

Package: Rdragmap (via r-universe)

October 10, 2026

Title An 'R' Wrapper for Portable 'DRAGMAP' Short-Read Alignment

Version 0.0.0.9000

Description Builds the pinned in-tree 'DRAGMAP' hash-table generator and short-read aligner with a portable baseline and runtime-gated AVX2 on supported x86 processors. Stateless R functions build reference indexes and align FASTQ input through one package-owned executable without searching 'PATH'. The same source also compiles for AArch64, where 'SIMDe' maps the imported 128-bit vector operations to NEON.

License GPL-3

Encoding UTF-8

Language en-US

Depends R (>= 4.4.0)

Imports processx (>= 3.8.0), S7 (>= 0.2.0)

Suggests tinytest

SystemRequirements A Unix-like system, GNU make, a C++17 compiler, zlib, bzip2, Boost Iostreams development files, and Boost Program Options development files

OS_type unix

Additional_repositories <https://rgenomicsetl.r-universe.dev>

URL <https://github.com/RGenomicsETL/Rdragmap>,
<https://rgenomicsetl.github.io/Rdragmap/>

BugReports <https://github.com/RGenomicsETL/Rdragmap/issues>

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/Needs/website pkgdown

Collate 'validation.R' 'results.R' 'process.R' 'executable.R'
'reference.R' 'align.R' 'Rdragmap-package.R'

Config/pak/sysreqs libboost-all-dev make zlib1g-dev

Repository <https://rgenomicsetl.r-universe.dev>
Date/Publication 2026-08-10 23:14:41 UTC
RemoteUrl <https://github.com/RGenomicsETL/Rdragmap>
RemoteRef main
RemoteSha 02217084ea247ba90d4427077a12c51cf934ad12
RemoteSubdir Rdragmap

Contents

rdragmap_align	2
rdragmap_build_index	4
rdragmap_contract_violation	5
rdragmap_error_value	6
rdragmap_executables	7
rdragmap_index	7
rdragmap_is_error	8
RdragmapErrorValue	8
RdragmapExecutables	9
RdragmapIndex	10
RdragmapRunResult	10
Index	12

rdragmap_align	<i>Align FASTQ reads with a DRAGMAP v8 reference index</i>
----------------	--

Description

Runs the package-owned dragen-os executable once and writes SAM plus native mapping metrics beside output_sam. Paired input also writes native insert-size statistics. The wrapper preserves map/align order by default so that the native SAM ordering remains deterministic.

Usage

```
rdragmap_align(  
  index,  
  read1,  
  output_sam,  
  read2 = character(),  
  executables = rdragmap_executables(),  
  read_group_id = "1",  
  sample_name = "none",  
  threads = 1L,  
  fastq_offset = 33L,  
  enable_sampling = TRUE,  
  preserve_order = TRUE,
```

```

    mmap_reference = FALSE
  )

```

Arguments

index	A validated RdragmapIndex.
read1	Absolute first FASTQ path, optionally gzip-compressed.
output_sam	Absolute, not-yet-existing .sam output path.
read2	Empty or one absolute mate FASTQ path.
executables	Explicit executable resolution from rdragmap_executables().
read_group_id	Read-group identifier recorded in the SAM header.
sample_name	Read-group sample recorded in the SAM header.
threads	Positive mapper/aligner worker count.
fastq_offset	FASTQ quality offset, 33L or 64L.
enable_sampling	Whether the native program estimates paired-end insert-size parameters from the input.
preserve_order	Whether to preserve deterministic map/align order.
mmap_reference	Whether to memory-map the uncompressed hash, extension, and reference files. The index must have been built with write_uncompressed = TRUE or otherwise contain hash_table.bin and extend_table.bin.

Value

RdragmapAlignmentResult or an RdragmapErrorValue.

Examples

```

work <- tempfile("rdragmap-example-")
dir.create(work)
built <- rdragmap_build_index(
  reference_fasta = system.file("extdata", "tiny.fasta", package = "Rdragmap"),
  index_directory = file.path(work, "index"),
  threads = 1L,
  hash_size = "16MB"
)
if (!rdragmap_is_error(built)) {
  aligned <- rdragmap_align(
    index = built@index,
    read1 = system.file("extdata", "one.fastq", package = "Rdragmap"),
    output_sam = file.path(work, "example.sam"),
    read_group_id = "tiny",
    sample_name = "tiny",
    threads = 1L,
    enable_sampling = FALSE
  )
  if (!rdragmap_is_error(aligned)) aligned@outputs
}

```

```

}
unlink(work, recursive = TRUE, force = TRUE)

```

rdragmap_build_index *Build a DRAGMAP v8 reference index*

Description

Runs the package-owned dragen-os hash-table generator once. The output directory must not already exist; its parent must exist. The returned index always contains the compressed hash table. It can also retain the much larger uncompressed hash and extension tables required for memory-mapped alignment.

Usage

```

rdragmap_build_index(
  reference_fasta,
  index_directory,
  executables = rdragmap_executables(),
  threads = 1L,
  hash_size = "0GB",
  memory_limit = "0GB",
  seed_length = 21L,
  seed_interval = 1,
  mask_bed = character(),
  decoys = character(),
  max_multi_base_seeds = integer(),
  write_uncompressed = FALSE
)

```

Arguments

reference_fasta	Absolute FASTA path.
index_directory	Absolute, not-yet-existing output directory.
executables	Explicit executable resolution from rdragmap_executables().
threads	Positive hash-generation worker count.
hash_size	Hash table size with B, KB, MB, or GB units. "0GB" requests the native automatic size calculation.
memory_limit	Hash generation memory limit in the same units. "0GB" requests the native automatic limit.
seed_length	Initial seed length.
seed_interval	Reference positions per seed.

mask_bed Empty or one absolute BED path passed to --ht-mask-bed.

decoys Empty or one absolute FASTA path passed to --ht-decoys.

max_multi_base_seeds Empty or one non-negative maximum passed to --ht-max-multi-base-seeds.

write_uncompressed Whether to retain hash_table.bin and extend_table.bin. These files require substantially more disk space but permit rdragmap_align(mmap_reference = TRUE) without decompressing the hash table for every alignment process.

Value

RdragmapIndexBuildResult or an RdragmapErrorValue.

Examples

```
work <- tempfile("rdragmap-example-")
dir.create(work)
built <- rdragmap_build_index(
  reference_fasta = system.file("extdata", "tiny.fasta", package = "Rdragmap"),
  index_directory = file.path(work, "index"),
  threads = 1L,
  hash_size = "16MB"
)
if (!rdragmap_is_error(built)) built@index
unlink(work, recursive = TRUE, force = TRUE)
```

rdragmap_contract_violation

Construct a typed Rdragmap contract condition

Description

Construct a typed Rdragmap contract condition

Usage

```
rdragmap_contract_violation(
  message,
  call = NULL,
  code = "invalid_contract",
  details = list()
)
```

Arguments

message	Human-readable description.
call	Call associated with the violation.
code	Stable machine-readable code.
details	Structured condition details.

Value

An rdragmap_contract_violation condition.

rdragmap_error_value *Construct a typed Rdragmap operational error*

Description

Construct a typed Rdragmap operational error

Usage

```
rdragmap_error_value(
  message,
  kind = c("input", "output", "process"),
  code,
  details = list(),
  source = NULL,
  operation = character(),
  status = integer()
)
```

Arguments

message	Human-readable description.
kind	Operational error category.
code	Stable machine-readable code.
details	Structured error details.
source	Optional source condition or backend value.
operation	Empty or one stable child-process operation identifier.
status	Empty or one child-process exit status.

Value

An RdragmapErrorValue subclass.

rdragmap_executables	<i>Resolve a packaged or explicitly supplied dragen-os executable</i>
----------------------	---

Description

This function never searches PATH. With no supplied path, it resolves only the executable installed by this package. Supply one absolute directory or one absolute executable path to use a separately built compatible program.

Usage

```
rdragmap_executables(directory = character(), dragen_os = character())
```

Arguments

directory	Empty or one absolute executable directory.
dragen_os	Empty or one absolute dragen-os path.

Value

RdragmapExecutables or an RdragmapInputErrorValue.

Examples

```
native <- rdragmap_executables()
if (!rdragmap_is_error(native)) native@dragen_os
```

rdragmap_index	<i>Validate a DRAGMAP v8 reference-index directory</i>
----------------	--

Description

The directory must contain hash_table.cfg.bin, hash_table.cmp, and reference.bin. An index built with write_uncompressed = TRUE also contains hash_table.bin and extend_table.bin for mmap_reference = TRUE alignment.

Usage

```
rdragmap_index(directory)
```

Arguments

directory	Absolute reference-index directory.
-----------	-------------------------------------

Value

RdragmapIndex or an RdragmapInputErrorValue.

<code>rdragmap_is_error</code>	<i>Test whether a value is an Rdragmap operational error</i>
--------------------------------	--

Description

Test whether a value is an Rdragmap operational error

Usage

```
rdragmap_is_error(value)
```

Arguments

<code>value</code>	Any R value.
--------------------	--------------

Value

A logical scalar.

<code>RdragmapErrorValue</code>	<i>A typed Rdragmap operational error value</i>
---------------------------------	---

Description

Invalid API calls signal `rdragmap_contract_violation()` conditions. Expected unavailable inputs, occupied outputs, unavailable executables, and native child-process failures are returned as values.

Usage

```
RdragmapErrorValue(
  message = character(0),
  code = character(0),
  details = list(),
  source = NULL,
  operation = character(0),
  status = integer(0)
)
```

```
RdragmapInputErrorValue(
  message = character(0),
  code = character(0),
  details = list(),
  source = NULL,
  operation = character(0),
  status = integer(0)
)
```

```

)

RdragmapOutputErrorValue(
  message = character(0),
  code = character(0),
  details = list(),
  source = NULL,
  operation = character(0),
  status = integer(0)
)

RdragmapProcessErrorValue(
  message = character(0),
  code = character(0),
  details = list(),
  source = NULL,
  operation = character(0),
  status = integer(0)
)

```

Arguments

message	Human-readable description.
code	Stable machine-readable code.
details	Structured error details.
source	Optional source condition or backend value.
operation	Empty or one stable child-process operation identifier.
status	Empty or one child-process exit status.

RdragmapExecutables	<i>Explicit dragen-os child executable</i>
---------------------	--

Description

Explicit dragen-os child executable

Usage

```
RdragmapExecutables(dragen_os = character(0))
```

Arguments

dragen_os	Absolute path to dragen-os.
-----------	-----------------------------

RdragmapIndex	<i>An Rdragmap reference index</i>
---------------	------------------------------------

Description

A reference index produced by `rdragmap_build_index()` or validated by `rdragmap_index()`. It records an absolute directory containing the native DRAGMAP v8 reference, compressed hash table, and binary configuration. An index may additionally retain uncompressed tables for memory-mapped alignment.

Usage

```
RdragmapIndex(directory = character(0))
```

Arguments

directory	Absolute reference-index directory.
-----------	-------------------------------------

RdragmapRunResult	<i>A completed Rdragmap child-process operation</i>
-------------------	---

Description

A completed Rdragmap child-process operation

A completed Rdragmap index build

A completed Rdragmap FASTQ alignment

Usage

```
RdragmapRunResult(
  operation = character(0),
  outputs = list(),
  status = integer(0),
  stdout = character(0),
  stderr = character(0)
)
```

```
RdragmapIndexBuildResult(
  operation = character(0),
  outputs = list(),
  status = integer(0),
  stdout = character(0),
  stderr = character(0),
  index = RdragmapIndex()
)
```

```
RdragmapAlignmentResult(  
  operation = character(0),  
  outputs = list(),  
  status = integer(0),  
  stdout = character(0),  
  stderr = character(0),  
  index = RdragmapIndex(),  
  paired = logical(0)  
)
```

Arguments

<code>operation</code>	Stable operation identifier.
<code>outputs</code>	Named output paths produced by the operation.
<code>status</code>	Child-process exit status.
<code>stdout</code>	Captured standard output.
<code>stderr</code>	Captured standard error.
<code>index</code>	The reference index used by the operation.
<code>paired</code>	Whether the operation received paired FASTQ input.

Index

`rdragmap_align`, 2
`rdragmap_build_index`, 4
`rdragmap_contract_violation`, 5
`rdragmap_error_value`, 6
`rdragmap_executables`, 7
`rdragmap_index`, 7
`rdragmap_is_error`, 8
`RdragmapAlignmentResult`
 (`RdragmapRunResult`), 10
`RdragmapErrorValue`, 8
`RdragmapExecutables`, 9
`RdragmapIndex`, 10
`RdragmapIndexBuildResult`
 (`RdragmapRunResult`), 10
`RdragmapInputErrorValue`
 (`RdragmapErrorValue`), 8
`RdragmapOutputErrorValue`
 (`RdragmapErrorValue`), 8
`RdragmapProcessErrorValue`
 (`RdragmapErrorValue`), 8
`RdragmapRunResult`, 10