On the Horizon
Using OpenStreetMap Data with Open-Source GIS

Michael Markieta | michael@spatialanalysis.ca

GIS Consultant
Toronto, Ontario, Canada
www.spatialanalysis.ca

INTRODUCTION

For many, free and open-source data and software represents accessibility to otherwise inaccessible geospatial workflows in terms of cost and availability. Commercial data used in geographic information systems (GIS) is available through a relatively small number of merchants or vendors, which produce highly accurate, precise, and detailed information. This is produced, however, at a cost that many small and large businesses, private consultants, and startups cannot afford. Open-source data, such as the volunteer geographic information on OpenStreetMap (OSM), represents a community effort to build one of the best web maps, and subsequently the best GIS database, available for free to the public. OpenStreetMap is a web-based map to which any registered user can submit data. These updates, over time, populate the now extensive web map that is the OpenStreetMap. At the same time, the data that lives on the OpenStreetMap can be downloaded and used inside of a GIS for geospatial analysis, cartographic rendering, and other geo-related tasks.

There are various workflows for extracting and consuming the data that is made available by the OpenStreetMap project. One of these methods is outlined in this tutorial. This tutorial will take Mac OSX users through a typical setup of a local PostgreSQL database, downloading and parsing raw OpenStreetMap data, and querying the database to extract data for use in QGIS, an open-source GIS package. Upon completing this tutorial, users will have hit the ground running, with the ability to run spatial queries—such as locating all the coffee shops within 500 metres of a subway station—or building cartographically pleasing reference map books with data that is of interest to the map reader.

THIS TUTORIAL WILL COVER THE FOLLOWING:

- Access and browse an OSM data repositories
- Download a subset (often called an "extract") of the planet.osm data package
- Install PostgreSQL (object-relational database system)
- Install PostGIS for use with PostgreSQL (spatial database extension for PostgreSQL)
- Install and utilize osm2pgsql (converts OSM data for use in the PostgreSQL database)
- Install QGIS and its dependencies (GIS package)
- Query and add data in QGIS from PostGIS/PostgreSQL

ALSO, THIS TUTORIAL MAKES THE FOLLOWING ASSUMPTIONS:

- Currently running Mac OS X 10.6 or later (tutorial verified using both 10.6.7 and 10.7)
- Access to administrative privileges so that we can perform installations
- Plenty of storage space (extracts from OSM range from a few MB to many GBs)
- Willingness to try something new!

LASTLY, THIS TUTORIAL WILL FOLLOW THE FOLLOWING FORMAT:

The tutorial will follow a guided, step-by-step instruction that will assume the role of a new user installing and processing all data from scratch. The included links and information above are for background to the project that we are about to begin. Please follow all the instructions (don't skip any steps if you don't know what you are doing) and download all the required data when prompted to.

We will be using Terminal in this guide but it is not expected, nor is it required, that any readers of this tutorial have any Terminal.app background or knowledge. Terminal allows users to interact with the computer through a command-line interface. If you have not seen or used the Terminal before, you may have come across some instance of command-line interfaces (perhaps on

Windows machines, à la Command Prompt). I will do my best to explain what we are doing during the phase of the tutorial which utilizes the Terminal.app.

If you would like more information about any of the items mentioned above, please feel free to visit the respective website/wiki as listed below (but remember to come back!):

OpenStreetMap **http://www.openstreetmap.org/**
Planet.osm **http://wiki.openstreetmap.org/wiki/Planet.osm**
PostgreSQL **http://en.wikipedia.org/wiki/PostgreSQL**
PostGIS **http://en.wikipedia.org/wiki/PostGIS**
Osm2pgsql **http://wiki.openstreetmap.org/wiki/Osm2pgsql**
QGIS **http://www.qgis.org/wiki/Welcome_to_the_QGIS_Wiki**

OPENSTREETMAP & THE REPOSITORIES

OpenStreetMap (OSM) is a collection of free geographic data that can be viewed within a browser (**http://www.openstreetmap.org/**). We can also access, download, and utilize the underlying OSM project data from various repositories. What makes this so attractive is the fact that it's a community driven project, which means that anyone can contribute to it. It is also free to use and distribute under the CC-BY-SA license (as long as we attribute OSM and the license itself). The standard package that OSM distributes is called the "planet.osm." It is a standard XML-formatted .osm file of global data which, at the time of writing (Aug 22, 2011), is over 220 GB (17 GB compressed). The planet.osm is updated weekly (every Thursday) and includes the latest revisions of nodes, ways and polygons (points, lines, and polygons). I strongly encourage you NOT to download the entire planet.osm and work with it, due to its size and the required computing power needed to work with such a database. We will be exploring an extract of the planet.osm—specifically the province of Ontario, Canada—which comes in a much smaller/cpu-friendly size (2 GB uncompressed/350 MB compressed).

http://wiki.openstreetmap.org/wiki/Planet.osm provides a list of mirrors from which the planet.osm file and its extracts can be downloaded. We will specifically be using the CloudMade directory located on **http://downloads.cloudmade.com/**

PREFACE

This tutorial will reflect what I am doing on my computer. This is done in an effort to address commonly made mistakes. Tailor the tutorial to suit your own needs or follow my instructions to a tee.

Let us make sure that we stay somewhat organized during this tutorial.

- Create a new folder on the desktop called "osm_tutorial"
- Create two subfolders within "osm_tutorial" called "downloads" and "data"

We will download all of our installation files to the “downloads” directory and we will download the OSM data to the “data” directory.

ACCESS, BROWSE, & DOWNLOAD OSM DATA

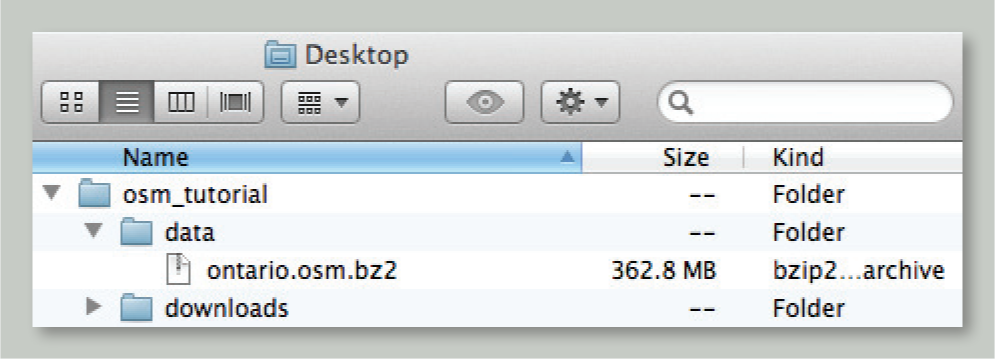
Let’s download an extract of the planet.osm file (remember that the planet.osm file is too large to handle on its own). We will use the CloudMade repository which updates their planet.osm and “extract.”osm (such that “extract” is the name of a location/place) weekly.

Go to **http://downloads.cloudmade.com/**

The repository is organized in a hierarchal structure (Region > SubRegion > Country > Province or State). Note that CloudMade does not have the entire planet.osm parsed into smaller extracts, such that only the most popular or demanded areas have been extracted for us to use. Some repositories will extract different regions than others. There are also some repositories that extract smaller scale areas (Cities, Towns, etc).

Here we will use the Ontario extract, which is located in Americas > Northern America > Canada > Ontario. Feel free to use any of the other extracts. However, I recommend that you choose an extract from the lowest level in the hierarchy (e.g., provinces or cities), and stay away from the larger extracts such as regions or countries.

- Follow the folder structure “Americas > Northern America > Canada > Ontario” to find the ontario.osm.bz2 extract.
- Right click > Save Link As on the “ontario.osm.bz2” link.
- Save the file to your “/Desktop/osm_tutorial/data” folder.



Name	Size	Kind
▼ osm_tutorial	--	Folder
▼ data	--	Folder
ontario.osm.bz2	362.8 MB	bzip2...archive
▶ downloads	--	Folder

Do not extract this file. A benefit of using the tools mentioned in this tutorial, is the ability to work with fully compressed OSM data. Why is the file named “ontario.osm.bz2”? BZIP2 (bz2) is an open-source compression tool that OSMutilizes to produce small production packages (eg. a 17 GB “planet.osm.bz2” v. 220 GB uncompressed “planet.osm”).

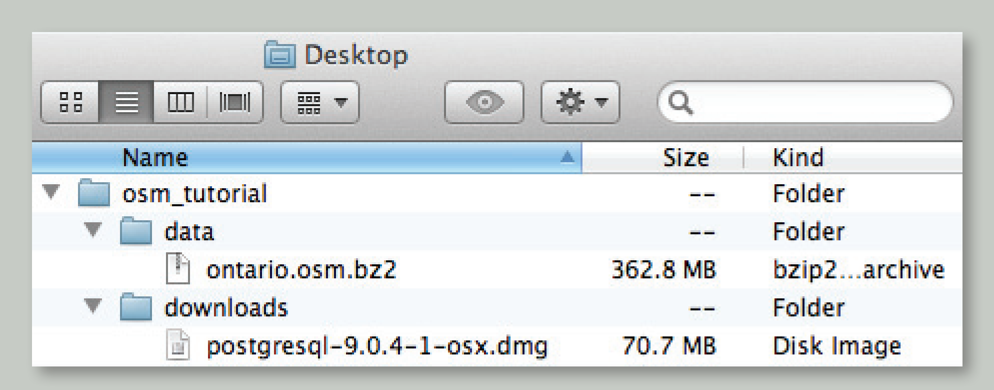
DOWNLOAD AND INSTALL POSTGRESQL

The next step is to download our database client, PostgreSQL. PostgreSQL is an enterprise-level database that is able to scale efficiently upon the demands of a single user or multiple users. By enabling spatial data storage with the PostGIS extension, PostgreSQL can store a wide variety of geometric objects, such as point, line polygons, multipoint, multiline, multipolygon, and geometric collections. Spatial databases empower the user by providing spatial functions, such as calculating the number of features within a specified radius of a point, or identifying the distance between two objects on the surface of the earth. PostGIS can also handle the reprojection of data as it is retrieved from the database, such that data can be stored in one common projection, but retrieved in a user-specified coordinate system.

The ESRI shapefile datatype is a common format for GIS data storage; however, with the amount of data provided by OpenStreetMap, it is unlikely that the shapefile specification can be used effectively. For example, the linestrings (polylines in ESRI-speak) table in our OpenStreetMap database contains all of the roads, paths, footways, creeks, rivers and so on. To effectively work with this data in an ESRI shapefile datatype, the features would require separation based on thematic content (roadways, pedestrian-ways, natural). However, while the data is in our PostgreSQL database, we can easily create three separate queries on our data to extract the same thematic content. This saves both space and time, as our data can be precisely extracted from one source database, based on the tagging system used in OpenStreetMap (**http://wiki.openstreetmap.org/wiki/Map_Features**), as opposed to producing three separate feature classes in the ESRI shapefile format. Lastly, it is important to note that queries on our database do not make changes to the data itself, and therefore can be easily modified if the user if the query results are not as expected.

There are many iterations of PostgreSQL and it can get quite confusing for the beginner user. Luckily, Dave Page at Enterprise DB maintains an easy-to-use, all-in-one installer that is available for Mac OS X (and other platforms as well).

- Go to **http://www.enterprisedb.com/products-services-training/pgdownload#osx**
- Select "Installer version Version 9.1.4" (latest at the time of writing) for Mac OS X.
- Save the file to your "/Desktop/osm_tutorial/downloads" folder.



Name	Size	Kind
osm_tutorial	--	Folder
data	--	Folder
ontario.osm.bz2	362.8 MB	bzip2...archive
downloads	--	Folder
postgresql-9.0.4-1-osx.dmg	70.7 MB	Disk Image

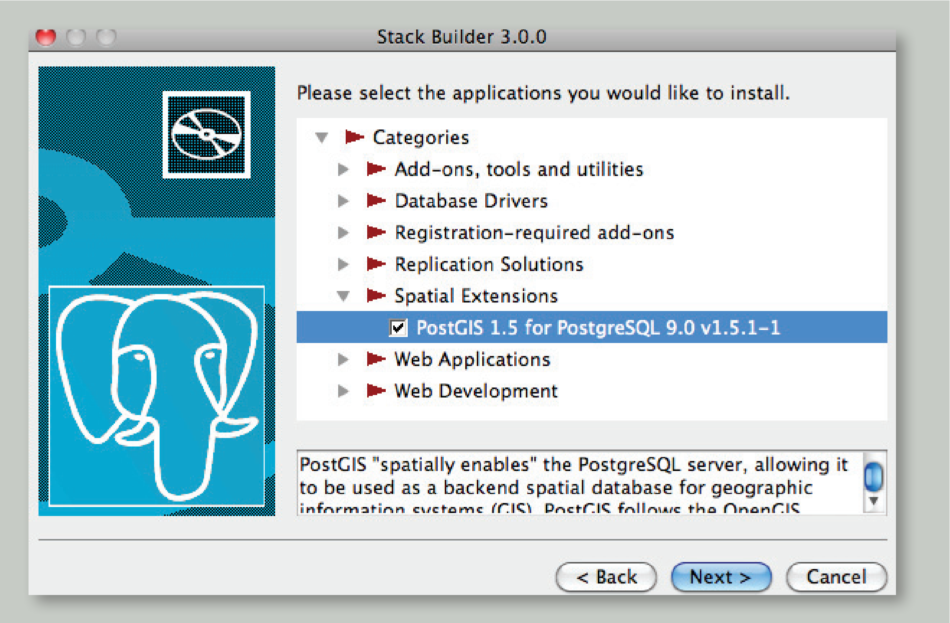
We will now try to install PostgreSQL. If this is your first attempt at installing PostgreSQL, you will be prompted with regard to your computer's "Shared Memory" configuration. Not to worry: PostgreSQL handles the changes that are necessary, and yes, it is safe to allow PostgreSQL make these changes. I've included a snippet from the PostgreSQL "readme."

SHARED MEMORY

PostgreSQL uses shared memory extensively for caching and inter-process communication. Unfortunately, the default configuration of Mac OS X does not allow suitable amounts of shared memory to be created to run the database server.

- Mount the PostgreSQL disk image by double-clicking on "postgresql-9.0.4–1-osx.dmg"
- Launch the installer and allow it to make the necessary changes.
- You will need to restart your computer.
- After restarting, re-mount the PostgreSQL disk image by double-clicking on "postgresql-9.0.4–1-osx.dmg" (if it is not already mounted on the desktop).
- The installer will now pass the shared memory check and load the setup wizard.
- Proceed through the setup wizard, accepting all default configuration settings. Remember to assign a unique password for your PostgreSQL database when prompted.
- Make special note of port number "5432" that PostgreSQL has been assigned to. We will need to call upon this number later on in this tutorial.

The installer will take a minute or so to complete, and will then ask if you would like to "Launch Stack Builder at exit?" This is required; we will need to install the PostGIS extension because PostgreSQL cannot handle our OSM data on its own. PostGIS will act as our database's forerunner, handling the spatial information in the OSM data for our PostgreSQL database.



Please select the applications you would like to install.

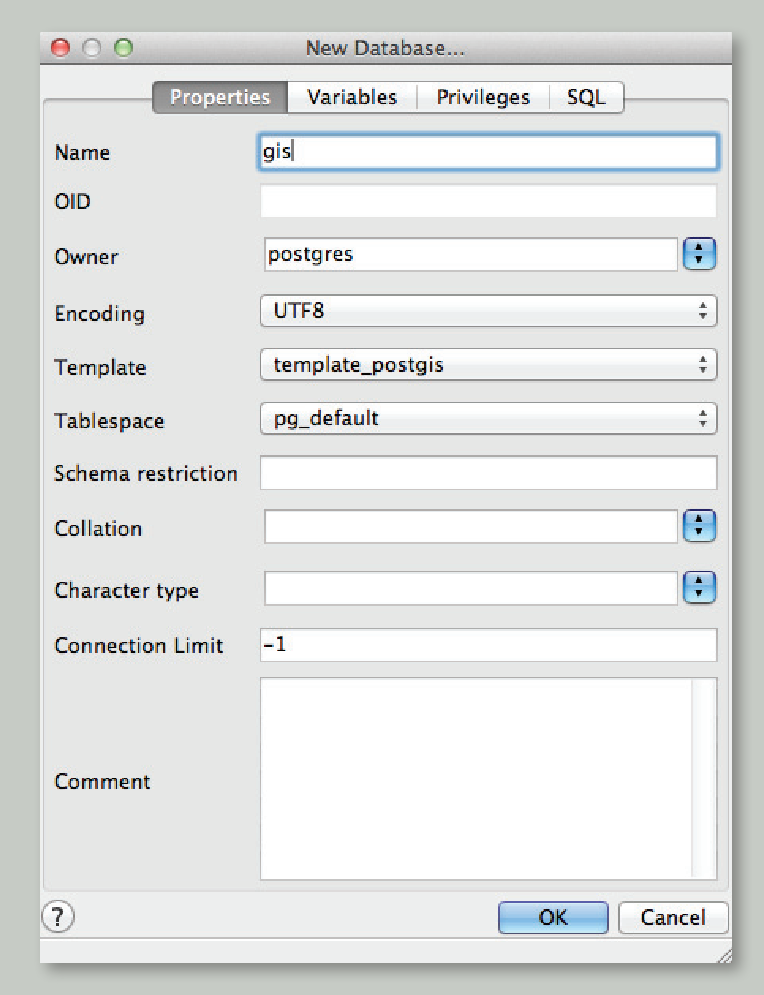
- Categories
  - Add-ons, tools and utilities
  - Database Drivers
  - Registration-required add-ons
  - Replication Solutions
  - Spatial Extensions
    - ☑ PostGIS 1.5 for PostgreSQL 9.0 v1.5.1-1
  - Web Applications
  - Web Development

PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS). PostGIS follows the OpenGIS

- Make sure the checkbox to launch stack builder on exit is ticked.
- When stack builder launches, it will ask you to select your PostgreSQL installation from the drop-down style menu.
- Select "Spatial Extensions > PostGIS" (as seen to right)
- Proceed through the rest of the installation, accepting all default configuration options.
- When prompted to, input your unique password for the PostgreSQL database. By default, the port number (5432)

and username (postgres) are correct

- Once the installation is completed, launch “pgAdmin III” from the “Applications” folder.
- Double-clicking on “PostgreSQL 9.0 (localhost:5432)” will start the database (located under “Object browser > Servers” in the left pane).
- Right click > New Database… in the central pane (making sure not to select the “postgres” or “template_postgis” items).
- Fill out the “New Database…” form with the settings as shown below.
- That’s it; we can exit the application.



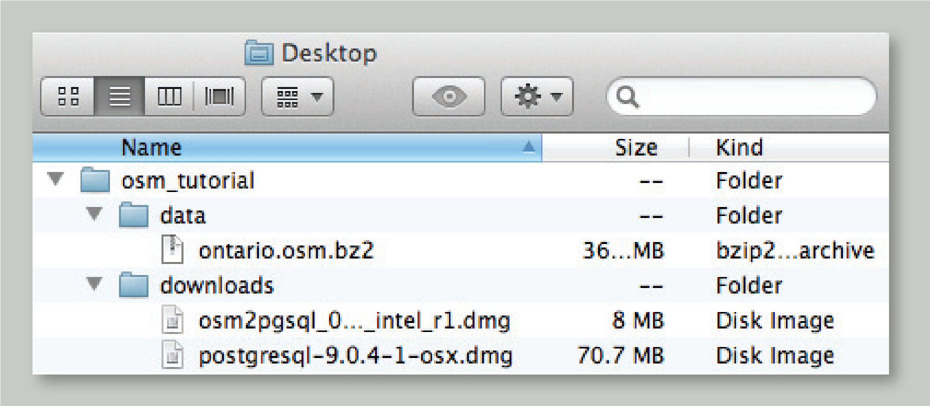
Name: gis
OID:
Owner: postgres
Encoding: UTF8
Template: template_postgis
Tablespace: pg_default
Schema restriction:
Collation:
Character type:
Connection Limit: -1
Comment:

DOWNLOAD AND INSTALL OSM2PGSQL

Our next step is to download and install osm2pgsql, which will expedite the “ontario.osm.bz2” file into our PostgreSQL database.

- Go to http://dbsgeo.com/downloads/#osm2pgsql
- Select and download “osm2pgsql 0.70.5 — Snow Leopard / OS 10.6 (Intel x86_64)” to your “/Desktop/osm_tutorial/downloads” folder.

- Mount the osm2pgsql diskimage by double-clicking "osm2pgsql_0.70.5_snow_intel_r1.dmg."
- You must install GEOS and PROJ frameworks before installing the osm2pgsql tool. Do this now. Proceed through the installations, accepting all default configurations.
- Once GEOS and PROJ are installed, we can move on to osm2pgsql. Accept all default configurations when installing osm2pgsql.



Name	Size	Kind
osm_tutorial	--	Folder
data	--	Folder
ontario.osm.bz2	36...MB	bzip2...archive
downloads	--	Folder
osm2pgsql_0..._intel_r1.dmg	8 MB	Disk Image
postgresql-9.0.4-1-osx.dmg	70.7 MB	Disk Image

USING OSM2PGSQL AND THE TERMINAL.APP

We will use osm2pgsql to parse our OSM data into the PostgreSQL database. Here is where we will encounter the use of the Terminal.app. As I mentioned earlier, I will try to explain what exactly we are telling Terminal to do.

- Launch Terminal. You can use the spotlight (magnify glass in the top right-hand corner) to search for it.

The first line of code—if you haven't opened terminal recently—should say something similar to this (replace my computer name and username with your own). Terminal is letting us know what {whose} computer we are on (michael-markietas-mac), and where are we performing the tasks (a directory; in this case, michael markieta's home directory)

```
Last login: Mon Aug 22 19:38:44 on console
Michael-Markietas-MacBook-Pro:~ michaelmarkieta$
```

We need to change the working directory from our user directory (michaelmarkieta) to the desktop. We use the "cd" command, which intuitively means "change directory." We also pass in the location to which we would like to change our directory. In this case, we will change directory to the "desktop." Type the following into Terminal (or copy and paste):

```
cd desktop
```

If done correctly, Terminal will switch the current working directory to the desktop and the repeating line of code should read something like this:

```
michael-markietas-mac:~desktop$>
```

Now that we know how to change working directories, let’s navigate to our ontario.osm.bz2 file, which is located in “desktop/osm_tutorial/data.” To move more than one folder when changing directories, we must insert our path between quotation marks and separate folders with forward slashes (/). Type the following into Terminal (or copy and paste):

- `cd “osm_tutorial/data”`

Terminal should now be working from within the “data” folder, which is located inside the “osm_tutorial” folder, which is also located inside {on} the “desktop”, which is also part of michael markieta’s user folder. The full path would look something like “/Users/Michael Markieta/Desktop/osm_tutorial/data.”

- `michael-markietas-mac:~data$>`

Now to perform some osm2pgl magic. The tool offers many configurable parameters. These can be seen by typing the following into Terminal:

- `osm2pgsql --help`

We will concern ourselves with a few of these parameters, but it’s useful to look over what is included with a tool when you install something new. We will be using –U, username, –d, database name to guide osm2pgsql in parsing the ontario.osm.bz2 file into our PostgreSQL database. The code will need to look like this:

- `osm2pgsql -d gis -U postgres -P 5432 ontario.osm.bz2`

LET’S PICK THIS APART ONE CHUNK AT A TIME.

- `osm2pgsql` — in words, we are telling Terminal.app to start the osm2pgsql script
- `-d gis` — in words, we are configuring the script to use the database called “gis”
- `-U postgres` — in words, we are configuring the script to access the database with username “postgres”
- `-P 5432` — in words, we are configuring the script to access the database on port 5432
- `ontario.osm.bz2` — in words, this is the file we would like our script to parse into our database

We will need to let it run for a few minutes. If everything went smoothly, you should have received the following confirmation in Terminal:

- `Completed planet_osm_point`
- `Completed planet_osm_roads`
- `Completed planet_osm_polygon`
- `Completed planet_osm_line`

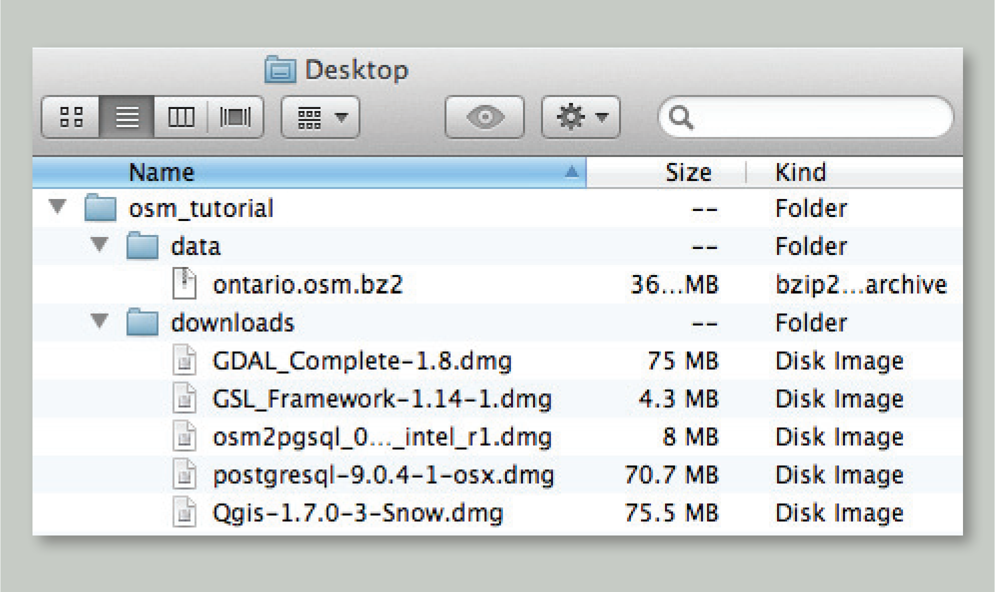
QGIS

QGIS is an open-source GIS software licensed under the GNU public license. This tutorial makes use of QGIS to query, extract, and visualize the OpenStreetMap data that currently lives inside of our PostgreSQL database. QGIS provides a vast array of functionality which makes it a competitor of commercial solutions. Also, due to its open-source roots, users will find direct compatibility with many other open-source projects such as the PostgreSQL database. QGIS also provides powerful Python scripting capabilities, making it a viable option for automating spatial workflows. Lastly, the QGIS community has a large and growing plugin repository where users can find useful tools or scripts that enhance the usability and efficiency of QGIS.

INSTALLING QGIS AND ITS DEPENDENCIES

The QGIS dependencies and the QGIS installer are located on the KyngChaos Wiki.

- navigate to http://www.kyngchaos.com/software/qgis
- download the "GDAL Complete 1.8 framework package" to your "/Desktop/osm_tutorial/downloads" folder
- download the appropriate "GSL framework" to your "/Desktop/osm_tutorial/downloads" folder
- download the appropriate "QGIS 1.7.x.x" build to your "/Desktop/osm_tutorial/downloads" folder
- proceed through the installations of GDAL and GSL before QGIS. Make sure to accept all the default configurations/options.
- QGIS installation is also straightforward. Install this after GDAL and GSL.

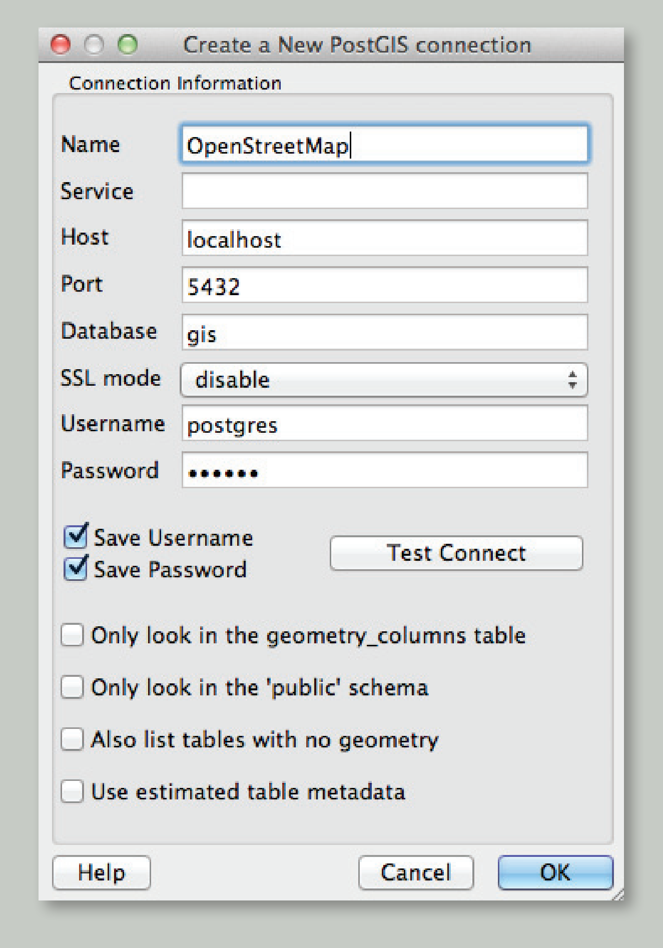


Name	Size	Kind
▼ osm_tutorial	--	Folder
▼ data	--	Folder
ontario.osm.bz2	36...MB	bzip2...archive
▼ downloads	--	Folder
GDAL_Complete-1.8.dmg	75 MB	Disk Image
GSL_Framework-1.14-1.dmg	4.3 MB	Disk Image
osm2pgsql_0..._intel_r1.dmg	8 MB	Disk Image
postgresql-9.0.4-1-osx.dmg	70.7 MB	Disk Image
Qgis-1.7.0-3-Snow.dmg	75.5 MB	Disk Image

QUERYING POSTGIS INSIDE QGIS

Let's launch QGIS for the first time and query our PostgreSQL/PostGIS database and load some OpenStreetMap data.

- Launch QGIS from the applications folder
- "File > Save project as" and save the project before we get started and forget to! Name it something relevant and place it in the "/Desktop/osm_tutorial/data" folder
- The toolbar at the top of the application window holds the tools that allow us to add new data to our map.
- By hovering over the icons we can locate the "Add PostGIS Layer" tool. Select this tool to open the Add PostGIS Tables dialogue.
- Select "new" to add our PostGIS database.
- Configure your new PostGIS layer as so (remember that the PostGIS layer uses your PostgreSQL configured username and password):



Create a New PostGIS connection

Connection Information

Name: OpenStreetMap

Service:

Host: localhost

Port: 5432

Database: gis

SSL mode: disable

Username: postgres

Password: ••••••

☑ Save Username

☑ Save Password

Test Connect

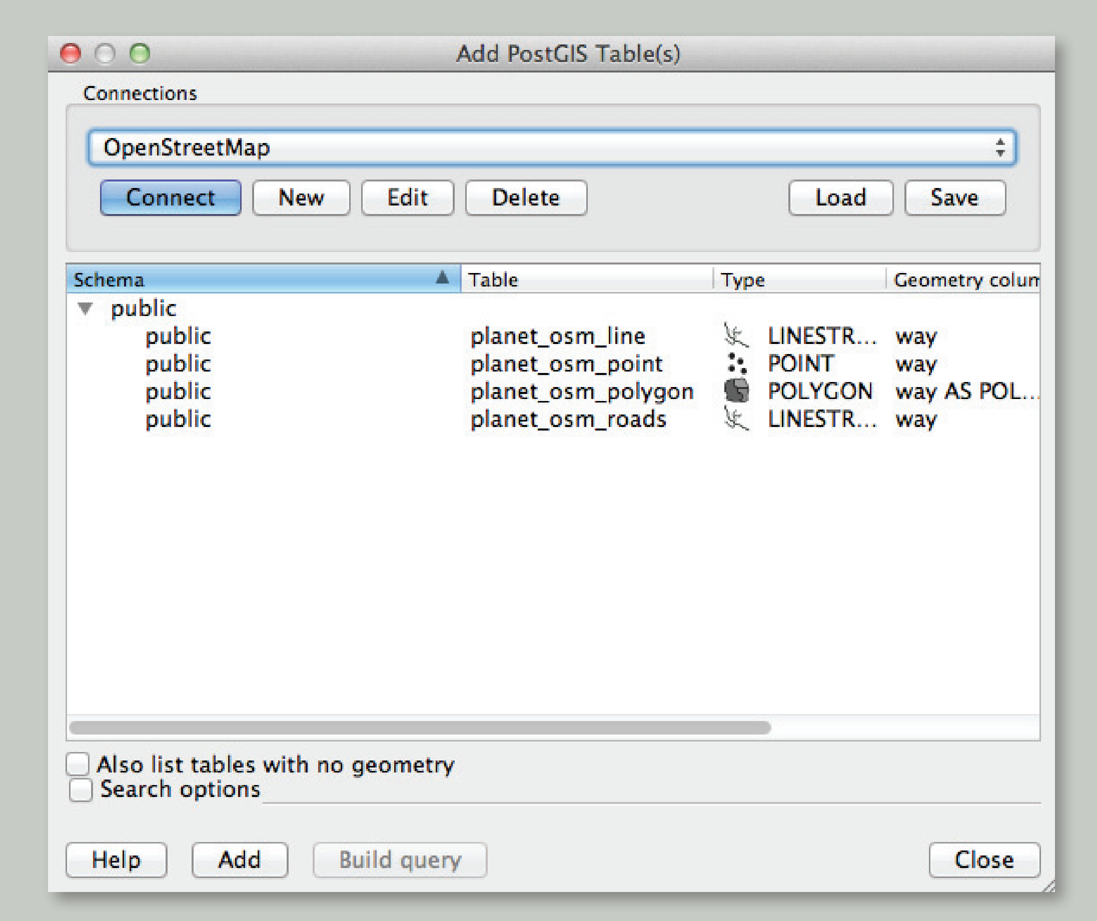
☐ Only look in the geometry_columns table

☐ Only look in the 'public' schema

☐ Also list tables with no geometry

☐ Use estimated table metadata

Help | Cancel | OK



Add PostGIS Table(s)

Connections

OpenStreetMap

Connect | New | Edit | Delete | Load | Save

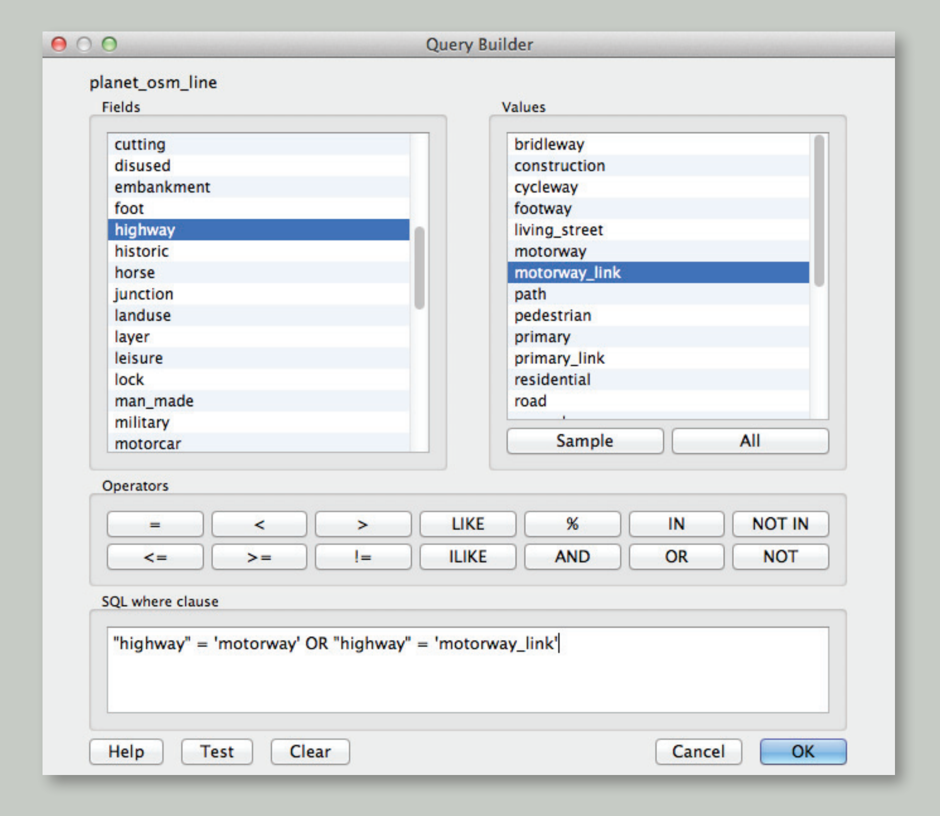
Schema	Table	Type	Geometry colum
public			
public	planet_osm_line	LINESTR...	way
public	planet_osm_point	POINT	way
public	planet_osm_polygon	POLYGON	way AS POL...
public	planet_osm_roads	LINESTR...	way

☐ Also list tables with no geometry

☐ Search options

Help | Add | Build query | Close

If this works, you will be able to connect to your OpenStreetMap PostGIS layer and you will get the following screen:



"highway" = 'motorway' OR "highway" = 'motorway_link'

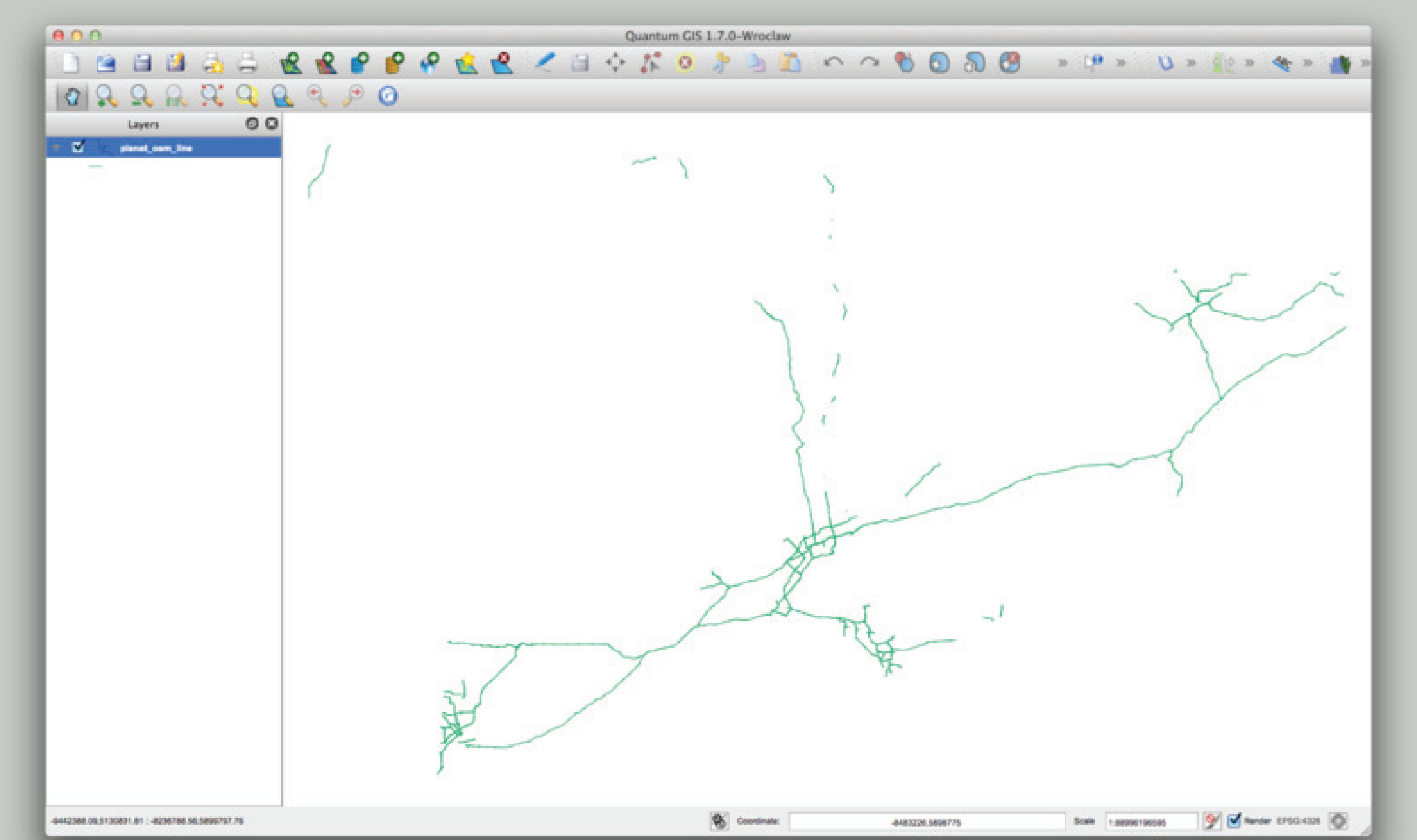
I will show you how to add some data using the query builder. All that you will need to do is get a grasp of the query builder syntax and the OSM tables so you can build your own custom queries and add OSM data to your projects at your heart's content.

- Select the first row, "public|planet_osm_line|LINESTRING|way."
- Select "Build Query."
- Using the interactive query builder, replicate the following query syntax:

Tip select "all" in the "Values" column to see all the available headings.

- Choose "OK."
- Select "Add."

The data will be queried from your PostgreSQL/PostGIS database and added to your map. If you have followed the tutorial using the Ontario extract you should have something that looks similar to the following image:



That is it! You are now ready to start using OSM data in your GIS. Note that you can export to shapefile in QGIS so that you can take the data and move it across computers and open it up in other GIS applications. Remember to save often in case QGIS crashes!