

# Package: dynafluxr (via r-universe)

June 6, 2026

**Type** Package

**Title** Retrieve Reaction Rate Dynamics from Metabolite Concentration Time Courses

**Version** 1.0.1

**Description** Reaction rate dynamics can be retrieved from metabolite concentration time courses. User has to provide corresponding stoichiometric matrix but not a regulation model (Michaelis-Menten or similar). Instead of solving an ordinary differential equation (ODE) system describing the evolution of concentrations, we use B-splines to catch the concentration and rate dynamics then solve a least square problem on their coefficients with non-negativity (and optionally monotonicity) constraints. Constraints can be also set on initial values of concentration. The package 'dynafluxr' can be used as a library but also as an application with command line interface `dynafluxr::cli("`-h")` or graphical user interface `dynafluxr::gui()`.

**License** GPL-2

**Encoding** UTF-8

**Depends** bspline (>= 2.5.0), nlsic (>= 1.1.1)

**Imports** optparse, qpdf, arrApply, slam, gmresls (>= 0.2), shiny, shinyjs, shinyFiles

**RoxygenNote** 7.3.2

**Suggests** RUnit, knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Serguei Sokol [aut, cre]

**Maintainer** Serguei Sokol <sokol@insa-toulouse.fr>

**Config/pak/sysreqs** cmake make libjpeg-dev libuv1-dev libssl-dev zlib1g-dev

**Repository** <https://sgsokol.r-universe.dev>

**Date/Publication** 2025-07-10 14:50:09 UTC

**RemoteUrl** <https://github.com/cran/dynafluxr>

**RemoteRef** HEAD

**RemoteSha** 2f55e54a64957ecc82c4213a98788e1f01c935fc

## Contents

cli . . . . .	2
fdyn . . . . .	3
gui . . . . .	5
txt2sto . . . . .	6
<b>Index</b>	<b>7</b>

---

cli	<i>Function to be called from shell command line</i>
-----	--

---

### Description

Function to be called from shell command line

### Usage

```
cli(args = commandArgs(trailingOnly = TRUE))
```

### Arguments

args                    Character vector, command line parameters (default `commandArgs(trailingOnly=TRUE)`)

### Details

run `cli("-h")` in R or `Rscript -e 'dynafluxr::cli()' -h` in shell to get a help page with available option description

### Value

a list resulting from `fdyn()` call

### See Also

fdyn

**Examples**

```

# from shell
# $ Rscript --vanilla -e 'dynafluxr::cli()' -m data_kinetics.tsv -s glycolysis.txt

# from R session
ddir=system.file("dataglyco", package="dynafluxr")
meas=file.path(ddir, "data.tsv")
sto=file.path(ddir, "network.txt")
res=cli(c("-m", meas, "-s", sto, "--skip", "24", "-o", ""))
tp=res$tp
np=length(tp)
tpp=res$tpp
# plot species
matplot(tpp, res$mtp(tpp), type="l")
matpoints(tp, res$mf[,-1], pch=".", cex=0.5)
legend("topright", legend=colnames(bsppar(res$mtp)$qw), lty=1:5, col=1:6, cex=0.75)
# plot rates
dev.new()
matplot(tpp, res$vsp(tpp), type="l")
# plot residuals
dev.new()
matplot(tpp, res$risp(tpp), type="l")
legend("topright", legend=colnames(bsppar(res$risp)$qw), lty=1:5, col=1:6, cex=0.75)

```

fdyn

*Retrieve flux dynamics from metabolic kinetics***Description**

Retrieve flux dynamics from metabolic kinetics

**Usage**

```

fdyn(
  mf,
  stofull,
  nsp = 4L,
  nki = 5L,
  lieq = NULL,
  monotone = 0,
  dls = FALSE,
  atomlen = NULL,
  npi = 300L,
  wsd = FALSE,
  nmsf = character(0L),
  sderr = NULL,
  tol = 1e-10,
  regular_grid = TRUE
)

```

**Arguments**

mf	Data-frame or matrix, specie kinetic measurements. Columns must be named with specie names and 'Time'.
stofull	Full stoichiometric matrix, <code>stofull[i, j]</code> means reaction 'j' produces specie 'i' with coefficient 'stofull[i,j]'. If <code>stofull[i, j] &lt; 0</code> , the specie 'i' is consumed. Columns must be named with reaction names. Rows must be names with the species. "Full" in the name means that matrix includes even NA species.
nsp	Integer, polynomial order of B-spline to use for species (default 4)
nki	Integer, number of internal knots for B-splines (default 5)
lieq	List, equality constraints on species (default NULL, i.e. no equality constraint)
monotone	Numeric scalar or vector, 1=species are monotonically increasing; -1=monotonically decreasing; 0=no constraint. If vector, each value constraints (or not) a corresponding data column in mf ('Time' column is excluded from counting) (default 0, i.e. no monotonicity constraint)
dls	Logical scalar, if TRUE, indicates that differential least squares should be resolved instead of integral least squares. (default FALSE, i.e. ILS will be used)
atomlen	Numerical named vector, indicates what is label length of a given specie used a vector item name. If provided, results will contain <code>lsp</code> and <code>ilsp</code> fields which are a B-spline function representing atom balance over <code>m</code> and <code>i</code> splines. (default NULL, i.e. no atom balance will be provided)
npi	Integer scalar, indicates a number of plot intervals to produce smooth plots. (default 300)
wsd	Logical scalar, if TRUE, indicates that differential least squares should be resolved with residuals weighted by a factor of covariance matrix. (default FALSE, i.e. no weighting is used)
nmsf	Character vector, list of species for which scaling factor must be estimated for -dls.
sderr	Numeric vector, use this SD of measured metabolites instead of automatically estimated. The name of each vector component is metabolite name, the value is SD value, e.g. <code>c(GLC=0.1)</code> (default NULL, i.e. SD are automatically estimated).
tol	Double scalar, tolerance for detecting singular matrices and solving linear systems
regular_grid	Logical scalar, use regular knot grid (default: TRUE)

**Details**

Each item in `lieq` corresponds to a specie and is a 2 column matrix (Time, Value). Each row of this matrix indicates what 'Value' must take corresponding specie at what 'Time'. Typically, it can be used to impose starting values at Time=0 for some species.

All specie fits are constraint to have values  $\geq 0$ .

**Value**

List with following components:

**mf:** specie data frame used for fitting  
**fp:** vector of time points for used measurements  
**tp:** vector of time points for plot (fine time resolution)  
**sto:** stoichiometric matrix used for fitting  
**stofull:** stoichiometric matrix before a possible NA elimination  
**stoinv:** pseudo-inverse of sto  
**mfp:** measured specie spline function  
**vfp:** estimated rates spline function  
**tfp:** estimated total flux ( $S \cdot v$ ) spline function  
**dfp:** first derivative of measured spline function  
**isp:** integrated specie spline function  
**asp:** atom balance over mfp spline function  
**iasp:** atom balance over isp spline function  
**vfp:** flux spline function  
**dfp:** measured specie first derivative spline function  
**rsp:** residual  $dM/dt - S \cdot v$  spline function  
**isp:** integral residual  $M - \int S \cdot v dt$  spline function  
**sdrate:** matrix of SD values for flux B-spline coefficients, of size (ncoef x nrate)  
**chi2tab:** data-frame with chi2-test results  
**sf:** named scale factor vector  
**internal\_knot\_ref:** number of internal knots used for estimation of var\_ref

---

gui

*Launch graphical user interface in web browser*

---

### Description

Launch graphical user interface in web browser

### Usage

gui()

### Value

no returned value

---

`txt2sto`*Translate plain text file with reactions to stoichiometric matrix*

---

**Description**

Translate plain text file with reactions to stoichiometric matrix

**Usage**

```
txt2sto(fn)
```

**Arguments**

`fn`                   String, file name

**Value**

Matrix of size `n_metab` x `n_reac`

# Index

cli, 2

fdyn, 3

gui, 5

txt2sto, 6