Lab-4: SeaDAS/IDL http://seadas.gsfc.nasa.gov/seadas/doc/toplevel/sds_command.html

Mar.-12, 2010

We are approaching the point where we can start to think about scientific applications and analyses that use Modis data from any of its 36 channels. We have already done the following:

- Select and download level *.L0_LAC file for a specific study area;
- Process this file to create level L1B files *.L1B_LAC, *.L1B_HKM, and *.L1B_QKM;
- Process data of individual MODIS frequency bands using fortran code getbin.f

And you will next perform the following tasks:

- Extract flat binary data of a specific channel (bin1) and location (nav) from the L1B files;
- Extract calibration and sensor information from L1B files to convert scaled integer (SI) to radiance or reflectance values in engineering units (see Xiong et al. (2005)
- Read flat binary files into a fortran program (getbin.f) for further processing to produce data sets for MATLAB or other processing and graphing routines (gridding, map projection, etc.)

10.	Create a file of IDL commands automatically within your processing script (proc2.csh) to extract band-1 (645 nm, "red) and band-2 (865 nm, "near-infrared") from the 250-m resolution HDF *.L1B_QKM file: echo "load, '"$name.L1B_QKM"',gfile='"$name.GEO"', prod_ix=[1,2]" >test.cmd echo "out, 'bin1', /data, band=1, ftype='FLAT',/geo" >>test.cmd echo "out, 'nav', /nav, band=1, ftype='FLAT'" >>test.cmd
11.	Run the command script test.cmd to extract flat binary files bin1 and nav; the following extracts these data from embedded IDL routines without requirements of a license for this widely used but expensive commercial software product (details on SeaDAS web-site): seadas –em –b test.cmd
12.	At this point we probably filled up the 80 GB server hard disk. Hence we all need to clean-up our scripts and use an attached 2,000 GB backup device. Access it at /Volumes/acomdat and create yourself a directory structure there to hold data files for subsequent processing: cd /Volumes/acomdat mkdir your_name

13.	Rewrite your proc2.csh script to organize your data flow so that no large files remain on the main server. execute processing steps only if the required output files do not exist. Make extensive use of the backup directory /Volumes/acomdat/your_name. The following csh-command structure file will prove most useful for checking for the existence of a file:
	set dir = some_directory_path_here_could_be_$argv[2]_if_set_in_step-1.csh
	set name = some_name_here_probably$argv[1]
	set ext = some_filename_extension_here
	if (! -e $dir/$name.$ext) then
	echo 'Do something here, like creating or copying a file'
	endif
	The 'Do something" part is only executed if the file $dir/$name.$ext does not (!) exist (-e). The "Do something" is the place where you "Do something."

14	Extract relevant information on calibration, file size, scan numbers, etc, from a header dump, e.g. hdfdump –h $name.LAC_QKM >HeaderInfo.ascii You will need to recall elements from our Lab2 session, such as constructs like nawk '$0 ~ "text_string" {print $0, substr($1,12,1)}' HeaderInfo.ascii >whatever_getbin.f_needs
15.	Add a segment to the proc2.csh script to compile and execute the fortran code getbin.f