Matlab functions for replicate regression

1 Installation

Getting started

1. Unpack the files from the github repository.
2. Include the path to the matlab directory 'replicate_regression' into your MATLAB path.
3. Run 'demo_replicate_regression.m' to see a single replicate regression.
4. Run 'demo_omics_data.m' to see the analysis of a small example omics data set (data and options files are provided in the same directory).

Requirements
The functions were developed and tested with matlab6.

State of the software
The functions are under development and provided 'as is'. If you would like to contribute extensions to the toolbox, please let me know.

Documentation
Documentation (in directory 'doc') has been built automatically with M2HMTL

License

Any questions?
Please send questions, comments, and bug reports to `wolfram.liebermeister@gmail.com`

2 Running a single replicate regression

The MATLAB function `replicate_regression.m` allows you to run a single replicate regression. All data are directly given in the form of MATLAB variables (vectors and structs).

MATLAB function `replicate_regression.m`

The function is called as follows:

[result, options] = replicate_regression(t, y, sigma, r, flag_fix_parameters, varargin)

Bayesian replicate regression for multiple time series measured in replicate.
Data must be provided as vectors and are transformed to logarithmic scale if desired.

Function arguments

t, y, sigma, r input data (times, values, standard errors, replicate labels)
given as row vectors (see replicate_regression_core.m)

flag_fix_parameters (Boolean, optional) If set to 1, the options given in the following argument(s) will be accepted
without changes (otherwise they will be checked and updated)

varagin (optional) Either a list of property/value pairs for algorithm options (list see appendix).
or a structure containing the property/value pairs (this is mandatory if flag_fix_parameters is set to 1)

Function output

result matlab struct with results from replicate regression

options matlab struct with options values that were used in the calculation

The options list is supposed to be ordered by priority; earlier options override later options. The function is a wrapper for the function 'replicate_regression_core'. In converting the data to logarithms, y and sigma are either taken to be medians and geometric standard deviations, or means and standard deviations of the data values. The choice is defined by the argument 'options.transformation'

The basis functions are adjusted to the final time interval [ta,tb](from tt)

'cos'	cosine function, zero slope at t=ta and t=tb
'sin'	sine function, zero value at t=ta and t=tb
'sin_half'	sine function, zero value at t=ta
'sin_horizontal'	sine function, zero value at t=ta, zero slope at t=tb
'cos+sin'	cosine and sine functions, no restriction
'polynomial'	polynomial function, zero value at t=ta
'exp'	exponentially relaxing functions (t<0 => f=0; t>0 => f = 1-exp(t/tau)

The entire curves are shifted by a constant basis function This can be suppressed by setting options.use_offset = 0

3 How to run a replicate regression for omics data

The MATLAB function `replicate_regression_omics.m` allows you to run replicate regression for an entire omics data set, including iterative updating of the curve priors. Data are given in the form of table files (both for omics data and for function options).

Matlab function `replicate_regression_omics.m`

The function is called as follows:

replicate_regression_omics(data_file, user_options_file, base_directory)

Bayesian replicate regression for omics data

Function arguments

data_file Omics data file (full directory path)

user_options_file Table file containing the options (full directory path). For list of options, see appendix.

base_directory Directory name for results (full directory path)

Function output Output data and graphics are written to files

How to prepare data and run `replicate_regression_omics.m`

1. Create a directory for the analysis
2. Create in this directory subdirecties "data", "options", "results", and "graphics"
3. Create a data file (tab-separated text table in the format described below) and save it to the "data" subdirectory
4. Create an options file (tab-separated text table in the format described below) and save it to the "options" subdirectory
5. Start matlab and run replicate regression
   (see matlab script replicate_regression/demo/omics_data_example/omics_data_example.m)

```
% Directory name for the omics set
base_DIR = [replicate_regression_DIR '/demo/omics_data_example/'];

% Name of options file
foptions_file = [base_DIR '/options/options_omics_data_example.csv'];

% Run script for replicate regression of omics data
replicate_regression_omics_analysis;
```

Examples of data and options files can be found in the subdirectory replicate_regression/demo/omics_data_example

Format of data file (tab-separated text file)

The following example table shows the format for data tables:

!UniprotID	S1R1	S2R1	S3R1	S4R1	S1R2	S2R2	S3R2	S4R2	S1R3	S2R3	S3R3	S4R3
!Time	0	10	20	30	0	10	20	30	0	10	20	30
!Replicate	R1	R1	R1	R1	R2	R2	R2	R2	R3	R3	R3	R3
Q04747	1.05	1.19	0.64	1.09	1.39	0.69	1.33	1.19	1.3	1.07	0.93	0.99
P27206	0.98	1.16	0.65	1.29	1.27	0.48	1.12	1.12	1.3	1.14	1.04	1.02
...	...	...	...	...	...	...	...	...	...	...	...	...

The first line contains the headers of the protein names column (e.g., !BSUnumber, !BGnumber, !GeneName, !UniprotID), followed by sample names (as headers of data columns). The second line contains the time points as numbers. The third line contains replicate names, and the fourth line (which is optional) contains the type of value given ('Value', 'Mean', or 'Std'). All further lines contain numerical data. The entry "!UniprotID" can be replaced by the name of gene identifiers used. The entries "!Time" and "!Replicate" are fixed. The sample names (in first line) and replicate names (third line) can be freely choosen; however, they must not start with a number and may not contain any special characters (e.g., ":" or ".") or whitespace characters.

Format of options file (tab-separated text file)

Options can be given in a tab-separated table file like in the following example:

```
% THIS IS A COMMENT
data_dir                    [base_DIR '/data/']
result_dir                  [base_DIR '/results/']
graphics_dir                [base_DIR '/graphics/']
data_file_csv               my_data_set_data.csv
data_file_matlab            my_data_set_data.mat
options_file                [replicate_regression_DIR '/resources/mcr_options_nonlogarithmic_data.csv']
options_out_csv             my_data_set_result_regression_options.csv
result_file_matlab          my_data_set_result.mat
result_file_csv             my_data_set_result.csv
graphics_file               my_data_set_graphics
log_file                    my_data_set_log.txt
data_time_unit              min
data_scale                  absolute
normalise_by_median         1
convert_to_logarithm        1
log_transformation          arithmetic
data_std_relative           0.25
data_min_data_points        3
fixed_prior                 0
prior_updating              3
updating_factor             1.2
t_smooth                    30
run_crossvalidation         1
```

Each line contains one attribute. The first entry contains the attribute name, the second column the attribute value (string or number). All further entries are ignored. Lines starting with the '%' character are ignored (can be used for comments). The attribute 'options_file' allows to declare another options file containing default options. The attribute 'data_file_csv' contains the name of the data file

A Function options

A.1 Options for matlab function `replicate_regression.m`

Options for the matlab function `replicate_regression` are given in a struct called `options`. Possible fields with default values are listed below.

OPTION	IN CORE	TYPE	DEFAULT	MEANING
options.verbose		Boolean	1	Output information during regression
options.is_logarithmic		Boolean	0	Declare that data are logarithmic
options.convert_to_logarithm		Boolean	1	Convert data to logarithms for regression
options.log_transformation		string	'arithmetic'	'arithmetic', 'geometric'
options.run_crossvalidation		Boolean	0	Run crossvalidation
options.set_std		float	nan	Value to replace all data standard deviations
options.insert_std		float	1	Value to replace missing data standard deviations
options.start_at_t		float	0	Start regression curves at starting time 'start_at_t' (instead of t=0)
options.start_value		float	nan	Fixed start value for regression curves
options.shift_data		string	'mean'	Policy for shifting data before regression 'none', 'fixed_start_value', 'mean', 'initial', 'fixed_1'
options.shift_value		float	nan	Shift used when shifting the data
options.basis	X	string	'cos+sin'	Type of basis functions (see table below)
options.n_comp	X	int	nan	Fixed number of basis functions
options.n_comp_min		int	1	Minimal number of basis functions
options.n_comp_max		int	20	Maximal number of basis functions
options.use_offset	X	Boolean	1	Use constant function as one of the basis functions
options.constant_before_start	X	Boolean	0	Set all basis functions constant for t¡0
options.deviation_same_start		Boolean	0	Enforce identical start values for all replicates
options.remove_offset	X	Boolean	0	Omit offset when creating the regression curves
options.t_smooth		float	nan	Time constant for setting decreasing prior widths
options.t_jump	X	float	nan	Time constant for initial jump basis function
options.t_interp		float	t	Time points for interpolated regression curves
options.average_std	X	string	'std_dev_mean'	Type of uncertainty to be reported for average curve
options.central_offset_mean	X	float	0	Prior mean sigma_alpha_0 (for alpha_0)
options.central_offset_width	X	float	1	Prior width sigma_alpha_0 (for alpha_0)
options.central_first_mode_mean	X	float	0	Prior mean sigma_alpha_1 (for alpha_1)
options.central_first_mode_width	X	float	1	Prior width sigma_alpha_1 (for alpha_1)
options.central_mode_mean	X	vector	[]	Prior means sigma_alpha_m (for alpha_m)
options.central_mode_width	X	vector	[]	Prior widths sigma_alpha_m (for alpha_m)
options.central_jump_mean	X	float	nan	Prior means sigma_alpha_jump (for alpha_jump)
options.central_jump_width	X	float	nan	Prior widths sigma_alpha_jump (for alpha_jump)
options.deviation_offset_mean	X	float	0	Prior mean sigma_beta_0 (for beta_0)
options.deviation_offset_width	X	float	1	Prior width sigma_beta_0 (for beta_0)
options.deviation_first_mode_mean	X	float	0	Prior mean sigma_beta_1 (for beta_1)
options.deviation_first_mode_width	X	float	1	Prior width sigma_beta_1 (for beta_1)
options.deviation_mode_mean	X	float	[]	Prior means sigma_beta_m (for beta_m)
options.deviation_mode_width	X	float	[]	Prior widths sigma_beta_m (for beta_m)
options.deviation_jump_mean	X	float	0	Prior means sigma_beta_jump (for beta_jump)
options.deviation_jump_width	X	float	1	Prior widths sigma_beta_jump (for beta_jump)
options.flag_draw_sample	X	Boolean	1	Draw sample curve parameters and curve from the posterior
options.flag_time_derivative	X	Boolean	0	Compute time derivative curves

The options marked in column "IN CORE" are used by the underlying function `replicate_regression_core.m`

A.2 Options for matlab function `replicate_regression_omics.m`

Options for the matlab function `replicate_regression_omics_analysis` are given in a struct called `foptions`. Possible fields with default values (see function `replicate_regression_omics_default_options`) are listed below. The same options can also be set in an options file.

OPTION	MEANING
data_dir	directory name for data files
result_dir	directory name for result files
graphics_dir	directory name for graphics
data_file_csv	filename for data file (tsv format, see examples)
data_file_matlab	filename for matlab data file (written during the analysis)
options_file	filename for default options file (tsv format)
options_out_csv	filename for completed options file (tsv format, written during analysis)
translation_table_file	filename for ID mapping table (see example)
result_file_matlab	filename for
result_file_csv	filename for hahne_salt_stress_cytosol_result.tsv
result_file_zip	filename for hahne_salt_stress_result.zip
graphics_file	file basename for graphics
data_time_unit	time unit ('min')
data_scale	'absolute' or 'log2' (also 'ln','log','log10','log2 ratio'; these are all treated like 'log2');
data_min_num_replicates	minimal number of valid replicates (genes with less valid replicates are discarded; default 1)
For non-logarithmic data:	
abs_data_adjust_std_upper	upper threshold; points above are outliers (increase std dev by factor of 3)
abs_data_adjust_std_lower	lower threshold; points below are outliers (increase std dev by factor of 3)
data_std_relative	default for relative standard deviation
data_std_minimal	minimal standard deviation
For logarithmic data	i.e., ('log2', 'ln','log','log10','log2 ratio')
data_std_log	default for standard deviation (on log scale)
log_data_adjust_std_threshold	threshold for data values (on chosen log scale) for which std dev is modified
	criterion: | [data value] - [median for this gene and replicate] | $<$ threshold
	for the inserted std dev, see next entry
log_data_adjust_std_factor	new std width = factor * absolute deviation from median
data_min_data_points	minimal number of data points required in the analysis (default 3)
	at least one replicate has to reach this number, points are times $t_i 0$ do not count
	replicates with less data points are ignored
convert_to_logarithm	convert (nonlogarithmic) data to logarithms for replicate regression (Boolean)
log_transformation	type of transformation
	'arithmetic': data=mean values and plotting on absolute scale
	'geometric' : data = median values and plotting on log scale (but data on absolute scale)
ignore_std_deviations	Boolean, ignore standard deviations given in data
fixed_prior	keeping the prior fixed? (Boolean, default 0)
prior_updating	number of prior updating iterations (default 10)
updating_factor	updating factor, default 1.1
update_prior_means	change parameter means from 0 to posterior means while updating? default 0
t_smooth	time constant defining how prior widths depend on the frequency
options_start_value	fixed starting value; to be inserted into options as options.start_value
options_start_at_t	Starting time point for changes (after constant behaviour)
	to be inserted into options as options.start_at_t
options_constant_before_start	Boolean (keep curves constant before starting time)
	to be inserted into options as options.constant_before_start
..	..

..	..
regression_t_interp	time points for regression (optional)
regression_tmin	start time for regression (optional)
regression_tmax	end time for regression (optional)
crossvalidation	run crossvalidation? (Boolean, default 0)
postprocess_normalise	Boolean, default 1
graphics_individual	file basename (used in script replicate_regression_omics_selected'
graphics_scale	default 'log2', 'linear'
graphics_format	'eps', 'png' (for technical reasons, 'eps' needs to be written in single quotes)
convenience_name	type of protein names to be used in graphics (default 'SubtiWiki_20090701')
normalise_by_median	(Boolean, default TRUE)
mark_outliers_percentage	percentage of data points to be marked as outliers based on crossvalidation error

Additional attributes in options file for individual graphics (function 'replicate_regression_omics_selected')

OPTION	MEANING
graphics_scale	'log2','linear'
postprocess_normalise	1
element_id	id (or list, selected by —)
element_name	name (or list, selected by —)
delimiter_symbol	symbol for delimiting list of elements (in element_id, element_name)
title_string	title for graphics
x_label	x label for graphics
y_label	y label for graphics
plot_data	produce plot for data (single element)
plot_replicates	produce plot with replicates (single element)
plot_regression	produce plot for regression curves (single element)
plot_all	produce joint plots for all elements