mcmc_eq - Bayesian Markov chain Monte Carlo simultaneous inversion for local earthquake hypocentres and 1-D velocity structure

T. Ryberg, Ch. Haberland, J. D. Pesicek

This is a short description of the Markov chain Monte Carlo inversion code suitable for inversion of P&S travel time data of local earthquakes to simultaneously locate earthquakes, derive 1D velocity models (Vp, Vp/Vs) and to derive station corrections. The code is an implementation of a transdimensional, hierarchical Markov chain Monte Carlo inversion. Hierarchical means, that the data noise (everything which can not be explained by the models) is treated as unknown and as a variable in the inversion. The noise model (for 4 P and 4 S classes) is assumed to have a Gaussian distribution. Transdimensional means, that the number of layers (Voronoi cells) to describe the model is treated as unknown and variable. This number represents a proxy to the model complexity. The variables to describe/estimate the data noise and the number cells will be determined during model space exploration (Markov chains). Technical details can be found in Bodin (PhD thesis), Bodin et al. (2012) and Ryberg & Haberland (2019).

The code basically represents a stochastic (Monte Carlo) equivalent of the inversion code *velest* (Kissling et al., 1994). In particular it is quite similar when following the approach to calculate the so-called “Minimum 1D model” which means to run *velest* with a variety of different initial models and parameters (e.g. starting models with different number of layers, range of velocities etc.). See Husen et al (1999) or Kissling et al. (1994) for examples of the Minimum-1D model approach. Of course, some limitations, assumptions and approximations of *velest* are treated differently in our code. Furthermore, our approach resembles somehow aspects of the statistical concept of NonLinLoc (Lomax et al., 2000), however, only regarding the hypocentre localization part.

The inversion run consists of two separate parts: the model space exploration by constructing Markov chains and, in a second step, the analysis of the models of those chains (averaging, statistical analysis, construction of reference model). In- and output files for all codes are ASCII files. A forward code, i.e. calculation of synthetic data files for a given model is provided as well.

To use the code it is assumed to have a gcc compiler installed on a Linux PC, the provided scripts need the GMT package installed. We have run the code on a computer with Ubuntu 20.04.01, gcc 9.4.0 and GMT 4.5.5 (quite old). Access to a (Linux) cluster with plenty of CPUs is HIGHLY recommended to run multiple Markov chains simultaneously. Our experience is that running multiple chains instead of a single one significantly accelerates the inversion process, i.e. actual time to wait for the inversion result.

The actual McMC code is provided together with some GMT scripts for displaying purpose, no GUI whatsoever.

Compilation

Firstly you have to compile the code by typing *make clean*
This step prepares the compilation, then type *make* to compile. This should produce the following codes
mcmc_eq - the inversion code
fw - forward code to generate synthetic data

analyse_eq - used to derive the reference solution by analysing the output of multiple runs of *mcmc_eq*.

Testing

Before running the inversion with your own data we recommend to generate a synthetic data set for a known model (velocity, earthquakes, station corrections) and run an inversion. The example provided here (from Ryberg & Haberland, 2019) has an ideal station and quake distribution (small azimuthal gaps). After the inversion the output can be compared with the true model. If this inversion run had been successful, we recommend to "replace" the travel times by the own data set, adjust the configuration file and run the actual inversion.

Example

You can generate the synthetic model and calculate the travel time data set by running *make_synthetics.*

This will generate the velocity model, calculate synthetic travel times, add some Gaussian distributed noise (time jitter, pick class dependent) to them – file *picks*.

Then you should run the inversion code multiple times (say 100) by either manual typing
./mcmc_eq config_eqx.dat rjx-000.out picks
./mcmc_eq config_eqx.dat rjx-001.out picks
./mcmc_eq config_eqx.dat rjx-002.out picks
...
./mcmc_eq config_eqx.dat rjx-099.out picks

Ignore the output on the command line (this is a "progress" report). Every run for the example will take around 1-3 hours (depending on your CPU speed).

Alternatively you might use a job queuing system on a Linux cluster system like for instance *slurm*. Alternatively, we provide two scripts to run chains for a *slurm* based systems:
run_sequence and *run.job*. For instance type *run_sequence 100* to start 100 chains (each calling *run.job*). Have a look at those scripts and adjust them to your *slurm* system.

When all calculations are done, 100 files containing a sequence of models is generated and stored in the rjx-???.out files.

In a next step you might look at the evolution of the misfit and number of layers by typing
dispe

It generates a PostScript file (*evo.ps*) where you can see the evolution of the misfit (bottom) and model dimension (number of layers) of the models of the Markov chain for all chains in a heat-map style. Since all chains start with random models the misfit is high at the beginning and than gradual decreases (burn-in phase), finally reaches a stable level where the model space exploration starts. Black are the misfits (cell numbers) for all models of the chains, blue are the values for all model beyond the burn-in phase and red are values for the 90% (see *dispe*) best fitting, post burn-in models. Expressed differently, red are the 90% best fitting from the "blue" models. Sometimes several chains might have difficulties to converge (for instance get trapped in local misfit minima), by taking only 90% of the best fitting models we can "automatically" exclude the models from

problematic chains from further analysis. Try other than 90% values in case you have more (less) problematic chains!

Once you determined a suitable value (90%) you could proceed with the next step: deriving a reference model by averaging post burn-in models.
Executing the script *disp_m_average_sl* generates the reference model from the inversion runs and stores it in *resmcnx.dat* (shown in *xselect.ps*).

Several other scripts are provided for reference:

Executing the script *disp_error* generates produces a comparison between the true and inverted model (*error.ps*).

disp_eq generates a map with all quake locations (*eq.ps*).

disp_eq_z shows the location distribution of an individual quake (# provide in *disp_eq_z*) in a heat-map (*eq_200.ps* for quake # 200).

disp_msft_dist show the input travel time picks and the predicted ones, sorted according to epicentral distance (*msftp.ps*).

Given the almost “perfect” distribution of sources and receivers, the recovery of the model (i.e. quake loactions, station corrections, velocity model, data noise) is quite good. Some smoothing at the edges of the velocity jumps is expected due to the fact that we added noise to the data (see *make_synthetics).* Less noise will lead to better recovered models, and vice-versa.

Input, output and configuration file description

All input and output files are in ASCII format.

Travel time picks (file *picks*):

The file containing the travel time picks (*picks*) is in the following format:

```
# 0 47 36 1104637590.560000
N408 073 P 12.300 24.420 -1.629 7.79909 3
N520 102 P -2.780 21.492 -0.022 5.84615 2
…
B407 019 P -2.187 -33.630 -0.240 12.3349 3
N408 073 S 12.300 24.420 -1.629 13.0245 0
N520 102 S -2.780 21.492 -0.022 11.7472 2
…
# 1 11 5 1104737417.580000
N521 103 P -37.895 48.354 -1.318 4.79113 0
…
```

The lines starting with “#” indicate information about an event (earthquake).
Event number (starting with 0, increasing monotonous, no gaps!)
number of P travel time picks, make sure it matches the actual number of picks
number of S travel time picks, make sure it matches the actual number of picks
reference time

In case you want to keep a position of an earthquake (i.e. depth at 10 km) fixed, please add -9999 -9999 10.0 to the line. “-9999” acts as a placeholder for a value to be inverted.

The next lines contain the actual arrival time picks:
Station name (actually not used later on)
station number (starting with 0, increasing monotonous, no gaps! Should match the station names)
wave type: P or S
x y z station coordinates (in km) with negative z values “above” sea level
relative pick time (with) respect to the reference time in the line starting with “#”
pick class (from 0 to 3 for P & S).

next event data subset.

Configuration file (*config_eqx.dat*) with line numbers added:

```
# value(s) # comment(s)
1 2.0           # forward dx
2 200           # forward NX
3 200           # forward NY
4 62            # forward NZ
5 -200.0        # model starts at X0
6 -200.0        # model starts at Y0
7 -4.0          # model starts at Z0
8 200           # max # of cells/layers
9 2.0           # minimum vel
10 12.0         # maximum vel
11 1.0          # minimum vpvs
12 3.0          # maximum vpvs
13 0.001        # minimum noise
14 10.0         # maximum noise
15 -5.0         # min residual
16 5.0          # max residual
17 10.0         # sdev for x dummy
18 10.0         # sdev for y dummy
19 5.0          # sdev for z
20 0.05         # sdev for vel
21 0.02         # sdev for vp/vs
22 0.01         # sdev for noise
23 1.0 2        # sdev x EQ, factor for epicenter search accelaration
24 1.0          # sdev y EQ
25 1.0          # sdev z EQ
26 0.02             # sdev residual
27 0.05             # minimum layer thickness in fractions of dz, i <0 no LVZ !!! should be !=0
28 1 0 0.0 0.0      # reference station + flag, =0: zero mean, =1: p fixed, =2: P&S fixed, =-1:
invert for P only, =-2 for S only
29 0                # Voronoi cells if = 0, triangulation if = 1
30 50000 250000     # number of models in chain
31 2000             # output every nth model
32 -77  1           # if <0 true random search, otherwise from seed n, 0=str 1=eik
```

33 QN QVRPBDMN # model modification tests
34 0 VRN # 0-mcmc, 1-prior only, 2-output regular grid, 3-input from model file: V and/or Q R N, i.e 3 QV
35 1 dummy 1 # dummy
36 5.0 0.5 0.03 # vp to start with
37 1.9 0.2 # vp/vs to start with
38 15 5 # cell number to start with
39 1.0 # start_noise
40 0.0 0.0 # delay & sdev to start with
41 0.5 0.5 # start EQ location from center (horizontal) or surface (vertical) 1 for full space

1. grid increment of the grid (forward calculation of travel times). Warning: halving the increment causes 4 time larger run times. For setting up the configuration file and testing: use a coarse grid

2. NX – number of grid points in X direction
3. NY – number of grid points in Y direction
4. NZ – number of grid points in Z direction
5. X0 – starting X position (i.e. left corner)
6. Y0 – starting Y position (i.e. left corner)
7. Z0 – top of model (negative if above “above” sea level)
Make sure that NX, NY, NZ, X0, Y0, Z0 includes all stations and potentially all quakes. Z0 & NZ should be deep enough for all quakes.
8. maximum number of layers, should be large enough, otherwise not essentially

Sampling ranges for model properties:
9. minimum Vp velocity
10. maximum Vp velocity, both should span a wide range (including non-physical values) to ensure sufficient sampling of the posterior distributions
11. minimum Vp/Vs
12. maximum VpVs, again wide range...
13. minimum noise level for all pick classes
14. maximum noise level, again wide range...
15. minimum station corrections
16. maximum station correction, …

Model proposal “steps”:
17. not used
18. not used
19. random movement steps of layers along depth in km, this is the standard deviation of a gaussian random number
20. random step for Vp velocity in km/s, this is the standard deviation of a gaussian random number for changing the velocity in a layer
21. same for Vp/Vs ratio
22. same for data noise
23. same for quake “movements” in X direction, second value is factor applied to the first one to accelerate the search in the first phase
24. same for Y direction
25. same for Z direction

26. random movement steps for station correction

Small proposal steps will result in a “steepest gradient search”, you might get trapped in a local misfit minimum. Huge steps will result in a classical Monte Carlo search where most of the proposed models will be discarded because of bad misfits. Make a test run with one chain and have a look at the end of the result file (rjx-000.out). There is a summary of the acceptance rate for every step. Try to adjust the step values so that acceptance rate between 30 and 50% is achieved.

27. only models are accepted if the minimum thickness of a layer is exceeded (fraction of grid increment). If this value is negative, only models without a Vp and/or Vs low velocity layer are accepted. Make it small positive if you allow low velocity zones or small negative if not.

28. reference station:

xx 0 xx xx : zero mean of P & S station corrections (recommended)
12 1 value_p value_s : if you want to have station #12 correction for P fixed at value_p
12 2 value_p value_s : if you want to have station #12 correction for P&S fixed at value_p/value_s
12 -1 xx xx : if you want to invert for P station corrections only
12 -2 xx xx : if you want to invert for S station corrections only

Recommended is xx 0 xx xx → zero mean station corrections.

29. if 0 then constant velocity layers (Voronoi equivalent), if 1 then linear interpolation. Recommended is 0!

30. two values: number of Markov chain models in phase 1 and phase 2. To accelerate the search we split the generation of the individual Markov chains into to phase: during the first phase the velocity model is kept unchanged, i.e. at the starting model (see line 33). This allows the quakes to roughly “move” to the epicenters. In the following phase a complete search is performed.

31. only every 2000th model is put into the output file (decimation).

32. if the first number is negative, the seed value for the random number generation is random. Positive numbers are used as a seed number and allow to reproduce Markov chains. Use negative numbers for a complete random search. The second number controls the calculation of travel times, use 1 for the eikonal solver.

33. controls the modification of the model (velocity, quake locations, etc.). First string is for the first search phase, second string for the second phase.
Q - means quake locations are perturbed
N - stands for data noise level perturbations (hierarchical search)
V - stands for Vp/Vs perturbations
P - stands for P velocity model perturbations
M - stands for moving a velocity layer
R - stands for station correction perturbations
B & D allows for adding/removing layers (transdimensional search)

34.
0 - full Markov chain searched
1 - takes all models, independent on misfit: this allows to get the prior distribution of parameters
2 - not used

3 - start search not from random points in model space but uses complete (or parts of) model from file model.dat, the following string controls what part of the model is used as a starting model:
V - reads the velocity model (Vp, Vp/Vs)
R - reads the station corrections
N - reads the data noise levels
Q - reads the quake locations
This feature can be used, for instance if you have a fixed velocity model, noise levels and station corrections and you want to locate quakes only. In this case use 1 VRN and put “Q Q” in line 33

35. not used

Starting model description:

36. “5.0 0.5 0.03 “ means random starting velocities (Vp) having a gaussian distribution with mean of 5.0 km/s and standard deviation of 0.5 km/s. The mean value is additionally changed according to the actual depth of that layer by adding a value according to the depth gradient of “0.03”. This allows to models which have an average gradient of “0.03”, i.e. the tendency of increasing velocity with depth. The standard deviation value adds a random component to the starting model.

37. same as line 36, but for starting Vp/Vs ratios. No gradient value is used.

38. Number of layers in starting model, same as line 37. If you want to have a fixed number of layers in the starting model, kep the second value 0. If you want to have a nearly equal distribution of layers, make the second value very large.

39. starting noise level in seconds.

40. starting values for station corrections. Same interpretation as line 38.

41. starting locations of quakes, first value controls the horizontal, second the vertical distribution . With “1.0 1.0” all starting positions will be equally distributed in the search space. “0.0 0.0” puts all starting positions in the center at z=Z0 of search space. “1.0 0.0” puts all quakes at the surface, but with equally distributed horizontal positions. “0.5 0.5” puts the starting positions in a cuboid (half the horizontal and vertical size of the entire search space) centered in the search space.

Inversion of own data

After successful inversion of the provided synthetic data set, try to convert your picks to the same format as *picks*. Then run the inversion.

Note on choice of grid increment (line 1 in configuration file): as mentioned above, it has a strong influence on the computational time. Given the nature of the eikonal solver used to calculate travel times for a given velocity model, sparse grids potentially introduce errors in the predicted travel times. This is especially true for short source-receiver distances, expressed in grid increments. Distances shorter than ~3-5 grid increments will be effected most. But: our experience is that even very sparse grid increments will still lead to a sufficient reference model. So it is worth to start with a quite sparse grid, adjust all the parameters in *config.dat*, run a test inversions. If the final model looks reasonable, try to decrease the grid increment, run the inversion again and compare with the

results of the sparser grids. Check if the final models converge, i.e. show similar major features (velocity model and quake locations).

Using the code, disclaimer

This *mcmc_eq* package is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
The *mcmc_eq* package is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. The code contains calls to subroutines (FDTIMES package) which are published under a GNU General Public License version 2. See the GNU General Public License for more details.

If you use this program for an academic project or a commercial project, citing this paper would be appreciated:
Bayesian simultaneous inversion for local earthquake hypocentres and 1-D velocity structure using minimum prior knowledge, Geophys. J. Int.(2019)218,840-854, doi: 10.1093/gji/ggz177.

Acknowledgments
Anybody?

References

Bodin, T., PhD thesis: Transdimensional Approaches to Geophysical Inverse Problems, download http://perso.ens-lyon.fr/thomas.bodin/publi.html

Bodin, T., Sambridge, M., Rawlinson, N., and Arroucau, P. (2012): Transdimensional tomography with unknown data noise, Geophys. J. Int., 189, 1536–1556, doi 10.1111/j.1365246X.2012.05414.x

S. Husen, E. Kissling, E. Flueh, G. Asch, Accurate hypocentre determination in the seismogenic zone of the subducting Nazca Plate in northern Chile using a combined on-/offshore network, *Geophysical Journal International*, Volume 138, Issue 3, September 1999, Pages 687–701, https://doi.org/10.1046/j.1365-246x.1999.00893.x

Kissling, E., W.L. Ellsworth, D. Eberhart-Phillips, and U. Kradolfer (1994) Initial reference models in local earthquake tomoqraphy, J. Geophys. Res., 99, 19635-19646, 1994.

Lomax, A., J. Virieux, P. Volant and C. Berge (2000) Probabilistic earthquake location in 3D and layered models: Introduction of a Metropolis-Gibbs method and comparison with linear locations, in *Advances in Seismic Event Location* Thurber, C.H., and N. Rabinowitz (eds.), Kluwer, Amsterdam, 101-134..

Ryberg & Haberland (2019): Bayesian simultaneous inversion for local earthquake hypocentres and 1-D velocity structure using minimum prior knowledge, Geophys. J. Int.(2019)218,840-854, doi: 10.1093/gji/ggz177