Parameter Balancing Version 2.0

Manual

Content

1. What is parameter balancing?

Parameter balancing is a method to determine consistent parameter sets for kinetic metabolic models. Experimentally measured values, when directly inserted into a model, are likely to yield incomplete and inconsistent parameter sets. Balanced parameter sets, which are complete and consistent, are computed from kinetic constants and other data collected from experiments or the literature, based on constraints between biochemical quantities and assumptions about typical ranges, represented by prior values and bounds.

The details of the mathematical background of parameter balancing can be received from the corresponding publication “Parameter Balancing in Kinetic Models of Cell Metabolism”, published in The Journal of Physical Chemistry B (2010, DOI: 10.1021/jp108764b). Furthermore, the work “Systematic Construction of Kinetic Models from Genome-Scale Metabolic Networks”, published in PLOS one (2013, DOI: 10.1371/journal.pone.0079195), exemplifies how parameter balancing can be embedded in a metabolic modelling workflow.

Parameter balancing can be employed in 4 ways:

1. **On our website** www.parameterbalancing.net, users can employ an online interface of the parameter balancing tool. (see Chapters 2-3).
2. It can be installed as **Python package** (see Chapter 4-5).
3. It can be employed as a **web2py standalone application** (see Chapter 4-5).
4. It can be used as a **commandline tool** (see Chapter 4-5).

This manual is supposed to guide users through the details of the four different parameter balancing employments. If you do not find your questions answered in this manual or the above publications, please contact Wolfram Liebermeister (wolfram.liebermeister@gmail.com) and Timo Lubitz (timo.lubitz@gmail.com).

2. Input data

For the online parameter balancing the user can employ SBML (www.sbml.org) and SBtab (www.sbtab.net) files as input format.

2.1 SBML

The user will need at least a mathematical model in the SBML format to perform online parameter balancing. This format is widespread in the mathematical modelling community and can be created with numerous tools, such as Copasi (www.copasi.org). The SBML model should contain species and reactions. The validity of an SBML model should be checked before parameter balancing on the official validation page http://sbml.org/Facilities/Validator.

2.2 SBtab

SBtab files can be created with every given text and spreadsheet editor. They are tab-separated table files (.tsv) which can hold very diverse content. We are using SBtab as input format for several different data kinds. A specification on SBtab and an online validation tool can be found on the official project homepage www.sbtab.net.

2.2.1 SBtab QuantityData (kinetic parameter file)

SBtab files of the table type QuantityData hold data on kinetic parameter values. The user can provide numeric values for the entities (species and reactions) of the SBML model. To correctly link the provided parameter values of the SBtab file to the entities of the SBML model it is of utmost importance to include the columns *!SBML:species:id* and *!SBML:reaction:id*. The entries of these columns must comply with the IDs (attention: **not** the names) of the SBML model entities.

Furthermore, the SBtab file urgently requires the columns *!QuantityType*, *Mean*, *!Std*, and *!Unit* for the provision of the parameter type, the numeric parameter value, its standard deviation, and the corresponding unit, respectively. The parameter types include equilibrium constants, concentration, enzyme concentration, rate constants, chemical potentials, Michaelis constants and more. More details can be found in the SBtab specification on www.sbtab.net and example files on www.parameterbalancing.net/pb/static/downloads.html.

The user must not provide a SBtab QuantityData file, this is optional. But the provision of the balancing with kinetic parameter values is crucial for its accuracy. If no SBtab parameter file is given, the parameter balancing will rely its computation solely on the prior distributions of the parameter types, which is very imprecise.

2.2.3 SBtab QuantityInfo (prior distribution file)

Parameter balancing is a Bayesian estimation which requires prior distributions for all included parameter types. The distributions are represented as mean values with corresponding standard deviations. We use distributions which are common for the individual parameter types and are provided with very broad standard deviations to not put too strict limitations on the estimation.

The column "PriorMedian" contains the median values of the parameter priors. For parameters with "original scaling", the median value coincides with the mean value. For parameters with "logarithmic scaling", the median value does not coincide with the parameter's own mean value, but is given by exp(x), where x is the mean value of the parameter's natural logarithm.

Depending on the type of parameter, the prior standard deviation is either defined in the column "PriorStd" or in the column "Log10PriorStd". For parameters with "original scaling", the column "PriorStd" contains the prior standard deviation of the parameter itself. For parameters with "logarithmic scaling", the column "PriorStd" contains the prior standard deviation of the parameter's decadic logarithm. For example, an entry of 2 would mean that a typical deviations from the median value is about 2 orders of magnitude.

Experienced users with special focus or abnormal reaction systems might want to change the prior distributions to serve their individual requirements. This can be done by providing an SBtab QuantityInfo file. The easiest way to create such a file is to download the default file from the website on www.parameterbalancing.net/pb/static/downloads.html and change the *!PriorMode* and/or *!PriorStd* columns.

As implied before, the provision of such a file is only optional and not recommended for inexperienced users.

2.2.4 SBtab PbConfig (parameter balancing options file)

Online parameter balancing is a complex process with many optional configurations. These can be changed by using a configuration file of the SBtab table type PbConfig. Again, the default configuration file can be downloaded on www.parameterbalancing.net/pb/static/downloads.html. The available configuration options are

- **use_pseudos** (Bool; *True* or *False*): usage of pseudo values as substitute for missing parameter values to improve balancing (see PLOS one publication for details)

- **pH** (float; 1-14): pH value that allows parameter balancing for correction of parameter values that do not correspond to the system's target pH value (see JoPC publication for details)

- **Temperature** (float, in Kelvin): temperature that allows parameter balancing for correction of parameter values that do not correspond to the system's target temperature (see JoPC publication for details)

- **overwrite_kinetics** (Bool; *True* or *False*): should existing kinetic laws in the SBML file be overwritten?

- **cell_volume** (float; in µm^3): volume of the cell

- **parametrisation** (string; *hal* or *cat* or *weg*): different parametrisation types for the convenience kinetics. See "Bringing metabolic networks to life" by Wolfram Liebermeister for details

- **enzyme_prefactor** (Bool; *True* or *False*): should a prefactor for the enzyme concentration be set in the rate law?

- **default_inhibition** (string; *complete*, *partial*, or *specific*): type of enzyme inhibition

- **default_activation** (string; *complete*, *partial*, or *specific*): type of enzyme activation

- **model_name** (string): name that is used for the output files

- **boundary_values** (string; *ignore*): ignore boundary values (temporary feature)

2.2.5 SBtab Document (multiple SBtab tables in one table file)

All of the above SBtab files can be provided in one single spreadsheet to ease and fasten up the online parameter balancing. Moreover, also the SBML model can be translated to SBtab (on www.sbtab.net/convert) and be added to the SBtab file. In sum, the user only has to upload a single file for online balancing. Here again, example files for whole systems (model, parameters, prior file, config file) can be found on the download page: www.parameterbalancing.net/pb/static/downloads.html

3. Online workflow

Online parameter balancing can be performed on

www.parameterbalancing.net/pb/default/balancing.html

and there are two options for the user: SBML + SBtab input and SBtab only input.

3.1 SBML + SBtab

The classic balancing requires at least an SBML model file to be uploaded. As soon as it is uploaded, a large ->Balance Parameters<- button appears on the bottom of the column. By clicking it, the SBML model is processed and parameter balancing is performed on grounds of the default prior distributions and default options. This is easy and fast, but for accuracy the user should provide more information.

Optionally, the user can provide an SBtab data file (see Section 2.2.1 of this manual) with experimental values. The more data are provided, the more accurate the balancing will get.

Also, the user can upload individual prior distributions for the different parameter types (see Section 2.2.3) and a parameter balancing options file (see Section 2.2.4). Example files for all table types are available on our download page.

3.2 Single SBtab file

Instead of uploading single files for SBML model and optionally parameter, prior distribution, and options SBtabs, the user can also simply upload a single SBtab file which holds all required information needed for parameter balancing. The SBML file can first be translated to SBtab (on www.sbtab.net/convert) and all other SBtab files can simply be added to the same file (see Section 2.2.5 and our online download section).

Regardless of choosing Option 1 or 2, after clicking the ->Balance Parameters<- button the user is referred to the results page. Here, they can download (i) the SBML model with balanced parameters and inserted convenience kinetics (as long as this option is not switched off with the options SBtab), (ii) an SBtab parameter file with all balanced parameters, and an SBtab model data file, which holds the model information, balanced parameter values,

prior file, and configuration file all in one SBtab. The files can be viewed online, downloaded, or removed.

4. Description of the output SBtab file

The output SBtab file holds the balanced parameters for each model entity. However, there are some details that motivate a closer look.

The parameter balancing algorithm computes a joint posterior distribution of all model parameters. This posterior distribution is multivariate Gaussian (where parameters with "logarithmic scaling" (see prior table) are represented by their natural logarithms). However, constraints on the parameters (e.g., individual upper and lower bounds) will restrict this distribution to a convex polytope. It may even happen that the mean value of our Gaussian distribution is outside this polytope and not a feasible point. To provide information about this, we characterise the posterior by different numbers in the output file:

(i) The columns are called "Mean" and "Std" contain the mean value and standard deviation of the UNCONSTRAINED posterior. In the case of parameters with "original scaling", the values refer to the Gaussian distribution itself; in the case of parameters with "logarithmic scaling", the values refer to the corresponding log-normal distribution (of the non-logarithmic parameter values).

(ii) The columns "MeanLn" and "StdLn" exist only for parameters with "logarithmic scaling". They describe the mean value and standard deviation of the underlying Gaussian distribution (for the parameters' natural logarithms).

(iii) The column "Median" contains, for every parameter, the UNCONSTRAINED median of the parameter itself. For parameters with original scaling, this is simply the mean value; for parameters with logarithmic scaling, this is exp(x), where x is the mean value of the underlying Gaussian distribution (of the parameter's natural logarithm).

(iv) The column "ConstrainedMode" contains the parameter value in the CONSTRAINED mode of the distribution. For parameters with original scaling, this is the point in the feasible polytope where the posterior has its maximal value; for parameters with logarithmic scaling, this is exp(x), where x is the point in the feasible polytope for logarithmic values where the posterior has its maximal value.

When inserted parameter values into a model, it is advisable to use the "ConstrainedMode" entries as model parameters, because any of the other values might violate the constraints. If all constraints are inactive, the standard deviations of the Gaussian distribution can be used to characterise the posterior. If some constraints are active, then the standard deviations may not be meaningful, and sampling from the posterior is required.

5. Download and installation instructions

Parameter balancing can be used via 4 different channels: (i) The online interface described above, (ii) as a Python package, (iii) as a web2py application for a standalone version of the online balancing interface, and (iv) as simple Python scripts that can be employed in the commandline.

4.1 The Online Interface

For the usage of the online interface, a user needs a browser of any choice, an internet connection, and the input files. All further details are explained in Sections 2 and 3.

4.2 The Python Package

To install Parameter Balancing as a Python3 package, first of all you need Python3. Next, you will need the pip3 installer. You can find information on how to achieve this installer here: https://pip.pypa.io/en/stable/installing/. Afterwards, install Parameter Balancing by typing in your command line:

> sudo pip3 install pbalancing

This will also install libsbml and tablib on your computer if these libraries are missing. You can now employ Parameter Balancing as a Python3 package by, e.g., writing a Python script such as

```
from pbalancing import parameter_balancing
parameter_balancing.parameter_balancing_wrapper('model.xml')
```

In this example case, 'model.xml' is the file name of an SBML model. Further optional arguments are an SBtab parameter file, an SBtab prior distribution file, and an SBtab configuration file. You will find information on all these file types in the online Download area www.parameterbalancing.net/pb/static/download.html.

4.3 The web2py Standalone Version

The parameter balancing interface can also be used as a standalone version. This requires initially the download of a web2py server from

http://www.web2py.com/examples/default/download

Next, the user needs to download the parameter balancing application from

https://bitbucket.org/tlubitz/pb.

The application folder pb needs to be moved to the web2py directory, namely to web2py/applications/pb. Now the web2py server can be initiated (see www.web2py.com for details on your specific operating system) and the parameter balancing standalone can be accessed on any browser calling

http://127.0.0.1:8000/pb/

The benefit of this version is that the user will not require a running internet connection and can still run the balancing GUI in their browser. Furthermore, they can make alterations on the parameter balancing source code in the pb directory, assuming the user is familiar with Python3 and web2py.

The prerequisites for running this interface are Python3 and the libsbml library. The former can be downloaded from

https://www.python.org/downloads/release/python-2713/

and the latter from

http://sbml.org/Software/libSBML/Downloading_libSBML.

If the user wants to contribute to the parameter balancing project, a bitbucket account is required, the pb repository needs to be cloned, and changes can be pushed to the repository on request.

4.4 Python Scripts

To run parameter balancing as a commandline tool, the package needs to be installed as explained in 4.2. Then, it can be executed in the commandline as follows:

python3 -m pbalancing.parameter_balancing model.xml

where model.xml corresponds to the path to your SBML model.

6. Parameter balancing in bigger workflows

As mentioned in Section 6.1, parameter balancing can be embedded in bigger workflows of mathematical modelling. An extensive example is given in “Systematic Construction of Kinetic Models from Genome-Scale Metabolic Networks”, published in PLOS one (2013, DOI: 10.1371/journal.pone.0079195). Here, a geometric FBA is employed as first step, but this is not necessarily required – any FBA that generates loopless flux distributions is appropriate. Similarly, other steps of the presented workflow can be replaced by other modelling techniques, which yield the same context as the methods used.

Furthermore, in parameter balancing we adhere to common standard file formats such as SBML and SBtab to ensure an easy interface for embedding.

Frequently asked questions

What is parameter balancing?

Parameter balancing is a way to determine consistent parameter sets for kinetic models of metabolism. Inserting experimentally measured values directly into a model will probably yield incomplete or inconsistent parameter sets, violating the thermodynamic Haldane relationships. Balanced parameter sets avoid this problem. They are computed based on kinetic constants and other data collected from experiments or the literature, but also based on known constraints between biochemical quantities and on assumptions about typical ranges, represented by prior values and bounds.

How can I run parameter balancing?

After preparing your model and data files, you can run parameter balancing interactively here. The workflow is described here. If you prefer working in the command line, or if you would like to include parameter balancing in your programs, you may use our code for Python and Matlab.

Which parameters of a metabolic model can be balanced?

In general, parameter balancing concerns the kinetic and thermodynamic constants in kinetic metabolic models. It can also cover metabolite concentrations, chemical potentials, and reaction Gibbs free energies (or, equivalently, "reaction affinities" or "driving forces"). Metabolic fluxes cannot be balanced, but they can be included in the analysis (this is described below). There are different typical application cases:

- Kinetic constants, where equilibrium constants are fixed and given
  This can be done separately for each individual reaction. If a network is large and equilibrium constants can be predefined (for instance, by parameter balancing), we suggest to split the network into single reactions and to run parameter balancing separately for every reaction.
- Kinetic constants and equilibrium constants in a network
- Equilibrium constants and concentrations in a network
- Equilibrium constants, kinetic constants, and concentrations in a network

What input data are needed?

Parameter balancing employs SBML (Systems Biology Markup Language) files or SBtab files for model structures and SBtab table files for data and

configuration files. Parameter balancing imports a model (SBML, obligatory) and a data table (SBtab, optional) with experimental data values. Furthermore, tables with information on the prior distributions and balancing options are possible. Parameter balancing produces tables with balanced parameters (SBtab) and a model with rate laws and balanced parameters included (SBML). Please prepare your SBtab files as described below. The validity of these files can be checked on the SBtab website. The quantities described in your data table will be linked to elements of the SBML model, via the entries in the columns !SBML:reaction:id and !SBML:species:id. The IDs in these columns must match the IDs chosen in the SBML file.

Can I also use flux data?

Metabolic fluxes do not directly fit into the dependence scheme that parameter balancing uses internally to link different quantitities. Therefore, fluxes cannot be used as input data, nor can they be predicted directly. However, they can be used in an indirect way, as described in Stanford et al. (2013). The idea is as follows: In parameter balancing including metabolite levels and reaction Gibbs free energies, known flux directions can be used to define the signs of all reaction Gibbs free energies. The resulting rate laws and metabolite levels will be consistent with the predefined fluxes. The kinetic model, parametrised in this way, and with the balanced metabolite levels, will yield reaction rates with the same signs as the predefined fluxes. By rescaling the Vmax values (i.e., scaling the catalytic constants, enzyme levels, or both), reaction rates and fluxes can be matched. The resulting model will correlate to the predefined flux distribution by construction. Note that, in order for this to work, the predefined fluxes must be thermodynamically feasible (i.e., loop-free, and realisable for the (potentially predefined) external metabolite levels.

Where can I find example files?

A number of example files (SBML models and SBtab data tables for parameter balancing) can be found on www.parameterbalancing.net.

Where can I find suitable input data for my own model?

Typical input data for estimating kinetic parameters comprise catalytic constants (kcat values), Michaelis-Menten constants (KM values), equilibrium constants, standard reaction Gibbs free energies, and Gibbs free energies of formation. Typical input data for estimating metabolic states also comprise metabolite concentrations.

- A large collection of kinetic data is provided by the BRENDA Enzyme Database.
- Thermodynamic data for many reactions can be obtained from the website eQuilibrator. This comprises calculated equilibrium constants and standard reaction Gibbs free energies for different values of pH and ionic strength. We provide a collection of these data for parameter balancing here.

How can I define or modify the priors on model parameters?

Experimental data alone will usually not suffice to determine all model parameters. To determine underdetermined parameters, and to keep parameters in realistic ranges, parameter balancing uses prior distributions and constraints for each type of parameter. These priors and constraints are defined in a data table, which can be customised by the user. The table is available on www.parameterbalancing.net.

Which kinetic rate laws are assumed in parameter balancing?

Parameter balancing is based on modular rate laws, a generalised version of the convenience kinetics. The modular rate laws include reversible mass-action and reversible Michaelis-Menten rate laws as special cases. Modular rate laws are also supported by SBMLsqueezer, which allows you to directly insert rate laws into SBML models. In parameter balancing, rate laws and rate constants can be directly inserted into your model at the end of the workflow. Note that all rate laws previously present in your model will be removed.

What physical units are used in parameter balancing?

The units are predefined in the prior table and cannot be changed, unless you provide your own customised prior table.

How does parameter balancing work mathematically?

Parameter balancing employs Bayesian estimation to determine a consistent set of all model parameters. To use it efficiently, it is good to know about some of its details. For technical reasons, all quantities are internally converted to natural scaling. This means that for energy quantities (in kJ/mol), we keep the original values while for all other quantities, we take the natural logarithms. Furthermore, we distinguish between basic quantities and derived quantities (which are uniquely determined by the basic quantities). See the overview of all quantities considered.

During balancing, we integrate information from data (values and standard errors), prior distributions (typical values and spread for basic quantities), and pseudo values (typical values and spread for derived quantities). All these values and spreads are represented by normal distributions (priors, data with standard errors, pseudo values, and posteriors) for the naturally scaled quantities. When converting back to non-logarithmic values, we obtain log-normal distributions, which makes it crucial to distinguish between median and mean values. Eventually, the median values (which are more realistic and guaranteed to satisfy the relevant constraints) are inserted into the model.

Where is the method described?

Parameter balancing is described in Lubitz et al. (2010). If you use parameter balancing in your work, please refer to this article. A modelling workflow based on parameter balancing is described in Stanford et al. (2013).

What are the known caveats?

Parameter balancing makes specific assumptions about the rate laws used. In some cases, this can lead to problems, or parameter balancing may not be suitable for your modelling. Here we list points that can typically lead to problems:

- Large models
  Large networks lead to large parameter sets, which increase the numerical effort of parameter balancing. One possibility to avoid this problem is to use fixed, precalculated equilibrium constants. Then, the kinetic constants of each reaction can be balanced separately, which reduces the effort.
- Biomass reaction or polymerisation reactions
  Many metabolic models (especially, models used in flux balance analysis) contain a "biomass" reaction that involves a large number of compounds with largely varying stoichiometric coefficients. Modular rate laws, as assumed in parameter balancing, use the stoichiometric coefficients as exponents in the formula. For biomass reactions or polyerisation reactions, this is not very realistic as assumption. Furthermore, these reactions usually do not have to be thermodynamically consistent. For both reasons, it is advisable to discard the automatically proposed kinetics, and insert a more realistic kinetics instead (see, for instance, the rate laws proposed in Hofmeyr et al., (2013)).
- Very large or small parameter values
  Very large or small parameter values can lead to unrealistic models and numerical problems. Extreme values should be avoided by using proper

bounds. In any case, we suggest to have a look at all balanced values and to see if they are in realistic ranges.

- Large uncertainties and strongly shifted mean values

  Each balanced parameter value comes with an uncertainty. The uncertainties are described by normal distributions for the logarithmic parameter values, so median and mean value on logarithmic scale will be identical. For the non-logarithmic values, we obtain a log-normal distribution, and median and mean value will differ. If the uncertainty is large (which can easily happen if a value is not constrained by any data values), the mean and median can become very different, and the mean value can become very high. This can be avoided by reducing the uncertainty range - by providing more data that constraints the parameter value, by using narrower priors or constraints, or by using "pseudo" values.

- What if the model cannot be simulated?

  Problems in simulating the model (e.g., using COPASI) may be caused by unrealistically high or low parameter values. If you notice such parameter values in your model and would like to avoid them, you may use tighter priors or pseudo values to exclude extreme parameter values.

How are enzymes handled in the balancing?

In SBML, enzymes are sometimes treated as SBML species and sometimes as SBML parameters; this is mainly up to the modeller. In parameter balancing, enzymes need to be parameters. If they are provided as species, they cannot correctly be assigned as reaction modifiers and thus are ignored.

Contact and Citations

Parameter balancing was developed in the group of Theoretical Biophysics at Humboldt Universität zu Berlin.

If you are using parameter balancing for your research, please cite the original publication “Parameter Balancing in Kinetic Models of Cell Metabolism”, published in The Journal of Physical Chemistry B (2010, DOI: 10.1021/jp108764b).

In case of any questions or feedback please do not hesitate to contact Wolfram Liebermeister (wolfram.liebermeister@gmail.com) and Timo Lubitz (timo.lubitz@gmail.com) for further questions.