**Step 1  Merge SV data samples in 1 master file**

**Command line code:**

**python Merge\_sv\_Germline.py –outf NameOutputFile.csv**

**or**

**python Merge\_sv\_Somatic.py –outf NameOutputFile.csv**

In the script make sure to change line 12 to the right directory:

Example for gbm file:

for root, dirs, files in os.walk('/icgc/dkfzlsdf/project/pedbrain/gbm/sequencing/whole\_genome\_sequencing/view-by-pid'):

**Or run**

**qsub Merge\_sv\_Germline.sh**

**or**

**qsub Merge\_sv\_Somatic.sh**

In the script make sure to set the right output file name

**Step 2  Main pipeline**

Example for gbm files:

**qsub SV\_gbm\_germline.sh**

**Example of .sh file with only Gene1\_exon as input:**

* In the files in my folders, Gene2\_exon, Gene1\_intron and Gene2\_intron are added as well.
* The easiest way to adapt the file is to change the folder where the output files are stored (ctrl+H; gbm\_output  NameOutputFolder).
* In red, green and blue, the output files of all the steps are shown which makes it clear in which step which file is needed. You don’t have to change the names if you’re fine with it.
* Check whether the right burden outputfile/file path is used
* Check whether the right input file (--inf) is used in the scipts 1\_Get\_Gene\_Names\*.py  These files are generated in the first step, so the master file **NameOutputFile.csv**
* Highlighted in yellow the final output files
* The first and third script are run twice, this is because we can generate 2 different Update\_Burden\_result\* output files (with or without sampleID). The sampleID is added in script \*2.py.

**python 1\_Get\_Gene\_Names.py**

--inf=SV\_gbm.csv

--comp=burden.out\_cases\_snpfile\_0.001\_HGGONLY\_CADD15.txt

--outf=**gbm\_output**/Genes\_Burden2nd\_Gene1\_exon\_germline.csv

--outfB=**gbm\_output**/Genes\_Burden\_Gene1\_exon\_germline.csv

--outfSV=**gbm\_output**/SV\_Gene1\_exon\_germline.csv

--input=Gene1\_exon;

**python 1\_Get\_Gene\_Names2.py**

--inf=SV\_gbm.csv

--comp=burden.out\_cases\_snpfile\_0.001\_HGGONLY\_CADD15.txt

--outf=**gbm\_output**/Genes\_Burden2nd\_Gene1\_exon\_germline2.csv

--outfB=**gbm\_output**/Genes\_Burden\_Gene1\_exon\_germline2.csv

--outfSV=**gbm\_output**/SV\_Gene1\_exon\_germline2.csv

--input=Gene1\_exon;

**python 2\_Get\_results.py**

--inf=**gbm\_output**/SV\_Gene1\_exon\_germline.csv

--comp=**gbm\_output**/Genes\_Burden2nd\_Gene1\_exon\_germline.csv

--outf=**gbm\_output**/Burden2nd\_Gene1\_exon\_germline.csv

--input=Gene1\_exon;

**python 3\_Update\_burden\_file.py**

--inf=burden.out\_cases\_snpfile\_0.001\_HGGONLY\_CADD15.txt

--comp=**gbm\_output**/Genes\_Burden\_Gene1\_exon\_germline.csv

--outf=**gbm\_output**/Update\_Burden\_result\_Gene1\_exon\_germline.csv

--input=Gene1\_exon;

**python 3\_Update\_burden\_file2.py**

--inf=burden.out\_cases\_snpfile\_0.001\_HGGONLY\_CADD15.txt

--comp=**gbm\_output**/Genes\_Burden\_Gene1\_exon\_germline2.csv

--outf=**gbm\_output**/Update\_Burden\_result\_Gene1\_exon\_germline2.csv

--input=Gene1\_exon;

**Step 1**

**python3 1\_Get\_Gene\_Names.py --inf sv1.csv --comp Burden\_results\_genes.csv --outf Intermediate\_files/Genes\_Burden2nd\_Gene1\_exon.csv --outfB Intermediate\_files/Genes\_Burden\_Gene1\_exon.csv --outfSV Intermediate\_files/SV\_Gene1\_exon.csv --input Gene1\_exon**

1\_Get\_Gene\_Names.py

--inf sv1.csv

Input file containing all the SV results from all the samples

--comp Burden\_results\_genes.csv

The genes which are the output of the burden test

--outf Intermediate\_files/Genes\_Burden2nd\_Gene1\_exon.csv

Output file with all the genes which are in the result of the burden test and the 2nd hit analysis with 2nd hit format

--outfB Intermediate\_files/Genes\_Burden\_Gene1\_exon.csv

Output file with all the genes which are in the result of the burden test and the 2nd hit analysis with Burden test format

--outfSV Intermediate\_files/SV\_Gene1\_exon.csv

Output file with sv1.csv file format containing only the genes of interest (see input option)

--input Gene1\_exon

On which of genes the analysis should be performed on (Gene1\_exon, Gene1\_intron, Gene2\_exon, Gene2\_intron)

**Step 2**

**python3 2\_Get\_results.py --inf Intermediate\_files/SV\_Gene1\_exon.csv --comp Intermediate\_files/Genes\_Burden2nd\_Gene1\_exon.csv --outf Result/Burden2nd\_Gene1\_exon\_result.csv --input Gene1\_exon**

2\_Get\_results.py

--inf Intermediate\_files/SV\_Gene1\_exon.csv

Input file derived from Sep 1 (Step 1, --outfSV)

--comp Intermediate\_files/Genes\_Burden2nd\_Gene1\_exon.csv

Input file derived from Sep 1 (Step 1, --outf)

--outf Result/Burden2nd\_Gene1\_exon\_result.csv

Output file with sv1.csv file format containing only the genes of interest (in Burden and 2nd Hit result)

--input Gene1\_exon

On which of genes the analysis should be performed on (Gene1\_exon, Gene1\_intron, Gene2\_exon, Gene2\_intron)

**Step3**

**python3 3\_Update\_burden\_file.py --inf Burden\_results\_genes.csv --comp Intermediate\_files/Genes\_Burden\_Gene1\_exon.csv --outf Result/Update\_Burden\_result\_Gene1\_exon.csv --input Gene1\_exon**

3\_Update\_burden\_file.py

--inf Burden\_results\_genes.csv

The genes which are the output of the burden test

--comp Intermediate\_files/Genes\_Burden\_Gene1\_exon.csv

Input file derived from Sep 1 (Step 1, --outfB)

--outf Result/Update\_Burden\_result\_Gene1\_exon

output file which can be used to update the burden test result file

--input Gene1\_exon

On which of genes the analysis should be performed on (Gene1\_exon, Gene1\_intron, Gene2\_exon, Gene2\_intron)