

# Package ‘fm.index’

April 8, 2022

**Title** Fast String Searching

**Version** 0.1.1

**Description** Wrapper for the Succinct Data Structure C++ library (SDSL v3) <<https://github.com/xxsds/sdsl-lite>> enabling fast string searching using FM indices. Partial string matching can be ~50-fold faster than simple string scans for many real-world string collections (corpora). A given corpus is converted into a compact in-memory FM index representation that can be efficiently queried for partial string matches.

**License** MIT + file LICENSE

**URL** <https://github.com/clemenshug/fm.index>

**BugReports** <https://github.com/clemenshug/fm.index/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**LinkingTo** Rcpp, stringi

**Imports** Rcpp, stringi

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** yes

**Author** Clemens Hug [aut, cre] (<<https://orcid.org/0000-0002-8299-3274>>)

**Maintainer** Clemens Hug <clemens.hug@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-04-08 09:12:29 UTC

## R topics documented:

fm_index_create	2
fm_index_locate	2
fm_index_save	3
<b>Index</b>	<b>5</b>

---

fm_index_create	<i>Construct new FM Index</i>
-----------------	-------------------------------

---

### Description

FM indices are data structures for memory efficient storage of large sets of strings (corpus). Searches for partial matches with the corpus are extremely fast.

### Usage

```
fm_index_create(strings, case_sensitive = FALSE)
```

### Arguments

`strings` Vector of strings (corpus) to construct FM index from  
`case_sensitive` Build case-sensitive index if TRUE

### Value

A FM Index object that can be passed to `fm_index_locate()` for finding matches in the corpus.

### See Also

Other FM Index functions: [fm\\_index\\_locate\(\)](#), [fm\\_index\\_save\(\)](#)

### Examples

```
data("state")
index <- fm_index_create(state.name, case_sensitive = FALSE)
```

---

fm_index_locate	<i>Locate given patterns</i>
-----------------	------------------------------

---

### Description

Finds all occurrences of all given patterns in the FM Index, analogous to [stringi::stri\\_locate\(\)](#) and [str\\_locate\(\)](#) from [stringr](#).

### Usage

```
fm_index_locate(patterns, index)
```

### Arguments

`patterns` Vector of strings to look for in the index  
`index` Index created with [fm\\_index\\_create\(\)](#)

**Value**

A data frame with three columns. `pattern_index` is the index of the query pattern, `corpus_index` is the index of the matching string in the corpus, and `position` is the starting position of the match within the corpus string. All indices are 1-based.

**See Also**

Other FM Index functions: [fm\\_index\\_create\(\)](#), [fm\\_index\\_save\(\)](#)

**Examples**

```
data("state")
index <- fm_index_create(state.name, case_sensitive = FALSE)
# Find all states with "new" in their names
hits <- fm_index_locate("new", index)
hits
# Show matching strings in library
state.name[hits$library_index]

hits <- fm_index_locate("ar", index)
hits
state.name[hits$library_index]
```

---

fm_index_save	<i>Save / load FM indices</i>
---------------	-------------------------------

---

**Description**

FM indices can be stored on disk and loaded into memory again in order to avoid re-computing the index every time a new R session is opened.

**Usage**

```
fm_index_save(index, path)
```

```
fm_index_load(path)
```

**Arguments**

<code>index</code>	FM Index to be saved to disk
<code>path</code>	Path where to save index to or load index from

**Value**

For `fm_index_load`, a FM Index object that can be passed to [fm\\_index\\_locate\(\)](#) for finding matches in the corpus.

For `fm_index_save`, no return value. Called for side-effects.

**Functions**

- `fm_index_save`: Save FM Index to disk
- `fm_index_load`: Load FM Index from disk

**See Also**

Other FM Index functions: [fm\\_index\\_create\(\)](#), [fm\\_index\\_locate\(\)](#)

**Examples**

```
data("state")
index_1 <- fm_index_create(state.name, case_sensitive = FALSE)

tmp_path <- tempfile()
fm_index_save(index_1, tmp_path)
index_2 <- fm_index_load(tmp_path)

identical(
  fm_index_locate("new", index_1),
  fm_index_locate("new", index_2)
)
```

# Index

## \* FM Index functions

fm\_index\_create, 2

fm\_index\_locate, 2

fm\_index\_save, 3

fm\_index\_create, 2, 3, 4

fm\_index\_create(), 2

fm\_index\_load (fm\_index\_save), 3

fm\_index\_locate, 2, 2, 4

fm\_index\_locate(), 2, 3

fm\_index\_save, 2, 3, 3

stringi::stri\_locate(), 2