

# Package ‘SMMT’

July 16, 2020

**Title** The Swiss Municipality Merger Tool Maps Municipalities Over Time

**Version** 1.0.2

**Description** In Switzerland, the landscape of municipalities is changing rapidly mainly due to mergers. The Swiss Municipality Merger Tool automatically detects these mutations and maps municipalities over time, i.e. municipalities of an old state to municipalities of a new state. This functionality is helpful when working with datasets that are based on different spatial references. The spatial reference in this context signifies a set of municipalities at a given point in time.

**Imports** dplyr, XML, tibble

**Suggests** testthat, roxygen2, knitr, rmarkdown

**VignetteBuilder** knitr

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Author** Valentin Knechtl [aut, cre] (<<https://orcid.org/0000-0002-8545-8883>>)

**Maintainer** Valentin Knechtl <[valentin.knechtl@unibas.ch](mailto:valentin.knechtl@unibas.ch)>

**Repository** CRAN

**Date/Publication** 2020-07-16 05:20:02 UTC

## R topics documented:

filter_date . . . . .	2
get_irreversible_municipality_mutations . . . . .	2
import_CH_municipality_inventory . . . . .	3
map_old_to_new_state . . . . .	4
smmt . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

filter_date	<i>Filter by date</i>
-------------	-----------------------

---

**Description**

Filter for existing municipalities at a specific point in time.

**Usage**

```
filter_date(tbl, date)
```

**Arguments**

tbl	A tibble
date	A POSIXct object

**Value**

A tibble which is a subset of tbl

---

get_irreversible_municipality_mutations	<i>Get irreversible municipality mutations</i>
---	--

---

**Description**

This function detects irreversible mutations.

**Usage**

```
get_irreversible_municipality_mutations(mutations)
```

**Arguments**

mutations	A tibble with municipality mutations (as created by <a href="#">import_CH_municipality_inventory</a> )
-----------	--

**Details**

Irreversible mutations are defined as mutations during which territories are split up. There are different types of irreversible mutations drawn from the below cited document. In contrast, normal mutations signify a simple merging of territory which accounts for most of the mutations in Switzerland since 1960 whereas irreversible mutations occurred only rarely. The aim of this function is to filter for these irreversible mutations. These can then be treated separately.

Definitions for different types of territory split ups are based on: Erläuterungen und Anwendungen - Historisierte Gemeindeverzeichnis der Schweiz (2017).

**Value**

A tibble with all the instances of irreversible mutations. The irreversibility cause is part of the output.

**Examples**

```
mutations <- structure(list(hist_id = c(11320L, 13668L, 13669L),
  district_hist_id = c(10024L, 10024L, 10024L),
  kanton_abbr = c("AG", "AG", "AG"),
  bfs_nr = c(4061L, 4061L, 4084L),
  name = c("Arni-Islisberg", "Arni (AG)", "Islisberg"),
  admission_nr = c(1000L, 1481L, 1481L),
  admission_mode = c(20L, 21L, 21L),
  admission_date = structure(c(-315619200, 410227200, 410227200),
  class = c("POSIXct", "POSIXt"), tzzone = ""),
  abolition_nr = c(1481L, NA, NA),
  abolition_mode = c(29L, NA, NA),
  abolition_date = structure(c(410140800, NA, NA),
  class = c("POSIXct", "POSIXt"), tzzone = ""),
  change_date = structure(c(410140800, 410227200, 410227200),
  class = c("POSIXct", "POSIXt"), tzzone = "")),
  row.names = c(NA, -3L), class = c("tbl_df", "tbl", "data.frame"))

irreversible_mutations <- get_irreversible_municipality_mutations(mutations)
```

---

```
import_CH_municipality_inventory
  Import the Swiss Municipality inventory
```

---

**Description**

This function imports the Swiss municipality inventory from the raw XML resource into R as a [tibble](#). The imported table is the basis to map the Swiss municipalities from an old to a new state (see [map\\_old\\_to\\_new\\_state](#)).

**Usage**

```
import_CH_municipality_inventory(file_path)
```

**Arguments**

file_path	Character vector of length one. It contains the file path to the Swiss municipality inventory XML file.
-----------	---

## Details

This imported Swiss municipality inventory is a database with the complete mutation history that occurred since 01.01.1960. The Swiss municipality inventory is made available by the Federal Statistical Office and updated regularly to keep track of new mutations.

### Download

See BfS webpage for infos about Swiss municipality inventory: [Historisiertes Gemeindeverzeichnis](#)

Direct download link: [Download XML](#)

## Value

A list with two tables in the form of tibble objects.

1. Municipality mutations.
2. Canton mutations

## See Also

[map\\_old\\_to\\_new\\_state](#)

---

map\_old\_to\_new\_state *Map municipalities of old state to municipalities of new state*

---

## Description

This function maps the Swiss municipalities of an old state to municipalities of a new state.

## Usage

```
map_old_to_new_state(mutations, state_old, state_new)
```

## Arguments

mutations	A tibble containing the municipality mutations inventory (see <a href="#">import_CH_municipality_inventory</a> )
state_old	A <a href="#">POSIXct</a> object vector of length one containing the date of the old state.
state_new	A <a href="#">POSIXct</a> object vector of length one containing the date of the new state.

## Details

### Approach

1. Download the [Swiss municipality inventory](#)
2. Import it into R workspace with [import\\_CH\\_municipality\\_inventory](#)
3. Set the old state and the new state (see example)
4. Get the mapping table with this function

**Example Daettwil / Baden**

On 1.1.1962 Daettwil (Bfs Nr. 4025) merged with Baden (Bfs Nr. 4021). Let's define

- `old_state <- as.POSIXct("1961-01-01")`
- `new_state <- as.POSIXct("1963-01-01")`
- Result:

bfs_nr_new	name_new	bfs_nr_old	name_old
4021	Baden	4021	Baden
4021	Baden	4025	Daettwil

**Value**

A list with 4 elements:

1. mapped: A tibble with the mapped municipalities
2. unmapped: A tibble with the unmapped municipalities
3. `state_old`: see above
4. `state_new`: see above

**Examples**

```
mutations <- structure(list(hist_id = c(11227L, 11240L, 13189L),
  district_hist_id = c(10025L, 10025L, 10025L),
  kanton_abbr = c("AG", "AG", "AG"),
  bfs_nr = c(4025L, 4021L, 4021L),
  name = c("Daettwil", "Baden", "Baden"),
  admission_nr = c(1000L, 1000L, 1004L),
  admission_mode = c(20L, 20L, 26L),
  admission_date = structure(c(-315619200, -315619200, -252460800),
    class = c("POSIXct", "POSIXt"), tzzone = ""),
  abolition_nr = c(1004L, 1004L, NA),
  abolition_mode = c(29L, 26L, NA),
  abolition_date = structure(c(-252547200, -252547200, NA),
    class = c("POSIXct", "POSIXt"), tzzone = ""),
  change_date = structure(c(-252547200,
    -252547200, -252460800), class = c("POSIXct", "POSIXt"), tzzone = "")),
  row.names = c(NA, -3L), class = c("tbl_df", "tbl", "data.frame"))
```

```
mapping_object <- map_old_to_new_state(mutations,
  as.POSIXct("1961-01-01"), as.POSIXct("1963-01-01"))
```

---

smtt

*SMMT - The Swiss Municipality Merger Tool Maps Municipalities Over Time*

---

### **Description**

In Switzerland, the landscape of municipalities is changing rapidly mainly due to mergers. The Swiss Municipality Merger Tool automatically detects these mutations and maps municipalities of an old state to municipalities of a new state. This functionality is helpful when working with datasets that are based on different spatial references. The spatial reference in this context signifies a set of municipalities at a given point in time.

### **Details**

For detailed information and examples, see [map\\_old\\_to\\_new\\_state](#)

# Index

`filter_date`, [2](#)

`get_irreversible_municipality_mutations`,  
[2](#)

`import_CH_municipality_inventory`, [2](#), [3](#),  
[4](#)

`map_old_to_new_state`, [3](#), [4](#), [4](#), [6](#)

`POSIXct`, [4](#)

`smtt`, [6](#)

`tibble`, [3](#)