Package ‘stoichcalc’

February 20, 2015

**Type** Package

**Version** 1.1-3

**Title** R Functions for Solving Stoichiometric Equations

**Date** 2013-02-06

**Author** Peter Reichert

**Maintainer** Peter Reichert <peter.reichert@eawag.ch>

**Description** Given a list of substance compositions, a list of substances involved in a process, and a list of constraints in addition to mass conservation of elementary constituents, the package contains functions to build the substance composition matrix, to analyze the uniqueness of process stoichiometry, and to calculate stoichiometric coefficients if process stoichiometry is unique. (See Reichert, P. and Schuwirth, N., A generic framework for deriving process stoichiometry in environmental models, Environmental Modelling and Software 25, 1241-1251, 2010 for more details.)

**License** GPL (>= 2)

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2013-02-06 08:22:10

**NeedsCompilation** no

**R topics documented:**

- stoeichcalc-package .......................................................... 2
- calc.comp.matrix ............................................................... 4
- calc.stoich.basis .............................................................. 6
- calc.stoich.coef .............................................................. 8

**Index** 12
Description

Given a list of substance compositions, a list of substances involved in a process, and a list of constraints in addition to mass conservation of elementary constituents, the package contains functions to build the substance composition matrix, to analyze the uniqueness of process stoichiometry, and to calculate stoichiometric coefficients if process stoichiometry is unique (see reference given below for more details).

Details

Package: stoichcalc
Type: Package
Version: 1.1-3
Date: 2013-02-06
License: GPL >= 2
LazyLoad: yes

The package contains the following three functions:

- `calc.comp.matrix` constructs the substance composition matrix from a list of substance composition vectors,
- `calc.stoich.basis` calculates the basis of the stoichiometry space that is compatible with mass balances of elementary constituents and additional constraints, `calc.stoich.coef` calculates the stoichiometric coefficients of a process from involved substances, their composition and constraints.

Author(s)

Peter Reichert <peter.reichert@eawag.ch>

References


See Also

`calc.comp.matrix, calc.stoich.basis, calc.stoich.coef`

Examples

```r
subst.comp <-
list(NH4 = c(H = 4*1/14, # gH/gNH4-N
      N = 1, # gN/gNH4-N
      charge = 1/14), # chu/gNH4-N
```
N03 = c(0 = 3*16/14, # g0/gNO3-N
N = 1, # gn/gNO3-N
charge = -1/14), # chu/gNO3-N
HP04 = c(0 = 4*16/31, # g0/gHP04-P
H = 1*1/31, # gh/gHP04-P
P = 1, # gp/gHP04-P
charge = -2/31), # chu/gHP04-P
HC03 = c(C = 1, # gC/gHC03-C
O = 3*16/12, # gO/gHC03-C
H = 1*1/12, # gh/gHC03-C
charge = -1/12), # chu/gHC03-C
O2 = c(0 = 1), # go/go2-O
H = c(H = 1, # gh/molH
charge = 1), # chu/molH
H2O = c(0 = 1*1/12, # go/molH2O
H = 2*1/12, # gh/molH2O
ALG = c(N = 0.06, # gn/gALG
P = 0.005, # gp/gALG
O = 0.50, # go/gALG
H = 0.07, # gh/gALG
C = 0.365), # gc/gALG
ZOO = c(N = 0.06, # gn/gZOO
P = 0.01, # gp/gZOO
O = 0.50, # go/gZOO
H = 0.07, # gh/gZOO
C = 0.36), # gc/gZOO
POM = c(N = 0.04, # gn/gPOM
P = 0.007, # gp/gPOM
O = 0.40, # go/gPOM
H = 0.07, # gh/gPOM
C = 0.483), # gc/gPOM
DOM = c(N = 0.04, # gn/gDOM
P = 0.007, # gp/gDOM
O = 0.40, # go/gDOM
H = 0.07, # gh/gDOM
C = 0.483)), # gc/gDOM

Y.ZOO <- 0.2; f.POM <- 0.2; f.DOM <- 0.1
alpha <- calc.comp.matrix(subst.comp)
subst.gro.ALG.NO3 <- c("NO3","HP04","HC03",
"O2","H","H2O","ALG")
basis.gro.ALG.NO3 <-
calc.stoich.basis(alpha,subst.gro.ALG.NO3)
u1.gro.ALG.NO3 <-
calc.stoich.coef(alpha = alpha,
name = "gro.ALG.NO3",
subst = subst.gro.ALG.NO3,
subst.norm = "ALG",
u1.norm = 1)
calc.comp.matrix

Construct Composition Matrix

Description

Construct substance composition matrix from list of substance composition vectors

Usage

calc.comp.matrix(subst.comp, verbose=TRUE)

Arguments

subst.comp Named list of named composition vectors. The list must contain entries labelled by the substance names containing vectors of the mass fractions of elementary constituents (typically chemical elements, charge or COD resp. ThOD) that characterize the composition of the substance. Each element of these vectors must be labelled by the name of the corresponding elementary constituent.

verbose indicator for whether or not to write basic information to the console.

Details

This function compiles the substance composition matrix used in the other functions of the stoichcalc package. It can alternatively be composed manually or by a user-defined function. The main advantage of the use of this function is that substance compositions can be maintained in lists. This makes it much easier to remove and add substances and elementary constituents.
calc.comp.matrix

Value

Composition matrix of all substances (labelled columns) and mass fractions of elementary constituents (labelled rows).

Author(s)

Peter Reichert <peter.reichert@eawag.ch>

References


See Also

calc.stoich.basis, calc.stoich.coef

Examples

```r
subst.comp <-
  list(NH4 = c(H = 4*1/14, # gH/gNH4-N
         N = 1,     # gN/gNH4-N
         charge = 1/14), # chu/gNH4-N
       NO3 = c(O = 3*16/14, # go/gNO3-N
                N = 1,     # gN/gNO3-N
                charge = -1/14), # chu/gNO3-N
       HP04 = c(O = 4*16/31, # go/gHP04-P
                 H = 1*1/12, # gh/gHP04-P
                 P = 1,      # gp/gHP04-P
                 charge = -2/31), # chu/gHP04-P
       HC03 = c(C = 1,   # gc/gHC03-C
                 O = 3*16/12, # go/gHC03-C
                 H = 1*1/12,  # gh/gHC03-C
                 charge = -1/12), # chu/gHC03-C
       O2 = c(O = 1),    # go/go2-O
       H = c(H = 1,      # gh/molH
              charge = 1), # chu/molH
       H20 = c(O = 1*12,  # go/molH20
                H = 2*1),    # gh/molH20
       ALG = c(N = 0.06,  # gN/gALG
                P = 0.005,   # gP/gALG
                O = 0.50,    # gO/gALG
                H = 0.07,    # gh/gALG
                C = 0.365),  # gc/gALG
       ZOO = c(N = 0.06,  # gN/gZOO
                P = 0.01,    # gP/gZOO
                O = 0.50,    # gO/gZOO
                H = 0.07,    # gh/gZOO
                C = 0.36),   # gc/gZOO
       POM = c(N = 0.04,  # gN/gPOM
                P = 0.007,   # gP/gPOM
                O = 0.40,    # go/gPOM
```
**Description**

Calculate the basis of the stoichiometry space that is compatible with mass balances of elementary constituents and additional constraints.

**Usage**

```r
calc.stoich.basis(alpha, subst = NA, constraints = list(), eps = 1e-5, verbose = TRUE)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>Substance composition matrix of all substances (labelled columns) and mass fractions of elementary constituents (labelled rows). Typically calculated by the function <code>calc.comp.matrix</code>.</td>
</tr>
<tr>
<td>subst</td>
<td>Character vector of names of substances to be used for analysis (this must be a subset of the column names of alpha).</td>
</tr>
<tr>
<td>constraints</td>
<td>List of stoichiometric constraints in addition to mass conservation of elementary constituents. Each stoichiometric constraint must be stored as a vector containing the coefficients of the linear equation in elementary constituents that defines the constraint. The elements of this vector must be labelled by the names of the corresponding elementary constituents.</td>
</tr>
<tr>
<td>eps</td>
<td>Relative tolerance for checking ratios of stoichiometric coefficients (only used for informing user about substance pairs with fixed stoichiometric ratio).</td>
</tr>
<tr>
<td>verbose</td>
<td>Indicator for whether or not to write basic information to the console.</td>
</tr>
</tbody>
</table>

**Details**

This function is primarily used in the function `calc.stoich.coef`. However, it can also be used to check the number of required stoichiometric constraints in addition to mass conservation of elementary constituents for a given process. In this case the composition matrix should only contain the substances relevant for this process. The number of required constraints is then equal to the row dimension of the output matrix minus 1.
Value

Matrix of basis vectors (in rows) that span the compatible stoichiometric space.

Author(s)

Peter Reichert <peter.reichert@eawag.ch>

References


See Also

calc.comp.matrix, calc.stoich.coef

Examples

```r
subst.comp <-
list(NH4 = c(H = 4*1/14, # gH/gNH4-N
          N = 1, # gN/gNH4-N
          charge = 1/14), # chu/gNH4-N
     NO3 = c(O = 3*16/14, # gO/gNO3-N
          N = 1, # gN/gNO3-N
          charge = -1/14), # chu/gNO3-N
     HP04 = c(O = 4*16/31, # gO/gHP04-P
          H = 1*1/31, # gH/gHP04-P
          P = 1, # gP/gHP04-P
          charge = -2/31), # chu/gHP04-P
     HC03 = c(C = 1, # gc/gHC03-C
          O = 3*16/12, # gO/gHC03-C
          H = 1*1/12, # gH/gHC03-C
          charge = -1/12), # chu/gHC03-C
     O2  = c(O = 1), # gO/gO2-0
     H   = c(H = 1, # gH/molH
              charge = 1), # chu/molH
     H20 = c(O = 1*12, # gO/molH20
              H = 2*1), # gH/molH20
     ALG = c(N = 0.06, # gN/gALG
              P = 0.005, # gP/gALG
              O = 0.50, # gO/gALG
              H = 0.07, # gH/gALG
              C = 0.365), # gc/gALG
     ZOO = c(N = 0.06, # gN/gZOO
              P = 0.01, # gP/gZOO
              O = 0.50, # gO/gZOO
              H = 0.07, # gH/gZOO
              C = 0.36), # gc/gZOO
     POM = c(N = 0.04, # gN/gPOM
              P = 0.007, # gP/gPOM
              O = 0.40, # gO/gPOM
              H = 0.07, # gH/gPOM
```
**Calculation of Stoichiometric Coefficients**

**Description**

Calculate stoichiometric coefficients of a process from involved substances, their composition and constraints.

**Usage**

```r
calc.stoich.coef(alpha, name, subst, subst.norm, nu.norm = 1, constraints = list(), eps = 1e-5, verbose = TRUE)
```

**Arguments**

- `alpha` Substance composition matrix of all substances (labelled columns) and mass fractions of elementary constituents (labelled rows). Typically calculated by the function `calc.comp.matrix`.
- `name` Name of the process

---

```r
c = 0.483,  # gC/gPOM
dom = c(N = 0.04,  # gN/gDOM
P = 0.007,  # gP/gDOM
O = 0.40,  # gO/gDOM
H = 0.07,  # gH/gDOM
C = 0.483))  # gC/gDOM

Y.ZOO <- 0.2; f.POM <- 0.2; f.DOM <- 0.1
alpha <- calc.comp.matrix(subst.comp)
subst.gro.ALG.NO3 <- c("NO3", "HP04", "HC03", "O2", "H", "H2O", "ALG")
basis.gro.ALG.NO3 <- calc.stoich.basis(alpha, subst.gro.ALG.NO3)
subst.gro.ZOO <- c("NH4", "HP04", "HC03", "O2", "H", "H2O", "ALG", "ZOO", "POM", "DOM")
basis.gro.ZOO <- calc.stoich.basis(alpha, subst.gro.ZOO)
const.gro.ZOO <- list(c("ZOO" = 1,"ALG" = Y.ZOO), c("POM" = 1,"ALG" = f.POM), c("DOM" = 1,"ALG" = f.DOM))
basis.gro.ZOO <- calc.stoich.basis(alpha, subst.gro.ZOO, const.gro.ZOO)
```
**calc.stoich.coef**

The function `calc.stoich.coef` is the key function of the package for the calculation of stoichiometric coefficients of individual processes. The results for different processes can easily be bound to the comprehensive stoichiometric matrix of all processes by using `rbind`.

### Value

Matrix consisting of one row of stoichiometric coefficients of the process or an error message if the process stoichiometry is not uniquely defined. The row name of the matrix is equal to the process name specified as an argument (to allow binding the stoichiometries of several processes to a comprehensive stoichiometric matrix), the column names are equal to the substance names provided by the substance composition matrix `alpha`.

### Author(s)

Peter Reichert &lt;peter.reichert@eawag.ch&gt;

### References


### See Also

calc.comp.matrix, calc.stoich.basis

### Examples

```r
subst.comp <-
  list(NH4 = c(H = 4*1/14, N = 1, charge = 1/14), NO3 = c(O = 3*16/14))
```

calc.stoich.coef

N = 1,  # gN/gNO3-N
charge = -1/14),  # chu/gNO3-N

HP04 = c(O = 4*16/31,  # go/gHP04-P
         H = 1*1/31,  # gh/gHP04-P
         P = 1,  # gP/gHP04-P
         charge = -2/31),  # chu/gHP04-P

HC03 = c(C = 1,  # gc/gHC03-C
         O = 3*16/12,  # go/gHC03-C
         H = 1*1/12,  # gh/gHC03-C
         charge = -1/12),  # chu/gHC03-C

O2  = c(O = 1),  # go/go2-O
H   = c(H = 1),  # gh/mo1H
         charge = 1),  # chu/mo1H

H2O = c(O = 1*12,  # go/mo1H2O
         H = 2*1),  # gh/mo1H2O

ALG = c(N = 0.06,  # gn/gALG
         P = 0.005,  # gp/gALG
         O = 0.50,  # go/gALG
         H = 0.07,  # gh/gALG
         C = 0.365),  # gc/gALG

ZOO = c(N = 0.06,  # gn/gZOO
         P = 0.01,  # gp/gZOO
         O = 0.50,  # go/gZOO
         H = 0.07,  # gh/gZOO
         C = 0.36),  # gc/gZOO

POM = c(N = 0.04,  # gn/gPOM
         P = 0.007,  # gp/gPOM
         O = 0.40,  # go/gPOM
         H = 0.07,  # gh/gPOM
         C = 0.483),  # gc/gPOM

DOM = c(N = 0.04,  # gn/gDOM
         P = 0.007,  # gp/gDOM
         O = 0.40,  # go/gDOM
         H = 0.07,  # gh/gDOM
         C = 0.483))  # gc/gDOM

Y.ZOO <- 0.2; f.POM <- 0.2; f.DOM <- 0.1

alpha <- calc.comp.matrix(subst.comp)

subst.gro.ALG.NO3 <- c("NO3","HP04","HC03","O2","H","H2O","ALG")

basis.gro.ALG.NO3 <-
calc.stoich.basis(alpha,subst.gro.ALG.NO3)
	nu.gro.ALG.NO3 <-
calc.stoich.coef(alpha = alpha,
name = "gro.ALG.NO3",
subst = subst.gro.ALG.NO3,
subst.norm = "ALG",
u.norm = 1)
calc.stoich.coef

```r
subst.gro.ZOO <- c("NH4","HPO4","HC03","O2","H",
                   "H2O","ALG","ZOO","POM","DOM")

basis.gro.ZOO <-
calc.stoich.basis(alpha,subst.gro.ZOO)

const.gro.ZOO <- list(c("ZOO" = 1,"ALG" = Y.ZOO),
c("POM" = 1,"ALG" = f.POM),
c("DOM" = 1,"ALG" = f.DOM))

nu.gro.ZOO <-
calc.stoich.coef(alpha = alpha,
                 name = "gro.ZOO",
                 subst = subst.gro.ZOO,
                 subst.norm = "ZOO",
                 nu.norm = 1,
                 constraints = const.gro.ZOO)

nu <- rbind(nu.gro.ALG.NO3,
             nu.gro.ZOO)

print(nu,digits=2)
```
# Index

*Topic process stoichiometry
- calc.comp.matrix, 4
- calc.stoich.basis, 6
- calc.stoich.coef, 8
- stoichcalc-package, 2

- calc.comp.matrix, 2, 4, 6–9
- calc.stoich.basis, 2, 5, 6, 9
- calc.stoich.coef, 2, 5–7, 8

- stoichcalc (stoichcalc-package), 2
- stoichcalc-package, 2