Package 'kgen'

March 17, 2023

Type Package

Title A Tool for Calculating Stoichiometric Equilibrium Constants (Ks) for Seawater

Version 0.2.1

License MIT + file LICENSE

Description A unified software package simultaneously implemented in 'Python', 'R', and 'Matlab' providing a uniform and internally-consistent way of calculating stoichiometric equilibrium constants in modern and palaeo seawater as a function of temperature, salinity, pressure and the concentration of magnesium, calcium, sulphate, and fluorine.

Encoding UTF-8

Depends R (>= 4.0)

Imports rjson (>= 0.2.21), reticulate (>= 1.26), rappdirs (>= 0.3.3), checkmate (>= 2.1.0), pbapply (>= 1.7.0), data.table (>= 1.14.6)

Suggests testthat (>= 3.0.0)

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation no

Author Dennis Mayk [aut, cre] (<https://orcid.org/0000-0002-5017-1495>)

Maintainer Dennis Mayk <dm807@cantab.ac.uk>

Repository CRAN

Date/Publication 2023-03-17 17:30:01 UTC

R topics documented:

`calc_fluorine` *Calculate fluorine From Dickson et al., 2007, Table 2 Note: Sal / 1.80655 = Chlorinity*

Description

Calculate fluorine From Dickson et al., 2007, Table 2 Note: Sal / 1.80655 = Chlorinity

Usage

```
calc_fluorine(sal)
```

Arguments

`sal`	Salinity

Value

fluorine

calc_K *Calculate a single equilibrium constant*

Description

Calculate a specified stoichiometric equilibrium constant at given temperature, salinity, pressure and the concentration of magnesium, calcium, sulphate, and fluorine.

Usage

```
calc_K(
  k,
  temp_c = 25,
  sal = 35,
  p_bar = NULL,
  magnesium = 0.0528171,
  calcium = 0.0102821,
  sulphate = NULL,
  fluorine = NULL,
  method = "MyAMI"
)
```

Arguments

k	K to be calculated
temp_c	Temperature (Celsius)
sal	Salinity
p_bar	Pressure (Bar) (optional)
magnesium	Magnesium concentration in mol/kgsw. If None, modern is assumed (0.0528171). Should be the average magnesium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.
calcium	Calcium concentration in mol/kgsw. If None, modern is assumed (0.0102821). Should be the average calcium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.
sulphate	Sulphate concentration in mol/kgsw. Calculated from salinity if not given.
fluorine	Fluorine concentration in mol/kgsw. Calculated from salinity if not given.
method	Options: `R_Polynomial`, `MyAMI_Polynomial` , `MyAMI` (defaults to "MyAMI").

Value

Specified K at the given conditions

Author(s)

Dennis Mayk

`calc_Ks` *Calculate equilibrium constants for seawater*

Description

Wrapper to calculate multiple stoichiometric equilibrium constants at given temperature, salinity, pressure and the concentration of magnesium, calcium, sulphate, and fluorine.

Usage

```
calc_Ks(
  ks = NULL,
  temp_c = 25,
  sal = 35,
  p_bar = NULL,
  magnesium = 0.0528171,
  calcium = 0.0102821,
  sulphate = NULL,
  fluorine = NULL,
  method = "MyAMI"
)
```

Arguments

`ks`	character vectors of Ks to be calculated e.g., c("K0", "K1")
`temp_c`	Temperature (Celsius)
`sal`	Salinity
`p_bar`	Pressure (Bar) (optional)
`magnesium`	Magnesium concentration in mol/kgsw. If None, modern is assumed (0.0528171). Should be the average magnesium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.
`calcium`	Calcium concentration in mol/kgsw. If None, modern is assumed (0.0102821). Should be the average calcium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.
`sulphate`	Sulphate concentration in mol/kgsw. Calculated from salinity if not given.
`fluorine`	Fluorine concentration in mol/kgsw. Calculated from salinity if not given.
`method`	Options: `R_Polynomial`, `MyAMI_Polynomial` , `MyAMI` (defaults to "MyAMI").

Value

Data.table of specified Ks at the given conditions

Author(s)

Dennis Mayk

`calc_pressure_correction` *Calculate pressure correction factor*

Description

Calculate pressure correction factor for a specified equilibrium constant.

Usage

```
calc_pressure_correction(k, temp_c, p_bar)
```

Arguments

`k`	K to be calculated
`temp_c`	Temperature (Celsius)
`p_bar`	Pressure (Bar)

Value

pressure correction factor

Author(s)

Dennis Mayk

`calc_sulphate` *Calculate sulphate From Dickson et al., 2007, Table 2 Note: Sal / 1.80655 = Chlorinity#'*

Description

Calculate sulphate From Dickson et al., 2007, Table 2 Note: Sal / 1.80655 = Chlorinity#'

Usage

```
calc_sulphate(sal)
```

Arguments

`sal`	Salinity

Value

sulphate

fn_Istr — *Ionic strength after Dickson (1990a); see Dickson et al. (2007)*

Description

Ionic strength after Dickson (1990a); see Dickson et al. (2007)

Usage

```
fn_Istr(sal)
```

Arguments

sal	Salinity

Value

Ionic strength

fn_K0 — *Calculate K0*

Description

Calculate K0

Usage

```
fn_K0(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

K0

fn_K1 *Calculate K1*

Description

Calculate K1

Usage

```
fn_K1(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

K1

fn_K2 *Calculate K2*

Description

Calculate K2

Usage

```
fn_K2(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

K2

fn_KB *Calculate KB*

Description

Calculate KB

Usage

```
fn_KB(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KB

fn_KF *Calculate KF*

Description

Calculate KF

Usage

```
fn_KF(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KF

fn_KP1 *Calculate KP1*

Description

Calculate KP1

Usage

```
fn_KP1(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KP1

fn_KP2 *Calculate KP2*

Description

Calculate KP2

Usage

```
fn_KP2(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KP2

fn_KP3 *Calculate KP3*

Description

Calculate KP3

Usage

```
fn_KP3(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KP3

fn_KS *Calculate KS*

Description

Calculate KS

Usage

```
fn_KS(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KS

fn_KSi *Calculate KSi*

Description

Calculate KSi

Usage

```
fn_KSi(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

KSi

fn_Ksp *Calculate Ksp*

Description

Calculate Ksp

Usage

```
fn_Ksp(p, temp_k, sal)
```

Arguments

p	Parameters for K calculation
temp_k	Temperature (Kelvin)
sal	Salinity

Value

Ksp

`fn_KW` *Calculate KW*

Description

Calculate KW

Usage

```
fn_KW(p, temp_k, sal)
```

Arguments

`p`	Parameters for K calculation
`temp_k`	Temperature (Kelvin)
`sal`	Salinity

Value

KW

`fn_pc` *Calculate pressure correction factor for Ks From Millero et al. (2007, doi:10.1021/cr0503557) Eqns 38-40*

Description

Calculate pressure correction factor for Ks From Millero et al. (2007, doi:10.1021/cr0503557) Eqns 38-40

Usage

```
fn_pc(p, temp_c, p_bar)
```

Arguments

`p`	Parameters for K calculation
`temp_c`	Temperature (Celsius)
`p_bar`	Pressure

Value

Pressure correction factor

install_pymyami *Install MyAMI from pypi*

Description

Install MyAMI from pypi

Usage

```
install_pymyami()
```

is_linux *Check if OS is Linux*

Description

Check if OS is Linux

Usage

```
is_linux()
```

is_osx *Check if OS is OSX*

Description

Check if OS is OSX

Usage

```
is_osx()
```

is_windows *Check if OS is Windows*

Description

Check if OS is Windows

Usage

```
is_windows()
```

kgen_poly *Kgen R polynomial function*

Description

Kgen R polynomial function

Usage

```
kgen_poly(sal, temp_k, magnesium = 0.0528171, calcium = 0.0102821)
```

Arguments

sal	Salinity (PSU)
temp_k	Temperature (Kelvin)
magnesium	magnesium concentration in mol/kgsw. If None, modern is assumed (0.0528171). Should be the average magnesium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.
calcium	calcium concentration in mol/kgsw. If None, modern is assumed (0.0102821). Should be the average calcium concentration in seawater - a salinity correction is then applied to calculate the magnesium concentration in the sample.

K_fns *List of all functions*

Description

List of all functions

Usage

```
K_fns
```

Format

An object of class `list` of length 13.

mc_exists *Check if miniconda exists*

Description

Check if miniconda exists

Usage

```
mc_exists(path = miniconda_path())
```

Arguments

path	Path to miniconda

miniconda_conda *Check if miniconda is installed*

Description

Check if miniconda is installed

Usage

```
miniconda_conda(path = miniconda_path())
```

Arguments

path	Path to miniconda

miniconda_path *Get miniconda path*

Description

Get miniconda path

Usage

```
miniconda_path()
```

`miniconda_path_default` *Get miniconda default path*

Description

Get miniconda default path

Usage

```
miniconda_path_default()
```

`pymyami_exists` *Check if pymyami is installed*

Description

Check if pymyami is installed

Usage

```
pymyami_exists()
```

Index