STALICLA R&D

Overview of the MPC and PolyPhen 2 annotation protocol

Gabriela Martínez

September 2019

The MPC-PolyPhen2 protocol

A command line interface program built in Python 3 to annotate MPC values and PolyPhen2 prediction labels to a set of patients and mutations. Given the initial dataset, this protocol will create the following columns:

- **MPC:** for missense variant mutations.
- **PolyPhen2 prediction label:** available options are **benign, possibly damaging, and probably damaging.**
- **PolyPhen2 prediction value:** probability associated to the PolyPhen2 prediction label.
- **Adjusted consequence:** further classification of consequences to find Missense, Missense3 and PTV mutations.

How to execute this protocol?

This module is written in Python 3 and uses up-to-date libraries for this version as well.

To run the main module in a Linux environment, simply open a new terminal and call the script together with the arguments it accepts:

```
python3 main.py
positional arguments:
  [inputFile] [inputMPC]
optional arguments:
  [-g GENE, --gene GENE, --genes GENE]
  [-id PATIENT, --patient PATIENT, --patients PATIENT]
  [-csq CSQ, --consequence CSQ, --consequences CSQ]
  [-mpc MPC, --mpc_gt MPC]
  [-pph2 PPH2, --polyphen PPH2, --polyphen2 PPH2]
  [-adj_csq ADJ-CSQ, --adjusted_consequence ADJ-CSQ, --adjusted_consequences ADJ-CSQ]
  [--path PATH-RESULTS]
```

MPC-PolyPhen2 protocol arguments

Mandatory parameters

- Input file containing patients and mutations.
- Folder containing pathways and their genes.
- Input file containing the official MPC mapped values.

Optional parameters: act as filters

- Genes.
- Patients.
- Consequences.

→ The protocol accepts filtering by these parameters before annotating MPC and pph2 information

- MPC.
- PolyPhen label.
- Adjusted consequences.

→ The protocol accepts filtering by these parameters after annotating MPC and pph2 information

Note: help() is available for all the parameters via the command line.

MPC-PolyPhen2 mandatory parameters: input file

The input file containing patients and mutations has the following columns*. Only observations with a valid **HGNC_symbol** will be considered.

Chr	Position	…	child_id	consequence	HGNC_symbol	constraint_score	…	pLI
11	881802		Patient_10	missense_variant	V595I	0.018		0.14
12	686702		Patient_19	Near_3splice	NA	0.651		0.21
3	981711		Patient_21	Intron	NA	0.006		0.35
15	1121834		Patient_31	Silent	R589R	0.124		0.05
20	541255		Patient_98	missense_variant	L239P	0.032		0.12

*and many more. This is not exhaustive.

MPC-PolyPhen2 mandatory parameters: MPC official values file

The input file containing official MPC values for all possible mutations. Columns of interest are:

- Chromosome
- Position
- Reference
- Alteration
- Consequence
- PolyPhen
- MPC

A preview of the file used

fordist_constraint_official_mpc_values_v2.txt

```
chrom,pos,ref,alt,Consequence,PolyPhen,MPC
1,69094,G,A,missense_variant,possibly_damaging(0.828),2.7340307082
1,69094,G,T,missense_variant,possibly_damaging(0.497),2.29135628242
1,69094,G,C,missense_variant,possibly_damaging(0.497),2.29135628242
1,69095,T,A,missense_variant,probably_damaging(0.99),4.31666214769
1,69095,T,C,missense_variant,probably_damaging(0.964),3.3059382823
1,69095,T,G,missense_variant,probably_damaging(0.99),3.50374879105
1,69097,A,T,missense_variant,benign(0.014),1.52357054751
1,69097,A,C,missense_variant,benign(0.03),2.071766814
1,69097,A,G,missense_variant,benign(0.031),1.6646576861000002
1,69098,C,A,missense_variant,possibly_damaging(0.463),2.4503748086000003
1,69098,C,T,missense_variant,benign(0.338),2.2094521780400003
1,69098,C,G,missense_variant,benign(0.014),1.52357054751
1,69100,G,A,missense_variant,possibly_damaging(0.556),2.5476542767400003
1,69100,G,C,missense_variant,possibly_damaging(0.556),2.3721794755400003
1,69101,A,T,missense_variant,probably_damaging(0.954),3.21115196292
1,69101,A,C,missense_variant,possibly_damaging(0.556),2.4715641076700003
```

*and many more. This is not exhaustive.

MPC-PolyPhen2 optional parameters

- ## Genes

Optional argument to filter specific genes. If no setting is provided, all available genes will be considered.

- Example: protocol will filter data for gene CTR9

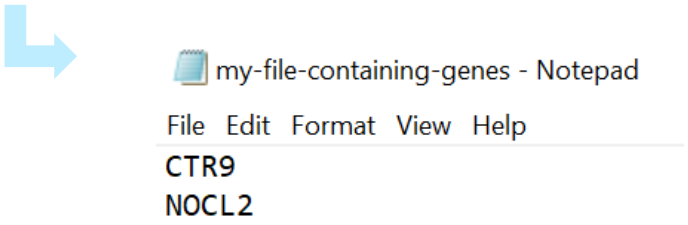
```
$ python3 main.py  ~/data/raw/mutations-file.txt
                   ~/data/raw/ mpc-official-values-file.txt
                   -g CTR9
```

*and many more. This is not exhaustive.

To filter by more than one gene, you can specify one of the following:

- Subset of genes separated by comma (without spaces):
  -g CTR9,NOCL2

- A **txt tab delimited file** with no headers and the desired genes to filter:
  -g ~/data/raw/filters/my-file-containing-genes.txt



- ## Patients

Optional argument to filter specific patient IDs. If no setting is provided, all available patients will be considered.

- Example: protocol will filter data for patient with ID 1

```
$ python3 main.py ~/PBPM/data/raw/mutations-file.txt
                  ~/PBPM/data/raw/pathways/
                  -id Patient_1
```

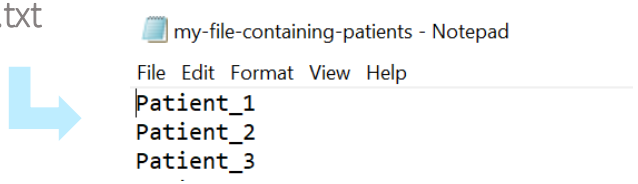
To filter by more than one patient, you can specify one of the following:

- A subset of patients IDs separated by comma (without spaces):

```
-id Patient_X,Patient_Y
```

- A **txt tab delimited file** with no headers and the desired patients to filter:

```
-id ~/PBPM/data/raw/filters/my-file-containing-patients.txt
```



- ## Consequences

Optional argument to filter specific consequences. If no setting is provided, all available consequences will be considered.

- Example: protocol will filter data for consequences of type 'missense_variant'

```
$ python3 main.py ~/PBPM/data/raw/mutations-file.txt
                  ~/PBPM/data/raw/pathways/
                  -csq missense_variant
```

To filter by more than one consequence, you can specify one of the following:

- A subset of mutations separated by comma (without spaces):

```
-csq missense_variant,Intron
```

- A **txt tab delimited file** with no headers and the desired consequences to filter.

```
-csq ~/data/raw/filters/my-file-containing-mutations.txt
```

- PolyPhen predictions

Optional argument to filter records with specific PolyPhen predictions. Available options for this parameter are: benign, possibly damaging, and probably damaging. Example:

```
$ python3 main.py ~/data/raw/mutations-file.txt
                  ~/data/raw/pathways/
                  -pph2 "possibly damaging"
```

Note that qualifiers must be enclosed with quotes as they hold blank spaces.

To filter by more than one qualifier, you must separate their labels with comma and without spaces while keeping the quotes:

```
-pph2 "probably damaging,possibly damaging"
```

- MPC

Optional argument to filter records with values greater than or equal to a specified MPC threshold. Example:

```
$ python3 main.py ~/data/raw/mutations-file.txt
                  ~/data/raw/pathways/
                  -mpc 0.7
```

- Path to store final annotations file

Optional argument to specify the path where the user wants to store the final MPC annotated patients set. Example:

```
$ python3 main.py ~/data/raw/mutations-file.txt
                  ~/data/raw/pathways//
                  --path location-where-I-want-to-save-annotated-patients/
```

When not specified, the final files will be stored in the **analysis** folder of the project.

What is the project organization?

It has the Stalicla Cookiecutter template for GitHub projects:

Available for the entire project settings

Execution logs are stored at this level and also PBPMs when an alternative path is not specified

Intermediate matrices are stored at this level

Specific libraries for the project can be found here

Where this presentation is

Source code for the project can be found here: PBPM.py and main.py

Where is this protocol stored?

Up to now, the MPC-PolyPhen2 repository is at my private GitHub:
https://github.com/mgmartinezl/Stalicla-MPC-PolyPhen

PENDING task: upload to the cluster and schedule some tests with final users.