

Empirical aggregation

As we now have multiple models available to some of our intensity measures we wish to be able to combine these intensity measures into aggregation files.

Because most models are able to create values for multiple intensity measures these values from the same model should be grouped where possible, as they are more likely to be aligned with each other.

Current functionality

The current workflow is only able to calculate and aggregate intensity measures for a single model per measure.

New functionality

The new workflow is able to use a config file specifying multiple models per intensity measure to get multiple values for each intensity measure, and then aggregate them into multiple files with one model per measure.

Currently all aggregation files are created, however this should be restricted based on grouping models together where possible.

Permutations

Currently all permutations of intensity measure files are generated, such that no aggregation contains the same measure twice.

The aggregated file naming scheme is as follows:

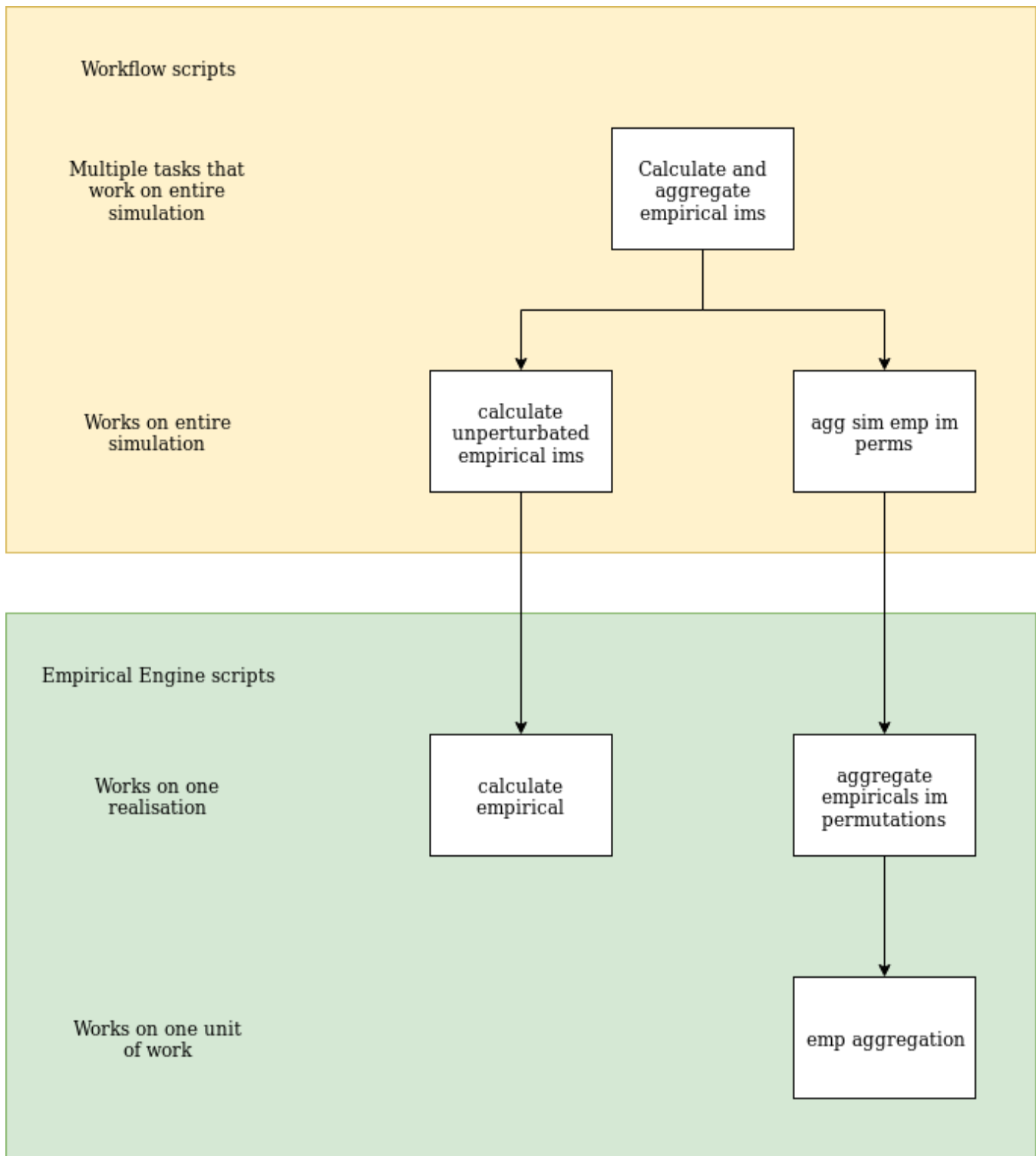
`<realisation>_<<model>_<measures>>.csv`

Where the section `<<model>_<measures>>` is repeated for every model in the aggregation, and each measure described by a model is separated with an underscore.

In the case that all measures are described by the same model for the given aggregation then the measures are omitted from the file name, which will then only contain the realisation name and model name.

Script breakdown

A diagram of the empirical calculation and aggregation scripts is available below:



Usage

If Rrups have been run for the first realisation of every event or fault in a simulation structured run, then the script `calculate_and_aggregate_empirical_ims.py` in the `scripts/empirical` directory of the `slurm_gmsim_workflow` repository can be used to generate aggregated empirical files for each event or fault.

```

$ python calculate_and_aggregate_empirical_ims.py -h
usage: Calculates and aggregates empirical IMs for a simulation directory
       [-h] [--version VERSION] [--n_processes N_PROCESSES]
       [--vs30_default VS30_DEFAULT] [--config CONFIG] [--extended_period]
       fault_selection_file [simulation_root]

positional arguments:
  fault_selection_file  Path to the fault selection file containing the list
                        of events to operate on
  simulation_root       The directory containing a simulations Data and Runs
                        folders

optional arguments:
  -h, --help            show this help message and exit
  --version VERSION, -v VERSION
                        The version of the simulation
  --n_processes N_PROCESSES, -n N_PROCESSES
                        number of processes
  --vs30_default VS30_DEFAULT
                        Sets the default value for the vs30
  --config CONFIG, -c CONFIG
                        configuration file to select which model is being used
  --extended_period, -e
                        Indicate the use of extended(100) pSA periods

```

For example to run this for the 19p5 cybershake the following command would be used:

```
python calculate_and_aggregate_empirical_ims.py /nesi/nobackup/nesi00213/RunFolder/Cybershake/v19p5/Data/Inputs
/list.txt /nesi/nobackup/nesi00213/RunFolder/Cybershake/v19p5/ -v 19p5 -n 40
```