

# The CSPP VIIRS EDR Package

Geoff Cureton, James E. Davies, Kathy Strabala, Scott Mindock,  
Ray Garcia and Hung-Lung Huang

Cooperative Institute for Meteorological Satellite Studies  
Space Science and Engineering Center  
University of Wisconsin - Madison  
Madison, Wisconsin, USA

CSPP/IMAPP Users Group Meeting 2013

# Outline

- 1 CSPP VIIRS-EDR
  - What is CSPP VIIRS-EDR?
  - CSPP VIIRS-EDR Usage
  - VIIRS Cross-Granule Dependency
  - VIIRS Pixel Trim
  - VIIRS Pixel Trim
- 2 CSPP VIIRS-EDR Outputs
  - East Coast Day
  - Gulf Aerosol
  - West Coast Night
  - SST Gulf Stream
  - Alaska
- 3 CSPP VIIRS-EDR Releases/Updates
  - CSPP VIIRS EDR v1.1

# What is CSPP VIIRS-EDR?

- The Community Satellite Processing Package (CSPP) is a collection of scripts, executables, ancillary and auxiliary files used to generate products from polar-orbiter satellite data.
- CSPP VIIRS-EDR package processes Scientific Data Record (SDR) data through to Intermediate Product (IP) and Environmental Data Record (EDR), for the VIIRS imager on Suomi-NPP.
- CSPP is built upon three things: ADL, Python, and bash scripting.

# What is CSPP VIIRS-EDR?

- The Community Satellite Processing Package (CSPP) is a collection of scripts, executables, ancillary and auxiliary files used to generate products from polar-orbiter satellite data.
- CSPP VIIRS-EDR package processes Scientific Data Record (SDR) data through to Intermediate Product (IP) and Environmental Data Record (EDR), for the VIIRS imager on Suomi-NPP.
- CSPP is built upon three things: ADL, Python, and bash scripting.

# What is CSPP VIIRS-EDR?

- The Community Satellite Processing Package (CSPP) is a collection of scripts, executables, ancillary and auxiliary files used to generate products from polar-orbiter satellite data.
- CSPP VIIRS-EDR package processes Scientific Data Record (SDR) data through to Intermediate Product (IP) and Environmental Data Record (EDR), for the VIIRS imager on Suomi-NPP.
- CSPP is built upon three things: ADL, Python, and bash scripting.

# What is CSPP VIIRS-EDR?

- The Community Satellite Processing Package (CSPP) is a collection of scripts, executables, ancillary and auxiliary files used to generate products from polar-orbiter satellite data.
- CSPP VIIRS-EDR package processes Scientific Data Record (SDR) data through to Intermediate Product (IP) and Environmental Data Record (EDR), for the VIIRS imager on Suomi-NPP.
- CSPP is built upon three things: ADL, Python, and bash scripting.

# The Algorithm Development Environment (ADL)

## The Guts...

- Collection of executables, XML and flat binary files which implement a subset of the IDPS.
- Runs on x86 (that means Linux as far as we're concerned).
- Binary file-based data ingest and intermediate output (as opposed to blobs flying around in memory). Incidentally (or not), we call the binary data files BLOBS (Binary Large Objects).
- ADL configuration and product descriptions manifested in a multitude of XML files.

# The Algorithm Development Environment (ADL)

## The Guts...

- Collection of executables, XML and flat binary files which implement a subset of the IDPS.
- Runs on x86 (that means Linux as far as we're concerned).
- Binary file-based data ingest and intermediate output (as opposed to blobs flying around in memory). Incidentally (or not), we call the binary data files BLOBS (Binary Large Objects).
- ADL configuration and product descriptions manifested in a multitude of XML files.



# The Algorithm Development Environment (ADL)

## The Guts. . .

- Collection of executables, XML and flat binary files which implement a subset of the IDPS.
- Runs on x86 (that means Linux as far as we're concerned).
- Binary file-based data ingest and intermediate output (as opposed to blobs flying around in memory). Incidentally (or not), we call the binary data files BLOBS (Binary Large Objects).
- ADL configuration and product descriptions manifested in a multitude of XML files.

# The Algorithm Development Environment (ADL)

## The Guts...

- Collection of executables, XML and flat binary files which implement a subset of the IDPS.
- Runs on x86 (that means Linux as far as we're concerned).
- Binary file-based data ingest and intermediate output (as opposed to blobs flying around in memory). Incidentally (or not), we call the binary data files BLOBS (Binary Large Objects).
- ADL configuration and product descriptions manifested in a multitude of XML files.

# The Algorithm Development Environment (ADL)

## The Guts...

- Collection of executables, XML and flat binary files which implement a subset of the IDPS.
- Runs on x86 (that means Linux as far as we're concerned).
- Binary file-based data ingest and intermediate output (as opposed to blobs flying around in memory). Incidentally (or not), we call the binary data files BLOBS (Binary Large Objects).
- ADL configuration and product descriptions manifested in a multitude of XML files.

# Python Scripting

The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*.

We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Python Scripting

The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Python Scripting

The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Python Scripting

The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Python Scripting

## The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.



# Python Scripting

## The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Python Scripting

## The Glue...

Running an ADL algorithm just requires an executable and an xml file for a particular granule ID. Everything else is just assumed to *be there*. We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input geolocation files to determine processing candidates.
- Granulate dynamic NCEP and static ancillary data using the available geolocation files.
- Create any granule specific XML files required at runtime.
- Parsing of log files looking for error conditions, and general error handling.

# Bash Scripting

## The Wrapper...

### Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment variables in the bash script, in a way probably familiar to the user.

# Bash Scripting

## The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment variables in the bash script, in a way probably familiar to the user.

# Bash Scripting

## The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment variables in the bash script, in a way probably familiar to the user.

# Bash Scripting

## The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment variables in the bash script, in a way probably familiar to the user.

# CSPP VIIRS-EDR Usage

The bare minimum invocation. . .

- Script name
- Bash glob (including path) of input files
- Algorithm (VCM, AOT or SST)
- ... and something to capture the logging output

# CSPP VIIRS-EDR Usage

The bare minimum invocation. . .

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \
```

- Script name
- Bash glob (including path) of input files
- Algorithm (VCM, AOT or SST)
- . . . and something to capture the logging output



# CSPP VIIRS-EDR Usage

The bare minimum invocation...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
      --input_files=/path/to/GMTCO_*.h5 \  
      \
```

- Script name
- Bash glob (including path) of input files
- Algorithm (VCM, AOT or SST)
- ... and something to capture the logging output

# CSPP VIIRS-EDR Usage

The bare minimum invocation. . .

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'
```

- Script name
- Bash glob (including path) of input files
- Algorithm (VCM, AOT or SST)
- . . . and something to capture the logging output

# CSPP VIIRS-EDR Usage

The bare minimum invocation. . .

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM' 2> err.log | tee out.log
```

- Script name
- Bash glob (including path) of input files
- Algorithm (VCM, AOT or SST)
- . . . and something to capture the logging output

# Other Use Cases...

## Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory

# Other Use Cases...

## Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'  --skip_sdr_unpack
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory

# Other Use Cases...

## Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'  --skip_sdr_unpack --skip_ancillary \  
    --skip_ancillary
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory

# Other Use Cases...

## Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'  --skip_sdr_unpack --skip_ancillary \  
    --skip_algorithm
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory

# Other Use Cases...

## Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM'  --skip_sdr_unpack --skip_ancillary \  
    --skip_algorithm --no_chain
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory



## Other Use Cases...

### Extra command line options...

```
bash $CSPP_EDR_HOME/viirs/viirs_edr.sh \  
    --input_files=/path/to/GMTCO_*.h5 \  
    --alg='VCM' --skip_sdr_unpack --skip_ancillary \  
    --skip_algorithm --no_chain --debug
```

- Skip unpacking HDF5 SDR to ADL internal format
- Skip granulation of dynamic and static ancillary to ADL internal format
- Do not run the algorithm(s)
- Do not run prerequisite algorithms for selected algorithm
- Do not clean out working directory

## Other Use Cases...

Use of the extra command line options allow you to...

- Use VIIRS SDR in ADL format from CSPP VIIRS-EDR package, without going through the HDF5 unpacking stage.
- Precompute granulated ancillary data for a pass, for multiple algorithms, without executing the algorithms.
- Repeatedly run algorithm(s) on cached SDR and ancillary data (say, after tweaking some threshold values).
- Perhaps others...

## Other Use Cases...

Use of the extra command line options allow you to...

- Use VIIRS SDR in ADL format from CSPP VIIRS-EDR package, without going through the HDF5 unpacking stage.
- Precompute granulated ancillary data for a pass, for multiple algorithms, without executing the algorithms.
- Repeatedly run algorithm(s) on cached SDR and ancillary data (say, after tweaking some threshold values).
- Perhaps others...

## Other Use Cases...

Use of the extra command line options allow you to...

- Use VIIRS SDR in ADL format from CSPP VIIRS-EDR package, without going through the HDF5 unpacking stage.
- Precompute granulated ancillary data for a pass, for multiple algorithms, without executing the algorithms.
- Repeatedly run algorithm(s) on cached SDR and ancillary data (say, after tweaking some threshold values).
- Perhaps others...

## Other Use Cases...

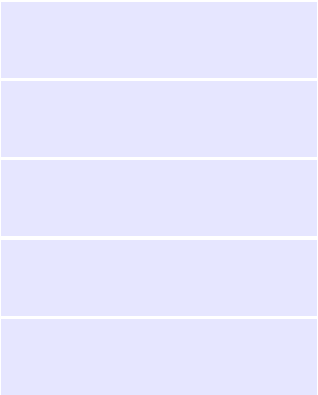
Use of the extra command line options allow you to...

- Use VIIRS SDR in ADL format from CSPP VIIRS-EDR package, without going through the HDF5 unpacking stage.
- Precompute granulated ancillary data for a pass, for multiple algorithms, without executing the algorithms.
- Repeatedly run algorithm(s) on cached SDR and ancillary data (say, after tweaking some threshold values).
- Perhaps others...

# VIIRS Cross-Granule Dependency

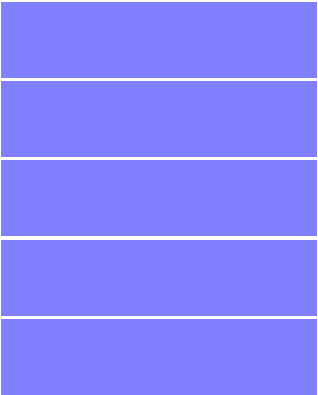


# VIIRS Cross-Granule Dependency



Various VIIRS algorithms require scans from previous and next granules. This complicates processing somewhat (and initially made OPS AOT code unreadable. . .

# VIIRS Cross-Granule Dependency

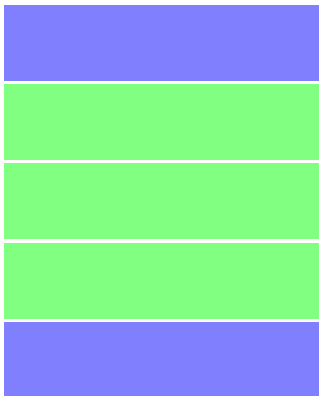


Various VIIRS algorithms require scans from previous and next granules. This complicates processing somewhat (and initially made OPS AOT code unreadable. . .

- SDR



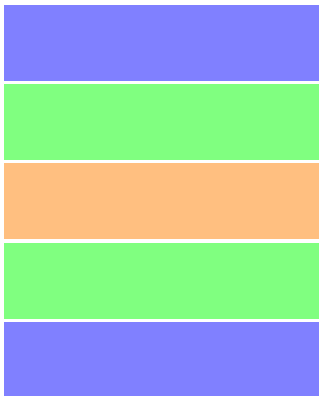
# VIIRS Cross-Granule Dependency



Various VIIRS algorithms require scans from previous and next granules. This complicates processing somewhat (and initially made OPS AOT code unreadable. . .

- SDR
- VCM: +/- 1 SDR granules.

# VIIRS Cross-Granule Dependency



Various VIIRS algorithms require scans from previous and next granules. This complicates processing somewhat (and initially made OPS AOT code unreadable. . .

- **SDR**
- **VCM**: +/- 1 SDR granules.
- **AOT/SST**: +/- 1 VCM granules.

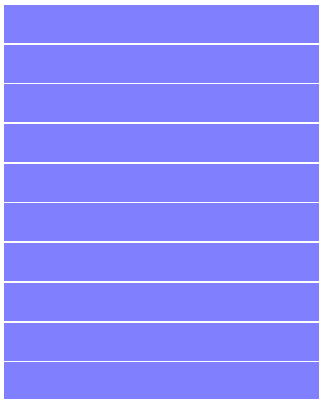
# VIIRS Cross-Granule Dependency



# VIIRS Cross-Granule Dependency

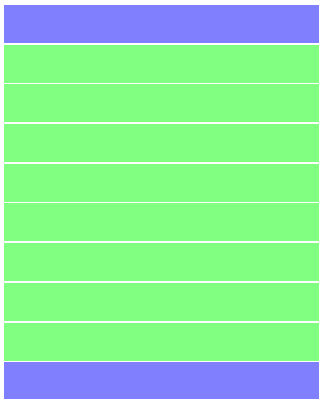
This time, we have a longer pass...

# VIIRS Cross-Granule Dependency



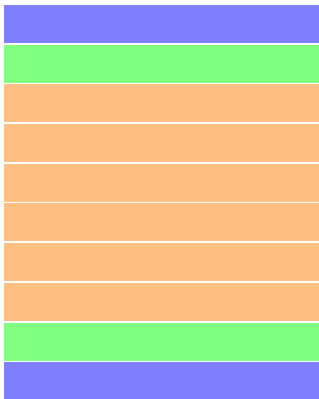
This time, we have a longer pass...  
SDR

# VIIRS Cross-Granule Dependency



This time, we have a longer pass...  
SDR , VCM

# VIIRS Cross-Granule Dependency



This time, we have a longer pass...  
SDR , VCM and AOT/SST.

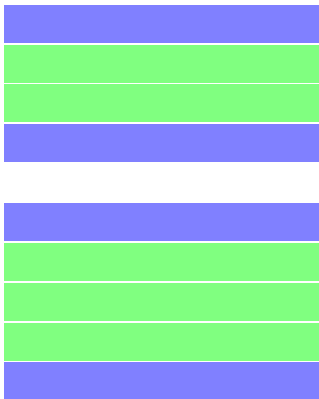
# VIIRS Cross-Granule Dependency

This time, we have a longer pass...  
SDR, VCM and AOT/SST.

- What happen if we have a missing granule in the SDR?



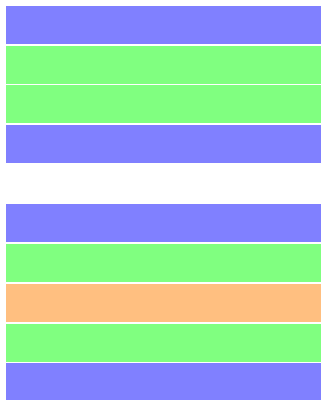
# VIIRS Cross-Granule Dependency



This time, we have a longer pass...  
SDR, VCM and AOT/SST.

- What happen if we have a missing granule in the SDR?
- CSPP EDR determines the allowable VCM granules...

# VIIRS Cross-Granule Dependency



This time, we have a longer pass...  
SDR, VCM and AOT/SST.

- What happen if we have a missing granule in the SDR?
- CSPP EDR determines the allowable VCM granules...
- ... and AOT/SST granules.

# VIIRS Pixel Trim



# VIIRS Pixel Trim

VIIRS SDR has selected observations deleted on the spacecraft (Suomi-NPP) to reduce bandwidth requirements.

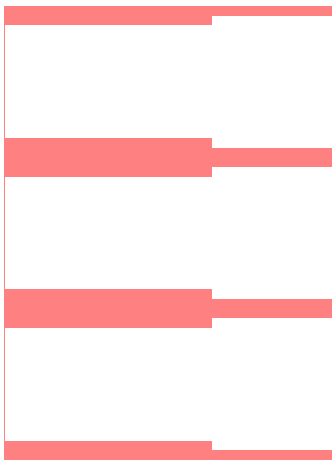
# VIIRS Pixel Trim

VIIRS SDR has selected observations deleted on the spacecraft (Suomi-NPP) to reduce bandwidth requirements. For granulated ancillary, and algorithms which depend on the SDRs, additional pixels are not processed, as they correspond to redundant observations near the edge of scans.

# VIIRS Pixel Trim

VIIRS SDR has selected observations deleted on the spacecraft (Suomi-NPP) to reduce bandwidth requirements. For granulated ancillary, and algorithms which depend on the SDRs, additional pixels are not processed, as they correspond to redundant observations near the edge of scans. Shown are three scans of a VIIRS granule (scan edge at left) . . .

# VIIRS Pixel Trim



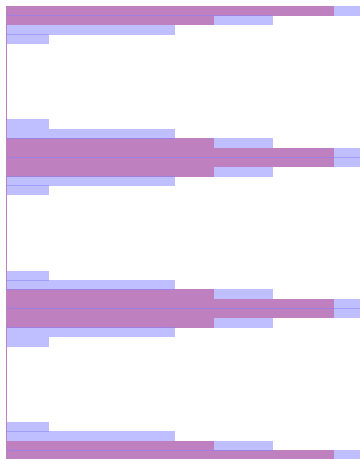
VIIRS SDR has selected observations deleted on the spacecraft (Suomi-NPP) to reduce bandwidth requirements.

For granulated ancillary, and algorithms which depend on the SDRs, additional pixels are not processed, as they correspond to redundant observations near the edge of scans.

Shown are three scans of a VIIRS granule (scan edge at left) . . .

- Onboard pixel trim.

# VIIRS Pixel Trim



VIIRS SDR has selected observations deleted on the spacecraft (Suomi-NPP) to reduce bandwidth requirements.

For granulated ancillary, and algorithms which depend on the SDRs, additional pixels are not processed, as they correspond to redundant observations near the edge of scans.

Shown are three scans of a VIIRS granule (scan edge at left) . . .

- Onboard pixel trim.
- Onground pixel trim.



# VIIRS Pixel Trim



# VIIRS Pixel Trim

Some things to remember when poking around in VIIRS SDRs and derived products. . .

# VIIRS Pixel Trim

Some things to remember when poking around in VIIRS SDRs and derived products. . .

- Pixel trim pixels have specific values depending on the datatype, documented in JPSS documentation. Definition of pixel trim is a defined pattern on a per-scan basis.

# VIIRS Pixel Trim

Some things to remember when poking around in VIIRS SDRs and derived products. . .

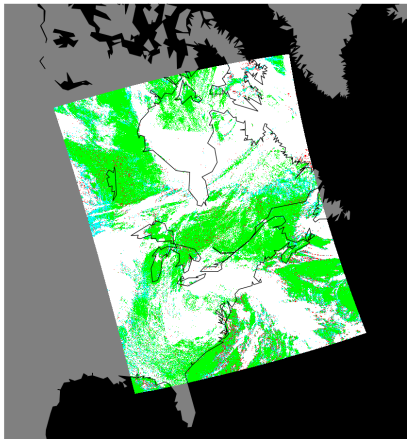
- Pixel trim pixels have specific values depending on the datatype, documented in JPSS documentation. Definition of pixel trim is a defined pattern on a per-scan basis.
- Unscaling of scaled data should then pixel trim mask applied with fill value appropriate to the unscaled datatype. . . search and replace based on value will get you in trouble.

# VIIRS Pixel Trim

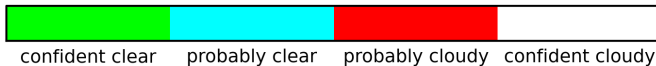
Some things to remember when poking around in VIIRS SDRs and derived products. . .

- Pixel trim pixels have specific values depending on the datatype, documented in JPSS documentation. Definition of pixel trim is a defined pattern on a per-scan basis.
- Unscaling of scaled data should then pixel trim mask applied with fill value appropriate to the unscaled datatype. . . search and replace based on value will get you in trouble.
- Don't assume that a particular VIIRS product will have pixel trim set to an appropriate fill value distinct from valid data values. . . more on this later.

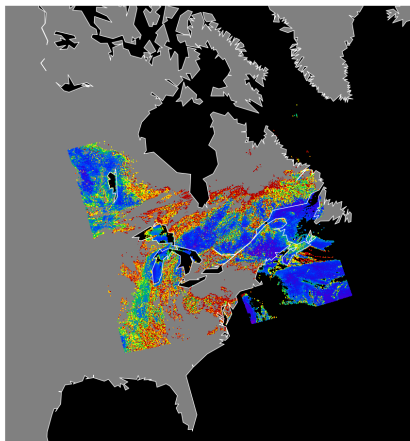
# VIIRS Cloud Mask - CONUS



VCM

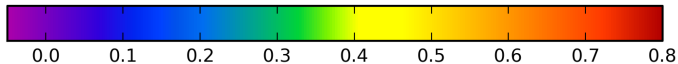


# VIIRS Aerosol Optical Thickness - CONUS

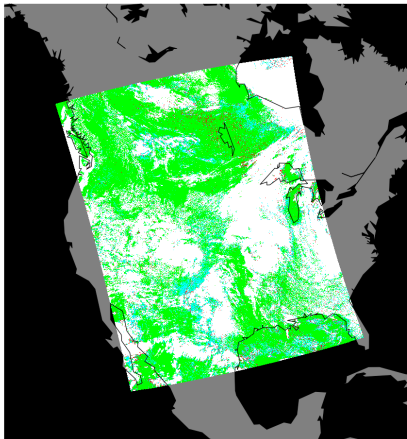


AOT

AOT



# VIIRS Cloud Mask - CONUS

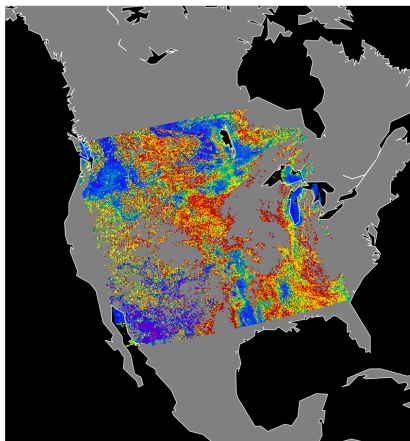


VCM



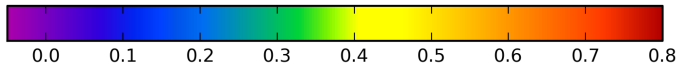


# VIIRS Aerosol Optical Thickness - CONUS

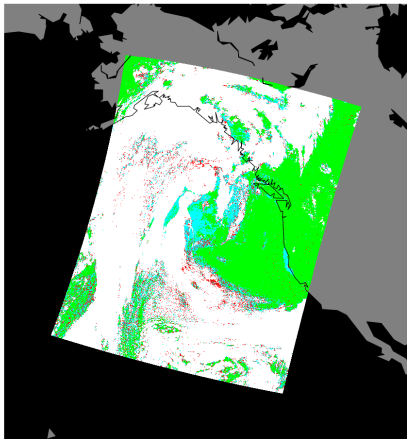


AOT

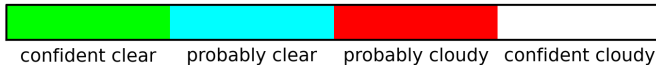
AOT



# VIIRS Cloud Mask - CONUS



VCM

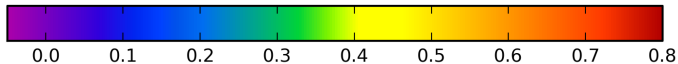


# VIIRS Aerosol Optical Thickness - CONUS

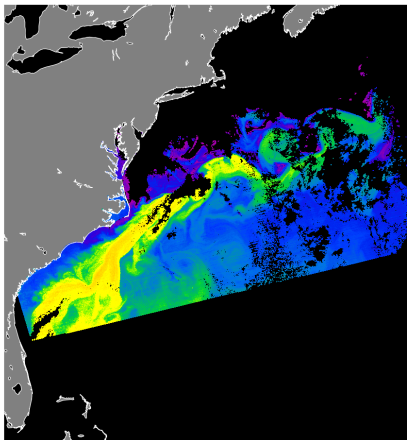


AOT

AOT

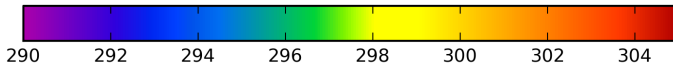


# VIIRS Sea Surface Temperature - CSPP

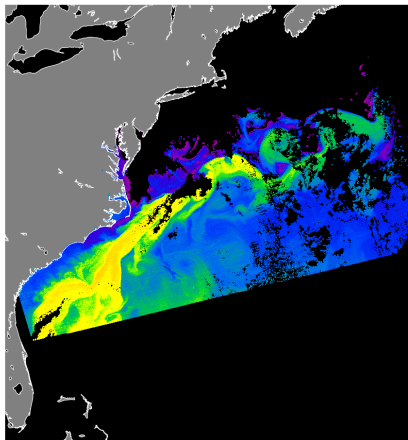


CSPP SST  
2013-05-15 1723h

Sea Surface Temperature (K)

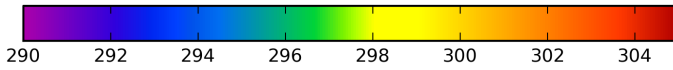


# VIIRS Sea Surface Temperature - OPS

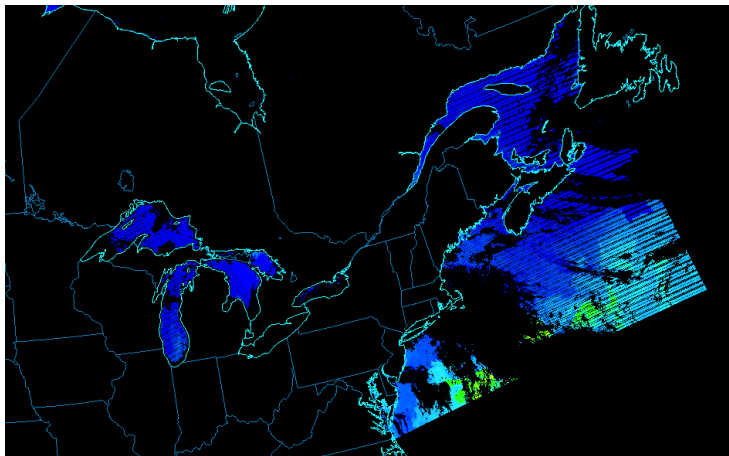


NOAA OPS SST  
2013-05-15 1723h

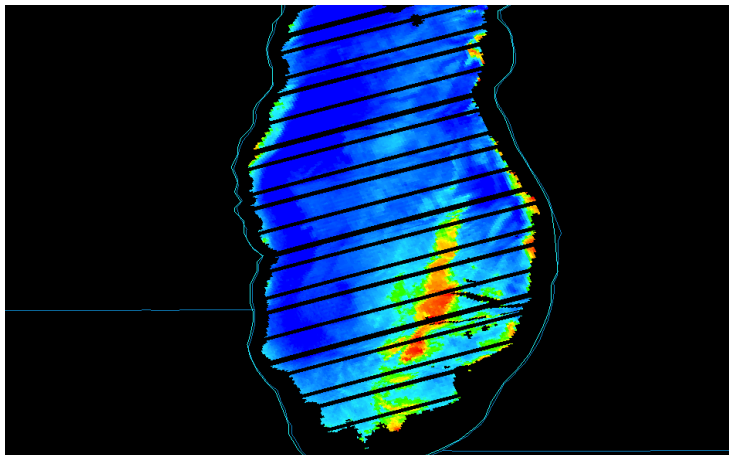
Sea Surface Temperature (K)



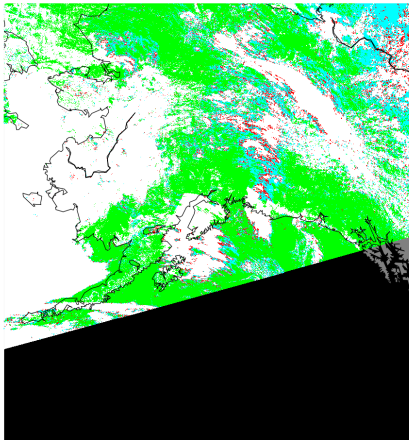
# VIIRS Lake Surface Temperature - CSPP



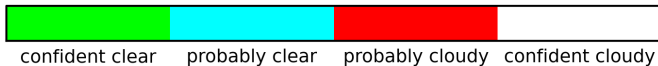
# VIIRS Lake Surface Temperature - CSPP



# VIIRS Cloud Mask - CSPP

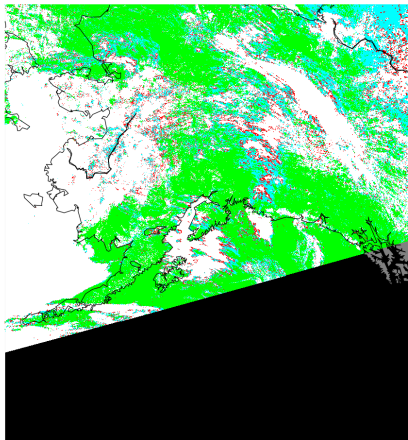


Alaska DB VCM  
npp.13119.2229

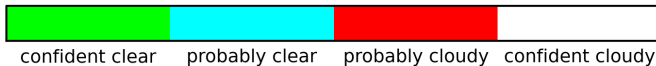




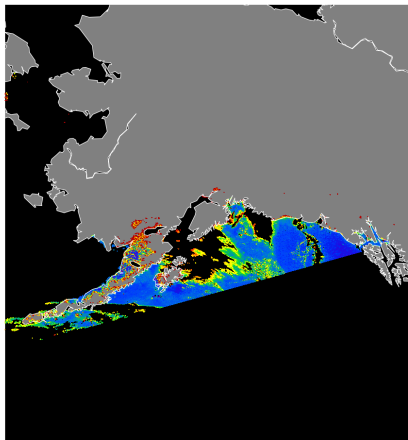
# VIIRS Cloud Mask - OPS



NOAA OPS VCM  
OPS.npp.13119.2229

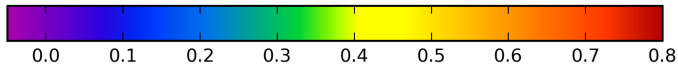


# VIIRS Aerosol Optical Thickness - CSPP

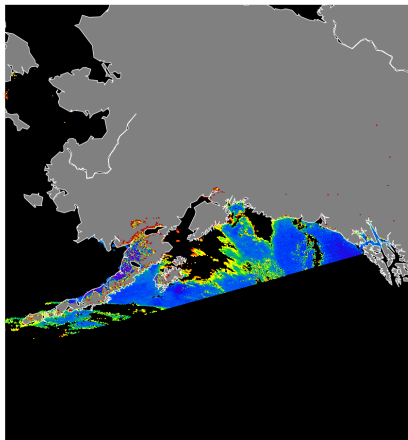


Alaska DB AOT  
npp.13119.2229

AOT

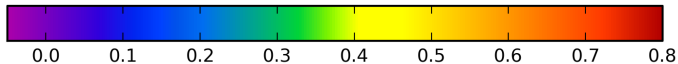


# VIIRS Aerosol Optical Thickness - OPS



NOAA OPS AOT  
OPS.npp.13119.2229

AOT



# CSPP VIIRS EDR v1.1

Since v1.0, we have...

- Added AOT and SST algorithms.
- Fixed a high latitude geolocation problem.
- NCEP temporal geolocation..

# CSPP VIIRS EDR v1.1

Since v1.0, we have...

- Added AOT and SST algorithms.
- Fixed a high latitude geolocation problem.
- NCEP temporal geolocation..