

Tutorial geoCnet.py

Author: Jéssica Domingues Santos (jessica.dominguess@gmail.com)

Linux - Ubuntu

1. Instalações:

- Igraph:
 - apt install libigraph-dev
 - PostgreSQL:
 - sudo apt install postgresql-9.5
 - PostGIS:
 - sudo apt install postgis postgresql-9.5-postgis-2.2
 - PgAdmin3:
 - sudo apt install pgadmin3
 - Python e libs:
 - Verificar se existe uma versão do Python 3 instalada (digitar python3 no terminal);
 - Instalar pip3 para gerenciamento de pacotes:
 - sudo apt-get install python3-pip
 - Import necessário, se não tiver instalar:
 - import numpy
 - import configparser
 - import os
 - import sys
 - import psycopg2
 - import subprocess
- sudo pip3 install <name>
- Visualização: QGIS
 - apt install qgis

2. Configurando o banco:

Criar senha para o usuário postgres (pelo terminal):

- sudo su postgres
- psql -c "ALTER USER postgres WITH PASSWORD 'masterkey'"
- createdb <nome_banco>
- psql -c "CREATE EXTENSION postgis; CREATE EXTENSION postgis_topology;" <nome_banco>

3. Diretórios:

- run: todos arquivos de execução
- src: fontes e Makefile
- shp:
 - in: shape de entrada (pontos)
 - out: shape de saída (pontos e linhas)
- shp.py e defaults.cfg

4. Arquivos:

- src:
 - geocnet.c: arquivo com as funções para calcular as propriedades da rede
 - geocnet.o : “objeto” das funções
 - geocnetApp.c: main da aplicação
 - Makefile: Bloco de comandos para compilar a aplicação.
- run:
 - matrix.in: arquivo de saída do main.cpp e entrada para geocnetApp.c, pode ser substituído por outro arquivo desde que obedeça às regras:
 1. Primeira linha representa a quantidade de matrizes
 2. Segunda o tamanho
 3. Seguido por cada matriz sem pular linha entre cada uma.
 - weighted_adjacency.txt: saída de geocnetApp de acordo com o limiar escolhido, também é arquivo de entrada para shp.py (descrito no defaults.cfg)
 - properties.txt: saída de geocnetApp, cada conjunto de dados representa as propriedades da rede. O arquivo também é entrada para o shp.py
 - main.x: executável do geocnetApp.c
- shp:
 - in: os arquivos de pontos devem ter uma coluna ID.
 1. Obs: devem ser equivalentes a matriz de adjacência
 - out: arquivo .shp de arestas e vértices com as propriedades

5. Pontos chaves de alterações:

- geocnetApp.c:
 - É possível alterar o threshold que representa a regra de criação das arestas
- geocnet.c:
 - É possível acrescentar funções, calcular novas propriedades e salvar arquivos.
- shp.py:
 - Alterar número de n_row, n_cul, n_times. A variável n_times representa a 3 dimensão da matriz
- defaults.cfg:
 - PATH da matriz de adjacência, das propriedades calculadas, da entrada do shape (in) e da saída (out).
 - Verificar o nome do banco

6. Executando a aplicação

- ir na pasta src:
 - executar: make
- ir na pasta run:
 - ./main.x
- voltar um diretório:
 - Se python instalado pelo anaconda: python geocnet.py
 - Se python instalado pelo terminal: python3 geocnet.py

geocnet.py Tutorial

Author: Jéssica Domingues Santos (jessica.dominguess@gmail.com)

Installation Guide for Ubuntu

1. Package installation:

- **Igraph:**
~\$ sudo apt-get install libigraph-dev
- **PostgreSQL:**
~\$ sudo apt-get install postgresql-9.5
- **PostGIS:**
~\$ sudo apt-get install postgis postgresql-9.5-postgis-2.2
- **PgAdmin3:**
~\$ sudo apt-get install pgadmin3
- **Python e libs:**
 - Check if a Python 3 version is installed (type python3 on the terminal);
 - Install pip3 for package management:
~\$ sudo apt-get install python3-pip
 - It's required to import the modules below:
 - import numpy
 - import configparser
 - import os
 - import sys
 - import psycopg2
 - import subprocess
 - If any module is missing, you can install it:
~\$ sudo pip3 install <name>
- **Visualization: QGIS**
 - sudo apt-get install qgis

2. Database Configuration:

Change the postgres user password (on the terminal):

```
~$ sudo su postgres
```

```
~$ psql -c "ALTER USER postgres WITH PASSWORD 'masterkey'"
```

- createdb <database_name>
~\$ psql -c "CREATE EXTENSION postgis; CREATE EXTENSION postgis_topology;" <database_name>

3. Folders:

- **run:** All executable files
- **src:** sources and Makefile
- **shp:**
 - in: input shape (points)
 - out: output shape (points and lines)
- **shp.py** and **defaults.cfg**

4. Files:

- **src:**
 - geocnet.c: file with the functions to calculate network properties
 - geocnet.o : "Object" of functions
 - geocnetApp.c: main file
 - Makefile: Commands to compile the application.
- **run:**
 - matrix.in: output file from main.cpp and input to geocnetApp.c can be replaced by another file as long as the rules follow:
 1. First row represents a quantity of matrices.
 2. Order of the matrix (size).
 3. Followed by each matrix without jumping line between each one.
 - weighted_adjacency.txt: output from geocnetApp according to the chosen
 - threshold, is also input file to shp.py (described in defaults.cfg)
 - properties.txt: output from geocnetApp, each data set represents the network properties. The file is also sent to shp.py
 - geocnetApp.x: geocnetApp executable file from geocnetApp.c
- **shp:**
 - in: the point files must have an ID column.
 1. Note: must be equivalent to adjacency matrix
 - out: .shp file of edges and nodes with the properties

5. Key points for changes:

- geocnetApp.c:
 - You can change the threshold that represents the edge creation rule.
- geocnet.c:
 - You can add functions, calculate new properties, and save files.
- geocnet.py:
 - Change number of n_row, n_cul, n_times. The variable n_times represents the 3 dimension of the matrix
- defaults.cfg:
 - PATCH of the adjacency matrix, calculated properties, input shape and output shape.
 - Check the database name

6. Running the application

- go to src folder and run:
~\$ make
- go to run folder and run:
~\$./main.x
- back one directory:
 - If python installed by anaconda:
~\$ python shp.py
 - If python is installed by the terminal:
~\$ python3 shp.py