Preparing NOVA data for analysis with Gudrun

A K Soper 3rd October 2019

Note: The following is based on what He Cheng has sent me, and may change as we gain more experience with analysing NOVA data. I am indebted to He Cheng for allowing me access to his data in order to make this development.

Introduction

Like SANDALS, GEM, NIMROD, etc., NOVA is a total scattering diffractometer with detectors in 5 banks that span angles from $2\theta = 1.9^{o}$ to ~ 170^{o}. The data are initially stored in the so-called "NeXus" (.nx) format, which in practice is simply a pseudonym for the HDF5 format.

(HDF5 is a general purpose data storage format that allows data to be accessible on a wide range of platforms in a highly compact and efficient manner. For example the speed of accessing an ISIS neutron data file written in the HDF5 format is just as quick as reading the native ISIS .raw format. Getting HDF5 to build on any particular platform, particularly Windows with the GNU compilers, can be a "pigs breakfast" as we say in the UK, but once built it appears to operate seamlessly. The point is that I would **strongly** advocate sticking to the HDF5 format for moving data around, and not convert these files to multiple .txt files, as appears to be happening, since the latter is going to generate vast numbers of files which can easily get lost, is extremely inefficient in the use of available disk space, and very slow to access for computer programs like Gudrun. Better to keep all the data from a particular sample in a single file. There is a program, HDFview which is freely available on most platforms which allows you view, indeed edit, the data in an HDF file.)

Unlike ISIS .nxs files, NOVA appears to produce separate .nx files, one for each detector bank with names like "014714_20_10steps.nx" and "014714_bs_10steps.nx", where the first number is a run number, the second is the bank label (sa, 20, 45, 90 and bs) and the remainder of the name is unclear to me as to its meaning. Since I believe NOVA has large detector arrays, I am assuming these files have already been grouped into the Debye-Scherrer cones to a significant extent since each bank contains only 11 distinct detector angles. With 5 detector banks, that makes a total of just 55 spectra to be analysed, which is hardly consistent with there being large detector arrays. However since each "detector" in these files has a flight path and a scattering angle, we can build a calibration file for NOVA from the numbers in the .nx files.

In their current form, the .nx files cannot be used directly in Gudrun, since Gudrun is expecting all the data from one run number to be in a single file. In particular the naming convention used within the .nx file is such that you cannot tell from the path names in the file which particular bank the data refer to. So the first step is to convert the multiple .nx files associated with a particular run number into a single file .nxs that can be read by Gudrun, and make the pathnames within the file tell you which particular bank they belong to.

Preparing the .nx files for Gudrun

To do this, there is an app called h5copy which comes as part of the HDF5 distribution. It can be found in the hdf5\bin folder. This program is used to convert multiple .nx files to a single .nxs file compatible with Gudrun. I have written a little batch file, "copyhdf.bat" to do this within a Command Prompt:-

```
set hdf5folder=C:\Users\user\Gudrun\Gudrun2014\Gudrun\bin\hdf5
set currentfolder=%CD%
cd NOVA-AllData\%1
%hdf5folder%\bin\h5copy -v -p -i %1_sa_10steps.nx -s Entry1/Data1 -o %currentfolder%\%1_10steps.nxs -d Entry1/Data_sa
%hdf5folder%\bin\h5copy -v -p -i %1_20_10steps.nx -s Entry1/Data1 -o %currentfolder%\%1_10steps.nxs -d Entry1/Data_20
%hdf5folder%\bin\h5copy -v -p -i %1_45_10steps.nx -s Entry1/Data1 -o %currentfolder%\%1_10steps.nxs -d Entry1/Data_45
```

```
%hdf5folder%\bin\h5copy -v -p -i %1_90_10steps.nx -s Entry1/Data1 -o %currentfolder%\%1_10steps.nxs -d Entry1/Data_90
%hdf5folder%\bin\h5copy -v -p -i %1_bs_10steps.nx -s Entry1/Data1 -o %currentfolder%\%1_10steps.nxs -d Entry1/Data_bs
cd %currentfolder%
```

The first line says where the HDF5 binaries are stored, the second sets the current folder where the .nxs files are to be generated, and the third line goes to the folder where the .nx files for the particular run number are stored.

The equivalent Bash shell script in Linux would be:-

```
cd NOVA-AllData/$1
/home/aks45/Utilities/hdf5files/hdf5/bin/h5copy -v -p -i $1_sa_10steps.nx -s Entry1/Data1 -o ../../$1_10steps.nxs -d Entry1/Data_sa
/home/aks45/Utilities/hdf5files/hdf5/bin/h5copy -v -p -i $1_20_10steps.nx -s Entry1/Data1 -o ../../$1_10steps.nxs -d Entry1/Data_20
/home/aks45/Utilities/hdf5files/hdf5/bin/h5copy -v -p -i $1_45_10steps.nx -s Entry1/Data1 -o ../../$1_10steps.nxs -d Entry1/Data_45
/home/aks45/Utilities/hdf5files/hdf5/bin/h5copy -v -p -i $1_90_10steps.nx -s Entry1/Data1 -o ../../$1_10steps.nxs -d Entry1/Data_90
/home/aks45/Utilities/hdf5files/hdf5/bin/h5copy -v -p -i $1_bs_10steps.nx -s Entry1/Data1 -o ../../$1_10steps.nxs -d Entry1/Data_bs
cd ../..
```

Obviously the user will need to adapt these scripts to their own file structure and definitions. Note that I have changed the extension from .nx to .nxs in combining these data files into a single file so that the files can be recognized by Gudrun, since Gudrun does not currently recognize .nx files.

These scripts produce a single data file for each run number, e.g. “014714_10steps.nxs”, with 5 entries, one for each detector bank. Gudrun ignores the first entry point, so the different detector banks are labelled at the second entry point, thus the entries in the .nxs files:-

/Entry1/Data_sa/ElementContainerArrayData/Vector_in_NeutronVector1/Header/ttheta
/Entry1/Data_20/ElementContainerArrayData/Vector_in_NeutronVector2/Header/ttheta
/Entry1/Data_45/ElementContainerArrayData/Vector_in_NeutronVector3/Header/ttheta

would give the 2-theta values for detector 1 in the sa bank, detector 2 in the 20º bank, and detector 3 in the 45º bank, respectively.

Inputs required by Gudrun

To run with the HDF5 files, Gudrun requires 5 other files to exist, besides the data files themselves:-

1. **Calibration file**. This I have called CalibrationFileForNOVA.calib. The extension .calib is important here: without it Gudrun will not be able to read the detector calibration. This file gives the flight path (in metres) and scattering angle (in degrees) for each spectrum. The format is hopefully transparent. Since there are 33 spectra in the sa, 20 and 45 detector banks, there must be 33 entries in this table. An example is to be found in Gudrun\StartupFiles\NOVA. I took the values from the .nx files, but obviously if different or better values become available this file should be rewritten with the correct values. So far I have not included the spectra in the 90 or bs banks in the calibration file, but it would not be difficult to add these spectra. If you add entries to this file remember to increment the number of spectra listed on the second line, otherwise the program won’t read the extra lines.

   The last item for each line of the calibration file, currently set to 0, can be used to store the azimuthal angle of the detector around the beam axis. Obviously if the actual detectors have been grouped into Debye-Scherrer cones, as was hinted above, then this value remains at 0. We have found with instruments like GEM when using highly absorbing cylindrical samples, the attenuation correction can be quite strongly affected by the azimuthal angle, so it would be important in such cases not group the detectors prior to analysis.

2. **NOVA.nexus_txt.** Again the .nexus_txt extension is essential so that the GUI can recognize the file. This file lists the HDF5 entry points for various parameters such as time-of-flight array, micro-amp hours (PROTONS), and the spectra data themselves. There is one entry for each spectrum in this case:

```
title = /Data_sa/ElementContainerArrayData/Header_in_NeutronVector/SAMPLETYPE
username = /Data_sa/ElementContainerArrayData/Header_in_NeutronVector/INSTRUMENT
starttime = /Data_sa/ElementContainerArrayData/Header_in_NeutronVector/MEASPERIOD
timechannels = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector0/ElementContainerData/TOF
periodspresent = no
muamphrs = /Data_sa/ElementContainerArrayData/Header_in_NeutronVector/PROTONS
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector0/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector1/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector2/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector3/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector4/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector5/ElementContainerData/Intensity 0 1 -1
spectrum = /Data_sa/ElementContainerArrayData/Vector_in_NeutronVector6/ElementContainerData/Intensity 0 1 -1
```

   etc.

   As more spectra are included in the analysis, the number of entries in this table will have to increase. **Important caveat:** traditionally Gudrun uses the same time-of-flight (TOF) scale for all detectors – that is the way things were done at ISIS. However I notice with NOVA there are small variations in the TOF scale for different detectors. For disordered materials work this is probably not a problem, but it might be a problem for accurate crystallography work. There is no easy workaround, other than to re-code Gudrun for multiple TOF scales as necessary. This is probably not difficult to do but so far has not been done.

3. **spec.bad.** I noticed detectors 11, 22 and 33 do not seem to contain useful data, so I have eliminated them from the data. They are listed in the file spec.bad that must be in the folder where the data analysis is taking place. They have a -1 next to them, which means they will not be used. Any number other than 0 next to spectrum number means that spectrum will not be used in the analysis.

4. **Groups file.** This determines how the spectra will be grouped in the output. Currently I have set there to be 9 groups of 3 or 5 spectra each, with 3 groups per detector bank. The groups are set by the NOVAgroups9.dat file found in the StartupFiles\NOVA folder. Obviously if the larger angle detectors are required, these extra groups will have to be added accordingly.

5. **Deadtime file.** The size of the deadtime correction is set by the NOVAdeadtime.cor file, which is also in the StartupFiles\NOVA folder. As I know nothing about the deadtime correction for NOVA detectors, the deadtime is set to 0, meaning that no deadtime correction will be performed.

Some final notes on running NOVA data analysis with Gudrun.

1. In Gudrun the spectra are numbered 1, 2, 3, 4, … whereas in the .nx and .nxs files they are numbered 0, 1, 2, 3, … This is a feature of Fortran not defaulting to 0 for the first number in a sequence.

2. In the Gudrun\run folder of the Gudrun2014 download I have included a NOVA folder with an example Gudrun input file, NOVAtest.txt, which should set things up correctly for NOVA analysis as it currently stands. In the Gudrun2014\RawData\NOVA folder I have included the scripts “copyhdf.bat” and “copyhdf.sh” to show how the .nxs file is created from the individual detector bank files.

3. When performing the iterative correction for inelastic scattering, you will get much better outputs if you click the button “Merge weights: By detector” or “… By channel”. Normally for ISIS data this has been set to “None”, but it seems NOVA may have significant mismatches between the different detector banks (possibly due to the high background) which throws off wavelength dependence if Merge weights is set to “none”.

4. I noticed that for the vanadium runs, it looks like the supplied files, 014710-014713_10steps, several runs have been added together. That is fine in principle, except that the PROTONS count did not get aggregated at the same time, so that the PROTONS count in that file was wrong. I have inserted the correct value using HDFview. In the future I would suggest to NOT aggregate the files prior to running Gudrun. Gudrun can accept multiple run numbers for any sample, and it will add the proton counts correctly in that case.

5. Because the empty instrument background is (apparently) so high on NOVA, it is necessary to use an “overall background factor” (BEAM tab) less than 1.0, otherwise the dcs levels (as reported by the .gud files) will be unreliable. To get the best factor, probably the best plan is run a sample with a known high Q limit (such as Al or Ni for example), and, making sure the sample density and dimensions are correct, adjust this factor until the correct high Q limit is achieved. Of course if a background factor of 1.0 works, then that is fine, but in the examples in the download I had to set the background facto to 0.976 to get reasonable high Q levels for all samples. It seems the same factor worked for all samples.

6. There is no transmission monitor on NOVA so the sample cross section via the sample transmission is not available. I have an application that can give an approximate cross section file for any sample, based on the ENDF/B libraries which are available at NIST. This is better than using TABLES on the sample tab, because neutron scattering cross sections are always energy dependent, and whilst you may get away with using TABLES for heavier samples, this does not apply if light atoms are present. Currently this application hasn’t been tested on Windows so it is not in the download, but I am hoping to make it available shortly.

7. Finally, just a reminder that if the operating system is Linux or OSX, Gudrun will have to be recompiled to make sure the binaries are consistent with the latest version, which allows NOVA data analysis.