User Manual SHERIFS

Thomas Chartier, Oona Scotti, Hélène Lyon-Caen Version 1.3

Last Updated January 2021

Contents

1 Introduction

The SHERIFS program is a code first developed in the framework of the PhD thesis of Thomas Chartier under the supervision of Oona Scotti (IRSN) and Hélène lyon-Caen (ENS). Since version 1.3, developments of the code have been realized at GEM. Seismic Hazard and Earthquake Rates In Fault Systems (SHERIFS) is a computer code written in python that allows computing earthquake rates on faults given a geometry of a fault system and of the background, a list of potential earthquake ruptures and specified rules to set the moment rate target for the fault system. The underlying approach used in the code is to first estimate the moment rate available for each fault and then to apply a set of rules that allow the slip rate of each fault to be consumed in either single or multi-fault rupture scenarios (FtF) allowed in the model depending on the picked magnitude until their slip-rate budget is exhausted.

Updates to this manual are available at: https://github.com/tomchartier/SHERIFS/

2 Previous versions

The first version of the code was distributed to the participants of the SHERIFS training (IRSN, Paris, France in December 2017). The present Version 1.1 of SHERIFS is published under a GNU Affero Global Public license together with a publication explaining the main features of SHERIFS (Chartier et al, submitted). The initial version of the code was used to model the fault system of the West Corinth rift published in NHESS special issue [1] (Chartier et al 2017). The code has evolved since. The following are the main aspects that have been implemented after the publication:

- The code now allows considering background seismicity. The user needs to define a zone surrounding the fault system.
- There is a new option in the code that allows correlating the sampling of the slip-rate on neighboring faults. In this way the slip rate budget of two faults that break together in many rupture scenarios will be sampled in the same way. If one of these faults is sampled in the upper part of the slip-rate distribution, it is admissible that the neighboring fault that breaks very often with this fault is similarly

in its upper part of the distribution. This option can be turned on or off.

- The code allows to define magnitude-frequency distribution MFD that deviate from the classical Gutenberg-Richter assumption (e.g. characteristic distribution, Youngs and Coppersmith 1985)
- A modified version of the Youngs and Coppersmith equation has been implemented.

New in version 1.1 (since 2019 article):

- More text files are generated during the Visualization to allow the users to generate more easily their own figures.
- For advanced users, in Sampling_analysis.py, comparison between the model and the data are automatically generated. See code for more details.

New in version 1.3 (2020 developments):

- Greatly improved computational speeds by saving intermediate results that are shared by different branches of the logic tree.
- SHERIFS now supports Geojson as input which allows for compatibility with the Global fault database and the output results coming from Oiler (GEM products).
- An automatic calculation of the possible rupture in a fault system according to the distance between faults.
- A general clean up and reorganisation of the libraries.
- The graphical user interface is no longer supported, running using Spyder is still possible but the command line is now preferred (note from January 2021 - beta testing is need for inter-platform compatibility)

3 Installing SHERIFS Version 1.3

TODO - clean up this part ! for now, we suggest to download from Github. Basemap is still a required library for the figures. In 2021, SHERIFS will start using GMT instead.

The current version of SHERIFS needs to be run using python 3.6.

For user not familiar with the command line , we suggest installing **Anaconda** that includes a lot of the dependencies required by SHERIFS. Installing Anaconda also installs spyder which is the easiest way to run SHERIFS. You will also need to install basemap, the library for plotting maps. Documentation is available online but if you are using anaconda you can run the following line in a terminal (warning! About 200Mo to download):

conda install -c conda-forge basemap

Then run this line to have high resolution coast lines:

conda install -c conda-forge basemap-data-hires

If basemap is still not loading, you might have to close and reopen your python console or spyder. Before running SHERIFS, you should run **test_SHERIFS.py**. This code ensures that you have all the python libraries necessary for running SHERIFS. If you have everything necessary for running SHERIFS, running test_SHERIFS.py will display a window confirming that all python libraries are correctly installed. The window might appear in the back of other opened windows, so it is worth looking around. If this is the case, the windows opened by SHERIFS might also appear in the back.

- If you have a python error saying a library is missing, please install the library using conda install or pip, if you are not using anaconda (documentation on installing libraries is available online with a quick google search – python install *name_of_the_library* -). Tips for non-frequent python users: - If the code is crashing, it will display exactly where it stopped, in spyder, you can click on the line number of the SHERIFS code where it crashed. If you use print(*name_of_variable*) the line before it crashed, you can have an idea of where the problem is coming from. Most crashes are due to problems in the input files (format not respected, wrong name of a fault...). But if it is a python issue, most of the big issues have already been encountered and solved by other people online. A copy and paste of the error in google will likely lead you to the answer to your problem.

- for any questions or bugs you cannot solve, please use the google group : https://groups.google.com/forum/#!forum/sherifs . If not already asked and answered, your question will be answered shortly.

4 Running SHERIFS Version 1.3

The SHERIFS code allows end-users to build the fault model with complex multifault ruptures. The files structure, the logic tree structure and the input files can be easily modified for a re-run if different parameters need to be tested. The code is written so as to build Openquake-compatible input files (OpenQuake V3.10). The user of SHERIFS should consult Openquake for further details about the hazard computation.

Flow chart

(optional : 0_build_ruptures.py)

1_SHERIFS.py

2_ Visualization.py

3_Weighting.py

The required input files and formats are listed below.

4.1 The SHERIFS input parameter file

4.2 Input file formats - using txt files

4.2.1 Faults_geometry.txt

This file contains the trace of the faults for the simple faults and the position of each point for complex faults.

The first line contains the column labels.

model_name	fault_name	longitude	latitude	type
Example_Model	F1	21.8461481	38.32598913	sf
Example_Model	F1	21.84745563	38.32597968	sf
Example_Model	F1	21.8494889	38.32590774	sf
Example_Model	F1	21.85130491	38.32589457	sf
Example_Model	F1	21.85341081	38.32582203	sf
Example_Model	F1	21.85508153	38.32580986	sf
Example_Model	F1	21.85711547	38.32579501	sf
Example_Model	F1	21.8590754	38.32566621	sf
Example_Model	F1	21.86096404	38.32565235	sf
Example_Model	F1	21.86372436	38.32563205	sf
Example_Model	F1	21.86495924	38.32562295	sf
Example_Model	F1	21.86684855	38.32566622	sf
Example_Model	F1	21.86815744	38.32577099	sf
Example_Model	F1	21.86997412	38.32581475	sf

list of the column:

model name = name of the model the fault belongs to.

fault name = name of the fault (one fault name should not contain the name of another fault).

longitude and latitude = The points of the faults are listed in rows, they need to be ordered or the fault will have loops.

type ('sf' or depth of the point) For describing a simple fault geometry (see OpenQuake definitions) input 'sf'; for a complex fault geometry, the user inputs the depth of the point - the complex fault geometry is described by at least by two edges (list of points) of uniform depth (top and bottom). Additional edges of intermediate depth can be added for more detailed geometries.

4.2.2 Faults_properties.txt

model_name	fault_name	dip	oriented	mechanism	upper_sismo_depth	lower_sismo_depth	slip_rate_min	slip_rate_moy	slip_rate_max	Domain	shear_modulus
Example_Model	F1	60	N	N	0	6	4.8	5.	5.2	Active_Shallow_Crust	30
Example_Model	F2	55	N	N	0	7	3.	3.2	3.4	Active_Shallow_Crust	30
Example_Model	F3	60	N	N	0	7	3.8	4	4.2	Active_Shallow_Crust	30
Example_Model	F4	60	N	N	0	7	0.7	0.9	1.1	Active_Shallow_Crust	30
Example_Model	F5	60	N	N	0	7	3.3	3.5	3.7	Active_Shallow_Crust	30

This file contains the geometry and kinematics of the faults. All the parameters required by Openquake are requested.

The first line contains the column labels. Each row is a fault in a model.

Description of the columns:

model name = name of the model the fault belongs to **fault name = name** of the fault (one fault name should not contain the name of another fault)

dip = dip of the fault

oriented = orientation of the dip (important for the "right hand rule" of OpenQuake)

mechanism = fault mechanism (N, S, R or value of the rake)

upper_sismo_depth = upper limit of the fault, (km) following the OpenQuake definition

lower_sismo_depth = lower limit of the fault, (km) following the OpenQuake definition

slip_rate_min = lower limit of the slip-rate distribution **slip_rate_moy** = mean value of the distribution **slip_rate_max** = higher value of the distribution slip-rate is picked in a uniform distribution. (sample 1 is always the mean value)

Domain = seismotectonic model used as a key in OpenQuake to attribute to each seismogenic source the correct GMPE.

Shear modulus = Shear modulus applied to this fault (in GPa, typical value 30 GPa)

4.2.3 Background_geometry.txt

```
Model_used  lon lat
Example_Model   21.77   38.450
Example_Model   22.018  38.400
Example_Model   22.330  38.300
Example_Model   22.330  38.163
Example_Model   22.018  38.210
Example_Model   21.77   38.311
```

This file contains the description of the geometry of the background zone.

4.3 Input file formats - using geojson files

4.3.1 Fault info : in a geojson

The geometry is described in a LineString describing the trace at the surface. SHERIFS will describe a kite fault surface between the min and max seismogenic depth.

Description of the parameters:

si = name of the section, can be a string of an integer

model = name of the model the fault belongs to **fault name = name** of the fault (one fault name should not contain the name of another fault)

dip = dip of the fault

oriented = orientation of the dip (important for the "right hand rule" of OpenQuake)

rake = value of the rake for this section

up_s_d = upper limit of the fault, (km) following the OpenQuake definition

lo_s_d = lower limit of the fault, (km) following the OpenQuake definition

sr_min = lower limit of the slip-rate distribution **sr_mean** = mean value of the distribution **sr_max** = higher value of the distribution slip-rate is picked in two uniform distribution between the min and mean, and mean and max. (sample 1 is always the mean value)

Domain = seismotectonic model used as a key in OpenQuake to attribute to each seismogenic source the correct GMPE.

shear_modulus = Shear modulus applied to this fault (in GPa, typical value 30 GPa)

Two parameters nb_rup and max_rup_length are also included in the file by the reprocessing tools but optional and are not directly used within SHERIFS.

4.3.2 Background geojson file

A geojson with a MultiPolygon describes the background geometry. Several backgrounds can be included in this file, the ones chosen for a given model is identified by the property "model" of the feature. It is important that the model property correspond with the one used in the logic tree file and the fault property file.

4.3.3 Mmax geojson file - optional

This file is used in the precomputational phase to define the maximum magnitude or length that rupture can reach in different regions described by polygons.

4.3.4 Shear modulus correction geojson file - optional

This file is used in the precomputational phase to modify the shear modulus of a large number of faults in specific regions.

4.4 Input file formats - common files when using txt or geojson

4.4.1 Background_properties.txt

```
Example_Model   upperSeismoDepth    0.
Example_Model   lowerSeismoDepth    8.
Example_Model   ruptAspectRatio 1.
Example_Model   nodalPlane  0.7 270.    60. -90.
Example_Model   nodalPlane  0.3 90. 60. -90.
Example_Model   hypoDepth   0.2 2.
Example_Model   hypoDepth   0.3 4.
Example_Model   hypoDepth   0.3 6.
Example_Model   hypoDepth   0.2 8.
```

This file contains the description of the source parameters for the background region used in OpenQuake. The details of this parameters are explained in the OpenQuake manual. New lines can be added if more options need to be in the model.

4.4.2 LT_log.txt

Models

Model_Corinth_1 Model_Corinth_2 Scaling laws

WC1994 Area m le2010 Area m

MFD b value

MFD_GR bmin_1.0_bmax_1.1 bmin_0.9_bmax_1.0

MFD_YC bmin_1.1_bmax_1.2 bmin_1.2_bmax_1.3

Background

bg_BG_1 bg_BG_2

Scenario set

sc_Set_1 sc_Set_2

For advance users: This file can be modified to add hypothesis and then SHERIFS.py can be run again to overwrite the files. Hypotheses are on the same line separated by a tab except for the MFD hypothesis that are in a row with the attached b value hypothesis in line separated by a tab. Be careful when editing by hand and check your results.

All lines must finish with a tab to avoid problems.

4.4.3 sherifs.in

Run_Name : name of the run (needs to be the same as run.info)

File_geom : path to file of fault geometry

File_prop : path to file of fault property

(if a geojson file is used they are the same)

File_bg : path to file of background geometry

file_prop_bg : path to file of background property

host_model_file : xml of the host source file (optional)

bgf : path to folder, or xml file with the smoothed seismicity background

overwrite_files : if True, every file is overwritten

use_host_model : tag True when using host model

fit_quality : default is 5 meaning the resulting mfd is withing 5% of the target shape

4.4.4 run.info

Information on run : Example

Option map: no

Site longitude : 22.090575

Site latitude : 38.250372

Vs30 : 800.0

Site Z1000 :100.0

Site Z2500 :5.0

nb_lT_samp :0

rup_mesh :0.5

source_discr :5.0

investigation_time : 50

Probability of exceedance : 0.1

trunc_lvl : 3

max_dist : 200 nb sample (sr,b_value, Mmax) : 20

Mmin : 5.0

Random seed : 805

SR correl : True

SR increment size (mm/yr) : 0.001

Mmax range : 0 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

intensity_i : PGA 0.01 1.5 10

This file contains the information entered in the first window of SHERIFS, see above for more detail. In the same manner as LT_log.txt, it can be modified for running again SHERIFS.py.

4.4.5 bg_seismicity.txt

bg ”name of the bg branch”

0.2 0.2 0.3 0.3 0.4 0.6 0.75 0.95 0.999

add new lines for new hypotheses.

This file contains the ratio of earthquakes on the faults for magnitude 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5 and 8.0.

4.5 Faults_n_scenario.txt

First line (list of the faults in the model)	F1 F2 F3 F5
Next lines, rupture scenarios	F1 F2
	F1 F2 F3
	F1 F2 F3 F5
	F2 F3
	F2 F3 F5
	F3 F5

The first line of this file is the list of faults in the model; the following lines are the possible FtF ruptures in the model. This file can also be modified manually before re-running SHERIFS.py. Filling all the possible FtF rupture of a model with the graphical interface can be a drag, if your model has a lot of possible FtF ruptures, it is much easier to change it afterwards. But as always, be careful when editing manually the files.

Each fault name is separated by a space. No tab and no newline at the end.

4.6 Running SHERIFS

In order to run SHERIFS with the command line, move to the SHERIFS repo and run the following line:

python 1_SHERIFS.py path$_{\rightarrow_{i}n.txt}$

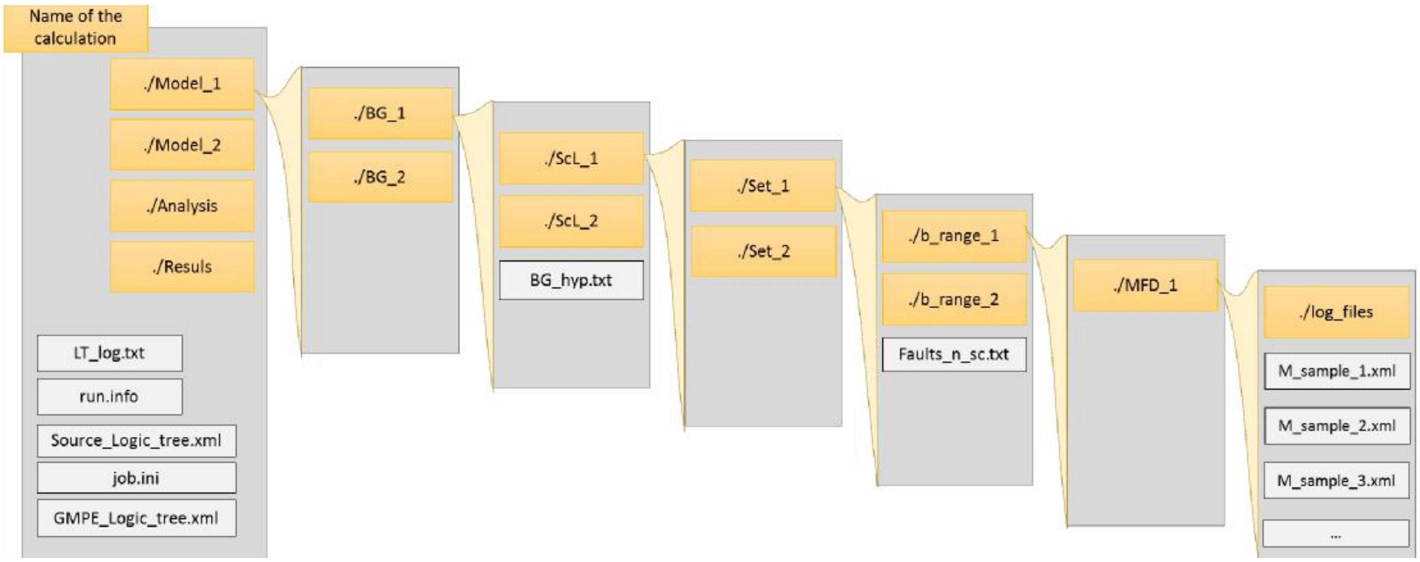
... The python code creates OpenQuake input files and many different log files for each branch of the logic tree (including the random sampling of the slip-rates, the Mmax and the b value)....

```
Console 3/A
invalid value encountered in true_divide
  shape_mfd_i = (moment_rate_in_bin)/sum(moment_rate_in_bin)
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 32
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 2
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3253.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 26
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 3
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3370.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 22
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 4
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3154.0
1%
25%
50%
- target set -
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 32
Example2/Example_Model/bg_BG_1/WC1994_A_m/sc_Set_1/bmin_0.9_bmax_1.1/MFD_GR sample : 5
7.736812675802851 21.149464617802565 18.031804789257524
number of dsr to spend : 3287.0
1%
25%
50%
- target set -
-target filled-
75%
90%
ratio between the target and the shape of the model : 0.99
ratio of NMS : 33
```

For each model, the name of the branch and the sample number are written. The number of dsr (slip-rate increment) to spend is displayed. The user can follow the advancement of the calculations since the code displays when 1%,25%,50%,73% and 90% of the slip-rate budget is spent. When the rate of the three largest bins of magnitude is limited, the target is set and the code writes 'target set'. At the end of the calculation for one branch, the code writes the ratio between the target shape and the actual shape. 1.0 is a perfect score. If the ratio is not good enough given the error accepted (indicated by the user in the 1_SHERIFS.py file), the model is ran another time with a smaller dsr.

5 Output files

Folder structure The whole hazard calculation is in the folder with the name of the calculation.



In this folder, there are three OpenQuake files (Source_logic_tree.xml, Job.ini and GMPE_logic_tree.xml if the option build the GMPE logic tree was selected), two files containing the information set by the user and the folders for each model explored.

Each level of the logic tree has its own level of the folder structure.

5.1 Description of the created files

6 Reality check of your hazard models

Choose the earthquake catalog you wish to use and format it as following:

The catalog should be in the shape of the example (first line is the name of the column). If the uncertainty of magnitudes in the catalog is not specified, input a default value.

lines to change or verify in Visualisation.py :

Name of the run

File_bg : file containing the geometry of the background. Same file used for creating the models. The global_GEM_isc_catalog.txt

Year	Mo	Da	time	Lat	Lon	Dep	M	MUnc	MType
1008	4	27	18	34.6	47.4	0	7	0.00	s
1033	12	5	0	32.5	35.5	0	7.4	0.40	s
1038	1	15	0	38.4	112.9	0	7.25	0.00	s
1045	4	5	0	40	38	0	7.4	0.40	s
1050	8	5	0	41	33.5	0	7.4	0.40	s
1052	0	0	0	31.5	50	0	6.8	0.00	s
1052	6	2	0	36.2	57.7	0	7	0.00	s
1058	11	0	0	35.8	43.6	25	7.2	0.00	s
1063	7	30	0	34.4	36.2	32	6.9	0.00	s
1063	9	23	0	40.867	27.411	0	7.14	0.30	w
1068	3	18	8	28.5	36.7	0	7	0.00	s
1096	12	11	8	34	137.5	0	8.3	0.00	jma
1099	2	16	6	33	135.5	0	8.2	0.00	jma
1107	2	12	3	45.7	26.6	150	7.1	0.30	w
1114	11	0	0	37.3	38.5	40	7.4	0.00	s
1114	11	29	0	37.5	37.2	0	6.9	0.30	s
1117	7	0	0	36	106	0	7	0.00	s
1125	9	6	0	36.1	103.7	0	7	0.00	s
1126	8	8	0	45.7	26.6	150	7.1	0.30	w
1139	9	30	0	40.3	46.2	23	7.7	0.00	s

geometry of the background will be used to extract the earthquake catalog for comparisons.

File_fault_geometry : file containing the geometry of the faults. Same file used for creating the models.

File_fault_data : (optional) file with information of earthquake rate of a specific fault. If information is available on a specific fault (rate of historical or instrumental earthquake, or rate of paleoearthquake located on this fault).

Structure :

Model	fault_name	type	M	sigma$_M$	rate	sigma$_rate$
Model_1	Fault_name	pal	6.4	0.4	0.003	0.002
Model_1	Fault_name	cat	6.0	0.4	0.006	0.001
...	...	...	...	...	...	...

Catalog_file : name of the catalog file you want to use. By default, the SHEEC catalog is used but it can be modified.

Completeness_file : File containing an estimation of the completeness for each magnitude. It is possible to explore several completeness hypotheses.

Name_completeness1	4.0,4.4	4.5,4.9	5.0,5.4	5.5,5.9	6.0,6.4	6.5,6.9	7.0,7.4	7.5,7.9	8.0,8.4
Weight	1996	1962	1958	1904	1725	1725	1725	1725	1725
Name_completeness2	4.0,4.4	4.5,4.9	5.0,5.4	5.5,5.9	6.0,6.4	6.5,6.9	7.0,7.4	7.5,7.9	8.0,8.4
Weight	1992	1960	1950	1900	1700	1725	1725	1725	1725

Sub_area_file : If you want to extract a sub region of your model to compare the model rate to the catalog, define the coordinate of this zone in this file.

Structure of the file:

Model_name Sub_area_name lat,lon lat,lon lat,lon ...

Model_2 ...

(! no empty line, tab at the end of each line!)

Coordinate of the **llcr** (lowerleftcorner)and **urcr** (upperrigthcorner) for setting the rectangle for the maps. In order to be able to visualize more rapidly different aspects of the model, Booleans can be turned on to activate the visualization of different parts of the model.

Plot_mfd	Plot the MFD for each node of the logic tree.
Plot_mfd_detailed	Plot the MFD for sub selection of the logic tree combining different hypothesis.
Plot_Mmax	Plot the distribution of maximum magnitude in the model.
Plt_as_rep	For each node of the logic tree, give the proportion of aseismic slip in the models.
Plot_rup_freq	Calculate the rupture rate for each fault of the model and for eventual subareas defined by the user.
plot_moment_rate	Plot the moment rate in different branches of the model and compare it to the moment rate in the catalog.
Visual_FtF	Draw map of all the FtF ruptures for each model hypothesis and each scenario set hypothesis.

The created figures and text files are located in the folder /analysis.

./analysis/txt_files/ contains text files containing the MFD of each branch and each source of the logic tree, the partitioning of the slip-rate between the single ruptures and the complex ruptures for each fault source and more...

list of the files and short description, see code for exact detail of what is being done, any user is of course welcome to generate more files:

- branch_cumMFD.txt : cumulative MFD of each branch of the logic tree.
- Branch_vs_catalog.txt : ration between the rate in the catalog and the modelled rate for each branch
- faults_MFD.txt : very rough file containing the array for the MFD of each individual source of the each individual branch of the logic tree.
- In V1.1 : lT_metrics.txt : each line is a branch of the logic tree with the mean slip-rate of the faults, and the scores when comparing to the catalog, the Mmax score, the MNS score and the paleo score. (see sample_analysis.py for more details)
- mean_parameters_faults.txt : for each model, set and fault, gives the mean slip-rate and the mean Mmax.
- slip_rate_sampling.txt : for each branch of the model and each fault, gives the slip-rate randomly selected for this sample. Sample 1 is always the mean slip-rate.
- slip_rep_on_faults_data : for each branch of the logic tree and fault, explains how the slip-rate has been used. The first columns of the files are describing the branch, then the name of the fault the first number is the percentage of slip budget spent on single fault ruptures, the second number is for ruptures involving two faults, then three faults and so on.. the last number is the NMS.
- slip_rep... : similar but average for whole parts of the logic tree corresponding to the name of the file

./analysis/figures/ contains a list of folders containing the different figures created.

./analyse_branch contains cumulative MFDs for each branch of the logic tree and the ratio of aseismic slip in the model. Exploring the folders and files in this folder will allow selecting some branches of the logic tree. In the folder Model, you will find more detail. /catalog contains the catalog MFD and maps as well as the number of earthquake of each magnitude in the catalog. If subareas have been used, the rates from the catalog are available in a sub folder.

./;compare moment_rate contains figures of box plots of the modeled moment rate, the model moment rate if no NMS was considered and the moment rate calculated from the catalog.

./FtF contains the map visualization of each FtF scenario taken as input in each FtF set hypothesis of the logic tree. This folder also contains the map of the maximum magnitude for each fault in the system, the map of mean slip-rate for each input model, the mean slip-rate that was considered as seismic on each fault and the map of NMS slip ratio on each fault. The name of the file details which branches of the logic tree are taken into account in order to calculate the average value presented for each fault on the map.

./mfd contains the MFD of the whole logic tree.

./Mmax contains statistics on the maximum magnitude in the logic tree, for the whole logic tree and in sub folders, for specific branches of the logic tree. This folder also contains the distribution of lengths and areas of the considered ruptures. If there is a gap in those distribution (some large rupture are considered but no intermediate ones, it can cause a lot of NMS slip in the SHERIFS calculations).

In V1.1 : In the folder named after each model, and the subfolders named after each scenario set, you will find a figure for each fault showing the number of ruptures considered on this fault section able to generate a magnitude larger or equal to a given magnitude. If for a fault, some very large magnitudes are considered, but no intermediate ones, it is likely that there is a problem in the SHERIFS calculation and that more FtF rupture need to be added in the intermediate range.

./rupture_rate_on_each_fault contains the MFD of each individual fault of fault section as well as the mfd of the sub_area if some have been defined. The plotted rates are the rates of each rupture including the fault section. Concerning the sub area, the rate of each FtF completely contained in the subarea is included and the rate of FtF rupture only partially rupturing in the subarea are multiplied by the ratio of number of sections of the FtF scenario in the subarea over the total number of sections of the FtF scenario.

In V1.1:This folder also contains txt file presenting the Magnitude of Median moment rate Mmmr for each fault for several branches of the logic tree. The Mmmr is the magnitude for which half of the moment rate of this fault is released in larger magnitude and half of the moment rate of this fault is released in smaller magnitudes. It's a good indicator of how much characteristic or GR the MFD of the fault is.

./sampling_analysis contains figures illustrating the fit of each branch to the catalog. Advanced SHERIFS users with python coding skills can use this tool to test hypotheses of the logic tree and sampling of parameters.

In V1.1: The **model_performance.png** figure exposes how well a model performs against data and if the MSN on some faults of the model is acceptable. (More details to be added once this feature is complete)

7 Weight the logic tree

Once you have looked at your model, performed consistency checks against the data, you are ready to set the weight for each branch of the logic tree.

The sum of the branches will be calculated and displayed. If this sum is not exactly one, the user is required to change manually the weights in the openquake logic tree input file. Once the logic Tree is weighted, you can run the Openquake Engine.

(For running OpenQuake : see `github.com/gem/oq-engine`)

OQ command lines :

oq engine –run job.ini

oq engine –exports-outputs #run ./results

oq export hcurves/all

References

[1] T. Chartier, O. Scotti, H. Lyon-Caen, and A. Boiselet. Methodology for earthquake rupture rate estimates of fault networks: example for the western corinth rift, greece. *Natural Hazards and Earth System Sciences*, 17(10):1857–1869, 2017.