Package 'blastar'

January 14, 2026

Title BLAST and Sequence Analysis Tools

Version 0.1.1

Description Description: Provides streamlined tools for retrieving sequences from NCBI, performing sequence alignments (pairwise and multiple), and building phylogenetic trees. Implements the Needleman-Wunsch algorithm for global alignment (Needleman & Wunsch (1970) <doi:10.1016/0022-2836(70)90057-4>), Smith-Waterman for local alignment (Smith & Waterman (1981) <doi:10.1016/0022-2836(81)90087-5>), and Neighbor-Joining for tree construction (Saitou & Nei (1987) <doi:10.1093/oxfordjournals.molbev.a040454>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports rentrez, ape, Biostrings, dplyr, tibble

Suggests msa, pwalign, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/loukesio/blastar

BugReports https://github.com/loukesio/blastar/issues

NeedsCompilation no

Author Loukas Theodosiou [aut, cre]

Maintainer Loukas Theodosiou <theodosiou@evolbio.mpg.de>

Repository CRAN

Date/Publication 2026-01-14 18:00:22 UTC

Contents

`align_sequences` *Align DNA Sequences (Pairwise or Multiple)*

Description

This function takes a tibble with a "sequence" column (and optional "accession" names) and performs either a pairwise alignment between two sequences or a multiple sequence alignment (MSA) across all.

Usage

```
align_sequences(
  df,
  method = c("pairwise", "msa"),
  pairwise_type = "global",
  msa_method = "ClustalOmega",
  seq_indices = c(1, 2)
)
```

Arguments

`df`	A tibble or data.frame containing at least: • `sequence`: character vector of DNA sequences • `accession` (optional): names for each sequence; if present, they will be used as identifiers in the alignment object.
`method`	One of: • "pairwise": perform a pairwise alignment between two sequences • "msa": perform a multiple sequence alignment on all sequences
`pairwise_type`	For pairwise only, alignment type: "global" (Needleman–Wunsch), "local" (Smith–Waterman), or "overlap".
`msa_method`	For MSA only, method name: "ClustalOmega", "ClustalW", or "Muscle".
`seq_indices`	Integer vector of length 2; indices of the two sequences to align when `method = "pairwise"`. Defaults to `c(1,2)`.

Value

If `method="pairwise"`, a list with:

- `alignment`: a `PairwiseAlignmentsSingleSubject` object
- `pid`: percent identity (numeric) If `method="msa"`, an object of class `MsaDNAMultipleAlignment` or similar.

Examples

```
# Pairwise alignment example (requires pwalign package)
if (requireNamespace("pwalign", quietly = TRUE)) {
  data <- data.frame(
    accession = c("seq1", "seq2"),
    sequence  = c("ACGTACGTACGT", "ACGTACGTTTGT"),
    stringsAsFactors = FALSE
  )

  res_pw <- align_sequences(
    df = data,
    method = "pairwise",
    pairwise_type = "global"
  )
  res_pw$pid
}

# Multiple sequence alignment (requires msa package)
if (requireNamespace("msa", quietly = TRUE)) {
  data_msa <- data.frame(
    accession = c("seq1", "seq2", "seq3"),
    sequence = c("ATGCATGC", "ATGCTAGC", "ATGGATGC")
  )
  res_msa <- align_sequences(data_msa, method = "msa", msa_method = "ClustalOmega")
  print(res_msa)
}
```

`build_nj_tree` *Build a Neighbor-Joining tree from a multiple sequence alignment*

Description

This function takes a Multiple Sequence Alignment (MSA) object (e.g., output of `align_sequences(method = "msa")`) and generates a Neighbor-Joining (NJ) tree.

Usage

```
build_nj_tree(msa, model = "raw", pairwise.deletion = TRUE)
```

Arguments

`msa`	A multiple alignment object (class `MsaDNAMultipleAlignment` or similar)
`model`	Evolutionary model for distance calculation passed to `ape::dist.dna` (e.g., "raw", "JC69", "K80", etc.)
`pairwise.deletion`	Logical. If TRUE, compute distances with pairwise deletion

Value

An object of class phylo (NJ tree)

Examples

```
# Build NJ tree from multiple sequence alignment (requires msa package)
if (requireNamespace("msa", quietly = TRUE)) {
  # Create example sequences
  df <- data.frame(
    accession = c("seq1", "seq2", "seq3"),
    sequence = c("ATGCATGC", "ATGCTAGC", "ATGGATGC")
  )

  # Generate MSA
  msa_result <- align_sequences(df, method = "msa", msa_method = "ClustalOmega")

  # Build NJ tree
  tree <- build_nj_tree(msa_result, model = "raw")
  print(tree)
}
```

---

fetch_metadata *Fetch Metadata (and optionally sequence ranges) from NCBI*

Description

Fetch Metadata (and optionally sequence ranges) from NCBI

Usage

```
fetch_metadata(accessions, db = c("nuccore", "protein"), seq_range = NULL)
```

Arguments

accessions	Character vector of accession numbers.
db	Either "nuccore" or "protein".
seq_range	Either: • NULL (default): fetch full sequence for every accession • numeric(2): fetch that same start–end for *all* accessions • named list: each element is a numeric(2) vector, names are accessions; will fetch only that slice for the named accession, full sequence for others.

Value

A tibble with columns accession, accession_version, title, organism, sequence

Examples

```
# Fetch metadata for a nucleotide sequence
result <- fetch_metadata("NM_000546", db = "nuccore")

# Fetch specific sequence range (positions 1-100)
result_range <- fetch_metadata("NM_000546", db = "nuccore", seq_range = c(1, 100))

# Fetch multiple accessions
result_multi <- fetch_metadata(c("NM_000546", "NM_001126"), db = "nuccore")
```

Index