**ENVI RX Anomaly Detection**

**RX Anomaly Detection** This task uses the Reed-Xiaoli Detector (RXD) algorithm to identify the spectral or color differences between a region to test its neighboring pixels or the entire dataset.

**Example Script: Run in IPython using the GBDXTools Interface**

pip install git+<https://github.com/DigitalGlobe/gbdxtools@feature-simpleworkflows>

***The Script can be found here:*** <https://github.com/DigitalGlobe/gbdxtools/blob/feature-simpleworkflows/examples/launch_simple_workflow.py>

from gbdxtools import Interface

gbdx = Interface()

# launch workflow ENVI -> S3

envitask = gbdx.Task("ENVI\_RXAnomalyDetection")

envitask.inputs.task\_name='RXAnomalyDetection'

envitask.inputs.file\_types='til'

envitask.inputs.kernel\_size='3'

data = "s3://receiving-dgcs-tdgplatform-com/054813633050\_01\_003/054813633050\_01/054813633050\_01\_P001\_MUL/"

envitask.inputs.input\_raster=data

workflow = gbdx.Workflow([ envitask ] )

workflow.savedata(envitask.outputs.task\_meta\_data, location='envi\_task\_output')

workflow.savedata(envitask.outputs.output\_raster\_uri, location='envi\_task\_output')

print workflow.execute()

**Description of Input Parameters and Options for the**\*\* "ENVI task":\*\*

**REQUIRED SETTINGS AND DEFINITIONS:**

* S3 location of input data:
  + Required = ‘True’
  + type = ‘directory’
  + name = ‘data’
* Define the ENVITask:
  + Required = ‘True’
  + envitask = gbdx.Task("ENVI\_RXAnomalyDetection")
* Define the Output Directory: (a gbd-customer-data location)
  + Required = ‘True’
  + type = ‘directory’
  + name = "destination"
* Define Stage to S3 location:
  + S3task = gbdx.Task("StageDataToS3",data=” ”,destination=” “)

**OPTIONAL SETTINGS: Required = False**

***The Default setting does not run the specified process. Some of these processes (e.g. "enable\_tiling" = “True”) may have dependencies that also require resetting. Some of the dependencies have “Auto” settings.***

* Comma-separated list of file type extensions, use this to filter input files.
  + Required = false,
  + Type = "string",
  + Name = "file\_types"
* Set this property to true to suppress vegetation anomalies in the RXD results. The options are true or false (default).
  + Default = 'false."
  + Required = false,
  + type = "string"
  + Name = " suppress\_vegetation "
* Select the mean calculation method: Specify one of the values from the CHOICE\_LIST, indicating which mean calculation method to use. Global: Derive the mean spectrum from the full dataset, Local: Derive the mean spectrum from the KERNEL\_SIZE around a given pixel.
  + Required = false,
  + type = "string"
  + name " mean\_calculation\_method "
* Select the RXD method to use. Specify one of the values from the CHOICE\_LIST, indicating which method to use. RXD: Standard RXD algorithm, UTD: Uniform Target Detector algorithm, RXD-UTD: Hybrid of the RXD and UTD algorithms.
  + Required = false
  + Type = "string"
  + Name = " anomaly\_detection\_method "
* Select the kernel size for the analysis. Specify the kernel size in pixels, around a given pixel that will be used to create a mean spectrum. Use an odd number. The minimum value is 3, and the maximum value is (number of columns - 1) less than (number of rows - 1). Specify KERNEL\_SIZE only when using the 'Local' option for MEAN\_CALCULATION\_METHOD.
  + Default is 3.
  + Required = false,
  + type = "string”,
  + name = " kernel\_size "
* GBDX Requirement. Output location for task meta data such as execution log and output JSON.
  + Required = false,
  + type = "directory",
  + name = " task\_meta\_data "

***Sample Output: Test Run 4/26/16, On R&D Cluster***

In [3]: envitask = gbdx.Task("ENVI\_RXAnomalyDetection")

In [4]: envitask.inputs.task\_name='RXAnomalyDetection'

In [5]: envitask.inputs.file\_types='til'

In [6]: envitask.inputs.kernel\_size='3'

In [7]: data = "s3://receiving-dgcs-tdgplatform-com/054813633050\_01\_003/054813633050\_01/054813633050\_01\_P001\_MUL/"

In [8]: envitask.inputs.input\_raster=data

In [9]:

In [9]: workflow = gbdx.Workflow([ envitask ] )

In [10]: workflow.savedata(envitask.outputs.task\_meta\_data, location=' s3://gbd-customer-data/7d8cfdb6-13ee-4a2a-bf7e-0aff4795d927/ENVI/')

In [11]: workflow.savedata(envitask.outputs.output\_raster\_uri, location=' s3://gbd-customer-data/7d8cfdb6-13ee-4a2a-bf7e-0aff4795d927/ENVI/')

In [12]:

In [12]: print workflow.execute()

4320185934977513853

###Postman status @ 16:22 4/26/16

"completed\_time": null,

"state": {

"state": "running",

"event": "started"

},

"submitted\_time": "2016-04-26T22:18:16.035312+00:00",

"owner": "creeder",

"id": "4320185934977513853"

For background on the development and implementation of the RX Anamaly Detection refer to the

[ENVI Documentation](https://www.harrisgeospatial.com/docs/rxanomalydetection.html)