|  |  |
| --- | --- |
| mm\_data {streamMetabolizer} | R Documentation |

Return the data types that may be used by metab\_models using the metab\_model\_interface.

**Description**

Produces a unitted data.frame with the column names, units, and data format to be used by metab\_models that comply strictly with the metab\_model\_interface. These are the columns that may be included:

* solar.time date-time values in mean solar time (see [calc\_solar\_time](http://127.0.0.1:9671/help/library/streamMetabolizer/help/calc_solar_time) and/or [convert\_UTC\_to\_solartime](http://127.0.0.1:9671/help/library/streamMetabolizer/help/convert_UTC_to_solartime)), in POSIXct format with a tzone attribute of 'UTC'. May be approximated by local, non-daylight-savings clock time (still with nominal UTC timezone but with clock noons close to solar noon), but mean solar time is better for matching model time windows to the diel cycle of light availability. Throughout this package, variables named "solar.time" are mean solar time, "app.solar.time" means apparent solar time, and "any.solar.time" means either.
* DO.obs dissolved oxygen concentration observations, *mg O2 / L*
* DO.sat dissolved oxygen concentrations if the water were at equilibrium saturation *mg O2 / L*. Calculate using [calc\_DO\_sat](http://127.0.0.1:9671/help/library/streamMetabolizer/help/calc_DO_sat)
* depth stream depth, *m*.
* temp.water water temperature, *degC*.
* light photosynthetically active radiation, *micro mols / m^2 / s*
* date dates of interest in Date format
* err.obs.sigma SD of observation error to use in simulating data
* err.obs.phi autocorrelation of observation error to use in simulating data
* err.proc.sigma SD of process error to use in simulating data
* err.proc.phi autocorrelation of process error to use in simulating data
* DO.obs dissolved oxygen concentration observations, *mg O2 / L*
* GPP daily estimates of GPP, *g O[2] m^-2 d^-1*
* ER daily estimates of ER, *g O[2] m^-2 d^-1*
* K600 daily estimates of K600, *d^-1*
* GPP.init daily initial values of GPP, *g O[2] m^-2 d^-1*, for use in maximum likelihood estimation
* ER.init daily initial values of ER, *g O[2] m^-2 d^-1*, for use in maximum likelihood estimation
* K600.init daily initial values of K600, *d^-1*, for use in maximum likelihood estimation
* discharge.daily daily mean river discharge, *m^3 s^-1*
* velocity.daily daily mean river flow velocity, *m s^-1*

**Usage**

mm\_data(..., optional = "none")

**Arguments**

|  |  |
| --- | --- |
| ... | column names to select, as passed to [select](http://127.0.0.1:9671/help/library/dplyr/html/select.html) |
| optional | one or more character strings listing the columns, if any, that may be excluded. If 'all', the entire data.frame may be omitted. If 'none', the entire data.frame must be included as prototyped. If specific column names are given, those columns may be omitted entirely or passed to[metab](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab)() as all NAs. |

**Details**

Most models will require a subset of these data columns. Specialized models may deviate from this format, but this is discouraged.

**Value**

data data.frame with columns as in the description

**Examples**

# all possible columns

mm\_data()

# columns typical of instantaneous data

mm\_data(solar.time, DO.obs, DO.sat, depth, temp.water, light)

# columns typical of daily data

mm\_data(date, K600.daily, discharge.daily, velocity.daily)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| mm\_name {streamMetabolizer} | R Documentation |

Find the name of a model by its features

**Description**

A model\_name concisely specifies the structure of a metabolism model. From a model\_name, an appropriate set of model specifications (parameters and runtime options) can be generated with [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs). From a complete specs list, a metabolism model can be run with [metab](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab).

**Usage**

mm\_name(

type = c("mle", "bayes", "night", "Kmodel", "sim"),

pool\_K600 = c("none", "normal", "normal\_sdzero", "normal\_sdfixed", "linear",

"linear\_sdzero", "linear\_sdfixed", "binned", "binned\_sdzero", "binned\_sdfixed",

"complete"),

err\_obs\_iid = c(TRUE, FALSE),

err\_proc\_acor = c(FALSE, TRUE),

err\_proc\_iid = c(FALSE, TRUE),

err\_proc\_GPP = c(FALSE, TRUE),

ode\_method = c("trapezoid", "euler", "rk2", "lsoda", "lsode", "lsodes", "lsodar",

"vode", "daspk", "rk4", "ode23", "ode45", "radau", "bdf", "bdf\_d", "adams",

"impAdams", "impAdams\_d", "Euler", "pairmeans", "NA"),

GPP\_fun = c("linlight", "satlight", "satlightq10temp", "NA"),

ER\_fun = c("constant", "q10temp", "NA"),

deficit\_src = c("DO\_mod", "DO\_obs", "DO\_obs\_filter", "NA"),

engine = c("stan", "nlm", "lm", "mean", "loess", "rnorm"),

check\_validity = TRUE

)

**Arguments**

|  |  |
| --- | --- |
| type | character. The model type. Options:   * mle: maximum likelihood estimation (see also [metab\_mle](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_mle)) * bayes: bayesian hierarchical models [metab\_bayes](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_bayes) * night: nighttime regression (see also [metab\_night](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_night)) * Kmodel: regression of *daily* estimates of K600.daily versus discharge, time, etc., usually for 3-phase estimation of K alone (by MLE or nighttime regression), K vs discharge (using this model), and then GPP and ER with fixed K (by MLE) (see also[metab\_Kmodel](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_Kmodel)) * sim: simulation of DO.obs 'data' for testing other models (see also [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)) |
| pool\_K600 | character. [How] should the model pool information among days to get more consistent daily estimates for K600? Options (see Details for more):   * none: no pooling of K600 * normal: *K600 ~ N(mu, sigma)* * linear: *K600 ~ N(B[0] + B[1]\*Q, sigma)* * binned: *K600 ~ N(B[Q\_bin], sigma)* where *mu ~ N(mu\_mu, mu\_sigma)* and *sigma ~ N(sigma\_mu, sigma\_sigma)* * complete: applicable only for type='Kmodel', which is generally used in conjunction with preceding estimates of K (e.g., by type='mle' or type='night') and subsequent estimates of GPP and ER (e.g., by type='mle' with daily K600 values specified) |
| err\_obs\_iid | logical. Should IID observation error be included? If not, the model will be fit to the differences in successive DO measurements, rather than to the DO measurements themselves. |
| err\_proc\_acor | logical. Should autocorrelated process error (with the autocorrelation term phi fitted) be included? |
| err\_proc\_iid | logical. Should IID process error be included? |
| err\_proc\_GPP | logical. Should IID process error in GPP be included? This kind of error occurs only during the day and is used to adjust GPP before passing that adjusted GPP into the dDO/dt equation. The GPP\_inst variable is the corrected GPP, and a new variable, GPP\_inst\_partial, contains the pre-adjustment GPP estimates |
| ode\_method | character. The method to use in solving the ordinary differential equation for DO. Options:   * euler, formerly Euler: the final change in DO from t=1 to t=2 is solely a function of GPP, ER, DO, etc. at t=1 * trapezoid, formerly pairmeans: the final change in DO from t=1 to t=2 is a function of the mean values of GPP, ER, etc. across t=1 and t=2. * for type='mle', options also include rk2 and any character method accepted by [ode](http://127.0.0.1:9671/help/library/deSolve/html/ode.html) in the deSolve package (lsoda, lsode, lsodes, lsodar, vode, daspk, rk4, ode23, ode45, radau, bdf, bdf\_d, adams, impAdams, and impAdams\_d; note that many of these have not been well tested in the context of streamMetabolizer models) |
| GPP\_fun | character. Function dictating how gross primary productivity (GPP) varies within each day. Options:   * linlight: GPP is a linear function of light with an intercept at 0 and a slope that varies by day.  GPP(t) = GPP.daily \* light(t) / mean.light   + GPP.daily: the daily mean GPP, which is partitioned into timestep-specific rates according to the fraction of that day's average light that occurs at each timestep (specifically, mean.light is the mean of the first 24 hours of the date's data window) * satlight: GPP is a saturating function of light.  GPP(t) = Pmax \* tanh(alpha \* light(t) / Pmax)   + Pmax: the maximum possible GPP   + alpha: a descriptor of the rate of increase of GPP as a function of light * satlightq10temp: GPP is a saturating function of light and an exponential function of temperature.  GPP(t) = Pmax \* tanh(alpha \* light(t) / Pmax) \* 1.036 ^ (temp.water(t) - 20)   + Pmax: the maximum possible GPP   + alpha: a descriptor of the rate of increase of GPP as a function of light * NA: applicable only to type='Kmodel', for which GPP is not estimated |
| ER\_fun | character. Function dictating how ecosystem respiration (ER) varies within each day. Options:   * constant: ER is constant over every timestep of the day.  ER(t) = ER.daily   + ER.daily: the daily mean ER, which is equal to instantaneous ER at all times * q10temp: ER at each timestep is an exponential function of the water temperature and a temperature-normalized base rate.  ER(t) = ER20 \* 1.045 ^ (temp.water(t) - 20)   + ER20: the value of ER when temp.water is 20 degrees C * NA: applicable only to type='Kmodel', for which ER is not estimated |
| deficit\_src | character. From what DO estimate (observed or modeled) should the DO deficit be computed? Options:   * DO\_mod: the DO deficit at time t will be (DO.sat(t) - DO\_mod(t)), the difference between the equilibrium-saturation value and the current best estimate of the true DO concentration at that time * DO\_obs: the DO deficit at time t will be (DO.sat(t) - DO.obs(t)), the difference between the equilibrium-saturation value and the measured DO concentration at that time * DO\_obs\_filter: applicable only to type='night': a smoothing filter is applied over the measured DO.obs values before applying nighttime regression * NA: applicable only to type='Kmodel', for which DO deficit is not estimated |
| engine | character. With which function or software should the model fitting be done?   * for type='mle': nlm only (the default) * for type='bayes': stan only (the default), an external software package that runs MCMC chains for Bayesian models (see http://mc-stan.org) * for type='night': lm only (the default) * for type='Kmodel': mean, lm, or loess enable different types of relationships between daily K600 and its predictors (nothing, discharge, time, etc.) * for type='sim': rnorm only (the default) |
| check\_validity | logical. if TRUE, this function checks the resulting name against mm\_valid\_names(type). |

**Details**

While the Usage shows all valid values for each argument, not all argument combinations are valid; the combination will also be checked ifcheck\_validity==TRUE. For arguments not explicitly specified, defaults depend on the value of type: any argument that is not explicitly supplied (besides type and check\_validity) will default to the values indicated by mm\_parse\_name(mm\_valid\_names(type)[1]).

***pool\_K600***

Here are the essential model lines (in Stan language) that distinguish the K pooling options.

|  |  |
| --- | --- |
| **pool\_K600** | **Model code** |
| none | K600\_daily ~ normal(K600\_daily\_mu, K600\_daily\_sigma) |
| normal | K600\_daily ~ normal(K600\_daily\_mu, K600\_daily\_sigma) |
|  | K600\_daily\_mu ~ normal(K600\_daily\_mu\_mu, K600\_daily\_mu\_sigma) |
|  | K600\_daily\_sigma ~ gamma(K600\_daily\_sigma\_shape, K600\_daily\_sigma\_rate) |
| linear | K600\_daily\_pred <- K600\_daily\_beta[1] + K600\_daily\_beta[2] \* discharge\_daily |
|  | K600\_daily ~ normal(K600\_daily\_pred, K600\_daily\_sigma) |
|  | K600\_daily\_beta ~ normal(K600\_daily\_beta\_mu, K600\_daily\_beta\_sigma) |
|  | K600\_daily\_sigma ~ gamma(K600\_daily\_sigma\_shape, K600\_daily\_sigma\_rate) |
| binned | K600\_daily\_pred <- K600\_daily\_beta[Q\_bin\_daily] |
|  | K600\_daily ~ normal(K600\_daily\_pred, K600\_daily\_sigma) |
|  | K600\_daily\_beta ~ normal(K600\_daily\_beta\_mu, K600\_daily\_beta\_sigma) |
|  | K600\_daily\_sigma ~ gamma(K600\_daily\_sigma\_shape, K600\_daily\_sigma\_rate) |
| complete | [This option refers to complete pooling via metab\_Kmodel in conjunction with preceding |
|  | estimates of K (e.g., by metab\_mle or metab\_night) and subsequent estimates of GPP and ER |
|  | (e.g., by metab\_mle with daily K600 values specified)] |
|  |  |

**See Also**

The converse of this function is [mm\_parse\_name](http://127.0.0.1:9671/help/library/streamMetabolizer/help/mm_parse_name).

**Examples**

mm\_name('mle')

mm\_name('mle', GPP\_fun='satlight', ER\_fun='q10temp')

mm\_name('night')

mm\_name('sim', err\_proc\_acor=TRUE)

mm\_name('bayes', pool\_K600='binned')

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| metab\_mle {streamMetabolizer} | R Documentation |

Maximum likelihood metabolism model fitting function

**Description**

Uses maximum likelihood to fit a model to estimate GPP and ER from input data on DO, temperature, light, etc. Discharge is only used, if at all, to identify and exclude days with any negative discharge.

**Usage**

metab\_mle(

specs = specs(mm\_name("mle")),

data = mm\_data(solar.time, DO.obs, DO.sat, depth, temp.water, light, discharge,

optional = "discharge"),

data\_daily = mm\_data(date, K600.daily, init.GPP.daily, init.Pmax, init.alpha,

init.ER.daily, init.ER20, init.K600.daily, optional = "all"),

info = NULL

)

**Arguments**

|  |  |
| --- | --- |
| specs | a list of model specifications and parameters for a model. Although this may be specified manually (it's just a list), it is easier and safer to use [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) to generate the list, because the set of required parameters and their defaults depends on the model given in themodel\_name argument to specs. The help file for [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) lists the necessary parameters, describes them in detail, and gives default values. |
| data | data.frame (not a tbl\_df) of input data at the temporal resolution of raw observations (unit-value). Columns must have the same names, units, and format as the default. The solar.time column must also have a timezone code ('tzone' attribute) of 'UTC'. See the **'Formatting data'** section below for a full description. |
| data\_daily | data.frame containing inputs with a daily timestep. See the **'Formatting data\_daily'** section below for a full description. |
| info | any information, in any format, that you would like to store within the metab\_model object |

**Value**

A metab\_mle object containing the fitted model. This object can be inspected with the functions in the [metab\_model\_interface](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_model_interface). The code column in get\_fit(mm) is defined in the Value subsection of ?nlm.

**See Also**

Other metab\_model: [metab\_Kmodel](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_Kmodel), [metab\_bayes](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_bayes), [metab\_night](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_night), [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)

**Examples**

dat <- data\_metab('3','30')

# PRK

mm <- metab\_mle(data=dat)

predict\_metab(mm)

# PR with fixed K on two days

dat\_daily <- data.frame(date=as.Date(c("2012-09-18","2012-09-20")), K600.daily=35)

metab\_mle(data=dat, data\_daily=dat\_daily)

# PRK with day-specific inits on some days

dat\_daily <- data.frame(date=as.Date("2012-09-19"),

init.GPP.daily=4, init.K600.daily=60)

metab\_mle(data=dat, data\_daily=dat\_daily)

# Nonlinear GPP or ER equations

metab\_mle(specs(mm\_name('mle', GPP\_fun='satlight')), data=dat)

metab\_mle(specs(mm\_name('mle', ER\_fun='q10temp')), data=dat)

## Not run:

plot\_DO\_preds(predict\_DO(mm))

## End(Not run)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| metab\_bayes {streamMetabolizer} | R Documentation |

Basic Bayesian metabolism model fitting function

**Description**

Fits a Bayesian model to estimate GPP and ER from input data on DO, temperature, light, etc. See [mm\_name](http://127.0.0.1:9671/help/library/streamMetabolizer/help/mm_name) to choose a Bayesian model and [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) for relevant options for the specs argument.

**Usage**

metab\_bayes(

specs = specs(mm\_name("bayes")),

data = mm\_data(solar.time, DO.obs, DO.sat, depth, temp.water, light, discharge,

optional = "discharge"),

data\_daily = mm\_data(date, discharge.daily, optional = "all"),

info = NULL

)

**Arguments**

|  |  |
| --- | --- |
| specs | a list of model specifications and parameters for a model. Although this may be specified manually (it's just a list), it is easier and safer to use [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) to generate the list, because the set of required parameters and their defaults depends on the model given in themodel\_name argument to specs. The help file for [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) lists the necessary parameters, describes them in detail, and gives default values. |
| data | data.frame (not a tbl\_df) of input data at the temporal resolution of raw observations (unit-value). Columns must have the same names, units, and format as the default. The solar.time column must also have a timezone code ('tzone' attribute) of 'UTC'. See the **'Formatting data'** section below for a full description. |
| data\_daily | data.frame containing inputs with a daily timestep. See the **'Formatting data\_daily'** section below for a full description. |
| info | any information, in any format, that you would like to store within the metab\_model object |

**Details**

As of summer and fall 2016, a new compilation of any Stan model gives deprecation warnings including typedef 'size\_type' locally defined but not used [-Wunused-local-typedefs], typedef 'index\_range' locally defined but not used [-Wunused-local-typedefs], typedef 'index' locally defined but not used [-Wunused-local-typedefs], and 'void stan::math::set\_zero\_all\_adjoints()' defined but not used [-Wunused-function]. THESE ARE OKAY. Subsequent runs of the compiled Stan model will be quieter, and the model will work.

**Value**

A metab\_bayes object containing the fitted model. This object can be inspected with the functions in the [metab\_model\_interface](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_model_interface) and also [get\_mcmc](http://127.0.0.1:9671/help/library/streamMetabolizer/help/get_mcmc).

**Author(s)**

Alison Appling, Bob Hall

**See Also**

Other metab\_model: [metab\_Kmodel](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_Kmodel), [metab\_mle](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_mle), [metab\_night](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_night), [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)

**Examples**

## Not run:

dat <- data\_metab('3', res='30')

# fast-ish model version, but still too slow to auto-run in examples

mm <- metab\_bayes(data=dat,

specs(mm\_name('bayes', err\_proc\_iid=FALSE),

n\_cores=3, n\_chains=3, burnin\_steps=300, saved\_steps=100))

mm

get\_fitting\_time(mm)

predict\_metab(mm)

plot\_DO\_preds(predict\_DO(mm))

# error and warning messages are printed with the mm object if present

dat <- data\_metab('3', res='30', flaws=c('missing middle'))

mm <- metab(specs(mm\_name('bayes', err\_proc\_iid=FALSE),

n\_cores=3, n\_chains=3, burnin\_steps=300, saved\_steps=100, verbose=FALSE),

data=dat)

predict\_metab(mm)

# view the Stan model file as stored on your system

file.edit(get\_specs(mm)$model\_path)

## End(Not run)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| metab\_night {streamMetabolizer} | R Documentation |

Nighttime regression for K estimation

**Description**

Fits a model to estimate K from nighttime input data on DO, temperature, light, etc. The default day start & end are 12 noon on the preceding to present day; the algorithm then filters the data to just those time points for which light is very low. Discharge is only used, if at all, to identify and exclude days with any negative discharge.

**Usage**

metab\_night(

specs = specs(mm\_name("night")),

data = mm\_data(solar.time, DO.obs, DO.sat, depth, temp.water, light, discharge,

optional = "discharge"),

data\_daily = mm\_data(NULL),

info = NULL

)

**Arguments**

|  |  |
| --- | --- |
| specs | a list of model specifications and parameters for a model. Although this may be specified manually (it's just a list), it is easier and safer to use [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) to generate the list, because the set of required parameters and their defaults depends on the model given in the model\_nameargument to specs. The help file for [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) lists the necessary parameters, describes them in detail, and gives default values. |
| data | data.frame (not a tbl\_df) of input data at the temporal resolution of raw observations (unit-value). Columns must have the same names, units, and format as the default. The solar.time column must also have a timezone code ('tzone' attribute) of 'UTC'. See the **'Formatting data'**section below for a full description. |
| data\_daily | data.frame containing inputs with a daily timestep. See the **'Formatting data\_daily'** section below for a full description. |
| info | any information, in any format, that you would like to store within the metab\_model object |

**Value**

A metab\_night object containing the fitted model. This object can be inspected with the functions in the [metab\_model\_interface](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_model_interface).

**Author(s)**

Alison Appling, Maite Arroita, Bob Hall

**See Also**

Other metab\_model: [metab\_Kmodel](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_Kmodel), [metab\_bayes](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_bayes), [metab\_mle](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_mle), [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)

**Examples**

dat <- data\_metab('3', day\_start=12, day\_end=35)

mm <- metab\_night(data=dat)

predict\_metab(mm)

## Not run:

plot\_DO\_preds(predict\_DO(mm))

## End(Not run)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| metab\_night {streamMetabolizer} | R Documentation |

Nighttime regression for K estimation

**Description**

Fits a model to estimate K from nighttime input data on DO, temperature, light, etc. The default day start & end are 12 noon on the preceding to present day; the algorithm then filters the data to just those time points for which light is very low. Discharge is only used, if at all, to identify and exclude days with any negative discharge.

**Usage**

metab\_night(

specs = specs(mm\_name("night")),

data = mm\_data(solar.time, DO.obs, DO.sat, depth, temp.water, light, discharge,

optional = "discharge"),

data\_daily = mm\_data(NULL),

info = NULL

)

**Arguments**

|  |  |
| --- | --- |
| specs | a list of model specifications and parameters for a model. Although this may be specified manually (it's just a list), it is easier and safer to use [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) to generate the list, because the set of required parameters and their defaults depends on the model given in the model\_nameargument to specs. The help file for [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) lists the necessary parameters, describes them in detail, and gives default values. |
| data | data.frame (not a tbl\_df) of input data at the temporal resolution of raw observations (unit-value). Columns must have the same names, units, and format as the default. The solar.time column must also have a timezone code ('tzone' attribute) of 'UTC'. See the **'Formatting data'**section below for a full description. |
| data\_daily | data.frame containing inputs with a daily timestep. See the **'Formatting data\_daily'** section below for a full description. |
| info | any information, in any format, that you would like to store within the metab\_model object |

**Value**

A metab\_night object containing the fitted model. This object can be inspected with the functions in the [metab\_model\_interface](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_model_interface).

**Author(s)**

Alison Appling, Maite Arroita, Bob Hall

**See Also**

Other metab\_model: [metab\_Kmodel](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_Kmodel), [metab\_bayes](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_bayes), [metab\_mle](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_mle), [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)

**Examples**

dat <- data\_metab('3', day\_start=12, day\_end=35)

mm <- metab\_night(data=dat)

predict\_metab(mm)

## Not run:

plot\_DO\_preds(predict\_DO(mm))

## End(Not run)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]

|  |  |
| --- | --- |
| metab\_Kmodel {streamMetabolizer} | R Documentation |

Combine a time series of K estimates to predict consistent values

**Description**

Takes daily estimates of K, usually from nighttime regression, and regresses against predictors such as discharge.daily. Returns a metab\_Kmodel object that only predicts daily K, nothing else.

**Usage**

metab\_Kmodel(

specs = specs(mm\_name("Kmodel")),

data = mm\_data(solar.time, discharge, velocity, optional = c("all")),

data\_daily = mm\_data(date, K600.daily, K600.daily.lower, K600.daily.upper,

discharge.daily, velocity.daily, optional = c("K600.daily.lower", "K600.daily.upper",

"discharge.daily", "velocity.daily")),

info = NULL

)

**Arguments**

|  |  |
| --- | --- |
| specs | a list of model specifications and parameters for a model. Although this may be specified manually (it's just a list), it is easier and safer to use [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) to generate the list, because the set of required parameters and their defaults depends on the model given in themodel\_name argument to specs. The help file for [specs](http://127.0.0.1:9671/help/library/streamMetabolizer/help/specs) lists the necessary parameters, describes them in detail, and gives default values. |
| data | data.frame (not a tbl\_df) of input data at the temporal resolution of raw observations (unit-value). Columns must have the same names, units, and format as the default. The solar.time column must also have a timezone code ('tzone' attribute) of 'UTC'. See the **'Formatting data'** section below for a full description. |
| data\_daily | data.frame containing inputs with a daily timestep. See the **'Formatting data\_daily'** section below for a full description. |
| info | any information, in any format, that you would like to store within the metab\_model object |

**Details**

Possible approaches:

* "mean"Predict K as the mean of all K values
* "weighted mean"Predict K as the mean of all K values, weighted by the inverse of the confidence intervals in the input K values
* "KvQ"Regress K versus Q, tending toward overall mean in ranges of Q with sparse data
* "weighted KvQ"Regress K versus Q, tending toward overall mean in ranges of Q with sparse data, weighting high-confidence K values more heavily
* "T smoother"Predict K using a loess or spline smoother over time
* "Q smoother"Predict K using a loess or spline smoother over discharge.daily
* "TQ smoother"Predict K using a loess or spline smoother over both time and discharge.daily

**Value**

A metab\_Kmodel object containing the fitted model. This object can be inspected with the functions in the [metab\_model\_interface](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_model_interface).

**Author(s)**

Alison Appling

**See Also**

Other metab\_model: [metab\_bayes](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_bayes), [metab\_mle](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_mle), [metab\_night](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_night), [metab\_sim](http://127.0.0.1:9671/help/library/streamMetabolizer/help/metab_sim)

**Examples**

library(dplyr)

# create example data

set.seed(24842)

example\_Ks <- data.frame(date=seq(as.Date("2012-08-15"),as.Date("2012-09-15"),

as.difftime(1,units='days')), discharge.daily=exp(rnorm(32,2,1)), K600.daily=rnorm(32,30,4)) %>%

mutate(K600.daily.lower=K600.daily-5, K600.daily.upper=K600.daily+6)

# mean

mm <- metab\_Kmodel(

specs(mm\_name('Kmodel', engine='mean')),

data\_daily=example\_Ks) # two warnings expected for engine='mean'

get\_params(mm)

## Not run:

plot(get\_params(mm)$date, get\_params(mm)$K600.daily)

## End(Not run)

# linear model

mm <- metab\_Kmodel(

specs(mm\_name('Kmodel', engine='lm'), predictors='discharge.daily'),

data\_daily=example\_Ks)

get\_params(mm)

## Not run:

plot(get\_data\_daily(mm)$discharge.daily, get\_params(mm)$K600.daily)

## End(Not run)

# loess

mm <- metab\_Kmodel( ### breaks ###

specs(mm\_name('Kmodel', engine='loess'), predictors='date', other\_args=list(span=0.4)),

data\_daily=example\_Ks)

get\_params(mm)

## Not run:

plot(get\_params(mm)$date, get\_params(mm)$K600.daily)

## End(Not run)

## 3-phase workflow (sort of like complete pooling) for estimating K within

## days, then K across days, then GPP and ER within days

# 1. data and specifications for both of the MLE models

dat <- data\_metab('10','15')

mle\_specs <- specs(mm\_name('mle'))

# fit a first-round MLE and extract the K estimates

mm1 <- metab\_mle(mle\_specs, data=dat)

K600\_mm1 <- get\_params(mm1, uncertainty='ci') %>%

select(date, K600.daily, K600.daily.lower, K600.daily.upper)

# smooth the K600s

mm2 <- metab\_Kmodel(specs(mm\_name('Kmodel', engine='mean'),

day\_start=-1, day\_end=23), data\_daily=K600\_mm1)

K600\_mm2 <- get\_params(mm2) %>% select(date, K600.daily)

# refit the MLE with fixed K

mm3 <- metab\_mle(mle\_specs, data=dat, data\_daily=K600\_mm2)

get\_params(mm3, fixed='stars')

predict\_metab(mm3)

## Not run:

plot\_metab\_preds(mm1)

plot\_metab\_preds(mm3)

## End(Not run)

[Package *streamMetabolizer* version 0.11.4 [Index](http://127.0.0.1:9671/help/library/streamMetabolizer/html/00Index.html)]