

Package ‘genoaligner’

September 24, 2026

Type Package

Title GPU-Portable Pairwise Sequence Alignment (WFA + Smith-Waterman)

Version 1.0.0

Description Pairwise sequence alignment from one portable C++17 core: edit-distance (Levenshtein / WFA-equivalent) and Smith-Waterman local alignment, both with score and CIGAR reconstruction. The core builds and runs anywhere (it is the CPU backend, so it needs no GPU toolchain); the wavefront/GPU backend (ROCm/CUDA) computes the same results with acceleration and is tracked in the sibling C++ repository. A batch API aligns many pairs in one call and reports how many were resolved, so silent under-serving is impossible. Designed to slot into data.frame/tibble pipelines.

URL <https://github.com/alrobles/genoaligner-r>,
<https://alrobles.github.io/genoaligner-r/>

BugReports <https://github.com/alrobles/genoaligner-r/issues>

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spelling

SystemRequirements C++17

NeedsCompilation yes

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Author Angel Robles-Fernandez [aut, cre]

Maintainer Angel Robles-Fernandez <a.l.robles.fernandez@gmail.com>

Repository CRAN

Date/Publication 2026-09-24 04:20:02 UTC

Contents

genoaligner-package	2
align	3
align_sw	4
Index	5

genoaligner-package	<i>genoaligner: GPU-portable pairwise sequence alignment</i>
---------------------	--

Description

Pairwise sequence alignment from one portable C++17 core: edit distance (Levenshtein / WFA-equivalent) and Smith-Waterman local alignment, both with score and CIGAR reconstruction. The core builds and runs anywhere (it is the CPU backend, so it needs no GPU toolchain); the sibling C++ library tracks the GPU (ROCm/CUDA) backend that computes the same results with wavefront acceleration.

Details

The two entry points are `align` (edit distance bounded by `smax`) and `align_sw` (local Smith-Waterman with affine gaps). Both are vectorised over pairs, so a two-column pipeline (query, reference) maps straight to a `data.frame` of scores and CIGARs.

Author(s)

Maintainer: Angel Robles-Fernandez <a.l.robles.fernandez@gmail.com>

Authors:

- Angel Robles-Fernandez <a.l.robles.fernandez@gmail.com>

See Also

Useful links:

- <https://github.com/alroble/genoaligner-r>
- <https://alroble.github.io/genoaligner-r/>
- Report bugs at <https://github.com/alroble/genoaligner-r/issues>

align	<i>Align pairs by edit distance</i>
-------	-------------------------------------

Description

Computes the Levenshtein edit distance (substitution = 1, insertion = 1, deletion = 1) between each pair of sequences, with an optional CIGAR reconstruction. This is the same distance the WFA formulation computes; wavefront/GPU speed is a backend concern, the result is identical.

Usage

```
align(pattern, text, smax = 64, with_cigar = TRUE)
```

Arguments

pattern	Character vector; query sequences (recycled to match text).
text	Character vector; reference sequences (recycled to match pattern). NA entries produce NA rows.
smax	Numeric; maximum edit distance searched, in $[0, 511]$. Pairs whose true distance exceeds it are unresolved by design.
with_cigar	Logical; if FALSE only scores are computed (cheaper, no traceback) and cigar is NA.

Details

A pair is *resolved* only when its true distance is at most `smax`; otherwise score is NA and the pair is reported as `resolved = FALSE`. The count of resolved pairs is deliberately visible: a caller that only reads scores can silently miss that most input was abandoned when `smax` is set too tightly.

Value

A data.frame with one row per pair and columns: score (edit distance, NA if unresolved), resolved (logical), cigar (over {M, X, I, D}, NA if unresolved or !with_cigar), rescore_ok and wellformed_ok (validation flags computed in the library).

CIGAR convention (identical for `align` and `align_sw`): M/X consume a text and a pattern base; I consumes an extra *text* base; D consumes an extra *pattern* base.

Examples

```
align("ACGTACGT", "ACGTTTCGT", smax = 8)
align(c("AAAACCC", "ACGT"), c("AAAATCC", "ACCT"), smax = 8)
```

align_sw	<i>Smith-Waterman local alignment</i>
----------	---------------------------------------

Description

Local alignment with affine gap penalties {match, mismatch, gap_open, gap_extend}. Penalties are positive and subtracted. A pair with score == 0 has no positive-scoring local alignment (disjoint sequences); all four coordinates are -1 and cigar is empty.

Usage

```
align_sw(text, pattern, scoring = c(2L, -3L, 5L, 2L), with_cigar = TRUE)
```

Arguments

text	Character vector; row-axis sequences (recycled).
pattern	Character vector; column-axis sequences (recycled).
scoring	Numeric vector of length 4, named or positional: (match, mismatch, gap_open, gap_extend). Recycled if a single vector is given. The defaults are an example, not a recommendation.
with_cigar	Logical; if FALSE, only score and end coordinates are computed (start coordinates and cigar are NA/-1).

Value

A data.frame with one row per pair and columns: score, start_i, start_j, end_i, end_j (0-based coordinates of the aligned span, -1 when score is 0/unknown), cigar (over the span), rescore_ok, wellformed_ok and too_large (the pair's matrix exceeded the supported size limit).

Examples

```
align_sw("TTTACGTGTT", "ACGTGT", scoring = c(2, -3, 5, 2))
align_sw(c("ACGTACGT", "GGGG"), c("ACGTTCGT", "ACGT"), c(2, -3, 5, 2))
```

Index

`align`, [2](#), [3](#)

`align_sw`, [2](#), [4](#)

`genoaligner` (`genoaligner-package`), [2](#)

`genoaligner-package`, [2](#)