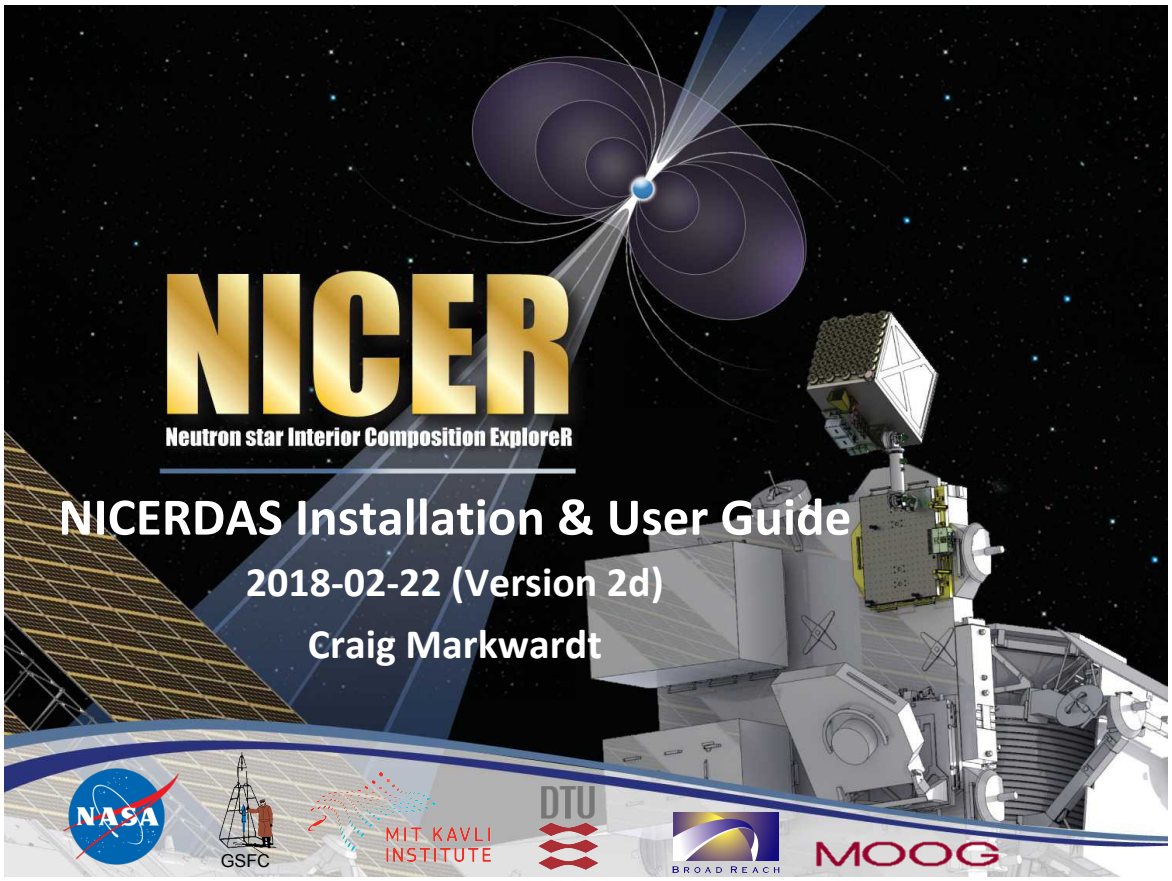

NICERDAS

- This presentation provides basic instructions to use NICER-specific tools, called NICERDAS
- NICERDAS version 3 is released with HEASoft 6.23
- NICER data in the public archive have been already processed with
  - NICERDAR version 3
  - CALDB 20180226
- Pages of this guide up to 7 are only relevant if new caldb or software is released

What is Included in NICERDAS

- NICER tools are included in this release
  - nicerl2 – apply full Level2 calibration & screening
  - nicertimecal – apply time calibration
  - nicercal – apply standard calibration
  - niextract-events – merge/filter event data
  - nimaketime – perform standard time screening
  - nicermergeclean – merge and clean event data
  - nimpumerge – merge per-MPU event data
  - nicerpi – apply single chain PI calibration
  - nimpucal – apply calibration for single MPU
  - niprefilter – create filter file (pipeline only)

Basic Analysis Steps

- One-stop shopping for calibration + screening (nicerl2)
  - Calls nicercal + nimaketime + nicermergeclean
- Apply NICER calibration (nicercal)
- Apply standard NICER time screening (nimaketime)
- Merge and clean event data (nicermergeclean)
  - Also used to merge data from multiple observations if desired
- Extract spectra or light curves (xselect)
- Perform barycenter correction (barycorr)
- See following slides for more information

One-Stop Analysis

- The 'nicerl2' script will perform all recommended NICER calibration and screening processes in one step.
  - This is the full 'Level2' process
  - It is equivalent to running nicercal, nimaketime, and nicermergeclean.
  - Most options for these individual tools are also available as options for nicerl2
- Run the task

```
nicerl2 $obs
```

- Here **$obs** is the name of your observation directory.
- The outputs are placed in $obs/xti/event_cl

Apply Calibration

- Apply calibration to data with 'nicercal'
  - As of NICERDAS 2018-02-22_V002d, nicercal computes the PI and PI_FAST energy scale, and also applies time calibration
  - Since nicercal is part of nicerl2, you only need to do one or the other
  - IMPORTANT NOTE: Initial-release data will have the best NICER team calibration at time of release. This step is only needed if a new calibration is released, or if you are applying new calibration to old or pre-release data.

```
nicercal indir=$obs
```

- Here **$obs** is the name of your observation directory.
- Outputs are placed in $obs/xti/event_cl
  (Note that previous versions of the task required you to use the outdir parameter explicitly; now it is hidden)

Apply Good Time Screening

- NICER has a task named 'nimaketime' which applies recommended screening criteria.
  - Since nimaketime is part of nicerl2, you only need to do one or the other
- Run the task

```
nimaketime infile="$obs/auxil/ni${obsroot}.mkf" \
   outfile=standard.gti
```

- The output is a GTI file which users can use downstream (here named standard.gti)
- Here $**obs** is the observation directory name and $**obsroot** is the observation number
- Type 'fhelp nimaketime' to see what kind of screening is applied.

Merge & Clean per-MPU Data

- Up until this point, data are kept on a per-MPU basis. The data must be merged and cleaned using the task nicermergeclean
  - Since nimaketime is part of nicerl2, you only need to do one or the other
- Run the task

```
ls $obs/xti/event_cl/ni${obsroot}_0mpu[0-6]_ufa.evt > ufalist.lis

nicermergeclean infiles=@ufalist.lis \
  ufafile=$obs/xti/event_cl/ni${obsroot}_0mpu7_ufa.evt \
  clfile=1706221428/xti/event_cl/ni1706221428_0mpu7_cl.evt \
  gtifile=standard.gti
```

- The first 'ls' command lists the calibrated 'ufa' files. The second nicermergeclean command merges and cleans the inputs. The GTI (standard.gti) is the result of the previous nimaketime command (see previous slide).
- Here $**obs** is the observation directory name and $**obsroot** is the observation number

Variation: merge per-MPU data from more than one observation

- The nicermerge command can be used to merge per-MPU data from multiple observations

```
ls $obs1/xti/event_cl/ni${obsroot1}_0mpu[0-6]_ufa.evt > ufalist.lis
ls $obs2/xti/event_cl/ni${obsroot2}_0mpu[0-6]_ufa.evt >> ufalist.lis

nicermergeclean infiles=@ufalist.lis \
  ufafile=merged_ufa.evt \
  clfile=merged_cl.evt
```

- Here **$obs1** and **$obs2** are the two observations to merge (corresponding to **$obsroot1** & **$obsroot2**). The 'ls' commands are used to print files from both observations into a single file list ufalist.lis, and then all of those files are merged.
- If you want to apply time screening, then you will need either:
  - Merge filter files (.mkf files) from obs1 & obs2, then run nimake time on result, OR
  - Make individual observation GTI files using nimaketime on obs1 and obs2, then merge the resulting GTIs with ftmgtime (and 'OR' combining)

Merge UFA "MPU7" data from more than one observation

- The previous example will not work for data from the archive which has already been merged by MPU. The result is a "ufa" file (lightly screened) labeled with "mpu7". Here "mpu7" means all MPUs have been merged into a single file.
- Instead, the nimpumerge command can be used to combine merged-MPU "ufa" data from multiple observations

```
ls */xti/event_cl/ni*_0mpu7_ufa.evt > ufalist.lis

nimpumerge infiles=@ufalist.lis outfile=merged_ufa.evt mpulist=7
```

- Here the wildcard "*" is used to select all observations in the current directory. You will want to change your "ls" command to select which desired observations (example. "ls 10120101*"). The 'ls' command is used to print files from multiple observations into a single file list ufalist.lis.
- You can then use nicerclean to screen this data.

```
nicerclean infile=merged_ufa.evt outfile=merged_cl.evt
```

- If you want to apply time screening, then you will need either:
  - Merge filter files (.mkf files) from the individual observations using ftmerge, then run nimake time on result, OR
  - Make GTI files using nimaketime on each observation, then merge the resulting GTIs with ftmgtime (and 'OR' combining)

What to do next?

- You can use the resulting cleaned event file with xselect to extract a spectrum or light curve.
  - Example xselect session:

```
> read events ./ni1707030136_0mpu7_cl.evt
> set binsize 1
> extract curve
> plot curve
> set phaname PI
> extract spectrum
> plot spectrum
```


Barycentering

- You can apply barycentering to NICER data using the barycorr command.

```
barycorr infile=orig.evt outfile=bary.evt \
  orbitfiles="$obs/auxil/ni${obsroot}.orb" ra=123.45 dec=-67.890 \
  refframe=ICRS
```

- Change the RA and Dec for your target, of course
- Here $**obs** is the observation directory name and $**obsroot** is the observation number
- Use the .orb orbit file for input, and not the filter file (.mkf). The .orb file has the highest precision. In rare cases, the standard filtered orbit solution may not be functioning. In that case the SPS_ORBIT extension can be used ni${obsroot}.orb[SPS_ORBIT]
- HIGHLY RECOMMEND: using refframe=ICRS, which selects a modern ephemeris. If you don't do this, there can be millisecond-errors!