

# Package ‘usearchlite’

February 12, 2026

**Title** Local Vector Search with 'USearch'

**Version** 0.1.0

**Description** A lightweight local vector index for approximate nearest neighbor (ANN) search using the vendored 'USearch' library Vardanian (2023) <[doi:10.5281/zenodo.7949416](https://doi.org/10.5281/zenodo.7949416)>. Provides a simple interface for adding vectors, searching for nearest neighbors, and persisting indexes to disk. Metadata filtering is performed in 'R' on the candidate set returned by the 'C++' layer.

**License** MIT + file LICENSE

**URL** <https://github.com/aljrigo/usearchlite>

**BugReports** <https://github.com/aljrigo/usearchlite/issues>

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**SystemRequirements** C++17

**NeedsCompilation** yes

**LinkingTo** cpp11

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Author** Alejandro Jiménez Rico [aut, cre]

**Maintainer** Alejandro Jiménez Rico <[aljrigo@gmail.com](mailto:aljrigo@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-02-12 20:30:02 UTC

## Contents

|                        |   |
|------------------------|---|
| index_add . . . . .    | 2 |
| index_meta . . . . .   | 2 |
| index_new . . . . .    | 3 |
| index_search . . . . . | 4 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>5</b> |
|--------------|----------|

---

|           |                                  |
|-----------|----------------------------------|
| index_add | <i>Add a Vector to the Index</i> |
|-----------|----------------------------------|

---

**Description**

Adds a vector with the given ID to the index, optionally with metadata.

**Usage**

```
index_add(index, id, vector, meta = NULL)
```

**Arguments**

|        |                                                          |
|--------|----------------------------------------------------------|
| index  | An usearchlite_index object.                             |
| id     | Integer. Unique identifier for the vector.               |
| vector | Numeric vector of length equal to the index dimension.   |
| meta   | Optional list of metadata to associate with this vector. |

**Value**

The index object (invisibly), for chaining.

**Examples**

```
tmp <- tempfile()
dir.create(tmp)
idx <- index_new(3L, tmp)
idx <- index_add(idx, 1L, c(1, 0, 0), meta = list(category = "a"))
unlink(tmp, recursive = TRUE)
```

---

|            |                           |
|------------|---------------------------|
| index_meta | <i>Get Index Metadata</i> |
|------------|---------------------------|

---

**Description**

Returns the metadata data.frame for all vectors in the index.

**Usage**

```
index_meta(index)
```

**Arguments**

|       |                              |
|-------|------------------------------|
| index | An usearchlite_index object. |
|-------|------------------------------|

**Value**

A data.frame with at least an 'id' column.

**Examples**

```
tmp <- tempfile()
dir.create(tmp)
idx <- index_new(3L, tmp)
idx <- index_add(idx, 1L, c(1, 0, 0), meta = list(name = "first"))
m <- index_meta(idx)
unlink(tmp, recursive = TRUE)
```

---

index\_new

---

*Create a New Vector Index*


---

**Description**

Creates or loads a vector index at the specified path.

**Usage**

```
index_new(dim, path)
```

**Arguments**

|      |                                                                                                                           |
|------|---------------------------------------------------------------------------------------------------------------------------|
| dim  | Integer. The dimensionality of vectors to be stored.                                                                      |
| path | Character. Directory path where the index will be stored. Will create index.usearch and meta.rds files in this directory. |

**Value**

An object of class usearchlite\_index containing the index state.

**Examples**

```
tmp <- tempfile()
dir.create(tmp)
idx <- index_new(3L, tmp)
unlink(tmp, recursive = TRUE)
```

---

|              |                                     |
|--------------|-------------------------------------|
| index_search | <i>Search for Nearest Neighbors</i> |
|--------------|-------------------------------------|

---

### Description

Searches the index for the k nearest neighbors of the query vector(s).

### Usage

```
index_search(index, query, k = 10L, filter = NULL, prefilter_k = 100L)
```

### Arguments

|             |                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| index       | An usearchlite_index object.                                                                                                               |
| query       | Numeric vector of length dim, or matrix with dim columns for batch queries.                                                                |
| k           | Integer. Number of nearest neighbors to return.                                                                                            |
| filter      | Optional function that takes the metadata data.frame and returns a logical vector indicating which rows to keep, or a filtered data.frame. |
| prefilter_k | Integer. Number of candidates to retrieve from the C++ layer before applying the filter. Should be >= k.                                   |

### Value

A list with components:

**ids** Integer vector (or matrix for batch) of neighbor IDs

**distances** Numeric vector (or matrix for batch) of distances

**meta** Data.frame of metadata for returned IDs

### Examples

```
tmp <- tempfile()
dir.create(tmp)
idx <- index_new(3L, tmp)
idx <- index_add(idx, 1L, c(1, 0, 0), meta = list(category = "a"))
idx <- index_add(idx, 2L, c(0, 1, 0), meta = list(category = "b"))
res <- index_search(idx, c(1, 0, 0), k = 2L)
res <- index_search(idx, c(1, 0, 0), k = 2L,
  filter = function(m) m$category == "a")
unlink(tmp, recursive = TRUE)
```

# Index

index\_add, [2](#)  
index\_meta, [2](#)  
index\_new, [3](#)  
index\_search, [4](#)