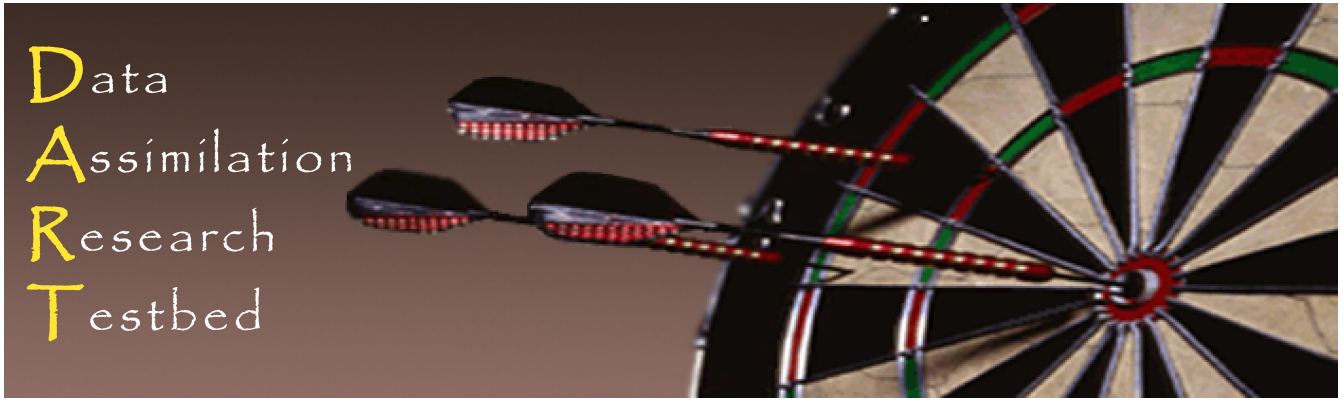
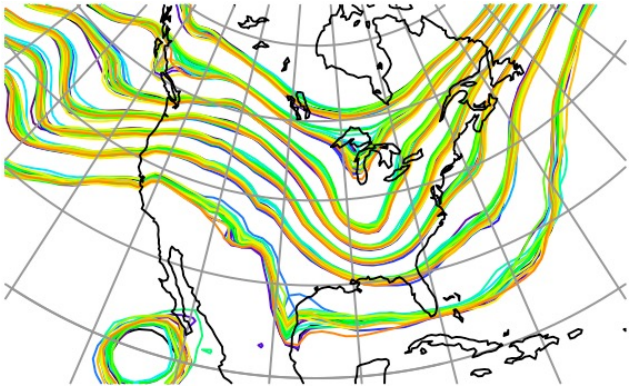


DART Tutorial Section 18: Lost in Phase Space: The Challenge of Not Knowing the Truth.



The National Center for Atmospheric Research is sponsored by the National Science Foundation. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR | National Center for
UCAR | Atmospheric Research

Reality Strikes

In real applications, **the truth is unknown**.

All that we have are observations.

Having the truth available has been convenient,
but also misleading.

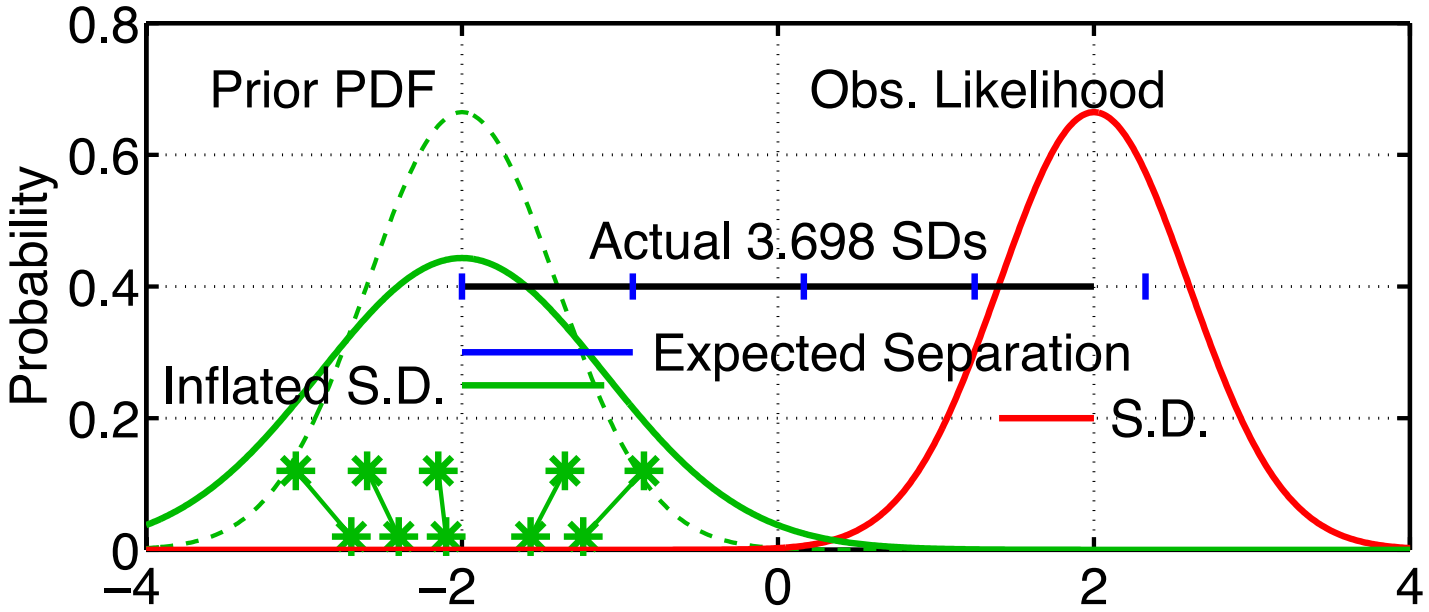
Much less information is available from the observations.

- They are generally functions of the state variables.
- They are always contaminated with observational errors.

What to expect …

Recall that

Expected(prior_mean – observation) = $\sqrt{\sigma_{prior}^2 + \sigma_{obs}^2}$



Error is dominated by observational noise if $\sigma_{obs}^2 \gg \sigma_{prior}^2$

Suppose $\sigma_{obs} = 1.0, \sigma_{prior} = 0.1$, then $\mathrm{E(RMS)} = 1.005$.

Halving σ_{prior} to 0.05 => $\mathrm{E(RMS)} = 1.001$; only a 0.4% reduction!

First Observation-space diagnostics:

Whether or not to assimilate or reject observations based on their Expected Separation is controlled during ***filter*** based on namelist settings in *input.nml*.

If $\left|\overline{y^p} - y^o\right| \Big/ \sqrt{\sigma^2_{prior} + \sigma^2_{obs}} > \text{outlier\_threshold}$ ⇒ Observation rejected! (DART QC ==7)

```
&quality_control_nml
   input_qc_threshold       = 3.0
   outlier_threshold        = -1.0
 /
```

Negative value means USE ALL

The program ***obs_diag*** post-processes *obs_seq.final*, calculates metrics like RMSE, bias, ensemble spread, totalspread, # of observations used or rejected … Start with the *lorenz_96* model.

Observation-space diagnostics

The observation sequence file is not in a particularly user-friendly format.
To aid in the evaluation and interpretation, a program named ***obs_diag*** must be run to produce a netCDF file with results that can be plotted in a manner of your choosing. DART has Matlab functions/scripts that create high-quality graphics.
For up-to-date information on the latest, greatest diagnostics, go to:
http://www.image.ucar.edu/DAReS/DART/DART2_Diagnostics.php#obs_diagnostics

```
&obs_diag_nml
   obs_sequence_name     = 'obs_seq.final',
   bin_width_days        = -1,
   bin_width_seconds     = -1,
   init_skip_days        = 0,
   init_skip_seconds     = 0,
   Nregions              = 3,
   trusted_obs           = 'null',
   lonlim1               = 0.00, 0.00, 0.50, -1.0
   lonlim2               = 1.01, 0.50, 1.01, -1.0
   reg_names             = 'whole', 'yin', 'yang', 'bogus,
   create_rank_histogram = .true.,
   outliers_in_histogram = .true.,
   use_zero_error_obs    = .false.,
   verbose               = .false.
   /
```

(Slightly different for 3D models.)

Observation-space diagnostics

The observation sequence file is not in a particularly user-friendly format.
To aid in the evaluation and interpretation, a program named ***obs_diag*** must be run to produce a netCDF file with results that can be plotted in a manner of your choosing. DART has Matlab functions/scripts that create high-quality graphics.
For up-to-date information on the latest, greatest diagnostics, go to:
http://www.image.ucar.edu/DAReS/DART/DART2_Diagnostics.php#obs_diagnostics

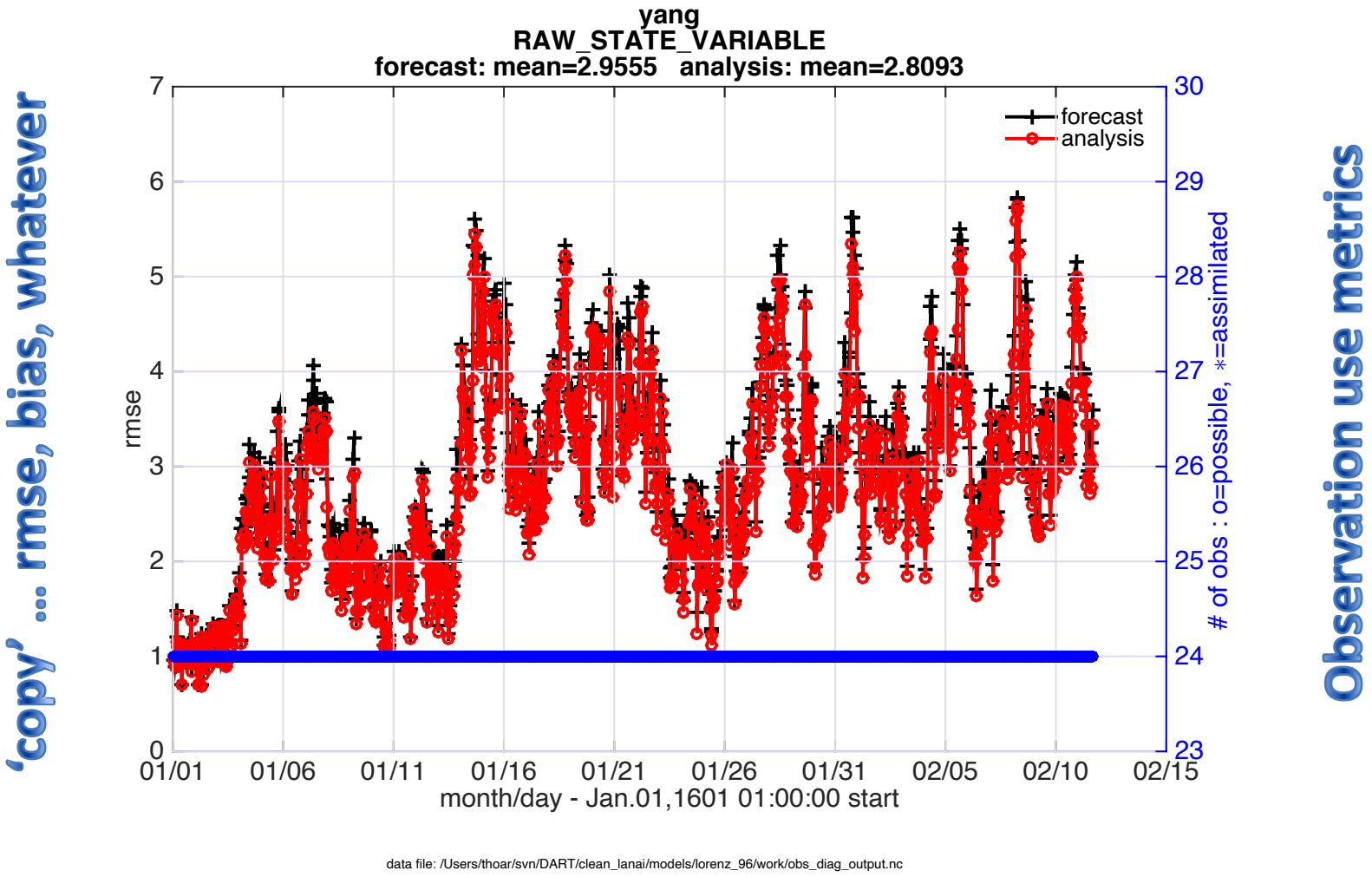
Here are a few of the Matlab functions available in *diagnostics/matlab*

- **plot_rank_histogram.m**
- **plot_evolution.m**
- **plot_rmse_xxx_evolution.m**
- **two_experiments_evolution.m**
- **plot_profile.m**
- **plot_bias_xxx_profile.m**
- **plot_rmse_xxx_profile.m**
- **two_experiments_profile.m**

These work with ANY *obs_seq.final* from ANY experiment with ANY model!

Lorenz 96 observation diagnostic example

outlier_threshold = -1.0



First Observation-space diagnostics:

Try setting the rejection threshold to a small positive number and rerunning ***filter***, and then rerunning ***obs_diag*** on the new output file.

```
&filter_nml
   …
   obs_sequence_in_name      = "obs_seq.out"
   obs_sequence_out_name     = "obs_seq.final"
   …
  /

&quality_control_nml
   input_qc_threshold        = 3.0,
   outlier_threshold         = -1.0,
   /
```

Change to what-ever you like.

Change to 3.0

Don't forget to rerun ***filter***!

Don't forget to rerun ***obs_diag***!

Don't forget to use the right filename in *obs_diag_nml*!

This is potentially **DANGEROUS**, but useful.

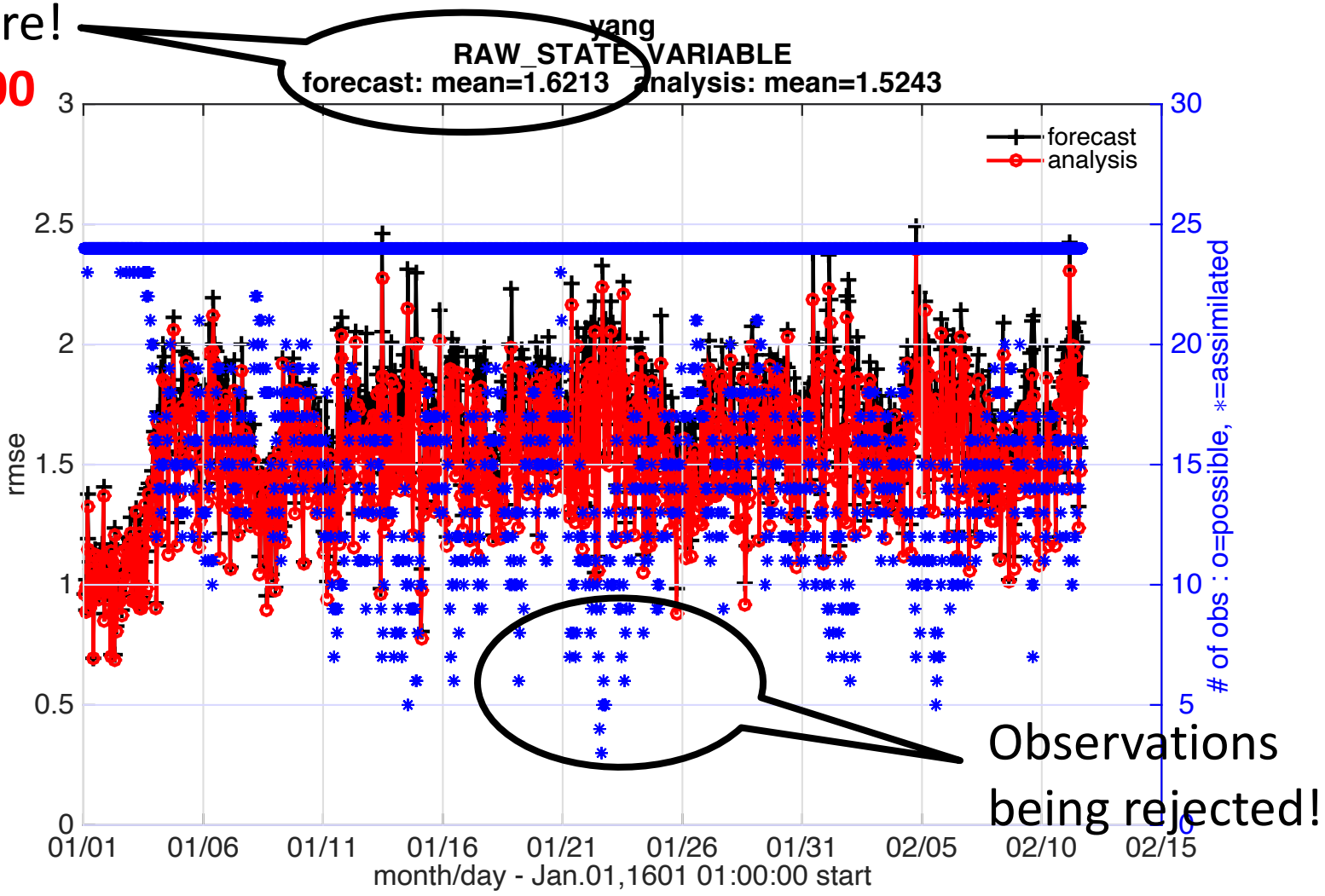
Rejecting 'good' observations can lead to inflated estimate of quality.

First Observation-space diagnostics:

Lower RMSE than before!

$1,000,000 question: Why?

outlier_threshold = 3.0



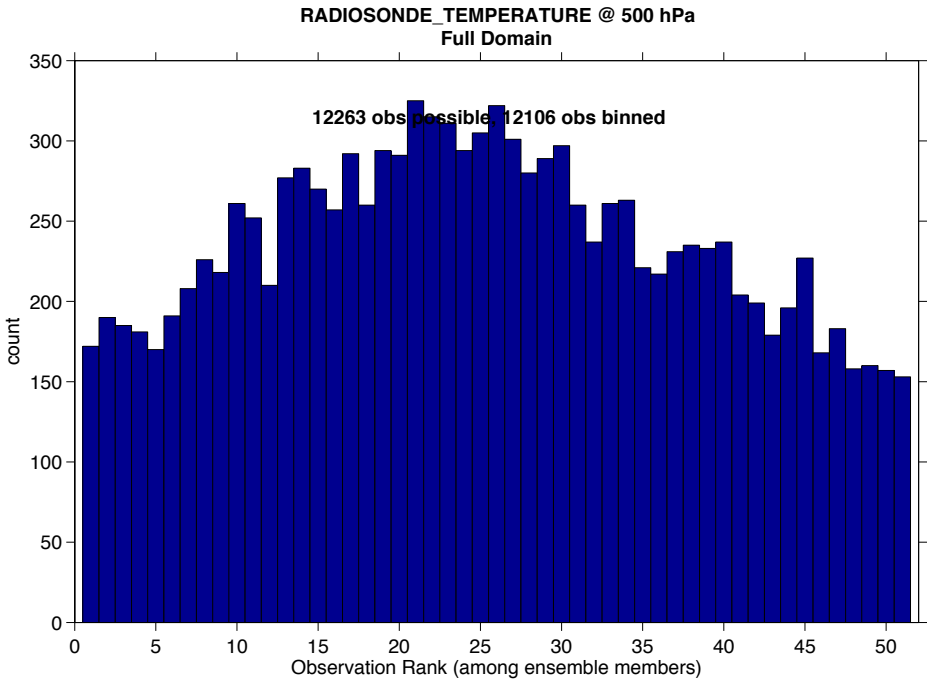
data file: /Users/thoar/svn/DART/clean_lanai/models/lorenz_96/work/obs_diag_output.nc

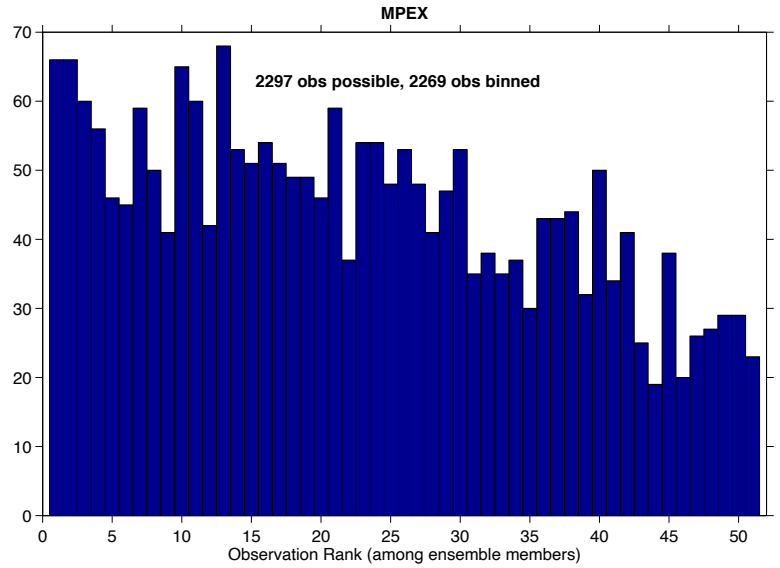
Lorenz_96 exercises:

- Pick a case that works relatively well and look at observation-space diagnostics.
- Pick a case that is similar, but clearly different, with physical-space diagnostics.
- See if you can detect the difference with observation-space diagnostics.
- Rerun ***obs_diag*** with different *bin_widths*.

Observation-space diagnostics: rank histograms

```
>> fname = 'obs_diag_output.nc';
>> timeindex = -1;
>> varname = 'RADIOSONDE_TEMPERATURE';
>> plot_rank_histogram(fname, timeindex, varname);
```



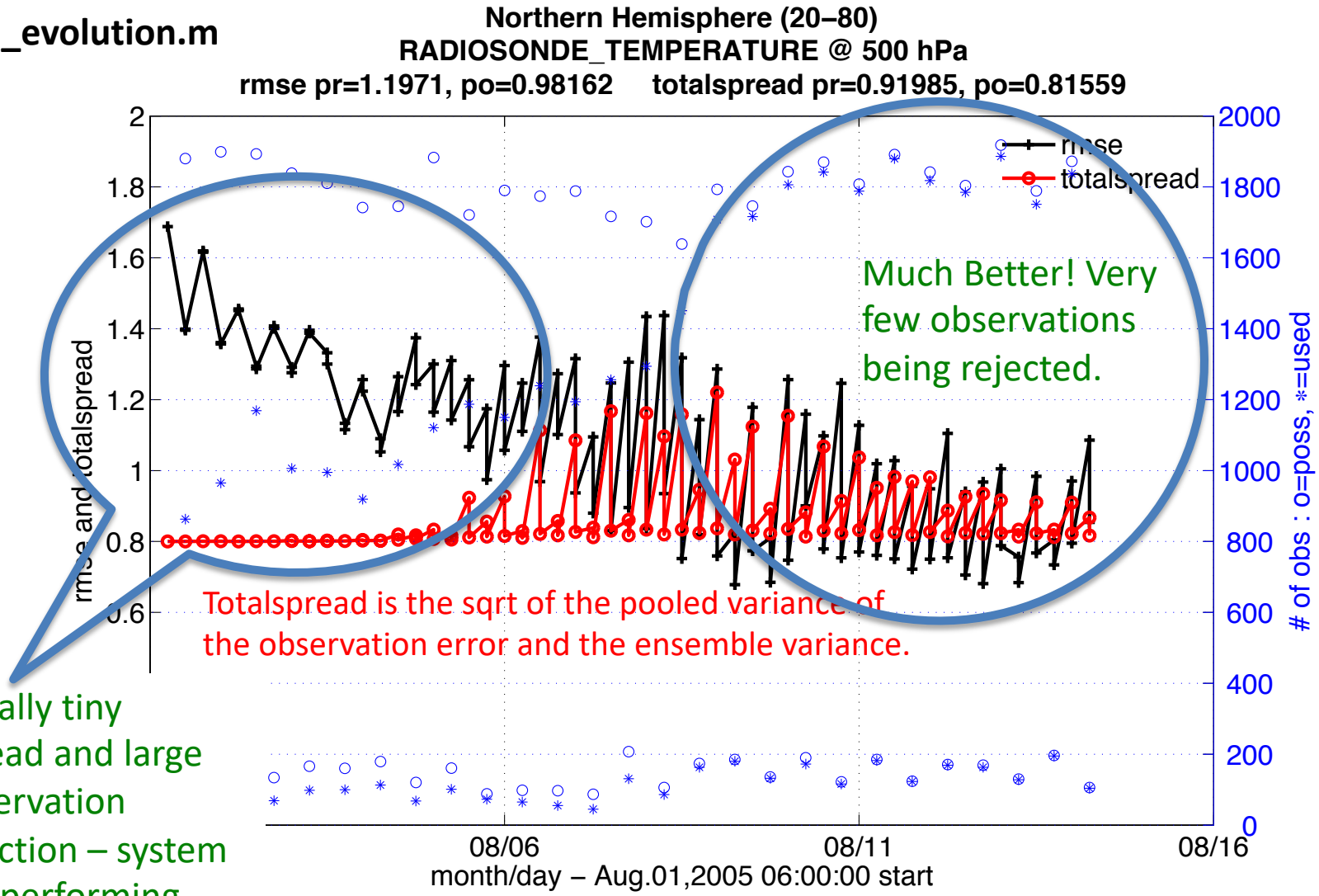


Results from WRF real-time forecasting.

Observation-space diagnostics: time evolution (by level)

plot_rmse_xxx_evolution.m

plot_evolution.m

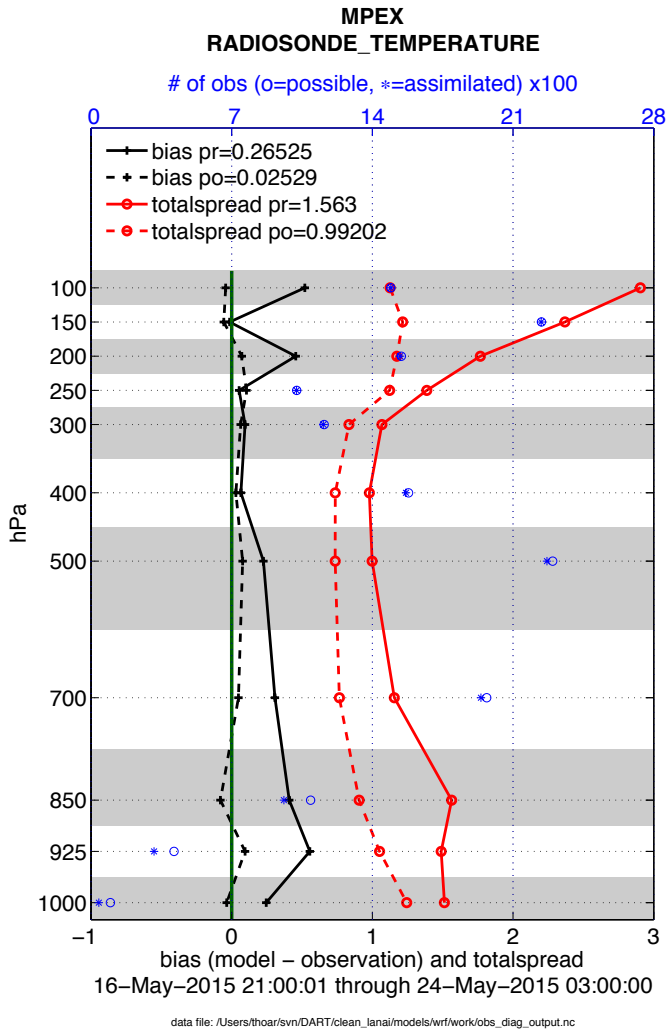


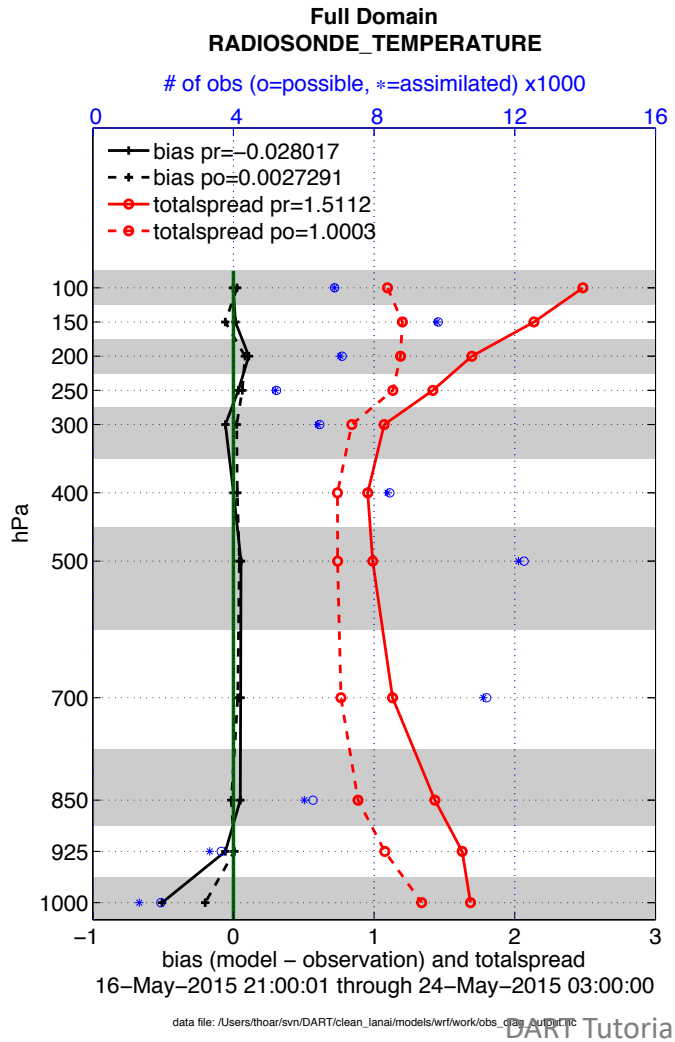
Initially tiny spread and large observation rejection – system not performing well – yet!

Observation-space diagnostics: time-averaged profiles

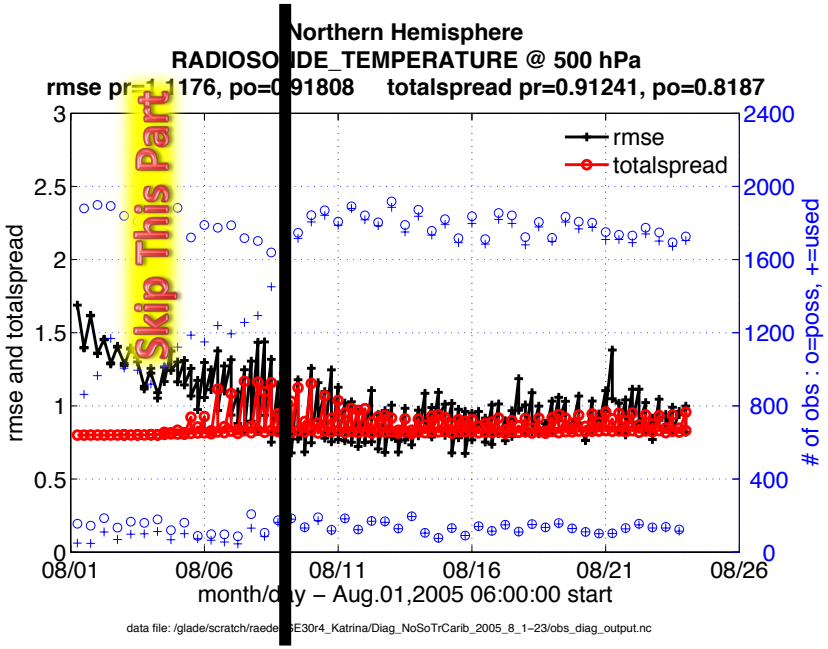
plot_profile.m plot_bias_xxx_profile.mplot_rmse_xxx_profile.m

Note: These are much more informative for models with levels! (i.e. the 1D models are not very interesting this way)





A word of warning …



NOTE: The `&obs_diag_nml` is different for low-order models and realistic models. Check carefully when using.

```
&obs_diag_nml
   obs_sequence_name = ''
   obs_sequence_list = 'file_list.txt'
   first_bin_center =  2005, 8, 1, 6, 0, 0
   last_bin_center  =  2005, 8,26, 0, 0, 0
   bin_separation   =     0, 0, 0, 6, 0, 0
   bin_width        =     0, 0, 0, 6, 0, 0
   time_to_skip     =     0, 0,10, 0, 0, 0
   max_num_bins     = 1000
   trusted_obs      = 'null'
   …
/
```

obs_diag '*time_to_skip*' setting will allow you to ignore the spinup before starting the time-averaging for the vertical profiles while still calculating metrics for the entire period of record for the time-evolution products.

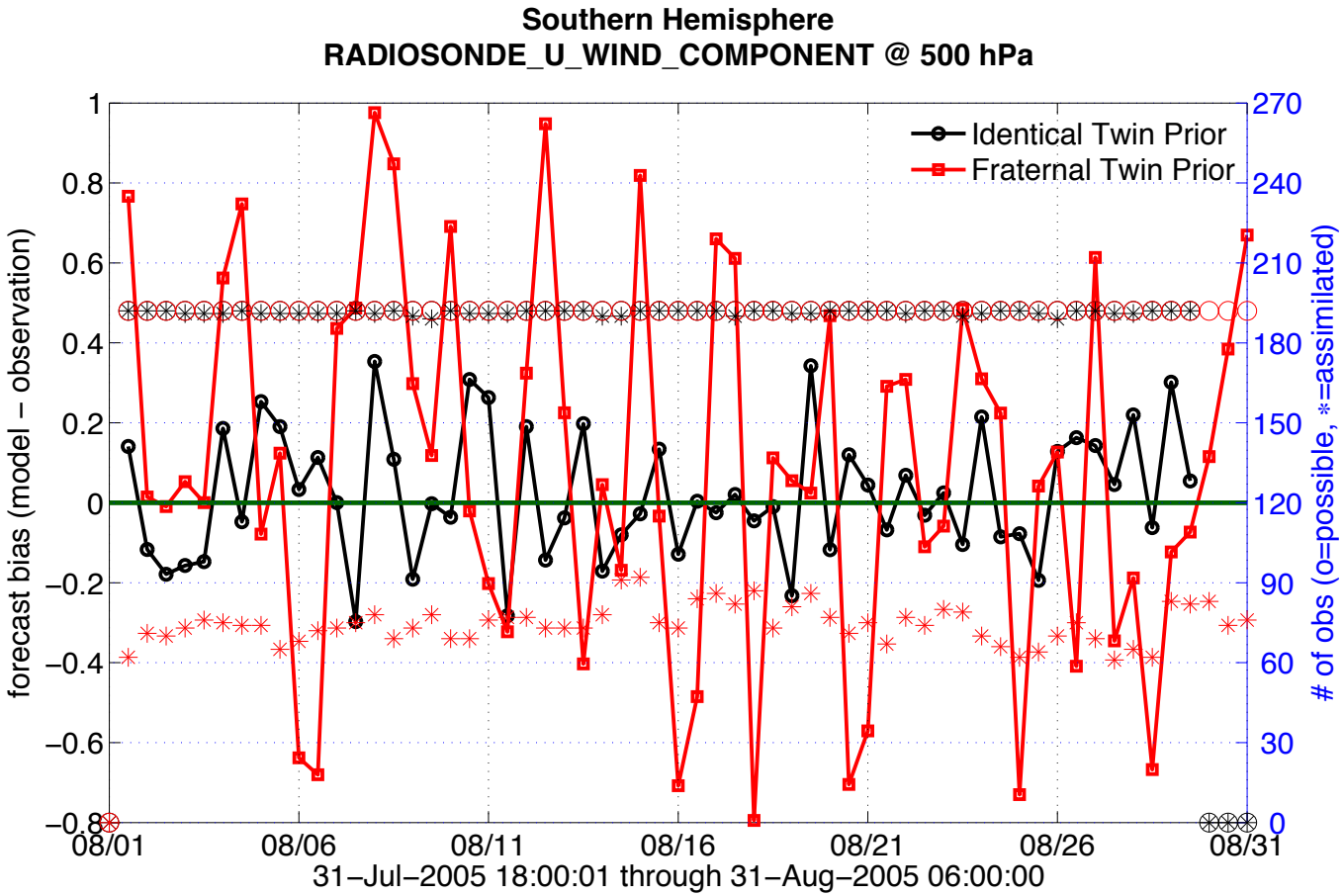
Observation-space diagnostics: comparing experiments

two_experiments_evolution.m

two_experiments_profile.m

This is useful for quick comparisons. Really 'fair' comparisons require more processing to compare the same set of observations across experiments.

assimilation_code/programs
obs_common_subset.html
obs_seq_coverage.html
obs_selection.html
obs_seq_verify.html



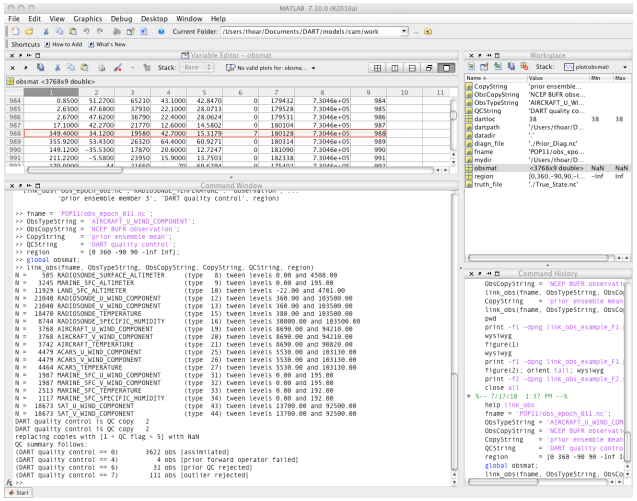
FYI:
'Identical' means the model that was used to generate the observations is also used for the assimilation.
'Fraternal' means the observations came from a different model.

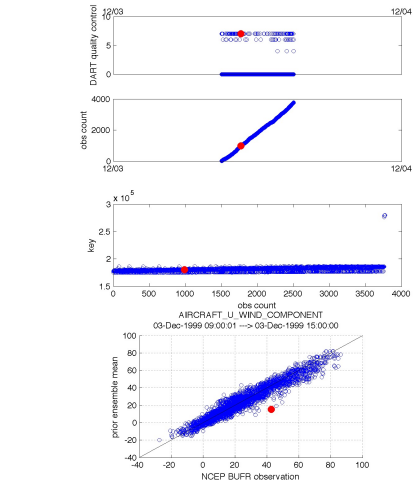
Observation-space diagnostics: netCDF

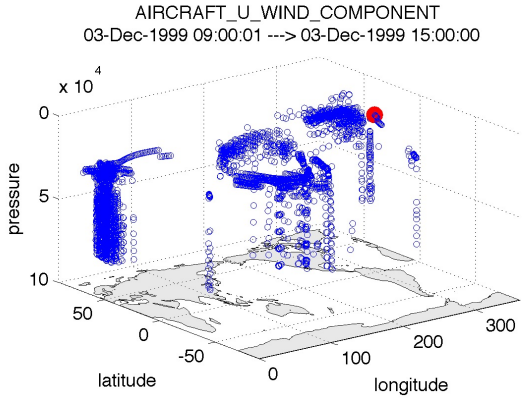
SOME of the information in the observation sequence files can be converted to netCDF and easily plotted. A program named ***obs_seq_to_netcdf*** must be run to produce the netCDF.

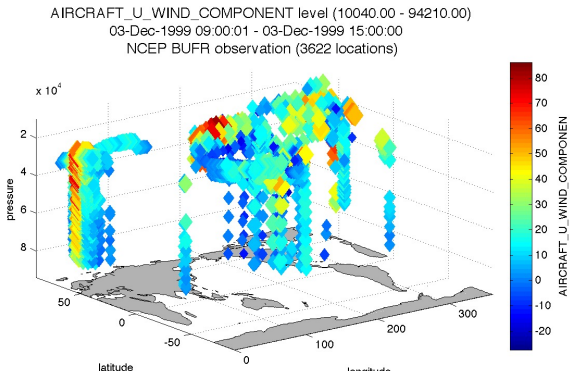
Here are a few of the Matlab functions available in *diagnostics/matlab*.

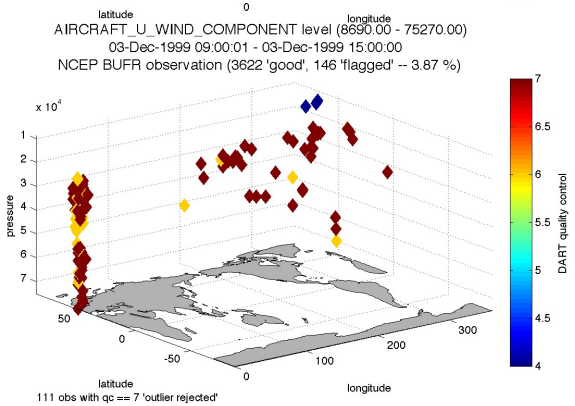
- **link_obs.m**
- **plot_obs_netcdf.m**
- **plot_obs_netcdf_diffs.m**
- **plot_coverage.m**











Complicated observation-space diagnostics.

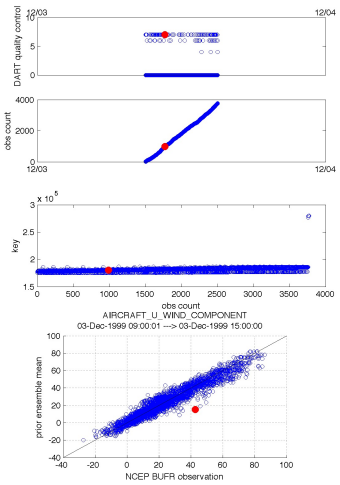
The program ***obs_seq_to_netcdf*** converts much of the information in an observation sequence file to a netCDF file. For now, we're going to explore a pre-computed file available at:

www.image.ucar.edu/pub/DART/Tutorial_Datasets/obs_epoch_SE30r4_Katrina.nc

It was generated with the following input:

```
&schedule_nml
    calendar           = 'Gregorian'
    first_bin_start    = 2005, 8, 13, 21, 0, 0
    first_bin_end      = 2005, 8, 14, 03, 0, 0
    last_bin_end       = 2005, 8, 14, 03, 0, 0
    bin_interval_days    = 10000
    bin_interval_seconds = 0
    max_num_bins         = 1000
    print_table          = .true.
  /
&obs_seq_to_netcdf_nml
    obs_sequence_name = 'cam_obs_seq.2005-08-14-00000.final'
    obs_sequence_list = ''
     lonlim1 = 160.
     lonlim2 = 40.
     latlim1 = 10.
     latlim2 = 65.
  /
```

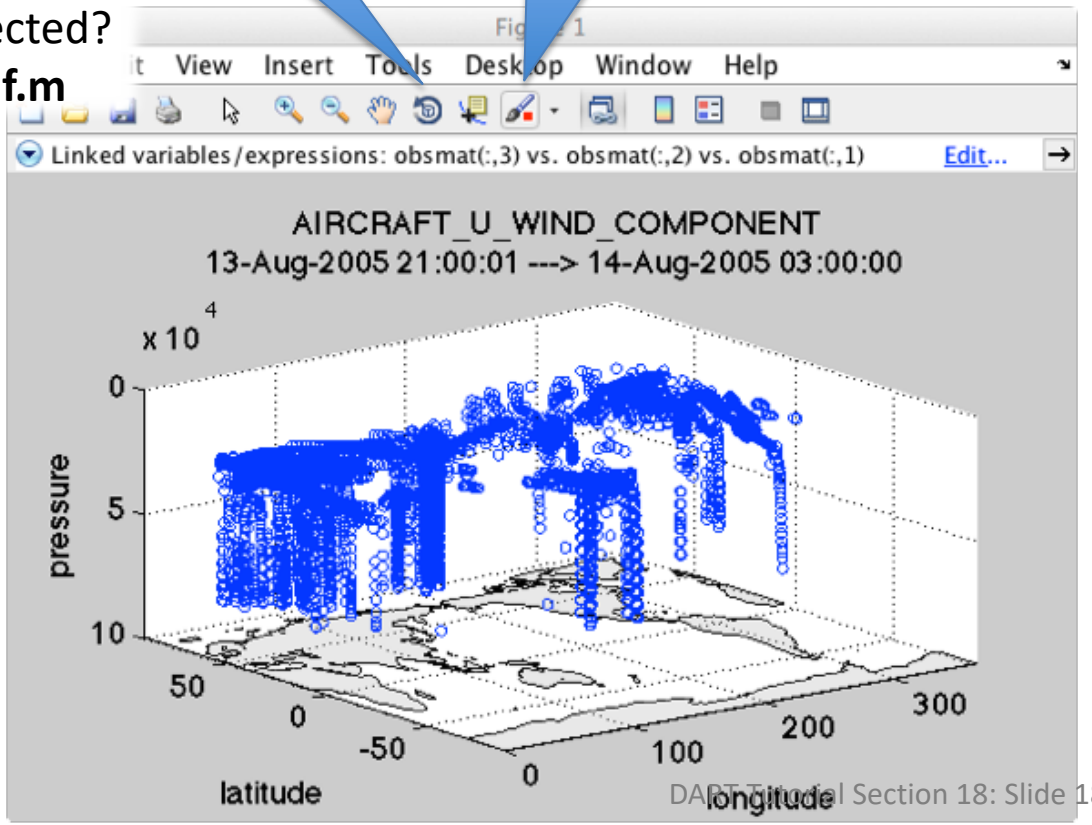
Matlab Hands-On: **link_obs** exploring observations

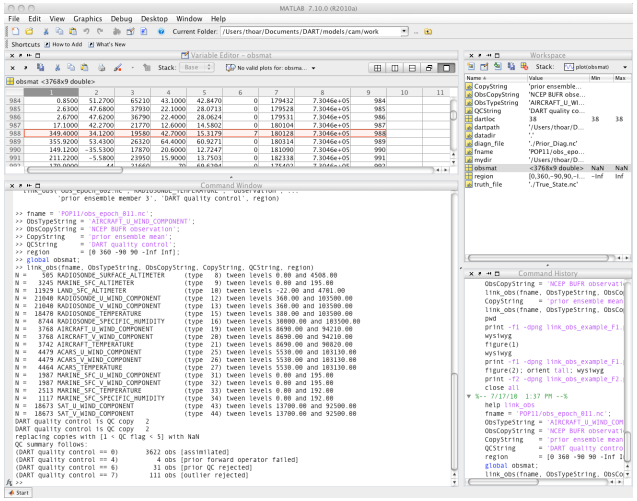


This enables rotation with the mouse.

'paintbrush' allows you to select observations for 'brushing'

Try different obs types,
Try to locate rejected obs,
Why were they rejected?
Try **plot_obs_netcdf.m**





DART Tutorial Index to Sections