininfo file and R dependency codes loessgctrain.R and glmnet.R

Inside RL folder also there must be rl\_bininfofile…csv all files of srl format and R dependency code glmnet.R

After Finishing all Training task, the parameter file should create inside each RL and Read folder and also copied inside training and outside Read and RL folder.

While testing (Predicting) the same process apply for bininfo file.

There must be read and rl bininfo files inside testing folders and all files must be there in same Folder and there also must be R dependency File

Loessgctest.R rlnormtest calenetrl.R and calenetread.R files

To run simply You need to open terminal in root file location (i.e. python code existing file)

**Type python python\_file\_name.py**

Python GenomomFF\_training.py

Python Genomom\_testing.py

**Explanation of each Functions inside Training and testing Codes**

**GenomomFF-training.py**

**read\_bin():**

read\_bin() is the first step and first function. This Function will check bininfo file for each Read and RL folder and also assign Global value for global variable.

**Rl\_normtrain():**

This Function will read bininfo file assigned in step 1 (read\_bin() function).

And also create 2 Out put files y and x. (ffy……) and (t\_rl)

**Arrangetrain\_rl()** : Arrangetrain\_rl() will receive y and x (ffy\_rl and t\_rl )and also make twoset of training y and x (rf) .

**Enet\_rl() :**

Enet\_rl() output Parameter value for testing.

**Loess\_gc\_train\_read():**

Same as function 2 but files are read files. Here Rscript code loessgctrain.R run with file parameter read\_bin\_info , ffy\_read and t\_read (ffy\_read and t\_read is the output files)

**Arrangetrain\_read():**

When Loess\_gc\_train\_read() output 2 files ffy\_.. and t\_read.. then these files prepared in proper format and output temptrain\_read\_y…. and temptrain\_read\_rf…. Files.

**Enet\_read():**When step 6 prepared train y and train x (rf) files then regression parameter values are created using R glmnet (E-Net-regression)

**GenomomFF\_testing.py**

Read\_bin() is same as training . Only little variable name may change.

**Get\_parameter\_file():**

to copy read\_parameter and rl\_parameter files which are created in training process and are in inside training folder after running GenomomFF\_training.py

**Loesstest\_read() :**

in this Function R script run which will calculate ffy\_read and t\_read using Loess regression .

**Arrangetest\_read():**

files Y and X file created in step 3 will be prepared for proper format to run prediction (cal\_enet\_read())

**Rlnormtest():**

In this Function Rscript rlnormtest.R executed which accept rlbininfo and output ffy\_rl… and t\_rl … files .

**Arrangetest\_rl():**

Accept ffy\_rl .. and t\_rl as input file and output temptrain\_rl\_rf and temptrain\_rl\_y for predicting Purpose.

**Cal\_enet\_rl() :**

Prediction rl using e\_net regression and parameter values calculated in training period.

**Cal\_Cor():**

This is last step in where Correlation is calculated by comparing predicted and actual Files.

**Step in Whole Process**

**Training**

|  |  |  |
| --- | --- | --- |
|  | **Read** | **RL** |
| 1 | **Loessgctrain🡪 read\_bininfo\_... .csv**  Out ffy\_read…….csv ,t\_read……..csv  (R script loessgctrain.R execute) | Rl\_normtrain()🡪  In🡪 bininfofile\_  Out 🡪 ffy\_rl……..csv, t\_rl….csv |
| 2 | **Arrangetrain\_read()🡪**  **In 🡪ffy\_read……..csv,t\_read……..csv**  **Out 🡪** temptrain\_read\_y..csv  and temp\_read\_rf…csv | **Arrangetrain\_rl()🡪**  **In 🡪 ffy\_rl …csv ,t\_rl ..csv**  **Out🡪temptrain\_rl\_rf.. temp\_train\_rl\_y** |
| 3 | **Enet\_read() :- IN** temptrain\_read\_y..csv, temp\_read\_rf…csv  **Out 🡪** temp\_train\_par\_read….csv | **Enet\_rl()->in**  **temptrain\_rl\_rf…… temp\_train\_rl\_y**  Out temp\_train\_par\_rl….. |

**Testing**

|  |  |  |
| --- | --- | --- |
|  | **RGC** | **SRL** |
| 1 | **Loessgctest\_read()**  **In 🡪** read\_bin\_info..  **Out🡪** ffy\_read\*\*\*\*\*.csv,  T\_read\*\*\*\*.csv  **(R code loessgctest.R** run by Python | **Srlnormtest()**  **In🡪** rl\_bin\_info………csv  **Out 🡪** ffy\_rl……csv t\_rl…..csv  **(R code run by Python)** |
| 2. | **Arrangetest\_read()**  **In🡪** ffy\_read\*\*\*\*\*.csv,  T\_read\*\*\*\*.csv  **Out🡪** temptrain\_read\_y and temptrain\_rf | Arrangetest\_rl.py  In 🡪 ffy\_rl……csv t\_rl…..csv  **Out** temptrain\_rl\_y….csv & temptrain\_rl\_rf….csv |
|  | **Cal\_enet\_read()**  In **🡪 temptrainpar**(From Training)  temptrain\_read\_y and temptrain\_rf    **Out 🡪** enet\_read.csv | **Cal\_enet\_rl()**  Calenet\_enetsrl.R  In 🡪 temptrainpar……csv (From Training Process)  temptrain\_rl\_y….csv & temptrain\_rl\_rf….csv Out 🡪 enet\_rl.csv()  **Run R code cal\_enetsrl.R** |
|  | **Cal\_Cor():**  **Input enet\_read.csv and enet\_rl.csv**  Out Correlation\_result.csv and also print value |  |