

# Package: SeqAlignR (via r-universe)

July 7, 2024

**Type** Package

**Title** Sequence Alignment and Visualization Tool

**Version** 0.1.1

**Description** Computes the optimal alignment of two character sequences.  
Visualizes the result of the alignment in a matrix plot.  
Needleman, Saul B.; Wunsch, Christian D. (1970) ``A general  
method applicable to the search for similarities in the amino  
acid sequence of two proteins"  
<[doi:10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4)>.

**Imports** plot.matrix, graphics

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Leonard Persson Norblad [aut, cre]  
(<<https://orcid.org/0009-0004-8562-4118>>), Kamil Slowikowski  
[ctb] (<<https://orcid.org/0000-0002-2843-6370>>)

**Maintainer** Leonard Persson Norblad <[leonard.norblad@gmail.com](mailto:leonard.norblad@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-02-07 17:50:13 UTC

## Contents

align_sequences . . . . .	2
plot.alignment . . . . .	3
print.alignment . . . . .	4
SeqAlignR . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

`align_sequences`*Sequence Alignment*

---

### Description

Performs sequence alignment on two sequences by a user specified alignment method. As of now, only the Needleman-Wunsch algorithm is supported.

### Usage

```
align_sequences(seq1, seq2, d, mismatch, match, method = "needleman")
```

### Arguments

<code>seq1</code>	First sequence to align.
<code>seq2</code>	Second sequence to align.
<code>d</code>	Gap penalty, should be negative.
<code>mismatch</code>	Mismatch penalty, should be negative.
<code>match</code>	Match score, should be positive.
<code>method</code>	Name of alignment algorithm. Currently only supports "needleman".

### Value

Object of class `alignment` representing the alignment result. This object can be utilized with the `plot.alignment` function to visualize the alignment matrix and the `print.alignment` function to display alignments in the console.

### References

For more details on the Needleman-Wunsch algorithm, see the [wikipedia page](#).

### Examples

```
seq1 <- "GCATGCG"  
seq2 <- "GATTACA"  
# Run the Needleman-Wunsch algorithm  
align_sequences(seq1, seq2, d = -1, mismatch = -1, match = 1)
```

---

plot.alignment	<i>Plot Alignment Matrix</i>
----------------	------------------------------

---

### Description

Produces a plot displaying the alignment matrix of seq1 and seq2.

### Usage

```
## S3 method for class 'alignment'  
plot(x, ...)
```

### Arguments

x	Object of class alignment (see <a href="#">align_sequences</a> ).
...	Additional parameters to be passed to the plot() function.

### Details

The first sequence (seq1) is represented by the columns and the second sequence (seq2) is represented by the rows. The first column and first row are left blank, meaning a gap. Each cell in the matrix displays the score. The subtitle states the match, mismatch and gap penalty d used in the algorithm. A mismatch is shown by the red arrows, a match by the blue arrows, and a gap by the green arrows. The alignment(s) with the highest score are highlighted with thick gray borders.

### Value

Plot of the alignment matrix.

### References

The implementation is inspired by the visualization ([code](#)) by Kamil Slowikowski ([ORCID](#)).

### Examples

```
seq1 <- "GCATGCG"  
seq2 <- "GATTACA"  
# Run the Needleman-Wunsch algorithm  
alignment1 <- align_sequences(seq1, seq2, d = -1, mismatch = -1, match = 1)  
# Plot the matrix  
plot(alignment1)
```

print.alignment      *Print Alignments*

---

### Description

Prints the alignments between seq1 and seq2 with the highest score.

### Usage

```
## S3 method for class 'alignment'  
print(x, ...)
```

### Arguments

x                      Object of class alignment (see [align\\_sequences](#)).  
...                     Additional parameters to be passed to the cat() function, displaying the alignment.

### Details

The printed message includes the alignment score. This function may display multiple alignments, as alignments with the same score are possible.

### Value

Console print of alignments.

### Examples

```
seq1 <- "GCATGCG"  
seq2 <- "GATTACA"  
# Run the Needleman-Wunsch algorithm  
alignment1 <- align_sequences(seq1, seq2, d = -1, mismatch = -1, match = 1)  
# Print the alignments  
print(alignment1)
```

---

SeqAlignR                      *Sequence Alignment Tool*

---

### Description

Sequence alignment and visualization tool designed to enhance understanding of sequence alignment algorithms, such as the Needleman-Wunsch algorithm. Through detailed matrix plot visualizations with arrows illustrating the path of different alignments, users can gain insights into how these algorithms score and identify optimal alignments between two sequences.

**Getting started**

1. Install the SeqAlignR package from CRAN using:  
`install.packages("SeqAlignR")`
2. Load the SeqAlignR package into your R session:  
`library(SeqAlignR)`
3. Explore the documentation for [align\\_sequences](#) or see the example below:  
`?align_sequences`

**Author(s)**

Leonard Persson Norblad <leonard.norblad@gmail.com> ([ORCID](#))

**References**

The graphics in the package were inspired by [this code](#) by Kamil Slowikowski ([ORCID](#)).

**Examples**

```
seq1 <- "GCATGCG"
seq2 <- "GATTACA"
# Run the Needleman-Wunsch algorithm
alignment1 <- align_sequences(seq1, seq2, d = -1, mismatch = -1, match = 1, method="needleman")
# Print the alignments
print(alignment1)
# Plot the matrix
plot(alignment1)
```

# Index

`align_sequences`, [2](#), [3–5](#)

`plot.alignment`, [2](#), [3](#)

`print.alignment`, [2](#), [4](#)

`SeqAlignR`, [4](#)