

Package: pkident (via r-universe)

October 19, 2026

Title Structural and Practical Identifiability for Pharmacokinetic and Pharmacodynamic Models

Version 0.1.0

Description Determines which parameters of pharmacokinetic and pharmacodynamic ordinary differential equation models can be identified from a given dosing and observation design, and which added measurement would make the others identifiable. Decisions are local and are stated as such.

License GPL-3

URL <https://github.com/DaikiKumakura/pkident>

BugReports <https://github.com/DaikiKumakura/pkident/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports rxode2, stats, symengine, tibble, utils

Suggests knitr, nlmixr2est, nlmixr2extra, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs cmake libgmp3-dev make libmpfr-dev

Repository <https://daikikumakura.r-universe.dev>

Date/Publication 2026-10-19 14:39:30 UTC

RemoteUrl <https://github.com/DaikiKumakura/pkident>

RemoteRef HEAD

RemoteSha b0b8f5354ee8abb88cc6caea90241a7c238afaa5

Contents

as_table.pkident_designs	2
bolus_dose	3
classify_profiles	4
compare_designs	5
experiment	7
pkpd_model	8
practical_identifiability	9
reference_models	11
residual_error	12
structural_identifiability	12
Index	15

as_table.pkident_designs
<i>Extract result tables</i>

Description

Extract result tables

Usage

```
## S3 method for class 'pkident_designs'
as_table(x, what = c("summary", "parameters"), ...)

## S3 method for class 'pkident_practical'
as_table(x, what = c("parameters", "directions", "correlation"), ...)

## S3 method for class 'pkident_profiles'
as_table(x, what = c("parameters", "profiles"), ...)

as_table(x, what, ...)

## S3 method for class 'pkident_structural'
as_table(x, what = c("parameters", "combinations", "rank", "agreement"), ...)
```

Arguments

x	A pkident result.
what	Which table.
...	Unused.

Value

A tibble.

bolus_dose	<i>Dose specifications</i>
------------	----------------------------

Description

Describe how a known dose enters the model.

Usage

```
bolus_dose(state, amount = "DOSE")
```

```
infusion_dose(state, rate = "RATE")
```

Arguments

state	Name of the state that receives the dose.
amount	Amount at time zero, as a character expression.
rate	Zero-order input rate, as a character expression.

Details

- `bolus_dose()` sets the initial amount of a state (an instantaneous dose at time zero), for example `bolus_dose("depot", "F*DOSE")`.
- `infusion_dose()` adds a known zero-order input rate to a state, for example `infusion_dose("central", "RATE")`.

Symbols used in amount or rate that are not states become parameters unless they are declared as known in `pkpd_model()`.

Value

An object of class `pkident_dose`.

Examples

```
bolus_dose("depot", "F*DOSE")  
infusion_dose("central", "RATE")
```

classify_profiles	<i>Classify likelihood profiles of an nlmixr2 fit</i>
-------------------	---

Description

Fixes each selected population parameter at values on either side of its estimate, re-estimates all other parameters with `nlmixr2extra::profileFixed()`, and classifies the change in objective function value (dOFV) following Raue et al. (2009, Bioinformatics 25:1923-1929):

Usage

```
classify_profiles(
  fit,
  which = NULL,
  grid = NULL,
  threshold = 3.84,
  steps = c(0.1, 0.25, 0.5, 1, 2),
  flat_tol = 1,
  structural = NULL,
  map = NULL
)
```

Arguments

<code>fit</code>	An <code>nlmixr2</code> fit (FOCEi or another estimation method supported by <code>nlmixr2extra::profileFixed()</code>).
<code>which</code>	Names of population parameters (thetas) to profile. By default all estimated thetas that are not residual error parameters.
<code>grid</code>	Optional named list of values (estimation scale) per parameter; replaces the step-wise search for that parameter.
<code>threshold</code>	dOFV threshold; 3.84 is the 95% point of a chi-squared distribution with one degree of freedom.
<code>steps</code>	Distances from the estimate, explored on each side in order.
<code>flat_tol</code>	dOFV below which a profile counts as flat (allows for the convergence noise of repeated estimation).
<code>structural</code>	Optional result of <code>structural_identifiability()</code> .
<code>map</code>	Named character vector from fit parameter names to pkident parameter names, for example <code>c(tc1 = "CL", tv = "V")</code> .

Details

- `identifiable`: dOFV exceeds threshold on both sides. An approximate profile-likelihood confidence interval is reported (interpolated on the signed-root scale; use `profile(fit, method = "llp")` from `nlmixr2extra` for a precise interval).

- `non_identifiable`: dOFV exceeds the threshold on one side only (the open direction is reported), or on neither side. A profile that stays within `flat_tol` of its minimum is reported as flat, which suggests structural non-identifiability; otherwise it is reported as shallow.
- `unresolved`: a profile fit failed.

Statements are limited to the explored range, which is reported.

Every profile fit is an upper bound of the profile likelihood: a failed or stuck re-estimation can only overstate dOFV. Each point more than `flat_tol` above the original fit, or above the lowest OFV found, is therefore re-estimated from the estimates of a neighbouring profile point (a warm start), and the lower value is kept. If profiling finds a lower OFV than the original fit, dOFV is measured from that lower value and the drop is reported (`ofv_drop`): the original fit was not at its minimum, which is common along non-identifiable directions.

By default, each side is explored by stepping away from the estimate by steps on the estimation scale, stopping once dOFV exceeds the threshold. The default steps suit log-transformed parameters (`cl <- exp(tcl + ...)`); for other parameters give grid.

If a `structural_identifiability()` result is given with `map`, each profile is placed next to the structural decision, which distinguishes "structurally identifiable but not determined by these data" from structural non-identifiability.

Value

An object of class `pkident_profiles`. Use `as_table()` with "parameters" or "profiles".

Examples

```
## Not run:
fit <- nlmixr2(one_compartment, data, est = "focei")
classify_profiles(fit, which = c("tcl", "tv"))

## End(Not run)
```

compare_designs

Compare candidate designs

Description

For each candidate design supplied by the user (for example the current design, the same design with an added measured output, with added sampling times, or with an added dose level), reports the structural decision for the outputs it observes and the expected relative standard errors from the Fisher information summed over its experiments. No design is optimised; only the supplied candidates are compared.

Usage

```
compare_designs(
  m,
  candidates,
  values,
  error = residual_error(prop = 0.1),
  inputs = list(),
  structural = TRUE,
  method = c("both", "lie", "sensitivity"),
  rse_limit = 30,
  seed = 1L,
  rtol = 1e-10,
  atol = 1e-12
)
```

Arguments

<code>m</code>	A <code>pkident_model</code> from <code>pkpd_model()</code> that defines every output that any candidate observes.
<code>candidates</code>	A named list of candidates; each is an <code>experiment()</code> or a list of experiments.
<code>values</code>	Named values of all parameters and known constants (the design point).
<code>error</code>	A <code>residual_error()</code> for every output, or a named list with one per output.
<code>inputs</code>	Known input signals shared by all experiments.
<code>structural</code>	If TRUE, run <code>structural_identifiability()</code> for each distinct set of observed outputs.
<code>method</code>	Method passed to <code>structural_identifiability()</code> .
<code>rse_limit</code>	RSE (%) above which a parameter is unresolved.
<code>seed</code>	Seed for the structural analysis.
<code>rtol, atol</code>	Solver tolerances.

Details

The structural decision depends only on which outputs are observed (it assumes continuous, noise-free observation after the doses of the model definition). Added sampling times and dose levels therefore change only the expected RSEs.

Value

An object of class `pkident_designs`. Use `as_table()` with "summary" or "parameters".

Examples

```
m <- pkpd_model(
  "d/dt(central) = -CL/V*central
  cp = central/V",
  outputs = "cp", doses = bolus_dose("central", "DOSE"), known = "DOSE"
```

```
)
compare_designs(
  m,
  candidates = list(
    early = experiment(c(0.25, 0.5, 1, 2), n = 10),
    extended = experiment(c(0.25, 0.5, 1, 2, 8, 24), n = 10)
  ),
  values = c(CL = 2, V = 20, DOSE = 100),
  error = residual_error(add = 0.05, prop = 0.1),
  structural = FALSE
)
```

experiment	<i>One experiment of a design</i>
------------	-----------------------------------

Description

A group of individuals with the same dosing and sampling.

Usage

```
experiment(times, events = NULL, values = NULL, n = 1, inputs = NULL)
```

Arguments

- | | |
|--------|---|
| times | Sampling times: a numeric vector used for every output, or a named list with one vector per observed output. Outputs not named are not observed in this experiment. |
| events | Optional dosing event table (see practical_identifiability()). |
| values | Optional named values that override the design values for this experiment, typically known constants such as a dose DOSE. |
| n | Number of individuals. |
| inputs | Optional known input signals for this experiment (see practical_identifiability()). |

Value

An object of class pkident_experiment.

Examples

```
experiment(c(0.5, 1, 2, 4, 8, 24), values = c(DOSE = 100), n = 12)
```

pkpd_model

*Define a model for identifiability analysis***Description**

Reads an ODE model through rxode2 and records, as symbolic expressions, the state equations, the observed outputs, the initial conditions, the known doses and input signals, and the unknown parameters. No identifiability decision is made here.

Usage

```
pkpd_model(
  model,
  outputs,
  doses = list(),
  inputs = character(),
  known = character(),
  parameters = NULL,
  initial = list()
)
```

Arguments

model	An rxode2 model: a compiled rxode2 object, model code as a character string, or a model function with <code>ini()</code> and <code>model()</code> blocks (residual error lines containing <code>~</code> are ignored).
outputs	Names of the observed quantities: variables assigned in the model or state names.
doses	A list of <code>bolus_dose()</code> and <code>infusion_dose()</code> specifications.
inputs	Names of known input signals (for example a plasma concentration C_p driving a pharmacodynamic model). Their values and time derivatives are treated as known.
known	Names of known constants (for example DOSE).
parameters	Names of the unknown parameters to analyse. By default, every symbol that is not a state, an input, a known constant or time.
initial	A named list of initial conditions as character expressions, for example <code>list(E = "kin/kout")</code> . States without a dose or an initial condition start at zero.

Value

An object of class `pkident_model`.

Examples

```
m <- pkpd_model(
  "d/dt(depot) = -ka*depot
  d/dt(central) = ka*depot - CL/V*central
  cp = central/V",
  outputs = "cp",
  doses = list(bolus_dose("depot", "F*DOSE")),
  known = "DOSE"
)
m
```

practical_identifiability

Practical identifiability for a given design

Description

Computes the expected Fisher information matrix of the unknown parameters for one typical individual (or n individuals with the same design), given the sampling times, the parameter values and the residual error model. The output sensitivities are obtained by solving the forward sensitivity equations with `rxode2`. Because the residual standard deviation may depend on the prediction (proportional error), the information includes the term from the variance as well as the term from the mean.

Usage

```
practical_identifiability(
  m,
  design,
  values,
  error = residual_error(prop = 0.1),
  inputs = list(),
  n = 1,
  events = NULL,
  rse_limit = 30,
  rtol = 1e-10,
  atol = 1e-12
)
```

Arguments

<code>m</code>	A <code>pkident_model</code> from <code>pkpd_model()</code> .
<code>design</code>	Sampling times: a numeric vector used for every output, or a named list with one vector per output.
<code>values</code>	Named numeric vector with a value for every parameter and every known constant of the model.

error	A <code>residual_error()</code> used for every output, or a named list with one per observed output.
inputs	For models with known input signals, a named list of character expressions in time t , for example <code>list(Cp = "10*exp(-0.2*t)")</code> .
n	Number of individuals with the same design (the information is multiplied by n).
events	Optional dosing event table (a data frame with columns <code>time</code> , <code>amt</code> , <code>cmt</code> and optionally <code>rate</code>), for example repeated infusions. It replaces the doses of the model definition; dose amounts and rates must be known.
rse_limit	RSE (%) above which a parameter is unresolved.
rtol, atol	Solver tolerances.

Details

The result reports, for each parameter, the expected standard error and relative standard error (RSE, %), the correlation matrix, and the eigen-decomposition of the information matrix on the log-parameter scale: the weakest directions show which combinations of parameters the design determines poorly, and the condition number summarises how poorly.

Statuses:

- `identifiable`: the RSE is finite and at most `rse_limit`.
- `non_identifiable`: the information matrix is singular in a direction involving the parameter; the design cannot determine it (structurally or because of too few samples).
- `unresolved`: the RSE is finite but above `rse_limit`. The expected information is a local, linear approximation that is unreliable when uncertainty is this large; examine likelihood profiles instead.

The default limit of 30% comes from the validation by repeated estimation on simulated data (`validation/fim-simulation.R`): with every RSE below about 15%, expected and observed RSEs agreed within 10%; with RSEs of 30-40%, estimates had skewed tails and the expected RSEs underestimated the spread, including that of a parameter whose own expected RSE was small. When any parameter is unresolved, the printed result therefore warns that the RSEs of all parameters are approximate.

Doses are those in the model (bolus doses at time zero and constant-rate infusions), whose amounts and rates are known constants supplied in values, or those of an event table given in events.

Value

An object of class `pkident_practical`. Use `as_table()` with `"parameters"`, `"directions"` or `"correlation"`.

Examples

```
m <- reference_model("A3")
p <- practical_identifiability(
  m,
  design = c(0.5, 1, 2, 4, 8, 12, 24),
```

```
values = c(ka = 1, CL = 2, V = 20, DOSE = 100),  
error = residual_error(add = 0.05, prop = 0.1)  
)  
p
```

reference_models	<i>Reference models with known identifiability</i>
------------------	--

Description

The acceptance models used to validate pkident. Each has an expected result fixed from a published table or an analytical derivation before any identifiability code was written (see the package's validation record).

Usage

```
reference_models()  
  
reference_model(id)
```

Arguments

id	Model identifier, for example "A2".
----	-------------------------------------

Details

reference_models() lists them; reference_model() builds one as a `pkpd_model()` with the expected result attached as attribute expected.

Value

reference_models() returns a data frame; reference_model() returns a pkident_model.

References

Janzén DLI, et al. Parameter identifiability of fundamental pharmacodynamic models. Front Physiol. 2016;7:590. doi:10.3389/fphys.2016.00590

Cheung SYA, Yates JWT, Aarons L. The design and analysis of parallel experiments to produce structurally identifiable models. J Pharmacokinet Pharmacodyn. 2013;40:93-100. doi:10.1007/s109280129291z

Examples

```
reference_models()  
reference_model("A2")
```

residual_error	<i>Residual error model</i>
----------------	-----------------------------

Description

Measurement error of an output. With add and prop: $sd = \sqrt{add^2 + (prop * f)^2}$, where f is the model prediction (the combined2 form of nlmixr2); use add alone for additive error and prop alone for proportional error. With exp: exponential error, that is additive error of standard deviation exp on the log scale (data fitted as $\log(y)$). Exponential error is not combined with the others.

Usage

```
residual_error(add = 0, prop = 0, exp = 0)
```

Arguments

add	Additive standard deviation (in the units of the output).
prop	Proportional standard deviation (for example 0.1 for 10%).
exp	Standard deviation on the log scale (exponential error).

Details

Proportional and exponential errors differ in the Fisher information: with proportional error the variance depends on the parameters and carries information of its own; with exponential error it does not.

Value

An object of class pkident_residual.

Examples

```
residual_error(prop = 0.1)
residual_error(add = 0.05, prop = 0.1)
residual_error(exp = 0.2)
```

structural_identifiability	<i>Structural identifiability</i>
----------------------------	-----------------------------------

Description

Decides local structural identifiability of the unknown parameters of a `pkpd_model()` with two independent methods.

Usage

```
structural_identifiability(
  m,
  method = c("both", "lie", "sensitivity"),
  points = 5L,
  seed = 1L,
  max_order = NULL,
  plateau = 3L,
  tol = 1e-07,
  max_seconds = 300,
  max_exponent = 2L,
  max_support = 4L,
  times = NULL,
  sens_tol = 1e-07,
  rtol = 1e-10,
  atol = 1e-12
)
```

Arguments

<code>m</code>	A <code>pkident_model</code> from <code>pkpd_model()</code> .
<code>method</code>	"both" (Lie method checked by sensitivities), "lie" or "sensitivity".
<code>points</code>	Number of random parameter points.
<code>seed</code>	Random seed for the points (the global random state is restored).
<code>max_order</code>	Highest derivative order of the Lie method. By default the number of parameters plus the number of states plus two, capped at 14.
<code>plateau</code>	Number of consecutive orders without a rank increase needed to conclude rank deficiency (Lie method).
<code>tol</code>	Relative singular value below which a direction counts as null (Lie method).
<code>max_seconds</code>	Time limit of the Lie method; beyond it the result is unresolved.
<code>max_exponent, max_support</code>	Search limits for monomial combinations.
<code>times</code>	Time grid of the sensitivity method. By default 0 and 150 log-spaced times from 0.001 to 200, which covers the rate constants implied by the random parameter points (between 0.2 and 5).
<code>sens_tol</code>	Relative singular value below which a direction counts as null (sensitivity method).
<code>rtol, atol</code>	Solver tolerances of the sensitivity method.

Details

Lie method (primary). The outputs are differentiated along the model (Lie derivatives), with known input signals contributing their own time derivatives, and evaluated at the known initial state at time zero (the Taylor-series approach). The parameters are locally identifiable if the Jacobian of these output derivatives with respect to the parameters has full rank. Using the known initial state matters for pharmacokinetic models: a bolus dose fixes the initial amount, which is often what

makes a volume identifiable. Methods that evaluate the rank at generic initial states ignore this information.

Sensitivity method (numerical check). The forward sensitivity equations are generated symbolically and solved with rxode2 at the same parameter points; known input signals are replaced by a smooth random signal. The rank of the noise-free output sensitivity matrix on a dense time grid is compared with the Lie rank, and the null spaces of the two methods are compared at every point.

With method = "both" (the default), a decision is made only if the two methods agree at every point; otherwise the result is unresolved. Ranks are evaluated at several random parameter points on a log scale, and a decision is made only if all points agree.

Rank-deficient parameters are reported with the directions along which they can change together without changing the outputs. Where these directions correspond to products of powers of parameters (for example V/F), the identifiable combinations are reported.

Decisions are **local**: a parameter reported as identifiable may still have a finite number of alternative values elsewhere in parameter space (for example the flip-flop of absorption and elimination rates).

Value

An object of class pkident_structural. Use `as_table()` to extract the "parameters", "combinations", "rank" (Lie method) and "agreement" (comparison of the methods at each point) tables.

Examples

```
r <- structural_identifiability(reference_model("A2"), method = "lie")
r
as_table(r, "combinations")
```

Index

`as_table(as_table.pkident_designs)`, 2
`as_table()`, 5, 6, 10, 14
`as_table.pkident_designs`, 2

`bolus_dose`, 3
`bolus_dose()`, 8

`classify_profiles`, 4
`compare_designs`, 5

`experiment`, 7
`experiment()`, 6

`infusion_dose(bolus_dose)`, 3
`infusion_dose()`, 8

`pkpd_model`, 8
`pkpd_model()`, 3, 6, 9, 11–13
`practical_identifiability`, 9
`practical_identifiability()`, 7

`reference_model(reference_models)`, 11
`reference_models`, 11
`residual_error`, 12
`residual_error()`, 6, 10

`structural_identifiability`, 12
`structural_identifiability()`, 4–6