# Least cost pipeline procedure

## Step 1. Create stacks of biogas cost & cost distance layers

[Create-Cost-Stack.ipynb](https://nbviewer.jupyter.org/github/johnpfay/BioGas_optimization/blob/master/scripts/Create-Cost-Stack.ipynb)

First, for each biogas source, we create two products used in the optimization procedure:

1. A cost layer whereby each pixel's value represents the cost to pipe gas from that source through that pixel. Cost is a factor of the volume produced at the source and the land type through which it is passing.
2. A cost-distance layer whereby each pixel's value represents the accumulated cost to construct pipeline from the biogas source to that pixel's location.

These are produced by iterating through a table of biogas sources, sorted in ascending order of transport cost (). A cost surface is generated (by multiplying the source's transport cost to each value in the MIT cost-factor surface), and then a cost distance surface is generated from that.

The cost and cost-distance layers produced are added to a cost and a cost-distance layer stack, respectively. These stacks are saved as both multi-band geoTIFFs (viewable in ArcGIS) and NumPy export arrays (easily imported into later Python scripts).

## Step 2. Construct pipeline networks

### 2a. Construct the optimal pipeline connecting sources via a single connection point

[LC-Pipeline-Construction-All-NG.ipynb](https://nbviewer.jupyter.org/github/johnpfay/BioGas_optimization/blob/master/scripts/LC-Pipeline-Construction-All-NG.ipynb)

This process builds off of the cost and cost distance stacks generated by Create-Cost-Stack.ipynb as well as a pipeline raster to compute a least cost pipeline configuration connecting biogas sources to the existing NG pipelines network via a single connection point..

The workflow for this process is:

1. Import the stack of cost distance layers (one for each biogas source): .
2. Import the pipeline raster, setting non-pipeline cells to no*data: $arr\*pipeline$.
3. Import the biogas source attributes as a dataframe and sort records by biogas production cost: .
   * *Sorting them on production cost will align them with the layers in the cost stacks.*
4. Determine which pixel among the pipeline pixels has the least cost among all farm cost distances rasters. This will serve as the location of the connection point to the existing pipeline: .
5. Determine which farm is the source of this minimum point, done by finding which layer (in the stack of cost distance rasters) has the minimum value at that location. This represents the least cost biogas source: .
6. Compute the least cost path connecting that farm () to the connection point (): .
7. Update the connection point layer () to include the least cost path (): .
8. Remove the layer associated with from the stack of cost distance rasters () and repeat steps 4-7:
   * Locate the minimum value among all remaining cost distance rasters to cells in the Pipes layer ()...
   * Identify the source farm associated with this minimum ()...
   * Compute the least cost path from from to ...
   * Update the connection point layer ()...

When all layers, i.e. all biogas sources, have been processed, the process will have produced:

* A list of biogas source IDs to link to other attributes ()
* A list of the marginal costs associated with linking each source to the preceding step's pipeline network ()
* A list of arrays depicting the LCP layer for the given biogas source ()
* A list of linestrings (for constructing vector geometries) depicting the path from source to pipeline ()

These products are combined to create a shapefile of the least cost paths from each source to the preceding step's pipeline network. Each segment is tagged with the biogas source ID (Ids), the marginal cost of adding the source to the network (Costs), the accumulated cost of the pipeline network as that source is added (SumCosts), and the order in which the source was added (Order).

### 2b. Construct the optimal pipeline connecting sources via multiple connection points

[LC-Pipeline-Construction-All-NG.ipynb](https://nbviewer.jupyter.org/github/johnpfay/BioGas_optimization/blob/master/scripts/LC-Pipeline-Construction-All-NG.ipynb)

As above, this process also builds off the cost and cost distance stacks generated previously as well as the pipeline raster. Here, however, biogas sources are connected not to a single connection point, but to any location along the existing NG pipeline network. The procedure is quite similar, the key difference being that the initial step of identifying the optimal connection point is omitted.

The workflow is as follows:

1. Import the stack of cost distance layers: .
2. Import the pipeline raster: .
3. Import the biogas source attributes as a dataframe sorted on production costs:
4. Run the following steps until all biogas source layers are processed:
   1. Find the least costly source to connect to the least cost pathway (originally the pipeline network itself) and the least cost connection point along this network.
   2. Find the coordinates of the biogas source corresponding with this least cost pathway.
   3. Construct the least cost path between the source and the identified connection point.
   4. Combine that least cost path with the existing least cost path.
   5. Remove the source's layer from the remaining stack of cost distance layers.

The remaining steps to construct a shapefile from the results are the same as above.

### 2c. Construct a "randomized" pipeline network (in progress)

[LC-Pipeline-Construction-Random.ipynb](https://nbviewer.jupyter.org/github/johnpfay/BioGas_optimization/blob/master/scripts/LC-Pipeline-Construction-Random.ipynb)

Similar to above, but pipelines are added in random fashion vs. added in order of cost. This is to explore the cost savings of developing an optimized network (either to one or to many connection points). Here the biogas sources are put in random order and added incrementally in that order.

## Step 3. View pipeline cost curves (in progress)

[Pipeline-Cost-Curves.ipynb](https://nbviewer.jupyter.org/github/johnpfay/BioGas_optimization/blob/master/scripts/Pipeline-Cost-Curves.ipynb)

Code to visualize the cost differences among the pipeline construction methods. Includes a plot of summed pipeline cost as sites are added to the pipeline network. Also includes plots comparing construction costs to revenues from tapping into sequentially more biogas sources for a given price per MMBtu,