Package ‘bigANNOY’

April 1, 2026

Type Package

Title Approximate k-Nearest Neighbour Search for 'bigmemory' Matrices with Annoy

Version 0.3.0

Date 2026-03-27

Author Frederic Bertrand [aut, cre]

Maintainer Frederic Bertrand <frederic.bertrand@lecnam.net>

Description Approximate Euclidean k-nearest neighbour search routines that operate on 'bigmemory::big.matrix' data through Annoy indexes created with 'RcppAnnoy'. The package builds persistent on-disk indexes plus sidecar metadata from streamed 'big.matrix' rows, supports euclidean, angular, Manhattan, and dot-product Annoy metrics, and can either return in-memory results or stream neighbour indices and distances into destination 'bigmemory' matrices. Explicit index life cycle helpers, stronger metadata validation, descriptor-aware file-backed workflows, and benchmark helpers are also included.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports methods, Rcpp, RcppAnnoy

LinkingTo BH, bigmemory, Rcpp, RcppAnnoy

Suggests bigmemory, knitr, litedown, testthat (>= 3.0.0)

VignetteBuilder litedown

Encoding UTF-8

NeedsCompilation yes

URL https://fbertran.github.io/bigANNOY/,
https://github.com/fbertran/bigANNOY

BugReports https://github.com/fbertran/bigANNOY/issues

RoxygenNote 7.3.3

Config/testthat/edition 3

Repository CRAN

Date/Publication 2026-04-01 08:00:33 UTC

Contents

`annoy_build_bigmatrix` *Build an Annoy index from a* `bigmemory::big.matrix`

Description

Stream the rows of a reference `bigmemory::big.matrix` into an on-disk Annoy index and write a small sidecar metadata file next to it. The returned `bigannoy_index` can be reopened later with `annoy_open_index()`.

Usage

```
annoy_build_bigmatrix(
  x,
  path,
  n_trees = 50L,
  metric = "euclidean",
  seed = NULL,
  build_threads = -1L,
  block_size = annoy_default_block_size(),
  metadata_path = NULL,
  load_mode = "lazy"
)
```

Arguments

`x`	A `bigmemory::big.matrix` or an external pointer referencing the reference matrix.
`path`	File path where the Annoy index should be written.
`n_trees`	Number of Annoy trees to build.
`metric`	Distance metric. bigANNOY v2 supports `"euclidean"`, `"angular"`, `"manhattan"`, and `"dot"`.

`seed` Optional positive integer seed used to initialize Annoy's build RNG.

`build_threads` Build-thread setting passed to Annoy's native backend. Use `-1L` for Annoy's default.

`block_size` Number of rows processed per streamed block while building the index.

`metadata_path` Optional path for the sidecar metadata file. Defaults to `paste0(path, ".meta")`.

`load_mode` Whether to keep the returned index metadata-only until first search (`"lazy"`) or eagerly load a live index handle immediately (`"eager"`).

Value

A `bigannoy_index` object describing the persisted Annoy index.

`annoy_close_index` *Close any loaded Annoy handle cached inside a* `bigannoy_index`

Description

Close any loaded Annoy handle cached inside a `bigannoy_index`

Usage

```
annoy_close_index(index)
```

Arguments

`index` A `bigannoy_index`.

Value

`index`, invisibly.

`annoy_is_loaded` *Check whether an index currently has a loaded in-memory handle*

Description

Check whether an index currently has a loaded in-memory handle

Usage

```
annoy_is_loaded(index)
```

Arguments

`index` A `bigannoy_index`.

Value

TRUE when a live native or debug-only handle is cached, otherwise FALSE.

---

annoy_load_bigmatrix *Load an existing Annoy index for bigmatrix workflows*

Description

Load an existing Annoy index for bigmatrix workflows

Usage

```
annoy_load_bigmatrix(
  path,
  metadata_path = NULL,
  prefault = FALSE,
  load_mode = "eager"
)
```

Arguments

path	File path to an existing Annoy index built by annoy_build_bigmatrix().
metadata_path	Optional path to the sidecar metadata file.
prefault	Logical flag indicating whether searches should prefault the index when loaded by the native backend.
load_mode	Whether to eagerly load the native index handle on open or defer until first search.

Value

A bigannoy_index object that can be passed to annoy_search_bigmatrix().

---

annoy_open_index *Open an existing Annoy index and its sidecar metadata*

Description

Open an existing Annoy index and its sidecar metadata

Usage

```
annoy_open_index(
  path,
  metadata_path = NULL,
  prefault = FALSE,
  load_mode = "eager"
)
```

Arguments

path	File path to an existing Annoy index built by annoy_build_bigmatrix().
metadata_path	Optional path to the sidecar metadata file.
prefault	Logical flag indicating whether searches should prefault the index when loaded by the native backend.
load_mode	Whether to eagerly load the native index handle on open or defer until first search.

Value

A bigannoy_index object that can be passed to annoy_search_bigmatrix().

---

annoy_search_bigmatrix *Search an Annoy index built from a* bigmemory::big.matrix

---

Description

Query a persisted Annoy index created by annoy_build_bigmatrix() or reopened with annoy_open_index(). Supply query = NULL for self-search over the indexed reference rows, or provide a dense numeric matrix, big.matrix, or external pointer for external-query search. Results can be returned in memory or streamed into destination big.matrix objects.

Usage

```
annoy_search_bigmatrix(
  index,
  query = NULL,
  k = 10L,
  search_k = -1L,
  xpIndex = NULL,
  xpDistance = NULL,
  prefault = NULL,
  block_size = annoy_default_block_size()
)
```

Arguments

index	A bigannoy_index returned by annoy_build_bigmatrix(), annoy_open_index(), or annoy_load_bigmatrix().
query	Optional query source. Supply NULL for self-search, another big.matrix or external pointer for streamed queries, or a dense numeric matrix.
k	Number of neighbours to return.
search_k	Annoy's runtime search budget. Use -1L for the library default.

`xpIndex` Optional writable `bigmemory::big.matrix` or external pointer receiving the 1-based neighbour indices.

`xpDistance` Optional writable `bigmemory::big.matrix` or external pointer receiving the Annoy distances. It may only be supplied when `xpIndex` is also provided.

`prefault` Optional logical override controlling whether the native backend prefaults the Annoy file while loading it for search.

`block_size` Number of queries processed per block.

Value

A list with components `index`, `distance`, `k`, `metric`, `n_ref`, `n_query`, `exact`, and `backend`.

`annoy_validate_index` *Validate a persisted Annoy index and its sidecar metadata*

Description

Validate a persisted Annoy index and its sidecar metadata

Usage

```
annoy_validate_index(index, strict = TRUE, load = TRUE, prefault = NULL)
```

Arguments

`index` A `bigannoy_index`.

`strict` Whether failed validation checks should raise an error.

`load` Whether to also verify that the index can be loaded successfully.

`prefault` Optional logical override used when `load = TRUE`.

Value

A list containing `valid`, `checks`, and the normalized `index`.

benchmark_annoy_bigmatrix

Benchmark a single bigANNOY build/search configuration

Description

Build or reuse a benchmark reference dataset, create an Annoy index, query it, and optionally compare recall against the exact bigKNN Euclidean baseline.

Usage

```
benchmark_annoy_bigmatrix(
  x = NULL,
  query = NULL,
  n_ref = 2000L,
  n_query = 200L,
  n_dim = 20L,
  k = 10L,
  n_trees = 50L,
  metric = "euclidean",
  search_k = -1L,
  seed = 42L,
  build_seed = seed,
  build_threads = -1L,
  block_size = annoy_default_block_size(),
  backend = getOption("bigANNOY.backend", "cpp"),
  exact = TRUE,
  filebacked = FALSE,
  path_dir = tempdir(),
  keep_files = FALSE,
  output_path = NULL,
  load_mode = "eager"
)
```

Arguments

x	Optional benchmark reference input. Supply NULL to generate a synthetic reference matrix, or provide a numeric matrix, big.matrix, descriptor, descriptor path, or external pointer.
query	Optional benchmark query input. Supply NULL for self-search, or provide a numeric matrix, big.matrix, descriptor, descriptor path, or external pointer.
n_ref	Number of synthetic reference rows to generate when x = NULL.
n_query	Number of synthetic query rows to generate when x = NULL and query is not NULL.
n_dim	Number of synthetic columns to generate when x = NULL.

`k` Number of neighbours to return.

`n_trees` Number of Annoy trees to build.

`metric` Annoy metric. One of `"euclidean"`, `"angular"`, `"manhattan"`, or `"dot"`.

`search_k` Annoy search budget.

`seed` Random seed used for synthetic data generation and, by default, for the Annoy build seed.

`build_seed` Optional Annoy build seed. Defaults to `seed`.

`build_threads` Native Annoy build-thread setting.

`block_size` Build/search block size.

`backend` Requested bigANNOY backend.

`exact` Logical flag controlling whether to benchmark the exact Euclidean baseline with `bigKNN` when available.

`filebacked` Logical flag; if `TRUE`, synthetic or dense reference inputs are converted into file-backed `big.matrix` objects before build.

`path_dir` Directory where temporary Annoy and optional file-backed benchmark files should be written.

`keep_files` Logical flag; if `TRUE`, leave the generated Annoy index on disk after the benchmark finishes.

`output_path` Optional CSV path where the benchmark summary should be written.

`load_mode` Whether the benchmarked index should be returned metadata-only until first search (`"lazy"`) or eagerly loaded once built (`"eager"`).

Value

A list with a one-row `summary` data frame plus the benchmark parameters and generated Annoy file paths.

`benchmark_annoy_recall_suite` *Benchmark a recall suite across multiple Annoy configurations*

Description

Run a grid of `n_trees` and `search_k` settings on the same benchmark dataset, optionally recording recall against the exact `bigKNN` Euclidean baseline.

Usage

```
benchmark_annoy_recall_suite(
  x = NULL,
  query = NULL,
  n_ref = 2000L,
  n_query = 200L,
  n_dim = 20L,
  k = 10L,
  n_trees = c(10L, 50L, 100L),
  search_k = c(-1L, 1000L, 5000L),
  metric = "euclidean",
  seed = 42L,
  build_seed = seed,
  build_threads = -1L,
  block_size = annoy_default_block_size(),
  backend = getOption("bigANNOY.backend", "cpp"),
  exact = TRUE,
  filebacked = FALSE,
  path_dir = tempdir(),
  keep_files = FALSE,
  output_path = NULL,
  load_mode = "eager"
)
```

Arguments

`x` Optional benchmark reference input. Supply `NULL` to generate a synthetic reference matrix, or provide a numeric matrix, `big.matrix`, descriptor, descriptor path, or external pointer.

`query` Optional benchmark query input. Supply `NULL` for self-search, or provide a numeric matrix, `big.matrix`, descriptor, descriptor path, or external pointer.

`n_ref` Number of synthetic reference rows to generate when `x = NULL`.

`n_query` Number of synthetic query rows to generate when `x = NULL` and `query` is not `NULL`.

`n_dim` Number of synthetic columns to generate when `x = NULL`.

`k` Number of neighbours to return.

`n_trees` Integer vector of Annoy tree counts to benchmark.

`search_k` Integer vector of Annoy search budgets to benchmark.

`metric` Annoy metric. One of `"euclidean"`, `"angular"`, `"manhattan"`, or `"dot"`.

`seed` Random seed used for synthetic data generation and, by default, for the Annoy build seed.

`build_seed` Optional Annoy build seed. Defaults to `seed`.

`build_threads` Native Annoy build-thread setting.

`block_size` Build/search block size.

`backend` Requested bigANNOY backend.

`exact` Logical flag controlling whether to benchmark the exact Euclidean baseline with `bigKNN` when available.

`filebacked` Logical flag; if `TRUE`, synthetic or dense reference inputs are converted into file-backed `big.matrix` objects before build.

`path_dir` Directory where temporary Annoy and optional file-backed benchmark files should be written.

`keep_files` Logical flag; if `TRUE`, leave the generated Annoy index on disk after the benchmark finishes.

`output_path` Optional CSV path where the benchmark summary should be written.

`load_mode` Whether the benchmarked index should be returned metadata-only until first search (`"lazy"`) or eagerly loaded once built (`"eager"`).

Value

A list with a `summary` data frame containing one row per `(n_trees, search_k)` configuration.

---

benchmark_annoy_volume_suite

Benchmark scaling across data volumes for bigANNOY and direct RcppAnnoy

---

Description

Run `benchmark_annoy_vs_rcppannoy()` over a grid of synthetic data sizes to study how build time, search time, and index size scale with data volume.

Usage

```
benchmark_annoy_volume_suite(
  n_ref = c(2000L, 5000L, 10000L),
  n_query = 200L,
  n_dim = c(20L, 50L),
  k = 10L,
  n_trees = 50L,
  metric = "euclidean",
  search_k = -1L,
  seed = 42L,
  build_seed = seed,
  build_threads = -1L,
  block_size = annoy_default_block_size(),
  backend = getOption("bigANNOY.backend", "cpp"),
  exact = FALSE,
  filebacked = FALSE,
  path_dir = tempdir(),
  keep_files = FALSE,
  output_path = NULL,
  load_mode = "eager"
)
```

Arguments

`n_ref` Integer vector of synthetic reference row counts.

`n_query` Integer vector of synthetic query row counts.

`n_dim` Integer vector of synthetic column counts.

`k` Number of neighbours to return.

`n_trees` Number of Annoy trees to build.

`metric` Annoy metric. One of `"euclidean"`, `"angular"`, `"manhattan"`, or `"dot"`.

`search_k` Annoy search budget.

`seed` Random seed used for synthetic data generation and, by default, for the Annoy build seed.

`build_seed` Optional Annoy build seed. Defaults to `seed`.

`build_threads` Native Annoy build-thread setting.

`block_size` Build/search block size.

`backend` Requested bigANNOY backend.

`exact` Logical flag controlling whether to benchmark the exact Euclidean baseline with `bigKNN` when available.

`filebacked` Logical flag; if `TRUE`, synthetic or dense reference inputs are converted into file-backed `big.matrix` objects before build.

`path_dir` Directory where temporary Annoy and optional file-backed benchmark files should be written.

`keep_files` Logical flag; if `TRUE`, leave the generated Annoy index on disk after the benchmark finishes.

`output_path` Optional CSV path where the benchmark summary should be written.

`load_mode` Whether the benchmarked index should be returned metadata-only until first search (`"lazy"`) or eagerly loaded once built (`"eager"`).

Value

A list with a `summary` data frame containing one row per implementation and data-volume combination.

benchmark_annoy_vs_rcppannoy *Benchmark bigANNOY against direct RcppAnnoy*

Description

Run the same Annoy build and search task through bigANNOY and through a direct dense RcppAnnoy baseline. The comparison reports both speed metrics and data-volume metrics such as reference bytes, query bytes, and generated index size.

Usage

```
benchmark_annoy_vs_rcppannoy(
  x = NULL,
  query = NULL,
  n_ref = 2000L,
  n_query = 200L,
  n_dim = 20L,
  k = 10L,
  n_trees = 50L,
  metric = "euclidean",
  search_k = -1L,
  seed = 42L,
  build_seed = seed,
  build_threads = -1L,
  block_size = annoy_default_block_size(),
  backend = getOption("bigANNOY.backend", "cpp"),
  exact = TRUE,
  filebacked = FALSE,
  path_dir = tempdir(),
  keep_files = FALSE,
  output_path = NULL,
  load_mode = "eager"
)
```

Arguments

x Optional benchmark reference input. Supply NULL to generate a synthetic reference matrix, or provide a numeric matrix, big.matrix, descriptor, descriptor path, or external pointer.

query Optional benchmark query input. Supply NULL for self-search, or provide a numeric matrix, big.matrix, descriptor, descriptor path, or external pointer.

n_ref Number of synthetic reference rows to generate when x = NULL.

n_query Number of synthetic query rows to generate when x = NULL and query is not NULL.

n_dim Number of synthetic columns to generate when x = NULL.

`k` Number of neighbours to return.

`n_trees` Number of Annoy trees to build.

`metric` Annoy metric. One of `"euclidean"`, `"angular"`, `"manhattan"`, or `"dot"`.

`search_k` Annoy search budget.

`seed` Random seed used for synthetic data generation and, by default, for the Annoy build seed.

`build_seed` Optional Annoy build seed. Defaults to `seed`.

`build_threads` Native Annoy build-thread setting.

`block_size` Build/search block size.

`backend` Requested bigANNOY backend.

`exact` Logical flag controlling whether to benchmark the exact Euclidean baseline with `bigKNN` when available.

`filebacked` Logical flag; if `TRUE`, synthetic or dense reference inputs are converted into file-backed `big.matrix` objects before build.

`path_dir` Directory where temporary Annoy and optional file-backed benchmark files should be written.

`keep_files` Logical flag; if `TRUE`, leave the generated Annoy index on disk after the benchmark finishes.

`output_path` Optional CSV path where the benchmark summary should be written.

`load_mode` Whether the benchmarked index should be returned metadata-only until first search (`"lazy"`) or eagerly loaded once built (`"eager"`).

Value

A list with a two-row `summary` data frame, one row for `bigANNOY` and one for direct `RcppAnnoy`, plus benchmark metadata and any validation report produced for the `bigANNOY` index.

`print.bigannoy_index` *Print a* `bigannoy_index`

Description

Print a `bigannoy_index`

Usage

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

Arguments

`x` A `bigannoy_index`.

`...` Unused.

Value

x, invisibly.

Index