Bacmeta – input/output manual

This manual describes how to set preferred simulation settings, and what kind of outputs are available for the simulation runs.

For further information of the effects of parameters in the simulation model, see **ModelManual.pdf**.
For information about running the program, e.g. command line options, see **README.md**.

See examples of the input and output files in **FeatureExample.zip** and **CaseExample.zip**, located in the ‘examples’ folder**.**

Developer welcomes all forms of feedback, help inquiries, developing suggestions and general correspondence about the program via: aleksi.sipola [at] helsinki.fi

Giving inputs:

Parameters are set with file `**simu.input**`. Custom filename of form `**simu123.input**` with identifying substring can also be used with the help of command line options.

Bacmeta does preliminary initializing of all parameters using values given in file `default.params`. Lines in that file can be copied straight into and modified in `simu.input`. Note that `default.params` should NOT be edited as it is always used for initializing necessary parameters and is assumed to be unaltered. Furthermore, only contents of `simu.input` are documented for simulation runs.

Parameter value is set with a line starting with corresponding parameter key (four letters and symbol ':'), followed by whitespace and then the value itself. A comment about the line can follow after these starting with symbol '#'. Note that the comment should not be longer than 180 characters. More thorough comments about the run can be written in the end of the file in the designated area.

Example line of setting amount of generations to 10000:

```
GENR:    10000    # This is a comment
```

Migration rate inputs:

Migration rates between populations can be set to common global value, with the parameter key 'MIGR', or if non-homogenous migration rate network is to be used, such can be given in file `**migration.input**` as adjacency matrix for weighted, directed graph, so that edge weights represent migration rates.

In essence, the input file starts with tab-separated migration rate values forming a matrix, where value in column j of row i defines migration rate from population i to population j, as a proportion of population size.

Note that the columns might not align visually, as white space can be displayed variosly across different platforms. In such case, it can be beneficial to regard columns as ordered positions, e.g. value in column 3 of row 2 is the third value in second row of the file. Using a spreadsheet program with suitable separator settings can also be helpful.

By default, Bacmeta reads file `**migration.input**`, but with command line options, custom file names of the form `**migration123.input**` can be used too.

Input documentation:

In the end of a simulation run, Bacmeta automatically saves the input parameters given in the file simu.input into output file `**rundetails*.txt**` and possible migration.input into file `**migration*input.txt**`, so that each run be replicated. Star symbol denotes the run id, see first parameter in the parameter list. `**rundetails*.txt**` contains also the seed that was used and runtime. Both of these files can be copied and renamed into the input file format for reruns, although if the run was made with seed=0, the time sampled seed number should be set as new seed value, if replication is desired.

Outputs:

Variety of output files can be created by a simulation run. Output files are created into folder `/bacmeta/outputs/`, which is created automatically if one does not already exist. All files have the filename modifier suffix given in `simu.input` or with the command line flag `-o`. The previous files are **overwritten** in each run if ones with same output filename modifier exist.

Descriptions of the files are given below along the corresponding parameters. Examples of all the files (excluding “MLSTsequences.txt”) are provided in **FeatureExample.zip**. **CaseExample.zip** displays an example use case, and contains further examples of some of the output files.

Note:

Output files can reach relatively large sizes if parameters for documentation are not set for big simulations.

Some information about the output file sizes are given below, but ultimately the file sizes are case dependent and affected by interplay of many factors, such as: size and length of simulation, event rates, diversity of populations, documentation sample sizes and frequency.

Thus before running bigger simulations, it is recommendable to: **1)** See the two example output cases in **FeatureExample.zip** and **CaseExample.zip** for concrete demonstrations of file sizes and contents in relation to the accompanying parameter sets **2)** Experiment with documentation settings for different use cases starting with following notions and gradually test to find suitable values:

-Toggle off unnecessary documentation outputs.
-Use at first only small proportions of populations for sample sizes, e.g. in sequence saving and especially in pairwise comparisons.
-MLST-sequences, i.e. sequence - allele type pairs, should not be saved without caution as they include every sequence that has occured during simulation.
-Any event documentation file might get excessively large if the event rate is high and the simulation is long. Fortunately these can be estimated quite well using short simulations.

Parameters:

Rest of this document lists the available parameters with descriptions. They are listed as follows:

Pages:	**Contents:**
Simulation:	
4	General, Initial stage
5	Genetic events
6	Population dynamics
Outputs:	
6	General, Strain-id composition
7-8	Pairwise distances, Sequences
9-10	Events, Multilocus sequence typing

	Key	Value type Default value (Switch-off)	Function	Notes
General	OPFN	Alpha-numeric 123	Run-id: Output file name modifier. i.e. the string to separate different runs.	Former files with same modifier will be overwritten!
	SEED	Integer 123456789	Seed to use Or 0 if random seed from time is to be used	Runs are deterministic in platform, but can differ between platforms.
	GENR	Positive integer 20000	Number of generations to run	
	PRIG	Positive integer 0 (0)	Debugging output generation, prints sequences and metadata to terminal.	
	PRIP	Positive integer 0	Debugging output population, prints sequences and metadata to terminal.	PRIG controls switching off.
	RORD	0 or 1 1	Keep order of the event types same (0) or randomize each generation (1)	
Initial Stage	NBAC	Positive integer 500	Number of bacteria in population	
	NLOC	Positive integer 10	Number of loci in bacteria	
	LOLE	Positive integer 1000	Length of locus	
	NPOP	Positive integer 4	Number of populations in metapopulation	
	PROA	[0,1] 0.25	Proportion of base A	These must sum to one
	PROT	[0,1] 0.25	Proportion of base T	
	PROG	[0,1] 0.25	Proportion of base G	
	PROC	[0,1] 0.25	Proportion of base C	

	Key	Value type Default value (Switch-off)	Function	Notes
Genetic events				
Mutations	MUTR	Double 2e-7	Mutation rate per nucleotide per population per generation	Typically in range [1e-5 , 1e-9],
	ATOT	[0,1] 0.3334	P(a->t) i.e. Probability of mutation from base A to T	Probabilities from each base x to the others, must sum to one
	ATOG	[0,1] 0.3333	P(a->g)	
	ATOC	[0,1] 0.3333	P(a->c)	
	TTOA	[0,1] 0.3334	P(t->a)	
	etc. from each base to all other bases			
Recombi-nations	RECR	Double 3.21	Recombination rate in relation to mutations	
	RECL	Integer 500	Recombination mean length. Give (0) if whole loci are to be recombined	
	RECA	Double 18.0	Recombination acceptance parameter for similarity test	
	RECS	0 or 1 0	Gather recombination site metadata (1) or not (0)	
Indels	INSR / DELR	Double 0 / 0	Insertion/Deletion rate in relation to mutations	
	INSL / DELL	Double 1.7 / 1.7	Insertion/Deletion length parameter for Zipf distribution	
	INDM	Double 0.02	Maximum indel length as a proportion of the loci length	

	Key	Value type Default value (Switch-off)	Function	Notes
Population dynamics				
Migrations	MIGI	0 or 1 0	Using migration rates from "migration.input"-file (1) or universal value given with MIGR (0)	
	MIGR	Double 0.01	Mean migration rate scaler	
	MIGP	Double 0.01	Migration probability	
Micro-Epidemics	MICA	Double 0.0	Mean microepidemic amount scaler	
	MICS	Double 0.0	Mean microepidemic size	
Random Selection	RSEL	0 or 1 1	Randomly select bacteria for new generation (1) or keep as is (0)	

Outputs

Example output filename and short description given under corresponding parameter(s).

Here “123” denotes example run-id and “20000” the documentation generation of the file.

General	SUMI	Integer 20000 (Negative Integer)	Summary info: Generation interval of computing summaries.	

Summaries123.txt:

Variety of information about the populations, e.g. amount of mutations, recombinations and other events.

Strain-id Composition	STRA	Integer -20000 (Negative Integer)	Strain-id composition documenting interval	

DiversityDistributionPop0-123.csv:

Number of differing **strain-ids**, i.e. C++ shared pointers addresses. Strain-id is hexadecimal string assigned to a "strain" that emerges in a specific genotype altering event. Same genotype can be represented by multiple different strain-ids as different events can lead to the same genotype. But each strain created by the specific event has unique strain-id specific only to it and its clones. Strain-id might be re-used for other strains when the previous one has become extinct and/or changed to a new genotype. Strain-id values, i.e. the hexadecimal strings, are random between runs.

	Key	Value type Default value (Switch-off)	Function	Notes
Mean pairwise distances:	MEAN	0 or 1 0 (0)	Compute and save mean distance summary every 1000 generation (1) or not (0)	
	MeanDistances123.csv:			
Mean distance of random pairs in a sample of population 0. Sample size is 0.3 times the population size. Given in intervals of 1000 generation.				
Intra-population pairwise distances:	GWDI	Integer -20000 (Negative Integer)	Loci-wise mismatches and mutation-only mismatches: Generation interval of computing and saving to file.	Possibly big file sizes. See * below.
	GWDS	[0,1] 0.01	Loci-wise mismatches and mutation-only loci-wise mismatches: Size as proportion of the population.	
	PairDistances20000-Pop0-123.csv:			
Distances, i.e. nucleotide differences, between bacteria pairs given "locus-by-locus" and in total. Given for each strain-id pair in the sample (see "Strain-id composition" above). The strain-ids in these files are random between runs, but contents are otherwise identical.				
	MGWD	0 or 1 0 (0)	Do mutation-only loci-wise mismatches along regular (1) or not (0). Activates recombination site metadata.	Possibly big file sizes. See * below.
	MutationPairDistances20000-Pop0-123.csv:			
As above, but considering only differences caused by mutations.				
Inter-population pairwise distances:	PGWI	Integer -20000 (Negative Integer)	Interpopulation: Loci-wise mismatches and mutation-only loci-wise mismatches: Size as proportion of the population.	Possibly big file sizes. See * below.
	PGWS	[0,1] 0.01	Interpopulation: Do mutation-only loci-wise mismatches along regular (1) or not (0). Activates recombination site metadata.	
	InterpopPairDistances20000-Pop0-Pop1-123.csv			
As "PairDistances20000-Pop0-123.csv:", but pairs are formed with one bacterium from population X and one from population Y.				

	Key	Value type Default value (Switch-off)	Function	Notes
	PMGW	0 or 1 0 (0)	Interpopulation: Loci-wise mismatches and mutation-only mismatches: Generation interval of computing and saving to file.	Possibly big file sizes. See ***** below.

InterpopMutationPairDistances20000-Pop0-Pop1-123.csv:

As above, but considering only differences caused by mutations.

*****

All the pair distance files can increase quadratically in size along sample size. The total amount of .csv-cells is up to approximately *n^2 * m* for intra-population files and *n^2*2m* for interpopulation files, when *n* is the sample size and *m* is the amount of loci. Though often there is multiple non-unique pairs and thus less rows. Still, each file can contain billions of cells of data, and with close by documentation intervals, the disk space usage by these files might accumulate to relatively large amount. For example, with N=500, M=200 file size for only one interpopulation distance file could be up to about 50 MB. But increasing to N=1000, the resulting file size would already be up to around 200 MB. Though, often the file sizes are actually far smaller, as there is multiple pairs with same strain-ids and thus less rows.

Sequences	ISEQ	0 or 1 0 (0)	Save initial genome (1) or not (0)

InitialSequences123.txt:

Contains the initial sequences of the simulation.

	SEQI	Integer -20000 (Negative Integer)	Sequence saving: Generation interval
	SEQS	[0,1] 0.01	Sequence saving: Sample size as proportion of population

Sequences20000-123.csv:

Sequences of the bacteria in the sample. Indels are shown in parentheses, deletions as dashes and mutations as uppercase letters.

	Key	Value type Default value (Switch-off)	Function	Notes
Events	MUTD	Integer -20000 (Negative Integer)	Do mutation documenting with this interval, give 0 to use recombination documenting interval RECI	

MutationsPop0-123.txt:

Mutation counts along the simulation in given generation intervals for each population.

	RECI	Integer -20000 (Negative Integer)	Do recombination count documenting with this interval	

RecCounts20000-Pop0-123.txt:

Numbers of recombinations, categorized by Hamming distance between donor and recipient sequence.

	RECT	Integer 0 (0)	Do recombination event documenting with this Hamming distance threshold, i.e. record every event with HD over this threshold	Outputfile can be big. See ** below.

RecEvents123.txt:

Recombination events where the Hamming distance between donor and recipient sequences was greater than given threshold parameter. i.e. only those recombination events are documented that cause larger amount of base changes in recipient loci than the threshold.

	INDD	0 or 1 0 (0)	Do indel documenting (1) or not (0),	

Indels123.txt:

Numbers of insertion and deletion events, given for different lengths. Deletions shown as negative integers, insertions as positive.

	MIGD	0 or 1 0 (0)	Do migration documenting (1) or not (0)	Outputfile can be big. See ** below.

Migrations123.txt:

Documentation of each migration event. Each line consists of the migration amount in specific source->destination route in some generation. If MLST documentation is on, amounts of different STs that migrated are also given.

	Key	Value type Default value (Switch-off)	Function	Notes
	MICD	0 or 1 0 (0)	Do microepidemic documenting (1) or not (0)	Outputfile can be big. See ** below.

Microepidemics123.txt:

Documentation of each microepidemic event. Each line describes size, population and generation of specific microepidemic event. If MLST documentation is on, the ST of the microepidemic strain is also given.

**

The sizes of event documentation files naturally grow along generation amount and depend most of all on three different factors: the type of the event, rate of that event and if MLST documentation is being made. The results of these documentations are very dependant on specific settings, especially as the rates of the specific events are each influenced by several parameters, and thus general estimates are not provided here. Instead, example outputs are provided in the **CaseExample.zip** and **FeatureExample.zip**, and equations for rates shown in **ModelManual.pdf** can be used to estimate end results of these documentations in specific scenarios.

Furthermore, testing with shorter runs is a good way to begin if disk space is limited. The file sizes are quite linear along generation amount, but it should be noted that initially the files can have smaller sizes than should be expected when the variation in the populations sets in. This follows naturally from the simulated populations having less variety initially, so there is less of different STs and recombinations have smaller Hamming distances than further on.

MLST	MLST	0 or 1 0 (0)	Do MLST documenting (1) or not (0).	Memory usage can be intensive.

MLST123.txt:

Contains multilocus sequence types of all the bacteria in the end of simulation. Note that storing all different sequences for many loci is quite memory-intensive when running very big or long simulations.

	STAL	0 or 1 0 (0)	Save sequence-allele type pairs (1) or not (0).	Outputfile can be big.

MLSTsequences123.txt:

Contains all different sequences that have occurred during simulation and the corresponding identifying integers for multilocus sequence typing.