

# Package ‘arkhe’

December 18, 2019

**Title** Representation of Archaeological Data

**Version** 0.2.0

**Maintainer** Nicolas Frerebeau

<nicolas.frerebeau@u-bordeaux-montaigne.fr>

**Description** A collection of classes that represent archaeological data. This package provides a set of S4 classes that extend the basic matrix data type (absolute/relative frequency, presence/absence data, co-occurrence matrix, etc.) upon which package developers can build subclasses. It also provides a set of generic methods (mutators and coercion mechanisms) and functions (e.g. predicates). In addition, a few classes of general interest (e.g. that represent stratigraphic relationships) are implemented.

**License** GPL-3

**URL** <http://arkhe.archaeo.science>, <http://github.com/nfrerebeau/arkhe>

**BugReports** <http://github.com/nfrerebeau/arkhe/issues>

**Depends** R (>= 3.2)

**Imports** methods

**Suggests** covr, knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Collate** 'AllClasses.R' 'AllGenerics.R' 'arkhe-package.R'  
'predicates.R' 'check.R' 'chronology.R' 'coerce.R'  
'conditions.R' 'deprecate.R' 'extract.R' 'geography.R'  
'initialize.R' 'show.R' 'utilities.R' 'validate.R'

**NeedsCompilation** no

**Author** Nicolas Frerebeau [aut, cre] (<<https://orcid.org/0000-0001-5759-4944>>)

**Repository** CRAN

**Date/Publication** 2019-12-18 14:50:10 UTC

## R topics documented:

AbundanceMatrix-class . . . . .	2
chronology . . . . .	4
coerce . . . . .	6
CountMatrix-class . . . . .	8
geography . . . . .	10
IncidenceMatrix-class . . . . .	12
mutator . . . . .	14
OccurrenceMatrix-class . . . . .	15
SimilarityMatrix-class . . . . .	17
StratigraphicMatrix-class . . . . .	19
subset . . . . .	21
<b>Index</b>	<b>23</b>

---

AbundanceMatrix-class *Relative Frequency Matrix*

---

### Description

An S4 class to represent a relative frequency matrix (i.e. the fraction of times a given datum occurs in a dataset).

### Slots

totals A `numeric` vector.

### Matrix ID

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

### Get and set

In the code snippets below, `x` is a `*Matrix` object.

`get_id(x)` Get the ID of `x`.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of `x`.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of `x`.

**Access**

In the code snippets below, `x` is a `*Matrix` object.

`dim(x)` Returns the dimension of `x`.

`nrow(x)` Returns the number of rows present in `x`.

`ncol(x)` Returns the number of columns present in `x`.

`dimnames(x)`, `dimnames(x) <- value` Retrieves or sets the row `dimnames` of `x` according to `value`.

`rownames(x)`, `rownames(x) <- value` Retrieves or sets the row names of `x` according to `value`.

`colnames(x)`, `colnames(x) <- value` Retrieves or sets the column names of `x` according to `value`.

**Subset**

In the code snippets below, `x` is a `*Matrix` object.

`x[i, j]` Extracts elements selected by subscripts `i` and `j`. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or `NULL`. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as `x`.

`x[[i]]` Extracts a single element selected by subscript `i`.

**Author(s)**

N. Frerebeau

**See Also**

[NumericMatrix](#)

Other matrix: [CountMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [SimilarityMatrix-class](#), [StratigraphicMatrix-class](#), [coerce](#)

**Examples**

```
## Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)

## Access
get_id(A1)
dim(A1) # Get the matrix dimensions
colnames(A1) <- letters[11:20] # Set the column names
colnames(A1) # Get the column names
rownames(A1) <- LETTERS[1:10] # Set the row names
rownames(A1) # Get the rownames

## Subset
A1[[1]] # Get the first value
A1[, ] # Get all values
A1[1, ] # Get the first row
```

```

A1[c("A", "B", "C"), ] # Get the first three rows
A1[c("A", "B", "C"), 1] # Get the first three rows of the first column
A1[, 1, drop = FALSE] # Get the first column

## Coerce counts to relative frequencies
B <- as_abundance(A1)
## Row sums are internally stored before coercing to a frequency matrix
get_totals(B) # Get row sums
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)
## Coerce to a co-occurrence matrix
B <- as_occurrence(A1)

```

---

chronology

*Chronological Information*


---

## Description

`set_dates` and `get_dates` allow to get or set the dates of an object.

## Usage

```

get_dates(object)

set_dates(object) <- value

## S4 method for signature 'Matrix'
get_dates(object)

## S4 replacement method for signature 'Matrix'
set_dates(object) <- value

```

## Arguments

<code>object</code>	A $m \times p$ matrix of count data (typically of class <a href="#">Matrix</a> ).
<code>value</code>	A possible value for the element(s) of object (see below).

## Details

An attempt is made to interpret the argument value in a suitable way. *Note* that errors are assumed to be given at 1 sigma.

If value is a:

character **vector** it is assumed to contain Roman numerals.

numeric **or** integer vector these values are assumed to represent dates without known errors.

list containing components "value" and "error", these are used to define dates and corresponding errors.

matrix or data.frame **with two or more columns** the first is assumed to contain the dates and the second the error values. *Note* that if value has columns named "value" and "error", these columns will be used.

## Value

get\_dates returns a two-columns [data.frame](#).

## Author(s)

N. Frerebeau

## Examples

```
## Create a count data matrix
A <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10,
                 dimnames = list(LETTERS[1:10], NULL))

## Set dates as a matrix
B <- matrix(data = sample(0:10, 20, TRUE), nrow = 10, ncol = 2)
set_dates(A) <- B
get_dates(A)

## Set dates as a data.frame
B <- data.frame(sample(0:10, 5, TRUE), sample(0:10, 5, TRUE),
               row.names = LETTERS[seq(1, 10, 2)])
set_dates(A) <- B
get_dates(A)

## Set dates as a list
B <- list(value = sample(0:10, 10, TRUE), error = sample(0:10, 10, TRUE))
set_dates(A) <- B
get_dates(A)

## Set dates as a numeric vector
B <- sample(0:10, 10, TRUE)
suppressWarnings(set_dates(A) <- B)
get_dates(A)

## Set dates as a character vector
B <- c("X", "IX", "VIII", "VII", "VI", "V", "IV", "III", "II", "I")
set_dates(A) <- B
get_dates(A)

## Unset dates
set_dates(A) <- NULL
get_dates(A)
```

---

coerce

*Coerce*

---

## Description

Coerce

## Usage

```
as_count(from)
```

```
as_abundance(from)
```

```
as_incidence(from)
```

```
as_occurrence(from)
```

```
as_similarity(from)
```

```
as_features(from)
```

```
as_stratigraphy(from)
```

```
## S4 method for signature 'ANY'  
as_count(from)
```

```
## S4 method for signature 'ANY'  
as_abundance(from)
```

```
## S4 method for signature 'ANY'  
as_incidence(from)
```

```
## S4 method for signature 'ANY'  
as_occurrence(from)
```

```
## S4 method for signature 'ANY'  
as_similarity(from)
```

```
## S4 method for signature 'ANY'  
as_stratigraphy(from)
```

```
## S4 method for signature 'Matrix'  
as_features(from)
```

## Arguments

from            A numeric [matrix](#) or [data.frame](#) to be coerced.

## Details

The following methods coerce a `matrix` or `data.frame` to a `*Matrix` object:

Method	Target	Details
<code>as_count</code>	<a href="#">CountMatrix</a>	absolute frequency data
<code>as_abundance</code>	<a href="#">AbundanceMatrix</a>	relative frequency data
<code>as_incidence</code>	<a href="#">IncidenceMatrix</a>	presence/absence data
<code>as_occurrence</code>	<a href="#">OccurrenceMatrix</a>	co-occurrence
<code>as_similarity</code>	<a href="#">SimilarityMatrix</a>	(dis)similarity
<code>as_stratigraphy</code>	<a href="#">StratigraphicMatrix</a>	stratigraphic relationships

`as_features` converts an [Matrix](#) object to a collection of features (i.e. `adata.frame` with dates and coordinates columns).

`as_stratigraphy` converts a set of stratigraphic relationships (edges) to a stratigraphic (adjacency) matrix. `from` can be a `matrix`, `list`, or `data.frame`: the first column/component is assumed to contain the bottom units and the second the top units.

## Value

A coerced object.

## Author(s)

N. Frerebeau

## See Also

Other matrix: [AbundanceMatrix-class](#), [CountMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [SimilarityMatrix-class](#), [StratigraphicMatrix-class](#)

## Examples

```
## Create a count matrix
A0 <- matrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)

## Coerce to absolute frequencies
A1 <- as_count(A0)

## Coerce to relative frequencies
B <- as_abundance(A1)

## Row sums are internally stored before coercing to a frequency matrix
## (use totals() to get these values)
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)

## Coerce to presence/absence
C <- as_incidence(A1)
```

```
## Coerce to a co-occurrence matrix
D <- as_occurrence(A1)

## Collection of features
set_dates(A1) <- matrix(sample(0:10, 20, TRUE), nrow = 10, ncol = 2)
set_coordinates(A1) <- matrix(sample(0:10, 30, TRUE), nrow = 10, ncol = 3)
as_features(A1)
```

---

CountMatrix-class      *Absolute Frequency Matrix*

---

### Description

An S4 class to represent an absolute frequency matrix (i.e. the number of times a given datum occurs in a dataset).

### Usage

```
CountMatrix(data = 0, nrow = 1, ncol = 1, byrow = FALSE,
             dimnames = NULL, ...)
```

### Arguments

data	an optional data vector (including a list or <a href="#">expression</a> vector). Non-atomic classed R objects are coerced by <a href="#">as.vector</a> and all attributes discarded.
nrow	the desired number of rows.
ncol	the desired number of columns.
byrow	logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.
dimnames	A <a href="#">dimnames</a> attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.
...	additional arguments to be passed to or from methods.

### Matrix ID

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

**Get and set**

In the code snippets below, `x` is a `*Matrix` object.

`get_id(x)` Get the ID of `x`.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of `x`.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of `x`.

**Access**

In the code snippets below, `x` is a `*Matrix` object.

`dim(x)` Returns the dimension of `x`.

`nrow(x)` Returns the number of rows present in `x`.

`ncol(x)` Returns the number of columns present in `x`.

`dimnames(x)`, `dimnames(x) <- value` Retrieves or sets the row `dimnames` of `x` according to `value`.

`rownames(x)`, `rownames(x) <- value` Retrieves or sets the row names of `x` according to `value`.

`colnames(x)`, `colnames(x) <- value` Retrieves or sets the column names of `x` according to `value`.

**Subset**

In the code snippets below, `x` is a `*Matrix` object.

`x[i, j]` Extracts elements selected by subscripts `i` and `j`. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or `NULL`. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as `x`.

`x[[i]]` Extracts a single element selected by subscript `i`.

**Author(s)**

N. Frerebeau

**See Also**

[NumericMatrix](#)

Other matrix: [AbundanceMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [SimilarityMatrix-class](#), [StratigraphicMatrix-class](#), [coerce](#)

**Examples**

```

## Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)

## Access
get_id(A1)
dim(A1) # Get the matrix dimensions
colnames(A1) <- letters[11:20] # Set the column names
colnames(A1) # Get the column names
rownames(A1) <- LETTERS[1:10] # Set the row names
rownames(A1) # Get the rownames

## Subset
A1[[1]] # Get the first value
A1[, ] # Get all values
A1[1, ] # Get the first row
A1[c("A", "B", "C"), ] # Get the first three rows
A1[c("A", "B", "C"), 1] # Get the first three rows of the first column
A1[, 1, drop = FALSE] # Get the first column

## Coerce counts to relative frequencies
B <- as_abundance(A1)
## Row sums are internally stored before coercing to a frequency matrix
get_totals(B) # Get row sums
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)
## Coerce to a co-occurrence matrix
B <- as_occurrence(A1)

```

---

geography

*Spatial Information*


---

**Description**

Deal with spatial information.

**Usage**

```

get_coordinates(object)

set_coordinates(object) <- value

## S4 method for signature 'Matrix'
get_coordinates(object)

## S4 replacement method for signature 'Matrix'
set_coordinates(object) <- value

```

**Arguments**

object            An object from which to get or set element(s).  
 value            A possible value for the element(s) of object (see below).

**Details**

An attempt is made to interpret the argument value in a way suitable for geographic coordinates.

If value is a:

list containing components "x", "y" and "z", these are used to define coordinates (longitude, latitude and elevation, respectively). If "z" is missing, the vertical coordinates will be ignored (and NA will be generated).

matrix **or** data.frame **with two or more columns** the first is assumed to contain the x values, the second the y and the third the z values. *Note* that if value has columns named "x", "y" and "z", these columns will be used. If value has only two columns or has columns named "x" and "y" but not "z", the vertical coordinates will be ignored (and NA will be generated).

**Note**

EXPERIMENTAL: subject to major changes in a future release.

**Author(s)**

N. Frerebeau

**Examples**

```
## Create a count data matrix
A <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10,
                 dimnames = list(LETTERS[1:10], NULL))

## Set geographic coordinates as a matrix
B <- matrix(data = sample(0:10, 30, TRUE), nrow = 10, ncol = 3)
set_coordinates(A) <- B
get_coordinates(A)

## Set geographic coordinates as a data.frame
B <- data.frame(sample(0:10, 5, TRUE), sample(0:10, 5, TRUE), sample(0:10, 5, TRUE),
               row.names = LETTERS[seq(1, 10, 2)])
set_coordinates(A) <- B
get_coordinates(A)

## Set geographic coordinates as a list
B <- list(X = sample(0:10, 10, TRUE), Y = sample(0:10, 10, TRUE))
set_coordinates(A) <- B
get_coordinates(A)

## Unset geographic coordinates
set_coordinates(A) <- NULL
get_coordinates(A)
```

---

 IncidenceMatrix-class *Incidence Matrix*


---

**Description**

An S4 class to represent an incidence (presence/absence) matrix.

**Usage**

```
IncidenceMatrix(data = FALSE, nrow = 1, ncol = 1, byrow = FALSE,
  dimnames = NULL, ...)
```

**Arguments**

<code>data</code>	an optional data vector (including a list or <a href="#">expression</a> vector). Non-atomic classed R objects are coerced by <a href="#">as.vector</a> and all attributes discarded.
<code>nrow</code>	the desired number of rows.
<code>ncol</code>	the desired number of columns.
<code>byrow</code>	logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.
<code>dimnames</code>	A <a href="#">dimnames</a> attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.
<code>...</code>	additional arguments to be passed to or from methods.

**Matrix ID**

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

**Get and set**

In the code snippets below, `x` is a `*Matrix` object.

`get_id(x)` Get the ID of `x`.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of `x`.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of `x`.

**Access**

In the code snippets below, `x` is a `*Matrix` object.

`dim(x)` Returns the dimension of `x`.

`nrow(x)` Returns the number of rows present in `x`.

`ncol(x)` Returns the number of columns present in `x`.

`dimnames(x)`, `dimnames(x) <- value` Retrieves or sets the row `dimnames` of `x` according to `value`.

`rownames(x)`, `rownames(x) <- value` Retrieves or sets the row names of `x` according to `value`.

`colnames(x)`, `colnames(x) <- value` Retrieves or sets the column names of `x` according to `value`.

**Subset**

In the code snippets below, `x` is a `*Matrix` object.

`x[i, j]` Extracts elements selected by subscripts `i` and `j`. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or `NULL`. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as `x`.

`x[[i]]` Extracts a single element selected by subscript `i`.

**Author(s)**

N. Frerebeau

**See Also**

[LogicalMatrix](#)

Other matrix: [AbundanceMatrix-class](#), [CountMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [SimilarityMatrix-class](#), [StratigraphicMatrix-class](#), [coerce](#)

**Examples**

```
## Create an incidence (presence/absence) matrix
## Data will be coerced with as.logical()
A <- IncidenceMatrix(data = sample(0:1, 100, TRUE, c(1, 1/3)), nrow = 10)

## Create a count data matrix
B <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10)
## Coerce to presence/absence
C <- as_incidence(B)
```

---

mutator

*Get or Set Parts of an Object*

---

## Description

Getters and setters to extract or replace parts of an object.

## Usage

```
get_id(object)

get_method(object)

get_units(object)

get_totals(object)

set_totals(object) <- value

## S4 method for signature 'ANY'
get_id(object)

## S4 method for signature 'AbundanceMatrix'
get_totals(object)

## S4 method for signature 'SimilarityMatrix'
get_method(object)

## S4 method for signature 'StratigraphicMatrix'
get_units(object)

## S4 replacement method for signature 'AbundanceMatrix'
set_totals(object) <- value
```

## Arguments

object	An object from which to get or set element(s).
value	A possible value for the element(s) of object (see below).

## Value

An object of the same sort as object with the new values assigned.

## Author(s)

N. Frerebeau

**See Also**

Other mutator: [subset](#)

**Examples**

```
## Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)

## Access
get_id(A1)
dim(A1) # Get the matrix dimensions
colnames(A1) <- letters[11:20] # Set the column names
colnames(A1) # Get the column names
rownames(A1) <- LETTERS[1:10] # Set the row names
rownames(A1) # Get the rownames

## Subset
A1[[1]] # Get the first value
A1[, ] # Get all values
A1[1, ] # Get the first row
A1[c("A", "B", "C"), ] # Get the first three rows
A1[c("A", "B", "C"), 1] # Get the first three rows of the first column
A1[, 1, drop = FALSE] # Get the first column

## Coerce counts to relative frequencies
B <- as_abundance(A1)
## Row sums are internally stored before coercing to a frequency matrix
get_totals(B) # Get row sums
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)
## Coerce to a co-occurrence matrix
B <- as_occurrence(A1)
```

---

OccurrenceMatrix-class

*Co-Occurrence Matrix*

---

**Description**

An S4 class to represent a co-occurrence matrix.

**Details**

A co-occurrence matrix is a symmetric matrix with zeros on its main diagonal, which works out how many times (expressed in percent) each pairs of taxa/types occur together in at least one sample.

**Matrix ID**

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

**Get and set**

In the code snippets below, `x` is a `*Matrix` object.

`get_id(x)` Get the ID of `x`.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of `x`.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of `x`.

**Access**

In the code snippets below, `x` is a `*Matrix` object.

`dim(x)` Returns the dimension of `x`.

`nrow(x)` Returns the number of rows present in `x`.

`ncol(x)` Returns the number of columns present in `x`.

`dimnames(x)`, `dimnames(x) <- value` Retrieves or sets the row dimnames of `x` according to `value`.

`rownames(x)`, `rownames(x) <- value` Retrieves or sets the row names of `x` according to `value`.

`colnames(x)`, `colnames(x) <- value` Retrieves or sets the column names of `x` according to `value`.

**Subset**

In the code snippets below, `x` is a `*Matrix` object.

`x[i, j]` Extracts elements selected by subscripts `i` and `j`. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or `NULL`. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as `x`.

`x[[i]]` Extracts a single element selected by subscript `i`.

**Author(s)**

N. Frerebeau

**See Also**

[NumericMatrix](#)

Other matrix: [AbundanceMatrix-class](#), [CountMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [SimilarityMatrix-class](#), [StratigraphicMatrix-class](#), [coerce](#)

**Examples**

```
## Create a count data matrix
A1 <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)

## Access
get_id(A1)
dim(A1) # Get the matrix dimensions
colnames(A1) <- letters[11:20] # Set the column names
colnames(A1) # Get the column names
rownames(A1) <- LETTERS[1:10] # Set the row names
rownames(A1) # Get the rownames

## Subset
A1[[1]] # Get the first value
A1[, ] # Get all values
A1[1, ] # Get the first row
A1[c("A", "B", "C"), ] # Get the first three rows
A1[c("A", "B", "C"), 1] # Get the first three rows of the first column
A1[, 1, drop = FALSE] # Get the first column

## Coerce counts to relative frequencies
B <- as_abundance(A1)
## Row sums are internally stored before coercing to a frequency matrix
get_totals(B) # Get row sums
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)
## Coerce to a co-occurrence matrix
B <- as_occurrence(A1)
```

---

SimilarityMatrix-class

*Similarity Matrix*

---

**Description**

An S4 class to represent a (dis)similarity matrix.

**Slots**

method A [character](#) string specifying the distance method used.

**Matrix ID**

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

### Get and set

In the code snippets below, `x` is a `*Matrix` object.

`get_id(x)` Get the ID of `x`.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of `x`.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of `x`.

### Access

In the code snippets below, `x` is a `*Matrix` object.

`dim(x)` Returns the dimension of `x`.

`nrow(x)` Returns the number of rows present in `x`.

`ncol(x)` Returns the number of columns present in `x`.

`dimnames(x)`, `dimnames(x) <- value` Retrieves or sets the row `dimnames` of `x` according to `value`.

`rownames(x)`, `rownames(x) <- value` Retrieves or sets the row names of `x` according to `value`.

`colnames(x)`, `colnames(x) <- value` Retrieves or sets the column names of `x` according to `value`.

### Subset

In the code snippets below, `x` is a `*Matrix` object.

`x[i, j]` Extracts elements selected by subscripts `i` and `j`. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or `NULL`. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as `x`.

`x[[i]]` Extracts a single element selected by subscript `i`.

### Author(s)

N. Frerebeau

### See Also

[NumericMatrix](#)

Other matrix: [AbundanceMatrix-class](#), [CountMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [StratigraphicMatrix-class](#), [coerce](#)

---

 StratigraphicMatrix-class

*Stratigraphic Matrix*


---

### Description

An S4 class to represent a stratigraphic matrix.

### Details

A stratigraphic matrix represents directed relationships between stratigraphic units. A stratigraphic matrix is an adjacency matrix (a non symmetric square matrix with zeros on its main diagonal), suitable to build a directed acyclic graph (DAG).

### Slots

units A [character](#) vector giving the stratigraphic unit names.

### Matrix ID

When a matrix is first created, an identifier is generated (UUID v4). This ID is preserved when coercing to another class. Thus, the object ID is unique within the same class, but two objects of different classes can have the same ID. This makes it possible to identify objects representing the same initial data and associate them with the results of specific computations.

### Get and set

In the code snippets below, x is a \*Matrix object.

`get_id(x)` Get the ID of x.

`get_dates(x)` **and** `set_dates(x) <- value` Get or set the dates of x.

`get_coordinates(x)` **and** `set_coordinates(x) <- value` Get or set the geographical coordinates of x.

### Access

In the code snippets below, x is a \*Matrix object.

`dim(x)` Returns the dimension of x.

`nrow(x)` Returns the number of rows present in x.

`ncol(x)` Returns the number of columns present in x.

`dimnames(x), dimnames(x) <- value` Retrieves or sets the row dimnames of x according to value.

`rownames(x), rownames(x) <- value` Retrieves or sets the row names of x according to value.

`colnames(x), colnames(x) <- value` Retrieves or sets the column names of x according to value.

**Subset**

In the code snippets below, x is a \*Matrix object.

`x[i, j]` Extracts elements selected by subscripts i and j. Indices are [numeric](#), [integer](#) or [character](#) vectors or empty (missing) or NULL. Numeric values are coerced to [integer](#) as by `as.integer` (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected. Returns an object of the same class as x.

`x[[i]]` Extracts a single element selected by subscript i.

**Author(s)**

N. Frerebeau

**See Also**

[LogicalMatrix](#)

Other matrix: [AbundanceMatrix-class](#), [CountMatrix-class](#), [IncidenceMatrix-class](#), [LogicalMatrix-class](#), [Matrix-class](#), [NumericMatrix-class](#), [OccurrenceMatrix-class](#), [SimilarityMatrix-class](#), [coerce](#)

**Examples**

```
# Principles of Archaeological Stratigraphy, fig. 12
harris <- matrix(
  data = c(2, 1,
           3, 1,
           4, 1,
           5, 2,
           5, 3,
           5, 4,
           6, 5,
           7, 1,
           7, 6,
           8, 1,
           8, 6,
           9, 7,
           9, 8),
  ncol = 2,
  byrow = TRUE,
  dimnames = list(NULL, c("lower", "upper"))
)

strati <- as_stratigraphy(harris)
get_units(strati)
```

---

subset

*Extract or Replace Parts of an Object*

---

## Description

Operators acting on objects to extract or replace parts.

## Usage

```
## S4 replacement method for signature 'Matrix'  
x[i, j] <- value  
  
## S4 replacement method for signature 'Matrix'  
x[[i, j]] <- value
```

## Arguments

x	An object from which to extract element(s) or in which to replace element(s).
i, j, ...	Indices specifying elements to extract or replace. Indices are <a href="#">numeric</a> , <a href="#">integer</a> or <a href="#">character</a> vectors or empty (missing) or NULL. Numeric values are coerced to <a href="#">integer</a> as by <a href="#">as.integer</a> (and hence truncated towards zero). Character vectors will be matched to the name of the elements. An empty index (a comma separated blank) indicates that all entries in that dimension are selected.
value	A possible value for the element(s) of x.

## Value

A subsetted object.

## Author(s)

N. Frerebeau

## See Also

Other mutator: [mutator](#)

## Examples

```
## Create a count data matrix  
A1 <- CountMatrix(data = sample(0:10, 100, TRUE), nrow = 10, ncol = 10)  
  
## Access  
get_id(A1)  
dim(A1) # Get the matrix dimensions  
colnames(A1) <- letters[11:20] # Set the column names  
colnames(A1) # Get the column names  
rownames(A1) <- LETTERS[1:10] # Set the row names
```

```
rownames(A1) # Get the rownames

## Subset
A1[[1]] # Get the first value
A1[, ] # Get all values
A1[1, ] # Get the first row
A1[c("A", "B", "C"), ] # Get the first three rows
A1[c("A", "B", "C"), 1] # Get the first three rows of the first column
A1[, 1, drop = FALSE] # Get the first column

## Coerce counts to relative frequencies
B <- as_abundance(A1)
## Row sums are internally stored before coercing to a frequency matrix
get_totals(B) # Get row sums
## This allows to restore the source data
A2 <- as_count(B)
all(A1 == A2)
## Coerce to a co-occurrence matrix
B <- as_occurrence(A1)
```

# Index

- .AbundanceMatrix
  - (AbundanceMatrix-class), 2
- .CountMatrix (CountMatrix-class), 8
- .IncidenceMatrix
  - (IncidenceMatrix-class), 12
- .OccurrenceMatrix
  - (OccurrenceMatrix-class), 15
- .SimilarityMatrix
  - (SimilarityMatrix-class), 17
- .StratigraphicMatrix
  - (StratigraphicMatrix-class), 19
- [,Matrix-method (subset), 21
- [<-,Matrix-method (subset), 21
- [[,Matrix-method (subset), 21
- [[<-,Matrix-method (subset), 21
  
- AbundanceMatrix, 7
- AbundanceMatrix-class, 2
- as.integer, 3, 9, 13, 16, 18, 20, 21
- as.vector, 8, 12
- as\_abundance (coerce), 6
- as\_abundance,ANY-method (coerce), 6
- as\_abundance-method (coerce), 6
- as\_count (coerce), 6
- as\_count,ANY-method (coerce), 6
- as\_count-method (coerce), 6
- as\_features (coerce), 6
- as\_features,Matrix-method (coerce), 6
- as\_features-method (coerce), 6
- as\_incidence (coerce), 6
- as\_incidence,ANY-method (coerce), 6
- as\_incidence-method (coerce), 6
- as\_occurrence (coerce), 6
- as\_occurrence,ANY-method (coerce), 6
- as\_occurrence-method (coerce), 6
- as\_similarity (coerce), 6
- as\_similarity,ANY-method (coerce), 6
- as\_similarity-method (coerce), 6
- as\_stratigraphy (coerce), 6
- as\_stratigraphy,ANY-method (coerce), 6
  
- as\_stratigraphy-method (coerce), 6
  
- character, 3, 9, 13, 16–21
- chronology, 4
- coerce, 3, 6, 9, 13, 16, 18, 20
- CountMatrix, 7
- CountMatrix (CountMatrix-class), 8
- CountMatrix-class, 8
  
- data.frame, 5–7
- dimnames, 8, 12
  
- expression, 8, 12
  
- geography, 10
- get (mutator), 14
- get\_coordinates (geography), 10
- get\_coordinates,Matrix-method (geography), 10
- get\_coordinates-method (geography), 10
- get\_dates (chronology), 4
- get\_dates,Matrix-method (chronology), 4
- get\_dates-method (chronology), 4
- get\_id (mutator), 14
- get\_id,ANY-method (mutator), 14
- get\_id-method (mutator), 14
- get\_method (mutator), 14
- get\_method,SimilarityMatrix-method (mutator), 14
- get\_method-method (mutator), 14
- get\_totals (mutator), 14
- get\_totals,AbundanceMatrix-method (mutator), 14
- get\_totals-method (mutator), 14
- get\_units (mutator), 14
- get\_units,StratigraphicMatrix-method (mutator), 14
- get\_units-method (mutator), 14
  
- IncidenceMatrix, 7

IncidenceMatrix  
    (IncidenceMatrix-class), 12  
IncidenceMatrix-class, 12  
integer, 3, 9, 13, 16, 18, 20, 21  
  
list, 7  
LogicalMatrix, 13, 20  
  
Matrix, 4, 7  
matrix, 6, 7  
mutator, 14, 21  
  
numeric, 2, 3, 9, 13, 16, 18, 20, 21  
NumericMatrix, 3, 9, 16, 18  
  
OccurrenceMatrix, 7  
OccurrenceMatrix-class, 15  
  
set (mutator), 14  
set\_coordinates, Matrix-method  
    (geography), 10  
set\_coordinates-method (geography), 10  
set\_coordinates<- (geography), 10  
set\_coordinates<- ,Matrix-method  
    (geography), 10  
set\_dates, Matrix-method (chronology), 4  
set\_dates-method (chronology), 4  
set\_dates<- (chronology), 4  
set\_dates<- ,Matrix-method (chronology),  
    4  
set\_totals, AbundanceMatrix-method  
    (mutator), 14  
set\_totals-method (mutator), 14  
set\_totals<- (mutator), 14  
set\_totals<- ,AbundanceMatrix-method  
    (mutator), 14  
SimilarityMatrix, 7  
SimilarityMatrix-class, 17  
StratigraphicMatrix, 7  
StratigraphicMatrix-class, 19  
subset, 15, 21