Bisous model – detecting filamentary pattern in point processes

The current document describes the parameters that control the Bisous model program. The current implementation of the Bisous model is described in the following papers:

1. Tempel E., Stoica R. S., Kipper R., Saar E., 2015, Astronomy & Computing (arXiv:)
2. Tempel E., Stoica R. S., Martínez V. J., Liivamägi L. J., Castellan G., Saar E., 2014, MNRAS, 438, 3465 (`arXiv:1308.2533`)

If you use the program in your research, please acknowledge these papers.

License

BisousModel is free software, governed by the CeCILL[1] license. Please read the license.txt file (or the last section of this manual).

Compiling

The program is written in fortran and tested using Intel Fortran Compiler. Assuming that fortran compiler is properly set in the Makefile and all necessary libraries are installed in the system, compiling should be as simple as typing `make` in command prompt.

To successfully compile the program, the following libraries should be available in the system:

- `cfitsio`. This library is not used during normal program execution. You can safely modify the code and remove any calls that require cfitsio library.
- `Numerical Recipe`. This library is only needed to calculate the incomplete beta functions. This is only used in the file `utilities.f90` – you are free to change the required library for incomplete beta function.

Parameters for configuration file

The parameters in a configuration file are divided by sections in order to keep them organised. The order of parameters within one section as well the order of sections can be arbitrary. Below, we very briefly describe all the parameters that can be set in a configuration file. Each parameter have a default value that is indicated in the parenthesis. The parameters marked with $\star$ should be kept to their default values, the change of these parameters is not needed for normal program execution. The minimum parameters that should be given when new data is analysed are given in magenta colour.

The description of parameters by sections is the following.

[1] http://www.cecill.info/index.en.html

General parameters [general]

main_option (1) Type of action to be executed. Option 1 is normal MCMC (produces set of realisations); option 2 is the analysis of results produced during MCMC (see analysis section in the configuration file). During the analysis a set of realisations are analysed in order to extract filament spines based on visit map and orientation map.

verbose (`True`) Print running statistics to terminal window. Some messages are printed regardless of this value.

resume (`True`) Resume previously interrupted MCMC simulation. If resume file does not exists program is started from beginning. The resume file is generated automatically during MCMC simulation.

resume_from_previous_snapshot⋆ (`False`) Resume from previous snapshot. Implemented to speed up the analysis of simulation data for several snapshots. Requires modification of input files to use it successfully.

max_time_hour (24.0) Maximum time (in hours) for program execution. Before this time limits, the program is terminated and resume file is generated. Interrupted program can be started from the resume file. This option is useful when running a program in a computer cluster where maximum allocated time is restricted. This parameter only affects the MCMC simulation and does not affect the analysis of the results (main_option=2).

input_type⋆ (1) The format of input data. Specific types should be defined in the program code.

file_data Name of the file that contains the input data. Datafile should contain three columns for cartesian xyz coordinates.

dummy_ Dummy variables for specific input types. Not in use for default input type (input_type=1).

root (`bm_`) The beginning of the output files. This value can be also given as an additional parameter from command line: it allows to start several programs simultaneously by external script (for simple parallelisation).

scale_factor⋆ (1.0) Scale factor for input data points. The input coordinates are multiplied by this value. In theory, it can be used to detect multi-scale filaments.

cylarr_size (100 000) The maximum size for cylinder array, maximum number of cylinders in a configuration at any given time. This is allocated in the beginning of the program and the actual size cannot exceed this value.

MCMC parameters [mcmc]

nr_cycles (10 000) Number of MCMC cycles. Temperature is adjusted after every cycle. One cycle consists of N moves and during that time the simulating temperature is constant.

nr_moves (10 000) Number of moves in one cycle. Temperature is fixed during one cycle. Number of moves should be larger than the number of cylinders in a configuration.

nr_moves_min_frac_con⋆ (1.0) Minimum number of moves in term of number of connected cylinders. It increases the "nr_moves" parameter if necessary. This is required to avoid unreasonable small number of moves between temperature updates.

every_cycle_to_stat (10) Every nth cycle is written to statistics file.

every_cycle_to_output (1000) Cylinder configuration is extracted after every nth cycle and resume point is generated.

cooling_schedule (4) Temperature cooling schedule for MCMC. 1 – constant temperature; 2 – simulated annealing; 3 – simulated tempering; 4 – combined simulated tempering plus simulated annealing.

temp_initial (2.0) Initial (maximum) temperature value.

temp_final (0.5) Final (minimum) temperature value.

st_temp_steps (10) Number of temperatures for simulated tempering.

st_expected_nrcyl_size (0.5) Adjusts the temperature change probability in simulated tempering. Larger values increase the probability for temperature change. This coefficient is given relative to the number of (expected) cylinders in a configuration.

nr_cycle_for_tempering (1000) *For combined cooling schedule.* Number of cycles for simulated tempering, before simulated annealing takes over.

nr_cycle_for_burnin (1000) *For combined cooling schedule.* Number of burn-in cycles for automatic temperature adjustments. Adjust initial minimum and maximum temperature values if they are unreasonable. Set to zero to avoid automatic temperature (minimum and maximum) adjustments. This option is provided to help user to choose reasonable minimum and maximum temperatures for simulated annealing.

st_adjust_temp_ladder_coef⋆ (0.4) *For automatic minimum and maximum temperature adjustment.* Fraction of connected cylinders for a given temperature compared with a maximum number of cylinders in any temperature.

st_adjust_temp_ladder_coef_free_min/max⋆ (0.05/1.0) *For automatic minimum and maximum temperature adjustment.* The fraction of minimum and maximum number of free cylinders in configuration compared with the number of connected cylinders.

prob_birth (0.5) Proposal probability for birth moves in MH.

prob_death (0.3) Proposal probability for death moves in MH.

prob_change (0.2) Proposal probability for change moves in MH.

prob_birth_connected (0.8) Proposal probability for connected birth for birth move.

change_delta_r (0.4) Maximum shift for cylinder centre for change move (in units of connection radius). This should be smaller than the cylinder radius in data term. If it is larger than cylinder radius, it only affects (in negative way) the computation time, not final results.

change_delta_cosi (0.95) Maximum cosine between old and new orientations for change move.

auto_sampling_volume⋆ (`True`) Set the sampling volume (volume_fraction) and normalising constant (volume_multiplier) automatically.

auto_volume_multiplier (1.0) Adjust the automatically calculated normalising constant. Increasing the value increases the number of cylinders in a configuration.

auto_samvol_acc (0.1) Accuracy for automatic sampling volume estimation. The exact value for sampling volume is not important, it have negligible effect for final results.

auto_samvol_maxtime (1.0) Maximum time (in minutes) for automatic sampling volume estimation. If maximum time is reached and accuracy is not achieved, the automatic estimation is aborted. The current best value is used for sampling volume.

volume_fraction Fraction of the total volume, where the data energy is defined, i.e. the region, where sampling is performed. Used if sampling volume parameters are not determined automatically.

volume_multiplier Sampling volume normalising constant. Used if sampling volume parameters are not determined automatically.

Data term parameters [data_term]

min_pts (3) Number of minimum points inside a cylinder. Data term is not defined if number of points is lower than that.

use_fixed_radius (*True*) Use a fixed radius for a cylinder. Otherwise, radius is determined automatically based on the radial density gradient.

cyl_rad_min (0.4) Minimum radius for a cylinder (in physical units). For fixed radius, this value is used.

cyl_rad_max (1.2) Maximum radius for a cylinder (in physical units).

cyl_len_min (3.0) Minimum length for a cylinder (in physical units).

cyl_len_max (10.0) Maximum length for a cylinder (in physical units).

cyl_rel_lvsd_min (3.0) Minimum ratio for cylinder length divided by diameter.

cyl_rel_lvsd_max (10.0) Maximum ratio for cylinder length divided by diameter.

cyl_shadow_rad⋆ (1.0) Size of cylinder shadow region in units of cylinder radius.

hyptest_uniform_den (4.0) Assumed uniform density for hypothesis testing.

hyptest_local_den (3.0) Assumed local density contrast for hypothesis testing.

hyptest_..._p_cut⋆ (0.001) Hypothesis testing minimum probability values. Can be used to reject extremely bad probabilities in hypothesis testing.

variance_coeff (0.5) Variance coefficient for data term.

hypothesis_coeff (1.0) Hypothesis coefficient for data term.

nr_of_rad_samples (10) Number of radiuses (between minimum and maximum radius) for automatic radius determination.

Interaction term parameters [interaction_term]

rad_connection (0.5) Connection radius for cylinders.

cos_orthogonal (0.5) Maximum cosine for orthogonal cylinders.

cos_parallel (0.85) Minimum cosine for parallel cylinders.

concyl_radius_difference (1.5) Maximum radius difference for two connected cylinders: one divided by the other.

repulsive_radius_difference (3.0) Maximum radius difference for repulsive cylinders. If difference is larger, then cylinders are not repulsive.

lg_gamma_use_self_regulation (`True`) Interaction potentials are determined based on data energy.

lg_gamma_use_interval (`True`) Use constant prior for interaction energies. Otherwise use fixed interaction energies.

lg_gamma_0/1/2_mean ($-1.5/-0.2/1.0$) Mean interaction potential for 0/1/2-connected cylinders in units of data potential.

lg_gamma_0/1/2_disp (0.8) Interval for interaction potentials in units of data energy standard deviation.

lg_gamma_0/1/2 Interaction energy value for 0/1/2-connected cylinder if interaction energy is not determined based on data energy.

lg_gamma_0/1/2_min/max Minimum and maximum values for interaction potential in case if self-regulation is not used.

Parameters for realisations [raw_data]

root_dir Root directory for input cylinder data. Cylinders produced during MCMC simulation(s) should be used here.

file_prefix Prefix for resume file. Should be set when multiple runs are analysed.

multiple_runs (`False`) Sets whether cylinder data comes from multiple MCMC runs. If true, each run have its own subdirectory in the main root directory.

root_dir_run (`run000`) Name of the run directory. The last digits should be numbers that define the separate simulations.

run_first (1) First run number.

run_last (1) Last run number.

rs_first/last Index for the first/last cylinder file. During MCMC several realisations were written to output. Only the last set of realisations should be used for analysis.

rs_every (1) Every nth cylinder realisation is used for analysis. This value should be increased, if the number of moves between two output realisations is not large enough.

weight_0/1/2⋆ (1.0) Weight for 0/1/2-connected cylinders in analysis.

r_smooth⋆ (1.0) Smoothing scale for analysis. Cylinder radius for analysis is multiplied by this value. This should not be changed for normal analysis.

old_format⋆ (`False`) For backward compatibility. Enables to analyse realisations produced using old code.

Parameters for analysis [analyse]

analyse_option Set the subtask that should be executed. 1 for spine extraction, -1 to run test subroutines, 2 to calculate filament statistics for input points.

output_spines Name of output filament spine file.

input_points Name of input file to analyse set of points.

output_points Name of output file for analysed set of points.

initial_grid_size (1.0) Grid size for initial visit map in physical units. This grid is used for spine extraction.

spine_location_accuracy Defines the accuracy for spine points. Must be smaller than initial grid size.

spine_location_accuracy_tuning Tuning parameter for final accuracy, sets how many times is the final accuracy better.

minimim_visitmap_value (0.05) Minimum visit map value for filament spines. Should be high enough to include reasonable number of cylinders for statistical analysis.

minimum_orientation_strength (0.75) Minimum orientation strength value for spine detection.

spine_spacing_rad_frac (1.0) Maximum spacing along filament spines in units of cylinder radius.

spine_spacing_max Physical maximum spacing along filament spines.

Boundary conditions and masking [cmask]

c_cmask_use_boundary_cyls⋆ (`False`) Use boundary cylinders in simulation. Currently, this parameter should be set to false.

c_cmask_plim_use⋆ (`False`) This parameter can be used to enable boundary limits for MCMC simulation. Currently, only rectangular box can be set. If set to false, the boundary limits are estimated automatically based on the input data points.

c_cmask_pmin/pmax_x/y/z Limiting values for simulation.

License information