# Package 'grantham'

October 13, 2022

**Type** Package

**Title** Calculate the 'Grantham' Distance

**Version** 0.1.1

**Description** A minimal set of routines to calculate the 'Grantham' distance <doi:10.1126/science.185.4154.862>. The 'Grantham' distance attempts to provide a proxy for the evolutionary distance between two amino acids based on three key chemical properties: composition, polarity and molecular volume. In turn, evolutionary distance is used as a proxy for the impact of missense mutations. The higher the distance, the more deleterious the substitution is expected to be.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Depends** R (>= 2.10)

**Imports** tibble, magrittr, vctrs, dplyr, tidyr, rlang, stringr

**URL** https://maialab.org/grantham/, https://github.com/maialab/grantham

**BugReports** https://github.com/maialab/grantham/issues

**Suggests** spelling

**Language** en-US

**NeedsCompilation** no

**Author** Ramiro Magno [aut, cre] (<https://orcid.org/0000-0001-5226-3441>), Isabel Duarte [aut] (<https://orcid.org/0000-0003-0060-2936>), Ana-Teresa Maia [aut] (<https://orcid.org/0000-0002-0454-9207>), CINTESIS [cph, fnd]

**Maintainer** Ramiro Magno <ramiro.magno@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-07 03:20:02 UTC

# R topics documented:



---

`amino_acids` *The 20 standard amino acids*

---

## Description

The 20 amino acids that are encoded directly by the codons of the universal genetic code.

## Usage

```
amino_acids()
```

## Value

Three-letter codes of the standard amino acids.

## Examples

```
amino_acids()
```

---

`amino_acids_properties`
*Amino acid side chain property values*

---

## Description

A dataset containing the amino acid side chain property values —composition, polarity and molecular volume. These values were obtained from Table 1, Grantham (1974), doi: 10.1126/science.185.4154.862.

## Usage

```
amino_acids_properties
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 20 rows and 4 columns.

## Source

Table 1, Grantham (1974), doi: 10.1126/science.185.4154.862.

## Examples

```
amino_acids_properties
```

---

`amino_acid_pairs` *Generate amino acid pairs*

---

## Description

This function generates combinations of amino acids in pairs. By default, it generates all pair combinations of the 20 standard amino acids.

## Usage

```
amino_acid_pairs(
  x = amino_acids(),
  y = amino_acids(),
  keep_self = TRUE,
  keep_duplicates = TRUE,
  keep_reverses = TRUE
)
```

## Arguments

| | |
|---|---|
| `x` | A character vector of amino acids (three-letter codes). |
| `y` | Another character vector of amino acids (three-letter codes). |
| `keep_self` | Whether to keep pairs involving the same amino acid. |
| `keep_duplicates` | Whether to keep duplicated pairs. |
| `keep_reverses` | Whether to keep pairs that are reversed versions of others. E.g. if `keep_reverses` is `TRUE` the pairs `"Ser"-"Arg"` and `"Arg"-"Ser"` will be kept in the returned tibble; however, if `keep_reverses` is `FALSE`, only the first pair is preserved in the output. |

## Value

A tibble of amino acid pairs.

## Examples

```
# Generate all pairs of the 20 standard amino acids
amino_acid_pairs()

# Remove the self-to-self pairs
amino_acid_pairs(keep_self = FALSE)

# Generate specific combinations of Ser against Ala and Trp.
amino_acid_pairs(x = 'Ser', y = c('Ala', 'Trp'))
```

---

# as_one_letter *Convert three-letter amino acid codes to one-letter codes*

---

## Description

Converts three-letter amino acid abbreviations to one-letter codes, e.g., Leu to L. The accepted codes in the input include the 20 standard amino acids and also Asx (Asparagine or Aspartic acid), converted to B, and Glx (Glutamine or Glutamic acid) converted to Z.

## Usage

```
as_one_letter(x)
```

## Arguments

| | |
|---|---|
| x | A character vector of three-letter amino acid codes, e.g. "Ser", "Arg", "Leu", or "Asx". |

## Value

A character vector of one-letter amino acid codes, e.g. "S", "R", "L", or "B".

## Examples

```
# Convert Ser to S, Arg to R and Pro to P.
as_one_letter(c('Ser', 'Arg', 'Pro'))

# The function `as_one_letter()` is case insensitive on the input but will
# always return the one-letter codes in uppercase.
as_one_letter(c('ser', 'ArG', 'PRO'))

# Convert the codes of the 20 standard amino acids. Note that the function
# `amino_acids()` returns the three-letter codes of the 20 standard amino
# acids.
as_one_letter(amino_acids())

# Convert also special case codes Asx (Asparagine or Aspartic acid) and Glx
# (Glutamine or Glutamic acid)
as_one_letter(c('Asx', 'Glx'))
```

```
# Invalid codes in the input are converted to NA.
# "Ser" is correctly mapped to "S" but "Serine" is not as it is not a
# three-letter amino acid code (the same applies to "Glucose").
as_one_letter(c('Ser', 'Serine', 'Glucose'))
```

---

`as_three_letter` *Convert one-letter amino acid codes to three-letter codes*

---

## Description

Converts amino acid one-letter abbreviations to three-letter codes, e.g., L to Leu. The accepted codes in the input include the 20 standard amino acids and also B (Asparagine or Aspartic acid), converted to Asx, and Z (Glutamine or Glutamic acid) converted to Glx.

## Usage

```
as_three_letter(x)
```

## Arguments

| | |
|---|---|
| x | A character vector of one-letter amino acid codes, e.g. "S", "R", "L", or "P". |

## Value

A character vector of three-letter amino acid codes, e.g. "Ser", "Arg", "Leu", or "Pro".

## Examples

```
# Convert S to Ser, R to Arg and P to Pro.
as_three_letter(c('S', 'R', 'P'))

# The function `as_three_letter()` is case insensitive on the input but will
# always return the three-letter codes with the first letter in uppercase.
as_three_letter(c('S', 's', 'p', 'P'))

# Convert also special case codes B (Asparagine or Aspartic acid) and Z
# (Glutamine or Glutamic acid)
as_three_letter(c('B', 'Z'))

# Invalid codes in the input are converted to NA.
# "S" is correctly mapped to "Ser" but "Ser" and "Serine" are not
# one-letter amino acid codes and are therefore converted to NA.
as_three_letter(c('S', 's', 'Ser', 'Serine'))
```

grantham_distance *Grantham distance*

## Description

This function calculates the Grantham distance for pairs of amino acids. Amino acid identities should be provided as three-letter codes in `x` and `y`. Amino acids identified in `x` and `y` are matched element-wise, i.e. the first element of `x` is paired with the first element of `y`, and so on.

The Grantham distance attempts to provide a proxy for the evolutionary distance between two amino acids based on three key chemical properties: composition, polarity and molecular volume. In turn, evolutionary distance is used as a proxy for the impact of missense substitutions. The higher the distance, the more deleterious the substitution is.

The distance calculation is provided by two methods. The so-called *original* method, meaning that the amino acid distances used are the ones provided by Grantham in his original publication in Table 2. This is the default method. In addition, you may choose the *exact* method, which uses the chemical properties provided in Grantham's Table 1 to compute the amino acid differences anew. The distances calculated with the *exact* method are not rounded to the nearest integer and will differ by ~1 unit for some amino acid pairs from the *original* method.

If you want to calculate Grantham's distance by providing the values of the amino acid properties explicitly, then use `grantham_equation()` instead.

## Usage

```
grantham_distance(
  x,
  y,
  method = c("original", "exact"),
  alpha = 1.833,
  beta = 0.1018,
  gamma = 0.000399,
  rho = 50.723
)
```

## Arguments

| | |
|---|---|
| `x` | A character vector of amino acid three-letter codes. |
| `y` | A character vector of amino acid three-letter codes. |
| `method` | Either `"original"` (default) or `"exact"`, see description for more details. |
| `alpha` | The constant $\alpha$ in the equation of Grantham's paper, in page 863. |
| `beta` | The constant $\beta$ in the equation of Grantham's paper, in page 863. |
| `gamma` | The constant $\gamma$ in the equation of Grantham's paper, in page 863. |
| `rho` | Grantham's distances reported in Table 2, Science (1974). 185(4154): 862–4 by R. Grantham, are scaled by a factor (here named $\rho$) such that the mean value of all distances are 100. The `rho` parameter allows this factor $\rho$ to be changed. By default $\rho = 50.723$, the same value used by Grantham. This value is originally mentioned in the caption of Table 2 of the aforementioned paper. |

### Value

A tibble of Grantham's distances for each amino acid pair.

### Source

doi: 10.1126/science.185.4154.862.

### Examples

```
# Grantham's distance between Serine (Ser) and Glutamate (Glu)
grantham_distance('Ser', 'Glu')

# Grantham's distance between Serine (Ser) and Glutamate (Glu)
# with the "exact" method
grantham_distance('Ser', 'Glu', method = 'exact')

# `grantham_distance()` is vectorised
# amino acids are paired element-wise between `x` and `y`
grantham_distance(x = c('Pro', 'Gly'), y = c('Glu', 'Arg'))

# Use `amino_acid_pairs()` to generate pairs (by default generates all pairs)
aa_pairs <- amino_acid_pairs()
grantham_distance(x = aa_pairs$x, y = aa_pairs$y)
```

---

## grantham_distances_matrix

*Grantham distance matrix*

---

### Description

A dataset containing Grantham distances in the format of a matrix. These values were obtained from Table 2, Grantham (1974), doi: 10.1126/science.185.4154.862.

### Usage

```
grantham_distances_matrix
```

### Format

An object of class `matrix` (inherits from `array`) with 20 rows and 20 columns.

### Source

Table 2, Grantham (1974), doi: 10.1126/science.185.4154.862.

### Examples

```
grantham_distances_matrix
```

`grantham_equation` *Grantham distance*

## Description

This function calculates Grantham's distance $d_{i,j}$ between two amino acids ($i$ and $j$) based on their chemical properties:

$$d_{i,j} = \rho((\alpha(c_i - c_j)^2 + \beta(p_i - p_j)^2 + \gamma(v_i - v_j)^2)^{\frac{1}{2}}$$

This calculation is based on three amino acid side chain properties that were found to be the three strongest correlators with the relative substitution frequency (RSF) (references cited in Grantham (1974)), namely:

- composition $c$, meaning the atomic weight ratio of hetero (noncarbon) elements in end groups or rings to carbons in the side chain.
- polarity $p$;
- molecular volume $v$.

Each property difference is weighted by dividing by the mean distance found with it alone in the formula. The constants $\alpha$, $\beta$ and $\gamma$ are squares of the inverses of mean distances of each property, respectively.

The distances reported by Grantham (1974) are further scaled by a factor —here coined $\rho$— such that the mean of all distances is 100. Although this factor is not explicitly included in Grantham's distance formula, it is actually used for calculating the amino acid pair distances reported in Table 2 of Grantham's paper. So, for all intents and purposes, this factor should be regarded as part of the formula used to calculate Grantham distance, and therefore we include it explicitly in the equation above.

If you want to calculate Grantham's distance right off from the identity of the amino acids, instead of using their chemical properties, then use `grantham_distance()`.

## Usage

```
grantham_equation(
  c_i,
  c_j,
  p_i,
  p_j,
  v_i,
  v_j,
  alpha = 1.833,
  beta = 0.1018,
  gamma = 0.000399,
  rho = 50.723
)
```

## Arguments

| | |
|---|---|
| c_i | composition value for the *ith* amino acid. |
| c_j | composition value for the *jth* amino acid. |
| p_i | polarity value for the *ith* amino acid. |
| p_j | polarity value for the *jth* amino acid. |
| v_i | molecular volume value for the *ith* amino acid. |
| v_j | molecular volume value for the *jth* amino acid. |
| alpha | The constant $\alpha$ in the equation of Grantham's paper, in page 863. |
| beta | The constant $\beta$ in the equation of Grantham's paper, in page 863. |
| gamma | The constant $\gamma$ in the equation of Grantham's paper, in page 863. |
| rho | Grantham's distances reported in Table 2, Science (1974). 185(4154): 862–4 by R. Grantham, are scaled by a factor (here named $\rho$) such that the mean value of all distances are 100. The rho parameter allows this factor $\rho$ to be changed. By default $\rho = 50.723$, the same value used by Grantham. This value is originally mentioned in the caption of Table 2 of the aforementioned paper. |

## Value

A double vector of Grantham's distances.

## See Also

Check amino_acids_properties for a table of the three property values that can be used with this formula. This data set is from Table 1, Science (1974). 185(4154): 862–4 by R. Grantham.

# Index