4 potential ways to get METIS to work on Windows:

Line that METIS is needed for:  
     n\_cuts, membership = metis.part\_graph(

        max(int(np.ceil(G.number\_of\_nodes() / chunk)), 1), adjacency=adj\_list(G))

Problem: membership is needed by later function

Solution 1: Replace with new function that does the same thing (the only solution I could get to work):

- Look for similar function to existing by seeing what function does:  
From [metis.part\_graph documentation](https://metis.readthedocs.io/en/latest/#metis.part_graph) membership is a list of partition indices corresponding to the minimization of the objective function.

np.ceil(G.number\_of\_nodes() / chunk) divides the number of nodes by chunks and rounds up

max(int(np.ceil(G.number\_of\_nodes() / chunk)), 1) ensures that at least one partition is created

adj\_list(G): This likely represents a function or operation that generates the adjacency list of the graph G. The adjacency list is a data structure used to represent which nodes are adjacent to each other in a graph.

**Other functions that do a partition/spectral clustering:**

[Community.best\_partition](https://python-louvain.readthedocs.io/en/latest/api.html)   
has dependencies : python-louvain & community  
or

[sklearn.cluster.SpectralClustering](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html)

Solution 2: Get pymetis to work on Windows (failed to fix problem but other methods mentioned in forum might work):

Problem: Get the following error when you try to install dependency:  
 regex.h cannot be found  
  
Tried installing POSIX through mingw

Possible untried solution follow recommendations on stack exchange:  
<https://stackoverflow.com/questions/25975673/posix-regex-functions-required-and-missing>

Solution 3: Get metis to work on Windows (failed to fix problem due to MSbuild failing to work):

-This solution involves downloading METIS directly and using the metis python wrapper:

Windows specific METIS install instructions:

<https://stackoverflow.com/questions/50675790/how-to-install-metis-package-in-python-on-windows>

General METIS install instructions:  
Download and unzip METIS:  
<https://www.gilith.com/software/metis/install.html>

pip install cmake

Or download from here:  
http://www.cmake.org/. CMake generates Visual Studio project files,

which then can be built using Visual Studio. There are two ways to

generate visual studio files: using the command line and using the

CMake GUI.

Using the command line

----------------------

Open the command prompt and cd to the METIS source directory. Run

> cmake --help

and look at the list of generators for the Visual Studio studio you

want to build for. For example, the generator for Visual Studio 2010

is called "Visual Studio 10".

After you have found the appropriate generator, run

> .\vsgen -G "<GENERATOR>"

to generate the project files. The project files will be placed

build\windows.

Building METIS requires CMake 2.8, found at http://www.cmake.org/, as

well as GNU make. Assumming CMake and GNU make are installed, two

commands should suffice to build metis:

$ make config

$ make

Configuration

-------------

METIS is primarily configured by passing options to make config. For

example:

$ make config shared=1 cc=gcc-4.2

would configure metis to be built as a shared library using GCC 4.2.

Common configuration options are:

cc=[compiler] - The C compiler to use [default is determined by CMake]

shared=1 - Build a shared library instead of a static one

[off by default]

prefix=[PATH] - Set the installation prefix [/usr/local/ by default]

Advanced debugging related options:

gdb=1 - Build with support for GDB [off by default]

debug=1 - Enable debugging support [off by default]

assert=1 - Enable asserts [off by default]

assert2=1 - Enable very expensive asserts [off by default]

METIS' index and real data type size can be configured by editing

include/metis.h.

Installation

------------

To install METIS, run

$ make install

The default installation prefix is /usr/local. To pick an installation

prefix for METIS pass prefix=[path] to make config. For example,

$ make config prefix=~/myroot/

will cause METIS to be installed in ~/myroot/ when make install is run.

Using the CMake GUI

-------------------

It is also possible to use the CMake GUI, distributed with CMake. To

do this, open the CMake GUI, and browse to the location of METIS'

source with the "Browse Source" button. You can also change the binary

directory. This is where the Visual Studio project files will be

placed. Click "Generate" to select the correct visual studio version

and build the project files.

Using the VS project files

--------------------------

The Visual Studio project will be called METIS.sln. Open it in Visual

Studio. If the configuration is not already "Release", set it to

"Release". Type F7 to build. The METIS library will be in

<BINARY\_DIR>\libmetis\Release and the executable programs will be in

<BINARY\_DIR>\programs\Release. (<BINARY\_DIR> will be build\windows if

you used the command line or whatever you choose if using the CMake

GUI.)

* Theoretically if everything went okay you would find METIS.dll in ...\build\windows\libmetis

note: … is a placeholder

* You need to copy the path to METIS.dll and either set it as an environmental variable METIS\_DLL or find metis.py (python wrapper) and change METIS\_DLL= ‘...\build\windows\libmetis’

Solution 4: Convert the needed function from METIS to python (did not try this because of lack of C++ backgraound)