SeaDAS Workshop

Ocean Optics XXII

26 October 2014

Portland, Maine

Course Agenda

- Morning Session 1 – 9:00am – 10:30am
  - SeaDAS Overview
  - Graphical User Interface Introduction
  - Load product, display band, manipulate color
  - Masks, geometries and shapefiles
- Moring Session 2 – 10:45am – 12:00pm
  - Introduction to layers
  - Math Band operator and Expression Editor
  - Reprojection, collocation, subset creation
- Afternoon Session 1 – 1:00pm – 2:30pm
  - "Shiptrack" and Analysis functions
  - Layout, tool bar customization, session management
- Afternoon Session 2 – 2:45pm – 4:00pm
  - Data Processing
    - GUI based
    - Command line based
    - Scripting
    - GPT

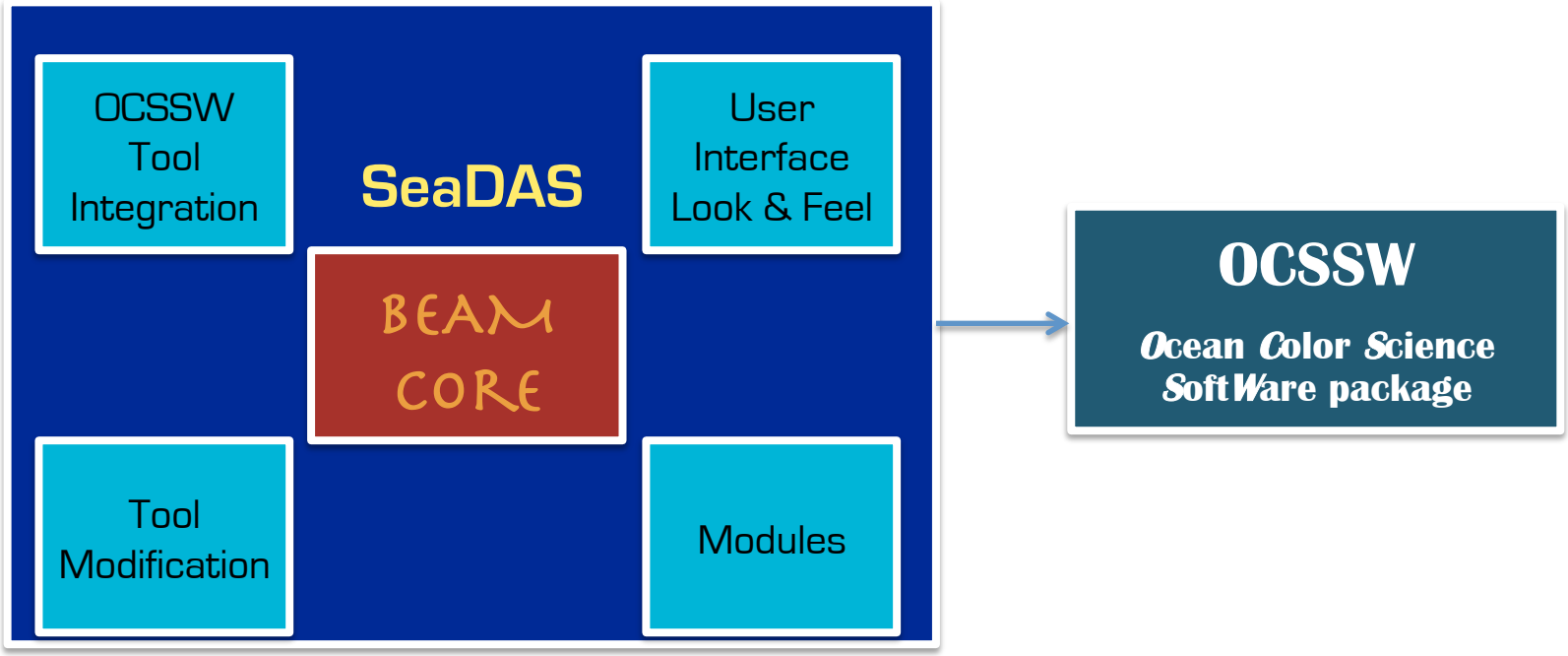
SeaDAS Overview

SeaDAS is a comprehensive open-source toolbox and development platform for viewing, analyzing and processing of remote sensing data.

- over 15 years as a tool for the ocean color community
- it has two primary components, visualization and science data processing that can exist independently
- original GUI package development used IDL

GUI redesign began in 2009 based on the BEAM, with extensions that provide the functionality provided by previous versions of SeaDAS

- first official release April 2013



BEAM

BEAM = Basic Envisat AASTR/MERIS toolbox

- originally developed to facilitate the utilization of image data from Envisat's optical instruments.
- now supports a growing number of other raster data formats:
  - GeoTIFF
  - NetCDF
  - HDF-EOS
  - Additional targeted support for specific data formats of other EO sensors such as MODIS, AVHRR, AVNIR, PRISM and CHRIS/Proba.
  - BEAM-DIMAP (Digital Image Format)
    - Originally developed to support the SPOT mission
      - data are stored as a single product header file with the suffix .dim in XML format containing the product meta-data
      - an additional directory with the same name plus the suffix .data containing ENVI®-compatible images for each band.
    - Chosen for its simplicity and portability

SeaDAS Highlights

- Very **fast image display and navigation** even of giga-pixel images
- Advanced **layer management** allows adding and manipulation of new overlays such as images of other bands, images from WMS servers or ESRI shapefiles
- Rich **region-of-interest** definitions for **statistics** and various **plots**
- Easy **bitmask** definition and overlay
- Flexible **band arithmetic** using arbitrary mathematical expressions
- Accurate **reprojection** and **ortho-rectification** to common map projections
- Geo-coding and rectification using **ground control points**
- Store and restore the current **session** including all opened files, views and layers

SeaDAS Features

- Integration of in-situ data and tools for analysis
  - Correlative plot
  - Transect plot
  - Profile Analysis
  - Time series
- Analyzing parameterized subsets (BEAM's mask concept)
  - Flagging
  - Creating masks as needed
  - Masks as tools for regional analyses
- Extraction of information
  - Pixel extraction of time series and match-ups
- Working with raster data processors

SeaDAS Features

- Visual Data Inspection
  - Manipulate colors to enhance visual representation
- Geo-Location
  - Display bands of multiple products in one view if they are in the same coordinate reference system
- Map Projection
  - Reproject data products
- Band Arithmetic
  - User defined expression for band combinations

BEAM Data Processors

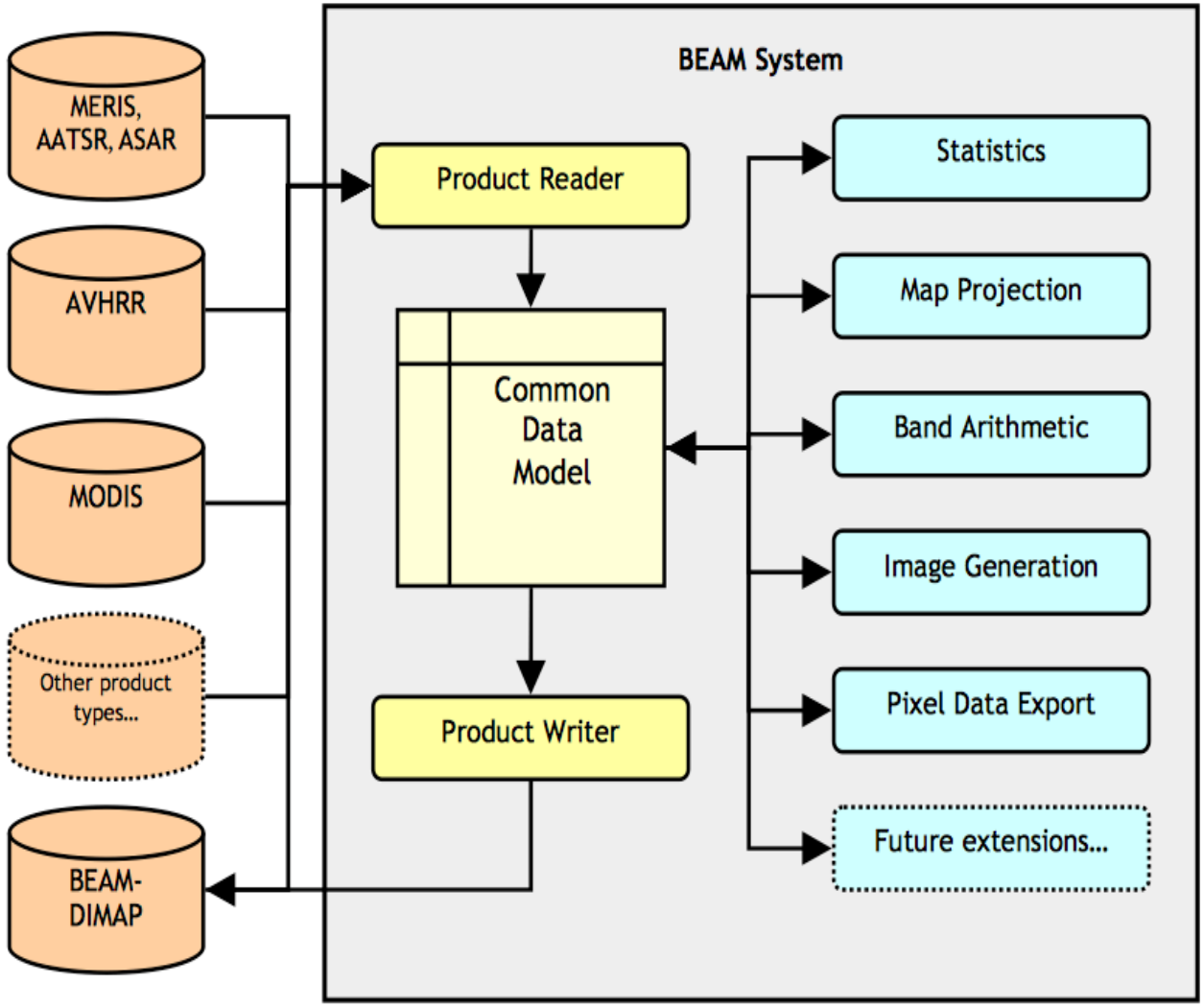
A standard set of **scientific data processors** is provided by BEAM, including:

- Level 3 Binning* and Mosaicing (all sensors)
- Collocation (all sensors)
- EM and K-Means Clustering, Linear Spectral Unmixing (all sensors)
- Radiance-to-Reflectance, Smile Effect Correction, Cloud Probability, SMAC Atmospheric Correction, Case 2 Water Constituents (MERIS) ⓘ
- Sea Surface Temperature (AATSR) ⓘ
- FLH/MCI, NDVI (MERIS) ⓘ

**a JAVA implementation the replicates the functionality of the OCSSW l3bin program*

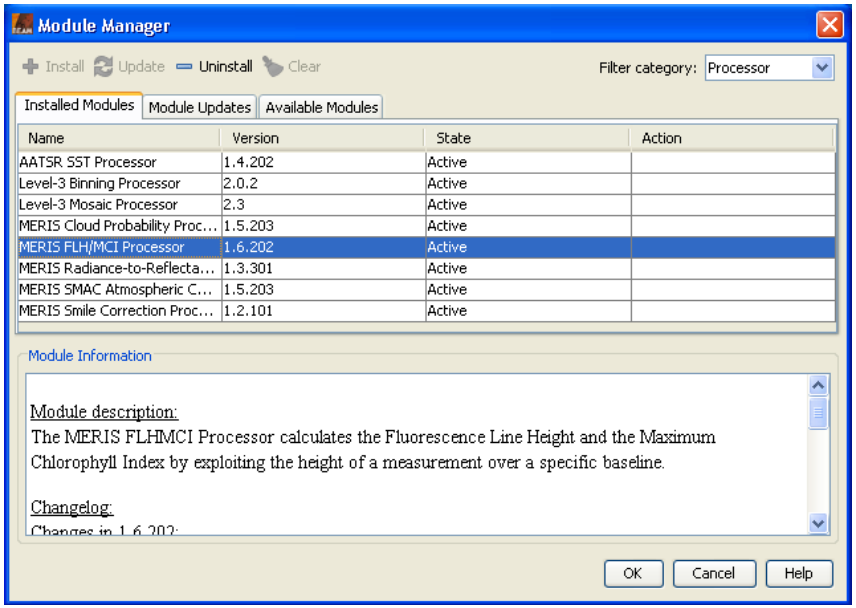
ⓘ Not included with default SeaDAS installation, but available via Module Manager

BEAM Generic Product Model Concept



Module Management

- SeaDAS (and BEAM) are extensible with module plug-ins
- SeaDAS can load BEAM modules
  - (and BEAM can load SeaDAS modules)



Satellite Data Primer

- Level 0 data

  unprocessed instrument/payload data at full resolution. Any artifacts of the communication (e.g. synchronization frames, communication headers) of these data from the spacecraft to the ground station have been removed. These data are (usually) the most raw format available.

- Level 1A data

  reconstructed, unprocessed instrument data at full resolution, time-referenced and annotated with ancillary information including radiometric and geometric calibration coefficients and georeferencing parameters (e.g. platform attitude and ephemeris data)

- Level 1B data

  Level 1A data that have had instrument/radiometric calibrations applied.

- Level 2 data

  derived geophysical variables at the same resolution as the source Level 1 data.

- Level 3 data

  derived geophysical variables that have been aggregated/projected onto a well-defined spatial grid over a well-defined time period

  - Binned

    Each Level 3 binned data product consists of the accumulated data for all L2 products in a product suite, for the specified instrument and resolution, corresponding to a period of time (e.g. daily, 8 days, monthly, etc.) and stored in a global, nearly equal-area, integerized sinusoidal grid (ISIN)

  - Mapped

    Created from the corresponding Level 3 binned products – Each Standard Mapped Image (SMI) file contains a Plate Carrée, pixel-registered grid of floating-point values (or scaled integer representations of the values) for a single geophysical parameter

GUI Introduction

File Manager

- Displays loaded files in a tree view
  - Metadata
  - Flag codings
  - Vector data
  - Bands (products)
- Enables selection of a band
- Convenient right-click options
  - Band properties and renaming
  - Vector export

Image View

- Screen interactive modes
  - Pan & zoom (default)
  - Selection
  - Rectangle zooming tool
- Auto-zoom options
  - Match screen resolution
  - Fill window
- Multiple image views
  - Window tiling
  - Synchronized boundaries/zoom
  - Show synchronized cursor
  - Windows are not floatable
- Stores/contains layers
- Stores/contains mask visibility
- Image rotation
- Image export

Color Schemes

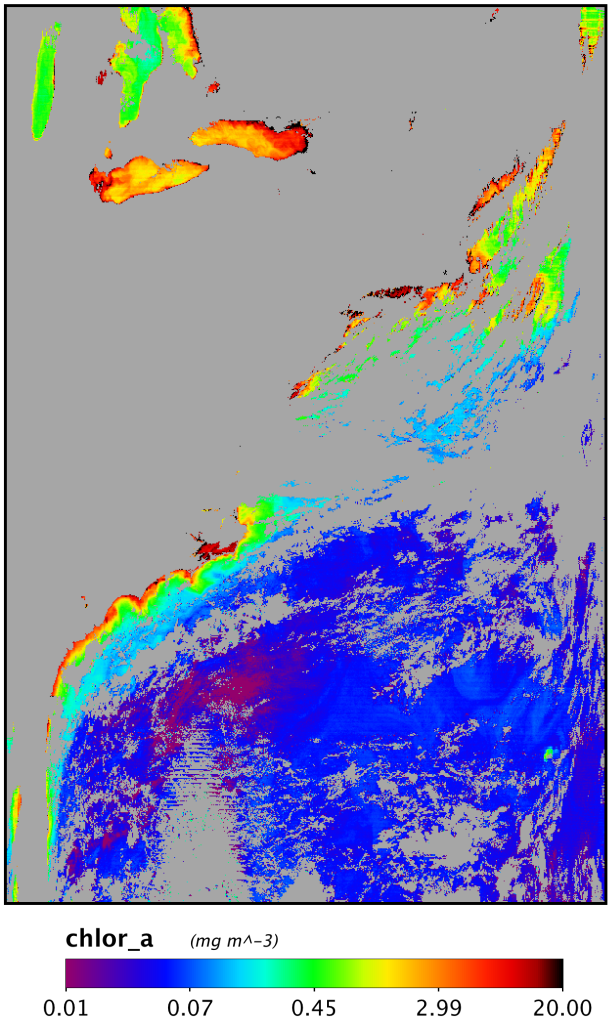
- Color Scheme
  - Color palette (.cpd file)
  - Value range (proportional distribution)
  - Log/linear scaling

- Band dependent defaults (keyed on band name)
- User selectable/definable schemes
- Create/modify palettes

- Scale value range to band statistics
  - Histogram displayed over color bar
  - Quickview "rough" vs full statistics

- Scheme and palette files location
  - ~/.seadas/beam-ui/auxdata/color-palettes

- Discrete colors mode
- Stored in band data

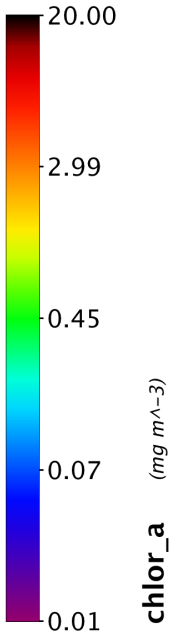


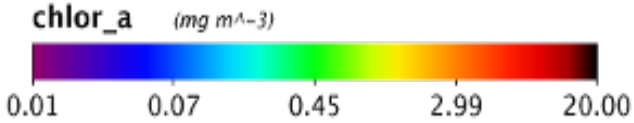
Color Bar

- General Options
  - Title and Units formatting
  - Dimensions
  - Transparency
  - Orientation (vertical, horizontal)
  - Border (width, color)

- Point distribution
  - Even (by tick mark count)
  - Manual (displays string exactly as typed)
  - Exact (displays all palette points)

- Data scaling option
- Export color bar to file
- Create color bar layer (*currently raster)
- Settings stored in band data



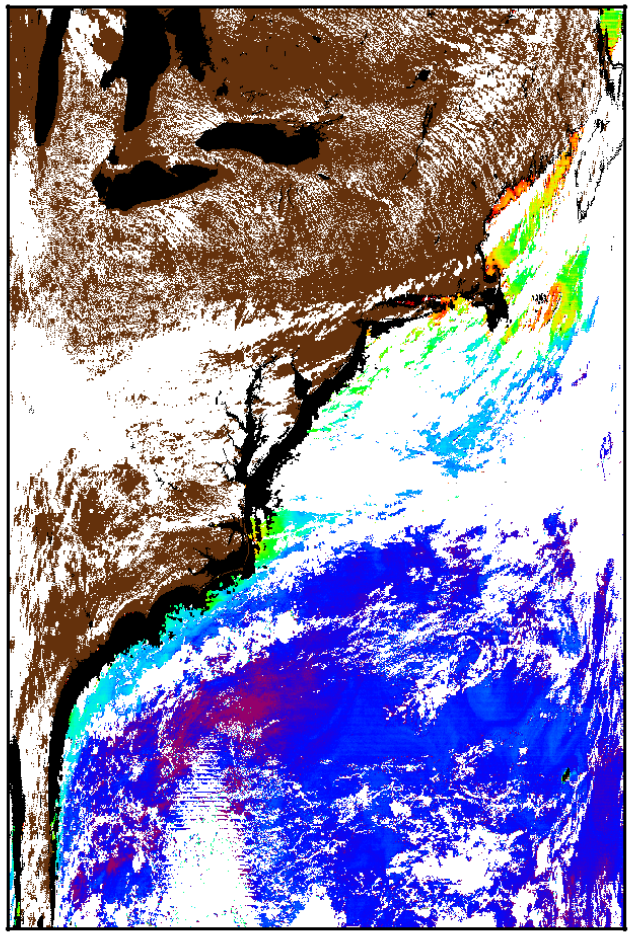


Exercise 1

- Launch SeaDAS
- Open File <filename>
- Display Band: Chlorophyll (chlor_a)
- Modify color scheme
- Create Color bar layer
- Export Color bar

Masks

- Masks
  - l2flags
  - Geometries
  - Logical math expression
  - Coastline
  - Land
  - Bathymetry
- Region of Interest (ROI) Mask
- Mask Manager Tool
  - add/modify/delete
  - show/hide
  - color, transparency
- Stored in file
  - shared by all bands

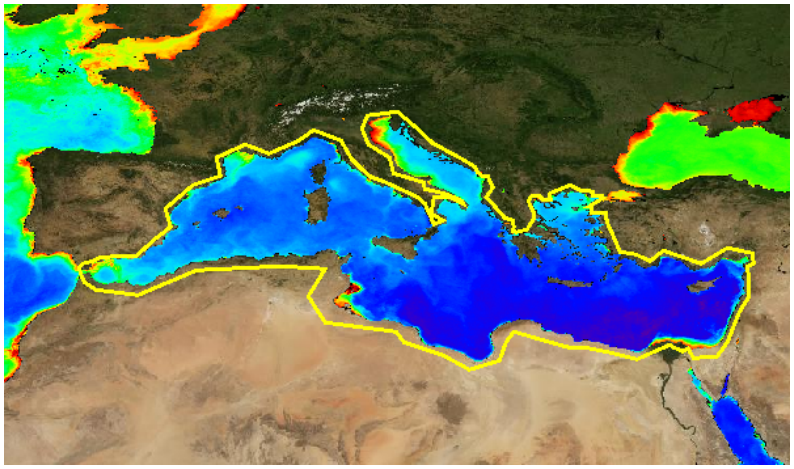


Chlorophyll level-2 image, masks displayed:
COASTZ (black), CLDICE (white), LAND (brown)

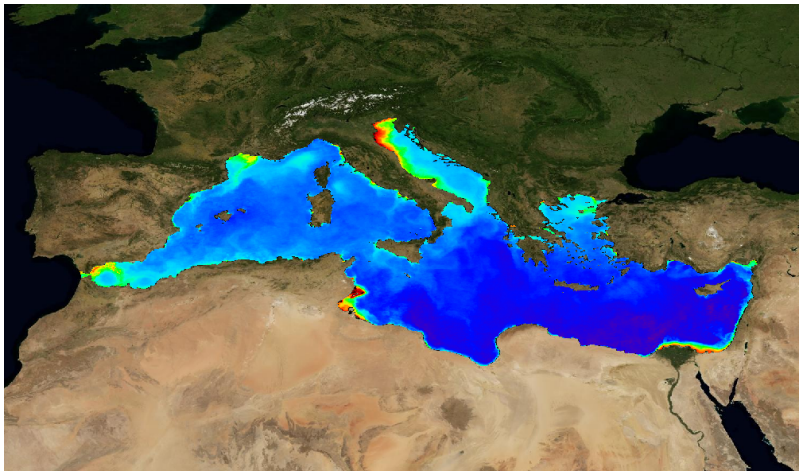
Geometries

- Interactive drawing/editing tools
  - Rectangle
  - Ellipse
  - Polygon
  - Line
  - Polyline

- Import/Export shapefiles
- Mask automatically created
- Stored in geometry containers
- Create geometries with coordinate points



Chlorophyll with polygon geometry

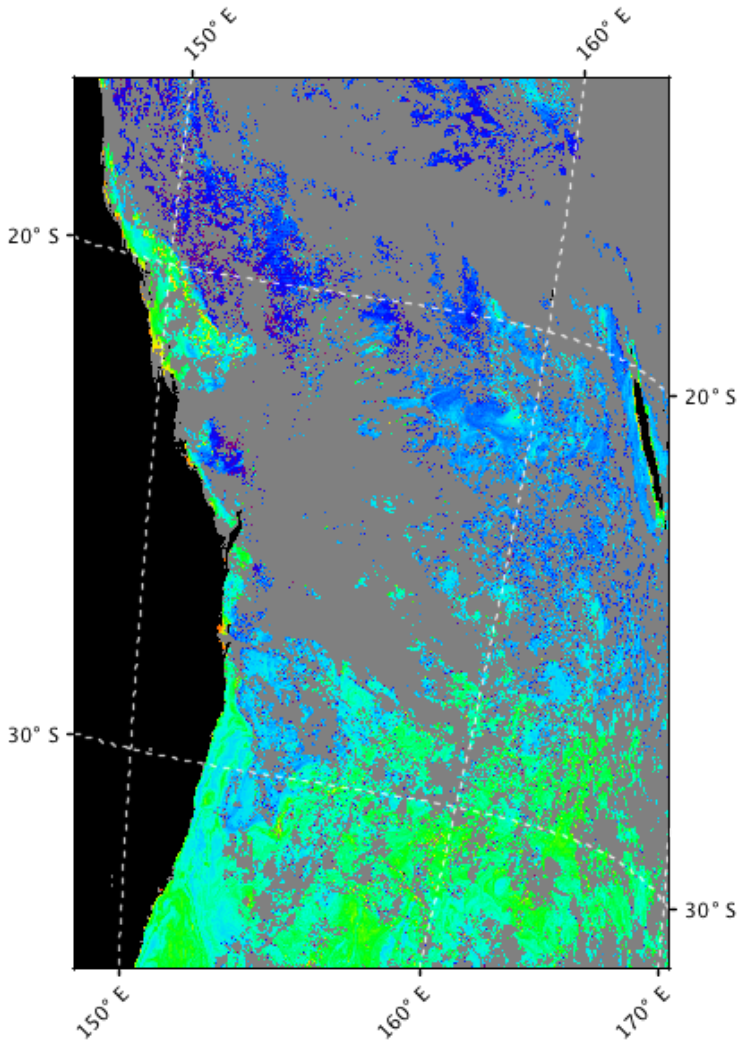


ROI masked Chlorophyll (using above geometry)

Map Gridlines

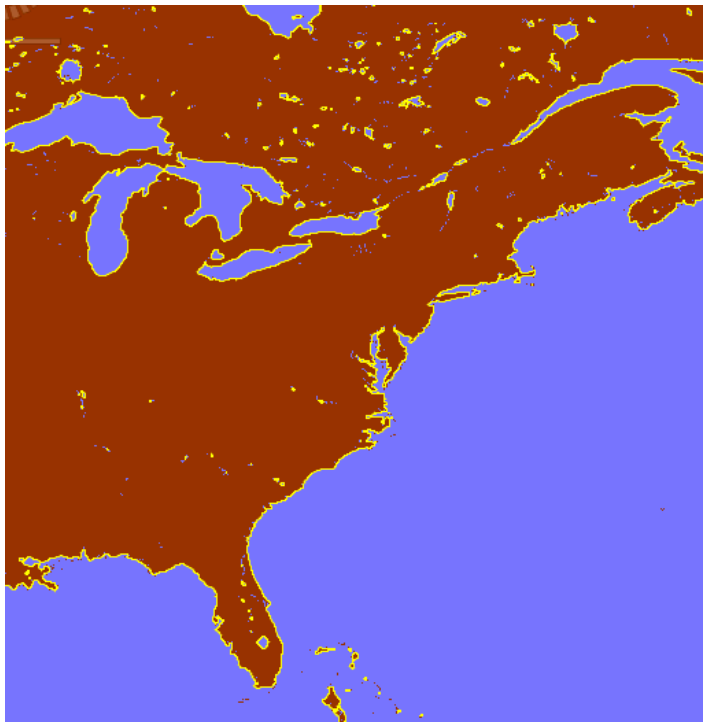
- Options editable with Layer Editor Tool
  - Auto/manual line spacing
  - Labels
    - location, size, angle, color
  - Tick marks
    - location, size, color
  - Gridlines
    - width, dashed/solid, transparency, color
  - Border
    - width, color

- Stored independently in each image view
- Limited preferences



Coastline & Land Masks

- Creates 2 bands
  - mask_data_water_fraction
  - mask_data_water_fraction_smoothed
    - 3x3 Filtered band
- Creates 3 masks
  - CoastLine
  - LandMask
  - WaterMask
- Source Resolutions
  - (10km, 1km, 150m*, 50m*)
- Supersampling
- Coastline contour
  - create a contour vector using "mask_data_water_fraction_smoothed" at value level = 50



LandMask (brown),
WaterMask (blue),
coastline contour (yellow)

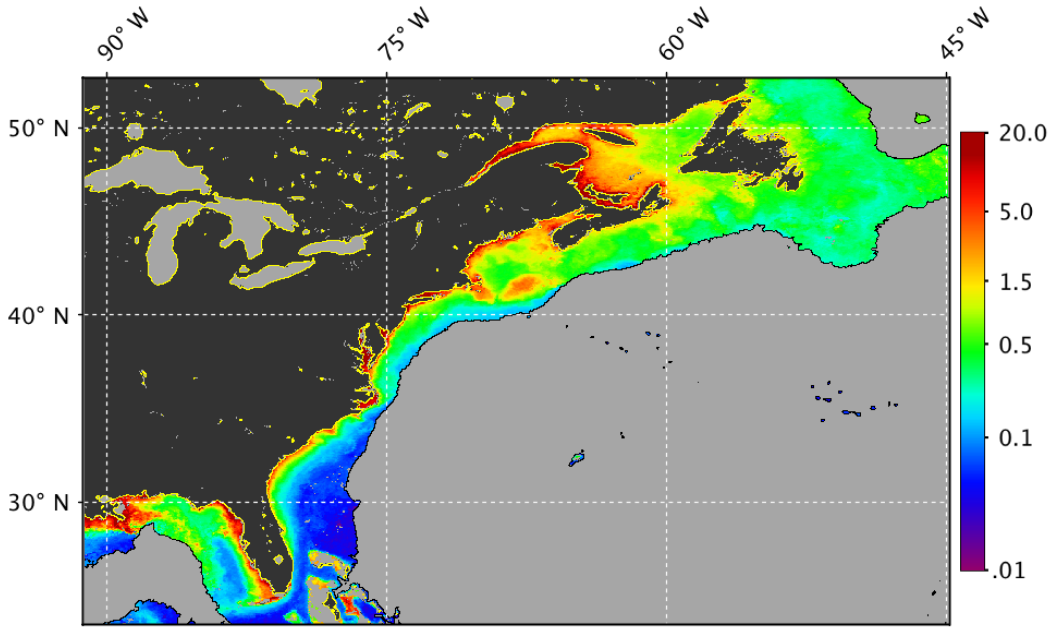
Exercise 2

- Display L2 flags as mask overlays
- Add gridlines
- Add coastline
- Manipulate masks with the Mask Manager Tool
- Import and export masks
- Add geometries
- Manipulate geometries
- Import and export geometries as ESRI shapefiles
- Generate a Region of Interest (ROI) mask

Layers

- Layers
  - band image
  - color bar
  - contours
  - GCP
  - geometries
  - masks
  - map gridlines
  - no-data
  - pins
  - world map

- Layer Manager Tool
  - add/remove
  - show/hide
  - stack order
  - transparency
  - edit (Layer Editor Tool)



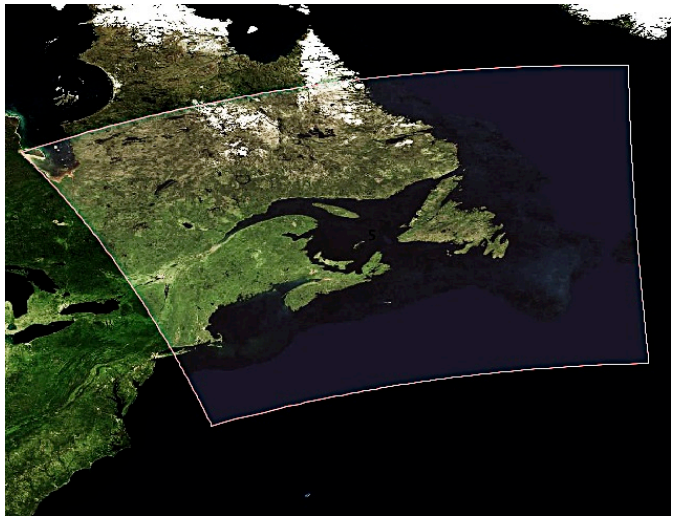
Chlorophyll on US Eastern continental shelf, layers displayed: land mask (black), no-data (grey), map gridlines, color bar, 2000m isobath contour, coastline contour

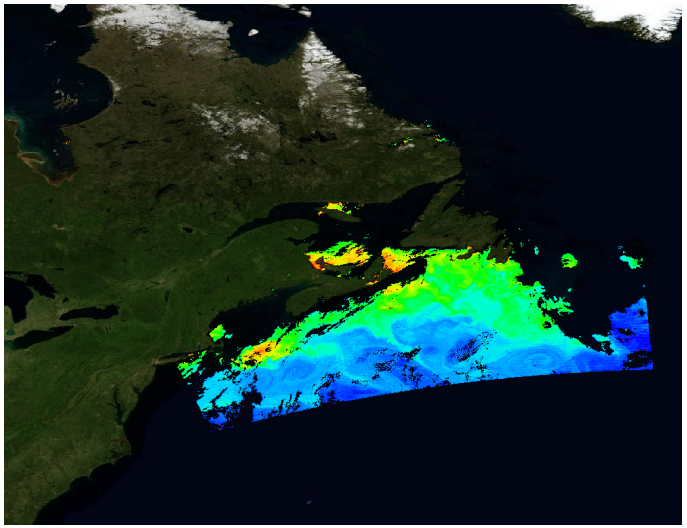
- Stored independently in each image view
- Can only be saved in a session

World Map

- World Map Layer
  - Can be displayed in the image view
  - File projection limitation (Geographic Lat/Lon (WGS 84)

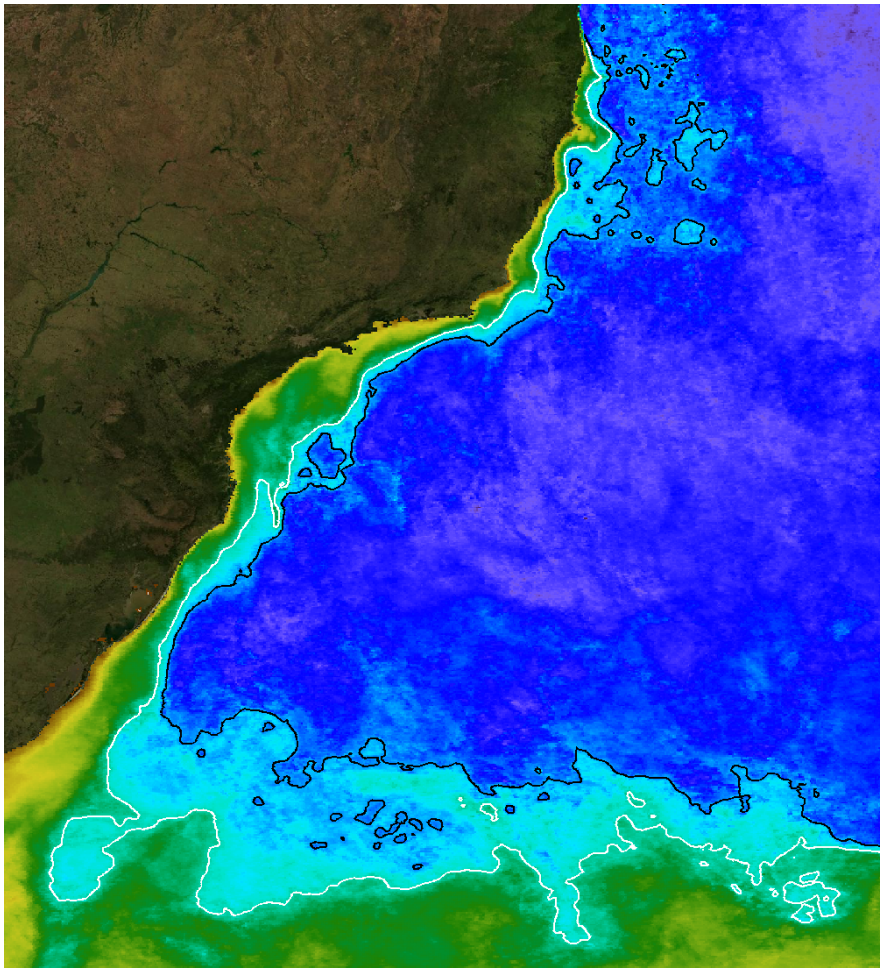
- World Map Location
  - Show outlines of all loaded file boundaries with file index number
  - No projection limitation





Contour

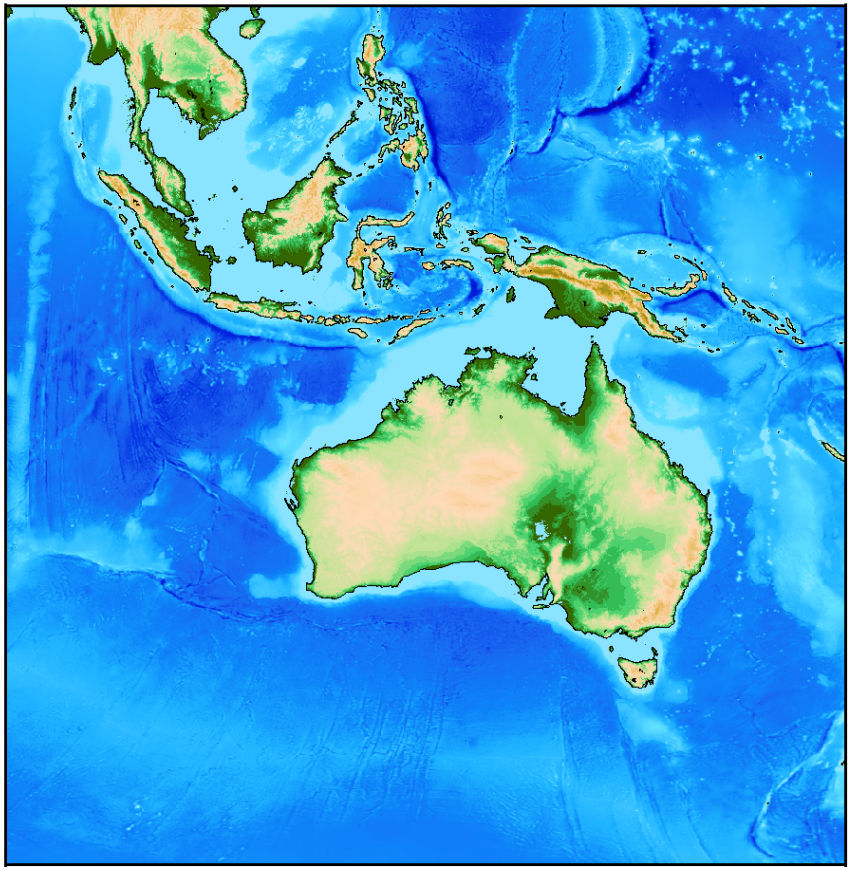
- Creates contour layer(s)
  - 1 contour level per layer
- Automated multi-level settings
  - User specifies number of levels
- Line formatting
  - Color
  - Width
  - Type (currently solid only)
- Band filter smoothing option



Euphotic Depth, contours made with 5x5 filter
100 m (black), 80 m (white)

Bathymetry

- Creates 3 bands
  - elevation
  - elevation_bathymetry
  - elevation_topography
- Bands are sea-level elevation
  - so not applicable to lakes
- Creates bathymetry mask
  - Depth range (elevation)
- Uses OCSSW file (~2km)
  - download



Exercise 3

- Overlay world map
- Show no-data layer
- Add data contour lines
- Add bathymetry band/masks
- Manipulate layers with Layer Manager

Math Band & Expression Editor

- Math Band Tool - creates a (virtual) band based on a mathematical expression of reference bands and masks
- Selectable equation reference bands
- Virtual band dependent on reference bands and masks
- Real band conversion
- Sample math expressions
  - Create a comparison difference band
    - {BAND1} – {BAND2}
  - Create a comparison ratio band
    - ({BAND2} != 0) ? {BAND1} / {BAND2} : NaN
  - Create a masked band
    - ({ROI_MASK && ! {MASK}) ? {BAND1} : NaN
- Masks and Math Band share same expression editor tool.

Exercise 4

- Generate user-defined bands
  - Difference band
  - Masked data band
  - Generate a new mask
  - Generate a user-defined algorithm band
    - MBR

      log10(max(Rrs_443,Rrs_488) / Rrs_547)
    - chl_oc3m

      pow(10,(0.2424 - (2.7423*MBR) + (1.8017*pow(MBR,2)) + (0.0015*pow(MBR,3)) - (1.228*pow(MBR,4))))
- Demonstrate product-to-product math

Reprojection & Orthorectification

- Select projection (CRS)
  - Default - Geographic Lat/Lon (WGS 84)

- Select resampling
  - Default – nearest neighbor
  - Bilinear
  - Bicubic

- Orthorectification
  - Select DEM – Digital Elevation Model

Collocation

- Copies all bands from multiple source files putting them into a single output file

- Enables
  - Band-to-band comparisons (pixel info, math difference band, …)
  - Multiband masking
  - Image alignment
  - Sharing layers and masks
  - Band layer stacking

- 2 Source files:
  - Reference
  - Dependent

- Output file:
  - contains all bands from both source files
  - reference bands are unaltered
  - dependent bands are cropped and resampled to reference file raster
  - bands renamed if identically named in source files

- Can be iterated to deal with many source files

Mosaic

- Averages desired bands from multiple source files to produce a single output file

- For output file user specifies
  - map projection
  - geographic boundaries
  - pixel size
  - orthorectification (optional)

- Optional math (masking) expression
  - Applied to each file BEFORE reprojection

- Currently no automated geographic boundary default
  - User enters coordinates or
  - Interactive map tool which displays input file boundaries

File Crop/Subset & Copy

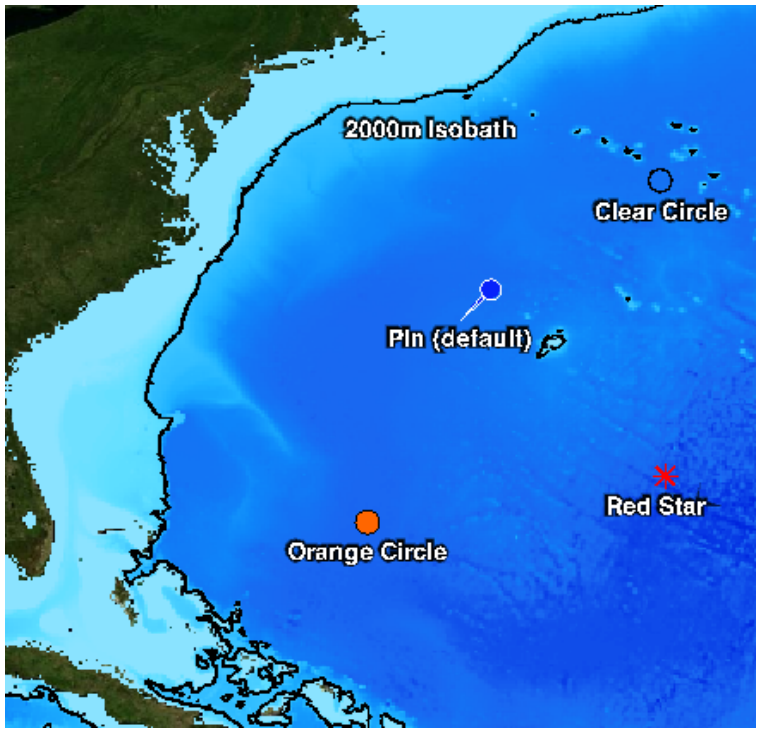
- Produces a file which is an exact copy, partial copy, or crop of the original
- Boundary reduction is in full pixel increments, no resampling needed
- User sets image boundaries
  - Pixel coordinates
  - Geo coordinates
  - Whole image
  - Default (image view window)
- User sets Subsampling
  - pixel step size
  - Default (all pixels)
- User selects which bands to retain
  - Default (all bands)
- User selects which metadata to retain
  - Default (all metadata)

Exercise 5

- Reproject a product
- Collocate multiple products from different sensors
- Generate a mosaic from multiple inputs
- Crop and export a regional/product subset file

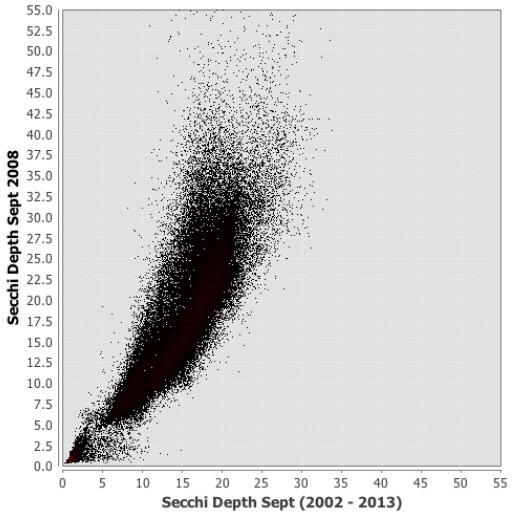
Pins

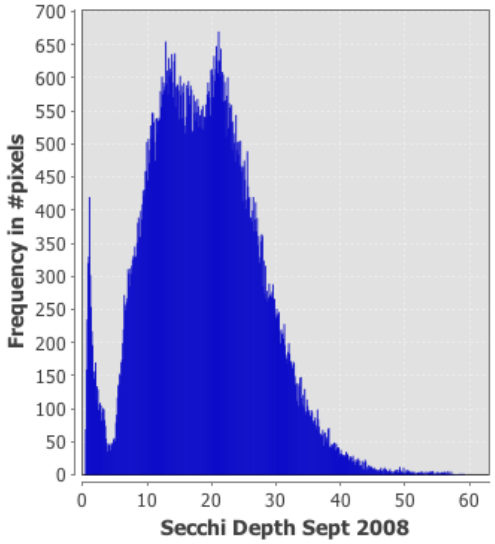
- Uses
  - Pixel Info
  - Ship Track
  - Spectrum View
  - Text Annotation (currently poor quality)
- Add/edit
  - Manual placement
  - Enter coordinates
- Formatting
  - Icon color
  - Icon transparency
  - Icon shape (pin, circle, star, cross, plus)
  - Text (not adjustable)
- Export / Import



Analysis & Statistics

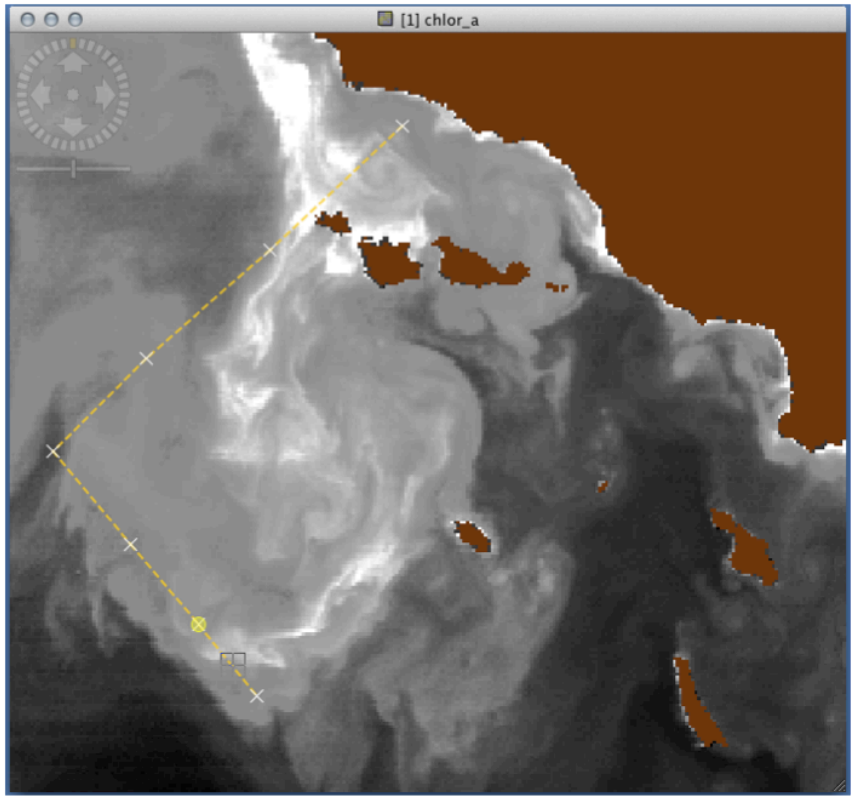
- Statistics
- Analysis Plots
  - Correlative Plot
  - Histogram Plot
  - Profile Plot
  - Scatter Plot
  - Spectrum Plot
- Uses ROI (Region-Of-Interest) Masking
- Analysis restrictions
  - All bands must be in same file (Collocate)
- File Processing Analysis Tools
  - Cluster (EM and K-Means)
  - Principal Component
  - Spectral Unmixing





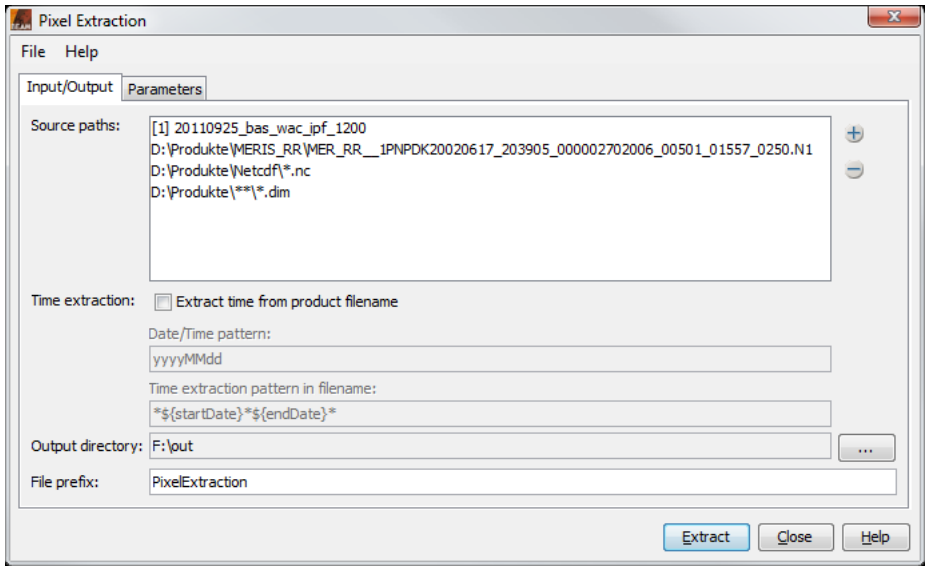
Ship Track

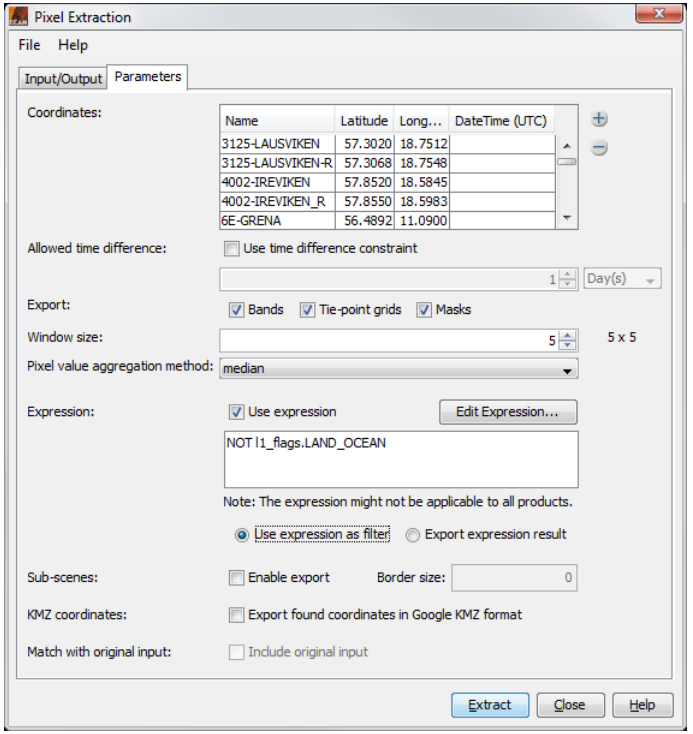
- Now called “Correlative Data”
- Input *in situ* data
  - SeaBASS format
  - CSV format
    - “Character Separated Values”
      - Tab
- Loaded data available for analyses
  - Correlative Plot
  - Histogram Plot
  - Profile Plot
  - Scatter Plot
- Pixel Extraction



Pixel Extraction

- Extract the pixel values given a user-specified list of geographic coordinates





Exercise 6

- Add and manipulate pins
- Load in situ data
- Display profile plot
- Display histogram plot
- Display scatter plot
- Display spectrum view
- Pixel extraction

Layout, Toolbars & Window Docking

- Toolbars
  - Small groupings (for better user customability)
  - Show/hide
  - Rearrangeable
  - Floating (best to avoid this)

- Tool windows
  - Can be docked/undocked
  - Dock location is fixed

- Layout
  - toolbars (visible, location),
  - tool windows (visible, dock state, location, size)

- Save/Load layout
  - Only 1 user layout
  - For multiple layouts (user layout file ~/.seadas/seadas-app/user.layout)

- Reset to Default layout

Sessions

- Save/load everything (SeaDAS sessions)

- Sessions include
  - Loaded files (location of file)
  - Image Views (contains mask and layer settings)
  - Current Layout

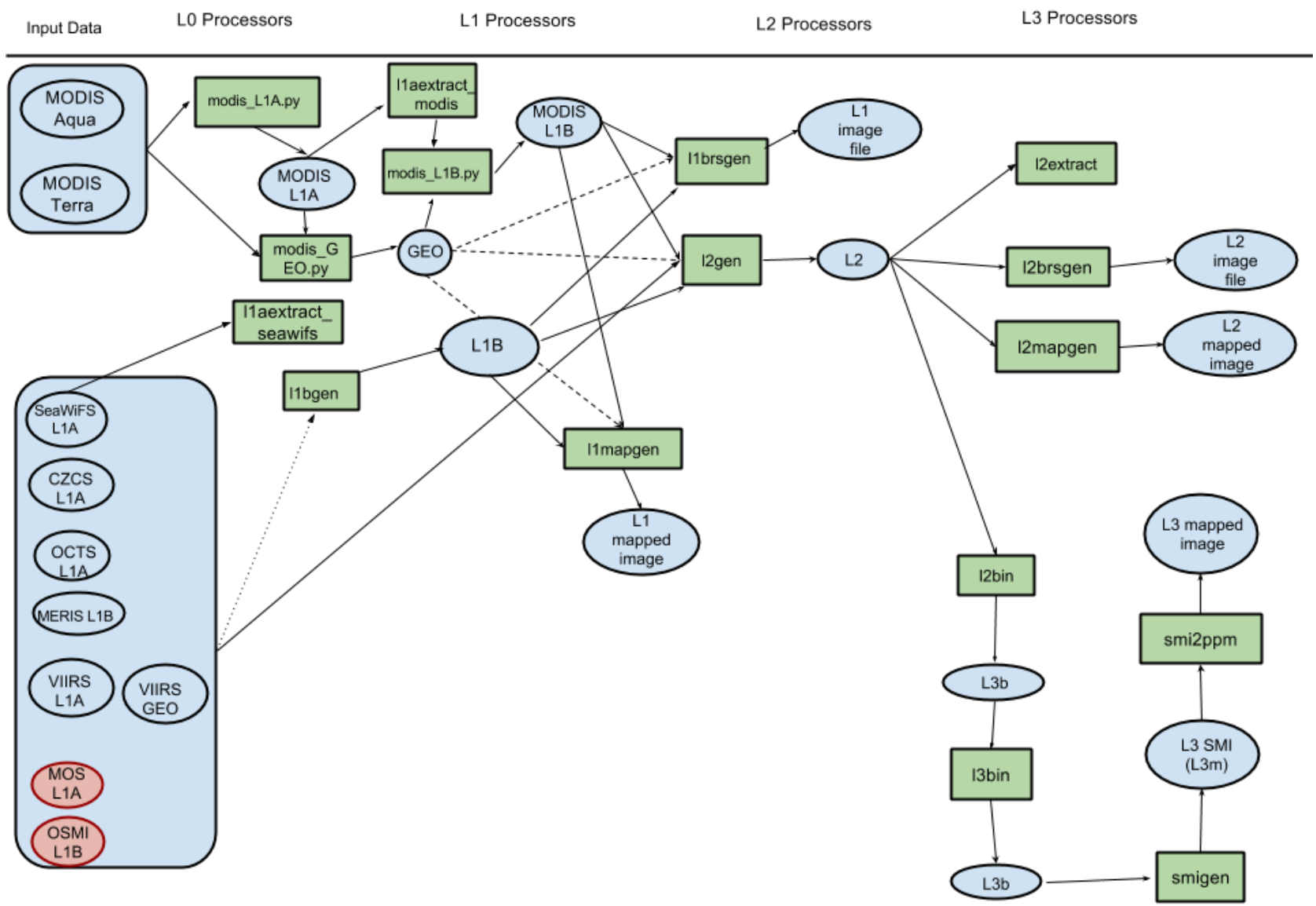
- Sessions are NOT a recording of actions but a single snapshot at a point in time

Exercise 7

- Personalize toolbars
  - Location
  - Show/hide
- Save/Restore layout
- Session Management

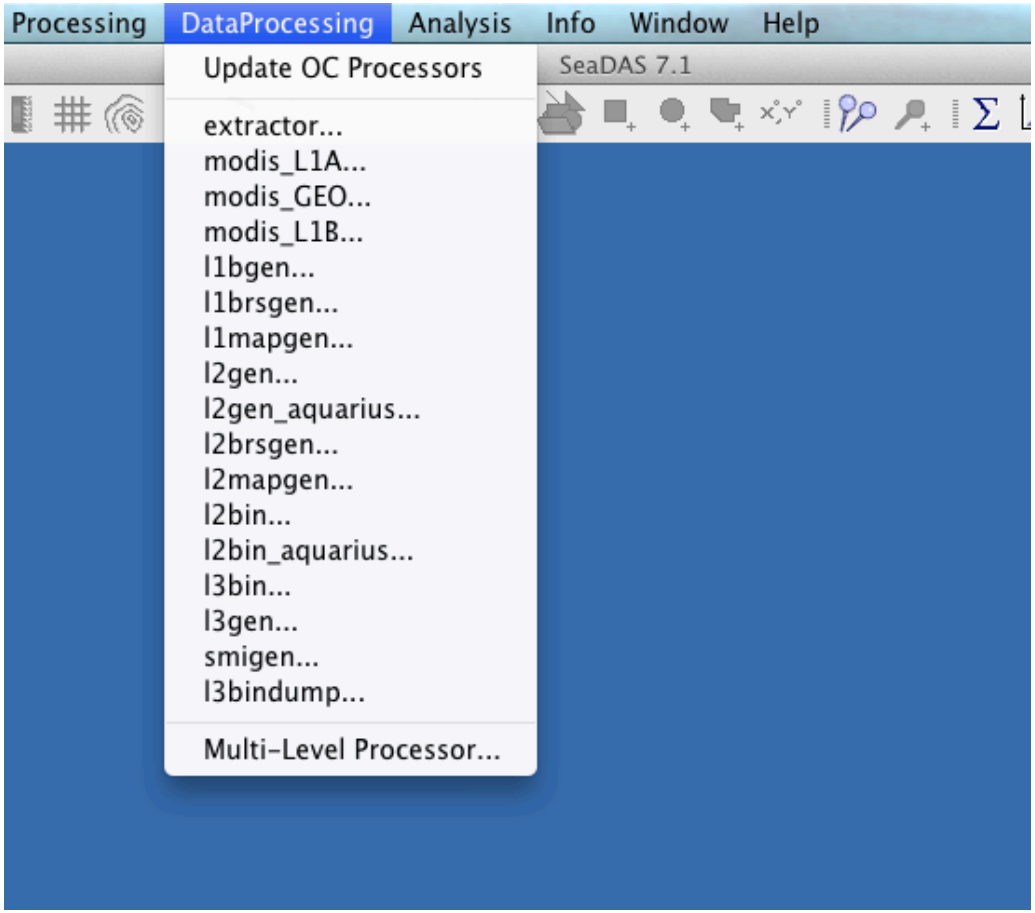
OCSSW Processors

OCSSW Data Flow Diagram

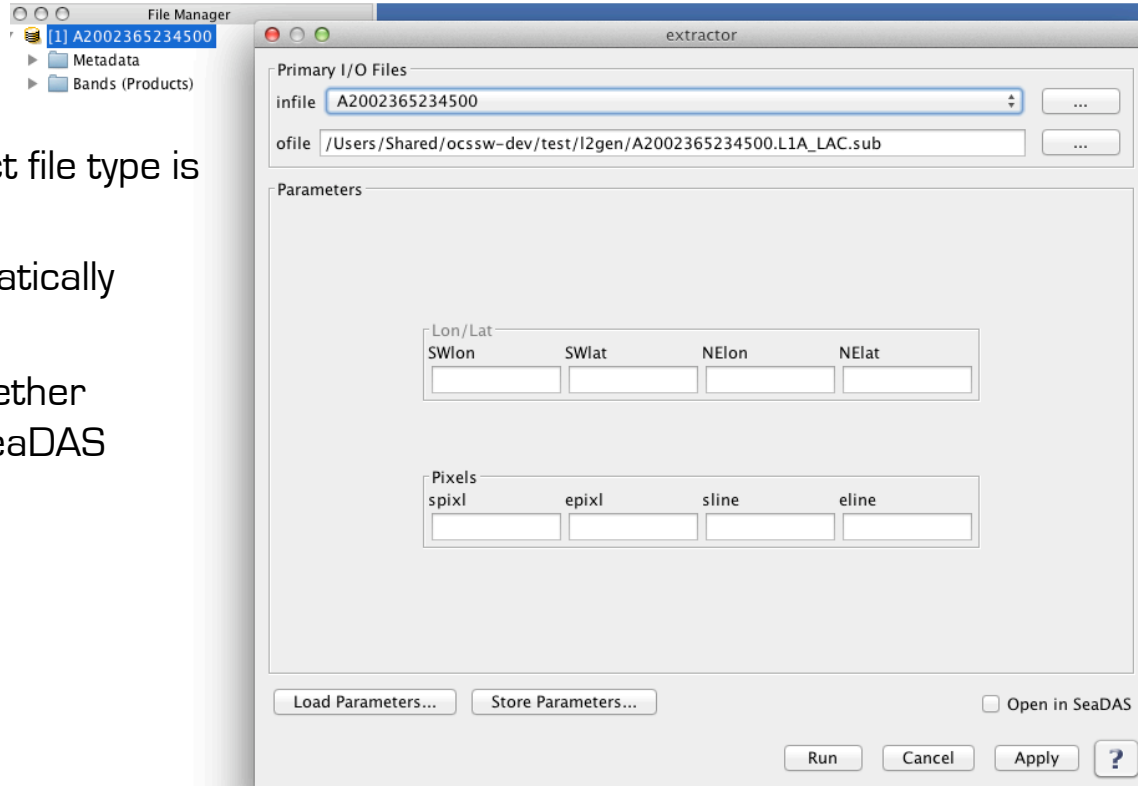


OCSSW Processors GUI

- All OCSSW processors have dedicated user interfaces in SeaDAS
- If a processor fails to execute inspect the following:
  - Input file type
  - Validity of arguments and options



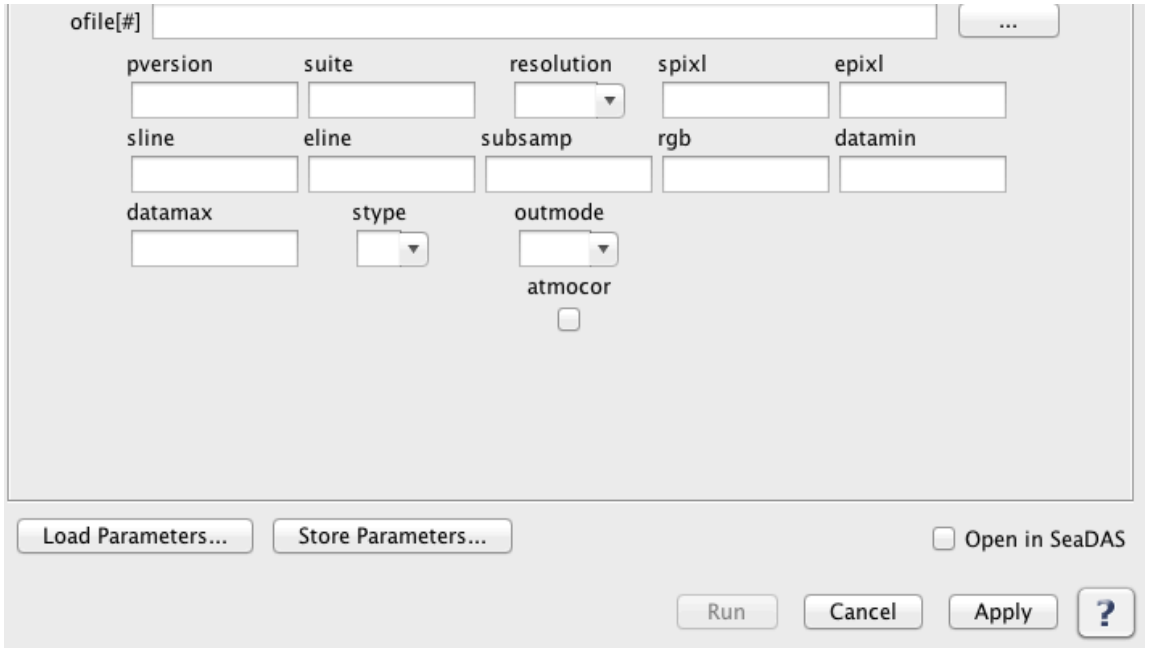
OCSSW Processors GUI



- ifile – open product with correct file type is used as default
- ofile name is computed automatically from ifile attributes
- "Open in SeaDAS" dictates whether generated ofile be opened in SeaDAS when execution is completed

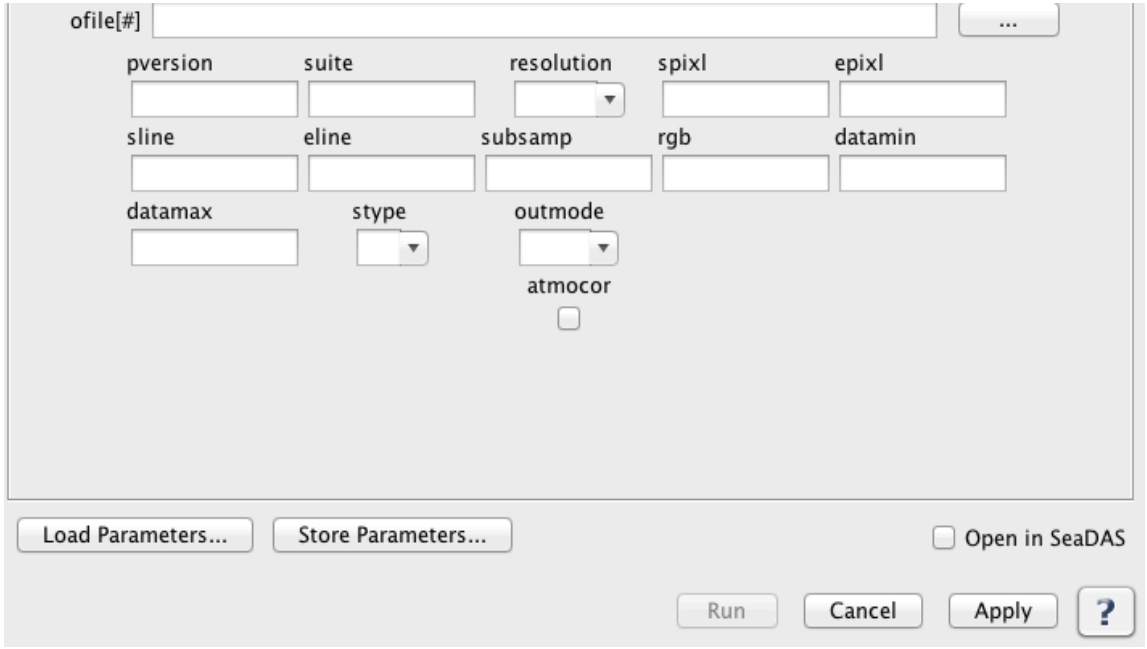
OCSSW Processors GUI

- Parameters on user interfaces are derived from corresponding command line program arguments, options, and flags
- Parameters can be loaded from files
  - Load Parameters …
- Parameters on UI can be saved into files
  - Store Parameters …



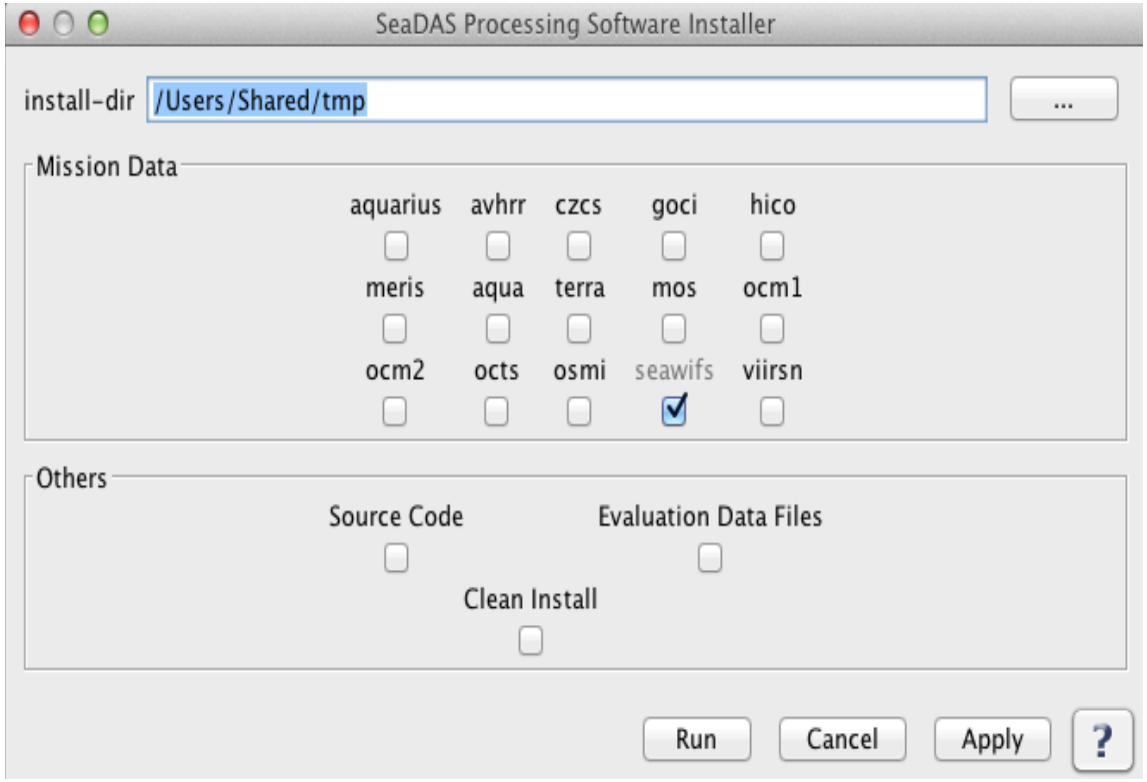
OCSSW Processors GUI

- "Run" button will be activated when "must have" parameters of a program have valid values
- "Cancel"
- "Apply"
- "?" – Displays help for a program



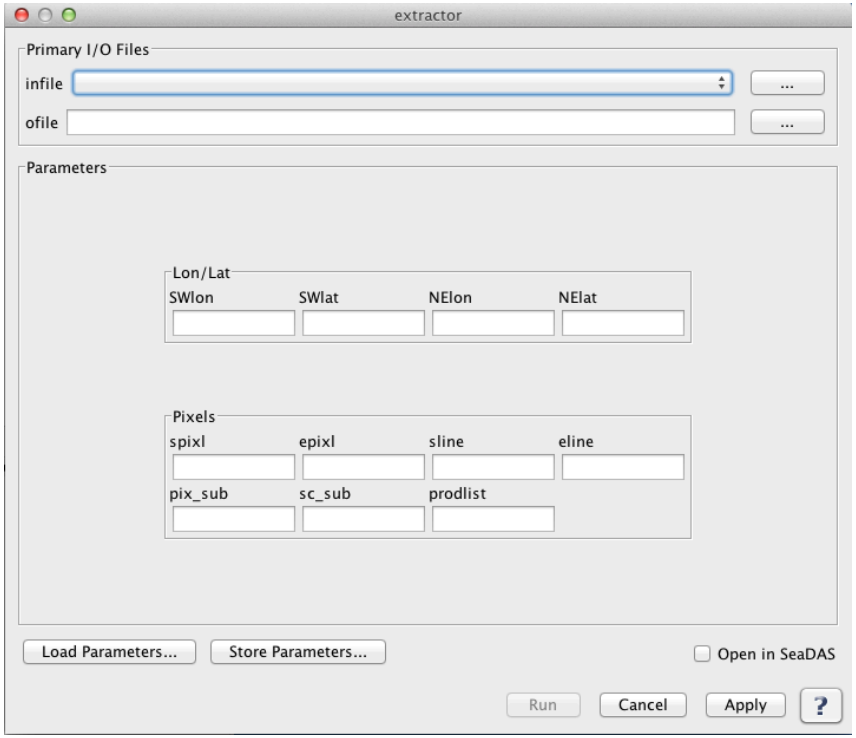
OCSSW Processors – Installer/Updater

- Default OCSSW installation directory is specified in seadas.config file
- Install if empty
- Update if already installed
- Mission selection
- Options
  - Source code
  - Evaluation data files
  - Clean install



OCSSW Processors - Extractor

- Interface for three file extractor programs
  - l1aextract_modis
  - l1aextract_seawifs
  - l2extract
- Accepts three files types
  - L1A MODIS
  - L1A SeaWiFS
  - L2
- Either “Lon/Lat” or “Pixel/Line” parameters need to be fully provided
- If “Lon/Lat” parameters are provided, pixel and line extents are computed automatically



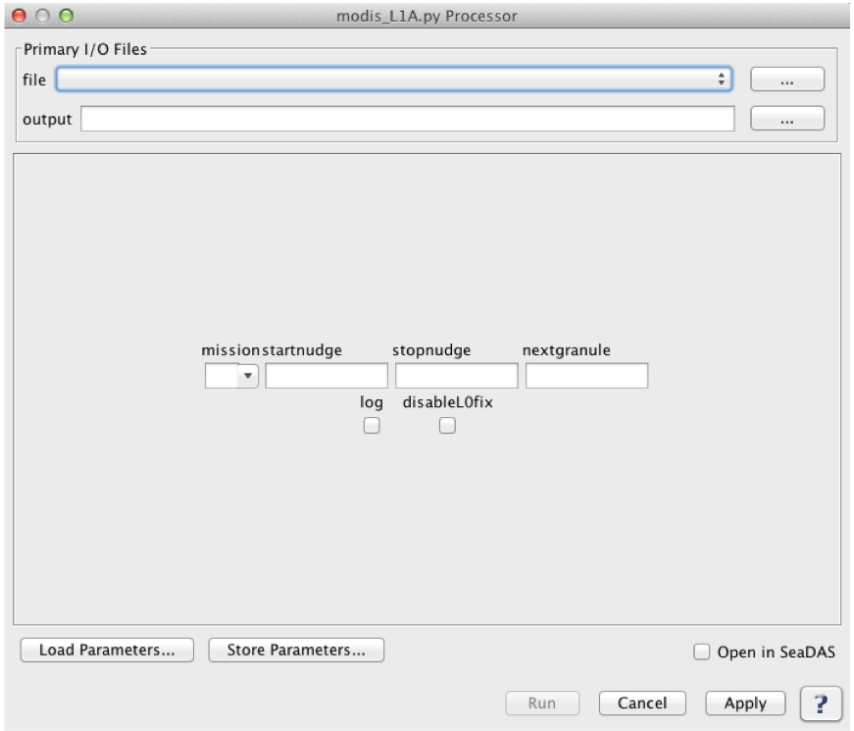
OCSSW Processors – modis_L1A.py

- Generate MODIS L1A file from L0 file

Command line usage

```
    modis_L1A.py [OPTIONS] MODIS_L0_file
            or
    modis_L1A.py --parfile=parameter_file [OPTIONS]


Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -p PARFILE, --parfile=PARFILE
                        Parameter file containing program
                        inputs
  -o L1AFILE, --output=L1AFILE
                        Output L1A filename - defaults to
                        '(A|T)YYYYDDDHHMMSS.L1A_LAC'
  -m MISSION, --mission=MISSION
                        MODIS mission - A(qua) or T(erra)
  -s STARTNUDGE, --startnudge=STARTNUDGE
                        Level-0 start-time offset (seconds)
  -e STOPNUDGE, --stopnudge=STOPNUDGE
                        Level-0 stop-time offset (seconds)
  -n NEXT, --nextgranule=NEXT
                        Next L0 granule (for geolocation of
                        last scan; sets stopnudge=0)
  -v, --verbose         print status messages
  --log                 Save processing log file(s)
  -d, --disableL0fix    Disable use of l0fix_modis utility for
                        corrupt packets
```



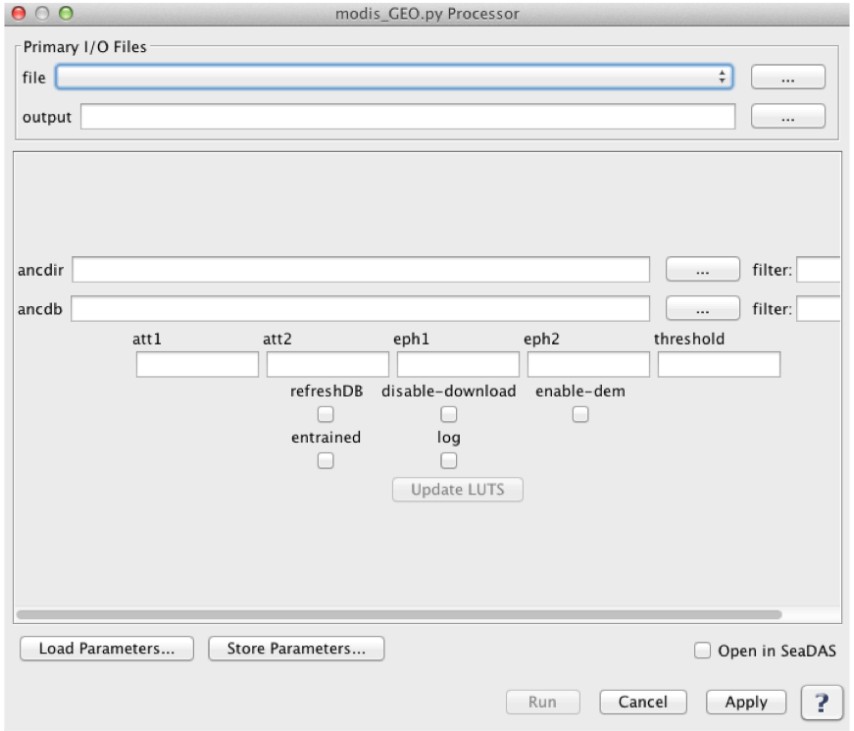
OCSSW Processors – modis_GEO.py

- Generate MODIS geolocation file from L1A file

Command line usage

```
    modis_GEO.py [OPTIONS] MODIS_L1A_file
            or
    modis_GEO.py --parfile=parameter_file [OPTIONS]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -p PARFILE, --parfile=PARFILE
                        Parameter file containing program inputs
  -o GEOFILE, --output=GEOFILE
                        Output filename
  -a ATT1, --att1=ATT1  Input attitude  file 1 (chronological)
  -A ATT2, --att2=ATT2  Input attitude  file 2 (chronological)
  -e EPH1, --eph1=EPH1  Input ephemeris file 1 (chronological)
  -E EPH2, --eph2=EPH2  Input ephemeris file 2 (chronological)
  --ancdir=ANCDIR       Use a custom directory tree for ancillary
                        files
  --ancdb=ANCDB         Use a custom file for ancillary database.
  --threshold=THRESHOLD % of geo-populated pixels required to pass
                        geocheck validation test
  -r, --refreshDB       Remove existing database records and re-
                        query for ancillary files
  --disable-download    Disable download of ancillary files not
                        found on hard disk
  -d, --enable-dem      Enable MODIS terrain elevation correction
  -v, --verbose         print status messages
  -n, --entrained       Use entrained attitude for Terra
  --log                 Save processing log file(s)
```



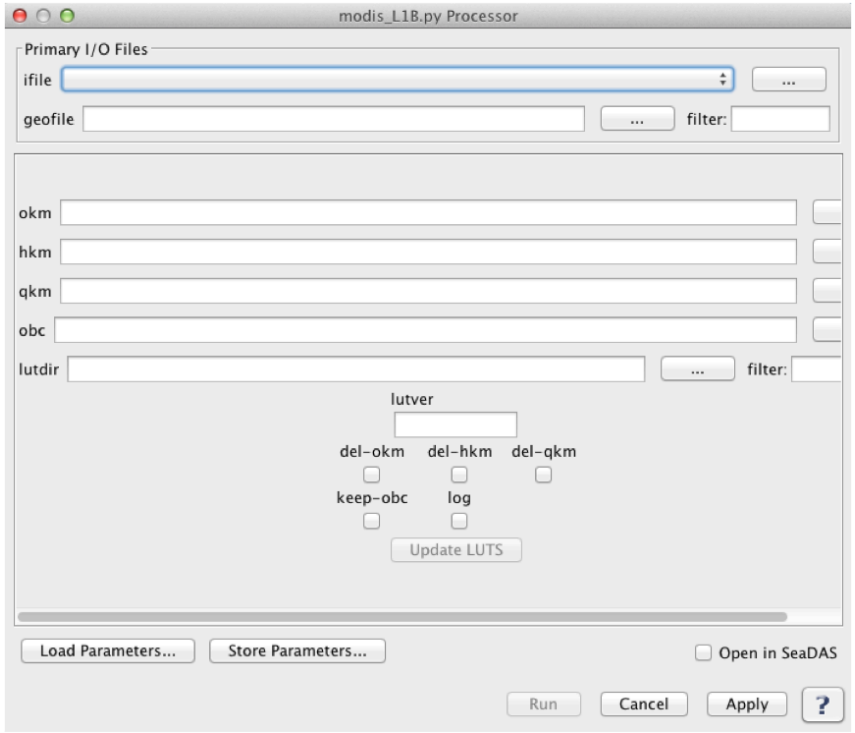
OCSSW Processors – modis_L1B.py

- Generates MODIS L1B file from L1A and geolocation files

Command line usage

```
    modis_L1B.py [OPTIONS] L1AFILE [GEOFILE]
        if GEOFILE is not provided, assumed to be basename of
        L1AFILE + '.GEO'
            or
    modis_L1B.py --parfile=parameter_file [OPTIONS]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -p PARFILE, --parfile=PARFILE
                        Parameter file containing program inputs
  -o 1KMFILE, --okm=1KMFILE
                        Output MODIS L1B 1KM HDF filename
  -k HKMFILE, --hkm=HKMFILE
                        Output MODIS L1B HKM HDF filename
  -q QKMFILE, --qkm=QKMFILE
                        Output MODIS L1B QKM HDF filename
  -c OBCFILE, --obc=OBCFILE
                        Output MODIS L1B OBC HDF filename
  -l LUTVER, --lutver=LUTVER
                        L1B LUT version number
  -d LUTDIR, --lutdir=LUTDIR Path of directory containing LUT files
  -x, --del-okm         Delete 1km  resolution L1B file
  -y, --del-hkm         Delete 500m resolution L1B file
  -z, --del-qkm         Delete 250m resolution L1B file
  --keep-obc            Save onboard calibration file
  -v, --verbose         print status messages
  --log                 Save processing log file(s)
```



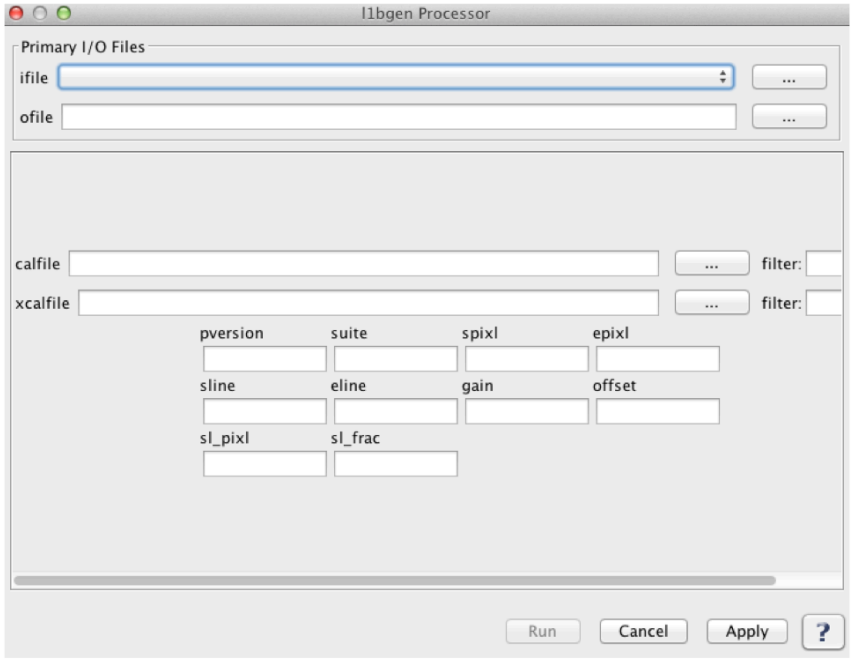
OCSSW Processors – l1bgen

- Generate a generic L1B file

Command line usage

```
l1bgen argument-list

 par (ifile) = input parameter file
 ifile (ifile) (alias=ifile1) = input L1 file name
 ofile (ofile) (alias=ofile1) (default=output) = output file
 oformat (string) (default=netCDF4) = output file format
      netcdf4: output a netCDF version 4 file
      hdf4:    output a HDF version 4 file
 spixl (int) (default=1) = start pixel number
 epixl (int) (default=-1) = end pixel number
 sline (int) (default=1) = start line number
 eline (int) (default=-1) = end line number
 calfile (ifile) = system calibration file
 xcalfile (ifile) = cross-calibration file
 gain (float) = calibration gain multiplier
 offset (float) = calibration offset adjustment
```



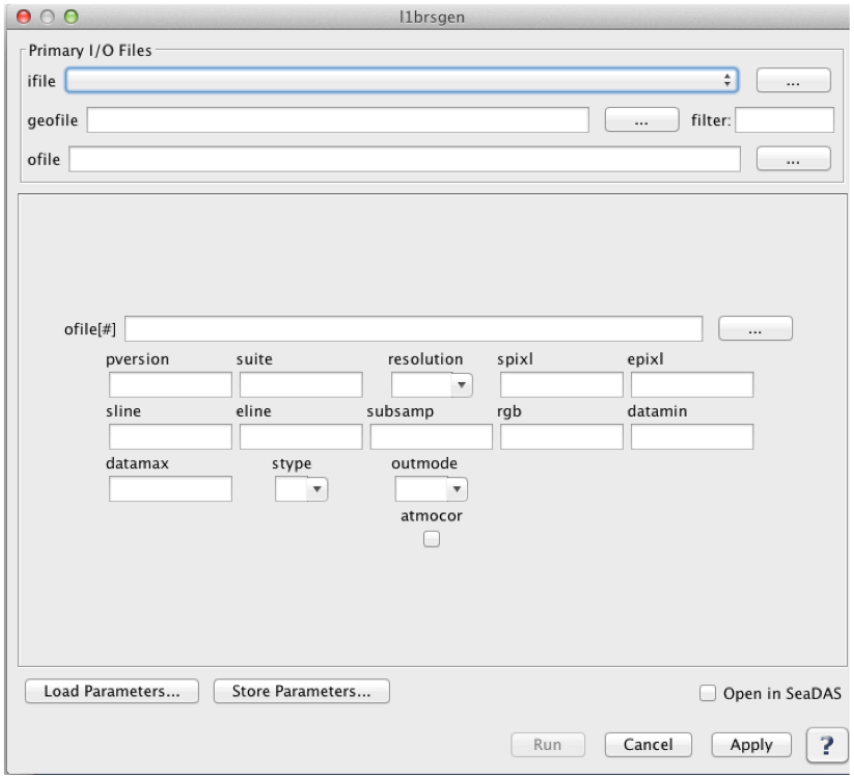
OCSSW Processors – l1brsgen

- Generate a pseudo-true color browse file

Command line usage

```
l1brsgen argument-list

par (ifile) = input parameter file
ifile (ifile) (alias=ifile1) = input L1 file name
geofile (ifile) = input L1 geolocation file name (MODIS only)
resolution (int) (default=-1) = processing resolution (MODIS)
ofile (ofile) (alias=ofile1) (default=output) = output file name
oformat (string) (default=hdf4) = output file format
        hdf4: output a HDF4 file
        bin:  output a flat binary file
        png:  output a PNG file
        ppm:  output a netPBM PPM file
oformat_depth (string) (default=8bit) = output file color depth
                                        for HDF4 file
spixl (int) (default=1) = start pixel number
epixl (int) (default=-1) = end pixel number (-1=the last pixel)
sline (int) (default=1) = start line number
eline (int) (default=-1) = end line number (-1=the last line)
subsamp (int) (default=1) = sub-sampling interval
rgb (int) (default=[1,1,1]) = bands to use for red, green and
                              blue
atmocor (boolean) (default=on) = toggle atmospheric correction
datamin (float) (default=0.01) = minimum reflectance for scaling
datamax (float) (default=0.9) = maximum reflectance for scaling
stype (int) (default=0) = scaling type
```

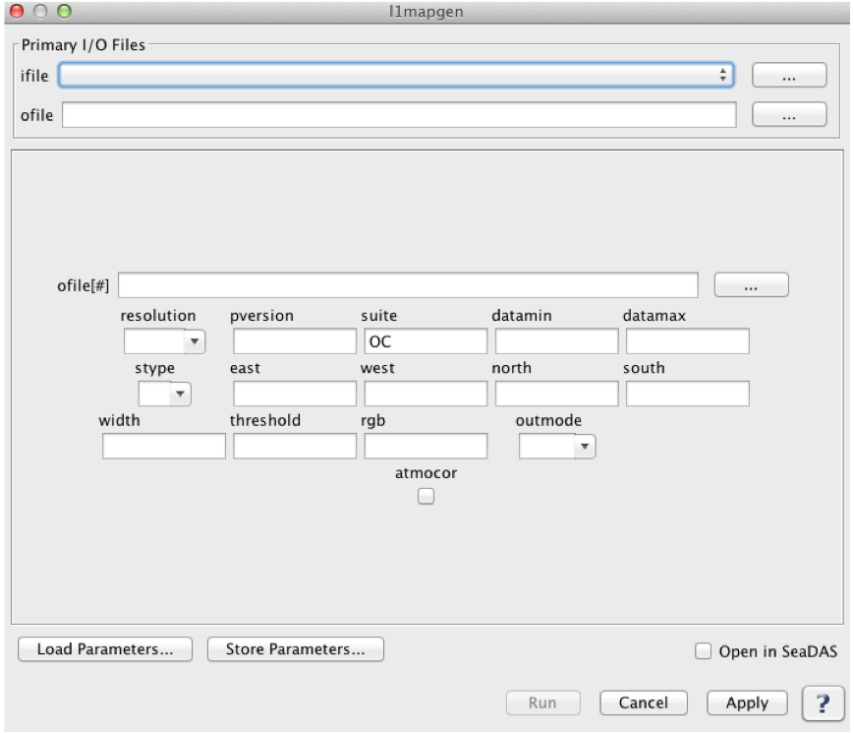


OCSSW Processors – l1mapgen

- Generate a pseudo-true color mapped file (Plate Carre projection)

Command line usage

```
l1mapgen argument-list
par (ifile) = input parameter file
ifile (ifile) (alias=ifile1) = input L1 file name
geofile (ifile) = input L1 geolocation file name (MODIS/VIIRS)
resolution (int) (default=-1) = processing resolution (MODIS)
ofile (ofile) (alias=ofile1) (default=output) = output file name
oformat (string) (default=ppm) = output file format
        ppm:  output a netPBM PPM file
        png:  output a PNG file
        tiff: output a geoTIFF file
north (float) (default=-999) = north boundary
south (float) (default=-999) = south boundary
east (float) (default=-999) = east boundary
west (float) (default=-999) = west boundary
width (int) (default=600) = width of output image
threshold (float) (default=0.1) = threshold for the number of
          good pixels before an image is produced
rgb (int) (default=[1,1,1]) = bands to use for red, green and
                              blue
atmocor (boolean) (default=on) = toggle atmospheric correction
datamin (float) (default=0.01) = minimum reflectance for scaling
datamax (float) (default=0.9) = maximum reflectance for scaling
stype (int) (default=0) = scaling type
```

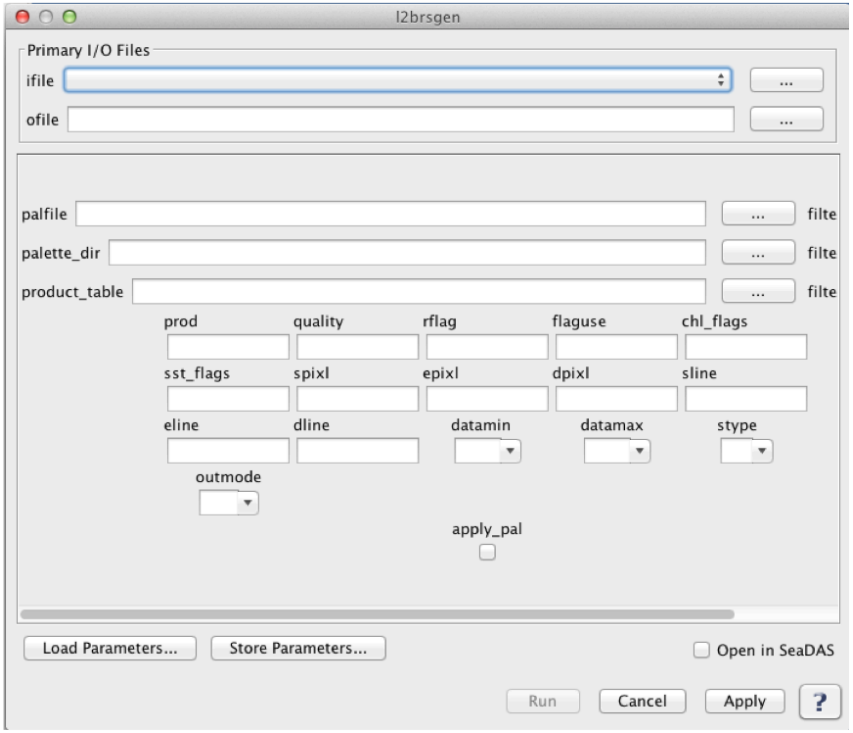


OCSSW Processors – l2brsgen

- Generate a product browse file

Command line usage

```
l1brsgen argument-list
par = input parameter file
ifile = input L2 file name
ofile (ofile) (default=output) = output filename
prod = product name
quality = highest quality value acceptable
flaguse = Flags used to mask data
chl_flags = Flags used to mask data for chl product if flaguse not set
sst_flags = Flags used to mask data for sst product if flaguse not set
spixl = start pixel number
epixl = end pixel number (-1=the last pixel)
dpixl = pixel subsampling interval
sline = start line number
eline = end line number (-1=the last line)
dline = line subsampling interval
apply_pal = apply color palette, false = grayscale
palfile = palette filename.
palette_dir = directory containing the palette files
datamin = minimum value for data scaling
datamax = maximum value for data scaling
stype = scaling type
        1: LINEAR
        2: LOG
oformat = format of the output file
        hdf4: (1) HDF browse file
        png:  (5) PNG color or grayscale image file
        ppm:  (7) PPM color or PGM grayscale image file
```

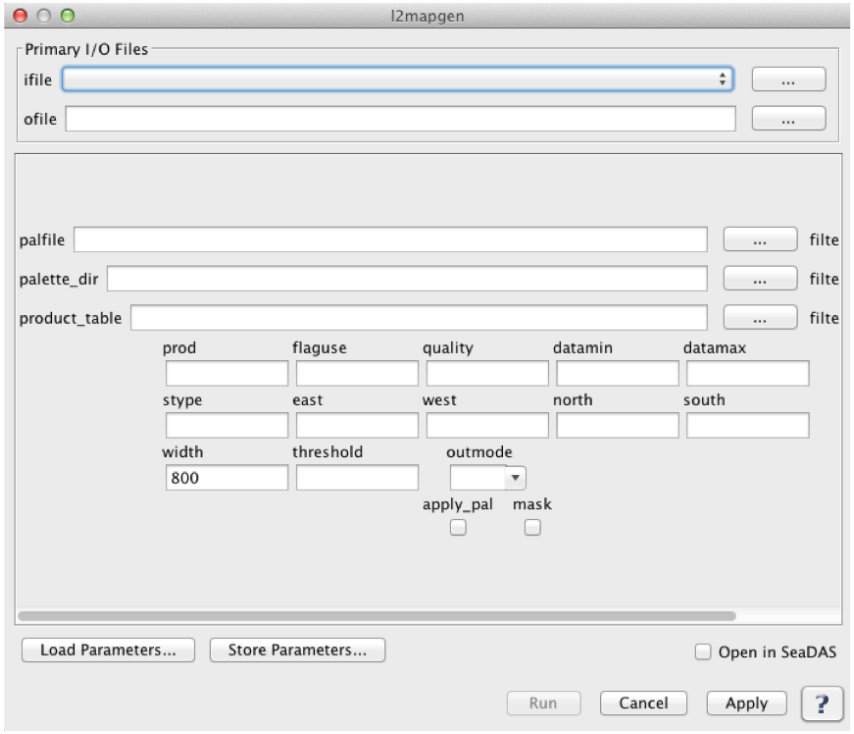


OCSSW Processors – l2mapgen

- Generate a L2 product mapped file (Plate Carre projection)

Command line usage

```
l2mapgen argument-list
par (string) = input parameter file
ifile = input L2 file name or file with a list of files names
ofile = output map filename (NULL=STDOUT)
prod = product name
apply_pal = apply color palette, false = grayscale
palfile = palette filename
palette_dir = palette directory
flaguse (string) = flags to be masked
mask = mask land, cloud and glint
datamin = minimum value for data scaling
datamax = maximum value for data scaling
stype = scaling type
east = Map East longitude
west = Map West longitude
north = Map North latitude
south = Map South latitude
width = width of the output image
threshold = minimum percentage of the area of interest
        that must receive valid pixel data before an image is
       generated
outmode = format of the output file
        ppm: PPM or PGM image file
         png: PNG color or grayscale image file
        tiff: TIFF color or grayscale geo tiff image file
```



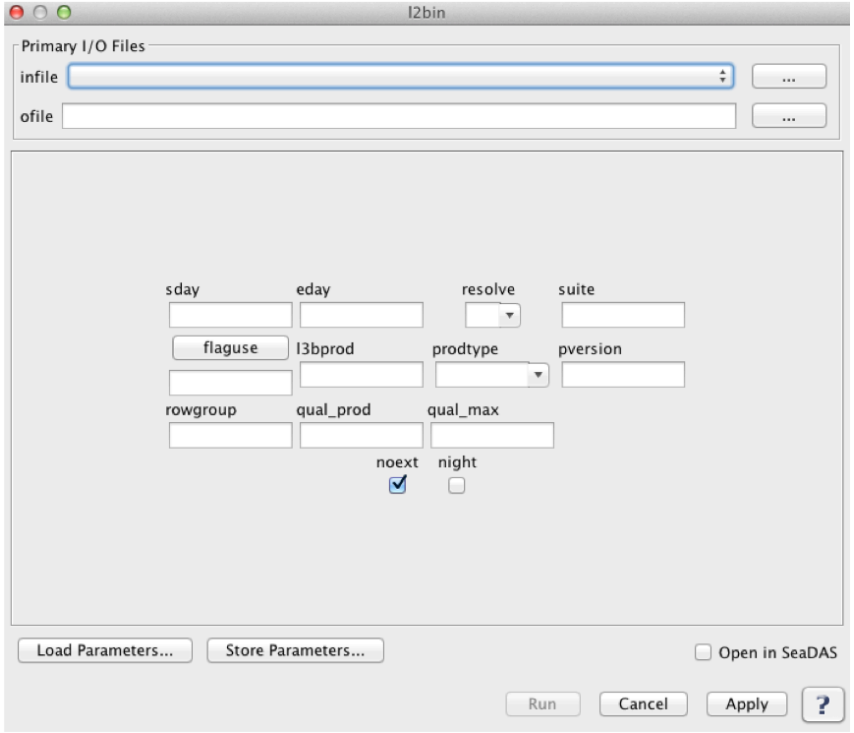
OCSSW Processors – l2bin

- Generate a L3 bin file from one or more L2 files

Command line usage

```
l2bin parfile=parfile or
            infile=infile ofile=ofile [sday=sday] [eday=eday]
            resolve=resolve [flaguse=flaguse] [l3bprod=l3bprod]
            [prodtype=prodtype] [noext=noext] [verbose=verbose
            [rowgroup=rowgroup] [night=night]

parfile   = parameter filename
infile    = input filename/filelist
ofile     = output bin filename
sday      = start datadate (YYYYDDD) [ignored for "regional"]
eday      = end datadate   (YYYYDDD) [ignored for "regional"]
resolve   = bin
flaguse   = flags masked [see /SENSOR/l2bin_defaults.par]
l3bprod   = bin products [default=all products]
prodtype  = product type (Set to "regional" to bin all scans.)
pversion  = production version [default=Unspecified]
noext     = set to 1 to suppress generation of external files
               [default=0, (1 for "regional" prodtype)]
rowgroup  = # of bin rows to process at once.
night     = set to 1 for SST night processing [default=0]
qual_prod = quality product field name
qual_max  = maximum acceptable quality [default=2]
verbose   = Allow more verbose screen messages [default=0]
```

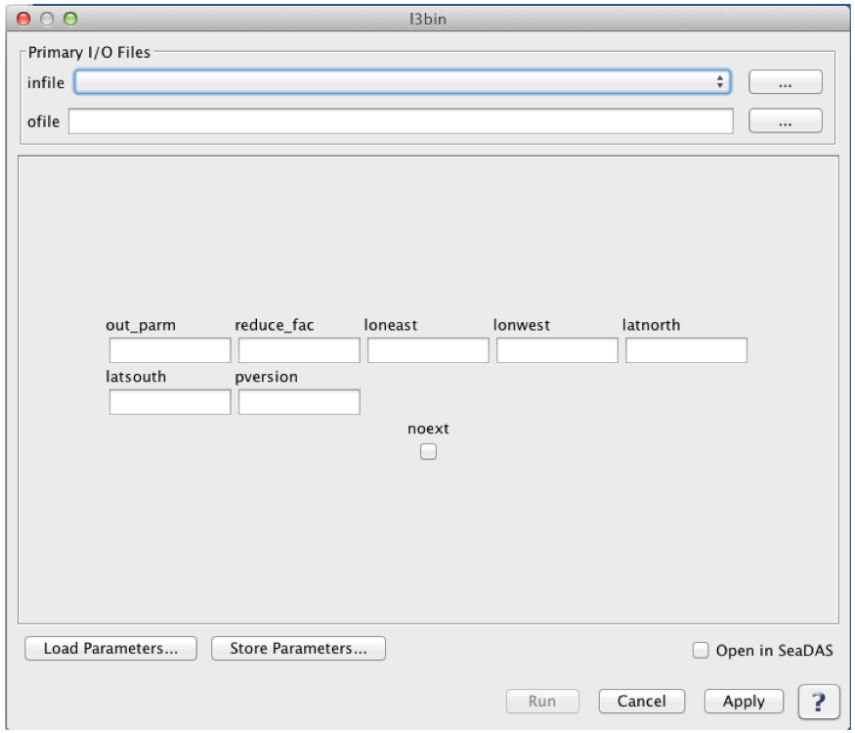


OCSSW Processors – l3bin

- Combine L3 bin files
  - used for temporal binning

Command line usage

```
l3bin in=input-file out=output-file out_parm=prodlist
            [reduce_fac=reduce_fac] [noext=noext]
input-file  = listfile of input binfiles
output-file = output bin filename
out_parm    = data products list
parfile     = parameter filename
reduce_fac  = scale reduction factor (power of 2)
loneast     = Easternmost longitude (default=+180)
lonwest     = Westernmost longitude (default=-180)
latnorth    = Northernmost latitude (default=+90)
latsouth    = Southernmost latitude (default=-90)
noext       = set to 1 to suppress generation of
external files
verbose     = Allow more verbose screen messages
pversion    = Processing Version [default=Unspecified]
oformat     = output format
              default is the same format as input files
deflate     = apply internal compression for netCDF
```

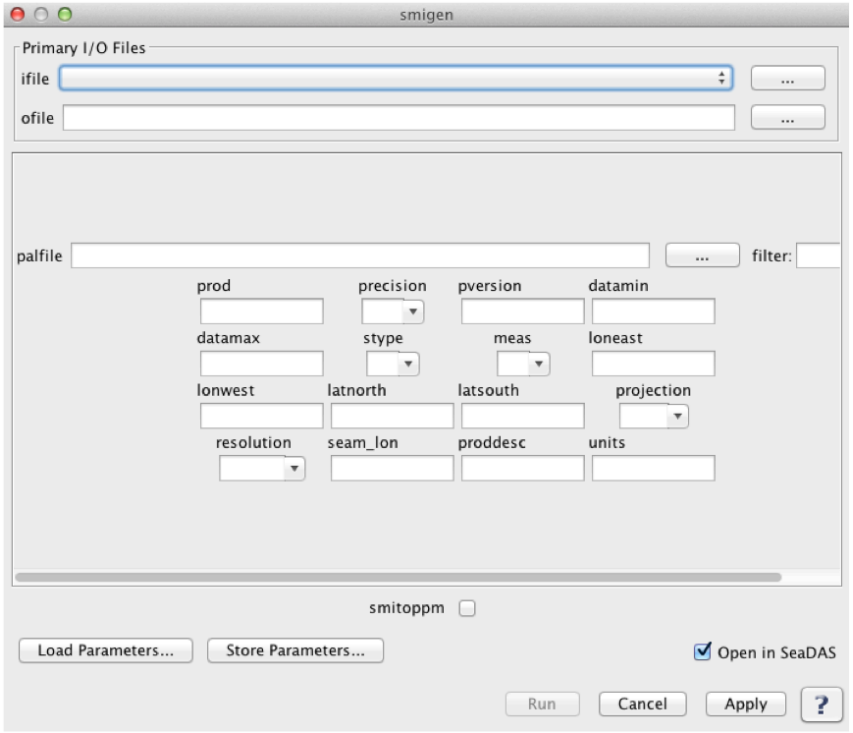


OCSSW Processors – smigen

- Generate a Standard Mapped Image (SMI) from a L3 bin file

Command line usage

```
par        = parameter filename
ifile      = input bin filename
ofile      = output map filename
oformat    = output format: 1 (HDF4), 2 (netCDF4), 3 (HDF5)
deflate    = apply internal compression for netCDF output
prod       = product name
precision  = output map precision: 'B' (default), 'I', 'F'
palfile    = palette filename
datamin    = minumum value for data scaling
datamax    = maximum value for data scaling
stype      = scaling type,1=LINEAR,2=LOG
meas       = measurement to map, 1=mean, 2=var, 3=stdev,
               4=pixels, 5=scenes (default=1)
loneast    = Easternmost longitude (default=+180)
lonwest    = Westernmost longitude (default=-180)
latnorth   = Northernmost latitude (default=+90)
latsouth   = Southernmost latitude (default=-90)
projection = SIN | RECT (default=RECT)
resolution = 36km | 18km | 9km | 4km | 2km | 1km | hkm | qkm
               1deg (one deg) | hdeg (0.5 deg) | qdeg (0.25
               deg) 10deg (0.1 deg) |udeg-#.# | ukm-#.#
              (default=9km)
seam_lon   = Longitude of Left Edge of Map (default=-180)
```



OCSSW Processors – l3bindump

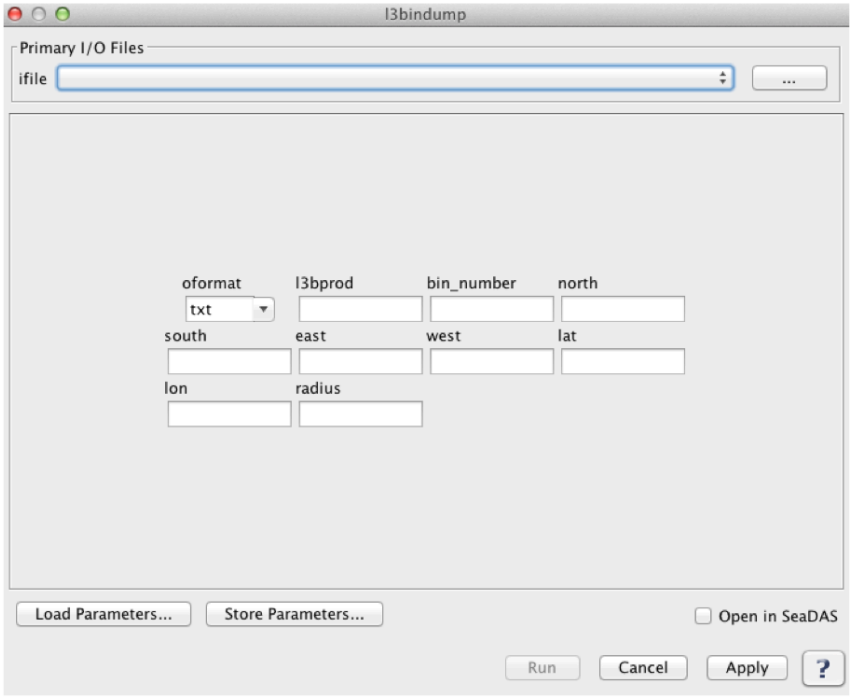
- Accepts L3 files
- Output data in a spreadsheet format

Command line usage

```
l3bindump argument-list

  There are 3 use cases:
       1) bin_number
       2) region defined by lat, lon, and radius (in km)
       3) region defined by north, south, west, east

par = input parameter file
ifile = input L1 file name
oformat = output file format
             txt:  plain text columnar format
             seabass:  SeaBASS format
l3bprod = binned product to extract
bin_number = bin number
north = north boundary
south = south boundary
east = east boundary
west = west boundary
lat = latitude
lon = longitude
radius = radius
verbose = verbose output
```



L2Gen Processor

- Files
  - Input file
    - selectable or uses loaded file
  - Output file
    - auto-named or user-named
  - Geofile
    - auto-named or user-named
  - Existence checks

- Par file
  - "Clean" par file
    - only params set to non-default values
  - Import/export

- Ancillary files
  - auto download

- Products categorized and selectable
- Params categorized with selectable value
- Loads with l2gen defaults
- Runs within SeaDAS

Exercise 8

- Process L1A file through to L2 mapped image using GUI interface

Multilevel Processor (MLP)

- Allows for sequential end-to-end processing via a single script
- Accessible via command line and the SeaDAS GUI
- Programs that can be accessed:
  - modis_L1A.py
  - modis_GEO.py
  - modis_L1B.py
  - l1aextract_modis.py
  - l1aextract_seawifs.py
  - l1mapgen
  - l1brsgen
  - l2gen
  - l2brsgen
  - l2mapgen
  - l2extract
  - l2bin
  - l3bin
  - smigen

MLP Parameter Files

- Defines
  - Lowest level input data files
  - Programs to be run
    - any options needed for those programs
- Format
  - Divided into sections with headers
    - Headers contain section names surrounded by square brackets ("[" and "]").
    - A [main] section is required
    - Other section names are the program to be described
    - There must be a section for the last program to be run
    - Sections for intermediate programs can be omitted if the default options are acceptable.
    - Entries must be formatted as "key=value" (no quotes)

MLP Processing Options

- Processing options
  - **`--ifile`** – sets input file
    - When specified in the parameter file's main section, several can be specified separated by commas. ***e.g.:***

      `--ifile``=file1.L1B, file2.L1B, file3.L1B`
    - A text file can be used to list several input files, with each file name on a separate line.
    - If **`--ifile`** is specified on the command line, only one file can be named
      - For multiple inputs, use the text list file method
  - **`--odir`** or **`--output_dir`** – sets output directory
    - default is to use current working directory
  - **`--keepfiles`** – keep all files created during processing
    - default is to keep only the final output files
  - **`--overwrite`** – allow existing data files to be overwritten, use
    - default is to stop processing when a file would be overwritten
  - **`--use_existing`** – use existing intermediate data files
    - default is to create new files
  - **`--timing`** print out timing information
  - **`--verbose`** – get extra information about the processing

Multilevel Processor GUI

- Accesses all SeaDAS processing GUIs

- File
  - Input file
    - selectable or uses loaded file
  - Intermediate Files
    - auto-named or user-named
    - keep files option
  - Output file
    - auto-named or user-named

- Par file
  - Import / export
  - Runnable with the python multi-level processor

- Runs within SeaDAS

Exercise 9

- Process a series of L1A files through to L2 mapped image using multilevel_processor.py

Help

- SeaDAS Internal Help
  - Located in Help menu
- Ocean Color Web Page
  - http://oceancolor.gsfc.nasa.gov
  - SeaDAS
    - http://seadas.gsfc.nasa.gov
    - Web Version of SeaDAS Internal Help
    - Tutorials
    - Latest downloads
  - Forum
- YouTube Channel
  - http://www.youtube.com/user/nasaoceancolor