A quick start guide to BaM (Bayesian Modeling) configuration files

1 Introduction

BaM (Bayesian Modeling) is a computational code aimed at: (1) estimating the parameters of a model using uncertain data; and (2) generating predictions from the calibrated model along with quantified uncertainties. It can be considered as a generalization of the BaRatin (Bayesian Rating curve) approach[1], where the rating curve model can be replaced by "any" other model.

BaM.exe is controlled by a few configuration files in simple text format. They can hence be created, opened or modified by the user. This quick start guide provides a few basic explanations on the content of these configuration files. It is primarily intended for users of the BaRatinAGE[2] software who wish to take advantage of the additional functionalities provided by BaM. This quick start guide does assume some familiarity with the basic concepts behind BaRatinAGE and BaM (calibration of a model, basic Bayesian concepts, etc.).

2 General configuration files

2.1 General controller

The file *Config_BaM.txt* is the general controller and should be located in the same folder as BaM.exe. Its content is as follows:

Line 1. Path[3] of the workspace. All configuration files should be located in the workspace, and all result files will be written in the workspace.
Line 2. Configuration file defining the actions performed by BaM.exe (see section 2.2).
Line 3. Configuration file defining the model to be calibrated (see section 2.3.1).
Line 4. Configuration file containing extra information needed by the model (see section 2.3.2).
Line 5. Configuration file defining the calibration data (see section 3.1).
Line 6. Configuration file(s) defining the structural error model(s) (see section 3.2). As many files as the number of output variables of the model (for BaRatin: 1).
Line 7. Configuration file defining the properties of MCMC simulations (see section 3.3.1).
Line 8. Configuration file defining the post-treatment of MCMC simulations (see section 3.3.2).
Line 9. Configuration file defining the summary of MCMC simulations (see section 3.3.3).
Line 10. Configuration file defining the residual analysis (see section 3.4).
Line 11. Configuration file defining the prediction experiments (see section 4).

The configuration files highlighted in red are those that you will have to modify for each case study. For other configuration files, the default values given in the example files presented throughout this document may be suitable in many cases.

[1] Le Coz, J., B. Renard, L. Bonnifait, F. Branger, and R. Le Boursicaud (2014). *Combining hydraulic knowledge and uncertain gaugings in the estimation of hydrometric rating curves: A Bayesian approach*. J. Hydrol.
[2] https://forge.irstea.fr/projects/baratinage_v2/files
[3] In the whole document, all paths can be written either as absolute paths or as relative paths with respect to the location of BaM.exe.

Config_BaM.txt

```
"BaM_BaRatin\"                                ! workspace
"Config_RunOptions.txt"                       ! run options
"Config_Model.txt"                            ! model
"Config_ControlMatrix.txt"                    ! xtra model information
"Config_Data.txt"                             ! data
"Config_RemnantSigma.txt"                     ! structural errors
"Config_MCMC.txt"                             ! MCMC
"Config_Cooking.txt"                          ! cooking of MCMC samples
"Config_Summary.txt"                          ! summary of MCMC samples
"Config_Residuals.txt"                        ! residual diagnostics
"Config_Pred_Master.txt"                      ! prediction experiments
```

2.2 Actions performed by BaM

Config_RunOptions.txt defines the actions performed by BaM.exe. Each line is a .true./.false. answer to the following questions:

Line 1. Perform MCMC simulations?
Line 2. Perform MCMC summary?
Line 3. Perform residual analysis?
Line 4. Perform prediction and uncertainty propagation experiments? (see section 4)

Config_RunOptions.txt

```
.true.    ! Do MCMC?
.true.    ! Do MCMC summary?
.true.    ! Do Residual diagnostics?
.true.    ! Do Predictions?
```

Typically, BaM.exe can be used in two successive steps: (1) parameter estimation (.true. for the first 3 items, .false. for the last one); (2) prediction / uncertainty quantification (.false. for the first 3 items, .true. for the last one).

2.3 Configuration of the model

A key difference between BaRatinAGE and BaM is that the latter is not restricted to rating curve models. However, we explain below the content of the configuration files for the particular BaRatin rating curve model. For other models, see section 5.3.

2.3.1 Model configuration and priors

The content of *Config_Model.txt* is as follows:

Line 1. Model ID.
Line 2. Number of input variables (for BaRatin: 1, stage).
Line 3. Number of output variables (for BaRatin: 1, discharge).
Line 4. Number of parameters (for BaRatin: depends on the hydraulic configuration, 3*(number of hydraulic controls)).
Line 5. For each parameter, a 4-line block as follows:

a. Parameter name.
b. Initial value.
c. Prior distribution (see Table 1 for a list of available distributions).
d. Prior parameters (comma-separated if several).

For BaRatin, the parameters should be specified by following the *k*-*a*-*c* pattern for each successive hydraulic control (*k* = activation stage, *a* = coefficient, *c* = exponent), as illustrated below. Note that an alternative version of BaRatin, named BaRatinBAC, uses the *b*-*a*-*c* pattern instead (*b* = offset). This alternative version is useful for modeling changing rating curves using the Stage-Period-Discharge approach[4].

Config_Model.txt

```
"BaRatin"                  ! Model ID
1                          ! nX: number of input variables
1                          ! nY: number of output variables
6                          ! nPar: number of parameters theta
"k1"                       ! Parameter Name
-0.5                       ! Initial guess
'Gaussian'                 ! Prior distribution
-0.5,0.25                  ! Prior parameters
"a1"                       ! Parameter Name
53                         ! Initial guess
'Gaussian'                 ! Prior distribution
53,10                      ! Prior parameters
"c1"                       ! Parameter Name
1.5                        ! Initial guess
'Gaussian'                 ! Prior distribution
1.5,0.025                  ! Prior parameters
"k2"                       ! Parameter Name
1.5                        ! Initial guess
'Gaussian'                 ! Prior distribution
1.5,0.5                    ! Prior parameters
"a2"                       ! Parameter Name
144                        ! Initial guess
'Gaussian'                 ! Prior distribution
144,45                     ! Prior parameters
"c2"                       ! Parameter Name
1.67                       ! Initial guess
'Gaussian'                 ! Prior distribution
1.67,0.025                 ! Prior parameters
```

[4] Mansanarez, V., B. Renard, J. Le Coz, M. Lang, and M. Darienzo (2019). *Shift happens! Adjusting stage-discharge rating curves to morphological changes at known times*. Submitted to Water Resour. Res.

Note that two additional advanced features may be of interest:

1. Allowing some parameters to vary: see section 5.1.
2. Specifying a correlated joint prior distribution: see section 5.2.

2.3.2 Extra information needed by the model

For BaRatin, the file *Config_ControlMatrix.txt* is used to specify the control matrix (see BaRatinAGE documentation). In the example below, two hydraulic controls are succeeding each other.

Config_ControlMatrix.txt
```
1 0
0 1
```

Note that for the BaRatinBAC version, the upper range of the domain of validity of the rating curve also needs to be specified as shown below.

Config_ControlMatrix.txt
```
1 0
0 1
8.    ! hmax [m]
```

3 Configuration files for model calibration

3.1 Configuration of calibration data

Config_Data.txt allows specifying the calibration data, and contains the following information:

Line 1. Path to the data file. The data file has to be organized in columns separated by tabulations or spaces.
Line 2. Number of header lines.
Line 3. Number of observations in the data file (i.e., number of lines, excluding headers).
Line 4. Number of columns in the data file.
Line 5. Column(s) containing input variable(s).
Line 6. Column(s) containing the standard deviations of random input errors. Use value 0 for a no-error assumption.
Line 7. Column(s) containing the standard deviations of systematic input errors. Use value 0 for a no-error assumption.
Line 8. Column(s) containing the index of systematic input errors. Use value 0 for a no-error assumption.
Line 9. Column(s) containing output variable(s).
Line 10. Column(s) containing the standard deviations of random output errors. Use value 0 for a no-error assumption.
Line 11. Column(s) containing the standard deviations of systematic output errors. Use value 0 for a no-error assumption.
Line 12. Column(s) containing the index of systematic output errors. Use value 0 for a no-error assumption.

The lines highlighted in orange correspond to the handling of systematic errors in observed inputs / outputs (stages / discharges for BaRatin). This is a new feature of BaM and at this stage it is still undocumented and highly experimental. So we strongly recommend deactivating systematic errors by putting zeros, as in the example below.

Config_Data.txt
```
'BaM_BaRatin\data\Gaugings.txt' ! path to data file
1                    ! number of header lines
38                   ! Nobs
8                    ! number of columns in the data file
1                    ! columns for X (observed inputs)
0                    ! columns for Xu (sdev of RANDOM errors in X)
0                    ! columns for Xb (sdev of SYSTEMATIC errors in X)
0                    ! columns for Xb_indx (index of SYSTEMATIC errors in X)
5                    ! columns for Y (observed outputs)
6                    ! columns for Yu (sdev of RANDOM errors in Y)
0                    ! columns for Yb (sdev of SYSTEMATIC errors in Y)
0                    ! columns for Yb_indx (index of SYSTEMATIC errors in Y)
```

3.2 Model for structural errors

The file *Config_RemnantSigma.txt* allows specifying the model for structural errors (a.k.a model errors or remnant errors). The default values used below may be suitable in many cases. The file contains the following information:

Line 1. ID of the structural error model (`'Constant'`, `'Linear'` `[recommended]`, `'Exponential'` ou `'Gaussian'`).
Line 2. Number of parameters to be inferred.
Line 3. For each parameter, a 4-line block as follows:
  a. Parameter name.
  b. Initial value.
  c. Prior distribution.
  d. Prior parameters.

Config_ RemnantSigma.txt
```
'Linear'                ! Function f used in sdev=f(Ysim)
2                       ! Number of parameters gamma for f
"intercept"             ! Parameter Name
1.                      ! Initial Guess
'Uniform'               ! Prior distribution
0,1000                  ! Prior parameters
"slope"                 ! Parameter Name
0.1                     ! Initial Guess
'Uniform'               ! Prior distribution
0,1000                  ! Prior parameters
```

3.3 Configuration of MCMC simulations

Three configuration files are used here to specify: (i) the properties of MCMC simulations (*Config_MCMC.txt*); (ii) its post-processing (*Config_Cooking.txt*); and (iii) its summarization (*Config_Summary.txt*). Understanding the content of these files requires some familiarity with the

MCMC algorithm used in BaM: see the documentation of the BaRatinAGE software or Renard et al. (2006)[5].

The default values used in the examples below may be suitable for many cases. The main thing you may want to change here is to decrease the number of MCMC simulations for fast preliminary runs: this can be achieved by decreasing `Ncycles` to 100 and `Nslim` to 10.

3.3.1 MCMC simulations

The content of *Config_MCMC.txt* is as follows:

Line 1. Name of the file where MCMC simulations will be stored.
Line 2. Number of iterations between each adaptation.
Line 3. Number of adaptation cycles.
Line 4. Minimal acceptance rate.
Line 5. Maximal acceptance rate.
Line 6. Factor for decreasing the jump standard deviation.
Line 7. Factor for increasing the jump standard deviation.
Line 8. Option for specifying the initial jump standard deviation.
Line 9. Cosmetic line (unused).
Line 10. Factor if option=0.
Line 11. Individual factors for $\boldsymbol{\theta}$ if option=1.
Line 12. Individual factors for $\boldsymbol{\gamma}$ if option=1.

Config_MCMC.txt

```
"Results_MCMC.txt" ! File name
100                ! NAdapt
1000               ! Ncycles
0.1                ! MinMoveRate
0.5                ! MaxMoveRate
0.9                ! DownMult
1.1                ! UpMult
0                  ! initial Std of the jump distribution option
****    DEFINITION OF INITIAL JUMP STD     **** ! Cosmetics
0.1                ! common MultFactor if option=0
                   ! individual MultFactor for theta if option=1
                   ! individual MultFactor for gamma if option=1
```

3.3.2 MCMC cooking

Postprocessing ("cooking") of raw MCMC simulations aims at decreasing the number of simulations that will be used in subsequent analyses (MCMC summary, residual analysis, predictions). This may avoid computing time and storage issues. The content of *Config_Cooking.txt* is as follows:

Line 1. File names where cooked MCMC simulations will be stored.
Line 2. Burn factor: the first Burn*Nsim iterations are discarded.
Line 3. Slim factor: after burning, only one iteration every Nslim is kept.

---

[5] Renard, B., V. Garreta, and M. Lang (2006). *An application of Bayesian analysis and MCMC methods to the estimation of a regional trend in annual maxima*. Water Resources Research.

Config_Cooking.txt

```
"Results_MCMC_Cooked.txt"    ! Result file
0.5                          ! BurnFactor
100                          ! Nslim
```

3.3.3 MCMC summary

MCMC summarization aims at providing a few basic descriptive statistics of the MCMC simulations (for each inferred parameter: mean, median, standard deviation, quartiles, max-posterior value, etc.). The file Config_Summary.txt contains a single line specifying the name of the file where the MCMC summary will be stored.

Config_Summary.txt

```
"Results_Summary.txt"             ! Result file
```

3.4 Residual Analysis

The associated configuration file contains a single line specifying the name of the file where the results of the residual analysis will be stored.

Config_Residuals.txt

```
"Results_Residuals.txt"            ! Result file
```

4 Configuration files for prediction and uncertainty propagation experiments

In BaM, all "prediction" experiments are performed using a single unified framework. For BaRatin, such prediction experiments may be used to compute:

- The prior rating curve (i.e. many rating curves representing the prior uncertainty, obtained by sampling rating curve parameters in their prior distribution).
- The maxpost rating curve (i.e. a single rating curve with parameters fixed at their maximum-posterior values).
- Partial or total uncertainties around the rating curve (i.e. many rating curves representing the uncertainty in discharge induced by parametric and/or structural uncertainties).
- A maxpost discharge time series (i.e. a single discharge time series computed by transforming a time series of stages with the maxpost rating curve).
- Partial or total uncertainties around the discharge time series (i.e. many discharge time series representing the uncertainty in discharge induced by stage and/or parametric and/or structural uncertainties).

4.1 Master prediction controller

The file *Config_PredMaster.txt* lists the prediction experiments that will be performed:

Line 1. Number of prediction experiments to be performed.
Line 2. Configuration file for experiment 1.
Line 3. Configuration file for experiment 2.
Line 4. etc.

Config_PredMaster.txt

```
7                            ! Number of prediction experiments
'Config_Pred_Prior.txt'      ! Exp. 1: prior rating curve (RC)
'Config_Pred_RCMaxpost.txt'  ! Exp. 2: Maxpost RC
'Config_Pred_RCParamU.txt'   ! Exp. 3: RC w/ parametric uncertainty (U) only
'Config_Pred_RCTotalU.txt'   ! Exp. 4: RC with total U (par. + struct.)
'Config_Pred_Maxpost.txt'    ! Exp. 5: Maxpost Q(t) time series
'Config_Pred_hU.txt'         ! Exp. 6: Q(t) with stage U only
'Config_Pred_TotalU.txt'     ! Exp. 7: Q(t) with total U (stage+par+struct)
```

4.2 Prediction configuration files

For each prediction experiment, a configuration file containing the following information needs to be written:

Line 1. **Path to input values**: for a rating curve, this will be a grid of stage values. For a discharge time series, this will be a time series of stages. Moreover, it is possible to provide several “spaghettis”, representing the uncertainty in these input values. In this case, each column in the input values file represents one spaghetti. Using a single column leads to ignoring stage uncertainty.
Line 2. **Number of observations for each spaghetti**: number of lines in the input values file.
Line 3. **Number of spaghettis**: number of columns in the input values file.
Line 4. **Propagation of parametric uncertainty?**
Line 5. **Propagation of structural uncertainty?**
Line 6. **Number of simulations for a prior prediction.** A positive number is interpreted as a request to sample the RC parameters in their prior distribution, and the provided number gives the number of simulations. If a negative number is provided, the RC parameters are sampled in their posterior distribution, i.e. MCMC simulations are used.
Line 7. **Result file for output spaghettis.** Each column contains one spaghetti for discharge.
Line 8. **Transpose spaghettis?** By default, spaghettis are written horizontally (row-wise) in the result file (for some algorithmic reason). Transposing them will bring them back vertical.
Line 9. **Compute uncertainty envelops?**
Line 10. **Result file for uncertainty envelops**.
Line 11. **Print a progress bar in console during computation?**
Line 12. **Predict state variables**?
Line 13. **Result file for state variable spaghettis.**
Line 14. **Transpose state variables spaghettis?**
Line 15. **Compute uncertainty envelops for state variables?**
Line 16. **Result file for uncertainty envelops for state variables**.

The lines highlighted in orange above are only useful for models having state variables, which is not the case for BaRatin: you can leave these lines blank.

The example below shows the configuration file for the seventh prediction experiment (compute discharge time series with total uncertainty). Configuration files for the other six experiments are provided in the attached example folder.

Config_Pred_TotalU.txt

```
'BaM_BaRatin\data\Ht_noisy.txt' ! Input file (spag. of stage time series)
250                             ! Nobs
100                             ! Nspag, number of spaghettis
.true.                          ! Propagate parametric uncertainty?
.true.                          ! Propagate structural uncertainty?
-1                              ! Nsim[prior]. If <=0: posterior sampling
'Qt_TotalU.spag'                ! File containing Q spaghettis
.true.                          ! Transpose spag file?
.true.                          ! Create uncertainty envelops?
'Qt_TotalU.env'                 ! Name of envelop file
.true.                          ! Print progress in console?
.false.                         ! Do state prediction?
```

5 Advanced features

5.1 Varying parameters

It is possible to assume that some of the parameters specified in the model configuration file (section 2.3.1) are not static for the entire calibration dataset. This can be specified by:

1. Using 'VAR' as a prior distribution.
2. Writing the name of a VAR-parameter configuration file (within quotes) in place of the prior parameters.

An example of such a VAR-parameter configuration file is given below. Its contents is the following:

Line 1. Number of distinct values taken by the VAR-parameter in the calibration dataset. If there are 5 distinct values, 5 additional parameters will have to be inferred.

Line 2. Column in the data file containing the index identifying which VAR-parameter value is associated with each calibration data.

Line 3. For each VAR-parameter value, a 4-line parameter block:

   a. Parameter name.
   b. Initial value.
   c. Prior distribution (see Table 1 for a list of available distributions).
   d. Prior parameters (comma-separated if several).

VAR-parameters can be used for instance for modeling changing rating curves (see Mansanarez et al. (2019)[4]).

Config_b1_VAR.txt

```
5           ! Number of distinct values
4           ! Column where the index is written in data file
"b1_1"      ! Parameter name
-0.6        ! Initial guess
"Gaussian"  ! Prior distribution
-0.6,0.5    ! Prior parameters
"b1_2"      ! Parameter name
-0.6        ! Initial guess
"Gaussian"  ! Prior distribution
-0.6,0.87   ! Prior parameters
"b1_3"      ! Parameter name
-0.6        ! Initial guess
"Gaussian"  ! Prior distribution
-0.6,1.12   ! Prior parameters
"b1_4"      ! Parameter name
-0.6        ! Initial guess
"Gaussian"  ! Prior distribution
-0.6,1.33   ! Prior parameters
"b1_5"      ! Parameter name
-0.6        ! Initial guess
"Gaussian"  ! Prior distribution
-0.6,1.50   ! Prior parameters
```

The use of varying parameters is not yet fully supported in BaM, and the following limitations hold:

1. Prediction with varying parameters is not supported (only estimation is feasible at the moment).
2. Varying parameters are not yet allowed with dynamic models such as hydrologic models (e.g. GR4J).

5.2 Prior correlations

The marginal prior distributions specified in the model configuration file (section 2.3.1) are assumed independent by default: the joint prior pdf is therefore obtained by multiplying marginal prior pdfs. It is possible to consider correlated prior distributions by using a Gaussian copula[6] to non-independently combine the marginal priors into a joint prior. This requires specifying a correlation matrix for all inferred parameters, which has to be written in a file named *PriorCorrelation.txt* (warning: file name should be respected and is case-sensitive).

The example given below corresponds to the Meyras case study of Mansanarez et al. (2019)[4]. The underlying model is a BaRatinBAC rating curve with 3 controls; offsets b_1 and b_2 are varying across 5 stability periods. The vector of inferred parameters is therefore $\left(b_1^{(1)},...,b_1^{(5)},a_1,c_1,b_2^{(1)},...,b_2^{(5)},a_2,c_2,b_3,a_3,c_3,\gamma_1,\gamma_2\right)$ and can be interpreted as the line/column header for the correlation matrix below.

[6] Renard, B., and M. Lang (2007). *Use of a Gaussian copula for multivariate extreme value analysis: some case studies in hydrology*. Adv. Water Resour.

PriorCorrelation.txt

```
1.00 0.58 0.45 0.38 0.33 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.58 1.00 0.77 0.65 0.58 0.00 0.00 0.00 0.41 0.33 0.29 0.26 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.45 0.77 1.00 0.85 0.75 0.00 0.00 0.00 0.32 0.52 0.45 0.40 0.00 0.00 0.01 0.00 0.00 0.00 0.00
0.38 0.65 0.85 1.00 0.88 0.00 0.00 0.00 0.27 0.44 0.57 0.51 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.33 0.58 0.75 0.88 1.00 0.00 0.00 0.00 0.24 0.39 0.51 0.60 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.70 0.58 0.50 0.45 0.00 0.01 0.00 0.00 0.00 0.00 0.00
0.00 0.41 0.32 0.27 0.24 0.00 0.00 0.70 1.00 0.82 0.71 0.63 0.00 0.00 0.01 0.00 0.00 0.00 0.00
0.00 0.33 0.52 0.44 0.39 0.00 0.00 0.58 0.82 1.00 0.87 0.78 0.00 0.00 0.01 0.00 0.00 0.00 0.00
0.00 0.29 0.45 0.57 0.51 0.00 0.00 0.50 0.71 0.87 1.00 0.90 0.00 0.00 0.01 0.00 0.00 0.00 0.00
0.00 0.26 0.40 0.51 0.60 0.00 0.00 0.45 0.63 0.78 0.90 1.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.01 0.01 0.01 0.01 0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00
```

Note that if no file named *PriorCorrelation.txt* is found in the workspace, the prior will be assumed independent, with no warning message. Consequently, beware of typos in the name of the file!

5.3 Using other models

The following models are currently implemented in BaM:

1. BaRatin
2. BaRatinBAC
3. Stage-Fall-Discharge rating curve model (e.g. twin gauges)
4. Stage-Gradient-Discharge rating curve model (e.g. hysteresis)
5. Vegetation-influenced rating curve models
6. Rating curve models using satellite (e.g. SWOT) information
7. Tide-influenced rating curve model
8. General Linear Model (in the statistical sense – linear w.r.t. parameters)
9. Mixture model (chemistry)
10. Orthorectification model
11. Hydrological model GR4J
12. Sediment transport models
13. Suspended load model

Unfortunately, models other than BaRatin / BaRatinBAC are not properly documented yet. This work is currently in progress.

6 Appendix

Table 1. Available prior distributions.

Distribution	ID	# par.	Parameters
Gaussian (a.k.a Normal)	`'Gaussian'`	2	mean, standard deviation
Uniform	`'Uniform'`	2	Lower bound, higher bound
Triangle	`'Triangle'`	3	Peak, lower bound, higher bound
Log-normal	`'LogNormal'`	2	Log-mean, log-standard-deviation
Exponential	`'Exponential'`	2	Threshold, scale
Generalized Pareto	`'GPD'`	3	Threshold, scale, shape
Gumbel	`'Gumbel'`	2	Location, scale
Generalized Extreme Value	`'GEV'`	3	Location, scale, shape
Inverse Chi-2	`'Inverse Chi2'`	2	Degrees of freedom, scale
Pearson III	`'PearsonIII'`	3	Location, scale, shape
Geometric	`'Geometric'`	1	Success probability
Poisson	`'Poisson'`	1	Rate
Binomial	`'Binomial'`	2	Success probability, number of trials
Negative binomial	`'NegBinomial'`	2	Success probability, number of failures
Flat prior*	`'FlatPrior'`	0	/
Positive flat prior*	`'FlatPrior+'`	0	/
Negative flat prior*	`'FlatPrior-'`	0	/
Fixed parameter (a.k.a Dirac)**	`'FIX'`	0	/
Varying parameter***	`'VAR'`	0	Name of the corresponding configuration file

* improper distributions with constant density over $\mathbb{R}$ (`'FlatPrior'`), $\mathbb{R}^+$ (`'FlatPrior+'`) or $\mathbb{R}^-$ (`'FlatPrior-'`). Mostly used to denote lack of knowledge.

** pseudo-distribution used to force a parameter at a fixed value (the parameter is therefore not inferred).

** pseudo-distribution used to denote a parameter whose value may vary (several values of the parameter are therefore inferred): see section 5.1.