Package 'supeRcrit'

April 1, 2025

Title Supercritical CO2 and Subcritical Water Process Development

Version 0.9.0

Description Provides tools for kinetic modeling, systematic co-solvent choice,
process optimization via experimental design and response surface methodology (RSM),
and Cost of Manufacturing (COM) analysis of supercritical CO2 and subcritical water processes.

License MIT + file LICENSE

URL https://github.com/Deniz-Koseoglu/supeRcrit

Depends R (>= 4.0.0)

Imports utils,
stats,
graphics,
grDevices,
stringr (>= 1.5.1),
grid (>= 4.3.2),
gridExtra (>= 2.3),
webchem (>= 1.3.0),
rcdk (>= 3.8.1),
data.table (>= 1.14.8),
ggpmisc (>= 0.5.4.1),
ggplot2 (>= 3.4.4),
nloptr (>= 2.0.3),
robustbase (>= 0.99.2),
segmented (>= 2.1.0),
factoextra (>= 1.0.7),
cluster (>= 2.1.6),
pracma (>= 2.4.4),
desirability2 (>= 0.0.1),
ggrepel (>= 0.9.6),
rJava (>= 1.0.11),
depict (>= 0.4.0),
scales (>= 1.3.0)

Suggests knitr,
R.rsp,
rmarkdown,
testthat (>= 3.0.0)

Remotes CDK-R/depict

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr, R.rsp

LazyData true

BugReports https://github.com/Deniz-Koseoglu/supeRcrit/issues

Contents

`add_cpts` *Add center points to an experimental design.*

Description

Part of the **experimental design generation** workflow.

Usage

```
add_cpts(input, cpts = 3)
```

Arguments

`input`	A `data.frame` containing the **coded** input factors and their levels. Columns (factor names) must be named alphabetically.
`cpts`	The number of center points to add (defaults to 3).

Value

The experimental design `data.frame` including the newly added center points.

See Also

`doe_frfd`, `doe_ffd`, `doe_ccd`, `doe_bbd`, `doe_tm`

Examples

```
#Load data
doe_base <- load_internal("doe_base")

#Get experimental design data.frame
doe_df <- doe_base[["FD_3"]]

#Add center points
res <- add_cpts(doe_df, 3)
```

add_stars *Add star points to an experimental design*

Description

Part of the **experimental design generation** workflow.

Usage

```
add_stars(input, factors, design = "CCC")
```

Arguments

input	A data.frame containing the **coded** input factors and their levels. Columns (factor names) must be named alphabetically.
factors	The number of factors (**between 2 and 4**).
design	The type of Central Composite experimental design. Currently supported designs are Circumscribed ("CCC", default) or face-centered ("CCF").

Value

The experimental design data.frame including the newly added star points.

See Also

doe_frfd, doe_ffd, doe_ccd, doe_bbd, doe_tm

Examples

```
#Load data
doe_base <- load_internal("doe_base")

#Get experimental design data.frame
doe_df <- doe_base[["FD_3"]]

#Add center points
res <- add_stars(doe_df, 3, "CCC")
```

bendens *Calculate CO2 density using Bender EoS*

Description

Uses the Bender Equation of State to calculate CO2 density (g/L or g/mL) and specific enthalpy (kJ/kg) between 1-1000 bar and -50 to 300 degress Celsius. Part of the bicmod workflow.

Usage

```
bendens(pres, temp, units = "g/L")
```

Arguments

`pres, temp`	Pressure (in bar) and temperature (in degrees Celsius) for which to return the CO2 density.
`units`	Units required for output density as a `character` value. One of: `"g/L"` or `"g/mL"`.

Details

For the complete Bender Equation of State used to calculate density and enthalpy herein, see **Eqs. 1.1-1.4** in Rizza (2014).

Value

A named `numeric` vector containing the density (`["rho"]`) in chosen `units` and the enthalpy

References

Bender, E. (1975), 'Equations of state for ethylene and propylene', *Cryogenics* **15** (11), pp. 667-673, DOI: https://doi.org/10.1016/0011-2275(75)90100-9.

Ghazouani, J., Chouaieb, O., Bellagi, A. (2005), 'Evaluation of the parameters of the Bender equation of state for low acentric factor fluids and carbon dioxide', *Thermochimica Acta* **432** (1), pp. 10-19, DOI: https://doi.org/10.1016/j.tca.2004.11.008.

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

Sievers, U. (1984), 'Die thermodynamischen Eigenschaften von Kohlendioxid', *Forsch Ing-Wes* **50**, p. 192, DOI: https://doi.org/10.1007/BF02560841.

See Also

`bicmod`

Examples

```
bendens(300, 45)
bendens(800, 70, "g/mL")
```

`bicmod` *Construct BIC models of varying complexity*

Description

Constructs various Broken and Intact Cells (BIC) kinetic models from overall extraction curves (OECs) of supercritical CO2, subcritical water, or any other process. Consult **References** and the **See Also** section for further information. **This workflow is currently a Work in Progress (WIP)**.

Usage

```
bicmod(
  oec,
  oec_vars,
  pars,
  opt_est = "default",
  etoh_frac = 0,
  flowpar = rep(NA, 2),
  ro_co2 = NA,
  tmax = NA,
  qmax = NA,
  cumulative = FALSE,
  mass_flow = FALSE,
  draw = TRUE,
  aggreg = "aard",
  modtype = "all",
  units = "default"
)
```

Arguments

oec	A data.frame of OEC data. Must include all oec_vars.
oec_vars	A **named** character vector of column names included in oec. These **must** include the extraction time ("x") and response ("y") with appropriate units. A third parameters indicating the solvent usage ("slv") is **optional** and may be provided as an **alternative** to an explicit flow rate normally provided in pars["flow"] - this is useful when the flow rate is uneven throughout the extraction process.
pars	A named numeric vector of input parameters for the model, divided into **mandatory** and **optional** parameters. **Mandatory** parameters include pressure ("pres"; bar), temperature ("temp"; degC), mass of material loaded ("mass_in"; g), moisture content ("moisture"; %), diameter ("D"; m) and length ("L"; m) of the extraction vessel, the fraction of ethanol co-solvent ("etoh"), the real ("dr"; g/L) and apparent ("dp"; g/L) densities of the raw material, and the number of observations/OEC points corresponding to the end of the CER ("n"). The **optional** variables default to NA and include the flow rate of solvent ("flow"; units set in units), and the maximum extractable material fraction ("cu"), which **must be included unless** modtype is set to "cu".
opt_est	Either "default" or a **named** numeric vector of initial parameter estimates for iterative optimization. May include any of grinding efficiency r ("r"), the product of the solid phase mass transfer coefficient (k_s) and the specific area between intact and broken cells (a_s) $k_s a_s$ ("ksas"), external mass transfer coefficient θ_e ("thetaf"), extraction duration of FER t_i ("ti"), the fluid phase mass transfer coefficient k_f ("kf"), and/or the relative amount of solvent expended at the end of CER q_m ("qc"). An additional parameter "c3" is related to the maximum extractable material fraction c_u and is **only required** when modtype is set to "cu". Any values not specified will be set to the following defaults: c(r = 0.4, ksas = 10e-5, qc = 400, thetaf = 1, ti = 30, kf = 0.001, c3 = 0.15).
etoh_frac	The single numeric fraction of ethanol co-solvent (between 0-0.99). Defaults to 0. Must be non-zero if **flow rate** is not provided in pars and pars["etoh"] is **non-zero**.

`flowpar`	Either `NA` (default) or a `numeric` vector of length 2 providing temperature and pressure at which flow rate of CO2 is measured.
`ro_co2`	The supercritical CO2 density (in g/L). If not provided (`NA`; default), it is calculated via the Bender Equation of State (see `bendens`).
`tmax`	Maximum x-axis value (time, min) to use for model predictions **exclusively** for `modtype = "ct"`. Setting to `NA` defaults to 120% of the maximum experimental value.
`qmax`	Maximum x-axis value (solvent expended, kg/kg insoluble solid) to use for model predictions for all models except `modtype = "ct"`. Setting to `NA` defaults to 120% of the maximum experimental value.
`cumulative`	A `logical` switch specifying whether the **response** and/or **solvent consumption** values provided in `oec` are cumulative or not (defaults to `FALSE`).
`mass_flow`	A `logical` indicating whether the flow rate provided in `pars` is **mass** or **volumetric** (`FALSE`; default).
`draw`	A `logical` switch. Should generated plots be plotted? Defaults to `TRUE`.
`aggreg`	A string specifying how the "best" results of non-linear optimization (`nlsm`) should be chosen. One of `"aard"` (default; value with minimum Average Absolute Relative Deviation is chosen) or `"mean"` (the arithmetic mean of all results is taken).
`modtype`	Specifies the type(s) of BIC models to generate. Either `"all"` (default) or one or more of: `"sim"` (simplified), `"ct"` (characteristic times), `"cmp3"` (complete with 3 OEC regions), and/or `"cmp2"` (complete with 2 OEC regions). A separate option `"cu"` is used when the extractable material fraction c_u is to be estimated.
`units`	A **named** `character` vector of length 1 or 2 specifying the units of `"flow"` (one of `"mL/min"`, the **default** `"g/min"`, `"kg/h"`, `"L/h"`, or `"none"`) and/or the response `"resp"` (one of `"g"`, the **default** `"percent"`, `"permille"`, `"ppm"`, or `"ppb"`). Where not provided, default values are used.

Details

This workflow derives and visualizes three different Broken and Intact Cells (BIC) kinetic models based on the work of Sovova (2005, 2012, 2017) and previously utilized by Rizza (2014) and many other authors. These include the **simplified**, **characteristic times**, and **complete** models - descriptions of associated parameters may be found in the descriptions of `bic_sm`, `bic_ct`, and `bic_cmp`, respectively, and their linked functions.

Value

The output takes two forms. When `modtype` includes `"cu"` (for estimation of the maximum extractable fraction c_u), the output is a `list` of the estimated c_u value (`$cu`) and the iteratively derived final estimate of the related constant **c3** (`$mod_coefs`). Alternatively, when `modtype` includes any of `c("sim","ct","cmp3","cmp2")`, output is a `list` containing the following elements:

1. **$data**: A `data.frame` of original `oec` input data with added values of Solvent-Material (S/M) ratio `$q`, as well as fractional yields `$e` (g/g insoluble solid) and `$estar` (g/g total dry solid used for `modtype == "ct"`).
2. **$sim, $ct, and/or $cmp**: Each a `list` of results for the eponymous BIC model, with output structure outlined in the documentation of `bic_sm`, `bic_ct`, and `bic_cmp`, respectively.
3. **$plots**: A `list` of plot objects of class `"ggplot"`.

4. **$input**: A named `numeric` vector of input parameters. In addition to those provided in argument `pars`, some calculated parameters are also included. These are the specific surface area per unit volume of extraction bed a_0 (`"a0"`; 1/m), the apparent porosity ε (`"porosity"`), the CO2 to insoluble solid ratio in the extraction bed γ (`"gamma"`; kg/kg), the number of experimental points m (`"m"`), total **dry** mass of material N (`"Ng"`; g), mass of **insoluble** material N_m (`"Nm"`; g), the ratio of solute to insoluble material x_u (`"xu"`; kg/kg), and the apparent extract solubility y_s (`"ys"`; g/g).
5. **$call**: The function call.

References

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

Sovova, H. (2005), 'Mathematical model for supercritical fluid extraction of natural products and extraction curve evaluation', *The Journal of Supercritical Fluids* **33** (1), pp. 35-52, DOI: https://doi.org/10.1016/j.supflu.2004.03.005.

Sovova, H. (2012), 'Steps of supercritical fluid extraction of natural products and their characteristic times', *The Journal of Supercritical Fluids* **66**, pp. 73-79, DOI: https://doi.org/10.1016/j.supflu.2011.11.004.

Sovova, H. (2017), 'Broken-and-intact cell model for supercritical fluid extraction: Its origin and limits', *The Journal of Supercritical Fluids* **129**, pp. 3-8, DOI: https://doi.org/10.1016/j.supflu.2017.02.014.

See Also

bendens, etoh_dens, bic_sm, bic_ct, bic_cmp, show_pars, kin_plot, kin_splot, kin_export

Examples

```
bic_res <- bicmod(oec = sfex[[2]][["data"]],
oec_vars = c(x = "Time_min", y = "Yield_g", slv = "Solvent_mL"),
pars = c(pres = 300,
         cu = 0.165,
         temp = 45,
         flow = NA,
         mass_in = 0.5125,
         moisture = 8.6,
         D = 0.015,
         L = 0.015,
         etoh = 0.5,
         dr = 1554,
         dp = 0.0004,
         n = 2),
opt_est = "default",
flowpar = c(1.01325, 25),
etoh_frac = 0.06, #Required when CO2 flow is not provided but 'etoh' is non-zero
ro_co2 = NA,
tmax = NA,
qmax = NA,
cumulative = FALSE,
mass_flow = FALSE,
draw = TRUE,
units = c(flow = "none", resp = "g"),
modtype = "all")
```

`bic_cmp` *Derive the complete BIC model*

Description

Derives, summarises, and visualises the results of the complete Broken-and-Intact Cells (BIC) model of Sovova (2005). Part of the `bicmod` workflow.

Usage

```
bic_cmp(
  yield,
  q,
  n,
  a0,
  gam,
  porosity,
  qaver,
  Nm,
  ys,
  xu,
  r0,
  ksas0,
  kf0 = 0.001,
  qc0 = 400,
  modpts = 100,
  maxq = round(max(q) * 1.2, -1),
  nlsm = c("MM", "tau", "CM", "mtl"),
  aggreg = "aard"
)
```

Arguments

`yield`	A `numeric` vector of yields (g/g insoluble solid).
`q`	A `numeric` vector of relative amounts of expended solvent (kg/kg insoluble solid).
`n`	Period corresponding to the end of the CER (in **number of experimental points**).
`a0`	Specific surface area per unit volume of extraction bed (1/m).
`gam`	Solvent to matrix ratio in the bed ($kg_{solvent}/kg_{insolublesolid}$).
`porosity`	Bed porosity (dimensionless).
`qaver`	Average solvent flow rate (kg/s).
`Nm`	Mass of insoluble material (g).
`ys`	Extract solubility in CO2 (g/g).
`xu`	Weight fraction (concentration) in the untreated solid.
`r0`	Initial estimate of the grinding efficiency (fraction of broken cells).

ksas0	Initial estimate of the product of the solid phase mass transfer coefficient (k_s; 1/s) and the specific area between intact and broken cells (a_s; 1/m), in 1/m/s.
kf0	An initial estimate of the fluid phase mass transfer coefficient (1/s). Defaults to 0.001.
qc0	Initial estimate of solvent expended at the end of the CER (kg/kg CO2). Defaults to 400.
modpts	A single numeric value specifying how many points should be modelled throughout the OEC curve range (100 by default).
maxq	Maximum x-axis value (solvent expended, kg/kg insoluble solid) to use for model predictions. Defaults to 120% of the maximum q value.
nlsm	A character vector of non-linear optimization methods to use for convergence of the model. One or more of: "MM", "tau", "CM", and/or "mtl" (**see** nlrob **for details**).
aggreg	A string specifying how the "best" results of non-linear optimization (nlsm) should be chosen. One of "aard" (default; value with minimum Average Absolute Relative Deviation is chosen) or "mean" (the arithmetic mean of all results is taken).

Details

For a detailed description of this variant of the Broken and Intact Cells (BIC) model, refer to bicmod.

Value

A named list containing the following elements:

1. **$ordt**: A data.frame containing the original input data including the temporal variable $t, the Solvent-Solute (S/M) ratio $x, the **actual** fractional yield ($y; g/g insoluble solid), as well as yields **modeled** via the 3-period and 2-period models ($y_cmp3 and $y_cmp2, respectively).
2. **$mdt**: A data.frame of modeled data including the model type ($model), the region of the extraction curve ($period), as well as S/M ratio $x, fractional yield $y, and time ($t) up to maxq.
3. **$mod_pars**: The model parameters including the external mass transfer resistance (θ_e; "thetae"), the product of the fluid phase mass transfer coefficient k_f and specific surface area per unit volume of extraction bed a_0 ($k_f a_0$; "kfa0"), the relative amount of expended solvent (kg/kg insoluble solid) at the end of the CER (q_m; "qm"), FER for the 3-period model (q_n; "qn"), and DC for the 2-period model (q_{cmp}; "qs"), coefficient β ("beta"), initial fraction of solute in broken cells (G; "G"), the fluid mass transfer coefficient (k_f; "kf"), grinding efficiency (r; "r"), as well as the product of the solid phase mass transfer coefficient k_s and the specific area between intact and broken cells a_s ($k_s a_s$; "ksas"). See show_pars for detailed description of these parameters.
4. **$fit_pars**: A character vector of which parameters in $mod_pars were iteratively fit to the model.
5. **$resid**: A named numeric vector of various error values for the model, including Average Absolute Relative Deviation AARD ("aard"), round mean squared error RMSE ("rmse"), and the R2 value ("r2").

References

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

Sovova, H. (2005), 'Mathematical model for supercritical fluid extraction of natural products and extraction curve evaluation', *The Journal of Supercritical Fluids* **33** (1), pp. 35-52, DOI: https://doi.org/10.1016/j.supflu.2004.03.005.

Sovova, H. (2017), 'Broken-and-intact cell model for supercritical fluid extraction: Its origin and limits', *The Journal of Supercritical Fluids* **129**, pp. 3-8, DOI: https://doi.org/10.1016/j.supflu.2017.02.014.

See Also

bicmod, show_pars, moderr

`bic_ct` *Derive the simplified BIC model based on characteristic times*

Description

Derives, summarises, and visualises the results of the characteristic times Broken-and-Intact Cells (BIC) model of Sovova (2005). Part of the bicmod workflow.

Usage

```
bic_ct(
  yield,
  t,
  n,
  qaver,
  cu,
  N,
  Nm,
  ys,
  modpts = 100,
  thetaf0 = 1,
  ti0 = 30,
  maxt = round(max(t) * 1.2, -1)
)
```

Arguments

`yield`	Fractional yield of extract (g/g total dry solid).
`t`	A `numeric` vector of times at which `yield` was recorded (s).
`n`	Period corresponding to the end of the CER (in **number of experimental points**).
`qaver`	Average solvent flow rate (kg/s).
`cu`	Solute content in the untreated solid (kg/kg), i.e. maximum possible yield as a **fraction**.
`N`	Total dry mass (**kg**).

Nm	Mass of insoluble material (g).
ys	Extract solubility in CO2 (g/g).
modpts	A single numeric value specifying how many points should be modelled throughout the OEC curve range (100 by default).
thetaf0	Initial estimate of the external mass transfer resistance. Defaults to 1.
ti0	Initial estimate of the characteristic time of the solid phase mass transfer (s). Defaults to 30 s.
maxt	Maximum x-axis value (time, s) to use for model predictions. Defaults to 120% of the maximum t value.

Details

For a detailed description of this variant of the Broken and Intact Cells (BIC) model, refer to bicmod.

Value

A named list containing the following elements:

1. **$ordt**: A data.frame containing the original input data including the time variable $x, the **actual** fractional yield ($y; g/g **total dry solid**), as well as the **modeled** yield ($mod_y.
2. **$mdt**: A data.frame of modeled data including the model type ($model), the region of the extraction curve ($period), as well as the time ($t) up to maxt and fractional yield $y.
3. **$mod_pars**: The model parameters including the external mass transfer resistance (θ_e; "thetaf"), extraction time/duration of the FER (t_i; "ti"), extraction time (t') and yield (e'; g/g total dry solid) at the end of CER ("tprime" and "$eprime$"), and the initial fraction of solute in broken cells (G; "G"). ($k_s a_s$; "ksas"). See show_pars for detailed description of these parameters.
4. **$fit_pars**: A character vector of which parameters in $mod_pars were iteratively fit to the model.
5. **$resid**: A named numeric vector of various error values for the model, including Average Absolute Relative Deviation AARD ("aard"), round mean squared error RMSE ("rmse"), and the R2 value ("r2").

References

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

Sovova, H. (2005), 'Mathematical model for supercritical fluid extraction of natural products and extraction curve evaluation', *The Journal of Supercritical Fluids* **33** (1), pp. 35-52, DOI: https://doi.org/10.1016/j.supflu.2004.03.005.

Sovova, H. (2017), 'Broken-and-intact cell model for supercritical fluid extraction: Its origin and limits', *The Journal of Supercritical Fluids* **129**, pp. 3-8, DOI: https://doi.org/10.1016/j.supflu.2017.02.014.

See Also

bicmod, show_pars, moderr

`bic_sm` *Build a simplified BIC model*

Description

Derives, summarises, and visualises the results of the simplified Broken-and-Intact Cells (BIC) model of Sovova (2005). Part of the `bicmod` workflow.

Usage

```
bic_sm(
  yield,
  q,
  n,
  xu = NA,
  qaver,
  Nm = NA,
  porosity,
  ys = NA,
  c3 = NA,
  Ng = NA,
  est_cu = FALSE,
  maxq = round(max(q) * 1.2, -1),
  r0 = 0.4,
  ksas0 = 1e-04,
  qc0 = 400,
  aggreg = "aard",
  modpts = 100,
  nlsm = c("MM", "tau", "CM", "mtl")
)
```

Arguments

`yield`	A `numeric` vector of yields (e; g/g insoluble solid).
`q`	A `numeric` vector of relative amount of expended solvent (q; kg/kg).
`n`	Period corresponding to the end of the CER (n; in **number of experimental points**).
`xu`	Weight fraction (concentration) in the untreated solid x_u. Defaults to `NA`. Must be provided only when `est_cu` is `FALSE`.
`qaver`	Average solvent flow rate ($\dot{Q}$; kg/s).
`Nm`	Mass of insoluble material (N_m; g). Must be provided only when `est_cu` is `FALSE`.
`porosity`	Bed porosity (ε; dimensionless).
`ys`	Extract solubility in CO2 (y_s; g/g). May be estimated using `oec_bp`. Defaults to `NA`. Must be provided only when `est_cu` is `FALSE`.
`c3`	An initial estimate of the asymptotic extraction yield at infinite time (c_u; fraction) to be provided **only** when `est_cu` is `TRUE`.
`Ng`	Total dry mass N (solute + insoluble material; g). Defaults to `NA`. Must be provided when `est_cu` is `TRUE`.

`est_cu`	A `logical` indicating whether solute content in the untreated solid (kg/kg), i.e. maximum possible yield, should be estimated (`FALSE` by default).
`maxq`	Maximum x-axis value (solvent expended, kg/kg insoluble solid) to use for model predictions. Defaults to 120% of the maximum `q` value.
`r0`	Initial estimate of the grinding efficiency r (fraction of broken cells).
`ksas0`	Initial estimate of the product of the solid phase mass transfer coefficient (k_s; 1/s) and the specific area between intact and broken cells (a_s; 1/m), in 1/m/s.
`qc0`	Initial estimate of solvent expended at the end of the CER (q_m; kg/kg CO2).
`aggreg`	A string specifying how the "best" results of non-linear optimization (`nlsm`) should be chosen. One of `"aard"` (default; value with minimum Average Absolute Relative Deviation is chosen) or `"mean"` (the arithmetic mean of all results is taken).
`modpts`	A single `numeric` value specifying how many points should be modelled throughout the OEC curve range (`100` by default).
`nlsm`	A `character` vector of non-linear optimization methods to use for convergence of the model. One or more of: `"MM"`, `"tau"`, `"CM"`, and/or `"mtl"` (**see** `nlrob` **for details**).

Value

A named `list` containing the following elements:

1. **$ordt**: A `data.frame` containing the original input data including the temporal variable `$t`, the Solvent-Solute (S/M) ratio `$x`, the **actual** fractional yield (`$y`; g/g insoluble solid), as well as the **modeled** yield (`$mod_y`).
2. **$mdt**: A `data.frame` of modeled data including the model type (`$model`), the region of the extraction curve (`$period`), as well as S/M ratio `$x`, fractional yield `$y`, and time (`$t`) up to `maxq`.
3. **$mod_pars**: The model parameters including the constants C_1 and C_2 (`"c1"` and `"c2"`), the relative amount of expended solvent (kg/kg insoluble solid) at the end of the CER (q_m; `"qm"`), the grinding efficiency (r; `"r"`), the product of the solid phase mass transfer coefficient k_s and the specific area between intact and broken cells a_s ($k_s a_s$; `"ksas"`), and the initial fraction of solute in broken cells (G; `"G"`). See `show_pars` for detailed description of these parameters.
4. **$fit_pars**: A `character` vector of which parameters in `$mod_pars` were iteratively fit to the model.
5. **$resid**: A named `numeric` vector of various error values for the model, including Average Absolute Relative Deviation AARD (`"aard"`), round mean squared error RMSE (`"rmse"`), and the R2 value (`"r2"`).

References

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

Sovova, H. (2005), 'Mathematical model for supercritical fluid extraction of natural products and extraction curve evaluation', *The Journal of Supercritical Fluids* **33** (1), pp. 35-52, DOI: https://doi.org/10.1016/j.supflu.2004.03.005.

Sovova, H. (2017), 'Broken-and-intact cell model for supercritical fluid extraction: Its origin and limits', *The Journal of Supercritical Fluids* **129**, pp. 3-8, DOI: https://doi.org/10.1016/j.supflu.2017.02.014.

See Also

bicmod, show_pars, oec_bp, moderr

---

calcom *Calculate COM for an SFE or SWE extraction process*

Description

Calculates the Cost of Manufacturing (COM) and its accompanying components including Cost of Labour (COL), Cost of Raw Materials (CRM), Cost of Ulitities (CUT) and Fixed Costs (FCI). See **Details** for further information.

Usage

```
calcom(
  input,
  invars = c(names(input)[1], names(input)[2]),
  pltlab = invars[1],
  cosol_loss = 0.05,
  gen,
  crm,
  auxpr = NA,
  auxfr = NA,
  cut,
  col,
  fci,
  taxrate = NA,
  flowpar = "auto",
  draw = TRUE,
  mass_flow = FALSE,
  comode = "sfe",
  use_coefs = FALSE,
  export = "none"
)
```

Arguments

input	A named character vector of length 2, or a data.frame with 2 columns containing the temporal (**element 1**) and response (**element 2**) components of the extraction process. When a data.frame is provided, the input is treated like an extraction curve, and COM is calculated separately for each row.
invars	A character vector containing names of the 2 input variables (temporal and response). By default, the names provided in input are used.
pltlab	The plot title. By default, invars[1] is used.
cosol_loss	A **single** numeric value between 0.001 and 0.99 denoting the **fraction** of co-solvent lost during post-processing (e.g. evaporation). Defaults to 0.05.
gen, crm, cut, col, fci	**Named** numeric vectors of input parameters required for **general calculations** (gen), **Cost of Raw Materials** (crm), **Cost of Labour** (col), **Cost of Utilities**

(cut), and/or **Fixed Costs** (fci). Some parameters are mandatory, while others are optional. For an exhaustive list of available **input** and **output** parameters, use show_pars("com").

auxpr, auxfr — **Identically named** numeric vectors containing the **price per unit mass** of any auxiliary materials used in the process (e.g. for pre- or post-processing), and the **fraction of said material required relative to the mass of raw material** (>0). Both default to NA (no auxiliary materials used).

taxrate — The fraction of gross profit to be removed as tax to calculate **net profit**.

flowpar — The parameters of temperature and pressure at which flow rate is given. Used for conversion of volumetric flow to mass flow when !mass_flow. If set to "auto" (default), uses the temperature and pressure specified for the recovery line in crm[c("recp","rect")].

draw — A logical switch specifying whether to plot the results. Only works if **an extraction curve is provided as** input.

mass_flow — A logical specifying whether the ["flow"] component of gen is provided in **mass (g/min)** or **volumetric (mL/min)** units.

comode — Which extraction process to calculate COM for? One of: "sfe" (Supercritical Fluid Extraction) or "swe" (Subcritical Water Extraction).

use_coefs — A logical denoting whether to use pre-defined coefficients to calculate COM (TRUE) from CUT, CRM, FCI, and COL, or whether to sue a simple sum instead (FALSE, **default**).

export — An **existing** folder path into which the results are exported as a .CSV file and accompanying graphics (if any). Defaults to "none" (the results are not exported).

Details

The calculation of Cost of Manufacturing (COM) is based on the formula provided by Turton et al. (1998) when use_coefs is TRUE, and combined the Cost of Labour (COL), Cost of Utilities (CUT), Cost of Raw Materials (CRM), and Fixed Costs (FCI). In the current implementation, the Cost of Waste Management (CWT) normally included in this method has been omitted since neither supercritical CO2 and subcritical water extraction do not produce significant amounts of waste.

$$COM = 0.304 \times FCI + 2.73 \times COL + 1.23 \times (CRM + CUT)$$

Alternatively (and by default), COM is calculated as a simple sum of **monthly** CUT, CRM, COL, and FCI. The various COM constituents are calculated as below.

The monthly COL is calculated using a simplified formula incorporating the average monthly wage (W_{pay}), the number of daily work shifts (W_{sh}), and the number of personnel required (W_{pers}):

$$COL = W_{sh} \times W_{pers} \times W_{pay}$$

The monthly CRM encompasses a sum of various costs associated with raw materials, calculated in multiple steps. First, the number of **monthly** extraction cycles N_{ex} is calculated using W_{sh}, the number of work hours per shift (W_{hr}), extraction time (T_{ex}), batch exchange time (T_{aux}), and the number of work days per month (W_{days}).

$$N_{ex} = (W_{sh} \times W_{hr})/((T_{ex} + T_{aux})/60) \times W_{days}$$

Any volumetric flow rates are converted to mass flow using CO2 (for SFE) or water (for SWE) densities at process conditions specified in crm or flowpar. The equations incorporate fluid density ρ calculated by the Bender Equation (bendens) or the IAPWS R6-95 formulation (h2o_dens).

$$F_{mass} = \rho/1000 \times F_{vol}$$

The density of main and co-solvent mixtures is calculated using the Linear Blend Rule (see `etoh_dens`).

Next, the main solvent usage (SL_{main}, water for SWE or a supercritical fluid like CO2 for SFE) and co-solvent usage (SL_{aux}) are calculated via the equations below. **Currently only EtOH for SFE and CO2 for SWE are supported as co-solvents**. The units are either **L** or **kg** depending on whether `mass_flow` is `TRUE`. For CO2, the equation depends on whether a recovery tank pressure was specified - if it was, the system is assumed to be equipped with a CO2 recirculation/recovery system, incorporating the extractor volume V_{ex}. If not, the flow rate is simply converted to kg/min and multiplied by extraction time (the same equation is used for SWE). In equations below, density ρ is assumed to be in units of **g/L**.

$$SL = \rho/1000 \times V_{ex} \text{ for CO2 as a main (SFE) or a co-solvent (SWE) with a recovery system}$$

$$SL = F_{mass}/1000 \times T_{ex} \times 0.05 \text{ for EtOH co-solvent (SFE) assuming a 5\% loss per batch}$$

$$SL = \rho_{CO2}/1000 \times (F_{aux}/(F_{aux} + F_{main})) \times V_{ex} \text{ for CO2 as a co-solvent for SWE}$$

$$SL = F_{mass}/1000 \times T_{ex} \text{ for all solvents where a recovery system is not present}$$

Finally, the requirement (REQ, kg), N_{ex}, and the price PR of **each** material i (including solvents) are combined as follows and the results summed to obtain the monthly CRM:

$$CRM = \sum REQ_i \times N_{ex} \times PR_i$$

Similarly, all main power requirements ($PWRM_i$, in **kWh**) are multiplied by N_{ex}, T_{ex} (in **hr**), and cost per kWh (PR_{kWh}) to obtain main power requirement PW_{main}:

$$PW_{main} = PWRM_i \times N_{ex} \times T_{ex} \times PR_{kWh}$$

Power requirements for **drying**, **comminution**, and **evaporation** (termed PW_{aux} when summed) are calculated by multiplying the respective power requirement $PWRA_i$ with the monthly consumption of the associated material(s) REQ_i divided by the respective processing capacity of required equipment CAP_i:

$$PW_{aux} = \sum PWRA_i \times (REQ_i/CAP_i)$$

The main and auxiliary power usage is finally summed to obtain CUT:

$$CUT = PW_{main} + PW_{aux}$$

The FCI is either used as-provided in the input data or incorporates the **depreciation fraction** that is deducted yearly from the CAPEX.

Once COM is calculated, the Specific Cost SC (USD/kg) of extract production is calculated by dividing monthly COM by the monthly yield of extract Y_{month} (kg). Sales Volume SV (USD/month) is then obtained by multiplying Y_{month} by sales price SP_{kg} (USD/kg). The manufacturing cost ($SC \times Y_{month}$) is then subtracted from SV to obtain the Gross Profit (GPr, USD/month), which may then be corrected for taxes to get Net Profit (NPr).

$$SC = COM/Y_{month}$$

$$SV = Y_{month} \times SP_{kg}$$

$$GPr = SV - SC \times Y_{month}$$

$$NPr = GPr - taxes$$

Finally, the Profit Margin (MRG, %) and Payback Period (PBK, yr) are obtained as follows:

$$MRG = NPr/SV \times 100$$

$$PBK = CAPEX/(NPr \times 12)$$

Value

A named `list` containing the input parameters (`$input`), the calculated output values (`$output`) its simplified form (`$simple_output`), extra parameters including FCI and COL (`$extra`), and the function call (`$call`). Optionally, specific cost and payback time plots of the results are also included when `draw` is `TRUE` and an extraction curve is provided as `input` (`$plots`). Any plot points removed due to economic **non-**viability are also noted in element `$rm_state`.

References

Turton, R., Bailie, R.C., Whiting, W.B., Shaeiwitz, J.A. (1998), *Analysis, Synthesis and Design of Chemical Process, PTR*, Prentice Hall, Upper Saddle River, NJ, USA.

See Also

`show_pars`, `bendens`, `h2o_dens`, `com_export`

Examples

```
#Calculate COM for a single process
comin <- c(time = 180, yield = 7)

#Calculate COM throughout an extraction curve and plot the results
comin <- data.frame(time = c(5,9,11,15,19,21,23,25,30,40,60,80,100,120,140,160,180),
yield = c(0.80, 2.18, 2.79, 3.40, 3.86, 4.17, 4.47, 4.63, 4.93,
          5.24, 5.70, 6.00, 6.31, 6.46, 6.77, 6.92, 7.00))

comres <- calcom(input = comin,
gen = c(volex = 30, load = 9, pres = 300, temp = 40, flow = 3000,
        extime = 30, csol_flow = 300, dilfac = 2, pr_sale = 90),
crm = c(bh = 155, id = 15.5, pr_mat = 0.5, pr_msol = 0.8, pr_csol = 2,
recp = 60, rect = 10, sept = 55),
cut = c(pw_main = 25, pr_kwh = 0.13, pw_dry = 2, pw_com = 1, pw_evap = 13.3,
cap_dry = 4.16, cap_com = 20, cap_evap = 60),
col = c(oper = 1, whr = 8, shifts = 3, wage = 940, wdays = 24),
fci = c(capex = 150000, maint = 250, other = 1210),
auxpr = c(oil = 1.52),
auxfr = c(oil = 1),
pltlab = "Time (min)")
```

---

compare_gcm *Compare estimated parameters from various GCM methods*

Description

Compares the boiling points, critical parameters (temperature, pressure, volume), and/or Hansen Solubility Parameters (HSPs) estimated for a solute by various Group Contribution Methods (GCMs).

Usage

```
compare_gcm(solute)
```

Arguments

solute	A (optionally named) character vector of solute information as provided to mol_find.

Value

A data.frame containing the type of estimated parameter, the method used, whether overlap was allowed in GCM SMARTS substructures, and various estimated parameters including boiling points ("Tb", "Tb_corr"), critical temperature ("Tc"), pressure ("Pc"), and volume ("Vc"), as well as dispersion ("dD"), polarity ("dP", "dP_low"), and hydrogen bonding ("dHB", "dHB_low") components of HSPs.

See Also

mol_find, sfe_mod

Examples

```
#Limonene
mol <- c("CC1=CCC(CC1)C(=C)C", "5989-54-8", "Limonene")
res <- compare_gcm(mol)

#Linoleic acid
mol2 <- c("CCCCCC=CCC=CCCCCCCCC(=O)O", "60-33-3", "Linoleic Acid")
res <- compare_gcm(mol2)
```

---

com_export *Export COM calculation results as .CSV and graphics*

Description

Exports the results and (if any) plots generated by calcom.

Usage

```
com_export(
  comres,
  expath = getwd(),
  plotpars = "default",
  plot_format = "png",
  silent = FALSE
)
```

Arguments

comres	The output of function calcom.
expath	The **existing** export folder path.
plotpars	A **named** numeric vector of plotting parameters, including: width (["w"], in inches), height (["h"]), point size (["psize"]), and dots per inch or DPI (["dpi"]). Initially set to "default", i.e. c(w = 10, h = 12, psize = 12, dpi = 300).

plot_format	Format of plots to export. One of "png" (default) or "pdf".
silent	Should console output be silenced? Defaults to FALSE).

Value

A .CSV file and any accompanying graphics (where present in comres) are expoted to expath.

See Also

calcom

---

desir_export *Export desirability optimization results*

Description

Exports the results of desirability function analysis generated by doe_desir into a summary .CSV file.

Usage

```
desir_export(input, expath = getwd(), detailed = FALSE, silent = FALSE)
```

Arguments

input	The output of function doe_desir.
expath	The **existing** export folder path.
detailed	Should detailed results be exported (FALSE by default).
silent	Should updates about function progress be suppressed in the console? (FALSE by default).

Value

A .CSV file exported into expath containing desirability function analysis results.

See Also

doe_desir

desir_sumods *Summarize key metrics of several DOE models*

Description

Summarizes performance metrics of 2 or more models output from function doe_analyze. Part of the doe_desir workflow.

Usage

```
desir_sumods(mods, rm_eqs = FALSE)
```

Arguments

mods	A list of **2 or more** models output from function doe_analyze.
rm_eqs	Should model equations be removed from the final output? Defaults to FALSE (equations are kept).

Value

A data.frame with each row containing key information about a single model, including the $response name, the $model_type and $order, performance metrics including the R2 ($r2), adjusted R2 ($adj_r2), residual standard error ($stnd_error), the F statistic ($F), degrees of freedom ($DOF_1 and $DOF_2), and the Lack-of-Fit test p-value ($lof_pvalue). When rm_eqs is FALSE, model equations (model_equation) are also included.

See Also

doe_desir, doe_analyze

doeopt_export *Export DOE analysis results*

Description

Export the results and visualisations of doe_analyze workflow as .CSV and .PDF (or .PNG) files.

Usage

```
doeopt_export(
  input,
  expath = getwd(),
  plotpars = "default",
  plot_format = "png",
  silent = FALSE
)
```

Arguments

input	The output of function doe_analyze.
expath	The **existing** export folder path.
plotpars	A **named** numeric vector of plotting parameters, including: width (["w"], in inches), height (["h"]), point size (["psize"]), and dots per inch or DPI (["dpi"]). Initially set to "default", i.e. c(w = 10, h = 12, psize = 12, dpi = 300).
plot_format	Format of plots to export. One of "png" (default) or "pdf".
silent	Should console output be silenced? Defaults to FALSE).

Value

A .CSV file and any accompanying graphics are expoted to expath.

See Also

doe_analyze

---

doex *Example data with DoE results*

Description

A named nested list containing Design of Experiments (DOE) data from scientific literature to be used with **supercRit**. Each element is itself a list including the following elements:

1. $data: A data.frame containing the run order, coded and uncoded factor levels, and corresponding response values of the design.
2. $opt_pars: A data.frame of optimized parameters as stated in the source publication(s).
3. $design: A brief description of the DOE type and factors.
4. $desc: A brief description of the study response variable(s).
5. $citation: A reference to the source publication(s).

Currently, the following designs and results are included encompassing Supercritical Fluid Extraction (SFE) and steam distillation processes:

1. $ccd1: Central Composite Inscribed Design (CCID) for ergosterol SFE from mushrooms (Almeida et al., 2024).
2. $ccd2: Central Composite Design (CCD) for SFE of black pepper (Bagheri et al., 2014).
3. $ccd3: Central Composite Face-centered Design (CCFD) for SFE of spearming flavonoids (Bimakr et al., 2011).
4. $ccd4: CCD for SFE of lavender essential oil (Danh et al., 2012).
5. $ccd5: CCFD for steam distillation of *Myrtus communis* essential oil (Kaya et al., 2020).
6. $ccd6: CCID for elimination of TiN peeling during CVD deposition (Buckner et al., 1997).
7. $bbd1: Box-Behnken Design (BBD) for SFE of rapeseed oil (Cvjetko et al., 2012).
8. $bbd2: BBD for steam distillation of *Eucalyptus tereticornis* essential oil (Galadima et al., 2012).

9. `$bbd3`: BBD for SFE of lavender flowers (Jerkovic et al., 2017).
10. `$bbd4`: BBD for SFE of Dalmatian sage leaves (Jokic et al., 2018).
11. `$bbd5`: BBD for SFE of antioxidants from Dalmatian sage (Pavic et al., 2019).
12. `$bbd6`: BBD for SFE of wheat germ oil (Satyannarayana et al., 2018).
13. `$ffd1`: Full Factorial Design (FFD, 2^5) for determining the effect of machining factors on ceramic strength (NIST/SEMATECH, 2024).
14. `$frfd1`: Fractional Factorial Design (FrFD, 2^(5-1)) for determining significant factors affecting the distance (response) a ball is thrown by a makeshift catapult (NIST/SEMATECH, 2024).

Usage

```
doex
```

Format

An object of class `list` of length 14.

References

Almeida, C.F., Manrique, Y.A., Lopes, J.C.B., Martins, F.G., Dias, M.M. (2024), 'Recovery of ergosterol from Agaricus bisporus mushrooms via supercritical fluid extraction: A response surface methodology optimisation', *HELIYON* **10** (2), article e21943, DOI: https://doi.org/10.1016/j.heliyon.2023.e21943.

Bagheri, H., Manap, M.Y.B.A., Solati, Z. (2014), 'Response surface methodology applied to supercritical carbon dioxide extraction of *Piper nigrum* L. essential oil', *LWT - Food Science and Technology* **57**, pp. 149-155, DOI: http://dx.doi.org/10.1016/j.lwt.2014.01.015.

Bimakr, M., Rahman, R.A., Ganjloo, A., Taip, F.S., Salleh, L.M., Sarker, M.Z.I. (2012), 'Optimization of Supercritical Carbon Dioxide Extraction of Bioactive Flavonoid Compounds from Spearmint (*Mentha spicata* L.) Leaves by Using Response Surface Methodology', *Food Bioprocessing Technology* **5**, pp. 912-920, DOI: https://www.doi.org/10.1007/s11947-010-0504-4.

Buckner, J., Cammenga, D.J., Weber, A. (1997), 'Elimination of TiN Peeling During Exposure to CVD Tungsten Deposition Process Using Designed Experiments', In: *Statistical Case Studies for Industrial Process Development* (Czitrom, V., Spagon, P.D., 1997, eds.), pp. 199-211.

Cvjetko, M., Jokic, S., Lepojevic, Z., Vidovic, S., Maric, B., Redovnikovic, I.R. (2012), 'Optimization of the Supercritical CO2 Extraction of Oil from Rapeseed Using Response Surface Methodology', *Food Technology and Biotechnology* **50** (2), pp. 208-215.

Danh, L.T., Triet, N.D.A., Han, L.T.N., Zhao, J., Mammucari, R., Foster, N. (2012), 'Antioxidant activity, yield and chemical composition of lavender essential oil extracted by supercritical CO2', *Journal of Supercritical Fluids* **70**, pp. 27-34, DOI: https://dx.doi.org/10.1016/j.supflu.2012.06.008.

Galadima, M.S., Ahmed, A.S., Olawale, A.S., Bugaje, I.M. (2012), 'Optimization of Steam Distillation of Essential Oil of *Eucalyptus tereticornis* by Response Surface Methodology', *Nigerian Journal of Basic and Applied Science* **20** (4), pp. 368-372.

Jerkovic, I., Molnar, M., Vidovic, S., Vladic, J., Jokic, S. (2017), 'Supercritical CO2 Extraction of *Lavandula angustifolia* Mill. Flowers: Optimisation of Oxygenated Monoterpenes, Coumarin and Herniarin Content', *Phytochemical Analysis* **28**, pp. 558-566, DOI: https://doi.org/10.1002/pca.2705.

Jokic, S., Molnar, M., Jakovlevic, M., Aladic, K., Jerkovic, I. (2018), 'Optimization of supercritical CO2 extraction of *Salvia officinalis* L. leaves targeted on Oxygenated monoterpenes, alpha-humulene, viridiflorol and manool', *Journal of Supercritical Fluids* **133**, pp. 253-262, DOI: https://doi.org/10.1016/j.supflu.2017.10.022.

Kaya, D.A., Ghica, M.V., Danila, E., Ozturk, S., Turkmen, M., Kaya, M.G.A., Dinu-Pirvu, C.-E. (2020), 'Selection of Optimal Operating Conditions for Extraction of *Myrtus communis* L. Essential Oil by the Steam Distillation Method', *Molecules* **25**, article 2399, DOI: http://dx.doi.org/10.3390/molecules25102399.

NIST/SEMATECH e-Handbook of Statistical Methods, available at: http://www.itl.nist.gov/div898/handbook (accessed 01.06.2024).

Pavic, V., Jakovlevic, M., Molnar, M., Jokic, S. (2019), 'Extraction of Carnosic Acid and Carnosol from Sage (*Salvia officinalis* L.) Leaves by Supercritical Fluid Extraction and Their Antioxidant and Antibacterial Activity', *Plants* **8**, article 16, DOI: http://dx.doi.org/10.3390/plants8010016.

Satyannarayana, S., Anjaneyulu, B., Neeharika, T.S.V.R., Rani, K.N.P., Charkabarti, P.P. (2018), 'Process optimization for the supercritical carbon dioxide (SC-CO2) extraction of wheat germ oil with respect to yield, and phosphorous and tocol contents using a Box Behnken design', *Grasas y Aceites* **69** (3), article e259, DOI: https://doi.org/10.3989/gya.0102181.

`doe_analyze` *Build, prune, analyze, and summarize RSM models from DoE*

Description

A complete workflow for modeling and analysing DoE design responses that includes the following elements:

1. Building of the **initial** linear or quadratic (RSM) model.
2. Carry out Lack-of-Fit (LoF) testing.
3. Removing insignificant effects via Stepwise Regression and/or p-value cutoff.
4. Building the **final** simplified model.
5. Minimizing and/or maximizing the response using `optim` and Canonical Analysis.
6. Compiling summaries, descriptive statements, and plots for all of the above. For further information, see **Details** and **References**.

Usage

```
doe_analyze(
  doe,
  uc_facs = NA,
  cent_id = NA,
  resp_var,
  time_var,
  mod_order = 1.5,
  canon_thres = "auto",
  p_cutoff = 0.1,
  trim_method = "stepwise",
  which_facs = "coded",
  export = "none",
  asprat = "default",
  verbose = TRUE
)
```

Arguments

doe	A data.frame containing DoE factors (**coded and uncoded**), response (resp_var), and temporal variable (e.g. run order, time_var). **Coded factor names are assumed to be capital letters ordered alphabetically**.
uc_facs	A character vector of **uncoded** factor names, in order of appearance in doe. If not provided (NA, default), uncoded factors are not included in the analysis.
cent_id	A unique character identifier for center points to be highlighted in Exploratory Data Analysis (EDA) plots. Defaults to NA (i.e. no center points are highlighted).
resp_var, time_var	Single character strings specifying the response and run order (temporal) variables as they appear in doe.
mod_order	The order of model to build. One of: 1 (first-order, linear), 1.5 (first order with interaction terms), or 2 (quadratic polynomial, suitable for RSM).
canon_thres	A threshold specific to Canonical Analysis (CA). When set to "auto", defaults to 10% of the maximum eigen value.
p_cutoff	The p-value cutoff to use when pruning the initial model. Only used when trim_method includes "p_cutoff".
trim_method	A character string specifying which method(s) to use for removing **in**significant terms from the initial model. One of "stepwise" (Stepwise Regression, default), "p_cutoff" (p-value cutoff), "both", or "none".
which_facs	A character string specifying which factors to use for model building. One of "coded" (default) or "uncoded".
export	An **existing** export folder path where plots and results are exported as .PDF (or .PNG) and .CSV files via function doeopt_export.
asprat	Aspect ratio of generated plots. When set to "default", a ratio of 1 is applied.
verbose	Should detailed updates be provided during processing? Defaults to TRUE.

Details

The key goal of SFE experimental design is process optimization, commonly achieved via Response Surface Methodology (RSM) which calculates both the interaction and quadratic terms and approximates the shape of the local response surface (Sharif et al., 2014; Yolmeh et al., 2017). Because a natural product SFE rate almost always decreases along a single smooth curvature seen in a typical OEC, RSM is especially suitable as it predicts a local response surface by a single 2nd-order (i.e. quadratic) polynomial function, thus failing to describe piecewise smooth or jagged response trajectories in a single experiment. An example model containing linear, two-way, and quadratic interaction terms for 3 factors is shown below.

$$y = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \epsilon$$

At least a 3-level design is required to estimate quadratic interactions. This requirement is not met by 2-level designs with center points, which are positioned within the experimental space such that all quadratic effects are aliased (NIST/Sematech, 2013). Three-level FFDs are useful but are not rotatable and become prohibitive due to the large number of runs required for more than 3 factors. Instead, rotatable Box-Wilson Central Composite (CCD) or Box-Behnken (BBD) designs are used, where response is dependent only on the distance of factor levels from the center of the experimental space (and not on direction); thus, points at an identical distance from the center of the experimental space exhibit the same prediction error/variance (Sharif et al., 2014). Cubic terms are seldom included when interpreting RSM. An ANOVA table should then be examined for

goodness of fit (consistently high R2 and adjusted R2) and insignificant terms of low-magnitude exhibiting a high p value (e.g. >0.05). Probability plots can be used as a complementary technique. Insignificant effects may then be separated systematically – using stepwise regression and/or Pareto charts (Kiratu, 2015), for example – and the simplified ANOVA model re-examined iteratively. Once the simplest model that explains most of the response variability is attained, the residual distribution should be examined to validate the normality and uniformity of variance assumptions (Yolmeh et al., 2017). Transformation of response data (e.g. via a natural logarithm or a Box-Cox approach) is useful for mitigating violations of these assumptions. The residuals can also be checked for curvature via if a 2nd-order model was not built. Once the residual distribution is satisfactory, the optimal factor settings can be determined by plotting the model coefficients/terms against factor levels. In case of RSM designs, a contour plot(s) for each response is used to both optimize each response separately and/or locate a compromise between multiple responses. The current workflow incorporates linear and quadratic model building, lack-of-fit testing, systematic removal of insignificant effects, visualisation of important factors and optimization of factor values to maximize or minimize the response via Canonical Analysis (CA).

Value

A complex `list` including the following elements:

1. **$models**: The results of **initial and final** models of class `"lm"`.
2. **$results**: Summarized key results of **initial and final** models, including a `data.frame` of model coefficients, aliasing, standard errors, t- and p-values, significance level, ANOVA sum of squares and F-values (`$Model_Results`). Other elements include a `data.frame` of main, interaction, and/or quadratic effects and residuals for each observation (`$Model_Data`), the original input `data.frame` (`$Orig_Data`), a `list` of performance metrics such as R2, F-statistic, LoF test results, and Canonical Analysis (`$Model_Metrics`). Finally, a summary of key input data such as the number of observations and both **coded and uncoded** model equations are also included.
3. **$plots**: A list of `ggplot`-class plots describing the results and including Box, EDA, Pareto, Cook's Distance, and other plots.
4. **$statements**: A vector of descriptive statements about the results.
5. **$call**: The function call.

References

NIST/SEMATECH (2013), 'Engineering Statistics Handbook', available at: https://www.itl.nist.gov/div898/handbook/index.htm (accessed 29.09.2024).

Kiratu, J., Raynie, D.E. (2015), 'Aiding the Development of Extraction Procedures with Response Surface Methodology', *LCGC North America* **33** (7), pp. 104-111.

Sharif, K.M., Rahman, M.M., Azmir, J., Mohamed, A., Jahurul, M.H.A., Sahena, F., Zaidul, I.S.M. (2014), 'Experimental design of supercritical fluid extraction – A review', *Journal of Food Engineering* **124**, pp. 105-116, DOI: https://doi.org/10.1016/j.jfoodeng.2013.10.003.

Yolmeh, M., Jafari, S.M. (2017), 'Applications of Response Surface Methodology in the Food Industry Processes', *Food Bioprocess. Technol.* **10**, pp. 413-433, DOI: https://www.doi.org/10.1007/s11947-016-1855-2.

See Also

`doeopt_export`

Examples

```
#Maximizing extraction yield of spearmint (SFE)
doe_optres <- doe_analyze(doe = doex[["ccd3"]][["data"]],
uc_facs = c("P_bar", "T_degC", "EtOH_gmin"),
cent_id = NA,
resp_var = "ExtYield",
time_var = "Actual_Order",
mod_order = 2,
canon_thres = "auto",
p_cutoff = 0.10,
trim_method = "both",
which_facs = "coded",
export = "none",
verbose = TRUE)
```

---

doe_bbd *Generate a Box-Behnken experimental design*

Description

Generates a Box-Behnken experimental design for 3-4 factors and 3 factor levels.

Usage

```
doe_bbd(
  factors = 3,
  cpts = 0,
  fnames,
  flims,
  randomize = TRUE,
  export = "none"
)
```

Arguments

`factors`	The number of factors to use (between 3-4). The value must equal to the length of `fnames` and `flims`.
`cpts`	The number of **additional** center points to add to the design (0 by default). BBD designs always have 3 center points by default.
`fnames`	A `character` vector of factor names of length 3-4.
`flims`	A `list` of numeric vectors containing the lower and upper factor level limits **in the same order as given in** `fnames`.
`randomize`	A `logical` indicating whether or not the run order of the design is randomized (`TRUE` by default).
`export`	Either `"none"` (default) or a path to export the generated design .TAB file to.

Value

A `data.frame` containing the generated design.

See Also

doe_export, doe_frfd, doe_ffd, doe_ccd, doe_tm

Examples

```
#Box-Behnken Design for 3 factors with 2 additional center points
doe_res <- doe_bbd(factors = 3,
                   cpts = 2,
                   fnames = c("Pressure", "Temperature", "Flow"),
                   flims = list(c(100, 300), c(35, 65), c(2, 4)))
```

---

doe_ccd *Generate a Central Composite experimental design*

Description

Generates either a Central Composite Circumscribed (CCC) or Central Composite Face-Centered (CCF) experimental design. Currently available for 2-4 factors.

Usage

```
doe_ccd(
  design = "CCC",
  levels = 2,
  factors = 3,
  cpts = 0,
  fnames,
  flims,
  randomize = TRUE,
  export = "none"
)
```

Arguments

design	Type of design to generate. One of "CCC" (default) or "CCF".
levels	The number of factor levels to use (5 and 3 when design is "CCC" and "CCF", respectively).
factors	The number of factors to use (between 2-4). The value must equal to the length of fnames and flims.
cpts	The number of center points to add to the design (0 by default).
fnames	A character vector of factor names of length 2-4.
flims	A list of numeric vectors containing the lower and upper factor level limits **in the same order as given in** fnames.
randomize	A logical indicating whether or not the run order of the design is randomized (TRUE by default).
export	Either "none" (default) or a path to export the generated design .TAB file to.

Details

CCC' and 'CCF' designs require 5 and 3 factor levels, respectively

Value

A `data.frame` containing the generated design.

See Also

doe_export, doe_frfd, doe_ffd, doe_bbd, doe_tm

Examples

```
#Central Composite Circumscribed for 4 factors (standard CCD)
doe_res <- doe_ccd(design = "CCC",
levels = 5,
factors = 3,
cpts = 6,
fnames = c("Pressure", "Temperature", "Co-Solvent"),
flims = list("hard"=c(100, 345), "hard"=c(35, 75), c(2, 8)))
```

---

doe_desir *Combine models with different responses into a Desirability Function*

Description

Calculates individual and overall desirability function from two or more models (linear or quadratic) output from the doe_analyze function. See **Details** and **References** for further information.

Usage

```
doe_desir(
  mods,
  dsrng,
  frng = "default",
  obj = rep("max", length(mods)),
  dtype = "coded",
  wts = lapply(obj, function(x) if (x == "trg") rep(1, 2) else 1),
  spts = c(100, 10),
  modbase = "final",
  optmet = "nlopt",
  kmed = NA,
  export = "none",
  silent = FALSE
)
```

Arguments

mods	A list of DoE model results output from function doe_analyze.
dsrng	A named list of numeric vectors of length 2 or 3 specifying lower and upper response limits as well as (**optionally**) the target for each model in mods. The target can **only** be specified if the corresponding element of obj is set to "trg". The list elements must have names identical to those of responses.
frng	An optional **named** list of numeric vectors of length 2 containing lower and upper limits of all factors. Names must correspond to those of **coded** factors (i.e. capital letters). If set to the "default" value, uses the full range of each factor.
obj	Either a single string or character vector denoting the objective of desirability function. Each element must be one of: "min" (response minimization), "max" (maximization), or "trg" (a specific response target). When a string is given, the value applies to all models in mods.
dtype	A character string specifying whether "coded" (default) or "uncoded" factor values should be used for processing. **Note** that ranges in frng must be provided based on this option.
wts	A list of desirability weights between **0.1 and 10** where each element corresponds to a model in mods and its associated obj and must either be a single numeric value or a vector of length 2 when the corresponding obj is set to "trg" (in such cases, the **first and second** elements are weights used for values **lower and higher** than the target, respectively).
spts	A numeric vector of length 2 providing the number of desirability optimization starting points to be randomly generated (**element 1**), and those to be retrieved from the original model data.frame (**element 2**). Defaults to c(100,10).
modbase	A character string specifying which model from each element of mods should be used for processing. One of "initial" or "final" (default).
optmet	optimization method. One of "optim" (uses the eponymous optim function) or "nlopt" (uses the **NLopt** library for non-linear optimization).
kmed	The **optional** number of clusters to use for Partitioning Around Medoids (PAM) clustering. May be set to a specific integer, "auto" for automatic choice of appropriate cluster number, or NA (**default**, clustering **not run**).
export	An **existing** folder path where a .CSV of the results should be exported.
silent	A logical specifying whether useful updates in the console during processing should be suppressed. Defaults to FALSE.

Details

This function retrieves two or more responses from various DoE-derived linear and/or quadratic models analyzed by doe_analyze and transforms each response into a desirability function using the approach of Derringer and Suich (1980) for simultaneous optimization of several responses. The overall desirability is then calculated and optimized by non-linear techniques including optim and the **NLopt** library.

The process allows for calculation of one of three types of desirability functions including:

1. **Larger-The-Best (LTB)**: A maximization function.
2. **Smaller-The-Best (STB)**: A minimization function.
3. **Nominal-The-Best (NTB)**: A function targeting a specific optimal response value.

The response $\hat{Y}_i$ as well as its minimum (Y_{i*}) and maximum (Y_i^*) acceptable values are incorporated into the calculation of a desirability d_i where $0 \leq d_i \leq 1$. A weight $0.1 \leq r1 \leq 10$ is also added to modify the relative influence of the response value on desirability. When $r < 1$, the desirability increases at a lower rate above Y_{i*}, decreasing the relative importance of the response. The reverse is true when $r > 1$, rapidly increasing desirability above Y_{i*} and increasing importance of the response. The overall desirability equations for **one-sided transformations** is:

$$d_i = 0 \text{ if } \hat{Y}_i \leq Y_{i*}$$

$$d_i = [(\hat{Y}_i - Y_{i*})/(Y_i^* - Y_{i*})]^{r1} \text{ if } Y_{i*} < \hat{Y}_i < Y_i^*$$

$$d_i = 1 \text{ if } \hat{Y}_i \geq Y_i^*$$

Upper, lower, or both limits can be set for each response.

For two-sided transformations required by the **NLB** approach, the value $\hat{Y}_i$ has both a **minimum and maximum** constraint and the value chosen for c_i is based on maximum desirability provided within the constraints of Y_{i*} and Y_i^*. The relative importance of each side of this target value is adjusted by factors $r1$ and $r2$. The associated equations are as follows:

$$d_i = ((\hat{Y}_i - Y_{i*})/(c_i - Y_{i*}))^{r1} \text{ if } Y_{i*} \leq \hat{Y}_i \leq c_i$$

$$d_i = ((\hat{Y}_i - Y_i^*)/(c_i - Y_i^*))^{r2} \text{ if } c_i < \hat{Y}_i \leq Y_i^*$$

$$d_i = 0 \text{ if } \hat{Y}_i < Y_{i*} \text{ or } \hat{Y}_i > Y_i^*$$

The **overall** desirability OD may then be calculated by taking the **geometric mean** of individual response desirability values:

$$OD = (d_1 \times d_2 \times ... \times d_k)^{1/k}$$

Both individual and overall desirability may be optimized using **non-linear techniques**.

Value

A named `list` containing the following elements:

1. **$factor_lims**: A `data.frame` summarizing the names, objectives, and lower/upper limits of all factors from `frng`.
2. **$response_lims**: A `data.frame` with names, desirability function objectives, lower/upper limits, and weights of all responses from `dsrng`.
3. **$mod_sums**: A `data.frame` summarizing key performance metrics from models.
4. **$orig_data**: The original input `data.frame` used to build the models, including coded and uncoded factor levels, responses, as well as corresponding individual and overall desirability values.
5. **$output_data**: The results of local desirability optimization from all starting points specified in `spts`, including individual and overall desirabilities.
6. **$unique_solutions**: A `data.frame` of statistically unique solutions derived from `$output_data`.
7. **$call**: The function call.

References

Cardoso, R.P., da Motta Reis, J.S., Silva, D.E.W., de Barros, J.G.M., Sampaio, N.A.S. (2023), 'How to perform a simultaneous optimization with several response variables', *Management and Administrative Professional Review* **14** (1), pp. 564-578, DOI: http://dx.doi.org/10.7769/gesec.v14i1.1536.

Cojocaru, C., Khayet, M., Zakrzewska-Trznadel, G., Jaworska, A. (2009), 'Modeling and multi-response optimization of pervaporation of organic aqueous solutions using desirability function approach', *Journal of Hazardous Materials* **167**, pp. 52-63, DOI: http://dx.doi.org/10.1016/j.jhazmat.2008.12.078.

Derringer, G., Suich, R. (1980), 'Simultaneous Optimization of Several Response Variables', *Journal of Quality Technology* **12** (4), pp. 214-219, DOI: https://doi.org/10.1080/00224065.1980.11980968.

See Also

doe_analyze, get_meq

Examples

```
#Calculate overall desirability among 3 responses
doe_lst1 <- load_internal("doe_lst1")

desires <- doe_desir(mods = doe_lst1,
                     dsrng = list(CarnosicAcid_mgg = c(0,150),
                     Carnosol_mgg = c(0,65), ExtYield = c(1,7)),
                     frng = list(B = c(40,60), A = c(10,30), C = c(1,3)),
                     obj = c("max", "max", "max"),
                     dtype = "uncoded",
                     wts = rep(1,3),
                     spts = c(100,10),
                     modbase = "final",
                     optmet = "nlopt",
                     kmed = "auto",
                     export = "none",
                     silent = FALSE)
```

---

doe_export *Export generated experimental design*

Description

Part of the **experimental design generation** workflow.

Usage

```
doe_export(input, export_name = NA, expath = getwd(), silent = FALSE)
```

Arguments

input	Output from one of the experimental design generator functions: doe_ffd, doe_frfd, doe_ccd, doe_bbd, or doe_tm.
export_name	An **optional** title to paste as the first line of the exported .TAB file.
expath	The directory path into which to export the output .TAB file.
silent	A logical specifying whether console output should be suppressed (FALSE by default).

Value

A .TAB file containing the experimental design data.frame is exported to the path specified in expath.

See Also

doe_frfd, doe_ffd, doe_ccd, doe_bbd, doe_tm

---

doe_ffd *Generate a Full Factorial experimental design*

Description

Generates a Full Factorial design with 2-3 levels and 2-5 factors. Three-level designs include 3 center points by default.

Usage

```
doe_ffd(
  levels = 2,
  factors = 3,
  cpts = 0,
  fnames,
  flims,
  randomize = TRUE,
  export = "none"
)
```

Arguments

levels	The number of factor levels to use (between 2-3).
factors	The number of factors to use (between 2-5). The value must equal to the length of fnames and flims.
cpts	The number of **additional** (to the default 3) center points to add to the design (0 by default).
fnames	A character vector of factor names of length 2-5.
flims	A list of numeric vectors containing the lower and upper factor level limits **in the same order as given in** fnames.
randomize	A logical indicating whether or not the run order of the design is randomized (TRUE by default).
export	Either "none" (default) or a path to export the generated design .TAB file to.

Value

A data.frame containing the generated design.

See Also

doe_export, doe_frfd, doe_ccd, doe_bbd, doe_tm

Examples

```
#Generate experimental design with 3 levels, 3 factors, and 3 additional center points (6 in total)
doe_res <- doe_ffd(levels = 3,
                   factors = 3,
                   cpts = 3,
                   fnames = c("Pressure", "Temperature", "Flow"),
                   flims = list(c(100, 300), c(35, 65), c(2, 4)))
```

---

doe_frfd *Generate a Fractional Factorial experimental design*

Description

Generates a 2-level Fractional Factorial experimental design.

Usage

```
doe_frfd(
  factors = 3,
  p = 1,
  cpts = 0,
  aliasing = "default",
  fnames,
  flims,
  randomize = TRUE,
  export = "none"
)
```

Arguments

factors	The number of factors to use (between 2-5). The value must equal to the length of fnames and flims.
p	A numeric value describing the fraction of the Full Factorial design used. Must be 1 when factors are between 2-4, and 1-2 for 5 factors.
cpts	The number of center points to add to the design (0 by default).
aliasing	Either "default" or a character vector containing the aliasing patterns to use.
fnames	A character vector of factor names of length 2-4.
flims	A list of numeric vectors containing the lower and upper factor level limits **in the same order as given in** fnames.
randomize	A logical indicating whether or not the run order of the design is randomized (TRUE by default).
export	Either "none" (default) or a path to export the generated design .TAB file to.

Value

A data.frame containing the generated design.

See Also

doe_export, doe_ffd, doe_ccd, doe_bbd, doe_tm

Examples

```
#Fractional Factorial (var. 3, 2^5-2)
doe_res <- doe_frfd(factors = 5,
                    p = 2,
                    cpts = 0,
                    aliasing = "default",
                    fnames = c("Pressure", "Temperature", "Flow", "Time", "PartSize"),
                  flims = list(c(100, 300), c(35, 65), c(2, 4), c(90, 240), c(0.2, 1.0)))
```

---

doe_optim *Optimize the model for either minimum or maximum response*

Description

Uses optim to both minimize and maximize the response variable in a linear/quadratic model.

Usage

```
doe_optim(model, mod_df, resp_var, effs, c_facs, uc_facs = NA)
```

Arguments

model	A linear or quadratic model of class lm.
mod_df	A data.frame of model input data containing **both** c_facs and uc_facs.
resp_var	A string specifying the response variable name.
effs	A data.frame of model effects, e.g. output as part of doe_prep.
c_facs, uc_facs	Character vectors of coded and uncoded factor names included in model (and mod_df), respectively.

Value

The model summary object of class "summary.lm", including a $tradopt list element

See Also

doe_analyze

doe_prep *Prepare DoE data for modeling and analysis*

Description

An auto-process function that performs necessary formatting and categorizing of an input DoE `data.frame`. Part of the `doe_analyze` workflow.

Usage

```
doe_prep(
  doe,
  time_var,
  resp_var,
  uc_facs = NA,
  which_facs = "coded",
  mod_order = 2
)
```

Arguments

`doe`	A `data.frame` containing all coded factors (named with **capital letters**), their uncoded equivalents with names as specified in `uc_facs`, the response variable as specified in `resp_var`, and the run order as provided in `time_var`.
`time_var, resp_var`	Character values specifying the column names in `doe` corresponding to the run order and response variables, respectively.
`uc_facs`	A character vector of column names in `doe` specifying **uncoded** factors.
`which_facs`	String specifying which type of factors to process in preparation for modeling. One of `"coded"` (default) or `"uncoded"`.
`mod_order`	The `numeric` model order, either one of 1, 1.5, or 2.

Value

A named `list` containing the original input `doe` (`$orig_df`), processed input ready for modeling (`$minput_df`), a sub-list of coded and uncoded factor names (`$all_facs`), as well as a `data.frame` of all main, interaction, and/or quadratic effects as appropriate for the set `mod_order` (`$all_effs`).

See Also

`doe_analyze`, `twoway`

doe_tm *Generate a Taguchi experimental design*

Description

Generates a Taguchi Method experimental design with 2-4 factor levels and 3-5 factors.

Usage

```
doe_tm(
  levels = 2,
  factors = 3,
  fnames,
  flims,
  randomize = TRUE,
  export = "none"
)
```

Arguments

levels	The number of factor levels to use (between 2-4).
factors	The number of factors to use (between 3-5). The value must equal to the length of fnames and flims.
fnames	A character vector of factor names of length 3-5.
flims	A list of numeric vectors containing the lower and upper factor level limits **in the same order as given in** fnames.
randomize	A logical indicating whether or not the run order of the design is randomized (TRUE by default).
export	Either "none" (default) or a path to export the generated design .TAB file to.

Value

A data.frame containing the generated design.

See Also

doe_export, doe_frfd, doe_ffd, doe_ccd, doe_bbd

Examples

```
#Taguchi Design for 3 levels and 3 factors
doe_res <- doe_tm(levels = 3,
                  factors =3,
                  fnames = c("Pressure", "Temperature", "Flow"),
                  flims = list(c(100, 300), c(35, 65), c(2, 4)))
```

`est_gcm` *Estimate various parameters via GCMs*

Description

Part of the `sfe_mod` workflow. Estimates boiling point, critical parameters (temperature, volume, pressure), and/or HSPs via Group Contribution Methods (GCM). See **Details** and **References**.

Usage

```
est_gcm(solute, tb, crit, hsp, hlight = TRUE, overlap = TRUE, silent = FALSE)
```

Arguments

`solute`	Input solute data as provided for `mol_find`.
`tb`, `crit`, `hsp`	The chosen GCM(s) for parameter estimation (see **Details**). Any of: 1. **For boiling point** (`tb`): `"JR"` (Joback-Reid), `"JR_corr"` (corrected Joback-Reid), `"SB"` (Stein-Brown), `"SB_corr"` (corrected Stein-Brown), or `"NL04"` (Nannoolal 2004). 2. **For critical parameters** (`crit`): `"JR"` (Joback-Reid), `"NL07"`, or `"NL07_robust"` (Nannoolal 2007). 3. **For HSP** (`hsp`): `"SP08"`, `"SP08_first"`, `"SP12"`, `"SP12_first"` (all variations of methods developed by Stefanis & Panayiotou in 2008-2012).
`hlight`	A `logical` indicating whether substructures are highlighted (see `plot_gcm`).
`overlap`	A `logical` indicating whether overlapping sub-structures should be allowed (`TRUE` by default). This argument is experimental (see `sub_smarts`).
`silent`	A `logical`. When `FALSE` (default), additional information is printed in the console. See `sfe_mod` for more details on the available methods.

Details

Group Contribution Methods (GCMs) predict thermodynamic and other properties from molecular structure using known properties of functional groups. In addition to first-order groups (i.e. primary groups), second-order groups (a.k.a. super-groups) may also be used and often encompass several first-order groups. Some newer methods utilize group interactions to further enhance predictions. Herein, several group contribution methods are utilized for prediction of boiling point, critical parameters (temperature, pressure, and volume), and Hansen Solubility Parameters. The approach of Joback & Reid (1987) remains the most popular method to estimate both boiling point T_b (K), critical temperature T_c (K), critical pressure P_c (bar), and critical molar volume V_c (ml/mole) using the equations below, where Σ is the sum-product of group contributions and the number of their occurrences in a given molecule:

$$T_b = 198.2 + \Sigma$$

$$T_c = T_b \times (0.584 + 0.965 \times \Sigma - \Sigma^2)^{-1}$$

$$P_c = (0.113 + 0.0032 n_A - \Sigma)^{-2}$$

$$V_c = 17.5 + \Sigma$$

The standard Joback-Reid approach tends to increasingly over-predict boiling points above 500 K. For this reason, Stein and Brown (1994) introduced a correction either side of 700 K by fitting a polynomial to a larger set of experimental data and minimizing the residuals:

$$T_b\ (corr.) = T_b - 94.84 + 0.5577 \times T_b - 0.0007705 \times T_b^2,\ \text{where}\ T_b \leq 700\ \text{K}$$

$$T_b\ (corr.) = T_b + 282.7 - 0.5209 \times T_b,\ \text{where}\ T_b > 700\ \text{K}$$

Both the Joback-Reid and Stein-Brown methods only utilize first-order groups, while second-order groups and possible group interactions are not considered. The newer approach by Nannoolal et al. (2004, 2007) incorporates both of these aspects. Same as above, Σ is the sum-product of group contributions, which **include first-order and second-order** groups as well as group interaction contributions. The `"NL07_robust"` variant of this method removes some group **interaction** contributions that were based on questionable or incomplete data according the authors. The letter n denotes the number of atoms in the molecule ***sans*-hydrogen**, while M is the molar mass (g/mol).

$$T_b = \Sigma/(n^{0.6583} + 1.6583) + 84.3395$$

$$T_c = T_b \times [0.6990 + 1/(0.9889 + \Sigma^{0.8607})]$$

$$P_c\ \text{(bar)} = M^{-0.14041}/(0.00939 + \Sigma)^2 \times 0.01$$

$$V_c\ \text{(mL/mol)} = \Sigma/n^{-0.2266} + 86.1539$$

Finally, in addition to boiling point and critical parameter estimation, HSP GCMs are also included as developed by Stefanis & Panayiotou (2008, 2012). These methods incorporate both first-order and (**optionally**) second-order groups to estimate the dispersion (δ_d), polarity (δ_p), and hydrogen bonding (δ_{HB}) HSPs. The earlier version of the model developed in 2008 uses these equations:

$$\delta_d\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 17.3231)\ MPa^{(1/2)}$$

$$\delta_p\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 7.3548)\ MPa^{(1/2)},\ \text{where}\ \delta_p > 3$$

$$\delta_p\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 2.7467)\ MPa^{(1/2)},\ \text{where}\ \delta_p < 3$$

$$\delta_{HB}\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 7.9793)\ MPa^{(1/2)},\ \text{where}\ \delta_{HB} > 3$$

$$\delta_{HB}\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 1.3720)\ MPa^{(1/2)},\ \text{where}\ \delta_{HB} < 3$$

An update provided in 2012 updates the equations to the following:

$$\delta_d\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 959.11)^{0.4126}\ MPa^{(1/2)}$$

$$\delta_p\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 7.6134)\ MPa^{(1/2)},\ \text{where}\ \delta_p > 3$$

$$\delta_p\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 2.6560)\ MPa^{(1/2)},\ \text{where}\ \delta_p < 3$$

$$\delta_{HB}\ (MPa^{0.5}) = (\sum_i N_i C_i + W \sum_j M_j D_j + 7.7003)\ MPa^{(1/2)},\ \text{where}\ \delta_{HB} > 3$$

Value

A `list` of estimated property/parameter(s) as named `numeric` vectors:

`solute_data` Solute data as output from function `mol_find`.

`pares` A `list` containing the boiling point and critical parameters (element `$Critical`), HSPs (element `$HSP`), GCMs used for their estimation (element `$Methods`), as well as a `list` of `data.frame` objects containing group contributions for each calculated value (element `$Contribs`).

`visres` Raster visualisations of the solute molecule with GCM-identified substructures highlighted (as output from function `plot_gcm`).

References

Joback, K.G., Reid, R.C. (1987), 'Estimation of Pure Components Properties from Group-contributions', *Chemical Engineering Communications* **57** (1-6), pp. 233-243, DOI: https://www.doi.org/10.1080/00986448708960487.

Nannoolal, Y., Rarey, J., Ramjugernath, D., Cordes, W. (2004), 'Estimation of pure component properties: Part 1. Estimation of the normal boiling point of non-electrolyte organic compounds via group contributions and group interactions', *Fluid Phase Equilibria* **226**, pp. 45-63, DOI: https://doi.org/10.1016/j.fluid.2004.09.001.

Nannoolal, Y., Rarey, J., Ramjugernath, D. (2007), 'Estimation of pure component properties: Part 2. Estimation of critical property data by group contribution', *Fluid Phase Equilibria* **252** (1), pp. 1-27, DOI: https://doi.org/10.1016/j.fluid.2006.11.014.

Stefanis, E., Panayiotou, C. (2008), 'Prediction of Hansen Solubility Parameters with a New Group-Contribution Method', *International Journal of Thermophysics* **29** (2), pp. 568-585, DOI: https://www.doi.org/10.1007/s10765-008-0415-z.

Stefanis, E., Panayiotou, C. (2012), 'A new expanded solubility parameter approach', *International Journal of Pharmaceutics* **426** (1), pp. 29-43, DOI: https://doi.org/10.1016/j.ijpharm.2012.01.001.

Stein, S.E., Brown, R.L. (1994), 'Estimation of normal boiling points from group contributions', *Journal of Chemical Information and Computer Sciences* **34** (3), DOI: https://www.doi.org/10.1021/ci00019a016.

See Also

`mol_find`, `sfe_mod`, `plot_gcm`, `sub_smarts`

Examples

```
#Define solute data
mol <- c("CC1=C(C(CCC1)(C)C)C=CC(=CC=CC(=CC=CC=C(C)C=CC=C(C)C=CC2=C(CCCC2(C)C)C)C)C",
"7235-40-7", "Beta-carotene")

#Estimate parameters by GCMs
estres <- est_gcm(mol, "JR", "JR", "SP08")
```

`etoh_dens` *Calculate density of ethanol and its mixtures with CO2*

Description

Calculates the density of pure ethanol at specific temperatures. The density of CO2-ethanol mixtures may also be calculated via the Linear Blend Rule, but density of CO2 must be provided (for example, from the `bendens` function). Part of the `bicmod` workflow.

Usage

```
etoh_dens(temp, co2_frac = 0, co2_rho = 0)
```

Arguments

`temp`	Temperature for which to calculate density (in degrees Celsius).
`co2_frac`	The volume fraction of CO2 (from 0 to 0.99).
`co2_rho`	Density of CO2 (**in g/L**).

Details

In this implementation ethanol is considered an incompressible fluid and the following equation (Poling et al., 2008) is used to calculate density ρ in g/L:

$$\rho = C_1/C_2^{1+(1-T/C_3)^{C_4}} \times RMM_{EtOH}$$

In this equation, C_1 to C_4 are constants: $C_1 = 1.6288$, $C_2 = 0.27469$, $C_3 = 514$, $C_4 = 0.23178$, T is temperature in Kelvin, and RMM_{EtOH} is the molar mass of ethanol (46.068 g/mol). The same expression is listed and described in **Eq. 3.14** (Rizza, 2014).

Density adjustment for CO2-Ethanol mixtures is carried out using the Linear Blend Rule. For example, when 6% ethanol is used as co-solvent (see also **Eq. 3.13** in Rizza, 2014):

$1/\rho_f = 0.94/\rho_{CO2} + 0.06/\rho_{EtOH}$

Value

A named `numeric` vector containing densities of pure ethanol (`["etoh"]`) and its mixture with CO2 (`["co2_etoh"]`).

References

Poling, B.E., Thomson, G.H., Friend, D.G., Rowley, R.L., Wilding, W.V. (2008), 'Physical and Chemical Data', In: Perry, R.H., Green, D.W. (eds.), *Perry's Chemical Engineers Handbook*, McGraw-Hill, chapter 2.

Rizza, C.S. (2014), *Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae*, MSc thesis, Universita Degli Studi Di Padova, Dipartimento Di Ingegneria Industriale.

See Also

`bicmod`

Examples

```
etoh_dens(55)
etoh_dens(55, 0.90, 880)
```

`flattenlist` *Flatten nested lists*

Description

Flattens nested lists into a one-level list.

Usage

```
flattenlist(x)
```

Arguments

`x`	A nested `list` object.

Value

A one-level list created from the input list `x`.

`get_meq` *Retrieve linear or quadratic model equation*

Description

Retrieves the coded or uncoded model equation from the output of function `doe_analyze`. Part of the `doe_desir` workflow.

Usage

```
get_meq(mod, mtype = "final", eqtype = "coded")
```

Arguments

`mod`	The output of function `doe_analyze`.
`mtype`	Which model to retrieve the equation from? One of `"initial"` or `"final"` (default).
`eqtype`	Which type of equation to retrieve from the model of type `mtype`? One of `"coded"` (default) or `"raw"`.

Value

A named string containing the **raw** or **coded** equation for `mod`.

See Also

doe_desir, desir_prep, doe_analyze

Examples

```
doe_lst1 <- load_internal("doe_lst1")
get_meq(doe_lst1[[1]])
```

---

h2o_dens *Calculate water density in the subcritical region*

Description

Calculates the density of pure water in the subcritical region (<374 degC, <221 bar) via the International Association for the Properties of Water and Steam formulation IAPWS-95 (Wagner & Pruß, 2002).

Usage

```
h2o_dens(temp = 25, pres = 1)
```

Arguments

temp	Water temperature in degrees Celsius. Defaults to 25.
pres	Water pressure in bar. Defaults to 1.

Value

The water density (in g/L) as a single numeric value.

References

Wagner, W., Pruß, A. (2002), 'The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use', *J. Phys. Chem. Ref. Data* **31**, pp. 387-535, DOI: https://doi.org/10.1063/1.1461829.

Examples

```
h2o_dens(200, 40)
```

hspex *Example data for estimation of solubility and critical parameters*

Description

A data frame with 33 rows and 7 columns containing example data about chemically diverse compounds to be used for Hansen solubility parameter (HSP), boiling point, and critical parameter estimation (pressure, temperature, volume). Data includes the following columns:

Name Name of the compound.

CAS The CAS number.

MVol The molar volume of the compound under standard temperature and pressure.

D The deltaD (dispersion, van der Waals forces) Hansen solubility parameter.

P The deltaP (dipole moment) Hansen solubility parameter.

HB The deltaH (hydrogen bonding) Hansen solubility parameter.

SMILES The **canonical** SMILES notation.

Usage

```
hspex
```

Format

An object of class `data.frame` with 33 rows and 7 columns.

References

Hansen Solubility Parameters, available at: https://www.hansen-solubility.com (accessed 02.06.2024).

National Library of Medicine, available at: https://pubchem.ncbi.nlm.nih.gov (accessed 02.06.2024).

hsp_export *Export results of HSP-based miscibility enhancement function*

Description

Exports the output of function `sfe_mod` as a .CSV file along any additional data of class `"ggplot"`.

Usage

```
hsp_export(
  input,
  addres = NA,
  plotpars = "default",
  plot_format = "png",
  expath = getwd(),
  silent = FALSE
)
```

Arguments

input	The output of function sfe_mod.
addres	Additional output of class "ggplot". Defaults to NA.
plotpars	Either "default" or a **named** vector containing one or more of the following parameters: "w" (width, defaults to 10 inches), "h" (height, defaults to 12 inches), "psize" (point size, default to 12), "res" (raster resolution, defaults to 800 px, only applies when plot_format is "png"), "dpi" (vector resolution, defaults to 300 dpi, only applies when plot_format is "pdf").
plot_format	The plot format. One of: "png" (default) or "pdf".
expath	The export directory as a character value. The path must exist on the system.
silent	A logical. When FALSE (default), additional information is printed in the console.

Value

A folder with a timestamp is created in the output directory expath, into which the output **.CSV** file is exported. Any plots are exported into an eponymous sub-folder.

See Also

sfe_mod

---

hsp_optim *Optimize HSP-based solubility enhancement for SFE*

Description

Part of the sfe_mod workflow. Assesses requirement and optimizes co-solvent choice for Supercritical CO2 Extraction (SFE) from Hansen Solubility Parameters (HSPs) and critical temperature of any solute.

Usage

```
hsp_optim(
  cp,
  hsp_vals,
  modif,
  pres = seq_last(80, 300, 20),
  temps = seq_last(32, 65, 3),
  vfrac = 0.1,
  silent = FALSE
)
```

Arguments

cp	The numeric critical temperature of the solute (**in Kelvin**).
hsp_vals	A named numeric vector of length 3-5 containing HSPs for the solute. Names must be: "dD", "dP", "dHB", optionally also including "dP_low" and/or "dHB_low".

modif	A character vector of one or more modifiers (i.e. co-solvents) to evaluate. For a list of modifiers, see sfe_mod.
pres, temps	A numeric vector of one or more pressure (75-700 bar) and temperature (31-70) values at which to evaluate co-solvents.
vfrac	The volume fraction of co-solvent(s) to use (between 0-1). Defaults to 0.1 (10% co-solvent).
silent	A logical indicating whether to show user updates when executing the function (FALSE by default).

Value

A list containing the following elements:

Modifiers A character vector of modifier names used for the evaluation.

Volume_Fraction The numeric volume fraction (between 0 and 1) used for the evaluation.

SoluteHSP_vs_Temp A list of 3 data.frame objects showing the variation of dD, dP, and dHB HSP parameters with temperature (as set by the temps argument).

SolventBlend_HSPs A **nested** list of data.frame objects showing the variation of HSP parameters with pressure and temperature **for each chosen modifier**.

Ra A list of data.frame objects showing the calculated HSP distances (R_a) for selected pressure and temperature values (pres and temps).

Miscib_Enhancement A list of data.frame objects showing the **percentage Miscibility Enhancement** for pure CO2 and CO2-modifier mixtures.

Best_Modifier A data.frame showing the modifier providing the highest Miscibility Enhancement for the solute at every combination of pressure and temperature.

Modifier_Ranking A **named** vector showing the percentage occurrence of each modifier in the Best_Modifier output list element.

See Also

sfe_mod

Examples

```
#Using limonene as an example
lim_cp <- 628.4626
lim_hsp <- c(dD = 14.00539, dP = 0, dHB = 1.7012, dP_low = 2.0448, dHB_low = 1.2838)
modif <- c("Methanol", "Ethanol", "Hexane")
res <- hsp_optim(lim_cp, lim_hsp, modif)
```

---

kin_export *Export the results of one or more kinetic models and their summaries*

Description

Compiles and exports the results and plots of Broken-and-Intact Cells (BIC) or Two-Site Kinetic Desorption (TWS) model into a specified directory.

Usage

```
kin_export(
  modres,
  sumres = NA,
  expath = getwd(),
  which_mod = "all",
  plotpars = "default",
  plot_format = "png",
  silent = FALSE
)
```

Arguments

modres	The output of bicmod or ktsmod.
sumres	**Optional** results summarizing **several** kinetic models output from function kin_splot.
expath	An **existing** directory where the results should be exported. Defaults to the working directory.
which_mod	A character vector specifying which model results should be exported. Either "all" (default) or any of "sim" (simple BIC), "ct" (characteristic times BIC), "cmp" (complete BIC), and/or "tws" (two-site kinetic desorption).
plotpars	Either "default" or a **named** vector containing one or more of the following parameters: "w" (width, defaults to 10 inches), "h" (height, defaults to 12 inches), "psize" (point size, default to 12), "dpi" (resolution, defaults to 300 dpi).
plot_format	The plot format. One of: "png" (default) or "pdf".
silent	A logical. When FALSE (default), additional information is printed in the console.

Value

Results as a .CSV files and plots as either .PDF or .PNG files are output into a new directory in expath.

See Also

bicmod, ktsmod, kin_splot, show_pars

---

kin_plot *Create kinetic model plots*

Description

Creates a plot of BIC or other kinetic model showing extraction curve points and up to three regions of the modeled curve including the Constant Extraction Rate (CER), Falling Extraction Rate (FER), and the Diffusion-Controlled (DC) regions. Part of the bicmod workflow.

Usage

```
kin_plot(
  pts,
  mod,
  ptvars = c("x", "y"),
  modvars = NA,
  grp = c(mod = "model", reg = "period"),
  cols = "default",
  pltlabs = c(x = ptvars[1], y = ptvars[2]),
  draw = FALSE
)
```

Arguments

pts	A data.frame of original Overall Extraction Curve (OEC) points (including both temporal and signal variables).
mod	A data.frame of the model curve to plot alongside pts.
ptvars, modvars	A character vector of length 2 specifying column names of the temporal and signal variables in pts and mod, respectively. If modvars is not provided, it is set to be equal to ptvars.
grp	An **optional** named character vector of length 2 specifying column names with grouping variables for **model type** ("mod") and **OEC region** ("reg").
cols	Either "default" or a named character vector of plot colours for points ("id"), model curve without sub-division into regions ("mod"), as well as with sub-division into CER ("cer"), FER ("fer"), and DC ("dc") regions of the extraction curve.
pltlabs	An **optional** named character vector of labels for the temporal ("x") and signal ("y") variables to display in plots. Defaults to ptvars.
draw	A logical switch specifying whether the generated plot(s) should be printed? Defaults to FALSE.

Value

A list of one or more plots of class "ggplot".

See Also

bicmod, ktsmod, show_pars

---

kin_splot *Visually compare several kinetic extraction models*

Description

Summarizes the results of between two and five kinetic models both visually and in a table.

Usage

```
kin_splot(
  m,
  which_mod,
  mvars = c(x = "x", y = "y"),
  cols = "default",
  leglab = NA,
  axlab = c(title = NA, x = NA, y = NA)
)
```

Arguments

m	A list of **2-5** output objects from function bicmod **OR** ktsmod.
which_mod	A string specifying which type of models to summarize and plot. **One** of "sim" (simplified BIC), "ct", "cmp2" or "cmp3" (complete BIC with 2 or 3 OEC regions, respectively), "tws" (two-site kinetic desorption).
mvars	A **named** character vector of column names specifying "x" and "y" values in all results in m. Defaults to c(x = "x", y = "y").
cols	Either "default" or a **named** character vector of point/line colours for models "one", "two", "three", "four", and/or "five". Unspecified colours are set to the following defaults: c(one = "darkgreen", two = "darkorange", three = "darkred", four = "blue", five = "purple").
leglab	A character vector of legend labels for models in the same order as in argument m.
axlab	A **named** character vector of plot labels for the x-axis ("x"), y-axis ("y"), and the title ("title").

Value

A list containing the following results:

1. **$plot**: A summary plot containing experimental points and modeled curves for all models of type which_mod in argument m.
2. **$summary**: A data.frame summarizing key input and calculated parameters of evaluated models, as well as three standard error metrics (see moderr). See/use bicmod, ktsmod, and show_pars for parameter descriptions.
3. **$model_type**: A character string stating the value of which_mod (the summarized model type) for user information.

See Also

bicmod, ktsmod, show_pars, moderr

Examples

```
#Prepare input data
bicdt <- list(sfex[["rizza1"]][["data"]],
sfex[["rizza2"]][["data"]],
sfex[["rizza3"]][["data"]])

#Estimates of extractable solute fraction
cuvals <- c(0.165, 0.21, 0.08)
```

```
#Generate several BIC models
biclst <- list()
for(i in seq_along(bicdt)) {
  biclst[[i]] <- bicmod(oec = bicdt[[i]],
                        oec_vars = c(x = "Time_min", y = "Yield_g", slv = "Solvent_mL"),
                        pars = c(pres = 300,
                                 cu = cuvals[i],
                                 temp = 45,
                                 flow = NA,
                                 mass_in = 0.5125,
                                 moisture = 8.6,
                                 D = 0.015,
                                 L = 0.015,
                                 etoh = 0.5,
                                 dr = 1554,
                                 dp = 0.0004,
                                 n = 2),
                        opt_est = "default",
                        flowpar = c(1.01325, 25),
                       etoh_frac = 0.06, #For when CO2 flow not provided but 'etoh' is >0
                        ro_co2 = NA,
                        tmax = NA,
                        qmax = NA,
                        cumulative = FALSE,
                        mass_flow = FALSE,
                        draw = FALSE,
                        units = c(flow = "none", resp = "g"),
                        modtype = "all") #"cu"
}

#Summarize the results
bic_summary <- kin_splot(m = biclst,
                         which_mod = "cmp3",
                         mvars = c(x = "x", y = "y"), #c("x", "y")
                         leglab = c("S. obliquus", "N. salina", "C. protothecoides"),
                        axlab = c(title = "BIC models", x = "q (kg/kg)", y = "e (kg/kg)"))
```

---

ktsmod *Construct two-site kinetic desorption models*

Description

Derives, evaluates, and visualizes the results of two-site kinetic desorption (TWS) models. Useful for subcritical water extraction (SWE) process evaluation. See **Details** for further information.

Usage

```
ktsmod(
  oec,
  oec_vars,
  pars,
  units = "default",
```

```
  opt_est = "default",
  plot_units = c(x = "time", y = "cc0"),
  flowpar = rep(NA, 2),
  ro_h2o = NA,
  tmax = NA,
  qmax = NA,
  cumulative = TRUE,
  mass_flow = FALSE,
  draw = TRUE,
  optmet = "nlopt"
)
```

Arguments

`oec`	A `data.frame` of OEC data. Must include all `oec_vars`.
`oec_vars`	A **named** `character` vector of column names included in `oec`. These **must** include the extraction time (`"x"`) and response (`"y"`) with appropriate `units`. A third parameters indicating the solvent usage (`"slv"`) is **optional** and may be provided as an **alternative** to an explicit flow rate normally provided in `pars["flow"]` - this is useful when the flow rate is uneven throughout the extraction process.
`pars`	A named `numeric` vector of input parameters for the model. **Mandatory** parameters include the pressure (`"pres"`; bar), temperature (`"temp"`; degrees Celsius), mass of raw material loaded (`"m_in"`; g), and the maximum possible yield (`"c0"`; units identical to those given in `units`). **Optional** parameters include the fraction of easily-desorbed solute (`"f"`) and the flow rate of water (`"flow"`; units given in `units`).
`units`	A **named** `character` vector of length 1 or 2 specifying the units of `"flow"` (one of `"mL/min"`, the **default** `"g/min"`, `"kg/h"`, `"L/h"`, or `"none"`) and/or the response `"resp"` (one of `"g"`, the **default** `"percent"`, `"permille"`, `"ppm"`, or `"ppb"`). Where not provided, default values are used.
`opt_est`	Either `"default"` or a **named** `numeric` vector of initial parameter estimates for iterative optimization. May include any of the first-order rate constants k_1 and k_2 (`"k1"` and `"k2"`), and/or the fraction of easily-desorbed solute F (`"f"`, **only required if it is not explicitly provided in** `pars`). If not provided, the following default values are used: `c(k1 = 0.1, k2 = 0.1, f = 0.5)`.
`plot_units`	A **named** `character` vector of length 2 specifying which units to use for the x- and y-axis of the model plot(s). Possible values are `c("time", "q")` for element `["x"]` and `c("abs", "cc0")` for element `["y"]`.
`flowpar`	Either `NA` (default) or a `numeric` vector of length 2 providing temperature and pressure at which flow rate of water is measured.
`ro_h2o`	The subcritical water density (in g/L) at extraction conditions. If not provided (`NA`; default), it is calculated via the IAPWS-95 formulation (see `h2o_dens`).
`tmax`, `qmax`	Maximum x-axis value (time, min for `tmax` and S/M ratio for `qmax`) to use for model predictions. Setting to `NA` defaults to 120% of the maximum experimental value.
`cumulative`	A `logical` switch specifying whether the **response** and/or **solvent consumption** values provided in `oec` are cumulative or not (defaults to `FALSE`).
`mass_flow`	A `logical` indicating whether the flow rate provided in `pars` is **mass** or **volumetric** (`FALSE`; default).

`draw`	A `logical` switch. Should generated plots be plotted? Defaults to `TRUE`.
`optmet`	The method of iterative curve fitting to use for estimating k_1, k_2, and (optionally) F. One of `"nlopt"` (Non-Linear Optimization via `nloptr`; default) or `"nlrob"` (Robust Fitting via `nlrob`).

Details

The TWS model is generated given the vector of time data t (e.g. extraction time), first-order rate constants k_1 and k_2 (in min^{-1}) describing the "fast" and "slow" component of kinetic desorption, and the fraction F of easily-desorbed target compound(s). The equation describing fractional yield e (ratio between extract collected at time t and the maximum amount of extractable material) is:

$$e = 1 - [F \times e^{-k_1 t}] - [(1 - F) \times e^{-k_2 t}]$$

See **References** for example applications of the TWS model for subcritical water extraction (SWE) studies.

Value

A named `list` with the following elements:

1. **$data**: A `data.frame` including time, response, S/M ratio (`$q`), mass yield of extract (`"yield_g"`), and fractional yield relative to the total amount of extractable solute (`"yield_cc0"`).
2. **$tws**: The summarized TWS model output from function `tws_cmp`.
3. **$plots**: The model visualisation (plot) of class `"ggplot"`.
4. **$input**: A list of input model parameters including pressure (`"pres"`), temperature (`"temp"`), flow rate (`"flow"` **in kg/s**), the maximum possible yield (`"c0"`), mass of loaded material (`"m_in"`), and the number of experimental observations (`"m"`).
5. **$call**: The function call.

References

Abidin, Z.Z., Samadi, M., Biak, D.R.A., Yunus, R. (2024), 'Mathematical Modelling Of Extraction Of Oil From *Aquilaria malacenssis* Wood Employing Subcritical Conditions', *Journal of Applied Science and Engineering* **27** (12), pp. 3725-3738, DOI: http://dx.doi.org/10.6180/jase.202412_27(12).0012.

Anepankul, T., Goto, M., Sasaki, M., Pavasant, P., Shotipruk, A. (2007), 'Extraction of anti-cancer damnacanthal from roots of *Morinda citrifolia* by subcritical water', *Separation and Purification Technology* **55**, pp. 343-349, DOI: http://dx.doi.org/10.1016/j.seppur.2007.01.004.

Jamaludin, R., Kim, D.-S., Salleh, L.M., Lim, S.-B. (2021), 'Kinetic Study of SubcriticalWater Extraction of Scopoletin, Alizarin, and Rutin from *Morinda citrifolia*', *Foods* **2021** (10), article 2260, DOI: https://doi.org/10.3390/foods10102260.

Pereira, D.T.V., Tarone, A.G., Cazarin, C.B.B., Barbero, G.F., Martinez, J. (2019), 'Pressurized liquid extraction of bioactive compounds from grape marc', *Journal of Food Engineering* **240**, pp. 105-113, DOI: https://doi.org/10.1016/j.jfoodeng.2018.07.019.

See Also

`tws_cmp`, `kin_splot`, `show_pars`, `moderr`, `bicmod`

Examples

```
twosite_res <- ktsmod(oec = swex[["duba1"]][["data"]],
oec_vars = c(x = "Time_min", y = "Yield_100C"),
pars = c(pres = 15, temp = 100, flow = 2, c0 = 77, m_in = 2, f = 0.24),
opt_est = "default",
units = c(flow = "mL/min", resp = "permille"),
plot_units = c(x = "q", y = "abs"), #"time" "q" "abs" "cc0"
cumulative = TRUE,
mass_flow = FALSE,
flowpar = rep(NA,2),
ro_h2o = NA,
tmax = NA,
qmax = NA,
optmet = "nlopt")
```

---

load_internal *Load internal package data where necessary*

Description

Load internal package data where necessary

Usage

```
load_internal(dtname)
```

Arguments

dtname	Name of dataset. One of `c("doe_base", "doe_lst1", "doe_lst2", "gcm_cnt", "gcm_int","gcm_smarts", "solv_dmass", "solv_dmol", "solv_mv")`

Value

The selected internal data.

Examples

```
## Not run:
load_internal("doe_lst1")

## End(Not run)
```

`miscomp` *Compare HSP-based miscibility enhancement for two or more solutes*

Description

Uses the `sfe_mod` framework to visually compare the miscibility enhancement provided by a chosen co-solvent for at least two solutes at various pressures (up to 6) and a set temperature.

Usage

```
miscomp(
  sols,
  tb = "SB_corr",
  crit = "NL07_robust",
  hsp = "SP12",
  modif = "Ethanol",
  pres = seq(100, 600, 100),
  pres_comp = pres,
  cols = "default",
  plt_title = TRUE,
  temp = 40,
  vfrac = seq(0.05, 0.4, 0.05),
  overlap = TRUE,
  draw = TRUE,
  silent = FALSE
)
```

Arguments

`sols`	A `list` where each element provides information for a solute as required by `mol_find`.
`tb`, `crit`, `hsp`	All `character` values of GCM methods to use. For available methods, see `est_gcm`.
`modif`	A `character` value of the co-solvent to use. For possible values, see `sfe_mod`.
`pres`	A `numeric` vector of **up to 6** pressures (in bar) to evaluate miscibility enhancement at.
`pres_comp`	A `numeric` pressure value to use for comparative plots between solutes. Must also be present in `pres`.
`cols`	Either `"default"` or a **named** vector of colours to use for up to six values of pressure (given in `pres`). Possible names are `"one"`, `"two"`, `"three"`, `"four"`, `"five"`, and/or `"six"`.
`plt_title`	A `logical` indicating whether titles should be added to plots.
`temp`	The single `numeric` temperature value to use (between 40-70 Celsius).
`vfrac`	A `numeric` vector of volume fractions of co-solvent to evaluate. Defaults to `seq(0.05, 0.40, 0.05)`.
`overlap`	A `logical` indicating whether overlapping sub-structures should be allowed (`TRUE` by default). This argument is experimental (see `sub_smarts`).

draw	A logical indicating whether generated plots should be printed automatically (TRUE by default).
silent	A logical. When FALSE (default), additional information is printed in the console.

Value

A named list containing the output data.frame ($data) with miscibility enhancement at all values of pressure and co-solvent volume fractions, and corresponding visualizations ($plots).

See Also

sfe_mod

Examples

```
#Get information about solute 1 (beta-carotene)
mol1 <- c("CC1=C(C(CCC1)(C)C)C=CC(=CC=CC(=CC=CC=C(C)C=CC=C(C)C=CC2=C(CCCC2(C)C)C)C)C",
"7235-40-7", "Beta-carotene")

#Get information about solute 2 (xanthohumol)
mol2 <- c(hspex[24,c("SMILES", "CAS", "Name")])

#Compare miscibility enhancement with ethanol as a co-solvent
gcm_comp <- miscomp(sols = list(mol1, mol2))
```

---

moderr *Calculate kinetic model errors (AARD, RMSE, R2)*

Description

Calculates the Average Absolute Relative Deviation (AARD), Round-Mean Squared Error (RMSE), and the R2 (R-Squared) value of any model given predicted and actual values alongside the number of recorded observations (data points).

Usage

```
moderr(act, pred, m = length(act))
```

Arguments

act, pred	Both numeric vectors of actual and model-predicted values **of equal length**.
m	**Optional** number of data points (by default, equals to the length of act).

Details

The Average Absolute Relative Deviation ($AARD$) is defined as:

$$AARD = 100 \times \sum |(x_{pred} - x_{act})/x_{act}|/m$$

The RMSE and R2 value are defined by:

$$RMSE = \sum \sqrt{(x_{pred} - x_{act})^2}/m$$

$$R2 = 1 - [\sum (x_{act} - x_{pred})^2]/[\sum (x_{act} - \bar{x}_{act})^2]$$

In all the above equations, m is the number of observations, while x_{pred} and x_{act} are predicted and actual response values, respectively. An overline denotes the mean value.

Value

A named `numeric` vector containing the calculated AARD (`"aard"`), RMSE (`"rmse"`), and R2 (`"r2"`) values.

See Also

`bicmod`, `ktsmod`, `bic_sm`, `bic_ct`, `bic_cmp`

`mol_find` *Get MOL from SMILES and derive molecular descriptors*

Description

Converts a SMILES string and (optionally) other information from SMILES to MOL. Part of the `sfe_mod` workflow.

Usage

```
mol_find(mol)
```

Arguments

`mol`	A `character` vector of `length` 1-3 containing the **mandatory CAS number** as well as **optional SMILES string** and **name** of a given molecule. The vector may **optionally** be named with `"Name"`, `"SMILES"`, and/or `"CAS"`. Otherwise the function will attempt automatic assignment based on values.

Value

A `list` of length 2 containing the following elements:

1. **IDs**: A named vector of descriptors, including SMILES, CAS number, Name, InChI, InChIKey, the number of atoms (`"Atom_Count"`), number of hydrogens (`"H_Count"`), molecular weight (`"MW"`), molecular formula (`"MF"`), and a `logical` specifying whether the molecule is aromatic (`"Aromaticity"`).
2. **Molfile**: A representation of the molecule in MOL format.

Examples

```
#Limonene input vector
mol <- c("CC1=CCC(CC1)C(=C)C", "5989-54-8", "Limonene")
#Get descriptors
res <- mol_find(mol)
```

---

oec_bp *Run segmented regression to estimate CER endpoint*

Description

Estimates the number of observation along an extraction curve using segreg or stepreg. May be used in preparation for the bicmod.

Usage

```
oec_bp(input, x, y, plt = TRUE, segmode = "step")
```

Arguments

input	A data.frame containing the time x and y values of the extraction curve.
x, y	Both are character column names identifying x- and y-values in the input data.
plt	A logical switch specifying whether to plot segmented regression results (defaults to TRUE).
segmode	Segmentation mode to use. One of: "step" (default) or "seg".

Value

A single numeric value showing the observation number at which the CER likely ends.

See Also

bicmod

Examples

```
oec_bp(input = sfex[[1]][["data"]], x = "time_min", y = "yield_g")
```

plot_gcm *Visualize molecules and sub-structures*

Description

Parts of the sfe_mod workflow. Visualizes a molecule from a MOL file and optionally highlights SMARTS substructures.

Usage

```
plot_gcm(solute, smarts, hlight = TRUE, draw_plot = TRUE, plot_title = TRUE)
```

Arguments

solute	MOL file and molecule descriptors output from function mol_find.
smarts	A character vector of SMARTS strings specifying substructures to be highlighted. Only relevant if hlight is TRUE.
hlight	A logical indicating whether substructures are highlighted.
draw_plot	A logical determining whether the plot is drawn.
plot_title	A logical determining whether the plot should be annotated.

Value

The molecule visualization as an array.

See Also

mol_find, sfe_mod

Examples

```
smarts <- c("[CX4H3]", "[CX3H2]", "[$([!R;#6X3H0]);!$([!R;#6X3H0]=[#8])]", "[R;CX4H2]", "[R;CX4H]",
"[R;CX3H1,cX3H1]", "[$([R;#6X3H0]);!$([R;#6X3H0]=[#8])]")
mol <- mol_find(c("CC1=CCC(CC1)C(=C)C", "5989-54-8", "Limonene"))
res <- plot_gcm(mol, smarts, draw_plot = TRUE)
```

scap *Capitalize first letter of every word in string*

Description

Capitalize first letter of every word in string

Usage

```
scap(x)
```

Arguments

x	A character string.

Value

The input character string with every word (separated by a space) capitalized.

Examples

```
scap("every word of this sentence should now be capitalized")
```

---

seq_last *Get sequence including the last number*

Description

Get sequence including the last number

Usage

```
seq_last(x, y, z)
```

Arguments

x, y	Starting and ending points of the sequence.
z	Increment of the sequence.

Value

A numeric vector containing the sequence inclusive of the last number.

Examples

```
seq_last(80,310,20)
```

---

sfex *Example Supercritical Fluid Extraction (SFE) kinetic modeling data*

Description

A named nested list containing extraction curve and Broken-and-Intact Cells (BIC) kinetic model data from scientific literature. Each element is itself a list including the following elements:

1. $data: A data.frame containing the OEC data (time, response and, optionally, amount of solvent expended).
2. $add_pars: Additional vector of logical parameters specifying whether the data is cumulative and whether the flow units are mass or volumetric.
3. $units: A named vector of units for the response (["resp"]) and flow (["flow"]).
4. $ext_pars: A vector of extraction parameters such as pressure and temperature.

5. `$input_pars`: Input model parameters (use `show_pars` for description and units of all parameters).
6. `$adj_pars`: Adjustable model parameters (use `show_pars` for description and units of all parameters).
7. `$proc_desc`: A short description of the modeled process.
8. `$citation`: A reference for the source publication.

Currently, the following Overall Extraction Curves (OEC) and BIC model results are included for SFE of:

1. `$dimic`: Cherry seed oil (Dimic et al., 2021).
2. `$rizza`: Lipids from microalgae (Rizza, 2014).

Usage

```
sfex
```

Format

An object of class `list` of length 4.

References

Dimic et al. (2021), 'Supercritical Fluid Extraction Kinetics of Cherry Seed Oil: Kinetics Modeling and ANN Optimization', *Foods* **10**, article 1513, DOI: https://doi.org/10.3390/foods10071513.

Rizza (2014), Experiments and Modeling of Supercritical CO2 Extraction of Lipids from Microalgae, Universita Degli Studi di Padova, Padua, Italy, MSc thesis.

See Also

`show_pars`

---

sfe_mod *Compare HSP-based utility of various co-solvents for SFE*

Description

Uses group contribution methods to estimate the boiling point, critical temperature, and/or Hansen Solubility Parameters of a solute given its SMILES string and MOL file (for example, output from `mol_find`). The estimated parameters are then used to calculate **solubility parameter distance** R_a between the solute and both pure carbon dioxide and its volumetric mixture with a given co-solvent. These distances may be compared for various co-solvents to determine the optimal solvent throughout a range of supercritical CO2 pressures and temperatures.

Usage

```
sfe_mod(
  solute,
  tb,
  crit,
  hsp,
  modif = "all",
  hlight = TRUE,
  overlap = TRUE,
  pres = seq_last(80, 300, 20),
  temps = seq_last(32, 65, 3),
  vfrac = 0.1,
  silent = FALSE
)
```

Arguments

`solute`	Information about the solute as required by function `mol_find`.
`tb`, `crit`, `hsp`	All are `character` values specifying the methods to use for GCMs (see `est_gcm`).
`modif`	A `character` value or vector indicating which co-solvent(s) to evaluate alongside pure CO2. One or more of: `"Acetone"`, `"Benzene"`, `"Toluene"`, `"OXylene"` (ortho-xylene), `"PXylene"` (para-xylene), `"Cyclohexane"`, `"DiethylEther"`, `"Methanol"`, `"Ethanol"`, `"Heptane"`, `"Hexane"`, and/or `"MethylOleate"`.
`hlight`	A `logical` indicating whether substructures are highlighted (see `plot_gcm`).
`overlap`	A `logical` indicating whether overlapping sub-structures should be allowed (`TRUE` by default). This argument is experimental (see `sub_smarts`).
`pres`, `temps`	Both are `numeric` values **or sequences** of pressures (75-700 bar) and temperatures (32-70 Celsius) at which to evaluate solute and solvent HSPs. Defaults are `seq(80,300,20)` and `seq(32,65,3)` for pressure and temperature, respectively.
`vfrac`	The volume fraction of co-solvent to use (defaults to `0.10`, or 10%).
`silent`	A `logical`. When `FALSE` (default), additional information is printed in the console.

Details

The workflow is based on various group contribution methods (see `est_gcm` for relevant sources), the Hansen Solubility Theory (Hansen, 2007), and the work of Tirado et al. (2018, 2019) and Diego & Calvo (2019). Given the SMILES string and molecular geometry (MOL file) of a solute, the boiling point, critical temperature, and Hansen Solubility Parameters (HSPs) are estimated via GCMs. For pure CO2 and co-solvents, these values as well as the molar volume (mL/mol) were compiled from CoolProp (Bell et al., 2014). The influence of temperature on **solute** HSPs is then calculated from **reduced temperatures** via the following equation where T_{r2} and T_{r1} are given by $T_{actual}/T_{critical}$ at the desired temperature and 298.15 K, respectively:

$$\delta_2 = (\frac{1 - T_{r2}}{1 - T_{r1}})^{0.34} \times \delta_1$$

The effects of temperature on dispersion (δ_d), polarity (δ_p), and hydrogen bonding (δ_{HB}) HSPs are calculated via the following equations:

$$\delta_{d\,ref}/\delta_d = (V_{ref}/V)^{-1.25}$$

$$\delta_{p\,ref}/\delta_p = (V_{ref}/V)^{-0.5}$$

$$\delta_{HB\,ref}/\delta_{HB} = exp[-1.32 \times 10^{-3} \times (T_{ref} - T) - ln(V_{ref}/V)^{-0.5}]$$

The HSP distance R_a between the solvent (pure CO2 and/or CO2+co-solvent mixture, subscript 1) and solute (subscript 2) is then derived using:

$$R_a = \sqrt{4 \times (\delta_{d1} - \delta_{d2})^2 + (\delta_{p1} - \delta_{p2})^2 + (\delta_{HB1} - \delta_{HB2})^2}$$

The above calculations are repeated for every combination of pressure and temperature. The miscibility enhancement (ME, in %) is finally derived as the percentage ratio between the solute R_a to that of CO2+co-solvent and pure CO2:

$$ME\ (\%) = (1 - (R_{a\ scCO_2 + co-solvent}/R_{a\ pure\ scCO_2})) \times 100$$

Marking the miscibility enhancement values at all combinations of pressure and temperature allows the determination of the best-suited co-solvent for any given solute.

Value

A named `list` with the following elements:

solute_ids Solute information as retrieved by `mol_find`.

parameters Solute parameters (boiling point, critical parameters, and HSPs) estimated by GCM via function `est_gcm`.

sfe Results of miscibility enhancement assessment as output from `hsp_optim`.

gcm_vis A `list` of raster visualisation included as part of `est_gcm` output.

call The function `call`.

References

Bell, Ian H., Wronski, Jorrit, Quoilin, Sylvain, Lemort, Vincent (2014), 'Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp', *Industrial & Engineering Chemistry Research* **53** (6), pp. 2498-2508, DOI: https://doi.org/10.1021/ie4033999.

Hansen, Charles M. (2007), *Hansen Solubility Parameters: A User's Handbook (2nd edition)*, CRC Press, London, United Kingdom.

Tirado, Diego F., Tenorio, Maria Jose, Cabanas, Albertina, Calvo, Lourdes (2018), 'Prediction of the best cosolvents to solubilise fatty acids in supercritical CO2 using the Hansen solubility theory', *Chemical Engineering Science* **190**, pp. 14-20, DOI: https://www.doi.org/10.1016/j.ces.2018.06.017.

Tirado, Diego F., Rousset, Amandine, Calvo, Lourdes (2019), 'The Selective Supercritical Extraction of High-value Fatty Acids from Tetraselmis suecica using the Hansen Solubility Theory', *Chemical Engineering Transactions* **75**, pp. 133-138, DOI: https://www.doi.org/10.3303/CET1975023.

Tirado, Diego F., Calvo, Lourdes (2019), 'The Hansen theory to choose the best cosolvent for supercritical CO2 extraction of beta-carotene from *Dunaliella salina*', *The Journal of Supercritical Fluids* **145**, pp. 211-218, DOI: https://www.doi.org/10.1016/j.supflu.2018.12.013.

See Also

`mol_find`, `est_gcm`, `plot_gcm`, `hsp_optim`, `show_solv`

Examples

```
#Retrieve molecule (beta-carotene)
mol <- c("CC1=C(C(CCC1)(C)C)C=CC(=CC=CC(=CC=CC=C(C)C=CC=C(C)C=CC2=C(CCCC2(C)C)C)C)C)C",
"7235-40-7", "Beta-carotene")

#Assess SFE miscibility enhancement with various co-solvents
optres <- sfe_mod(solute = mol,
tb = "SB_corr",
crit = "NL07_robust",
hsp = "SP12")
```

---

show_pars *Show kinetic model parameters and their description*

Description

Displays names, units, and descriptions of kinetic model and Cost of Manufacturing calculation parameters implemented in **supeRcrit**.

Usage

```
show_pars(type)
```

Arguments

type	The type of model to show parameters for. One or more of: "bic" (BIC model), "ts" (two-site kinetic desorption model), and/or "com" (Cost of Manufacturing).

Value

A data.frame with types, names, units, and descriptions of model parameters.

Examples

```
#Show parameter list for the Broken and Intact Cells (BIC) model
show_pars("bic")
```

---

show_solv *HSP data for various solvents*

Description

Displays the Hansen Solubility Parameters (HSP), molecular weight, molar volume, boiling point, critical temperature and pressure,

Usage

```
show_solv()
```

Value

A `data.frame` of solvent data to be used for HSP-based calculations of relative solubility improvement. The columns include solvent name (`"Solvent"`), abbreviation for function arguments (`"Abbreviation"`), the `"CAS"` number, molecular weight (`"MW"`), molar volume in mL/mol (`"MV"`), Hansen Solubility Parameters (dispersion `"dD"`, dipole moment `"dP"`, and hydrogen bonding `"dH"`), boiling point temperature (`"Tb"`), triple point temperature (`"Ttp"`), as well as the critical temperature (`"Tc"`), pressure (`"Pc"`), and density in mol/L (`"RHOc"`). All temperatures are in Kelvin.

---

splitAt *Split vector at specific position(s)*

Description

Splits an input atomic vector into two or more vectors at specific indices. Function adapted from a StackOverflow answer (see **References**).

Usage

```
splitAt(x, pos)
```

Arguments

`x`	An `atomic` vector to split.
`pos`	A `numeric` position (or vector thereof) at which to split the input vector `x`.

Value

A `list` of output vectors resulting from splitting `x` at position(s) given in `pos`.

References

StackOverflow answer by **flodel**: https://stackoverflow.com/questions/16357962/r-split-numeric-vector-a

Examples

```
splitAt(seq(1,5,1), c(2,4))
```

---

sub_smarts *Identify GCM groups in a molecule*

Description

Part of the `sfe_mod` workflow. Identifies GCM groups in a solute provided SMARTS substrings.

Usage

```
sub_smarts(solute, method, overlap = TRUE)
```

Arguments

solute	Molecular descriptors and MOL file output from function mol_find.
method	The GCM method to use as a character string. One of: Joback & Reid ("JR"), Stein & Brown ("SB"), their corrected counterparts ("JR_corr" and "SB_corr"), Nannoolal (2004; "NL04"), Nannoolal (2007; "NL07") and its robust counterpart ("NL07_robust", Stefanis & Panayiotou (2008 and 2012; "SP08" and "SP12", respectively), and their counterparts including only **first-order** groups ("SP08_first", "SP12_first"). See **Details** and **References**.
overlap	A logical indicating whether overlapping sub-structures should be allowed (TRUE by default). This argument is experimental.

Value

A data.frame of all GCM group contributions for the chosen method ($contrib_data), corresponding SMARTS strings ($smarts_data), group occurrences in the input molecule ($occurrences), and additional settings ($method and $overlap).

See Also

mol_find, sfe_mod

Examples

```
mol <- mol_find(c("CC1=CCC(CC1)C(=C)C", "5989-54-8", "Limonene"))
res <- sub_smarts(mol, "JR")
```

---

swex *Example Subcritical Water Extraction (SWE) kinetic modeling data*

Description

A named nested list containing extraction curve and two-site kinetic desorption model data from scientific literature. Each element is itself a list including the following elements:

1. $data: A data.frame containing the OEC data (time and response).
2. $resp_unit: The response value units.
3. $ext_pars: A vector of extraction parameters such as pressure and temperature.
4. $input_pars: Input model parameters (use show_pars for description and units of all parameters).
5. $output_pars: Output model parameters (use show_pars for description and units of all parameters).
6. $proc_desc: A short description of the modeled process.
7. $citation: A reference for the source publication.

Currently, the following Overall Extraction Curves (OEC) and kinetic desorption model results are included for SWE of:

1. $abidin: Cherry seed oil (Dimic et al., 2021).

2. `$duba1`: Grape skins (Duba, 2015).
3. `$duba2`: Grape seeds (Duba, 2015).
4. `$jamaludin1`: Alizarin from *Morinda citrifolia* (Jamaludin et al., 2021).
5. `$jamaludin2`: Scopoletin from *Morinda citrifolia* (Jamaludin et al., 2021).
6. `$kim`: Citrus flavonoids (Kim et al., 2020).
7. `$vasquez`: Anthraquinones from *Heterophyllaea pustulata* (Barrera Vasquez et al., 2015).

Usage

```
swex
```

Format

An object of class `list` of length 7.

References

Abidin, Z.Z., Samadi, M., Biak, D.R., Awang Yunus, R. (2024), 'Mathematical Modelling of Subcritical Water Extraction of Essential Oil From Aquilaria Malacenssis Wood', *Journal of Applied Science and Engineering* **27** (12), pp. 3725-3738, DOI: *https://www.doi.org/10.6180/jase.202412_27(12).0012*.

Barrera vazquez, M.F., Comini, L.R., Milanesio, J.M., Nunez Montoya, S.C., Cabrera, J.L., Bottini, S., Martini, R.E. (2015), 'Pressurized hot water extraction of anthraquinones from *Heterophyllaea pustulata* Hook f. (Rubiaceae)', *Journal of Supercritical Fluids* **101**, pp. 170-175, DOI: https://dx.doi.org/10.1016/j.supflu.2015.02.029.

Duba (2015), Supercritical Technologies for the Valorization of Wine Industry By Products, University of Trento, PhD thesis.

Jamaludin, R., Kim, D.-S., Salleh, L.M., Lim, S.-B. (2021), 'Kinetic Study of SubcriticalWater Extraction of Scopoletin, Alizarin, and Rutin from *Morinda citrifolia*', *Foods* **10**, article 2260, DOI: https://doi.org/10.3390/foods10102260.

Kim, D.-S. (2020), Subcritical Water Extraction and Hydrolysis of Citrus Flavonoids: Kinetics, Optimization, and Biological Activities, Jeju National University, South Korea, PhD thesis.

See Also

show_pars

---

twoway *Get all two-way interactions from a linear/RSM model*

Description

Obtains all interaction and quadratic terms of a linear or quadratic model given labels of coded factors alongside model order. Part of the doe_analyze workflow.

Usage

```
twoway(c_facs, mod_order = 2)
```

Arguments

c_facs	A character vector of **coded** factors. **Elements must all be capital letters**.
mod_order	The numeric model order, either one of 1, 1.5, or 2.

Value

A data.frame of 3 columns containing the main, interaction, and/or quadratic terms (depending on mod_order) in 3 different representations (data, formula, and label).

Examples

```
twoway(c("A","B","C"), 2)
```

---

tws_cmp *Build two-site kinetic desorption model*

Description

Generates and summarizes a two-site kinetic desorption (TWS) model. Part of the ktsmod workflow.

Usage

```
tws_cmp(
  x,
  y,
  c0,
  q = NA,
  f = NA,
  k1_0 = 0.5,
  k2_0 = 0.5,
  f0 = NA,
  est_f = TRUE,
  modpts = 100,
  aggreg = "aard",
  maxt = round(max(x) * 1.2, -1),
  maxq = round(max(q) * 1.2, -1),
  optmet = "nlopt"
)
```

Arguments

x, y	Both numeric vectors of time (t; min) and response (e_t; e.g. yield), respectively. Must be of equal length. Missing data is not allowed.
c0	The maximum possible yield e_0. Must have **units identical to those of** y.
q	An **optional** numeric vector of Solvent-Material (S/M) ratio q to be included in the final output.
f	An **explicit** value of the fraction of easily-desorbed extract F (between 0 and 1). When provided (NA by default), iterative curve fitting of this parameter is not carried out (est_f is ignored).

k1_0, k2_0	**Mandatory** initial estimates for first-order rate constants k_1 and k_2.
f0	**Optional** initial estimate for F.
est_f	A logical indicating whether F should be estimated via iterative curve fitting (TRUE by default).
modpts	The number of points to model between 0 and maxq (or maxt). Defaults to 100.
aggreg	A string specifying how the "best" results of non-linear optimization (nlsm) should be chosen. One of "aard" (default; value with minimum Average Absolute Relative Deviation is chosen) or "mean" (the arithmetic mean of all results is taken).
maxt, maxq	Maximum x-axis value (solvent expended, kg/kg insoluble solid for maxq, time for maxt) to use for model predictions. Defaults to 120% of the maximum experimental value.
optmet	The method of iterative curve fitting to use for estimating k_1, k_2, and (optionally) F. One of "nlopt" (Non-Linear Optimization via nloptr; default) or "nlrob" (Robust Fitting via nlrob).

Value

A list with the following elements:

1. **$ordt**: A data.frame containing the original input data including the extraction time (x), **optionally** the S/M ratio (q), the actual response (y, usually yield) and associated model predictions ("pred_y"), as well as the actual and predicted fractional yield ("cc0" and "pred_cc0") defined as the ratio of yield at time e_t and the maximum possible yield e_0.
2. **$mdt**: A data.frame of modeled data including the model type ($model), the extraction time ("x"), the **optional** S/M ratio ("q"), the response ("y"), and the fraction of maximum attainable yield ("cc0").
3. **$mod_pars**: A named numeric vector of input and estimated model parameters including the first-order rate constants k_1 and k_2 (c("k1","k2")), the fraction of easily desorbed solute F ("f"), and the maximum possible yield of extractable material e_0 ("c0").
4. **$fit_pars**: A character vector of which parameters in $mod_pars were iteratively fit to the model.
5. **$resid**: A named numeric vector of various error values for the model, including Average Absolute Relative Deviation AARD ("aard"), round mean squared error RMSE ("rmse"), and the R2 value ("r2").

See Also

ktsmod, tws_eq, moderr

Index