MOD14 Science Processing Algorithm (MOD14_SPA) User's Guide

Version 6.2.1

January 2017

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

MOD14 Science Processing Algorithm

MOD14_SPA

General

The NASA Goddard Space Flight Center's (GSFC) Direct Readout Laboratory (DRL), Code 606.3 developed this software for the International Polar Orbiter Processing Package (IPOPP). IPOPP maximizes the utility of Earth science data for making real-time decisions by giving fast access to instrument data and derivative products from the Suomi National Polar-orbiting Partnership (SNPP), Aqua, and Terra missions and, in the future, the Joint Polar Satellite System (JPSS) mission.

Users must agree to all terms and conditions in the Software Usage Agreement on the DRL Web Portal before downloading this software.

Software and documentation published on the DRL Web Portal may occasionally be updated or modified. The most current versions of DRL software are available at the DRL Web Portal:

http://directreadout.sci.gsfc.nasa.gov/?id=software

Questions relating to the contents or status of this software and its documentation should be addressed to the DRL via the Contact DRL mechanism at the DRL Web Portal:

http://directreadout.sci.gsfc.nasa.gov/?id=dspContent&cid=66

Algorithm Wrapper Concept

The DRL has developed an algorithm wrapper to provide a common command and execution interface to encapsulate multi-discipline, multi-mission science processing algorithms. The wrapper also provides a structured, standardized technique for packaging new or updated algorithms with minimal effort.

A Science Processing Algorithm (SPA) is defined as a wrapper and its contained algorithm. SPAs will function in a standalone, cross-platform environment to serve the needs of the broad Direct Readout community. Detailed information about SPAs and other DRL technologies is available at the DRL Web Portal.

Software Description

This software package contains the Level 2 MODIS Active Fire Product (MOD14) SPA. This algorithm primarily uses brightness temperatures derived from MODIS 4 and 11 micrometer channels (bands 21, 22 and 31) to detect fires. MODIS bands 1, 2, 7 and 32 are used to reject false alarms and to mask clouds. The algorithm reads a MODIS 1-km Level 1B file along with the associated geolocation file and identifies active fires. The output is a two-dimensional fire mask in Hierarchical Data Format (HDF). Additionally the SPA creates a text file containing useful information (latitude, longitude, channel 21/22 brightness temperature, along scan pixel dimension, along track pixel dimension, fire detection confidence, and Fire Radiative Power [FRP]) for each fire pixel detected. The MOD14_SPA functions in two modes: Standalone, or as an IPOPP plug-in.

Software Version

Version 1.1 of the DRL algorithm wrapper was used to package the SPA described in this document. The MOD14 algorithm has been ported from Version 6.2.1 of the MODIS Active Fire Product.

Enhancements to this SPA include:

a) optional extension of processing to include oceans and other large water bodies;

b) additional test to reject false fire pixels caused by small forest clearings;

c) dynamic adjustment of potential fire thresholds;

d) improved cloud mask;

e) improved sun glint rejection.

This software will execute on a 64-bit computer, and has been tested with the following operating systems:

a) Fedora 18 X86_64;

b) CentOS Linux 6.4 X86_64;

c) OpenSUSE Linux 12.1 X86_64;

d) Kubuntu 12.04 X86_64.

Credits

The MODIS Active Fire Product (MOD14) Production Code Version 6.2.1 was provided to the DRL by the MODIS Land Rapid Response Team.

Prerequisites

To run this package, you must have the Java Development Kit (JDK) or Java Runtime Engine (JRE) (Java 1.6.0_25 or higher) installed on your computer, and have the Java installation bin/ subdirectory in your PATH environment variable. This package contains 64-bit binaries statically pre-compiled on an x86-compatible 64-bit computer running under Fedora 14, using gcc 4.5.1.

Program Inputs and Outputs

This SPA uses the MODIS 1-km L1B Calibrated Geolocated Radiances (MOD021KM, MYD021KM) HDF product and MODIS Geolocation HDF product (MOD03, MYD03) as inputs. The SPA output is the MODIS Level 2 Fire product.

Installation and Configuration

This section contains instructions for installing an SPA in a standalone configuration. SPAs may also be installed dynamically into an IPOPP framework; instructions for this type of installation are contained in the IPOPP User's Guide.

Download the MOD14_6.2.1_SPA_1.1.tar.gz and MOD14_6.2.1_SPA_1.1_testdata.tar.gz (optional) files into the same directory.

Decompress and un-archive the MOD14_6.2.1_SPA_1.1.tar.gz and MOD14_6.2.1_SPA_1.1_testdata.tar.gz (optional) files:

```
$ tar -xzf MOD14_6.2.1_SPA_1.1.tar.gz
$ tar -xzf MOD14_6.2.1_SPA_1.1_testdata.tar.gz
```

This will create the following subdirectories:

- SPA
  - mod14
    - algorithm
    - ancillary
    - station
    - testdata
    - testscripts
    - wrapper

Software Package Testing and Validation

The testscripts subdirectory contains test scripts that can be used to verify that your current installation of the SPA is working properly, as described below. Note that the optional MOD14_6.2.1_SPA_1.1_testdata.tar.gz file is required to execute these testing procedures.

Step 1: cd into the testscripts directory.
Step 2: There is a script named run-mod14 inside the testscripts directory.
To run the MOD14 algorithm, use

```
$ ./run-mod14
```

A successful execution usually requires one minute or more, depending on the speed of your computer and the size of the granule. If everything is working properly, the scripts will terminate with a message such as:

```
Output modis.firedetection is /home/ipopp/SPA/mod14/testdata/output/MYD14.07054183325.hdf
Output modis.fireloc.txt is /home/ipopp/SPA/mod14/testdata/output/MYD14.07054183325.txt
```

You can cd to the output directory to verify that the science products exist. Test output product(s) are available for comparison in the testdata/output directory. These test output product(s) were generated on a 64-bit PC architecture computer running Fedora 14. The output products serve as an indicator of expected program output. Use a comparison utility (such as diff, etc.) to compare your output product(s) to those provided in the testdata/output directory. Locally generated files may differ slightly from the provided output files because of differences in machine architecture or operating systems.

If there is a problem and the code terminates abnormally, the problem can be identified using the log files. Log files are automatically generated within the directory used for execution. They start with stdfile* and errfile*. Other log and intermediate files may be generated automatically within the directory used for execution. They are useful for traceability and debugging purposes. However it is strongly recommended that users clean up log files and intermediate files left behind in the run directory before initiating a fresh execution of the SPA. Intermediate files from a previous run may affect a successive run and produce ambiguous results. Please report any errors that cannot be fixed to the DRL.

Program Operation

In order to run the package using your own input data, you can either use the run scripts within the wrapper subdirectories, or modify the test scripts within the testscripts subdirectory.

To Use the Run Scripts

Identify the 'run' scripts: The wrapper directory within this package contains one subdirectory named mod14. The subdirectory contains an executable called 'run'. Execute 'run' within the correct wrapper subdirectory to generate the corresponding product. For instance, the 'run' within wrapper/mod14 is used for creating MOD14 outputs. Note that to execute 'run', you need to have java on your path.

Specify input parameters using <label value> pairs: To execute the 'run' scripts, you must supply the required input and output parameters. Input and output parameters are usually file paths or other values (e.g., an automatic search flag). Each parameter is specified on the command line by a <label value> pair. Labels are simply predefined names for parameters. Each label must be followed by its actual value. Each process has its own set of <label value> pairs that must be specified in order for it to execute. Some of these pairs are optional, meaning the process would still be able to execute even if that parameter is not supplied. The two types of <label value> pairs that the MOD14_SPA uses are:

a) Input file label/values. These are input file paths. Values are absolute or relative paths to the corresponding input file.

b) Output file label/values. These are output files that are produced by the SPA. Values are absolute or relative paths of the files you want to generate.

The following tables contain labels, and their descriptions, required by the MOD14_SPA.

Input File Labels	Description	Source
modis.mxd021km	MODIS 1-km L1B Calibrated Geolocated Radiances HDF file (MOD021KM,MYD021KM)	DRL ftp site for real-time MODIS L1B data over the eastern US region: Terra: ftp://is.sci.gsfc.nasa.gov/gsfcdata/terra/modis/level1/ Aqua: ftp://is.sci.gsfc.nasa.gov/gsfcdata/aqua/modis/level1/
modis.mxd03	MODIS Geolocation HDF file (MOD03,MYD03)	

Output File Labels	Description
modis.firedetection	MODIS L2 Active Fire product
modis.fireloc.txt	Fire Information text file

The Fire Information text file contains the following information:

Column 1: Latitude
Column 2: Longitude
Column 3: Channel 21/22 Brightness Temperature
Column 4: Along Scan Pixel Dimension
Column 5: Along Track Pixel Dimension
Column 6: Fire Detection Confidence
Column 7: Fire Radiative Power

NOTE: Fire Detection Confidence is a percentage value ranging from 0 to 100 percent. The value for FRP is extracted directly from SDS FP_power.

Execute the 'run': The following script shows an example of the command line to run the MOD14 algorithm from the testscripts directory:

```
$ ../wrapper/mod14/run \
modis.mxd021km ../testdata/input/MYD021KM.07054183325.hdf \
modis.mxd03 ../testdata/input/MYD03.07054183325.hdf \
modis.firedetection ../testdata/output/MYD14.07054183325.hdf \
modis.fireloc.txt ../testdata/output/MYD14.07054183325.txt
```

A successful execution usually requires 1 minute or more, depending on the speed of your computer and the size of the granule. If execution fails, you will see an error message indicating the cause of failure (e.g., a file cannot be found, or a label cannot be recognized). Correct it and run again. If the problem has some other cause, it can be identified using the log files. Log files are automatically generated within the directory used for execution. They start with stdfile* and errfile* and can be deleted after execution. Other log and intermediate files may be generated automatically within the directory used for execution. They are useful

for traceability and debugging purposes. However it is strongly recommended that users clean up log files and intermediate files left behind in the run directory before initiating a fresh execution of the SPA. Intermediate files from a previous run may affect a successive run and produce ambiguous results. The 'run' can be executed from any directory the user chooses. This can be done by prefixing it with the file path for the 'run' script.

To Use the Scripts in the testscripts Directory

One simple way to run the algorithms from the directory of your choice using your own data is to copy the run-mod14 script from the testscripts directory to the selected directory. Change the values of the variables like WRAPPERHOME, INPUTHOME and OUTPUTHOME to reflect the file paths of the wrapper directories and the input/output file paths. Then modify the input/output file name variables. Run the script to process your data.