This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

Observations (2):
Converting observations to IODA format & HofX Application

Presented by Ivette Hernández Baños
Based on materials prepared by Junmei Ban

Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

NCAS MPAS-JEDI Tutorial, St Andrews, UK
June 25-26, 2025

Outline

IODA: Interface for Observational Data Access

1. Observation types in MPAS-JEDI
2. Converting observations to IODA format
3. HofX Application

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_b)^{\mathbf{T}}\mathbf{B}^{-1}(\boldsymbol{x}-\boldsymbol{x}_b) + \frac{1}{2}[H(\boldsymbol{x})-\boldsymbol{y}]^{\mathbf{T}}\mathbf{R}^{-1}[H(\boldsymbol{x})-\boldsymbol{y}]$$

This talk focus on:

- ***y*** → Observations
- ***H*(x)** → calculate model equivalents of the observations; computed through the forward operator

Observation types in MPAS-JEDI

❑ **Non-Radiances**:

- Aircraft (U, V, T, spechum)
- Sondes (U, V, T, spechum)
- Surface pressure (surface synoptic observations (SYNOP), METAR, ships, drifting buoys and CMAN station reports)
- atmospheric motion vectors (AMVs) (NCEP prepBURF and BURF files)
- GNSS radio occultation
  - bending angle
  - atmosphere refractivity

❑ **Radiances** (using CRTM or RTTOV):

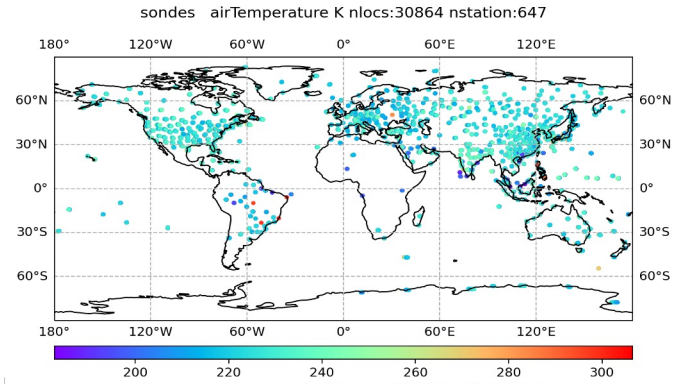
- AMSU-A (NOAA-15–16, NOAA-18–19, EOS-Aqua, MetOp-A–B)
- MHS (NOAA-18–19, MetOp-A–B)

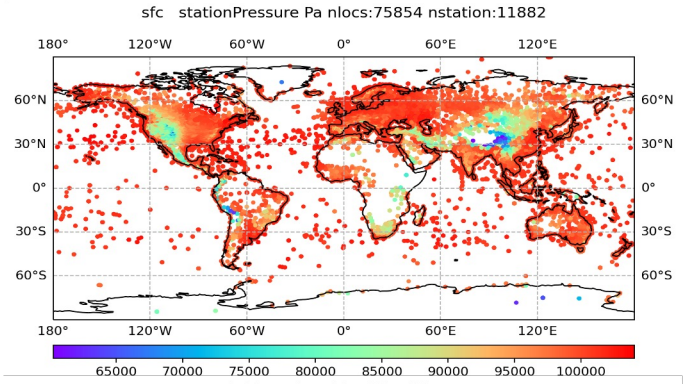
Testing mode:

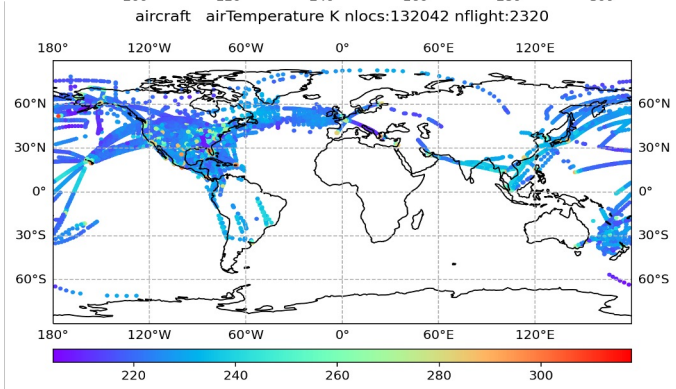
- ATMS (Suomi NPP, NOAA-20–21)
- IASI (MetOp-A–B)
- CrIS (Suomi NPP, NOAA-20, JPSS-2)
- GMI (GPM)

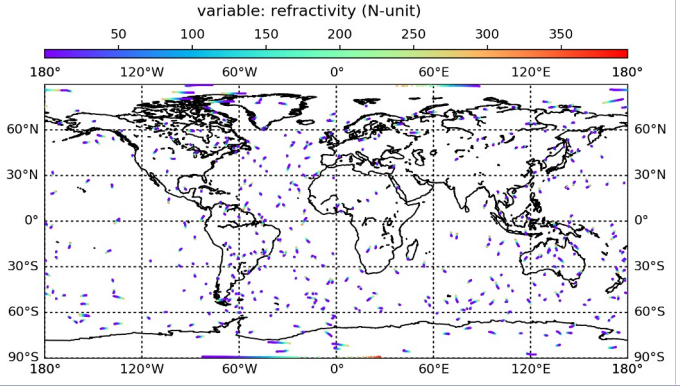
Observation types in MPAS-JEDI

Obs coverage
00Z 15 April 2018







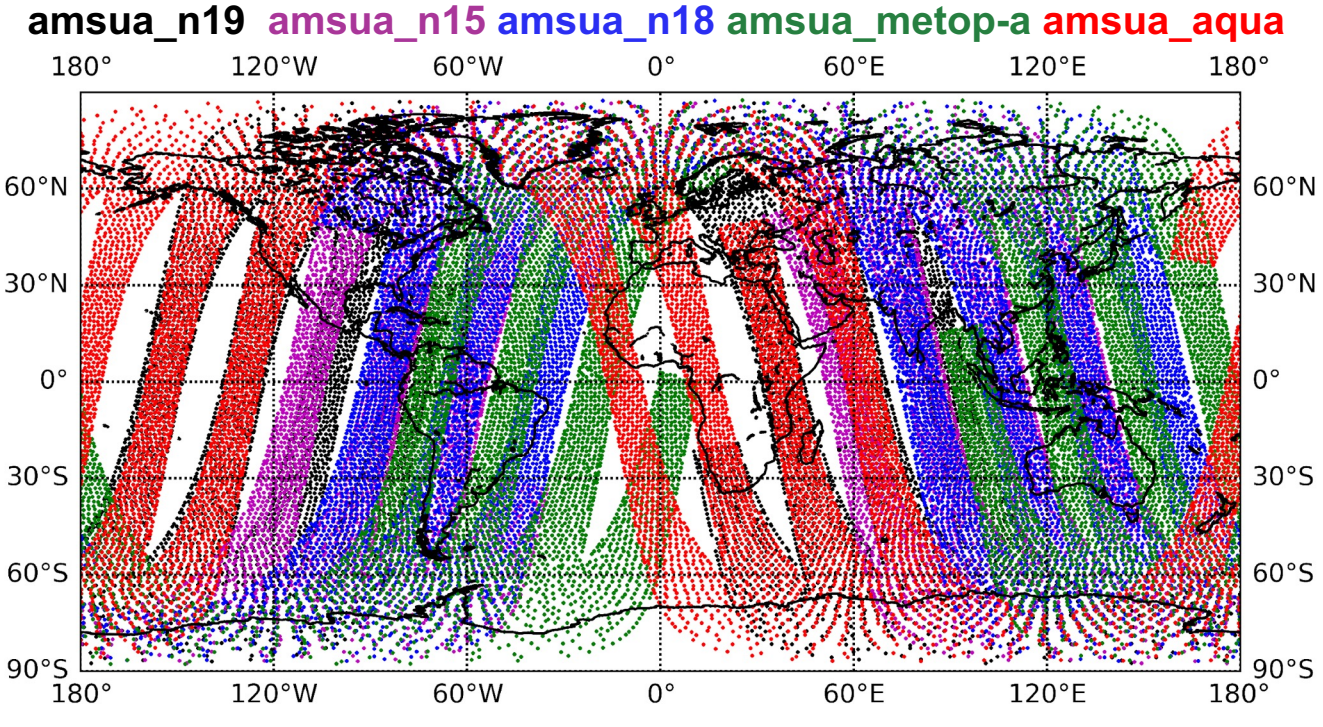


Observation types in MPAS-JEDI

AMSU-A

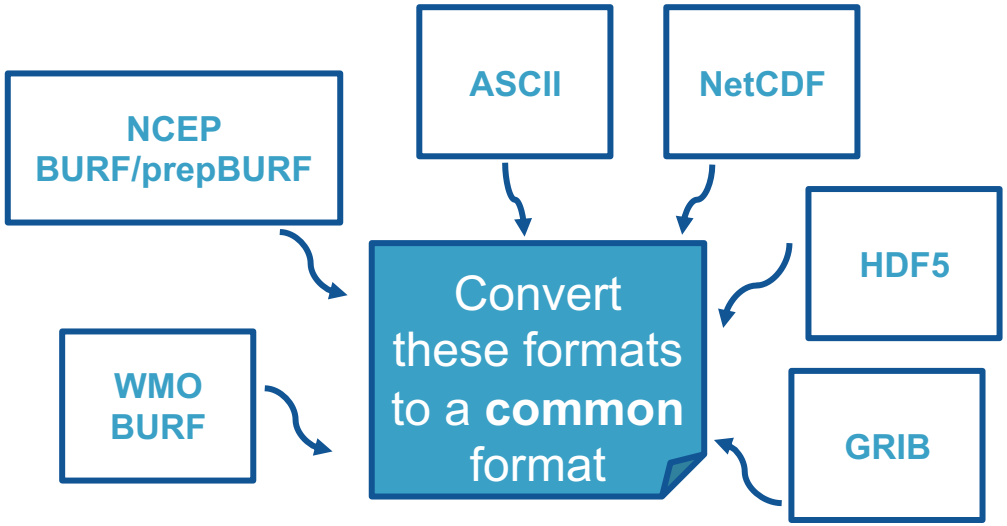
Obs coverage
12Z 18 April 2018

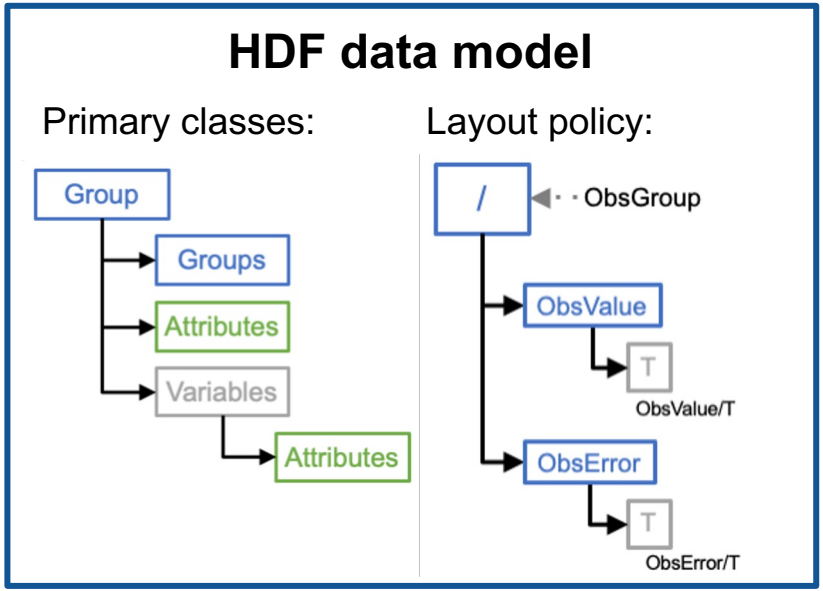
Thinning : 145km



Converting observations to IODA format

What is the IODA format?





https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/ioda/introduction.html

Converting observations to IODA format

NSF NCAR obs2ioda converter

1. Source code: git clone https://github.com/NCAR/obs2ioda

   Dependencies:

   - ❑ CMake
   - ❑ NCEP BUFR library:
     git clone https://github.com/NOAA-EMC/NCEPLIBS-bufr
   - ❑ NetCDF library
   - ❑ Intel or **GNU** compilers
   - ❑ Python packages pytest, NetCDF4 requests
     - ★ only to run the whole test suite

Converting observations to IODA format

NCAR obs2ioda converter

2. Set NCEP BUFR library path and build type (**Release**, RelWithDebInfo, and Debug):

```
cmake <OBS2IODA_ROOT_DIR> -
DNCEP_BUFR_LIB=<NCEP_BUFR_LIB_PATH> -
DCMAKE_BUILD_TYPE=<BUILD_TYPE>
```

★ Optionally, you can build the IODA converter for GOES-ABI data, by explicitly setting a build flag as -DBUILD_GOES_ABI_CONVERTER=ON (not covered in this tutorial)

Converting observations to IODA format

NCAR obs2ioda converter

3. Make to compile the code:

```
make
```

4. Run ctest to the compilation is correct:

```
ctest
```

4. (Optionally) Run the test suite:

```
pytest
```

Successful compilation produces the executable: obs2ioda_v3

Converting observations to IODA format

BUFR and PREPBUFR format

NCEP operational observation files in BUFR and PREPBUFR format:

- ❑ NCEP real-time data

  http://www.ftp.ncep.noaa.gov/data/nccf/com/gfs/prod

- ❑ NSF NCAR CISL archive

  http://rda.ucar.edu/datasets/ds337.0

  http://rda.ucar.edu/datasets/ds735.0

**If you have an account on Derecho (or Casper) HPC:*

/glade/campaign/collections/rda/data/ds337.0 ⇒ prepBUFR

/glade/campaign/collections/rda/data/ds735.0 ⇒ BURF

Files to look for:

prep48h

└── ${yyyy}/prepbufr.gdas.${yyyy}${mm}${dd}.t${hh}z.nr.48h

satwnd

└── ${yyyy}/satwnd${yyyy }${mm}${dd}.tar.gz

gpsro

└── ${yyyy}/gpsro${yyyy}${mm}${dd}.tar.gz

1bamua

└── ${yyyy}/1bamua.${yyyy}${mm}${dd}.tar.gz

Converting observations to IODA format

BUFR and prepBUFR to IODAv3-HDF5

Usage: obs2ioda_v3 [-i input_dir] [-o output_dir] [bufr_filename(s)_to_convert] [-noqc] [-split]

**input_dir and output_dir: optional augment*

Output: IODA v3 format

Example output files:

aircraft_obs_YYYYMMDDHH.h5
satwind_obs_YYYYMMDDHH.h5
sfc_obs_YYYYMMDDHH.h5
sondes_obs_YYYYMMDDHH.h5

amsua_aqua_obs_2018041500.h5
amsua_n15_obs_2018041500.h5
amsua_metop-b_obs_2018041500.h5
gnssrobndropp1d_obs_2018041500.h5
…

Converting observations to IODA format

BUFR and prepBUFR to IODAv3-HDF5

Usage: obs2ioda_v3 [-i input_dir] [-o output_dir] [bufr_filename(s)_to_convert] [-noqc] [-split]

About observation errors:

- Observation errors of conventional data are either extracted from the input prepBUFR or from an external error table (if obs_errtable exists in the working directory)
- Observation errors of AMSU-A/MHS radiances are coded in **define_mod.f90**
    - Observation errors of **satwnd-decoded AMVs** are from an external error table (**obs_errtable**)

About quality controls (QC):

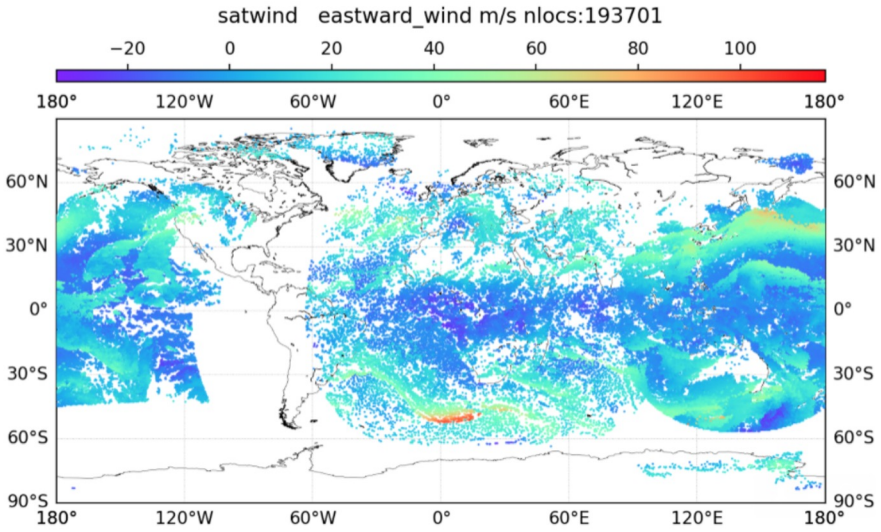
* Subroutine **filter_obs_conv** applies some additional QC for conventional observations as in **GSI's read_prepbufr.f90** for the global model and can be deactivated through **-noqc** command-line option

About time window:

- Default is 6-hourly; **-split** enables hourly data

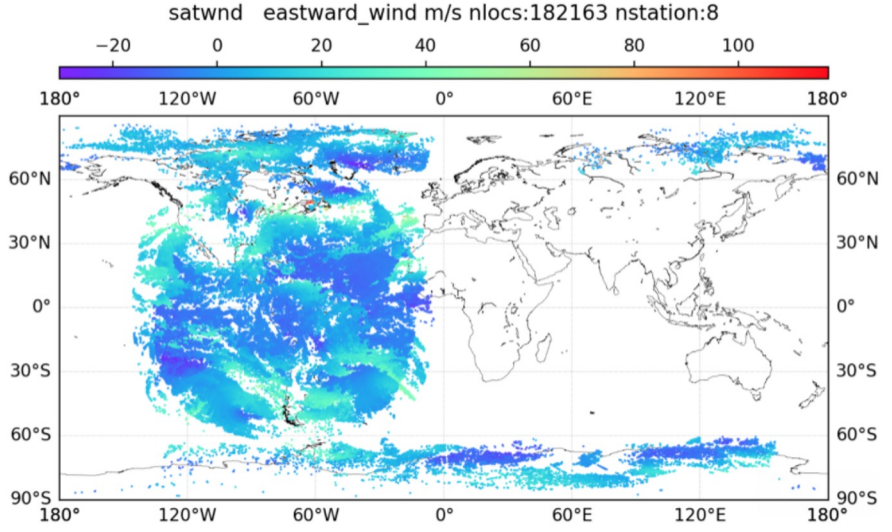
Converting observations to IODA format

Satellite wind converted from prepBUFR and BUFR are complementary and should be assimilated **together**



Satellite wind converted from prepBUFR file

Other AMVs are from PREPBUFR files



Satellite wind converted from BUFR file

Includes **GOES-16/GOES-17, AVHRR (METOP/NOAA) and VIIRS (NPP/NOAA) polar AMVs,** also LEOGEO AMVs

Converting observations to IODA format

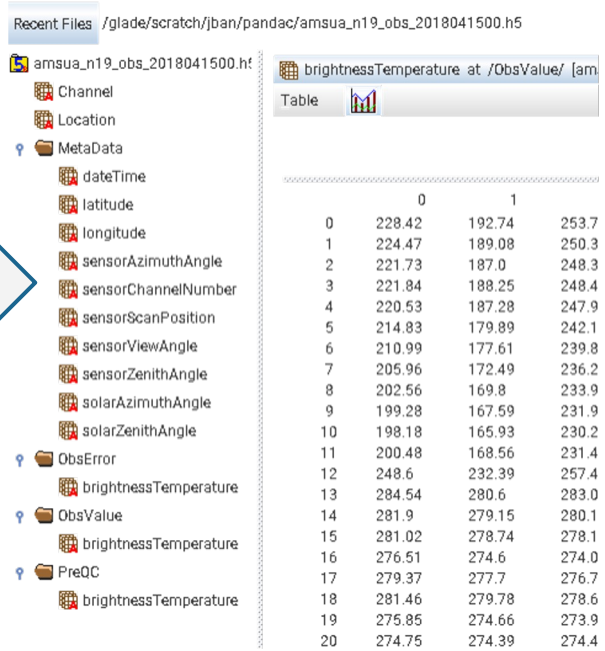
Tools to check IODA observations

- ncdump, h5dump, hdfview, Python h5py, …

h5dump –n filename

```
HDF5 "obs_iodav3/obsiodav3_221216/newobs_2018/raw_obs/20180
FILE_CONTENTS {
 group      /
 dataset    /Channel
 dataset    /Location
 group      /MetaData
 dataset    /MetaData/dateTime
 dataset    /MetaData/latitude
 dataset    /MetaData/longitude
 dataset    /MetaData/sensorAzimuthAngle
 dataset    /MetaData/sensorChannelNumber
 dataset    /MetaData/sensorScanPosition
 dataset    /MetaData/sensorViewAngle
 dataset    /MetaData/sensorZenithAngle
 dataset    /MetaData/solarAzimuthAngle
 dataset    /MetaData/solarZenithAngle
 group      /ObsError
 dataset    /ObsError/brightnessTemperature
 group      /ObsValue
 dataset    /ObsValue/brightnessTemperature
 group      /PreQC
 dataset    /PreQC/brightnessTemperature
 }
}
```

hdfview

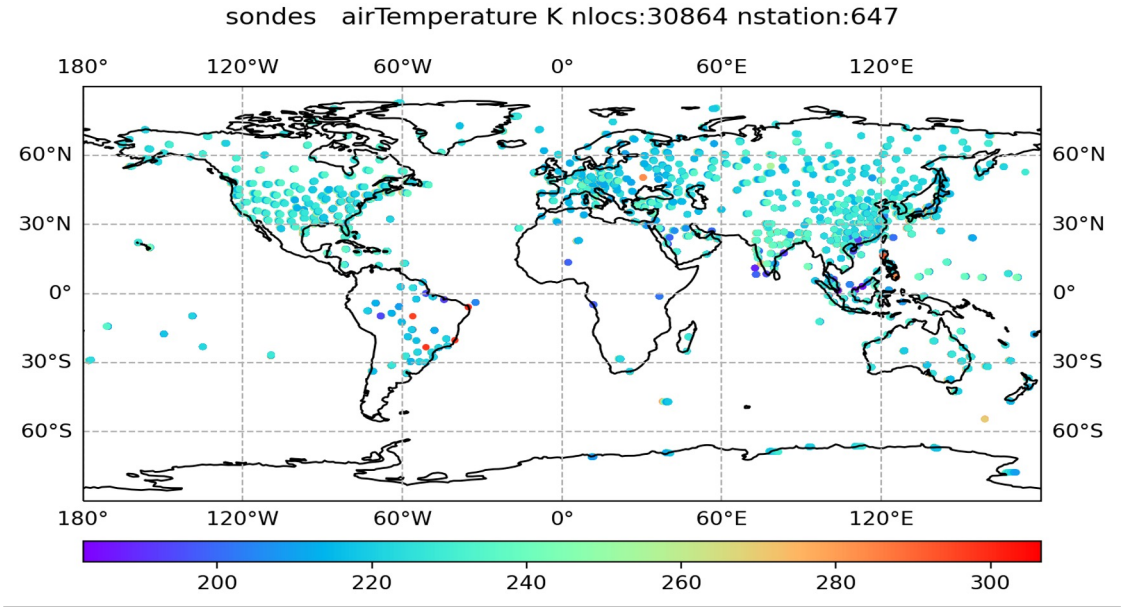


Converting observations to IODA format

Plotting observation locations

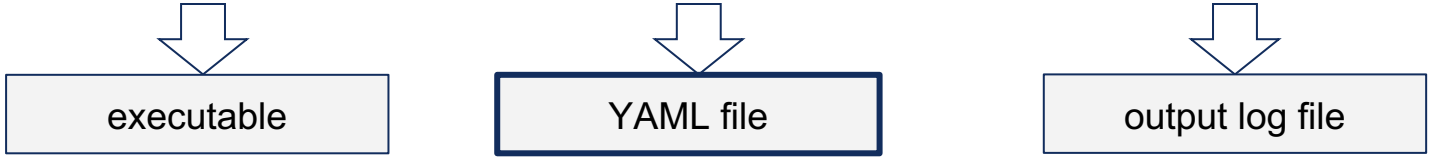
Under **graphics**:

```
├── standalone
    ├── plot_obs_loc_tut.py
```



HofX Application

mpasjedi_hofx3d.x ./hofx3d.yaml ./mpasjedi_hofx3d.log




```
24 window begin: 2018-04-14T21:00:00Z
25 window length: PT6H
26 geometry:
27   nml_file: ./namelist.atmosphere_240km
28   streams_file: ./streams.atmosphere_240km
29   deallocate non-da fields: true
30 state:
31   state variables: [spechum,surface_pressure,temperature,uReconstructMer
   typ,isltyp,snowh,vegfra,u10,v10,lai,smois,tslb,pressure_p]
32   filename: ./bg/bg.2018-04-15_00.00.00.nc
33   date: 2018-04-15T00:00:00Z
34 observations:
```

window begin:

 # datetime in ISO format

window length:

 # duration in ISO format

geometry:

 # geometry of the model

state:

 # model state used for computing H(x)

HofX Application

mpasjedi_hofx3d.x ./hofx3d.yaml ./mpasjedi_hofx3d.log

executable | YAML file | output log file

```
11 observations:
12   observers:
13   - obs space:
14       name: Aircraft
15       obsdatain:
16         engine:
17           type: H5File
18           obsfile: ./aircraft_obs_2018041500.h5
19       obsdataout:
20         engine:
21           type: H5File
22           obsfile: ./obsout_hofx_aircraft.h5
23       simulated variables: [airTemperature, windEastward, windNorthward, specificHumidity]
24     obs operator:
25       name: VertInterp
```

observation:

observers:

-obs space:

obs operator:

-obs space:

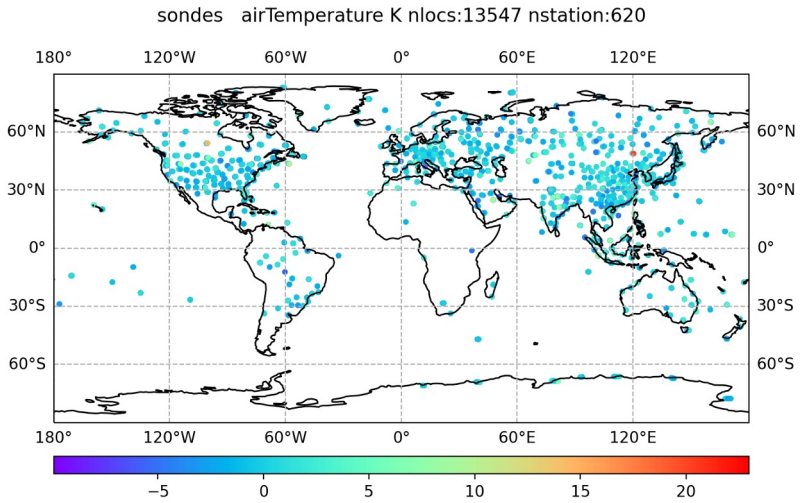
obs operator:

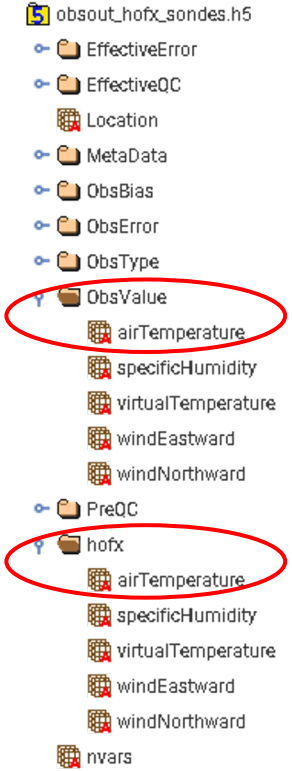
INDENTATION!!!

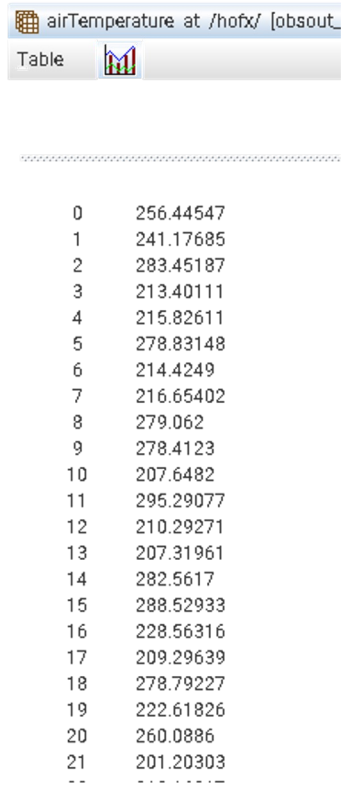
HofX Application

Output: obsout_hofx_sondes.h5

Observations departure: O-B







To learn more

- IODA: Interface for Observation Data Access
  https://github.com/JCSDA/ioda
    - Other converters: https://github.com/JCSDA/ioda-bundle
- UFO: Unified Forward Operator
  https://github.com/JCSDA/ufo