Bayesian Earthquake Dating (BED) v1.0.1

Joakim Beck (joakim.beck@kaust.edu.sa) and Sören Wolfers (soeren.wolfers@gmail.com)

CEMSE, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, KSA

August 22, 2018

We demonstrate how to set up a new case study and to run the BED code, based on the Fiamignano (FIAM) case study. We recommend MATLAB 2016a or newer because of the improved compression available in these versions for the storage of output files.

1 Setup

First, create a directory with the name of the case study, e.g., `CaseStudies/FIAM/`, in the `CaseStudies/` directory. The directory must include the data files `rock.txt`, `colluvium.txt`, and `magfield.txt` in the same format required by the code of Schlagenhauf et al. (2010). Next, create a MATLAB file with the same name as the directory, `FIAM.m` and the following minimal structure:

```
function [parameters,prior,settings,truth]=FIAM(parameters,prior,settings)
parameters.H_sc = 2705;            % scarp height along fault (cm)
parameters.H_tr = 115;             % trench depth along fault (cm)
parameters.alpha = 23;             % colluvium wedge dip (degrees)
parameters.beta = 42;              % preserved scarp dip (degrees)
parameters.gamma = 33;             % upper eroded scarp dip (degrees)
parameters.rho_rock = 2.7;         % scarp rock mean density (g cm^{-3})
parameters.rho_coll = @(~) UniformProposal(1.2,1.8); % colluvial wedge mean density (g cm^{-3})
prior.d_max = 110;                 % maximal displacement (cm)
end
```

2 Run simulations

Next, run the BED code as follows:

```
>> save_interval=60;
>> RunBED('FIAM',save_interval)
```

where `save_interval` specifies how frequently (in minutes) the results are saved. (Note, however, that the initial offline phase takes around 20 minutes, and the first interval is only started afterwards.) The results are stored in `results.mat` in the case study directory. To terminate a run, press `Ctrl+C`; by default, the code keeps running indefinitely. The resume a terminated run, simply use `>> RunBED('FIAM',save_interval)` again. If changes to the settings of the case study were made, however, the old `results.mat` must be manually removed first.

3 Visualize results

To visualize the results, use `>> VisualizeBED('FIAM')`. This command can be run as soon as the output file exists, and if desired in parallel to an ongoing run in a separate MATLAB instance. It shows a wide range of posterior distributions and the Gelman-Rubin MCMC convergence diagnostic. Note that, depending on the number of generated samples, running this command may take multiple minutes. The amount of burn-in (in percent), the time period to visualize (in kyr), the width of confidence intervals, and the time-resolution of plots (when applicable) can be controled in the main window.

To obtain statistically significant plots, the burn-in period should be tuned such that the maximal Potential Scale Reduction Factor (displayed in figure 'Convergence diagnostic') is below 1.1 (the maximum can be read off the plot but is also printed in the terminal). If this is not possible for any value of the burn-in, more simulations are necessary.

If the simulations were done on a different computer, the contents of a `results.mat` file may manually be loaded into MATLAB and visualized using `>> VisualizeBED(results)` instead of by reference to the case study name.

4 Load results for manual analysis

To analyze the results by hand, use `>> [scenarios,weights] = LoadBED('FIAM')` or `>> [scenarios,weights] = LoadBED('FIAM',burn,thinning)` with `burn` being the burn-in ratio, between 0 and 1, and `thinning` stating that every `thinning`-th sample of the MCMC algorithm will be returned. This returns a list of scenarios (MCMC samples) together with their weights (with larger weights indicating larger posterior probability).

For example, to extract the highest likelihood scenario, use

```
>> [~,ii] = sort(cellfun(@(scenario) scenario.likelihood,scenarios));
>> scenarios{ii(end)}
```

To compute the posterior mean of a quantity of interest, form a weighted average. For example, to compute the posterior mean of the next earthquake time, use

```
>> sum(weights.*cellfun(@(scenario) scenario.time_after,scenarios))
```

Remarks

The following default settings and prior parameters can be found in `MCMC/add_defaults.m` and can be changed in the case study file (`CaseStudy/FIAM/FIAM.m`) simply by adding and modifying the corresponding line:

```
prior.recurrence_mean_min = 200; % Average interevent times are drawn from Inverse Gamma distribution
prior.recurrence_mean_max = 2000;% with parameters `recurrence_mean` and `recurrence_alpha`;
prior.recurrence_alpha_min = 1;  % these parameters are inferred within the ranges given here.
prior.recurrence_alpha_max = 10; % ...
prior.switch_distance_min = 3e3; % Distance between different long-scale slip-activity regions is drawn
prior.switch_distance_max = 3e4; % from exponential distribution with (inferred) mean `switch_distance`
prior.tau_min = 0.5;             % Minimal short-scale variability of inter-event times
prior.tau_max = 1.5;             % Maximal ...
prior.d_min = 10;                % Minimal displacement (along fault plane)
prior.d_max = 300;               % Maximal displacement (along fault plane)
prior.T_init_min = -20000;       % Lower bound on demise of LGM
prior.T_init_max = -12000;       % Upper bound on demise of LGM
prior.p_zero_freq_max = 0;       % Likelihood of a long-scale activity region with zero activity
prior.no_more_slips = 0;         % Time after which no more earthquakes occured
prior.previous_earthquakes = []; % 2xN array of known earthquake times and displacement sizes
settings.T_min = -30000;         % Start time of simulations
settings.debug = false;          % Display debug information during runs
settings.group_size = 2;         % Number of independent Markov Chains.
settings.pt_levels = gamma(linspace(2,2+19/6,20)); % Parallel tempering levels
settings.dT = 50;                % Time discretization in simulations
settings.rho_max = 0.1;          % Model error is inferred between 0 and rho_max
settings.correlation_length_max = 0;  % Correlation length of model error
settings.modelscarp =  @simulateCl36; % Function that simulates 36Cl concentrations
settings.L = 6;                  % Accuracy of sparse grid interpolation in offline phase
settings.thinning = 5;           % One out of how many samples should be saved
parameters.Psi_s = @(~) NormalProposal(48.8,1.7); % Production rate of 36Cl by ...
                                 % spallation of 40Ca  (atoms/g/year) (Gosse & Phillips (2001))
parameters.Psi_mu = @(~) NormalProposal(190,19); % Production rate of 36Cl by muon capture(atoms/g/year)
parameters.Lambda_sp = @(~) UniformProposal(180,220); % Attenuation length, spallation
parameters.Lambda_mu = @(~) UniformProposal(1300,1700); % Attenuation length, muon capture
```

In particular, to change the 36Cl simulation code, for example to that of Schlagenhauf et. al (2010), add the line `settings.modelscarp = @<alternativeSimulator>` to the case study file.

The variable `truth` that is returned by the case study function can be used to add the truth in case it is known (synthetic test cases). For this purpose, add
`truth.jumps = <array of displacements (cm), e.g. [10,30]>` and
`truth.times = <array of event times (yr), e.g. [-1e4,-1e3]>`.

The settings of the offline phase of our 36Cl simulator can be found in `SimulateCl36/offline_phase.m`:

```
rho_min = 1.2;
rho_max = 1.8;
Lambda_sp_min = 180;
Lambda_sp_max = 220;
Lambda_mu_min = 1300;
Lambda_mu_max = 1700;
```

The ranges specified here must cover those of the corresponding prior distributions in the case study file (or the defaults file), namely:

```
parameters.rho_coll = @(~) UniformProposal(1.2,1.8);
parameters.Lambda_sp = @(~) UniformProposal(180,220);
parameters.Lambda_mu = @(~) UniformProposal(1300,1700);
```