

OpenTripPlanner for R

27 Nov 2019

Summary

opentripplanner provides functions that enable multi-modal routing in R. It provides an interface to the OpenTripPlanner (OTP) Java routing service, which allows calculation of routes on large transport networks, locally or via calls to a remote server. The package contains three groups of functions for: (1) setting up and managing a local instance of OTP; (2) connecting to OTP locally or remotely; and (3) sending requests to the OTP API and importing the results and converting them into appropriate classes for further analysis.

Motivation

Routing, the process of calculating paths between points in geographic space that follow a transport network, is a fundamental part of transport planning and vital to solving many real-world transport problems. The outputs of a routing service, which can calculate many routes over a large area, comprises coordinates and other data representing a movement that is in some way ‘optimal’, based on a range of criteria (that the user should understand and be able to change). Routing services, such as the one provided by Google Maps, are a well-known and increasingly vital component of personal travel planning for many people (Bast et al. 2010). Less well-known, but perhaps equally important, is that routing services are also key to understanding aggregate travel patterns and guiding policy and commercial decisions (Giusti et al. 2018). To meet this need for route planning capabilities, a wide range of both proprietary and open source tools have been created.

Functionality

OpenTripPlanner (OTP) is written in Java and designed to work with Open Street Map (OSM) data for road-based modes (Car, Bike, Walking) and General Transit Feed Specification (GTFS) data for public transit (Bus, Train, Metro). OTP is unusual among open source routing tools in its ability to account for a wide range of modes of travel, e.g. bicycle hire, and support for complex multi-stage multi-modal journeys such as park and ride. However, OTP’s primary purpose is to support public-facing websites such as TriMet; thus its analytical capabilities are limited. Conversely, the R language is well suited to statistical and spatial analysis but has no route planning capabilities.

The OpenTripPlanner for R package aims to bridge the gap between OTP and R by supplying simple ways for R to connect to OTP either on a local machine or on a remote server, via OTP’s API. The package has been designed to ease bulk routing by allowing the input of multiple origins and destinations as two column matrices of longitude-latitude pairs. The package also supports multi-core operation to take advantage of OTP’s multicore functionality. Results are returned in the widely used **sf** data frame format. Although performance is dependant on the size of the map being routed over, it typically can achieve more than 10 routes per second per core.

The package has been developed from a set of R functions that formed part of an intermediate-level OTP tutorial as part of research at Centre for Research into Energy Demand Solutions and the Institute of Transport Studies.

Reproducible demonstration

Example data for the Isle of Wight, UK is provided with the package. The example below uses this data to demonstrate the basic functionality of the package. A full explanation is provided in the package vignettes. First, download the demo data and OTP.

```

library(opentripplanner)           # Load Package
path_data <- file.path(tempdir(), "OTP") # Create folder for data
dir.create(path_data)
path_otp <- otp_dl_jar(path_data)    # Download OTP
otp_dl_demo(path_data)              # Download demo data

```

Second, build the OTP graph, start up OTP server and connect to the server

```

log <- otp_build_graph(otp = path_otp,      # Build Graph
                      dir = path_data)
otp_setup(otp = path_otp, dir = path_data) # Start OTP
otpcon <- otp_connect()                   # Connect R to OTP

```

Finally, find routes

```

route <- otp_plan(otpcon,
                  fromPlace = c(-1.17502, 50.64590),
                  toPlace = c(-1.15339, 50.72266))

```

References

Bast, Hannah, Erik Carlsson, Arno Eigenwillig, Robert Geisberger, Chris Harrelson, Veselin Raychev, and Fabien Viger. 2010. “Fast Routing in Very Large Public Transportation Networks Using Transfer Patterns.” In *Algorithms - ESA 2010*, edited by Mark de Berg and Ulrich Meyer, 290–301. Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/978-3-642-15775-2_25.

Giusti, Riccardo, Daniele Manerba, Guido Perboli, Roberto Tadei, and Shuai Yuan. 2018. “A New Open-Source System for Strategic Freight Logistics Planning: The SYNCHRO-NET Optimization Tools.” *Transportation Research Procedia*, EURO Mini Conference on "Advances in Freight Transportation and Logistics", 30 (January): 245–54. doi:10.1016/j.trpro.2018.09.027.