README file for sample CrIS PCA/RED data

Joe Taylor, Dave Tobin, Greg Quinn
CrIS L1B Science and Software Team
Contact: cris.l1b.support@ssec.wisc.edu
09 May 2024

This is a README file to explain the contents of sample NASA NOAA 20 CrIS PCA/RED data files and to show example use and reconstruction of radiance spectra. This README will form the basis for more official documentation (ATBD, User Guide) to come later.

This sample data is expected to be very similar to the first version of the official data. A few possible exceptions are: (1) addition of other meta-data variables depending on user feedback, (2) More local PCs (perhaps 10 instead of 5) depending on further analysis of rare events, (3) refinement of RED spectral regions and scaling.

Data may be obtained locally (at SSEC) from:
    /ships22/crisl1b/CrIS-J1/PCA_RED/cris_l1b_fsr/atmos_sips/0.2.0
Or from GES-DISC at:
    https://disc.gsfc.nasa.gov/datacollection/SNDRJ1CrISL1BPCARED_3.0.html

The example data file used for this README is:
SNDR.J1.CRIS.20240120T1812.m06.g183.PCA_RED.beta.v03_00.W.240322145711.nc

This is a NetCDF file which can be read with a range of software packages. Matlab is used here. The attached Matlab function rd_pca_red_file.m reads the data from the PCA_RED file, and also performs the reconstruction of radiance spectra depending on input flags. This additionally uses pca_expandM.m which is also included. We have also developed Python code to perform the same operation, which is available at the following URL:
    https://gitlab.ssec.wisc.edu/cris_l1b/user/-/tree/master/PCA_RED/py/cris_pca_red

Contents of the PCA_RED files:

Using rd_pca_red_file.m to read the contents:

```
>> pca_file = 'SNDR.J1.CRIS.20240120T1812.m06.g183.PCA_RED.beta.v03_00.W.240322145711.nc';
>> d = rd_pca_red_file(pca_file);
```

There are two basic classes of variables in the file: ones which are unique to the PCA_RED file and ones which are replicated from the corresponding L1B file, for convenience so the user does not have to have both L1B and PCA_RED files available. Note that rd_pca_red_file converts variables to double precision floats as they are read in.

The typical dimension sizes are FOV (9), cross-track FOR (30), along-track FOR (45), and wavenumber (717 LW, 869 MW, 637 SW, 2223 all 3 bands). Note that the order of array dimensions presented here uses Matlab's convention. Dimension ordering will be reversed when inspecting the data with NetCDF tools like ncdump or with programming languages that use the opposite convention like Python.

Variables carried over from the L1B file:

(See the L1B documentation for descriptions)

obs_time_tai93:	[30 x 45 double]
lat:	[9 x 30 x 45 double]
lon:	[9 x 30 x 45 double]
lat_bnds:	[8 x 9 x 30 x 45 double]
lon_bnds:	[8 x 9 x 30 x 45 double]
land_frac:	[9 x 30 x 45 double]
sol_zen:	[9 x 30 x 45 double]
sat_zen:	[9 x 30 x 45 double]
sat_azi:	[9 x 30 x 45 double]
asc_flag:	[45x 1 double]
mean_anom_wrt_equat:	[45 x 1 double]
rad_lw_qc:	[9 x 30 x 45 double]
rad_mw_qc:	[9 x 30 x 45 double]
rad_sw_qc:	[9 x 30 x 45 double]
scan_sweep_dir:	[30 x 1 double]
for_num:	[30 x 1 double]
fov_num:	[9 x 1 double]
wnum_lw:	[717 x 1 double]
wnum_mw:	[869 x 1 double]
wnum_sw:	[637 x 1 double]

Variables unique to PCA_RED:

wnum_all:	[2223 x 1 double]	Wavenumber vector
nz_norm:	[2223 x 1 double]	NEDN vector
global_pc_score:	[150 x 9 x 30 x 45 double]	Global PC scores
local_pc_eig:	[2223 x 10 double]	Local PCs
local_pc_score:	[10 x 9 x 30 x 45 double]	Local PC scores
local_pc_mean:	[2223 x 1 double]	Local PC mean spectrum
pca_qc:	[9 x 30 x 45 double]	QC for PCA product
rad_outlier:	[2223 x 100 double]	Outlier radiance spectra
pca_red:	[25 x 9 x 30 x 45 double]	Rapid Event Detection scores

Variables not carried over from the L1B file:

There are many other variables in the L1B file that are not carried over into the PCA/RED files. See the L1B file documentation for a list of those variables. And, please let us know if there are additional variables that you would like to be included in the PCA/RED files.

Related to this, we are also considering adding a few variables from the IMG files into the PCA/RED files, for example the basic cloud fraction variable.

Computing Reconstructed Radiances:

To compute reconstructed radiances using rd_pca_red_file.m, provide as inputs the name of the Global PC file and the reconstruction option. The Global PC file can be obtained at the following URL:

https://gitlab.ssec.wisc.edu/cris_l1b/user/-/raw/a1b5144bb003c4578d89e8ef7c6c5b9257f6c4af/PCA_RED/global_pcs/SNDRGBLPCv2.J1.CRIS.20180325T0000.20190324T2354.L1B.v03_08_nznorm1_latwt1_qc2.nc?inline=false

For full (hybrid) reconstruction, use rflag = 1. Global-only and local-only radiance reconstruction are also possible.

```
>> GPCfile = 'SNDRGBLPCv2.J1.CRIS.20180325T0000.20190324T2354.L1B.v03_08_nznorm1_latwt1_qc2.nc';
>> [d,GPC] = rd_pca_red_file(pca_file,GPCfile,1);
```

Returns the additional variables of reconstructed radiances in structure d:

rad_lw	[717 x 9 x 30 x 45 double]
rad_mw	[869 x 9 x 30 x 45 double]
rad_sw	[637 x 9 x 30 x 45 double]

And the global PC variables are in structure GPC:

U	[2223 x 150]	Global PCs
M	[2223 x 1]	Global mean spectrum
D	[2223 x 1]	Global eigenvectors
v	[2223 x 1]	wavenumbers
nedn	[2223 x 1]	NEDN

The basic equation for this reconstruction is:

Rad = (GPC.U x d.global_pc_score + GPC.M) + (d.local_pc_eig x d.local_pc_score + d.local_pc_mean)

where the “x” represent matrix multiply operations, and noise normalized radiances using the NEDN values of d.nz_norm (equal to GPC.nedn) are used, as shown in rd_pca_red_file.m. This is done using the [2223 x 1] wavenumber dimensions, and then extracted into LW, MW, SW rad variables.

The **pca_qc** variable captures the Quality control of the PCA process on a footprint level basis with values of:
0: No issues detected
1: Hybrid reconstructed radiance residual exceeds threshold in any spectral channel. Original radiance stored in rad_outlier variable.
2: Hybrid reconstructed radiance residual exceeds threshold in any spectral channel. Original radiance not stored in rad_outlier variable.
3: PCA not performed on observation (not currently implemented in beta product)

For cases where the PCA QC fails, the original L1B radiance spectra are saved into the **rad_outlier** variable, with up to 100 outliers per granule. To determine where these outlier spectra are indexed (into the 9 x 30 x 45) array, the following matlab calculations can be used:

```
>> index = find(d.pca_qc == 1);
>> n = length(index);
```

```
>> Rad(:,index) = d.rad_outlier(:,1:n);
```

Subscripts can be computed from the linear indices using ind2sub.m, e.g.:

```
>> [i,j,k] = ind2sub(size(d.pca_qc),index);
```

PCA_RED

The pca_red variable flags interesting observations contained in the local PC data for 25 selected wavenumber regions. Radiance "anomalies" determined from the local PC contributions to the reconstructed radiances are scaled to have values from 0 to 127, so that these features can be easily identified. Table 1 contains the wavenumber regions and scaling information for each of the 25 cases. Feedback from beta users will be used to refine the PCA RED regions and channels within each region.

Table 1. PCA RED spectral regions and radiance scaling implemented in the beta product.

bin #	Name		wavenumbers	note	min radiance	max radiance	basis
1	longwave band	OTHER	650 - 1095		0	0.25	Hunga Tonga
2	midwave band	OTHER	1210 - 1750		0	0.1	Hunga Tonga
3	shortwave band	OTHER	2155 - 2550		0	0.02	Aus Fire
4	CO2 (LW)	OTHER	650 - 710	Cutoff at 710 to avoid C2H2, Furan, and HCN channels	0	0.4	Aus Fire, Global Maps
5	HCN (Hydrogen Cyanide)	FIRE	712.5		0	0.65	Aus Fire
6	C2H2 (Acetylene) nu5	FIRE	729.375, 730.000	Ethyne	0	0.7	Aus Fire, Quebec Fire
7	C4H4O (Furan)	FIRE	744.375		0	0.65	Aus Fire
8	C2H3NO5 (PAN)	OTHER	786.8750, 787.5000, 788.1250		0	0.25	2022 Sep 12
9	C2H6 (Ethane) nu9	OTHER	814.375,825		0	0.19	hassi messaoud oil field, Quebec fire, World Map
10	HNO3 (Nitric acid) nu5 2nu9	VOLCANO	878.125, 878.750, 879.375, 895.625, 896.25		0	0.2	Hunga Tonga, Global Maps
11	C5H8 (Isoprene) nu28	OTHER	893.125, 893.750	Channels from Wells SSTM meeting and material. Not getting good detection results in beta PCA RED.	0	0.2	Global Maps
12	C3H6 (Propylene)	FIRE	911.875, 912.500	Propene	0	0.28	Aus Fire
13	C2H4 (Ethylene) nu7	FIRE	922.5,949.375	Ethene	0	1.5	Aus Fire, Quebec Fire

bin #	Name		wavenumbers	note	min radiance	max radiance	basis
14	NH3	FIRE	930.000, 931.250, 931.875, 964.375, 965.000, 965.625, 966.250, 966.875, 967.500		0	0.25	Aus Fire, Quebec Fire
15	O3	OTHER	1049.375, 1055, 1056.25, 1059.375	Should move to bin #16 for next release (if sorting by wavenumber)	0	0.3	Global Maps
16	CH3OH (Methanol)	FIRE	1033.75		0	0.55	Aus Fire, Global Maps
17	HCOOH (Formic Acid)	FIRE	1070 - 1080		0	0.32	Aus Fire, Global Maps
18	CH4, N2O nu1 region 1	OTHER	1240 - 1275	Trying to avoid SO2, HNO3 overlap; would be good to identify specific channels to use	0	0.15	Global Maps
19	CH4, N2O nu1 region 2	OTHER	1275 - 1310	Trying to avoid SO2, HNO3 overlap; would be good to identify specific channels to use	0	0.15	Global Maps
20	HNO3 (Nitric acid) nu3	VOLCANO	1323.75, 1324.375, 1325	CH4 interfering, consider removing	0	0.23	Hunga Tonga
21	SO2 nu3	VOLCANO	1365 - 1377.500	would be good to identify specific channels to use	0	0.23	Hunga Tonga
22	1653.125 H2O line	OTHER	1653.125	1653.125	0	0.11	Hunga Tonga
23	CO	FIRE	2174.3750, 2175.625, 2178.1250, 2181.8750	near band edge: ringing?	0	0.07	Aus Fire
24	N2O nu3	OTHER	2220.625, 2221.875, 2222.5, 2223.125, 2223.75, 2225.6250	2220.6250, 2221.8750, 2222.5000, 2223.1250, 2223.7500, 2225.6250	0	0.022	Global Maps
25	CO2 SW	OTHER	2250 - 2400	will be susceptible to glint, consider creating blue spike PCA RED range	0	0.018	Quebec Fire, Global Maps

Beta Product Notes and Caveats

1. PCA RED score scaling currently uses a pre-set radiance range for each PCA RED region. The minimum radiance value is zero for all 25 PCA RED spectral regions, while the upper radiance limit is PCA RED region dependent. The lower and upper radiance limits for the PCA RED score scaling will continue to be assessed and refined as needed (beta product user feedback is appreciated).
2. PCA RED wavenumber indices: developer analysis and feedback from beta product users will be used to refine the PCA RED regions and channels used within each region for the PCA RED scores.
3. Future releases will include QC improvements to make optimal use of the L1B QC vars.
4. Future releases will further optimize performance and robustness for PCA RED product generation from L1B data with QC≠0, fill values, or missing bands.
5. The threshold value for assessing reconstructed radiance accuracy is approximately 3xNESR in the beta sample data product. This threshold may need further refinement.
6. PCA RED scores are scaled the same for ascending and descending orbits. It may produce better results to have different scaling for ascending and descending orbits.
7. PCA RED processing is currently conducted on a single granule basis without context granules. Adding context granules would likely remove any obvious granulation that is evident in the current beta product and could also improve the local reconstruction.
8. The current beta product uses the same PCA RED score scaling for all CrIS sensors.
9. The current beta product can store a maximum of 100 "outlier radiances" for each 6-minute granule. This limit could be increased in future versions if needed. Extremely large events such as the Australian wildfires or the Hunga Tonga eruption result in more than 100 outlier radiances per granule.

10. Determining the along-track, across-track, and field-of-view indexes for spectra stored in rad_outlier is error prone. The official product will provide a simplified way to retrieve the outlier indexes.
11. The current beta product uses 150 Global Eigenvectors and 10 Local Eigenvectors for the radiance reconstruction. Extensive assessment and feedback from beta users will be used to determine if the number of eigenvectors needs to be modified for future releases.
12. There is a higher rate of reconstruction residual outliers in South Atlantic Anomaly (SAA) region. This is under investigation and will be improved in future releases.
13. This beta sample product is missing many global NetCDF attributes needed to fully describe the overall file contents. A complete set of global attributes will be present on the official product.
14. The spectral_resolution is missing from the general attributes in the beta sample product.

rd_pca_red_file.m

```
function [d,GPC] = rd_pca_red_file(fname,gpcfile,rflag);

%
% function [d,GPC] = rd_pca_red_file(fname,gpcfile,rflag);
%
% Read contents of CrIS PC file and (optionally) compute reconstructed radinces
%
% Inputs
%  fname:      Name of CrIS PC file
%              e.g. SNDR.J1.CRIS.20220118T0154.m06.g020.PCA_RED.v00_01...nc
%  gpcfile     (Optional): Name of Global PC file
%              e.g.
% SNDRGBLPCv2.J1.CRIS.20180325T0000.20190324T2354.L1B.v03_08_nznorm1_latwt1_qc2.nc
%              If gpcfile is input, the output includes reconstructed radiances,
%                         and global PC data in structure GPC
%
%  rflag:      (Optional):Flag to determine what type of radiance reconstruction.
%              values: 1 or not input: Hybrid; 2: Global-only; 3: Local-only
%
% Uses: pca_expandM.m
%
% 25-Aug-2023
%

warning off

d.info = ncinfo(fname);
for vv = 1:length(d.info.Variables)
        try
                varname = d.info.Variables(vv).Name;
                d.(varname) = ncread(fname,varname);
        catch
%               fprintf(1,'Could not read %s\n',varname);
        end
end

if nargin >= 2

 GPC.U = double(ncread(gpcfile,'U'));
 GPC.M = double(ncread(gpcfile,'M'));
 GPC.D = double(ncread(gpcfile,'D'));
 GPC.v = double(ncread(gpcfile,'v'));
 GPC.nedn = double(ncread(gpcfile,'nedn'));

 globalCoef = double(d.global_pc_score(:,:));
 localCoef = double(d.local_pc_score(:,:));
 nednm = repmat(double(d.nz_norm),[1,size(globalCoef,2)]);
 localPC.U = double(d.local_pc_eig);
 localPC.M = double(d.local_pc_mean);
 RecRadGlobal = pca_expandM(globalCoef,GPC) .* nednm;  % double, [nCHAN_all x nrec]

 if nargin == 2 | (nargin == 3 & rflag == 1) % Hybrid
   RecRes = pca_expandM(localCoef,localPC) .* nednm;
   RecRad = RecRadGlobal + RecRes;
```

```
 elseif nargin == 3 & rflag == 2  % Global only
   RecRad = RecRadGlobal;
 elseif nargin == 3 & rflag == 3  % Local only
   RecRes = pca_expandM(localCoef,localPC) .* nednm;
   RecRad = RecRes;
 end

 lw_indx = find(d.wnum_all >= nanmin(d.wnum_lw) & d.wnum_all <= nanmax(d.wnum_lw));
 mw_indx = find(d.wnum_all >= nanmin(d.wnum_mw) & d.wnum_all <= nanmax(d.wnum_mw));
 sw_indx = find(d.wnum_all >= nanmin(d.wnum_sw) & d.wnum_all <= nanmax(d.wnum_sw));

 sz = size(d.global_pc_score);
 sz(1) = length(d.wnum_all);
 RecRad = reshape(RecRad,sz);
 d.rad_lw = RecRad(lw_indx,:,:,:);
 d.rad_mw = RecRad(mw_indx,:,:,:);
 d.rad_sw = RecRad(sw_indx,:,:,:);

end

return
```

pca_expandM.m

```
function [x,rc] = pca_expandM( coef, PC )
[ntr,nX] = size(coef);
PCM = repmat(PC.M,1,nX);
col = PC.U(:,1:ntr) * coef + PCM;
x   = col;
rc = 0;
```