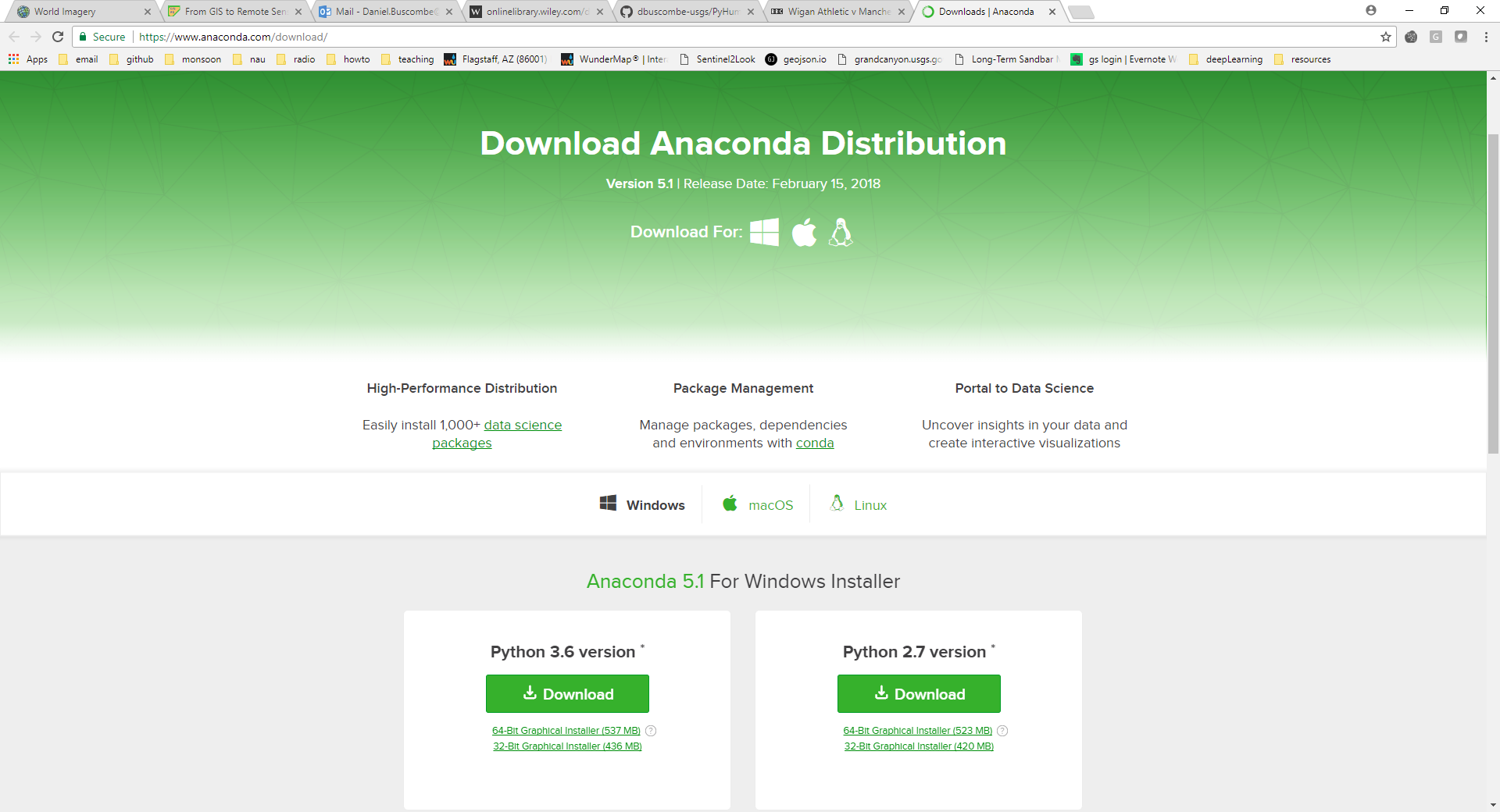
PriSM toolbox: user manual (Windows)

Written by Daniel Buscombe, [Daniel.Buscombe@nau.edu](mailto:Daniel.Buscombe@nau.edu). Version 0.1. February 2018

# Download and install Python

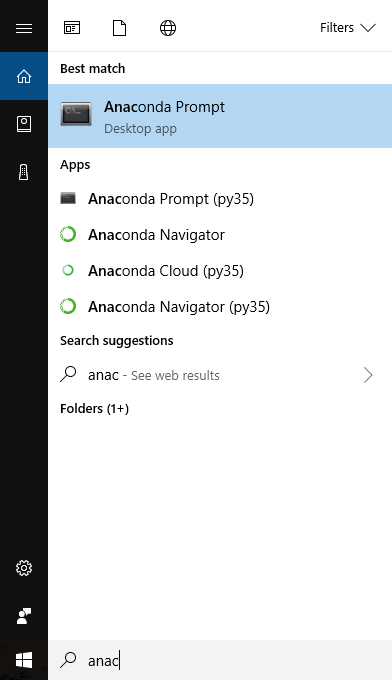
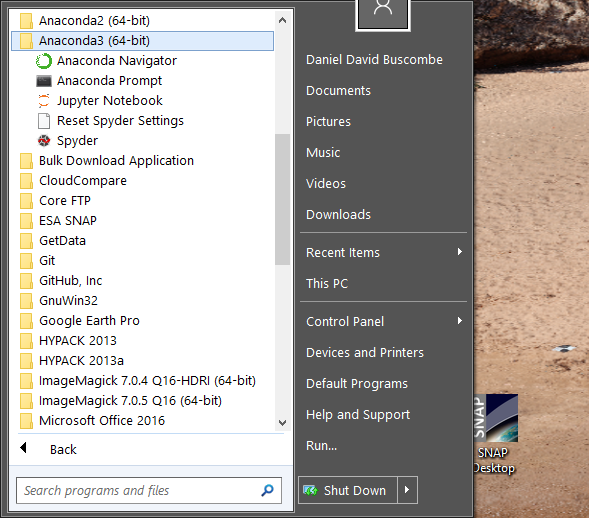
We recommend using the Anaconda Python distribution because it ships with a lot of libraries that PriSM uses. Anaconda is free, available for all major operating systems, 32 and 64 bit, lots of different versions of Python. Head to [www.anaconda.com/download](http://www.anaconda.com/download) and download the installer.



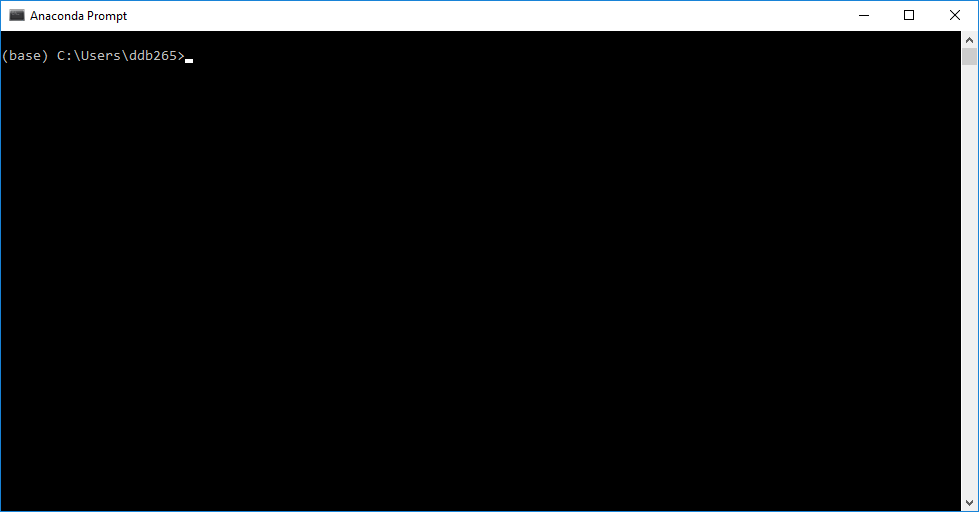
We recommend that you use Python 3. As of Feb. 19 2018, Python 3.6 is the latest version available from Anaconda. Go ahead and install that. For more experienced Python users who may prefer Python 2, you may use PriSM with versions of Python 2 greater than 2.7. For everyone else, Python 3 is the way to go.

# Launch python

From the search or start menus, find Anaconda. Anaconda3 refers to Python 3. Anaconda2 refers to Python 2. You may install them both and use either. Launch the ‘Anaconda prompt’



Below is what the Anaconda prompt looks like. It’s like a regular command prompt and has much of the same functionality

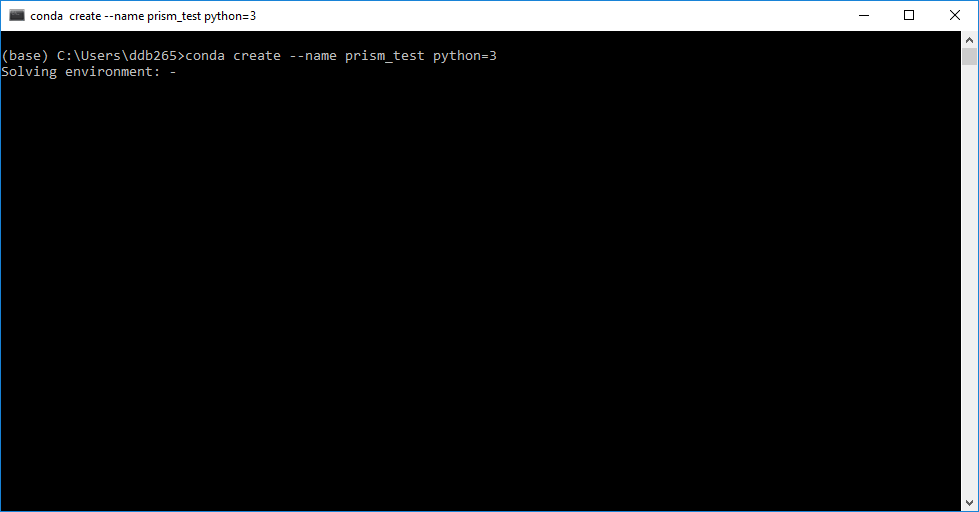


# Using PriSM in a Python virtual environment

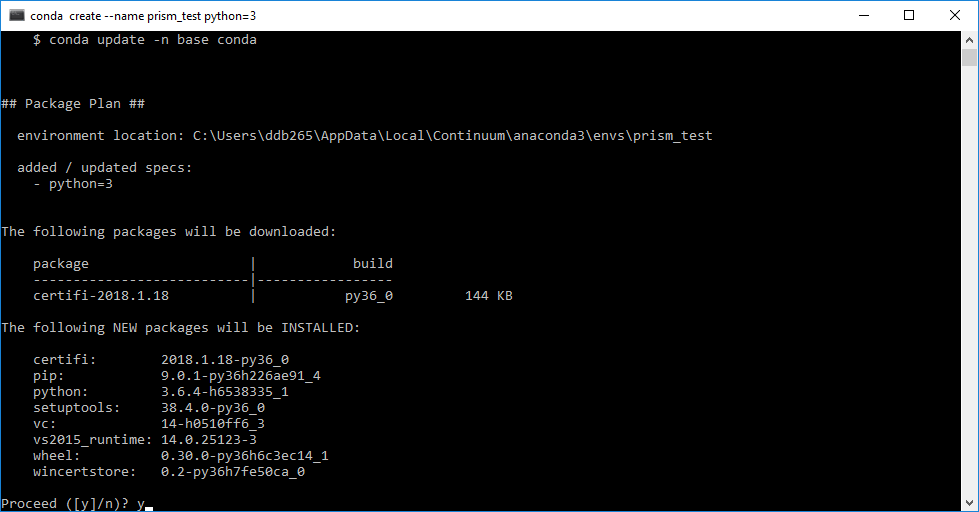
We recommend that you try the PriSM toolbox out first using a conda virtual environment (<https://conda.io/docs/user-guide/tasks/manage-environments.html>). This is a safe way to run python programs if you already have python installed on your computer. It is also a good way to make sure all the libraries you’ll need to install to use PriSM play nicely with each other.

## 1. Setting up the virtual environment

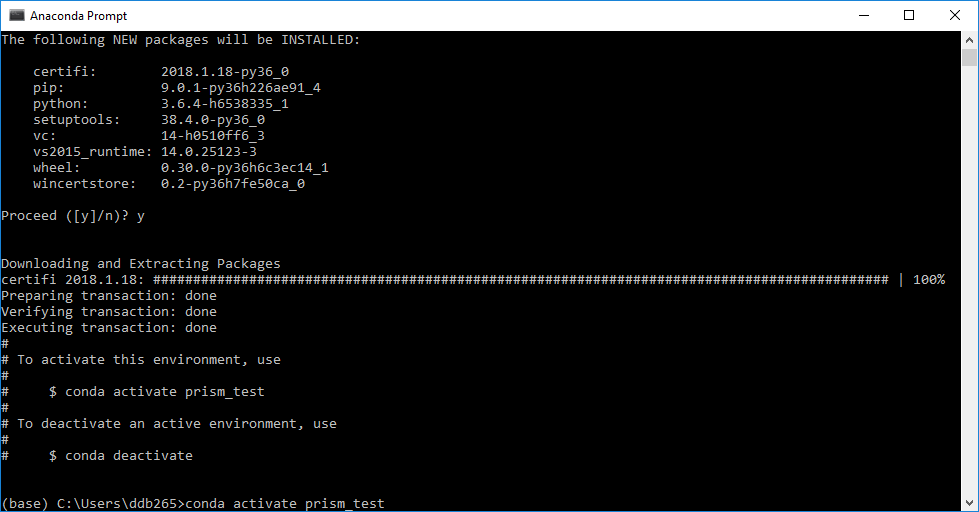
You may call this environment whatever you like. Below, I called my environment ‘prism\_test’. Write the following and press enter



When it asks you to proceed, hit enter (or type y, then press enter)



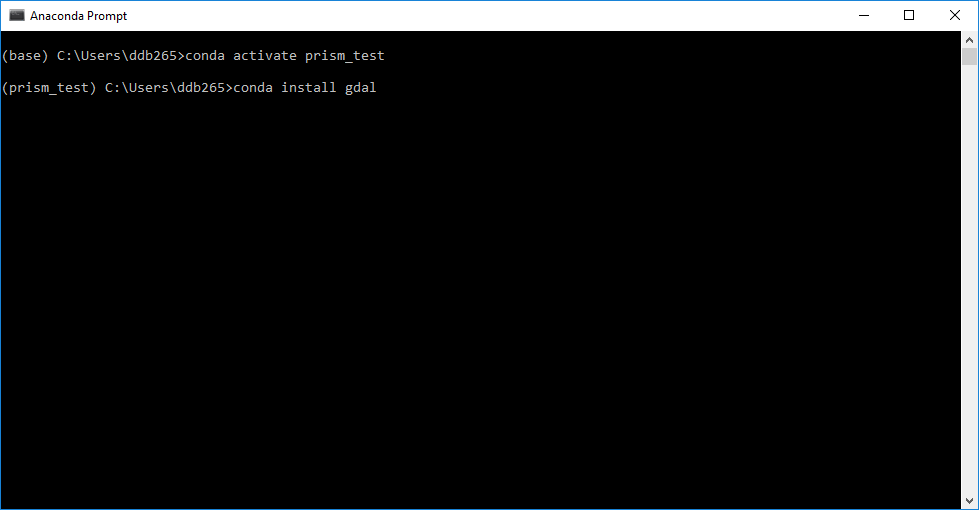
Next you need to activate the environment: it tells you what to do (in my case ‘conda activate prism\_test’ [Enter])



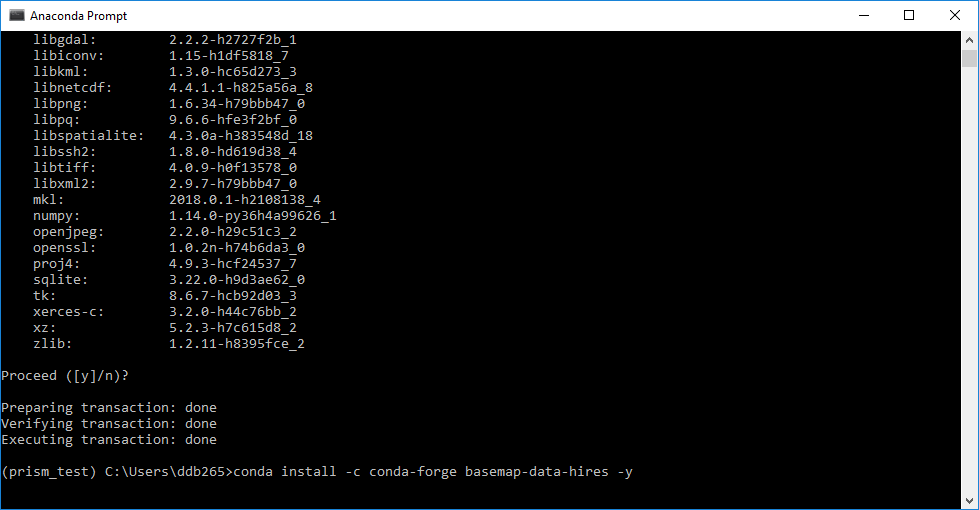
## 2. Installing required libraries

First you need to install the libraries that PriSM needs. There is quite a few, mostly providing us with geospatial and plotting routines but also some related to machine learning and other numerical computations. We’ll use conda (<https://conda.io/docs/>) and pip (<https://en.wikipedia.org/wiki/Pip_(package_manager)>) to install what we need automatically

1. GDAL (<http://www.gdal.org/>) is first (conda install gdal [Enter], then hit [Enter] again when it asks you if you’d like to proceed). This takes a few minutes to install



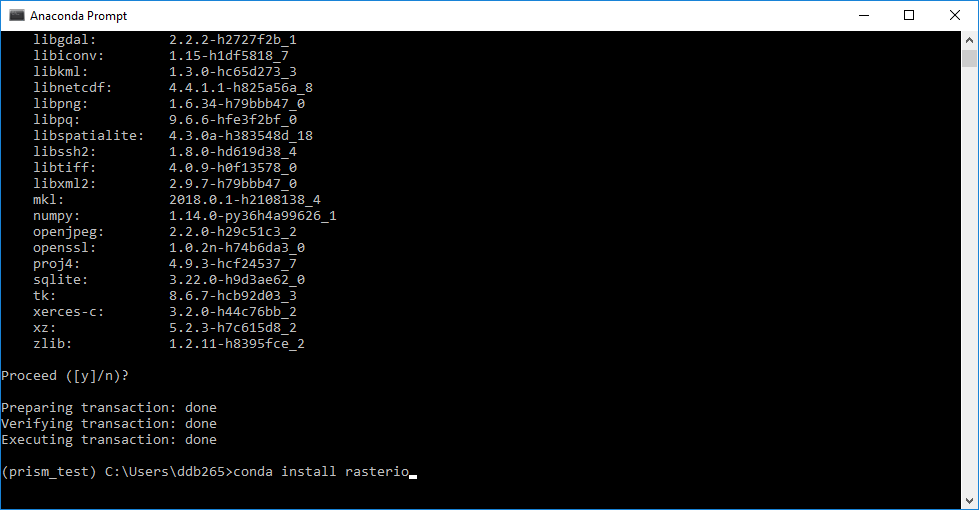
2. basemap (<https://matplotlib.org/basemap/>)



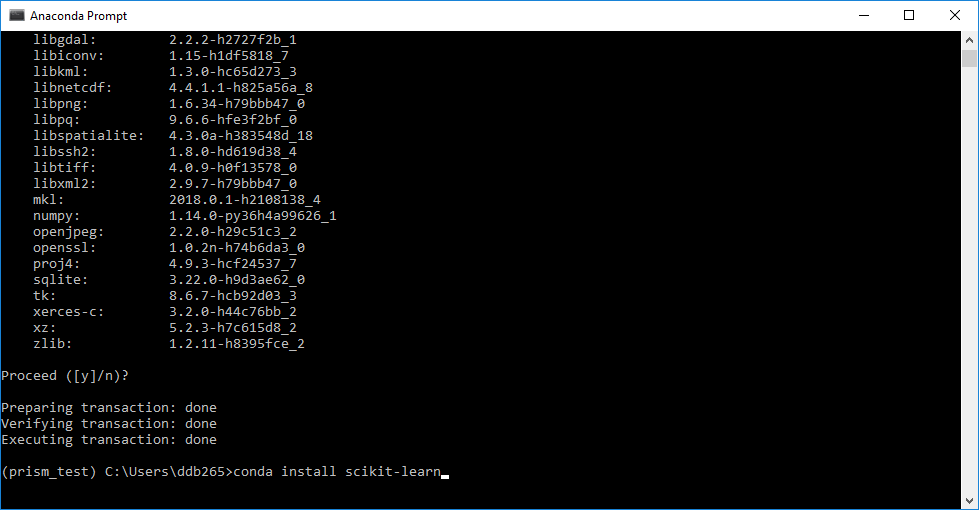
3. Fiona (<http://toblerity.org/fiona/>)



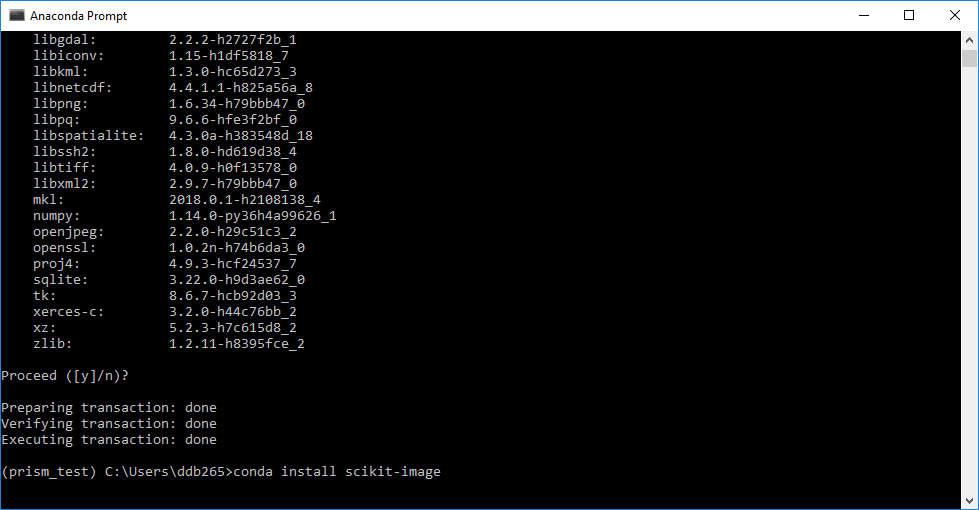
4. rasterio (<https://mapbox.github.io/rasterio/>)



5. scikit-learn (<http://scikit-learn.org/stable/>)

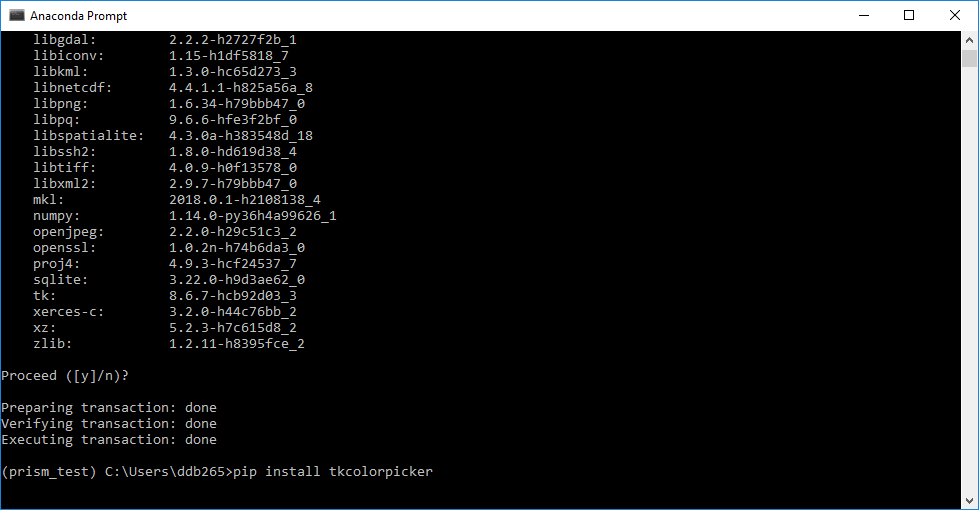


6. scikit-image (<http://scikit-image.org/>)

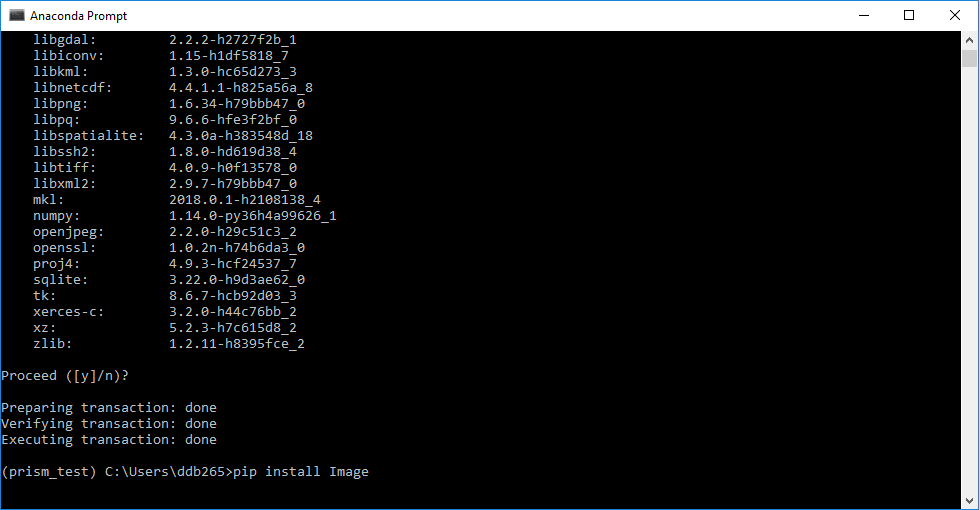


Finally, we install four libraries using pip.

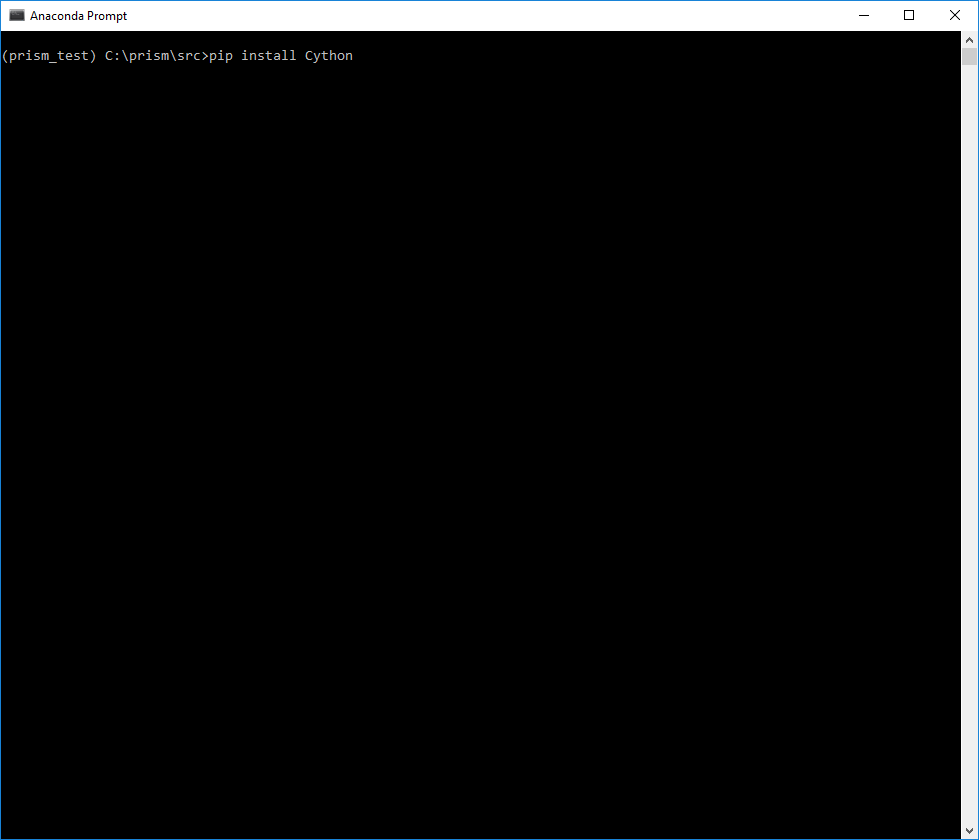
First, tkcolorpicker (<https://pypi.python.org/pypi/tkcolorpicker>)



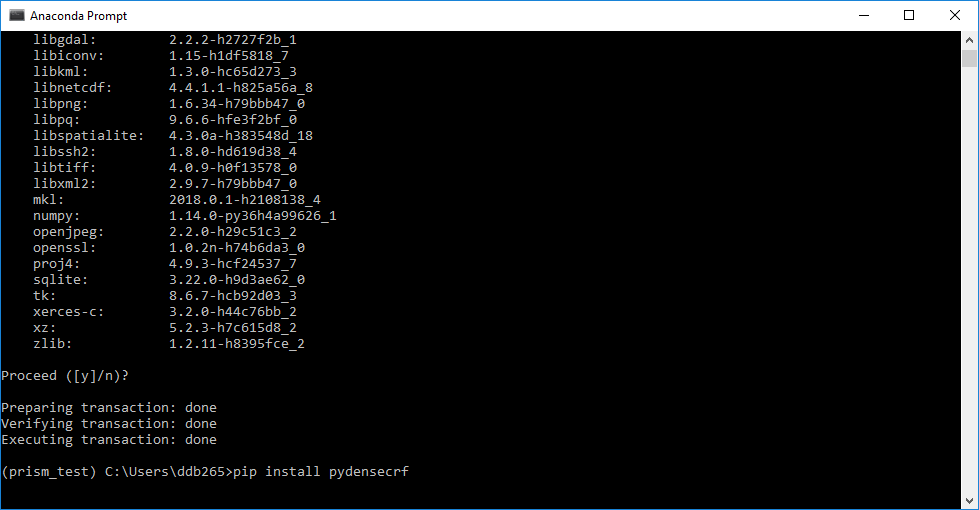
Then, the Imaging library (<https://pillow.readthedocs.io/en/latest/>)



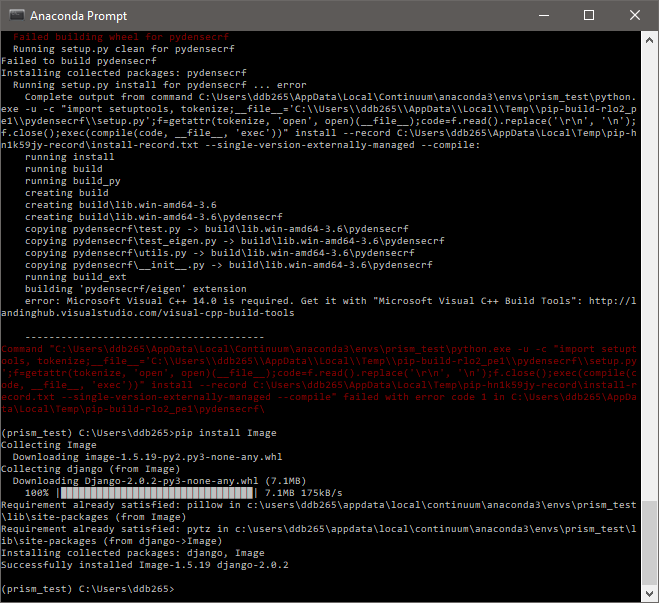
Then Cython (<http://cython.org/>)



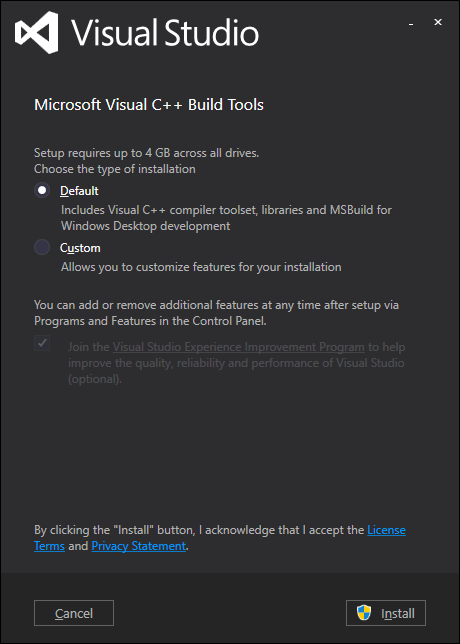
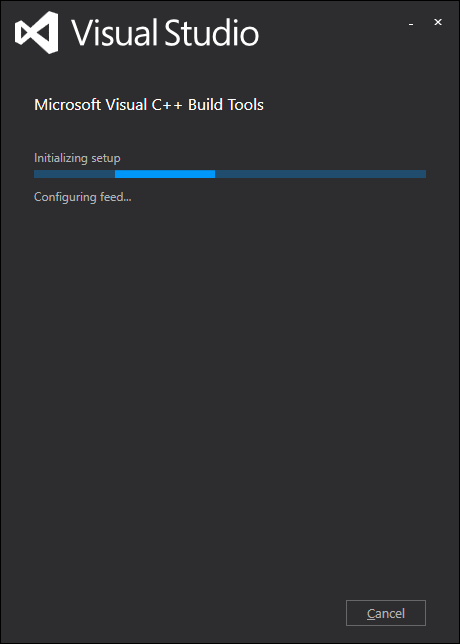
Finally, pydensecrf (<https://github.com/lucasb-eyer/pydensecrf>)



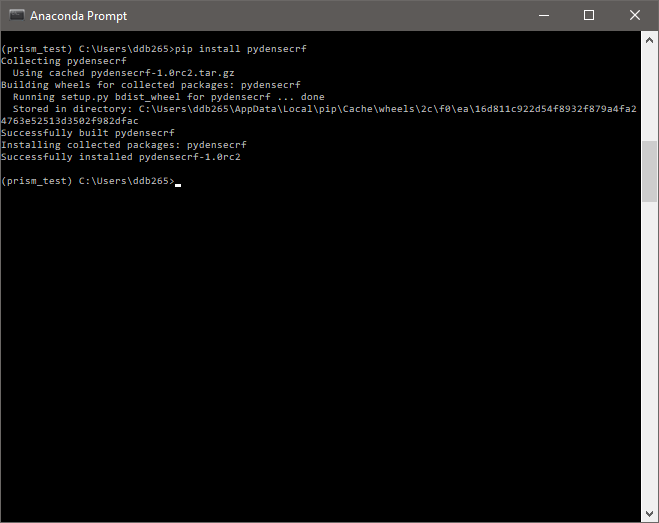
If you get an error that looks like this:



This means you’ll need to install a C++ compiler. First, go here: <http://landinghub.visualstudio.com/visual-cpp-build-tools> and download Visual C++ 2015 Build Tools



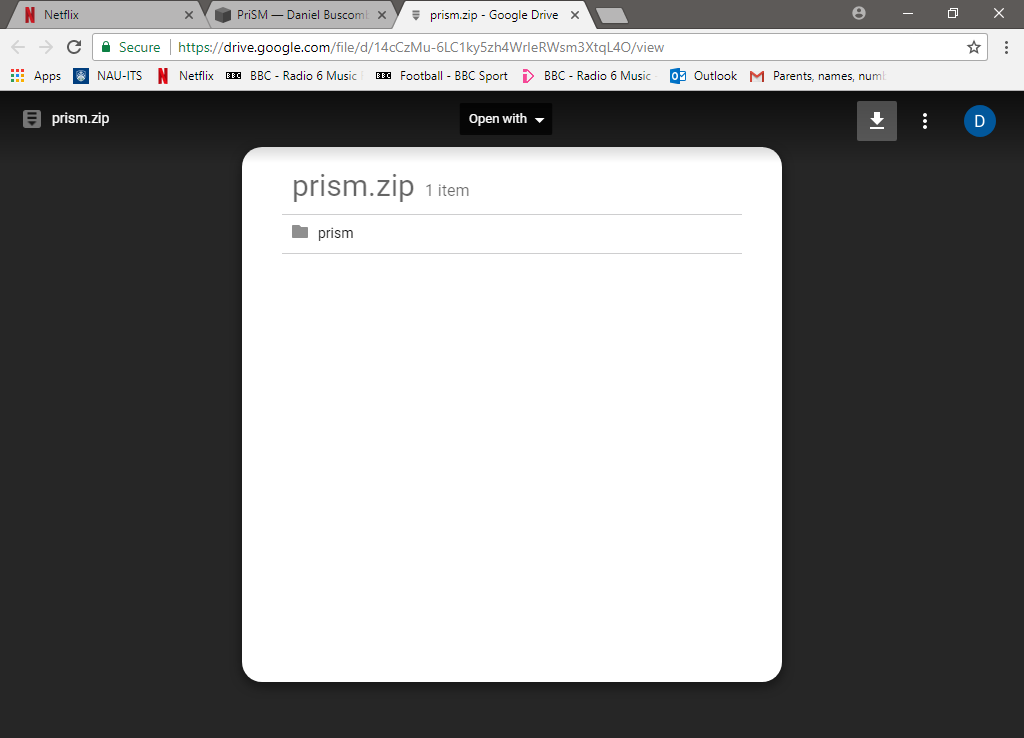
When it is finished, try to install pydensecrf again – it should now work



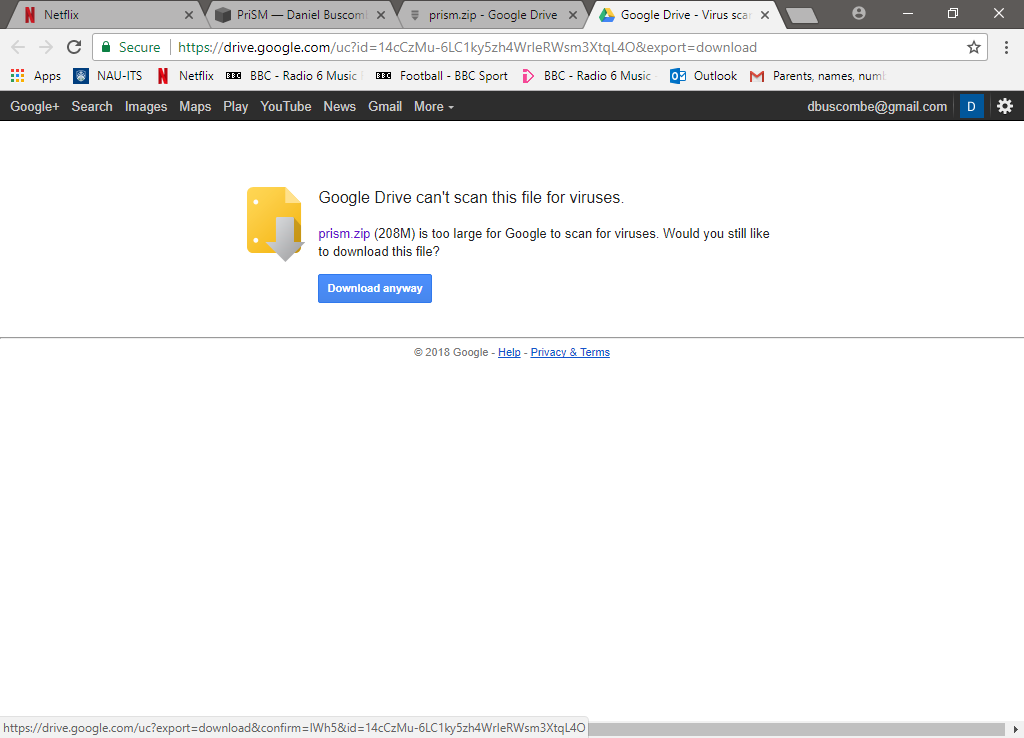
Ok, that should be all the libraries you need. Now you’re ready to download the toolbox and start using it.

## 3. Download PriSM

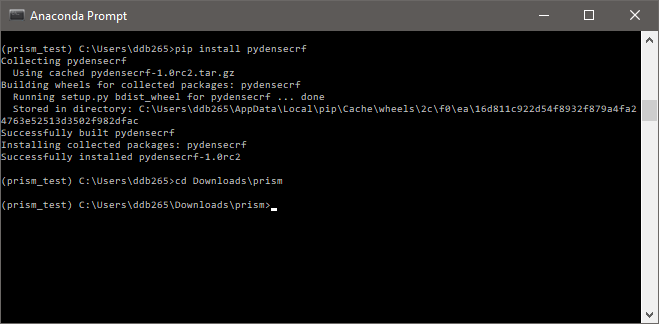
Go to [www.danielbuscombe.com/prism](http://www.danielbuscombe.com/prism). Hit ‘Download program’, then “download” (downward arrow)



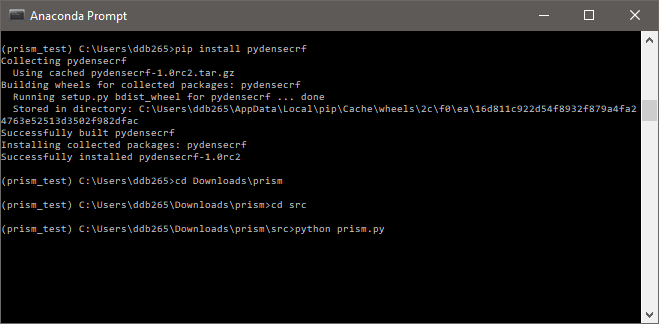
Then ‘download anyway’



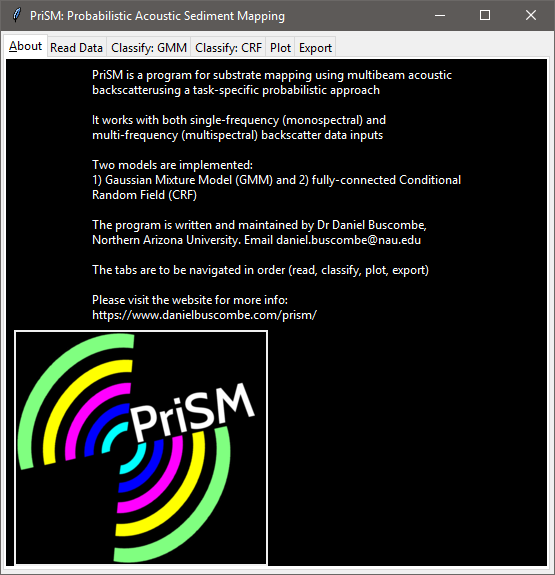
Unzip / extract to somewhere on your filesystem. Navigate there from the Anaconda prompt using the ‘cd’ (change directory) command



Navigate to the src directory. Now you can run the program by typing: python prism.py



And you should see this screen after a few moments

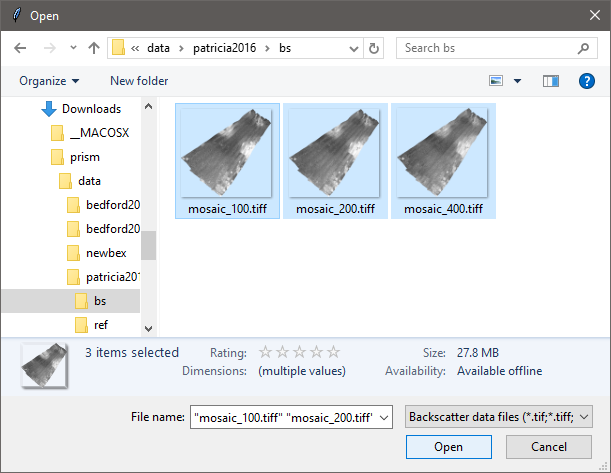
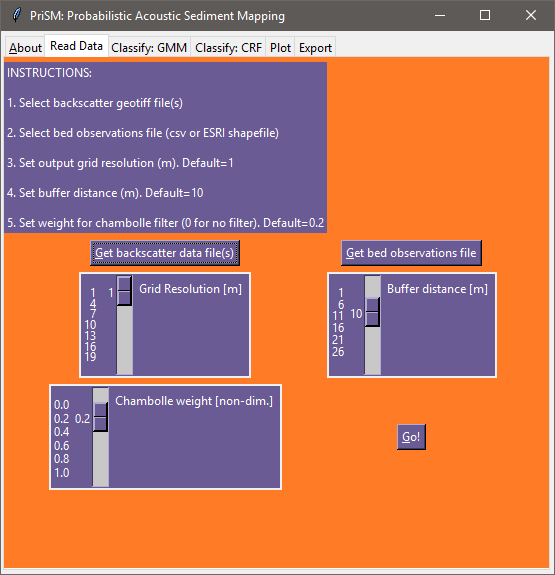


# 4. Running the program: Quickstart guide

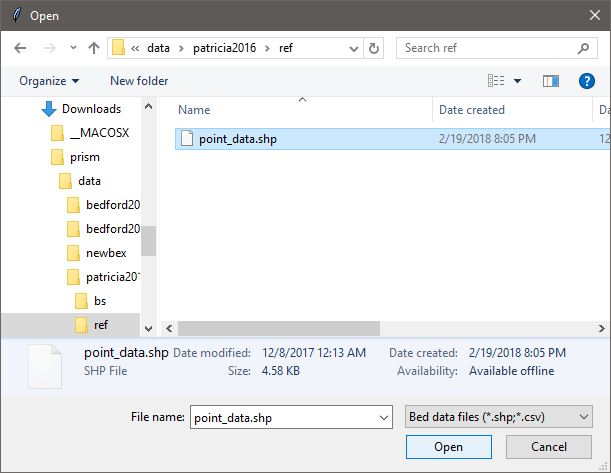
We will demonstrate the functionality of the program using the Patricia Bay dataset shipped with the program. All data were collected by R2Sonic for use with the R2Sonic 2017 Multispectral Backscatter competition.

## Read module

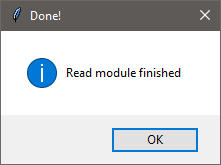
From the ‘Read Data’ tab, select backscatter data by hitting the ‘Get backscatter data file(s)’ button, then navigate to the data directory 🡪 Patrcia2016 🡪 select all three tiff files 🡪 Open



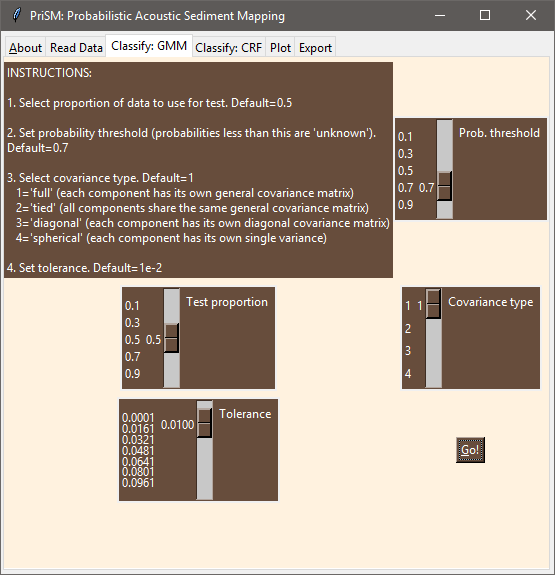
Do the same for the ‘bed observations file’ 🡪 select the shapefile in the ‘ref’ directory 🡪 Open



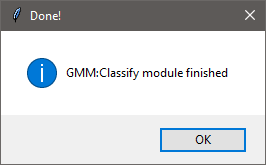
You may use the sliders to change the grid size (default is 1 m), buffer size (the distance in meters that grids are ascribed to each bed observation, default is 10 m), and a filter weight. The filter is for removing high-frequency (speckle) noise. For now, we recommend using the defaults. Hit the ‘Go’ button. After a few seconds, the following message should appear

 Hit OK and move onto the ‘Classify: GMM’ tab

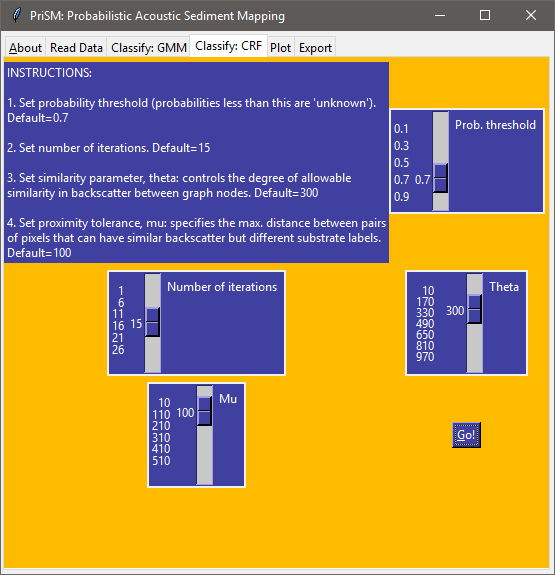
## Classify: GMM



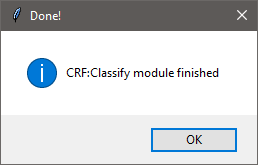
You may use the sliders to change the parameters. ‘Prob. Threshold’ is the threshold probability below which, a classification is considered indeterminate (‘unknown’). The higher this number, the fewer grid cells are classified but the greater confidence in those grid cells that are classified. ‘Test proportion’ is the amount of the data used for testing the model. The remaining proportion is used to train the model. Covariance type is explained in the instructions within the tab. ‘Tolerance’ is a numerical value that determines how the model decides when to finish (very generally, smaller = more accurate model). To run the GMM classification with all the defaults (recommended), simply hit ‘Go’. After a few seconds, you should see this screen

 Hit OK and move onto the ‘Classify: CRF’ tab

## Classify: CRF



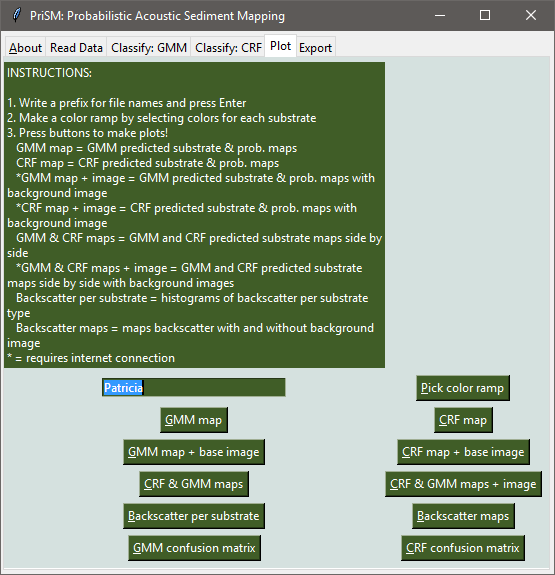
You may use the sliders to change the parameters. ‘Prob. Threshold’ is the threshold probability below which, a classification is considered indeterminate (‘unknown’). The higher this number, the fewer grid cells are classified but the greater confidence in those grid cells that are classified. ‘Number of iterations’ is the number of times the model iterates through before returning a solution. ‘Theta’ and ‘Mu’ parameters are explained in the instructions within the tab. To run the CRF classification with all the defaults (recommended), simply hit ‘Go’. When the model is finished, you should see this screen

 Hit OK and move onto the ‘Plot’ tab

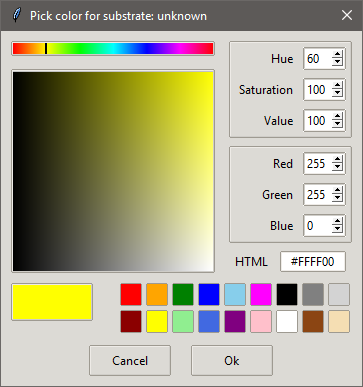
## Plot

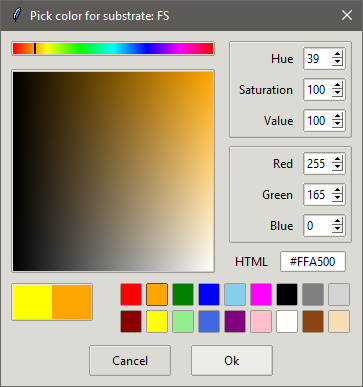
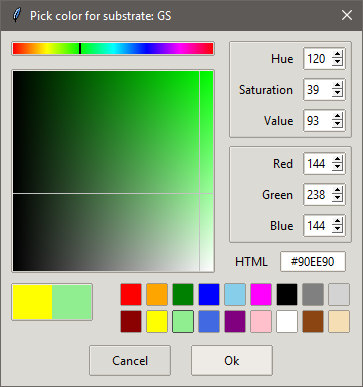
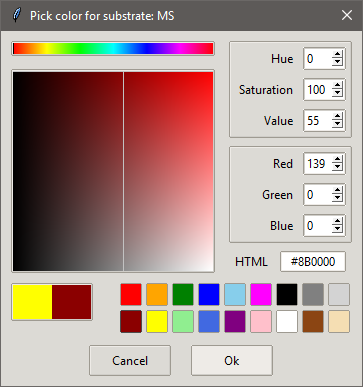
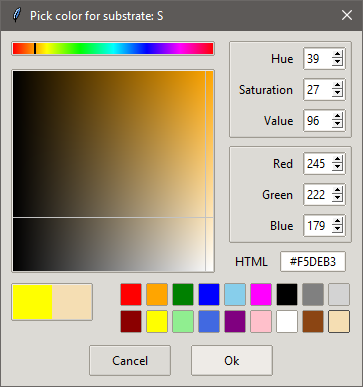
This tab is for generating figures and maps of the results. The files are written to the ‘outputs’ folder within the program

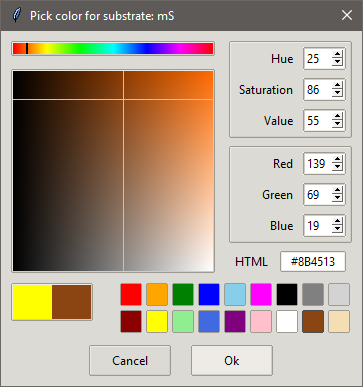
First, you need to provide a file prefix. Below I used ‘Patricia’. Remember to hit [Enter] after typing – the prefix should be highlighted, as below



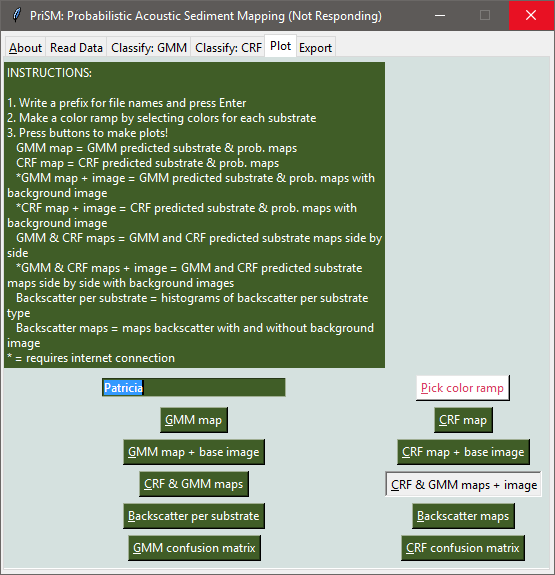
Then you’ll need to select colors for each of the substrates by selecting the ‘Pick color ramp’ button. You’ll see a dialog like this:

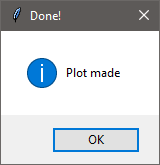
You are being asked to select a color for the ‘unknown’ substrate category. I tend to use gray. Cycle through each of the substrates assigning a color to each:

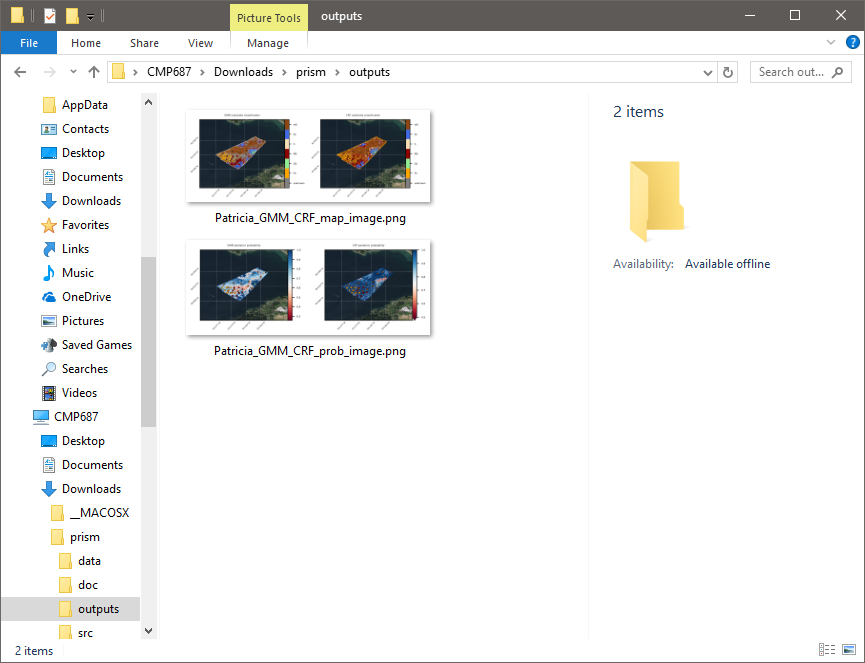
   

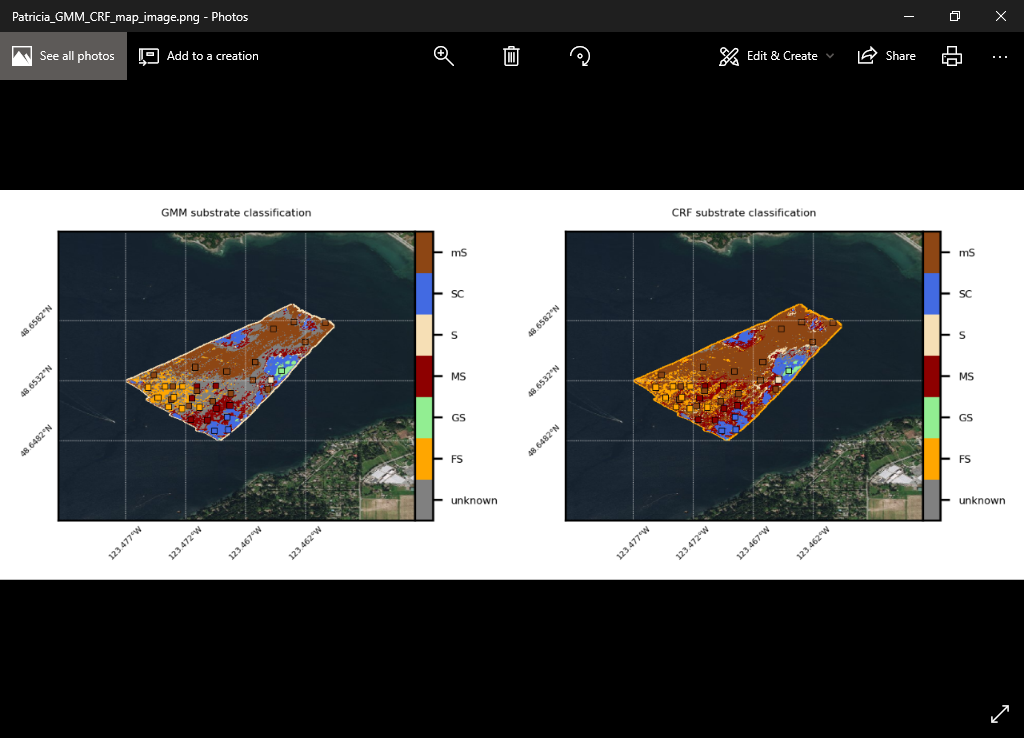
Now you’re ready to make some plots. The instructions within the tab explain what each button creates. Since we ran both the GMM and CRF modules, we can hit the ‘CRF & GMM maps + images’ button that will generate lots of outputs from both models:

 After a little while, you’ll get a message:

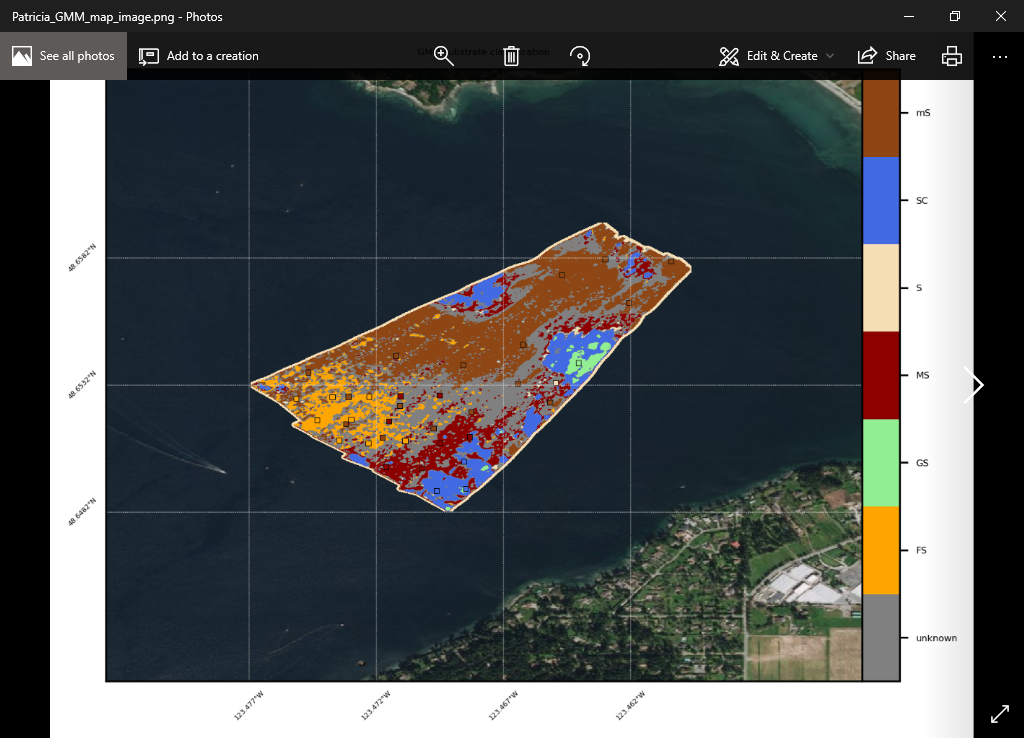
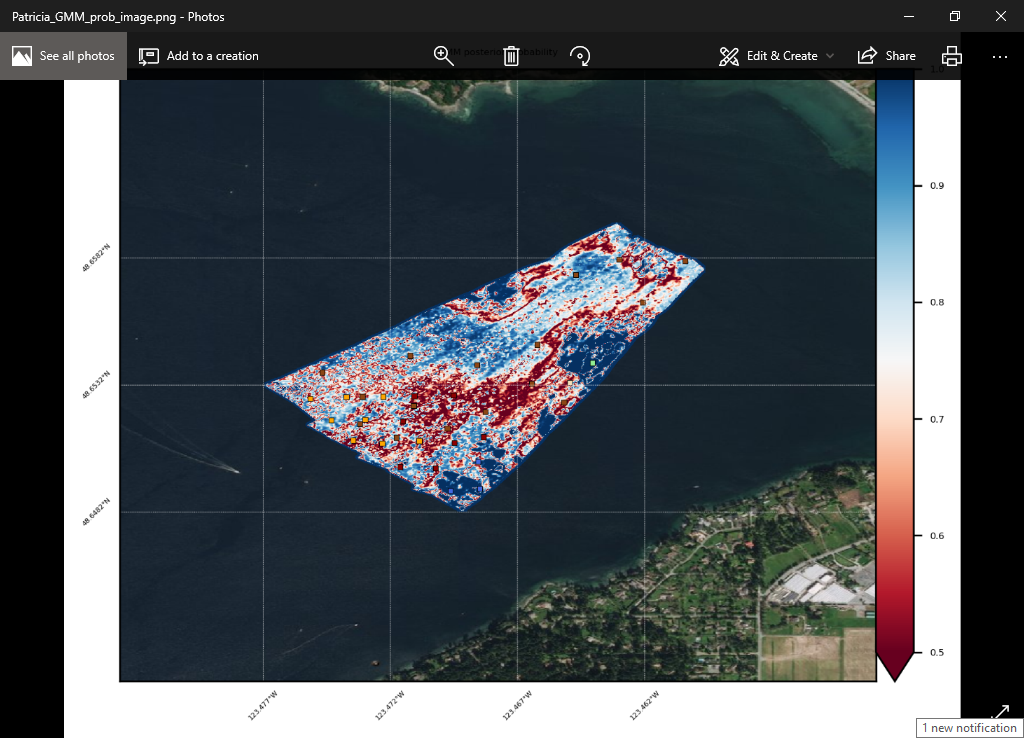
 and you can navigate to your browser window (prism 🡪 outputs ) to see the generated plots



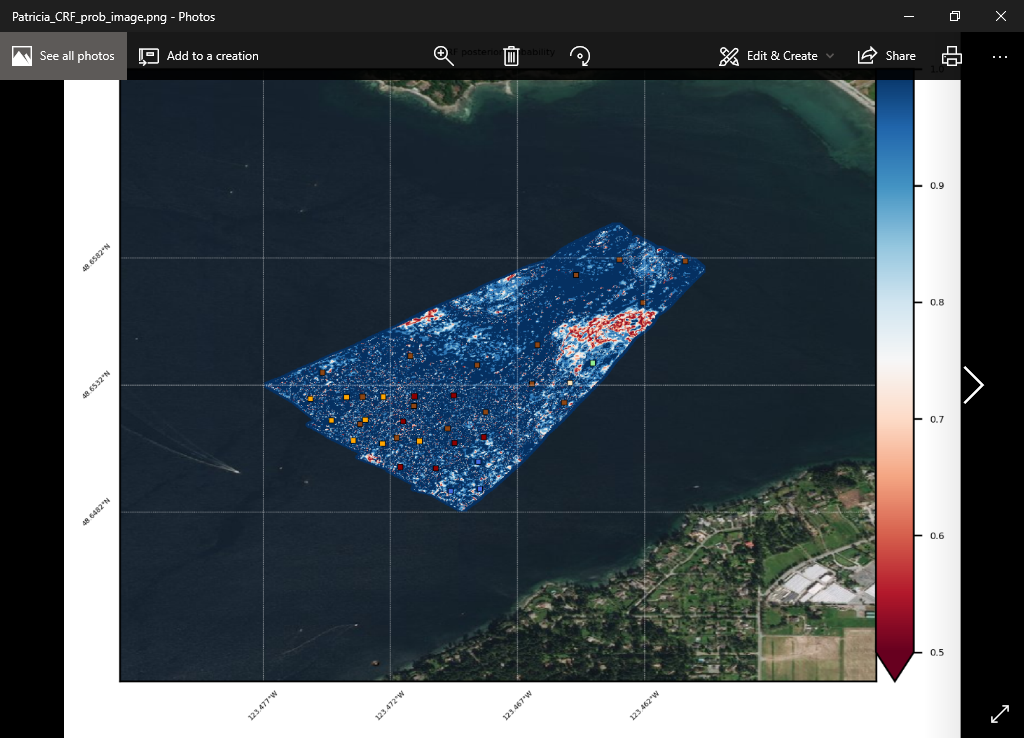
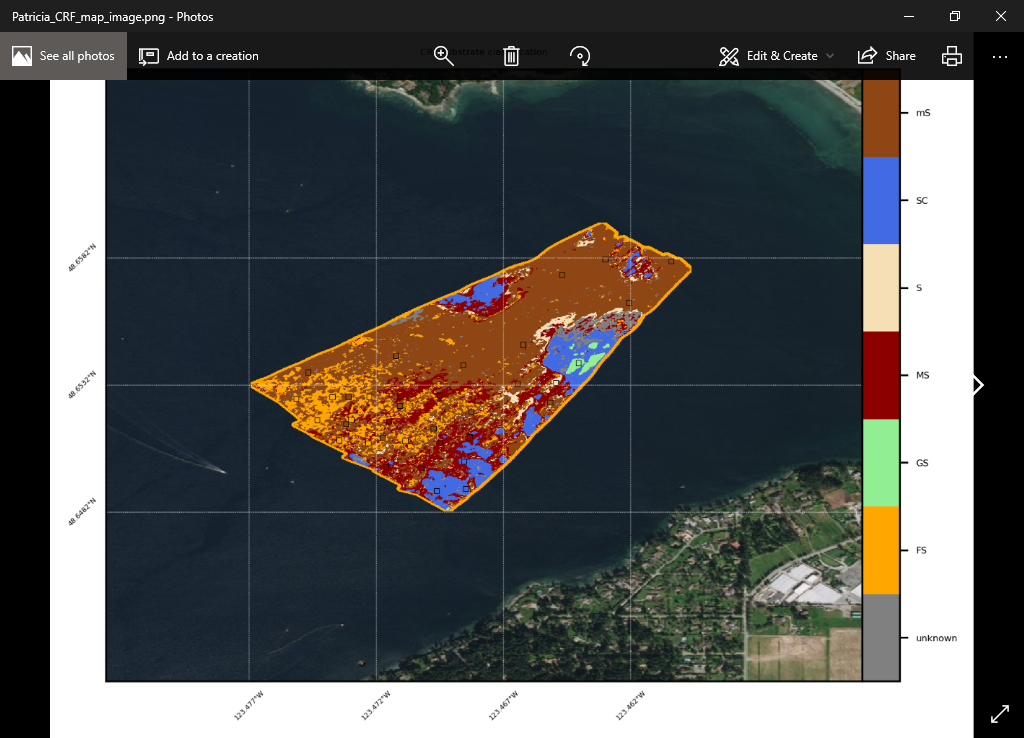
And of course you can view the png files in your image viewer:



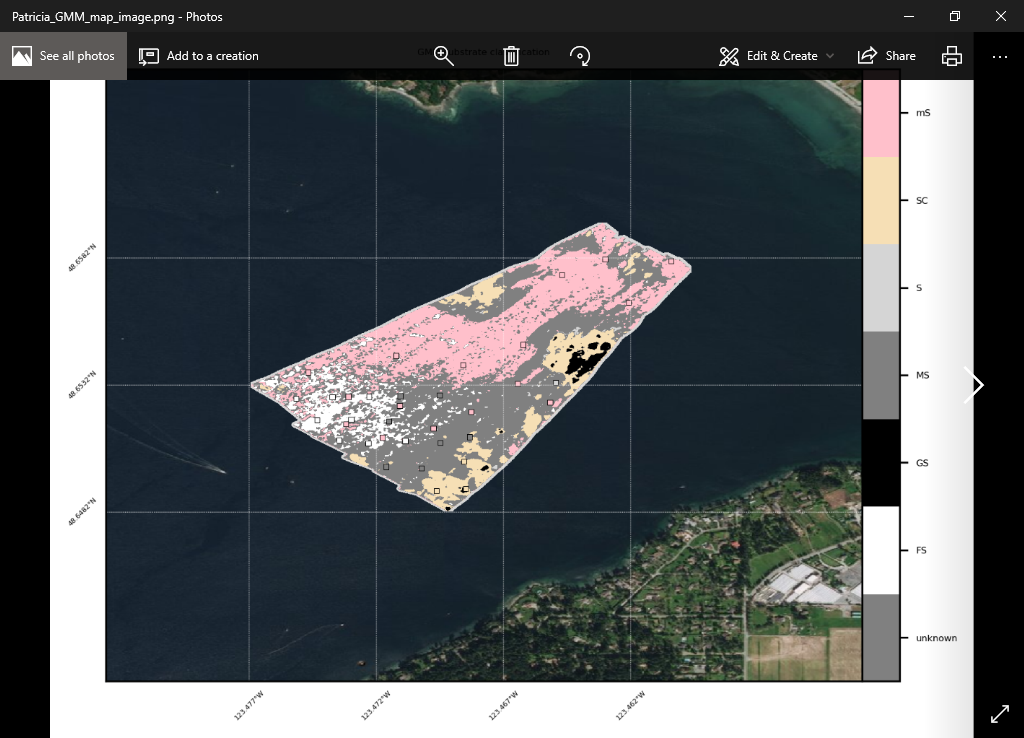
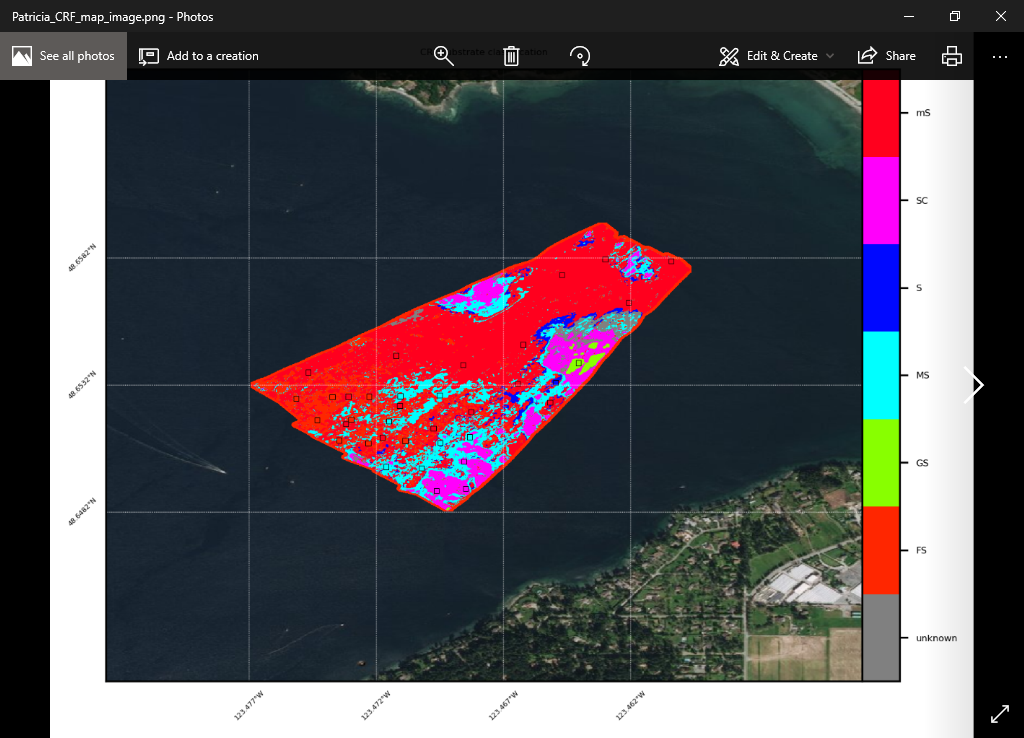
These are substrate maps for GMM and CRF outputs side-by-side, overlain onto an aerial image downloaded from the internet. You may generate other plots for GMM and CRF models individually. For example, ‘GMM map + base image’ will generate substrate and ‘posterior probability of substrate’ maps:

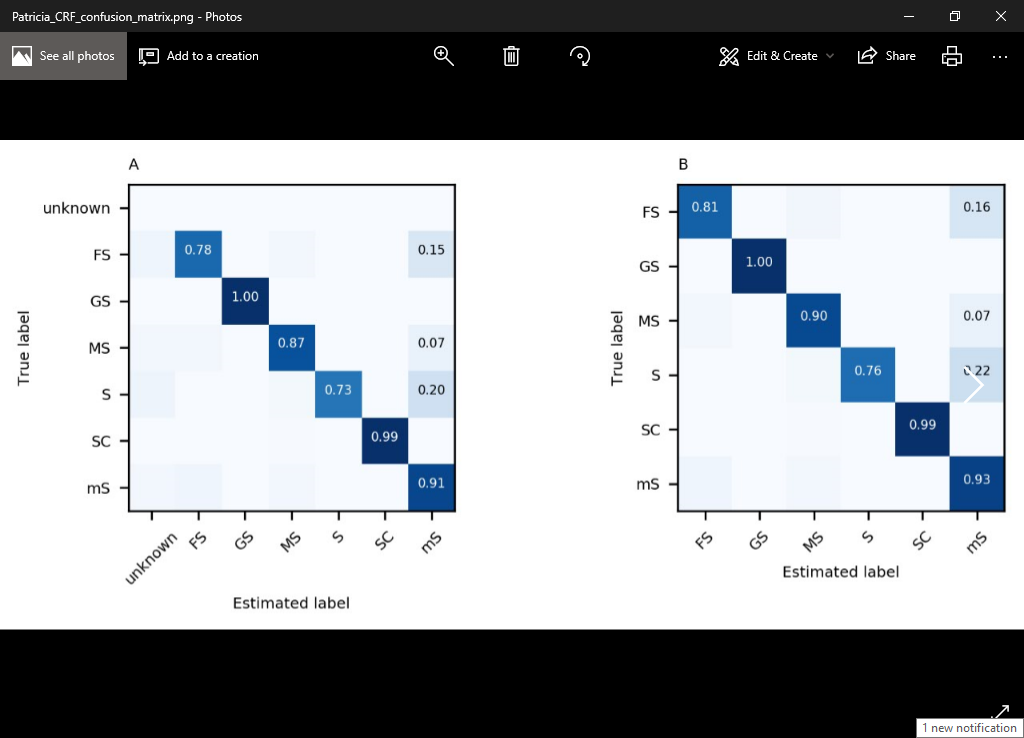
And ‘CRF map + base image’ will create the same things from the CRF outputs:



At any point, you may choose a different color scheme and redo the plots, for example:

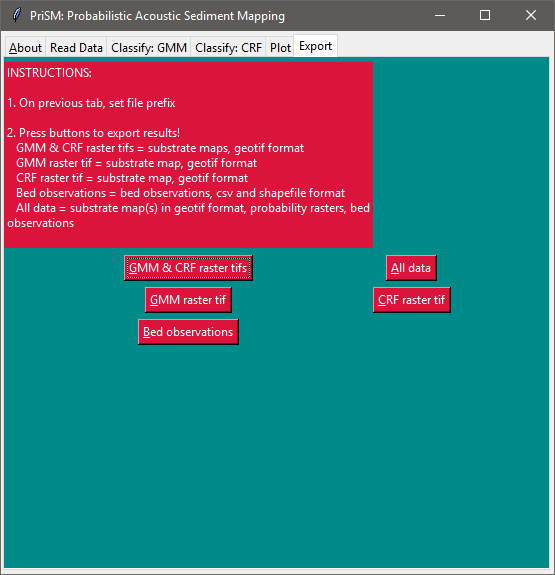
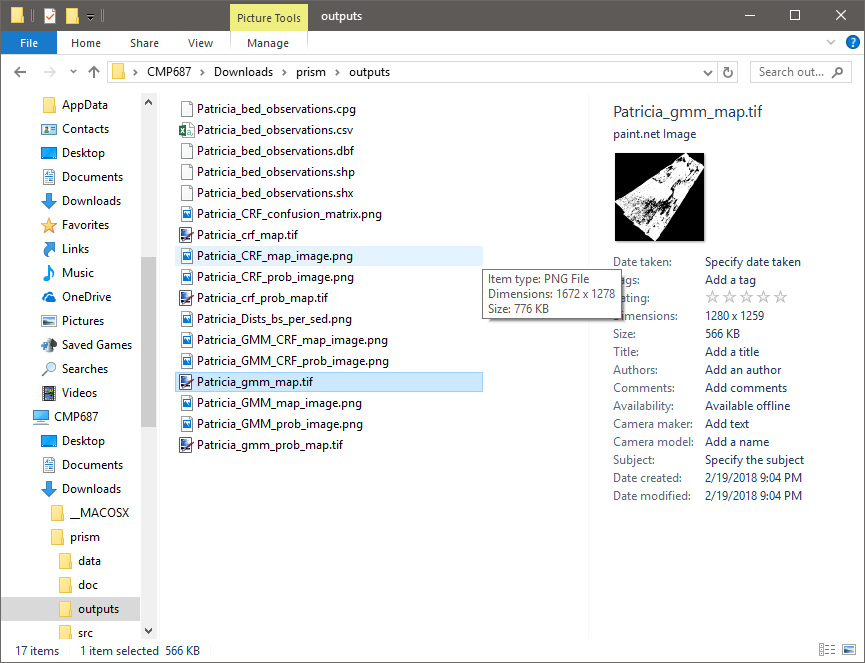


You can even generate plots of confusion matrices to see how well the classifier worked:

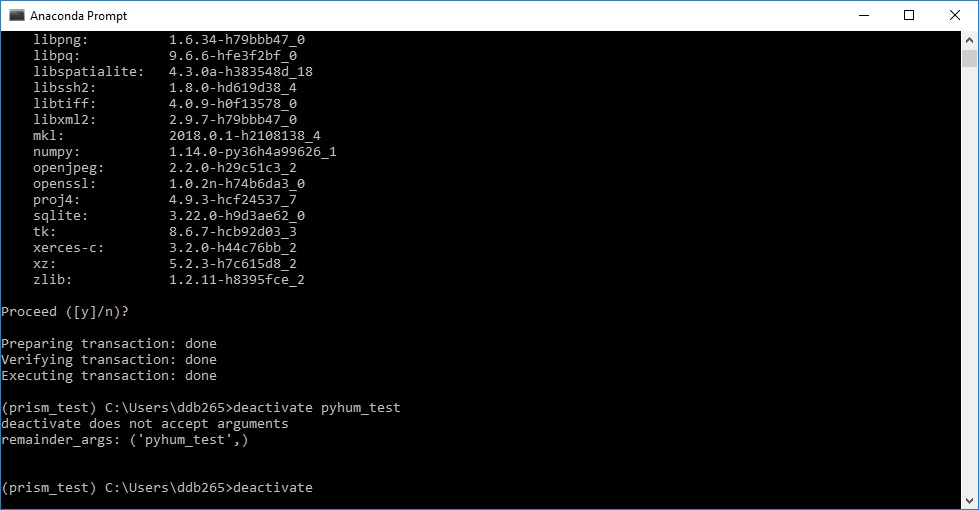


## Export

Finally, this tab is for generating outputs that you may use in GIS or another program. The files are written to the ‘outputs’ folder within the program. It is relatively simple: you may generate maps in geoTIFF format (that may be loaded into a GIS), you may export bed observations within the surveyed multibeam extent as a shapefile (again, that is compatible with GIS) or csv, or you may do both!

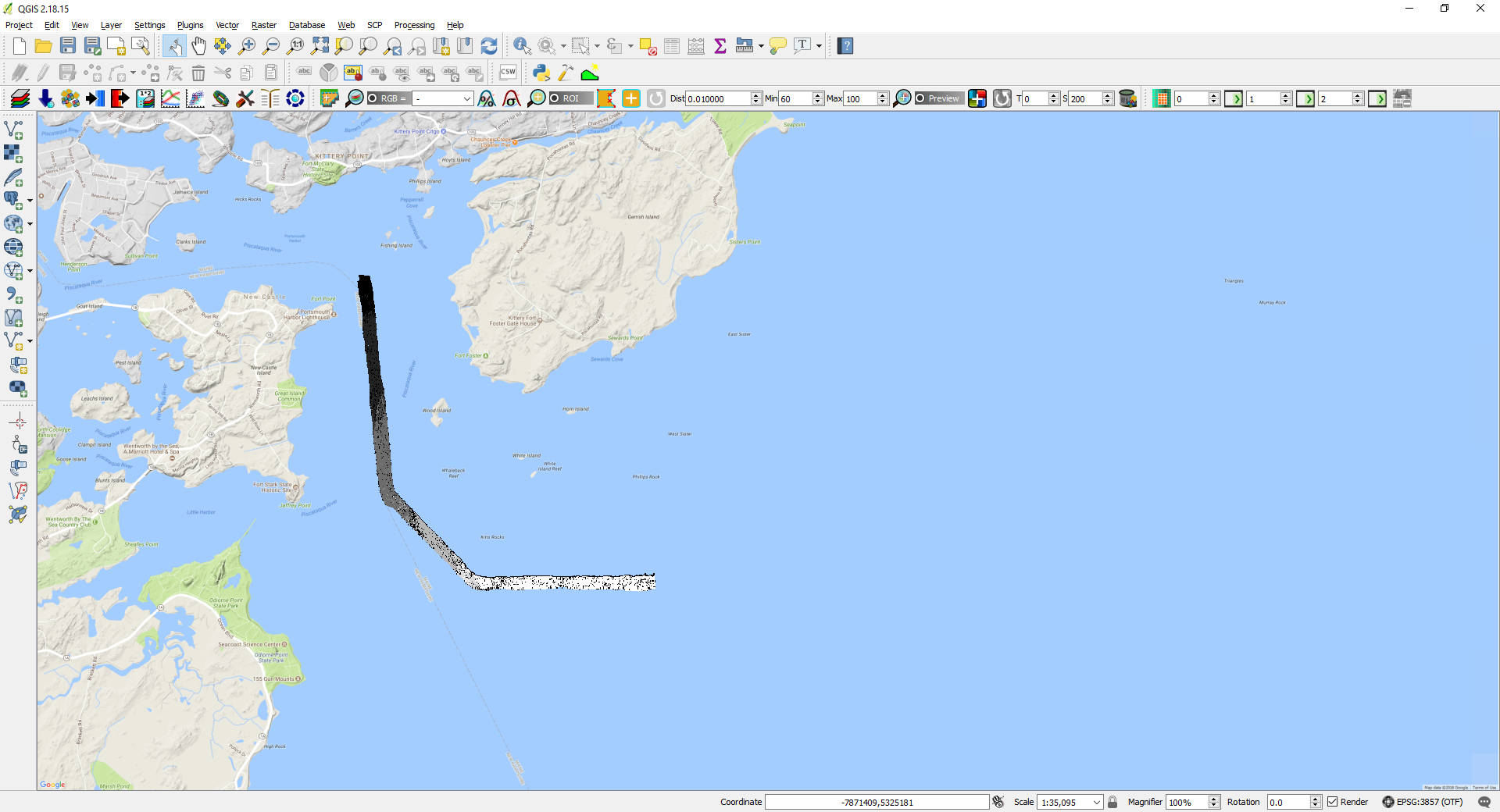
When you are finished, remember to deactivate your conda environment, like so:



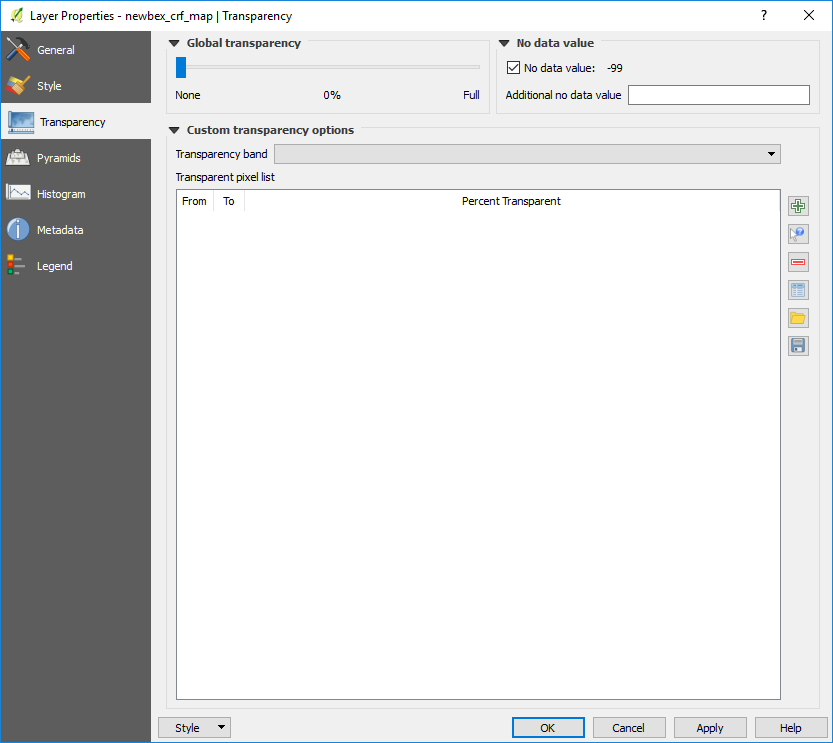
## Using data exports in GIS

### Quick Example Using QGIS

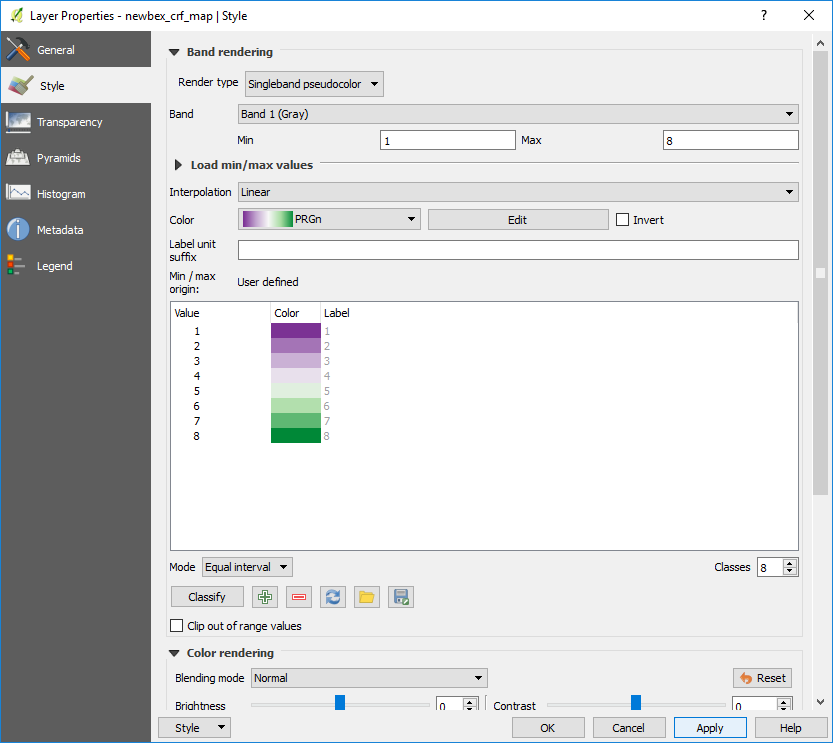
Open a new project and drag GeoTIFFs into the main window. This automatically creates a new layer



Here is the CRF substrate map tiff for the NEWBEX site (lower Portsmouth Harbor, New Hampshire, USA)



Right click the layer in layer panel 🡪 Properties

In the Style tab, render type ‘singleband psuedocolor’, set min and max, choose a color scale, set mode to ‘equal interval’, choose the number of classes to match the number of integer values in the map, and hit Apply

