## Package ‘LLSR’

March 5, 2019

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Data Analysis of Liquid-Liquid Systems using R</td>
</tr>
<tr>
<td>Version</td>
<td>0.0.2.19</td>
</tr>
<tr>
<td>Date</td>
<td>2019-03-05</td>
</tr>
<tr>
<td>Author</td>
<td>Diego F Coelho <a href="mailto:diegofcoelho@gmail.com">diegofcoelho@gmail.com</a> [aut, cre], Pedro Victor Oliveira Menezes <a href="mailto:pedrod841@hotmail.com">pedrod841@hotmail.com</a> [dct], Carla Corina dos Santos Porto <a href="mailto:carlacorina@hotmail.com.br">carlacorina@hotmail.com.br</a> [dct], Jon George Huddleston <a href="mailto:Jonathan.Huddleston@brunel.ac.uk">Jonathan.Huddleston@brunel.ac.uk</a> [rev], Elias Basile Tambourgi <a href="mailto:eliastam@feq.unicamp.br">eliastam@feq.unicamp.br</a> [rev]</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Diego F Coelho <a href="mailto:diegofcoelho@gmail.com">diegofcoelho@gmail.com</a></td>
</tr>
<tr>
<td>Description</td>
<td>Originally design to characterise Aqueous Two Phase Systems, LLSR provide a simple way to analyse experimental data and obtain phase diagram parameters, among other properties, systematically. The package will include (every other update) new functions in order to comprise useful tools in liquid-liquid extraction research.</td>
</tr>
<tr>
<td>License</td>
<td>GPL-3</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>LazyData</td>
<td>TRUE</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.5)</td>
</tr>
<tr>
<td>Imports</td>
<td>rootSolve, XLConnect, digest, svDialogs, minpack.lm, ggplot2, svglite, dplyr, nleqslv, crayon</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://CRAN.R-project.org/package=LLSR">https://CRAN.R-project.org/package=LLSR</a></td>
</tr>
<tr>
<td>BugReports</td>
<td><a href="https://github.com/diegofcoelho/LLSR/issues">https://github.com/diegofcoelho/LLSR/issues</a></td>
</tr>
<tr>
<td>Suggests</td>
<td>testthat</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>6.1.1</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>no</td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2019-03-05 22:20:11 UTC</td>
</tr>
</tbody>
</table>
\section*{R topics documented:}

\begin{itemize}
\item AQSearch ........................................... 2
\item AQSearch.Binodal .................................. 3
\item AQSearch.Parameter ............................... 4
\item AQSearch.Slope ................................... 5
\item AQSearch.Tieline ................................. 6
\item AQSys .............................................. 7
\item AQSys.CritPoint ................................. 9
\item AQSys.data .................................... 10
\item AQSys.LevArmRule ............................ 11
\item AQSys.plot ..................................... 13
\item AQSysBancroft ................................ 14
\item AQSysCurve .................................... 15
\item AQSysDB ......................................... 16
\item AQSysDOE ....................................... 17
\item AQSysEval ...................................... 18
\item AQSysList ....................................... 19
\item AQSysOthmer .................................. 20
\item AQSysPlot ...................................... 21
\item export_data .................................... 22
\item export_template ................................ 22
\item llsr_data ...................................... 23
\item peg4kslt ........................................ 23
\end{itemize}

\section*{Index}

AQSearch \hfill \textit{Search function for ATPS Systems data}

\subsection*{Description}

This function allow the user to search the LLSR database to find any ATPS that matches the used criteria.

This function allow the user to search the package database to find any ATPS that matches the available criteria.

\subsection*{Usage}

AQSearch(db = LLSR::llsr_data, ...)

\# Default S3 method:
AQSearch(db = LLSR::llsr_data, db.CompA = NULL,
  db.CompB = NULL, db.CompC = NULL, db.Temp = NULL, db.ph = NULL,
  db.uid = NULL, stacked = FALSE, ...)

Arguments

- db: A highly structured db containing data from previously analysed data. LLSR database is used by default but user may input his own db if formatted properly.
- Additional optional arguments. None are used at present.
- db.compA: A String variable containing either the CAS, chemical formula or name of the upper phase enriched component.
- db.compB: A String variable containing either the CAS, chemical formula or name of the lower phase component.
- db.compC: A String variable containing either the CAS, chemical formula or name of the additive component.
- db.temp: A numeric variable containing the Temperature (in Kelvin) to be searched within DB.
- db.ph: A numeric variable containing the pH to be searched within DB.
- db.uid: An Unique md5 hash Identification. User can retrieve data for a specific system if in possession of its UID.
- stacked: A boolean variable used to return value as a nested list or a data.frame. Used internally to organize data output.

Details

The function returns the systems that match the criteria submitted by the user.

Value

Returns a data.frame containing system’s parameters which match searched conditions.

Examples

```r
## Not run:
AQSearch(db.compA = "Ammonium")

## End(Not run)
```

Description

This function allows the user to search the package database to find any ATPS that matches the available criteria.

Usage

```r
## S3 method for class 'Binodal'
AQSearch(db = LLSR::llsr_data, db.compA = NULL, 
          db.compB = NULL, db.compC = NULL, db.temp = NULL, db.ph = NULL, 
          db.uid = NULL, stacked = FALSE, ...)
```
Arguments

- **db**: A highly structure db containing data from previously analysed data. LLSR database is used by default but user may input his own db if formatted properly.
- **db.CompA**: A String variable containing either the CAS, chemical formula or name of the upper phase enriched component.
- **db.CompB**: A String variable containing either the CAS, chemical formula or name of the lower phase component.
- **db.CompC**: A String variable containing either the CAS, chemical formula or name of the additive component.
- **db.Temp**: A numeric variable containing the Temperature (in Kelvin) to be searched within DB.
- **db.ph**: A numeric variable containing the pH to be searched within DB.
- **db.uid**: An Unique md5 hash Identification. User can retrieve data for a specific system if in possession of its UID.
- **stacked**: A boolean variable used to return value as a nested list or a data.frame. Used internally to organize data output.
- ... Additional optional arguments. None are used at present.

Details

The function return the systems that matches the criteria submitted by the user.

Value

Returns a data.frame containing system’s parameters which match searched conditions

Examples

```r
## Not run:
AQSearch.Binodal(db.CompA="Ammonium")

## End(Not run)
```

Description

This function allow the user to search the package database to find any ATPS that matches the available criteria.

Usage

```r
## S3 method for class 'Parameter'
AQSearch(db = LLSR::llsr_data, db.CompA = NULL,
          db.CompB = NULL, db.CompC = NULL, db.Temp = NULL, db.ph = NULL,
          db.uid = NULL, stacked = FALSE, ...)
```
Arguments

- `db` A highly structure db containing data from previously analysed data. LLSR database is used by default but user may input his own db if formatted properly.
- `db.CompA` A String variable containing either the CAS, chemical formula or name of the upper phase enriched component.
- `db.CompB` A String variable containing either the CAS, chemical formula or name of the lower phase component.
- `db.CompC` A String variable containing either the CAS, chemical formula or name of the additive component.
- `db.Temp` A numeric variable containing the Temperature (in Kelvin) to be searched within DB.
- `db.ph` A numeric variable containing the pH to be searched within DB.
- `db.uid` An Unique md5 hash Identification. User can retrieve data for a specific system if in possession of its UID.
- `stacked` A boolean variable used to return value as a nested list or a data.frame. Used internally to organize data output.
- `...` Additional optional arguments. None are used at present.

Details

The function return the systems that matches the criteria submitted by the user.

Value

Returns a data.frame containing system’s parameters which match searched conditions.

Examples

```r
### Not run:
AQSearch.Parameter(db.CompA="Ammonium")
### End(Not run)
```

Description

This function allow the user to search the package database to find any ATPS that matches the available criteria.

Usage

```r
### S3 method for class 'Slope'
AQSearch(db = LLSR::llsr_data, db.CompA = NULL,  
db.CompB = NULL, db.CompC = NULL, db.Temp = NULL, db.ph = NULL,  
db.uid = NULL, stacked = FALSE, ...)
```
Arguments

db       A highly structure db containing data from previously analised data. LLSR database is used by default but user may input his own db if formatted properly.
db.CompA A String variable containing either the CAS, chemical formula or name of the upper phase enriched component.
db.CompB A String variable containing either the CAS, chemical formula or name of the lower phase component.
db.CompC A String variable containing either the CAS, chemical formula or name of the additive component.
db.Temp   A numeric variable containing the Temperature (in Kelvin) to be searched within DB.
db.ph     A numeric variable containing the pH to be searched within DB.
db.uid    An Unique md5 hash Identification. User can retrieve data for a specific system if in possession of its UID.
stacked  A boolean variable used to return value as a nested list or a data.frame. Used internally to organize data output.
...      Additional optional arguments. None are used at present.

Details

The function return the systems that matches the criteria submitted by the user.

Value

Returns a data.frame containing system’s parameters which match searched conditions

Examples

```r
## Not run:
AQSearch.Slope(db.CompA="Ammonium")
## End(Not run)
```

Description

This function allow the user to search the package database to find any ATPS that matches the available criteria.

Usage

```r
## S3 method for class 'Tieline'
AQSearch(db = LLSR::llsr_data, db.CompA = NULL,
      db.CompB = NULL, db.CompC = NULL, db.Temp = NULL, db.ph = NULL,
      db.uid = NULL, stacked = FALSE, ...)
```
Arguments

- **db**: A highly structured db containing data from previously analyzed data. LLSR database is used by default but user may input his own db if formatted properly.

- **db.NcompA**: A string variable containing either the CAS, chemical formula or name of the upper phase enriched component.

- **db.NcompB**: A string variable containing either the CAS, chemical formula or name of the lower phase component.

- **db.NcompC**: A string variable containing either the CAS, chemical formula or name of the additive component.

- **db.Ntemp**: A numeric variable containing the Temperature (in Kelvin) to be searched within DB.

- **db.Nph**: A numeric variable containing the pH to be searched within DB.

- **db.Nuid**: An unique md5 hash identification. User can retrieve data for a specific system if in possession of its UID.

- **stacked**: A boolean variable used to return value as a nested list or a data.frame. Used internally to organize data output.

- **...**: Additional optional arguments. None are used at present.

Details

The function returns the systems that match the criteria submitted by the user.

Value

Returns a data.frame containing system's parameters which match searched conditions.

Examples

```r
## Not run: 
AQSearch.Tieline(db.NcompA="Ammonium")

## End(Not run)
```

Description

- Perform a nonlinear regression fit using any of the several mathematical descriptors implemented in order to calculate the equation's parameters.
Usage

AQSys(dataSet, ...)

## Default S3 method:
AQSys(dataSet, modelName = "merchuk", Order = "xy", ...

Arguments

dataSet - Binodal Experimental data that will be used in the nonlinear fit ... Additional optional arguments. None are used at present.
modelName - Character String specifying the nonlinear empirical equation to fit data. The default method uses Merchuk's equation. Other mathematical descriptors can be listed using AQSysList().
Order - Defines how the data is organized in the Worksheet. Use "xy" whether the first column corresponds to the lower phase fraction and "yx" whether the opposite.

Details

The function returns functions parameters after fitting experimental data to the equations listed in AQSysList().

Value

A list containing three data.frame variables with all data parsed from the worksheet and parameters calculated through the available mathematical descriptions.

References


TANG, X. et al. The study of phase behavior of aqueous two-phase system containing [Cnmim] BF 4 (n=2, 3, 4)+(NH4)2SO4 + H2O at different temperatures. Fluid Phase Equilibria, v. 383, p. 100-107, 2014. ISSN 0378-3812. (ScienceDirect)


AQSys.CritPoint

See Also

• AQSys.default
• AQSys.plot
• AQSys LeveArmRule
• AQSys0thmer
• AQSys Bancroft

Examples

# Populating variable dataSet with binodal data
dataset <- peg4kst[, 1:2]
# Fitting dataSet using Merchuk's function
AQSys(dataset)

AQSys.CritPoint

ATPS Critical Point Calculation

Description

This function implements methods available in current literature to calculate an ATPS critical point based on its experimental data.

Usage

## S3 method for class 'CritPoint'
AQSys(dataSet, tldData, method, modelName = "merchuk",
slope = NULL, NP = 100, xmax = 30, xlbl = "", ylbl = "",
Order = "xy", ext = FALSE, ...)

Arguments

dataSet - Binodal Experimental data that will be used in the nonlinear fit. [type:data.frame]
tldData - A data.frame with two columns containing a set of Tieline's Slopes (S) and its bottom-rich component composition in the bottom phase (XB). [type:data.frame]
method - Binodal Experimental data that will be used in the nonlinear fit. [type:string]
   "algebraic" - Uses the critical point own definition to set up constraints and solve a system of equations. Still in development.
   "numerical" - A number of tie-lines are calculated successively until TLL is close to zero and concentration of components are numerically equal. A constant slope is assumed.
   "polynomial" - Calculate the intercept point between the chosen mathematical description and a third order polynomial fitting the tie-lines mid-points
modelName - Mathematical descriptor that will be used for non-linear fitting. Use AQSysList() to list the available equations. [type:string]
slope The method assumes all tielines for a given ATPS are parallel, thus only one slope is required. [type:double]
**Description**

The function returns a plot after fitting a dataset to the mathematical descriptor chosen by the user.

**Usage**

```r
## S3 method for class 'data'
AQSys(dataSET, modelName = "merchuk", Order = "xy",
      xmax = "", ymax = "", ...)```

**Details**

The Critical Point is one in which both the composition and volume of the phases become equal, and the tie-line length (TLL) tends to zero. Thus, the methods here implemented the methods described by KAUL, A (2000) to calculate a theoretical critical point.

**Value**

(XC, YC) - The function returns TieLine’s Critical Point Composition

**References**


**Examples**

```r
## Not run:
AQSys.CritPoint(dataSET, tldata)

## End(Not run)
```
Arguments

- **dataSet** - Binodal Experimental data that will be used in the nonlinear fit. It might hold multiple systems stacked side-by-side. [type:data.frame]
- **modelName** - Character String specifying the nonlinear empirical equation to fit data. [type: String] The default method uses Merchuk’s equation. Other mathematical descriptors can be listed using AQSysList().
- **Order** - Defines how the data is organized in the Worksheet. Use "xy" whether the first column corresponds to the lower phase fraction and "yx" whether the opposite. [type:String]
- **xmax** - Maximum value for the Horizontal axis’ value - optional [type:double]
- **ymax** - Maximum value for the Vertical axis’ value - optional [type:double]
- **NNN** - Additional optional arguments. None are used at present.

Details

This version uses the plot function and return a regular orthogonal plot.

Value

return a data.frame with data fitted using the chosen mathematical descriptor.

Examples

```r
# Populating variable dataSet with binodal data
dataSet <- peg4kslt[, 1:2]
# Fitting dataSet using Merchuk's function
data <- AQSys.data(dataSet, Order = "xy")
```

Description

Merchuk et al. described a very straightforward method to calculate the concentration of each component in the tie-line giving only its global composition and phase’s properties (such as volume and density). Here this method is implemented and generalized for multiple mathematical descriptors.

Usage

```r
## S3 method for class 'LevArmRule'
AQSys(dataSet, modelName = "merchuk", Xm, Ym,
       Vt = NULL, Vb = NULL, dyt = NULL, dyb = NULL, WT = NULL,
       WB = NULL, byW = TRUE, Order = "xy", ...)
```
**Arguments**

- **dataSET**: Binodal Experimental data that will be used in the nonlinear fit.
- **modelName**: Character String specifying the nonlinear empirical equation to fit data. The default method uses Merchuk’s equation. Other mathematical descriptors can be listed using AQSysList().
- **Xm**: Component X’s concentration in the tieline’s global composition.
- **Ym**: Component Y’s concentration in the tieline’s global composition.
- **Vt**: Tieline’s TOP phase volume.
- **Vb**: Tieline’s BOTTOM phase volume.
- **dyt**: Tieline’s TOP phase density.
- **dyb**: Tieline’s BOTTOM phase density.
- **WT**: ATPS upper phase weight.
- **WB**: ATPS bottom phase weight.
- **byW**: Use weight (TRUE) or volume and density (FALSE) during lever arm rule calculation.
- **Order**: Defines how the data is organized in the Worksheet. Use "xy" whether the first column corresponds to the lower phase fraction and "yx" whether the opposite.
- **NNN**: Additional optional arguments. None are used at present.

**Details**

Using any implemented binodal data mathematical descriptor, the global composition of a chosen tieline and its phases properties.

**Value**

The function returns the Critical Point (X,Y), Tieline Length (TLL), Tieline’s Equivolume point (xVRe2o,yVRe2o), and Tieline’s Slope.

**References**


**Examples**

```r
## Not run:
AQSys.LevArmRule(dataSET, Xm, Ym, Vt, Vb, dyt, dyb, WT, WB, byW = FALSE)

## End(Not run)
```
AQSys.plot

Dataset and Fitted Function plot

Description

The function returns a plot after fitting a dataset to the mathematical descriptor chosen by the user.

Usage

```r
## S3 method for class 'plot'
AQSys(dataSET, xlbl = "", ylbl = "", main = NULL,
      col = "blue", type = "p", cex = 1, cexlab = 1, cexaxis = 1,
      cexmain = 1, cexsub = 1, modelName = "merchuk", np = 100,
      xmax = "", ymax = "", Order = "xy", save = FALSE, HR = FALSE,
      filename = NULL, wdir = NULL, silent = FALSE, ...)
```

Arguments

- `dataSET` - Binodal Experimental data that will be used in the nonlinear fit. It might hold multiple systems stacked side-by-side. [type:data.frame]
- `xlbl` - Plot’s Horizontal axis label.
- `ylbl` - Plot’s Vertical axis label.
- `main` - Legacy from plot package. For more details, see `plot.default`
- `col` - Legacy from plot package. For more details, see `plot.default`
- `type` - Legacy from plot package. For more details, see `plot.default`
- `cex` - Legacy from plot package. For more details, see `plot.default`
- `cexlab` - Legacy from plot package. For more details, see `plot.default`
- `cexaxis` - Legacy from plot package. For more details, see `plot.default`
- `cexmain` - Legacy from plot package. For more details, see `plot.default`
- `cexsub` - Legacy from plot package. For more details, see `plot.default`
- `modelName` - Character String specifying the nonlinear empirical equation to fit data. The default method uses Merchuk’s equation. Other possibilities can be seen in `AQSysList()`.
- `np` - Number of points used to build the fitted curve. Default is 100. [type:Integer]
- `xmax` - Maximum value for the Horizontal axis’ value
- `ymax` - Maximum value for the Vertical axis’ value
- `Order` - Defines how the data is organized in the Worksheet. Use "xy" whether the first column corresponds to the lower phase fraction and "yx" whether the opposite.
- `save` - Save the generated plot in the disk using path and filename provided by the user. [type:Boolean]
- `HR` - Adjust Plot’s text to be compatible with High Resolution size [type:Boolean]
filename  Filename provided by the user to save a given plot. [type: String]
wdir  The directory in which the plot file will be saved. [type: String]
silent  save plot file without actually showing it to the user. [type: Boolean]
...  Additional optional arguments. None are used at present.

Details
This version uses the plot function and return a regular orthogonal plot.

Value
A plot containing the experimental data, the correspondent curve for the binodal in study and the curve’s raw XY data.

Examples

```
#Populating variable dataSet with binodal data
dataSet <- peg4kslt[, 1:2]
# Plot dataSet using Merchuk’s function
# AQSys.plot(dataSet)
```

Description
Bancroft’s equation to correlate tie-line’s data.

Usage

```
AQSysBancroft(dataSet, ...)
```

Arguments

```
dataset  - Tie-line Experimental data that will be used in the nonlinear fit
...  Additional optional arguments. None are used at present.
```

Value
Parameters k1, r and Statistical data

References
TUBIO, G. et al. Liquid-liquid equilibrium of the Ucon 50-HB5100/sodium citrate aqueous two-phase systems. Separation and Purification Technology, v. 65, n. 1, p. 3-8, 2009. ISSN 1383-5866. (ScienceDirect)
Examples

# data set is a data.frame which contains series of tie line's mass fraction
# (upper-rich component, bottom-rich component and water)
# Each column in the data.frame represents a series of one component mass fraction
# For example, an empty data.frame for four tie lines can be obtaining using:
dataSet <- matrix(NA,nrow=4,ncol=6)
# Variables order must follows the sequence presented below:
# "mfXt","mfYt","mfXb","mfYb","mfWt","mfWb"
# In which: mf stands for mass fraction; X and Y for the component
# rich in bottom and upper phase, respectively; t or b for top and
# bottom phases and W for water.
# Then you just need to load the data.frame in the function:
## Not run:
AQSysBancroft(dataSet, Order = "xy")
## End(Not run)

AQSysCurve

This functions plot a curve based in the chosen model and its parameters.

Description

The function returns a plot after using the parameters and model given by the user.

Usage

AQSysCurve(modelName, modelPars, seriesNames = NULL, xlbl = "", ylbl = "", col = "black", type = "p", cex = 1, cexlab = 1, cexaxis = 1, cexmain = 1, cexsub = 1, xmax = 35, HR = FALSE, NP = 100, filename = NULL, wdir = NULL, save = FALSE, silent = FALSE, ...)

Arguments

modelName        Equation to be used: merchuk, murugesan [type:string]
modelPars        Model’s parameters [type::data.frame]
seriesNames      A list of sequential names which will identify each system provided by the user in the dataSet variable. [type:List]
xB             Plot’s Horizontal axis label.
ylbl             Plot’s Vertical axis label.
col             Legacy from plot package. For more details, see plot.default
type             1-character string giving the type of plot desired. The following values are possible, for details, see plot: "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
cex Legacy from plot package. For more details, see `plot.default`
cexlab Legacy from plot package. For more details, see `plot.default`
cexaxis Legacy from plot package. For more details, see `plot.default`
cexmain Legacy from plot package. For more details, see `plot.default`
cexsub Legacy from plot package. For more details, see `plot.default`
xmax Maximum value for the Horizontal axis’ value (bottom-rich component) [type:double]
HR Adjust Plot’s text to be compatible with High Resolution size [type:Logical]
NP Number of points used to build the fitted curve. Default is 100. [type:Integer]
filename Filename provided by the user to save a given plot. [type:String]
wdir The directory in which the plot file will be saved. [type:String]
save Save the generated plot in the disk using path and filename provided by the user. [type:Logical]
silent save plot file without actually showing it to the user. [type:Logical]
... other graphical parameters (see `par` and section ‘Details’ below).

Details

The function owns predefined set of equations that can be seen below and must be used, with adequate parameters, to return a plot which represent the chosen model.

Value

A plot using the input model within the chosen interval and the curve’s raw XY data. If no interval is selected, xmax = 0.4.

Examples

```r
## Not run:
AQSysCurve("murugesan", data.frame(90.389, -34.897, 2.924), col = "red")

## End(Not run)
```

Description

Import DB data from an Excel Worksheet and process it through mathematical descriptors to output a highly structured variable comparable to a Database and which hold a list of references, chemicals and parameters for any implemented mathematical descriptors.

Usage

`AQSysDB(path)`
Arguments

- `path`: String containing the full path to the XLS or XLSX file.

Examples

```r
## Not run:
AQSysDB("C:/data.xlsx")

## End(Not run)
```

Description

The function uses a ATPS characterization data to build a Design Of Experiments (DOE) matrix based on Tie-Line Length (TLL) and Volume Ratio. See `AQSysEval` for more details.

Usage

```
AQSysDOE(dataSET, db = LLSR::llsr_data, slope = NULL, xmax = NULL, modelName = "merchuk", nTL = 3, nPoints = 3, tol = 1e-05)
```

Arguments

- `dataSET`: Binodal Experimental data that will be used in the nonlinear fit.
- `db`: A highly structure db containing data from previously analysed data. LLSR database is used by default but user may input his own db if formatted properly.
- `slope`: The method assumes all tielines for a given ATPS are parallel, thus only one slope is required. [type:double]
- `xmax`: Maximum value for the Horizontal axis’ value (bottom-rich component). [type:double]
- `modelName`: Character String specifying the nonlinear empirical equation to fit data. The default method uses Merchuk’s equation. Other mathematical descriptors can be listed using `AQSysList()`. [type:string]
- `nTL`: Number of tielines plotted for a given ATPS. Default is 3. [type:Integer]
- `nPoints`: Number of points chosen for a given tieline. Default is 3. [type:Integer]
- `tol`: Limit of tolerance to reach to assume convergence. Default is 1e-5. [type:Integer]
Examples

```
# dataSet is a data.frame which contains series of Tieline's mass fraction and information
# from both components and # extraction conditions (T, pH). The function perform a system
# characterization based on data stored in LLSR's database (or provided by the user)
# and then calculate a DOE based on the input.
## Not run:
dataSet <- AQSearch.Binodal(db.uid='56b53a50f500c502fa4a65d197fc6d84')
ans <- AQSysDOE(dataSet2, nTL = 5, nPoints = 5)
View(ans$DOE)

## End(Not run)
```

### Description

The function performs a full ATPS characterization (parameters, tie-line boundaries and critical point), generating a brief report.

### Usage

```
AQSysEval(dataSet, db = LLSR::llsr_data, xmax = NULL, ymax = NULL,
          NP = 100, slope = NULL, modelName = "merchuk",
          convrgncelines = FALSE, nTL = 3, nPoints = 3, tol = 1e-05,
          xlbl = "", ylbl = "", seriesNames = NULL, save = FALSE,
          HR = FALSE, autoname = FALSE, wdir = NULL, silent = TRUE)
```

### Arguments

- `dataSet` - Binodal Experimental data that will be used in the nonlinear fit. [type:data.frame]
- `db` - A highly structure db containing data from previously analysed data. LLSR database is used by default but user may input his own db if formatted properly.
- `xmax` - Maximum value for the Horizontal axis' value (bottom-rich component). [type:double]
- `ymax` - Maximum value for the vertical axis' value (bottom-rich component). [type:double]
- `NP` - Number of points used to build the fitted curve. Default is 100. [type:Integer]
- `slope` - The method assumes all tielines for a given ATPS are parallel, thus only one slope is required. [type:double]
- `modelName` - Character String specifying the nonlinear empirical equation to fit data. The default method uses Merchuk’s equation. Other mathematical descriptors can be listed using AQSysList(). [type:string]
- `convrgncelines` - Magnify Plot's text to be compatible with High Resolution size [type:Logical]
- `nTL` - Number of tielines plotted for a given ATPS. Default is 3. [type:Integer]
- `nPoints` - Number of points chosen for a given tieline. Default is 3. [type:Integer]
The function returns a list of all mathematical descriptors available at the time.

Usage

AQSysList(npars = FALSE)

Arguments

npars Logic option to return a List variable containing the number of required parameters for each equation.
Othmer’s Equation - Tieline’s correlation

Description

Othmer’s equation to correlate tieline’s data applying the lever’s rule.

Usage

AQSysOthmer(dataSET, ...)

Arguments

dataSET - Tieline Experimental data that will be used in the nonlinear fit
...

Additional optional arguments. None are used at present.

Value

Parameters A, B and Statistical data

References


Examples

# dataSET is a data.frame which contains series of Tieline's mass fraction
# (upper-rich component, bottom-rich component and water)
# Each column in the data.frame represents a series of one component mass fraction
# For example, an empty data.frame for four tielines can be obtaining using:
dataSET<-matrix(NA,nrow=4,ncol=6)
# Variables order must follows the sequence presented below:
# "mfXt","mfYt","mfXb","mfYb","mfWt","mfWb"
# In which: mf stands for mass fraction; X and Y for the component
# rich in bottom and upper phase, respectively; t or b for top and
# bottom phases.
# Then you just need to load the data.frame in the function:
# Not run:
AQSysOthmer(dataSET, Order = "xy")

# End(Not run)
**Description**

This function plots binodal data as a curve in a pre-defined high quality theme ready for publication.

**Usage**

```r
AQSysPlot(dataSet, Order = "xy", xlbl = "", ylbl = "",
seriesNames = NULL, save = FALSE, filename = NULL, HR = FALSE,
wdir = NULL, silent = FALSE)
```

**Arguments**

- `dataSet`: Binodal Experimental data that will be used in the nonlinear fit. It might hold multiple systems stacked side-by-side. [type: data.frame]
- `Order`: Defines how the data is organized in the Worksheet. Use "xy" whether the first column corresponds to the lower phase fraction and "yx" whether the opposite. [type: string]
- `xlbl`: Plot's Horizontal axis label.
- `ylbl`: Plot's Vertical axis label.
- `seriesNames`: A list of sequential names which will identify each system provided by the user in the dataSet variable. [type: List]
- `save`: Save the generated plot in the disk using path and filename provided by the user. Default is False. [type: Logical]
- `filename`: A filename provided by the user to save a given plot. No default is provided. [type: String]
- `HR`: Adjust Plot's text to be compatible with High Resolution size [type: Logical]
- `wdir`: The directory in which the plot file will be saved. [type: String]
- `silent`: save plot file without actually showing it to the user. Default is FALSE. [type: Logical]

**Details**

The function have a predefined set of ggplot2 features adjusted to return a high quality picture. More suitable for plotting fitted data, once binodal data usually determined through cloudy-phase point will look irregular.

**Value**

The plot is returned as a ggplot2 object that can be manipulated accordingly.
Examples

# Populating variable dataSet with binodal data
dataSet <- peg4kSlt[, 1:2]
# Fitting dataSet using Merchuk's function
data <- AQSys.data(dataSet, Order = "xy")
AQSysPlot(data)

Description

The function saves a copy of a specified variable to a file in the folder pointed by the user.

Usage

export_data(localData = NULL)

Arguments

localData A variable existing in R environment and that will be saved locally.

Description

The function makes a copy of LLSR’s template file and copy it to the folder pointed by the user.

Usage

export_template()
**llsr_data**

---

**llsr_data**

**LLSR’s database**

**Description**

A database is a highly structured collection of data generally stored and accessed from a computer system and stores raw data and parameters for all analyzed ATPS phase diagrams.

**Format**

multi-level data.frame()

- **db.ref** Table from LLSR’s database in which the references for all used manuscripts are stored.
- **db.sys** Table from LLSR’s database in which parameters for all implemented mathematical descriptors are stored. Statistic data is also available.
- **db.cas** Table from LLSR’s database in which information regarding all chemicals (such as name and CAS number) used in the collected data.
- **db.data** Table from LLSR’s database in which raw experimental data is tabulated. The data was used to calculate all properties made available in this package.
- **db.tielines** Table from LLSR’s database in which raw experimental data, used to calculate tie-lines compositions and slopes, are stored.

**Source**

https://github.com/diegofcoelho/LLSR/

**Examples**

```r
#
XYdt <- llsr_data[["db.data"]][, 1:2]
Xdt <- llsr_data[["db.data"]][, 1]
Ydt <- llsr_data[["db.data"]][, 2]
```

---

**peg4kslt**

**Dataset of experimental binodal data of an ATPS**

**Description**

A dataset containing the experimental binodal data for a PEG/SALT Aqueous Two-Phases System (ATPS)

**Format**

multi-level data.frame()

- **XC** Ammonium Sulphate mass fraction
- **YC** Poly(ethylene glycol) mass fraction...
Examples

```r
#
XYdt <- peg4kslt[,1:2]
#
Xdt<-peg4kslt[,1]
#
Ydt<-peg4kslt[,1]
```
# Index

AQSearch, 2  
AQSearch.Binodal, 3  
AQSearch.Parameter, 4  
AQSearch.Slope, 5  
AQSearch.Tieline, 6  
AQSys, 7  
AQSys.CritPoint, 9  
AQSys.data, 10  
AQSys.default, 9  
AQSys.LevArmRule, 9, 11  
AQSys.plot, 9, 13  
AQSysBancroft, 9, 14  
AQSysCurve, 15  
AQSysDB, 16  
AQSysDOE, 17  
AQSysEval, 17, 18  
AQSysList, 19  
AQSysOthmer, 9, 20  
AQSysPlot, 21  

export_data, 22  
export_template, 22  

graphical parameters, 16  

llsr_data, 23  

par, 16  
peg4kslt, 23  
plot, 15  
plot.default, 13, 15, 16