

Package: m4ra (via r-universe)

November 4, 2024

Title Many-to-Many Multi-Modal Routing Aggregator

Version 0.1.1.041

Description Many-to-many multi-modal routing aggregator, combining 'dodgr' and 'gtfsrouter' for complete routing through urban networks.

License GPL-3

URL <https://github.com/UrbanAnalyst/m4ra>

BugReports <https://github.com/UrbanAnalyst/m4ra/issues>

Depends R (>= 2.10)

Imports checkmate, cli, digest, dodgr, fs, gtfsrouter, methods, memoise, parallel, rappdirs, Rcpp (>= 0.12.6), RcppParallel

Suggests dplyr, geodist, geosphere, hms, osmdata, osmextract, sf, testthat (>= 3.0.0)

LinkingTo Rcpp, RcppParallel, RcppThread

Encoding UTF-8

LazyData true

NeedsCompilation yes

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

SystemRequirements GNU make

Repository <https://urbananalyst.r-universe.dev>

RemoteUrl <https://github.com/UrbanAnalyst/m4ra>

RemoteRef HEAD

RemoteSha 65cde90e8edbbba61a284ca07e0c8eef4de4cefbfd

Contents

m4ra_batch_weight_networks	2
m4ra_bike_car_ratio_areas	3
m4ra_cache_network	4
m4ra_dists_n_pts	4
m4ra_gtfs_traveltimes	5
m4ra_hampi	6
m4ra_load_cached_network	6
m4ra_network_hash	7
m4ra_parking	8
m4ra_prepare_data	9
m4ra_times_bike_car	10
m4ra_times_mm_car	11
m4ra_times_multi_mode	13
m4ra_times_single_mode	14
m4ra_times_to_gtfs_stops	15
m4ra_vertices	16
m4ra_weight_networks	17
Index	18

m4ra_batch_weight_networks

Cache a directory full of street networks for routing queries

Description

This function runs of a directory which contain a number of **silicate** or sc-formatted street networks, generated with the `dodgr_streetnet_sc` function of the **dodgr** package. The function uses a default cache location specified by `rappdirs::user_cache_dir()`. This location can be overridden by specifying a local environment variable, "M4RA_CACHE_DIR".

Usage

```
m4ra_batch_weight_networks(net_dir, remove_these = NULL)
```

Arguments

<code>net_dir</code>	Path of local directory containing 'sc'-format street networks.
<code>remove_these</code>	Names of any 'sc'-format files which should not be converted into weighted network form.

Value

A character vector of local locations of cached versions of the variously weighted network representations used in the various routing functions.

See Also

Other cache: [m4ra_cache_network\(\)](#), [m4ra_load_cached_network\(\)](#), [m4ra_network_hash\(\)](#), [m4ra_weight_networks\(\)](#)

m4ra_bike_car_ratio_areas

Convert bicycle and automobile times into equivalent areas for a specified range of ratios of the two travel times.

Description

Convert bicycle and automobile times into equivalent areas for a specified range of ratios of the two travel times.

Usage

```
m4ra_bike_car_ratio_areas(bike_car_dat, ratio_lims = 1 + 0:10/10)
```

Arguments

bike_car_dat Result of [m4ra_times_bike_car](#) function for a specified vector of from points.

ratio_lims Vector of ratio limits used to calculate areas within which ratio of bicycle to automobile times is less than specified limits.

Value

A data.frame with one row for each from point used to calculate [m4ra_times_bike_car](#), and columns including the OSM "id" value and corresponding coordinates ("x" and "y"), followed by a series of columns, one for each specified value of **ratio_lims**, containing the areas in square kilometres over which bicycle travel times are within the specified ratio of car travel times.

See Also

Other analyses: [m4ra_times_bike_car\(\)](#), [m4ra_times_mm_car\(\)](#)

m4ra_cache_network	<i>Save a weighted street network in both full and contracted forms to local cache.</i>
--------------------	---

Description

Save a weighted street network in both full and contracted forms to local cache.

Usage

```
m4ra_cache_network(graph, city)
```

Arguments

graph	The full network to be cached.
city	Name of city; used to name cached network files.

See Also

Other cache: [m4ra_batch_weight_networks\(\)](#), [m4ra_load_cached_network\(\)](#), [m4ra_network_hash\(\)](#), [m4ra_weight_networks\(\)](#)

m4ra_dists_n_pts	<i>Calculate network distances to nearest 'n' points.</i>
------------------	---

Description

Calculate network distances to nearest 'n' points.

Usage

```
m4ra_dists_n_pts(graph, from = NULL, to = NULL, npts = 3L)
```

Arguments

graph	A <code>dodgr</code> network returned from the weight_streetnet function using a network obtained with the <code>osmdata</code> <code>osmdata_sc</code> function.
from	Vector of OpenStreetMap ID values from which route distances are to be calculated.
to	Vector of OpenStreetMap ID values to which nearest 'npts' distances are to be calculated.
npts	Number of points to which distances are to be calculated.

See Also

Other main: [m4ra_gtfs_traveltimes\(\)](#), [m4ra_times_multi_mode\(\)](#), [m4ra_times_single_mode\(\)](#)

m4ra_gtfs_traveltimes *Construct a travel time matrix between all pairs of stops in a 'GTFS' feed.*

Description

Construct a travel time matrix between all pairs of stops in a 'GTFS' feed.

Usage

```
m4ra_gtfs_traveltimes(
  gtfs,
  start_time_limits,
  day,
  from_stops = NULL,
  next_interval = TRUE,
  quiet = FALSE
)
```

Arguments

gtfs	A 'GTFS' feed extracted with the gtfsrouter function, 'extract_gtfs'.
start_time_limits	As for the 'gtfs_traveltimes' function of gtfsrouter , a vector of two values specifying the earliest and latest departure times from each station.
day	As for the 'gtfs_traveltimes' function of gtfsrouter , the day for which the matrix is to be calculated.
from_stops	If not NULL (default), calculate travel times only from the stops identified in this parameter. All values must be stops taken from <code>gtfs\$stops\$stop_id</code> .
next_interval	If TRUE, calculate time intervals to subsequent trips after identified fastest trips. These subsequent trips may not necessarily be as fast as the initial trips, and so this requires a second calculation of all travel times. Setting this parameter to TRUE will therefore generally double the calculation time. Note that intervals to subsequent trips may be negative where alternative connections with greater numbers of transfers leave earlier than initial, minimal-transfer trips.
quiet	If FALSE, display progress information on screen.

Value

A list of two or three integer matrices:

- "duration": The fastest travel times between all pairs of stops for the specified 'start_time_limits'; and
- "ntransfers": The corresponding numbers of transfers.
- (Only if 'next_interval = TRUE') "intervals": a matrix of intervals (in seconds) until the next fastest service after that corresponding to the times in the 'duration' item.

See Also

Other main: [m4ra_dists_n_pts\(\)](#), [m4ra_times_multi_mode\(\)](#), [m4ra_times_single_mode\(\)](#)

m4ra_hampi

m4ra_hampi

Description

A sample street network from the township of Hampi, Karnataka, India. The same as the `dodgr::hampi` dataset, but in `sc` rather than `sf` format.

Format

A **silicate** `sc` representation of the street network of Hampi.

Note

Can be re-created with the following commands.

Examples

```
## Not run:
hampi <- dodgr_streetnet_sc ("hampi india")

## End(Not run)
```

m4ra_load_cached_network

Load cached file for one city and mode

Description

Load cached file for one city and mode

Usage

```
m4ra_load_cached_network(
  city = NULL,
  mode = "foot",
  contracted = TRUE,
  filename = NULL
)
```

Arguments

city	City for which file is to be loaded.
mode	One of "foot", "bicycle", or "motorcar".
contracted	If TRUE, load the contracted version of the graph; otherwise load the full graph.
filename	Optional name of specific file to load.

Value

Previously cached, weighted streetnet for specified city and mode.

See Also

Other cache: [m4ra_batch_weight_networks\(\)](#), [m4ra_cache_network\(\)](#), [m4ra_network_hash\(\)](#), [m4ra_weight_networks\(\)](#)

m4ra_network_hash	<i>Return a unique hash for the input network.</i>
-------------------	--

Description

This can be used to identify cached versions of a given network, in particular to determine whether or not weighted versions of a given network have been previously cached.

Usage

```
m4ra_network_hash(graph)
```

Arguments

graph	A silicate , "SC", format object containing network data used to generate weighted street networks.
-------	--

Value

Single character value with unique hash of given network.

See Also

Other cache: [m4ra_batch_weight_networks\(\)](#), [m4ra_cache_network\(\)](#), [m4ra_load_cached_network\(\)](#), [m4ra_weight_networks\(\)](#)

m4ra_parking	<i>Get data on parking availability and building volumes to estimate time penalties for automobile travel.</i>
--------------	--

Description

Get data on parking availability and building volumes to estimate time penalties for automobile travel.

Usage

```
m4ra_parking(
  bb,
  city_name,
  mode = "motorcar",
  planet_file = NULL,
  dlim = 5000,
  k = 1000,
  quiet = FALSE
)
```

Arguments

bb	Bounding box of city for query to extract parking data.
city_name	Name of city used to name cached files.
mode	Mode of transport used to extract OSM node IDs at which to estimate relative parking availability (should generally be "motorcar").
planet_file	Optional file path to local .osm.pbf or .osm.bz2 file encompassing specified bounding box. If given, data are extracted with system-level calls to "osmium", which must be installed.
dlim	Distance limit in metres out to which contributions of parking and buildings will be aggregated.
k	With of exponential function used to weight contributions with distance, $\exp(-d / k)$ for distance d. Default value of 1000 metres decreases weight to 37% at 1km, 14% at 2km, and 0.6% at 5km.
quiet	If FALSE, display progress information on screen.

Value

A `data.frame` of the vertices of the (contracted) network, with additional columns quantifying number of parking spaces associated with each vertex, as well as the total volume of all surrounding buildings.

See Also

Other prep: [m4ra_prepare_data\(\)](#), [m4ra_times_to_gtfs_stops\(\)](#)

m4ra_prepare_data	<i>Prepare network and GTFS components for 'm4ra' queries.</i>
-------------------	--

Description

This is a meta-function that calls the following functions:

- [m4ra_weight_networks](#) to generate several versions of the input street network weighted for different kinds of transport.
- [m4ra_gtfs_traveltimes](#) to generate

It also identifies the closest GTFS stops to every network point. The function stores several differently-weighted version of the street networks in a local cache directory defined by the `user_cache_dir()` function of the **rappdirs** package, or by a local environment parameter, "M4RA_CACHE_DIR", if set.

Usage

```
m4ra_prepare_data(
  net_sc = NULL,
  gtfs = NULL,
  city_name = NULL,
  planet_file = NULL,
  day = NULL,
  start_time_limits = NULL,
  final_mode = "foot",
  fast = FALSE,
  n_closest = 10L,
  batch_size = NULL,
  parking = FALSE,
  quiet = FALSE
)
```

Arguments

<code>net_sc</code>	Local file path to a silicate , "SC", format object containing network data used to generate weighted street networks.
<code>gtfs</code>	Local file path to a gtfsrouter object saved via <code>saveRDS</code> , containing GTFS (General Transit Feed Specification) data for network represented in <code>net_sc</code> . This <code>.Rds</code> object may include additional timetable or transfer information in addition to data represented in original zip-format data of provided GTFS feed.
<code>city_name</code>	Name of city used to name cached files.
<code>planet_file</code>	Optional file path to local <code>.osm.pbf</code> or <code>.osm.bz2</code> file encompassing specified bounding box. If given, data are extracted with system-level calls to "osmium", which must be installed.

day	As for the 'gtfs_traveltimes' function of gtfsrouter , the day for which the matrix is to be calculated.
start_time_limits	As for the 'gtfs_traveltimes' function of gtfsrouter , a vector of two values specifying the earliest and latest departure times from each station.
final_mode	The mode of transport used for the final stage from GTFS stops to destination points.
fast	Values of TRUE generate potentially enormous matrices which may not fit in local memory. The default value of FALSE is generally safer, but calculation may take considerably longer.
n_closest	Final travel times to each destination point are calculated by tracing back times to this number of closest GTFS stops. Lower values will result in faster calculation times, yet with potentially inaccurate results.
batch_size	Default routine calculates full travel time matrices between all pairs of stops. For very large GTFS feeds, this matrix may not fit into memory, in which case this parameter can be set to a positive integer value to sequentially calculate only portions of the full matrix, before final re-assembly into single result.
parking	If TRUE, calculate local densities of parking availability and building volumes, and convert to a score used to calculate time penalties for automobile routing.
quiet	If FALSE, display progress information on screen.

See Also

Other prep: [m4ra_parking\(\)](#), [m4ra_times_to_gtfs_stops\(\)](#)

m4ra_times_bike_car	<i>Calculate relative times from a specified point by bicycle versus private automobile, along with (optionally) corresponding walking distances.</i>
---------------------	---

Description

This city expects weighted networks to have been generated with the [m4ra_batch_weight_networks](#) function, and for corresponding networks to exist in the m4ra cache directory for the specified city.

Usage

```
m4ra_times_bike_car(city = NULL, from = NULL, walk_dists = TRUE)
```

Arguments

city	City for which values are to be calculated
from	Vector of OSM ID values of vertices from which ratios are to be calculated. Typically obtained by loading one weighted network, and sampling or extracting vertices from the function m4ra_vertices .
walk_dists	If TRUE, also calculate equivalent walking distances.

Value

A data.frame of destination vertices, including Open Street Map ID values, and longitude and latitude values, and four variables:

- 'car_t' Times in seconds to travel with automobile.
- 'bike_t' Times in seconds to travel by bicycles.
- 'walk_d' Equivalent walking distance in kilometres.
- 'ratio' Ratio of bicycle to automobile travel times.

See Also

Other analyses: [m4ra_bike_car_ratio_areas\(\)](#), [m4ra_times_mm_car\(\)](#)

Examples

```
## Not run:
city <- "<city_name>"
net <- m4ra_load_cached_network (city = city, mode = "foot")
v <- m4ra_vertices (net, "<city_name>")
from <- sample (v$id, size = 10)
dat <- m4ra_bike_car_ratios (city = city, from = from)

## End(Not run)
```

m4ra_times_mm_car	<i>Calculate relative times from a specified point by multi-modal transport versus private automobile, along with (optionally) corresponding walking distances.</i>
-------------------	---

Description

This city expects weighted networks to have been generated with the [m4ra_batch_weight_networks](#) function, and for corresponding networks to exist in the m4ra cache directory for the specified city.

Usage

```
m4ra_times_mm_car(
  net_sc = NULL,
  gtfs = NULL,
  city_name = NULL,
  day = NULL,
  start_time_limits = NULL,
  initial_mode = "foot",
  final_mode = "foot",
  from = NULL,
  duration_max = NULL,
  fast = FALSE,
```

```

n_closest = 10L,
walk_dists = TRUE,
quiet = FALSE
)

```

Arguments

net_sc	Local file path to a silicate , "SC", format object containing network data used to generate weighted street networks.
gtfs	Local file path to a gtfsrouter object saved via saveRDS, containing GTFS (General Transit Feed Specification) data for network represented in net_sc. This .Rds object may include additional timetable or transfer information in addition to data represented in original zip-format data of provided GTFS feed.
city_name	Name of city used to name cached files.
day	As for the 'gtfs_traveltimes' function of gtfsrouter , the day for which the matrix is to be calculated.
start_time_limits	As for the 'gtfs_traveltimes' function of gtfsrouter , a vector of two values specifying the earliest and latest departure times from each station.
initial_mode	Initial mode of transport from origin points towards public transport stop.
final_mode	The mode of transport used for the final stage from GTFS stops to destination points.
from	Vector of OSM ID values of vertices from which ratios are to be calculated. Typically obtained by loading one weighted network, and sampling or extracting vertices from the function m4ra_vertices.
duration_max	If specified, only calculate times from each point to the nearest GTFS stops out to specified maximal duration in seconds. Values may be provided if overall travel times are only of interest out to some defined range or maximal value. Specifying values for this parameter can considerably reduce calculation times.
fast	Values of TRUE generate potentially enormous matrices which may not fit in local memory. The default value of FALSE is generally safer, but calculation may take considerably longer.
n_closest	Final travel times to each destination point are calculated by tracing back times to this number of closest GTFS stops. Lower values will result in faster calculation times, yet with potentially inaccurate results.
walk_dists	If TRUE, also calculate equivalent walking distances.
quiet	If FALSE, display progress information on screen.

Value

A data.frame of destination vertices, including Open Street Map ID values, and longitude and latitude values, and four variables:

- 'car_t' Times in seconds to travel with automobile.
- 'mm_t' Times in seconds for equivalent multi-modal transport.

- 'walk_d' Equivalent walking distance in kilometres.
- 'ratio' Ratio of multi-modal to automobile travel times.

See Also

Other analyses: [m4ra_bike_car_ratio_areas\(\)](#), [m4ra_times_bike_car\(\)](#)

Examples

```
## Not run:
city <- "<city_name>"
net <- m4ra_load_cached_network (city = city, mode = "foot")
v <- m4ra_vertices (net, "<city_name>")
from <- sample (v$id, size = 10)
dat <- m4ra_mm_car_ratios (city = city, from = from)

## End(Not run)
```

m4ra_times_multi_mode *Calculate matrix of pair-wise travel times between points using multiple modes of transport.*

Description

Calculate matrix of pair-wise travel times between points using multiple modes of transport.

Usage

```
m4ra_times_multi_mode(
  net_sc = NULL,
  gtfs = NULL,
  city_name = NULL,
  day = NULL,
  start_time_limits = NULL,
  initial_mode = "foot",
  final_mode = "foot",
  from = NULL,
  duration_max = NULL,
  fast = FALSE,
  n_closest = 10L,
  quiet = FALSE
)
```

Arguments

net_sc Local file path to a **silicate**, "SC", format object containing network data used to generate weighted street networks.

gtfs	Local file path to a gtfsrouter object saved via saveRDS, containing GTFS (General Transit Feed Specification) data for network represented in net_sc. This .Rds object may include additional timetable or transfer information in addition to data represented in original zip-format data of provided GTFS feed.
city_name	Name of city used to name cached files.
day	As for the 'gtfs_traveltimes' function of gtfsrouter , the day for which the matrix is to be calculated.
start_time_limits	As for the 'gtfs_traveltimes' function of gtfsrouter , a vector of two values specifying the earliest and latest departure times from each station.
initial_mode	Initial mode of transport from origin points towards public transport stop.
final_mode	The mode of transport used for the final stage from GTFS stops to destination points.
from	List of OSM vertex IDs from which to calculate total multi-modal travel times. These must be vertices from the largest connected component of the contracted graph.
duration_max	If specified, only calculate times from each point to the nearest GTFS stops out to specified maximal duration in seconds. Values may be provided if overall travel times are only of interest out to some defined range or maximal value. Specifying values for this parameter can considerably reduce calculation times.
fast	Values of TRUE generate potentially enormous matrices which may not fit in local memory. The default value of FALSE is generally safer, but calculation may take considerably longer.
n_closest	Final travel times to each destination point are calculated by tracing back times to this number of closest GTFS stops. Lower values will result in faster calculation times, yet with potentially inaccurate results.
quiet	If FALSE, display progress information on screen.

See Also

Other main: [m4ra_dists_n_pts\(\)](#), [m4ra_gtfs_traveltimes\(\)](#), [m4ra_times_single_mode\(\)](#)

m4ra_times_single_mode

Calculate matrix of pair-wise travel times between points using a single mode of transport.

Description

Calculate matrix of pair-wise travel times between points using a single mode of transport.

Usage

```
m4ra_times_single_mode(graph, from = NULL, to = NULL, path = NULL)
```

Arguments

graph	A <code>dodgr</code> network returned from the weight_streetnet function using a network obtained with the <code>osmdata</code> <code>osmdata_sc</code> function.
from	Vector or matrix of points from which route distances are to be calculated (see Notes)
to	Vector or matrix of points to which route distances are to be calculated (see Notes)
path	If specified, save individual travel time vectors for each 'from' point at that local directory. from and to values can be either two-column matrices or equivalent of longitude and latitude coordinates, or else single columns precisely matching node numbers or names given in <code>graph\$from</code> or <code>graph\$to</code> . If <code>to</code> is <code>NULL</code> , pairwise distances are calculated between all points specified in <code>from</code> . If both <code>from</code> and <code>to</code> are <code>NULL</code> , pairwise distances are calculated between all nodes in <code>graph</code> .

See Also

Other main: [m4ra_dists_n_pts\(\)](#), [m4ra_gtfs_traveltimes\(\)](#), [m4ra_times_multi_mode\(\)](#)

Examples

```
net <- dodgr::weight_streetnet (m4ra_hampi, wt_profile = "foot")
traveltimes <- m4ra_times_single_mode (net)
```

m4ra_times_to_gtfs_stops

Construct a travel time matrix to or from all stops in a 'GTFS' feed from or to to all points in a street network.

Description

Construct a travel time matrix to or from all stops in a 'GTFS' feed from or to to all points in a street network.

Usage

```
m4ra_times_to_gtfs_stops(graph, gtfs, city, from = NULL, graph_to_gtfs = TRUE)
```

Arguments

graph	A <code>dodgr</code> network returned from the weight_streetnet function using a network obtained with the <code>osmdata</code> <code>osmdata_sc</code> function.
gtfs	A 'GTFS' feed extracted with the <code>gtfsrouter</code> function, 'extract_gtfs'.
city	Name of city being analysed; used to name and extract cache files.

from	Vector or matrix of points from which route distances are to be calculated. If not given, times are calculated from all points in the network. Only has any effect if graph_to_gtfs is TRUE.
graph_to_gtfs	If TRUE, generate matrix of times from all network junctions in 'graph' (or all from points if specified) to each stop in the 'gtfs\$stops' table; otherwise generate matrix of times from all stops to all network junctions.

Value

An integer matrix of fastest travel times either between all 'gtfs' stops and all network points (for 'graph_to_gtfs = FALSE'), or the other way around (for 'graph_to_gtfs = TRUE').

See Also

Other prep: [m4ra_parking\(\)](#), [m4ra_prepare_data\(\)](#)

m4ra_vertices	<i>m4ra_vertices</i>
---------------	----------------------

Description

Extract vertices of graph, including spatial coordinates if included. Modified version of equivalent **dodgr** function that uses different hash comparisons. This version also does none of the checks implemented in the **dodgr** version, and assumes graphs remain unmodified throughout.

Usage

```
m4ra_vertices(graph, city)
```

Arguments

graph	A flat table of graph edges. Must contain columns labelled from and to, or start and stop. May also contain similarly labelled columns of spatial coordinates (for example from_x or stop_lon).
city	Name of city; used to name cached network files.

Value

A data.frame of vertices with unique numbers (n).

Note

Values of n are 0-indexed

Examples

```
graph <- dodgr::weight_streetnet (dodgr::hampi)
v <- m4ra_vertices (graph, "hampi")
```

m4ra_weight_networks *Cache weighted networks for routing query*

Description

Uses a default cache location specified by `rappdirs::user_cache_dir()`. This location can be over-ridden by specifying a local environment variable, "M4RA_CACHE_DIR". The "city" parameter is only used as a prefix for the cached networks.

Usage

```
m4ra_weight_networks(net, city, quiet = TRUE)
```

Arguments

net	A silicate , "SC", format object containing network data used to generate weighted street networks.
city	Name of city; only used to name cached network files.
quiet	If FALSE, display progress information on screen.

Value

A character vector of local locations of cached versions of the variously weighted network representations used in the various routing functions.

See Also

Other cache: [m4ra_batch_weight_networks\(\)](#), [m4ra_cache_network\(\)](#), [m4ra_load_cached_network\(\)](#), [m4ra_network_hash\(\)](#)

Index

- * **analyses**
 - m4ra_bike_car_ratio_areas, 3
 - m4ra_times_bike_car, 10
 - m4ra_times_mm_car, 11
 - * **cache**
 - m4ra_batch_weight_networks, 2
 - m4ra_cache_network, 4
 - m4ra_load_cached_network, 6
 - m4ra_network_hash, 7
 - m4ra_weight_networks, 17
 - * **datasets**
 - m4ra_hampi, 6
 - * **data**
 - m4ra_hampi, 6
 - * **main**
 - m4ra_dists_n_pts, 4
 - m4ra_gtfs_traveltimes, 5
 - m4ra_times_multi_mode, 13
 - m4ra_times_single_mode, 14
 - * **misc**
 - m4ra_vertices, 16
 - * **prep**
 - m4ra_parking, 8
 - m4ra_prepare_data, 9
 - m4ra_times_to_gtfs_stops, 15
- m4ra_batch_weight_networks, 2, 4, 7, 10, 11, 17
- m4ra_bike_car_ratio_areas, 3, 11, 13
- m4ra_cache_network, 3, 4, 7, 17
- m4ra_dists_n_pts, 4, 6, 14, 15
- m4ra_gtfs_traveltimes, 4, 5, 9, 14, 15
- m4ra_hampi, 6
- m4ra_load_cached_network, 3, 4, 6, 7, 17
- m4ra_network_hash, 3, 4, 7, 7, 17
- m4ra_parking, 8, 10, 16
- m4ra_prepare_data, 8, 9, 16
- m4ra_times_bike_car, 3, 10, 13
- m4ra_times_mm_car, 3, 11, 11
- m4ra_times_multi_mode, 4, 6, 13, 15
- m4ra_times_single_mode, 4, 6, 14, 14
- m4ra_times_to_gtfs_stops, 8, 10, 15
- m4ra_vertices, 16
- m4ra_weight_networks, 3, 4, 7, 9, 17
- weight_streetnet, 4, 15