# Background Fit to Satellite Observations

William F. Campbell
U.S. Naval Research Laboratory, Monterey, CA

## Background and Methodology

### What is it?

- Background fit to observations is a measure of how close, on average, a short forecast (background) is to the entire set of observations valid at the forecast time
- The measure of closeness typically chosen is the **innovation** (observation minus background) **standard deviation**
- Unlike most verification metrics, it is computed in **observation space** rather than model space
- Observations used to measure the background fit ***need not be assimilated***, but the observation operator must be computed

### How to Compute

- For each observation (e.g. NPP ATMS channel 4) and for each assimilation window, ***measure*** how well the ***background fit***s all of the observations by computing the ***innovation standard deviation***
- Form the **ratio** of innovation standard deviations of the **experiment** to the **control**
- If the **experiment** fits the background **better** than the control, the **ratio** will be **less than 1**; if the **control** fits **better,** the **ratio** will be **greater than 1**
- Form a **time series of** these **ratios**, and evaluate whether the mean is **statistically significant**ly different from **1** (Student's t-test on the log of the time series with lag-1 autocorrelation correction)
- For that particular observation and time period:
  - If the confidence interval around the mean **includes 1**, the null hypothesis that **experiment** and **control** are indistinguishable cannot be rejected; else
  - If the mean is **less than 1**, the **experiment** is **better** than the **control**
  - If the mean is **greater than 1**, the **control** is **better** than the **experiment**

### Verification

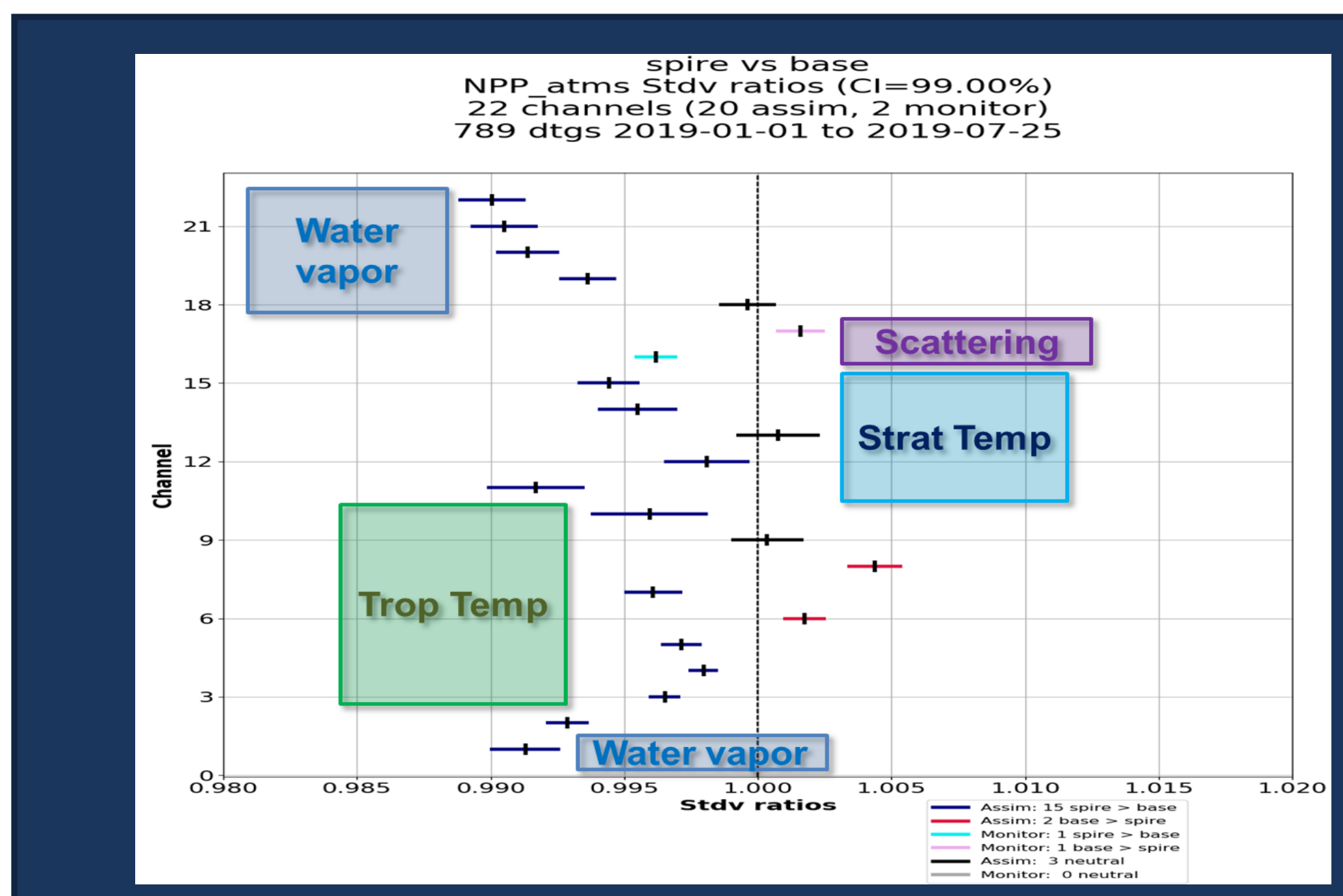
- ECMWF uses background fit to observations as their ***primary metric*** in evaluating short-range (up to 5 days) forecasts*
- Decisions on the improvement or degradation of numerical experiments (relative to a control run) can be made ***much faster*** and (potentially) ***more reliably*** than with traditional forecast metrics
- ***Any type of change*** between experiment and control can be evaluated using this method: dynamics, numerics, observations, error covariances, physics, data assimilation, …
- The savings in ***computer time*** and, more importantly, ***researcher time*** are substantial

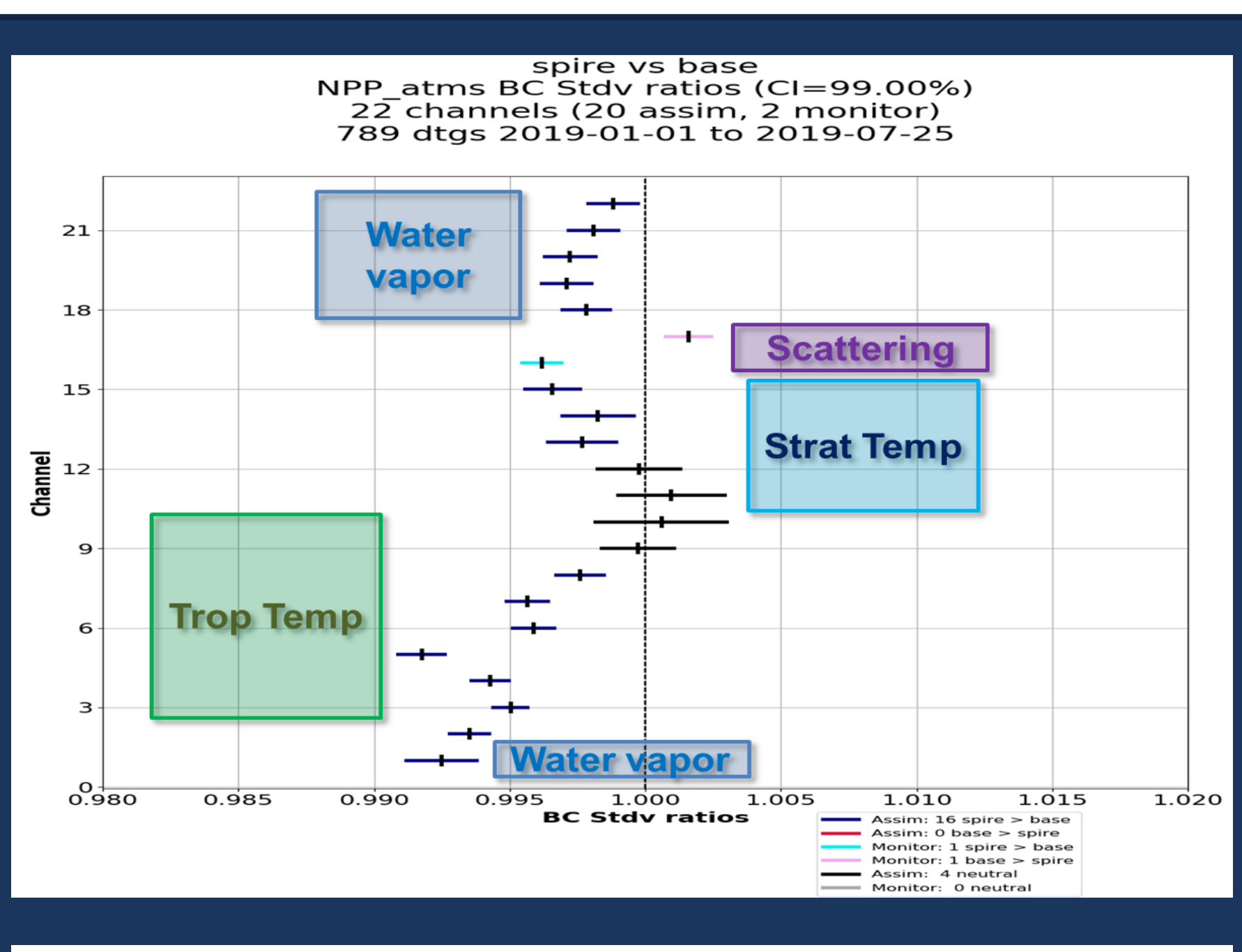
*Alan Geer, personal communication

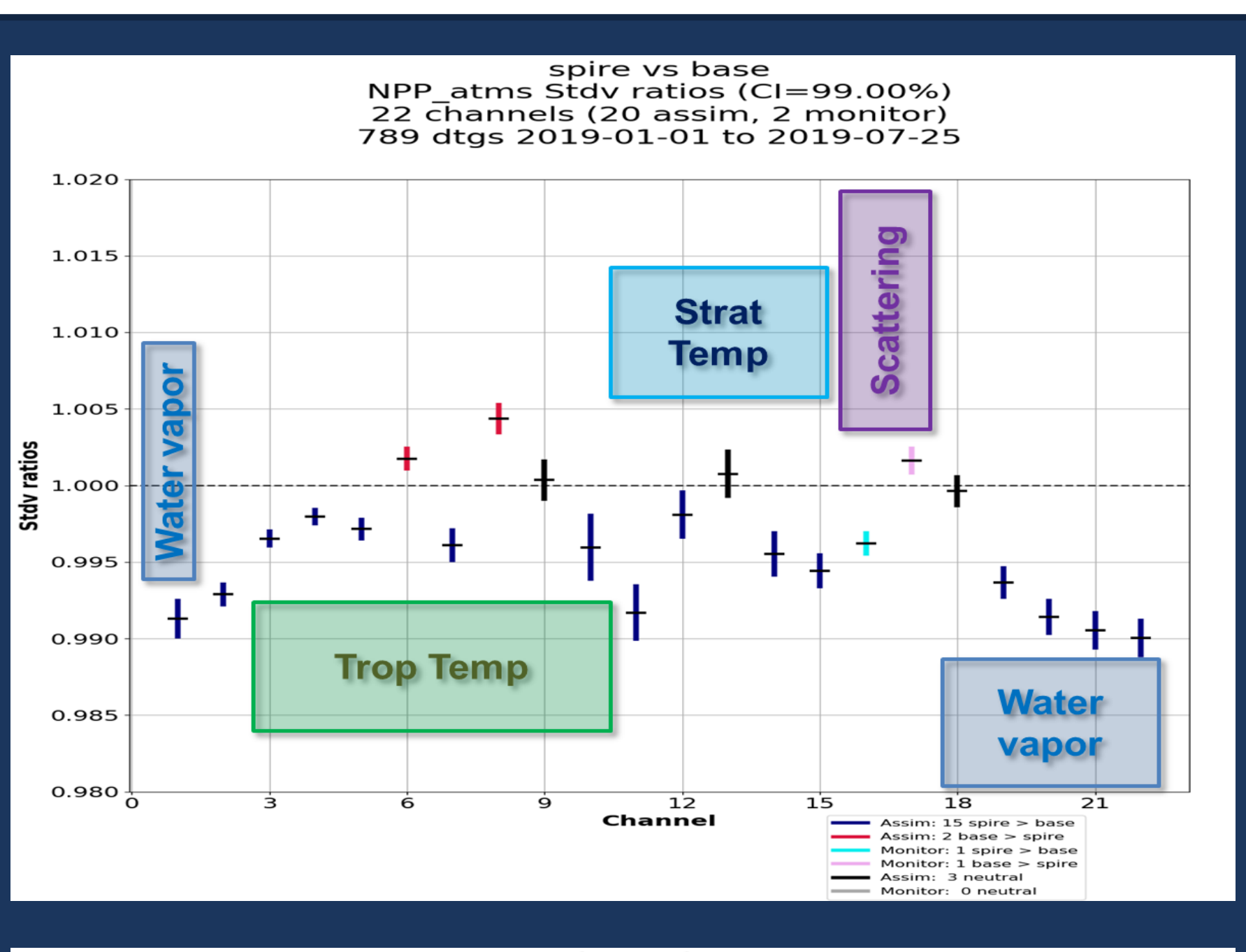
### Diagnostic

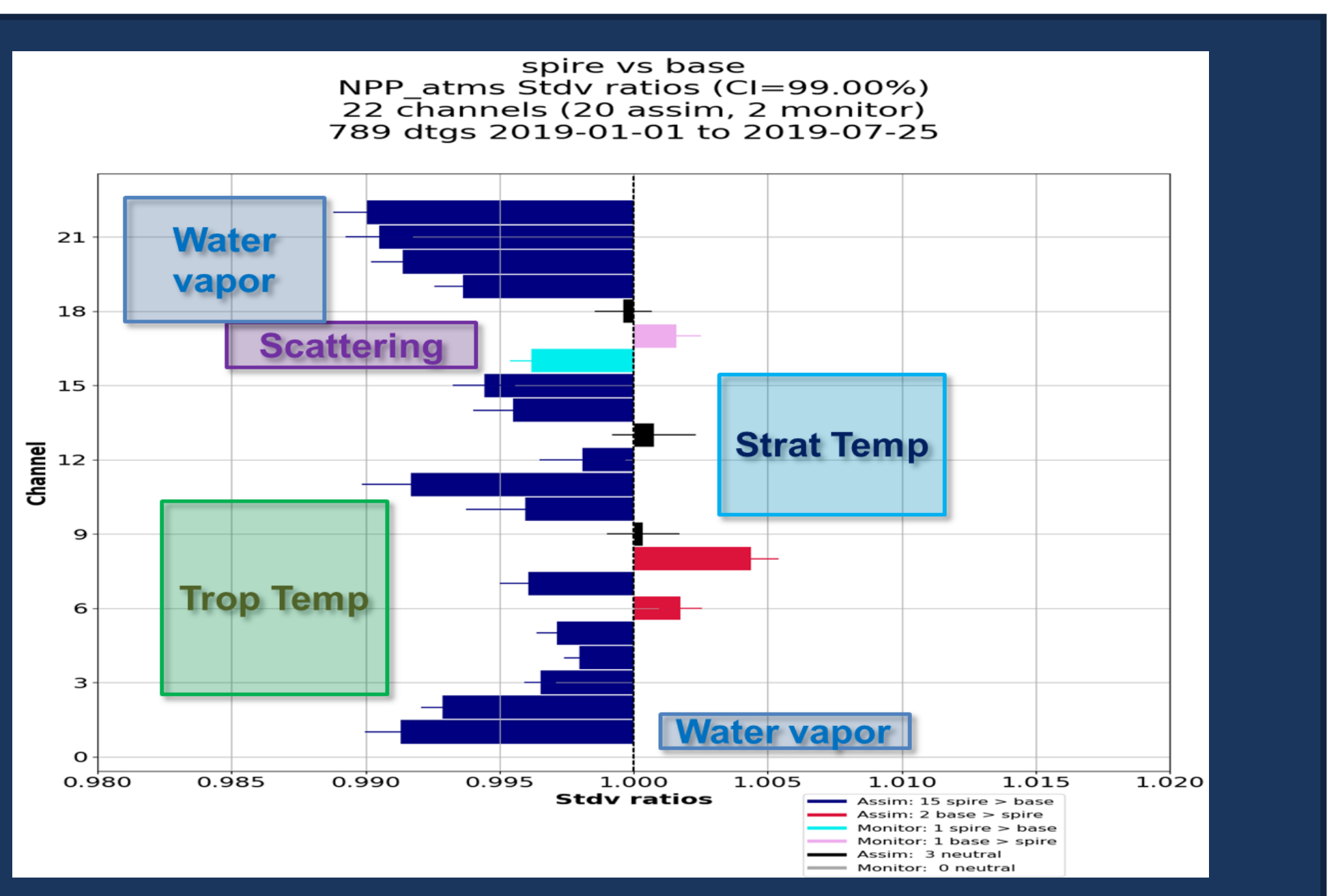
- Fit to observations is also a ***diagnostic tool***
- Millions of monitored and assimilated **observations** can be **used an additional time** to evaluate the prior forecast in observation space
- We can **aggregate similar observations** in plots and scorecards to facilitate **causal inferences**
  - For example, an experiment improves the background fit to radiances sensitive to moisture in the boundary layer, but degrades the background fit to radiances sensitive to stratospheric temperature
  - Diagnostic information like this provides guidance on how to resolve issues, resulting in an acceleration of the software development cycle

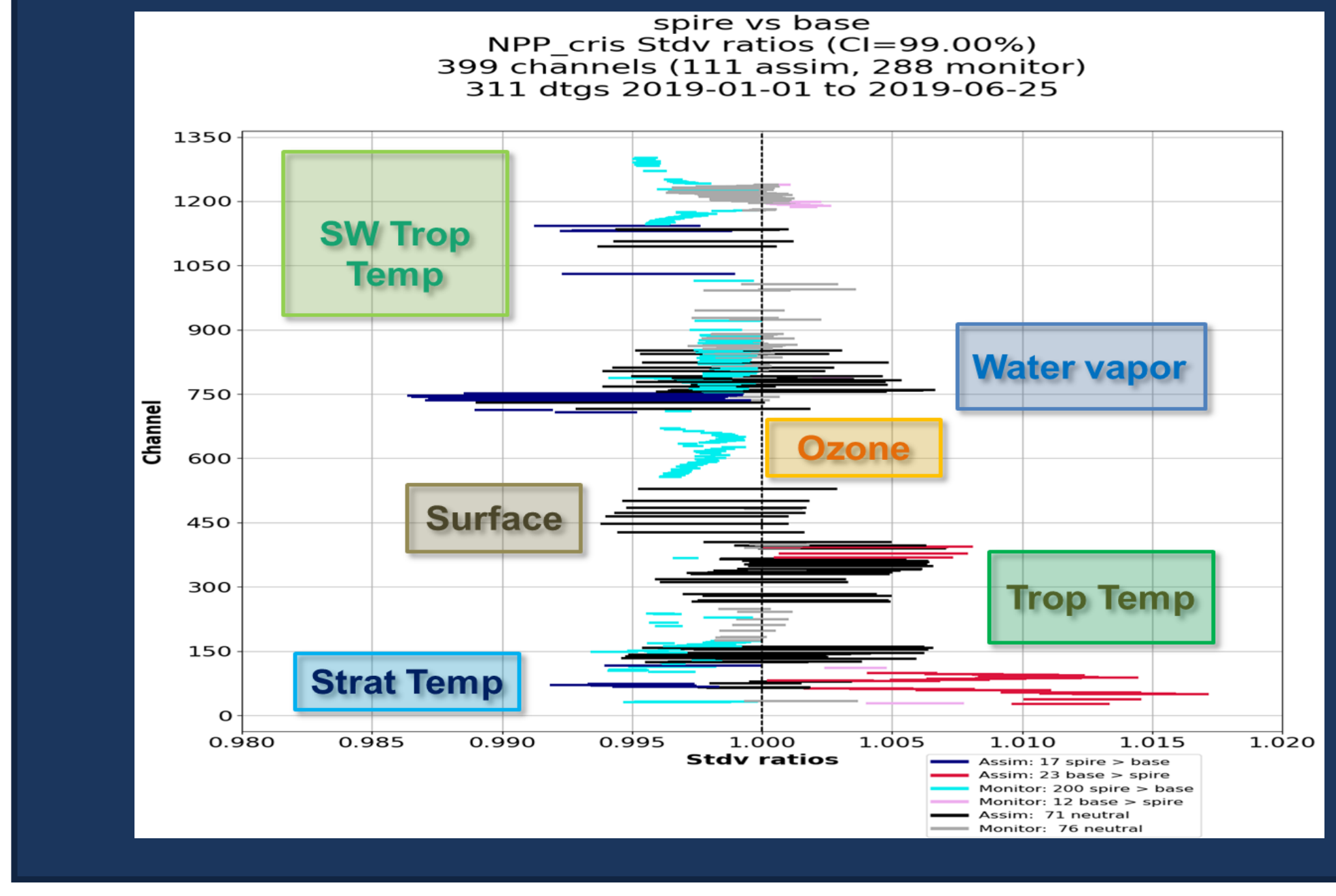
## Graphics Gallery

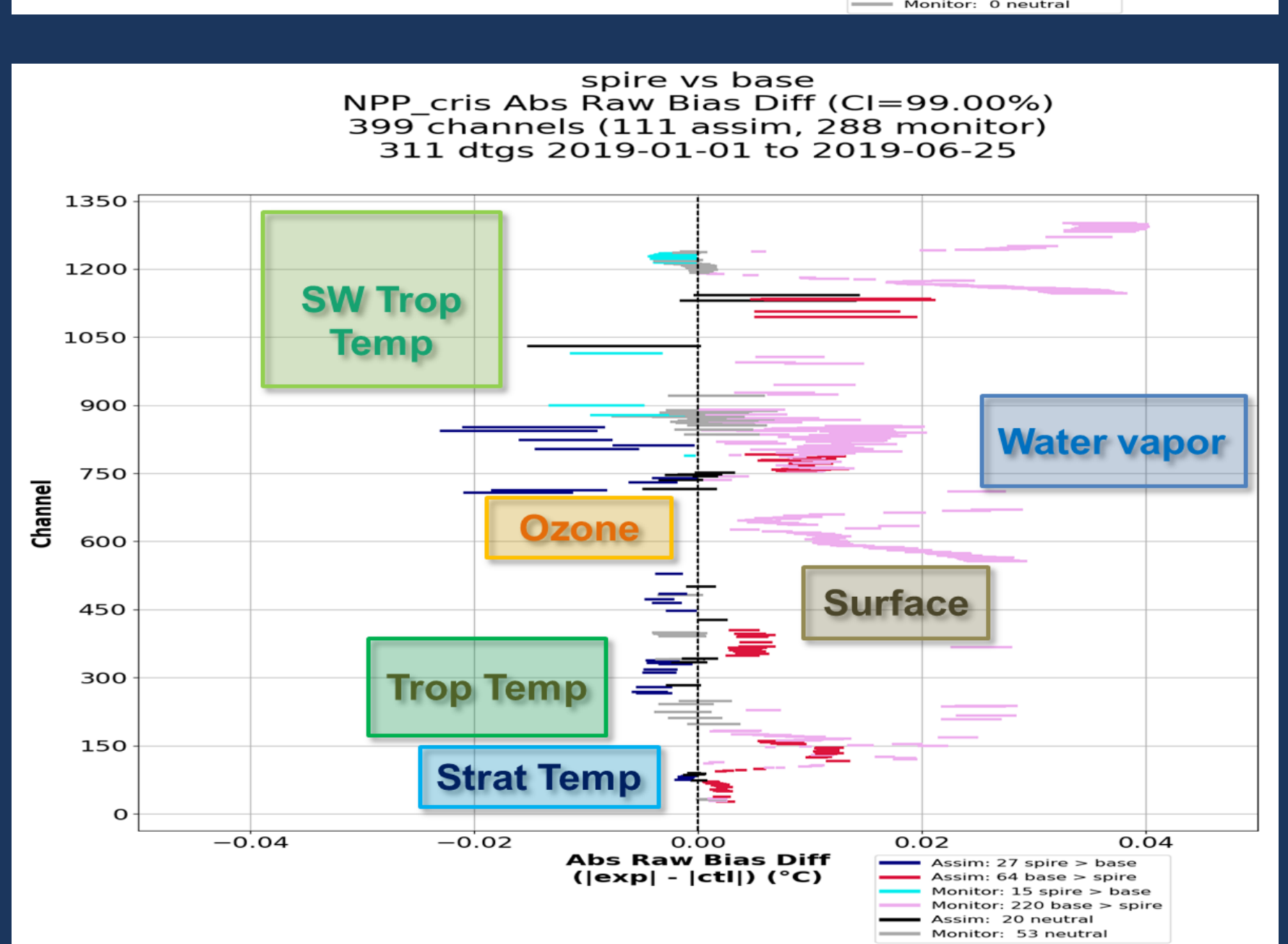


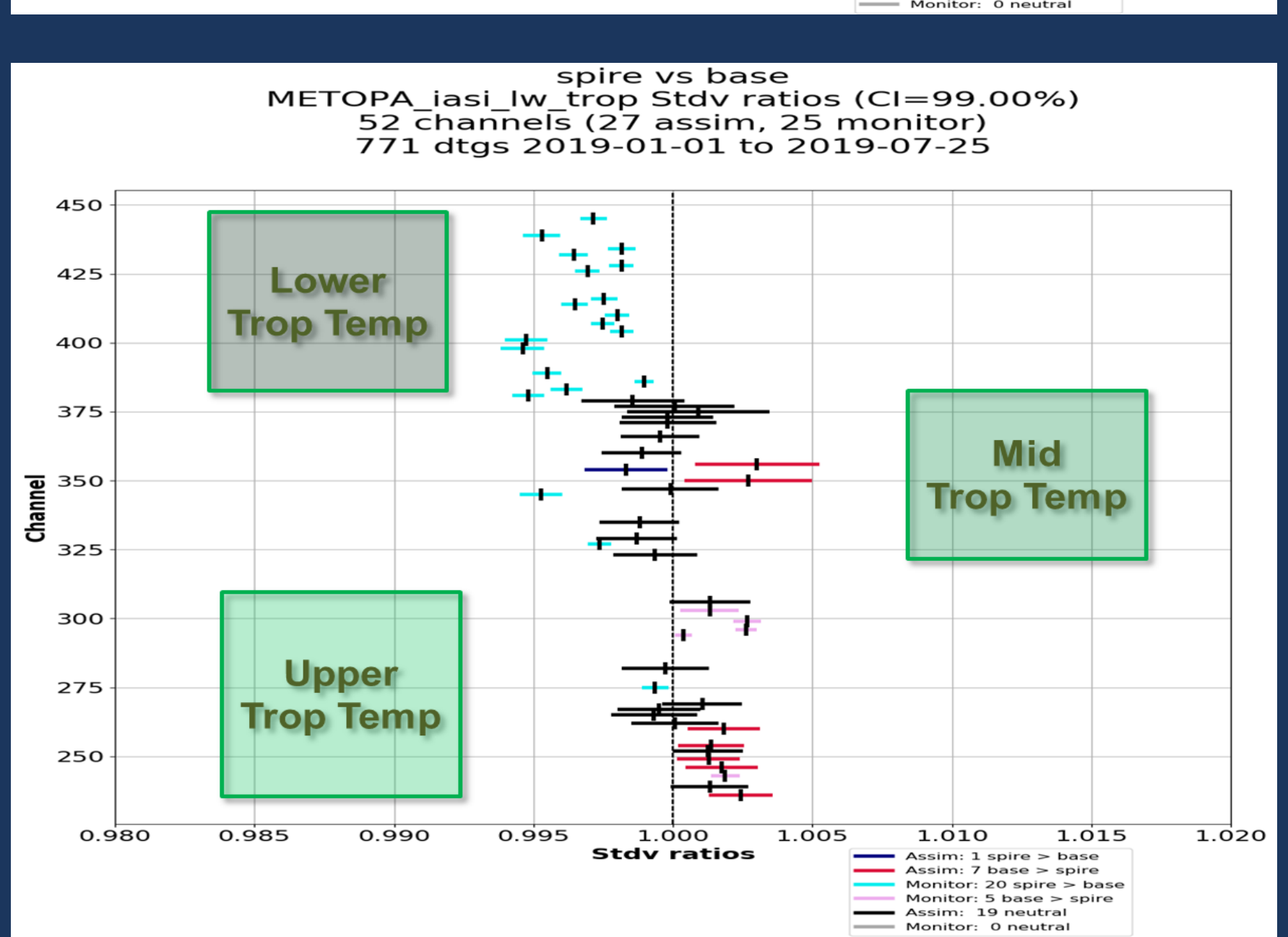


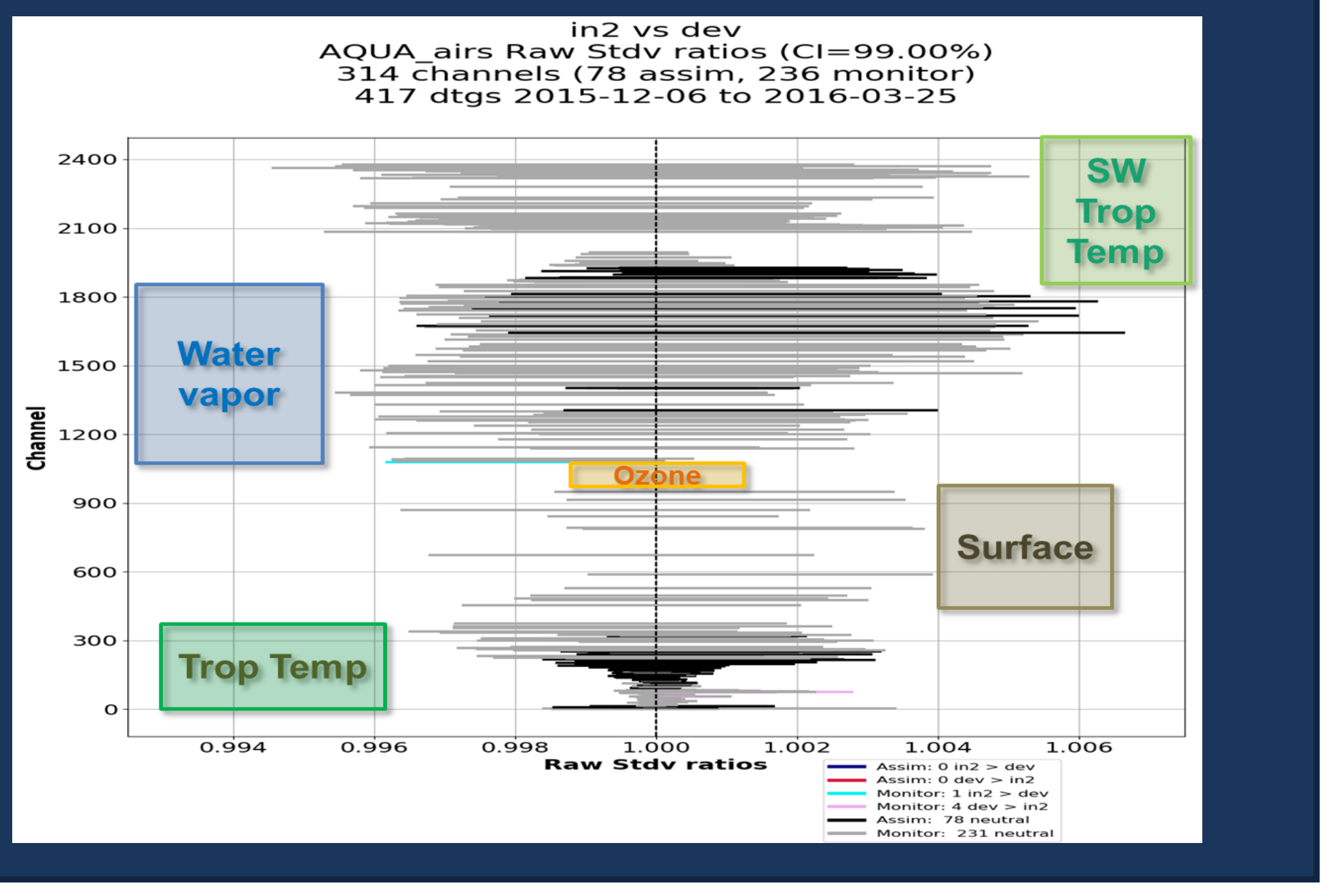






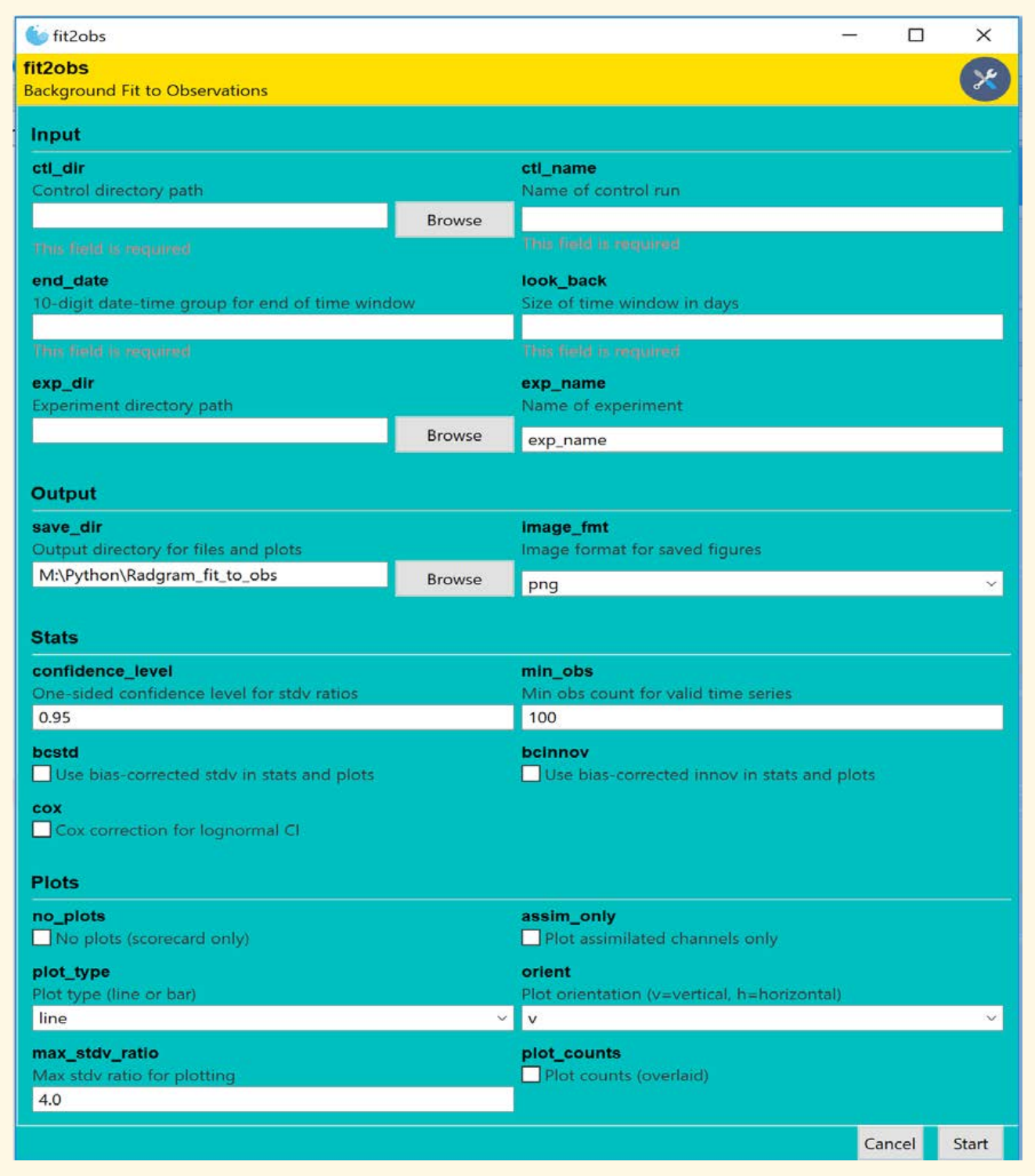






## Input and Output

### GUI



fit2obs
Background Fit to Observations

Input

ctl_dir
Control directory path
This field is required

ctl_name
Name of control run
This field is required

end_date
10-digit date-time group for end of time window
This field is required

look_back
Size of time window in days
This field is required

exp_dir
Experiment directory path

exp_name
Name of experiment
exp_name

Output

save_dir
Output directory for files and plots
M:\Python\Radgram_fit_to_obs

image_fmt
Image format for saved figures
png

Stats

confidence_level
One-sided confidence level for stdv ratios
0.95

min_obs
Min obs count for valid time series
100

bcstd
□ Use bias-corrected stdv in stats and plots

bcinnov
□ Use bias-corrected innov in stats and plots

cox
□ Cox correction for lognormal CI

Plots

no_plots
□ No plots (scorecard only)

assim_only
□ Plot assimilated channels only

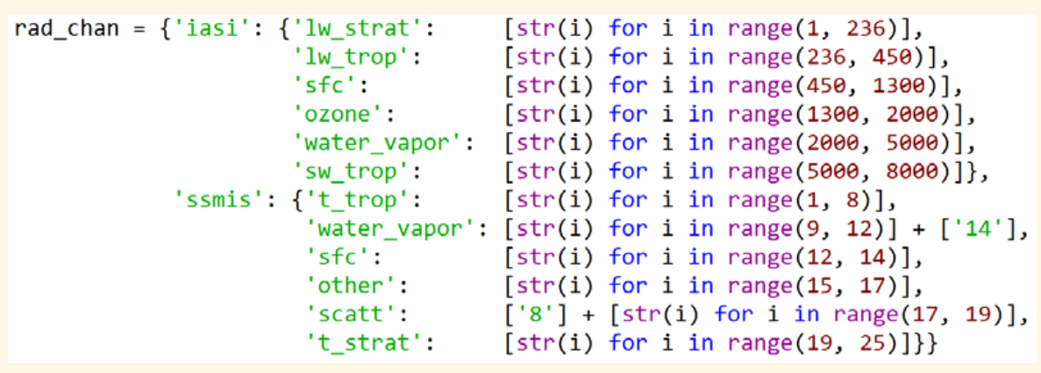
plot_type
Plot type (line or bar)
line

orient
Plot orientation (v=vertical, h=horizontal)
v

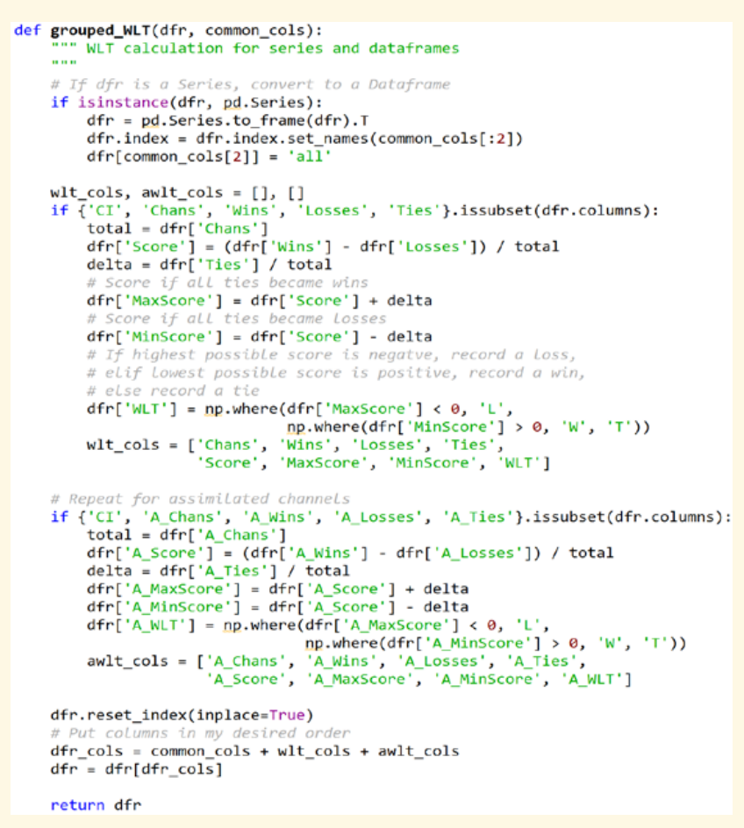
max_stdv_ratio
Max stdv ratio for plotting
4.0

plot_counts
□ Plot counts (overlaid)

### Python Snippets



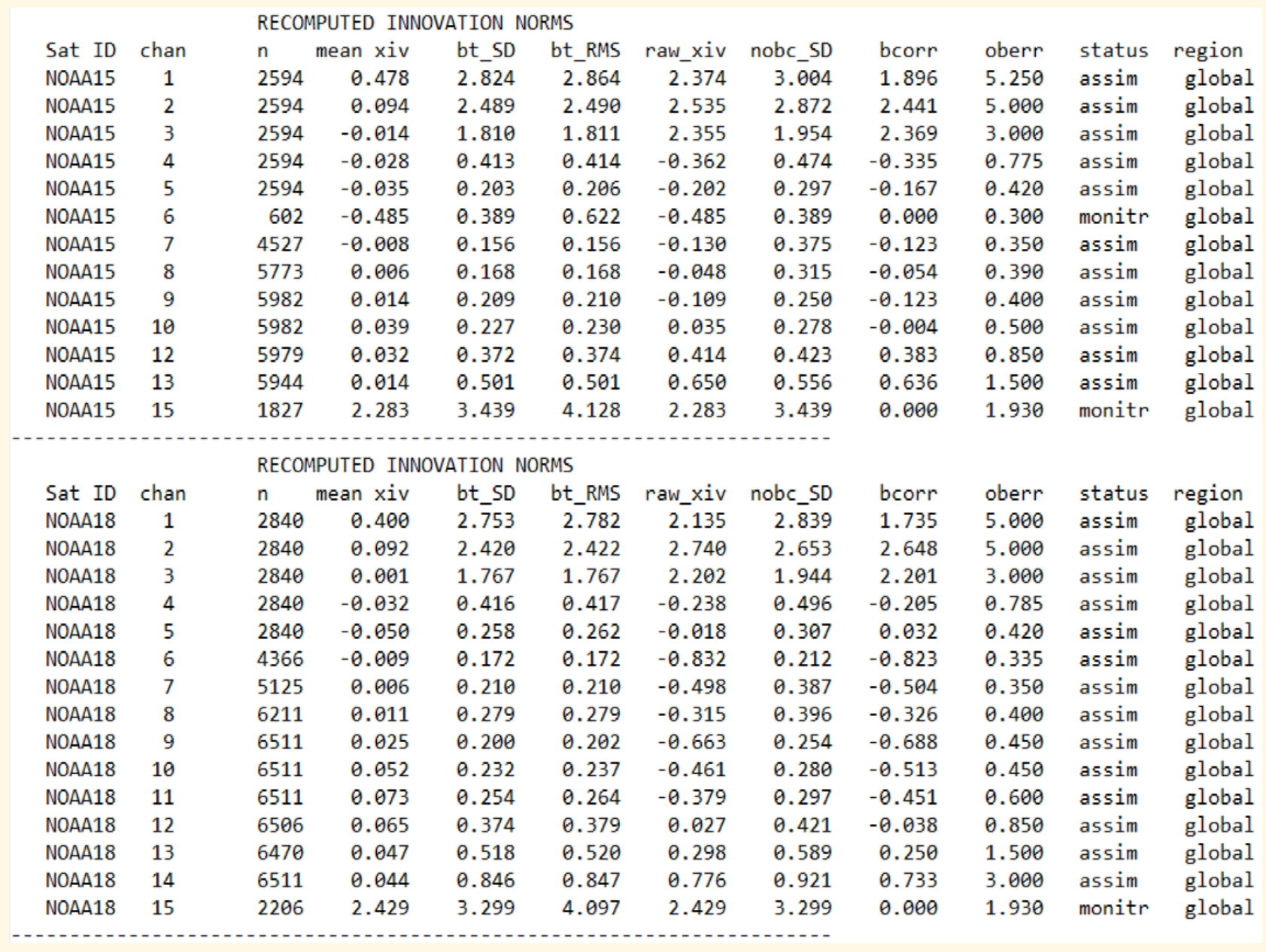
```
rad_chan = {'iasi': {'lw_strat':     [str(i) for i in range(1, 236)],
                     'lw_trop':      [str(i) for i in range(236, 450)],
                     'sfc':          [str(i) for i in range(450, 1300)],
                     'ozone':        [str(i) for i in range(1300, 2000)],
                     'water_vapor':  [str(i) for i in range(2000, 5000)],
                     'sw_trop':      [str(i) for i in range(5000, 8000)]},
            'ssmis': {'t_trop':      [str(i) for i in range(1, 8)],
                      'water_vapor': [str(i) for i in range(9, 12)] + ['14'],
                      'sfc':         [str(i) for i in range(12, 14)],
                      'other':       [str(i) for i in range(15, 17)],
                      'scatt':       ['8'] + [str(i) for i in range(17, 19)],
                      't_strat':     [str(i) for i in range(19, 25)]}}
```



```
def grouped_WLT(dfr, common_cols):
    """ WLT calculation for series and dataframes
    """
    # If dfr is a Series, convert to a Dataframe
    if isinstance(dfr, pd.Series):
        dfr = pd.Series.to_frame(dfr).T
        dfr.index = dfr.index.set_names(common_cols[:2])
        dfr[common_cols[2]] = 'all'

    wlt_cols, awlt_cols = [], []
    if {'CI', 'Chans', 'Wins', 'Losses', 'Ties'}.issubset(dfr.columns):
        total = dfr['Chans']
        dfr['Score'] = (dfr['Wins'] - dfr['Losses']) / total
        delta = dfr['Ties'] / total
        # Score if all ties became wins
        dfr['MaxScore'] = dfr['Score'] + delta
        # Score if all ties became losses
        dfr['MinScore'] = dfr['Score'] - delta
        # If highest possible score is negatve, record a loss,
        # elif lowest possible score is positive, record a win,
        # else record a tie
        dfr['WLT'] = np.where(dfr['MaxScore'] < 0, 'L',
                              np.where(dfr['MinScore'] > 0, 'W', 'T'))
        wlt_cols = ['Chans', 'Wins', 'Losses', 'Ties',
                    'Score', 'MaxScore', 'MinScore', 'WLT']

    # Repeat for assimilated channels
    if {'CI', 'A_Chans', 'A_Wins', 'A_Losses', 'A_Ties'}.issubset(dfr.columns):
        total = dfr['A_Chans']
        dfr['A_Score'] = (dfr['A_Wins'] - dfr['A_Losses']) / total
        delta = dfr['A_Ties'] / total
        dfr['A_MaxScore'] = dfr['A_Score'] + delta
        dfr['A_MinScore'] = dfr['A_Score'] - delta
        dfr['A_WLT'] = np.where(dfr['A_MaxScore'] < 0, 'L',
                                np.where(dfr['A_MinScore'] > 0, 'W', 'T'))
        awlt_cols = ['A_Chans', 'A_Wins', 'A_Losses', 'A_Ties',
                     'A_Score', 'A_MaxScore', 'A_MinScore', 'A_WLT']

    dfr.reset_index(inplace=True)
    # Put columns in my desired order
    dfr_cols = common_cols + wlt_cols + awlt_cols
    dfr = dfr[dfr_cols]

    return dfr
```

- The input for fit2obs.py consists of a set of ASCII files with the naming convention as follows: {instrument}_fit2obs_{date}.{region}
- In each file are the counts and  raw and bias-corrected innovation statistics for each satellite and channel of that instrument for that date.
- The user must compute these statistics and produce a set of input files in a directory structure that looks like {expname}/g{date}
- In each folder will be files for all of the instruments and all of the regions for that date
- Fit2obs.py is called separately for each named region

### amsua_fit2obs_2019041900.global



```
                       RECOMPUTED INNOVATION NORMS
  Sat ID   chan        n    mean xiv    bt_SD   bt_RMS  raw_xiv  nobc_SD    bcorr    oberr   status   region
  NOAA15    1       2594     0.478     2.824    2.864    2.374    3.004    1.896    5.250    assim    global
  NOAA15    2       2594     0.094     2.489    2.490    2.535    2.872    2.441    5.000    assim    global
  NOAA15    3       2594    -0.014     1.810    1.811    2.355    1.954    2.369    3.000    assim    global
  NOAA15    4       2594    -0.028     0.413    0.414   -0.362    0.474   -0.335    0.775    assim    global
  NOAA15    5       2594    -0.035     0.203    0.206   -0.202    0.297   -0.167    0.420    assim    global
  NOAA15    6        602    -0.485     0.389    0.622   -0.485    0.389    0.000    0.300    monitr   global
  NOAA15    7       4527    -0.008     0.156    0.156   -0.130    0.375   -0.123    0.350    assim    global
  NOAA15    8       5773     0.006     0.168    0.168   -0.048    0.315   -0.054    0.390    assim    global
  NOAA15    9       5982     0.014     0.209    0.210   -0.109    0.250   -0.123    0.400    assim    global
  NOAA15   10       5982     0.039     0.227    0.230    0.035    0.278   -0.004    0.500    assim    global
  NOAA15   12       5979     0.032     0.372    0.374    0.414    0.423    0.383    0.850    assim    global
  NOAA15   13       5944     0.014     0.501    0.501    0.650    0.556    0.636    1.500    assim    global
  NOAA15   15       1827     2.283     3.439    4.128    2.283    3.439    0.000    1.930    monitr   global
----------------------------------------------------------------------------
                       RECOMPUTED INNOVATION NORMS
  Sat ID   chan        n    mean xiv    bt_SD   bt_RMS  raw_xiv  nobc_SD    bcorr    oberr   status   region
  NOAA18    1       2840     0.400     2.753    2.782    2.135    2.839    1.735    5.000    assim    global
  NOAA18    2       2840     0.092     2.420    2.422    2.740    2.653    2.648    5.000    assim    global
  NOAA18    3       2840     0.001     1.767    1.767    2.202    1.944    2.201    3.000    assim    global
  NOAA18    4       2840    -0.032     0.416    0.417   -0.238    0.496   -0.205    0.785    assim    global
  NOAA18    5       2840    -0.050     0.258    0.262   -0.018    0.307    0.032    0.420    assim    global
  NOAA18    6       4366    -0.009     0.172    0.172   -0.832    0.212   -0.823    0.335    assim    global
  NOAA18    7       5125     0.006     0.210    0.210   -0.498    0.387   -0.504    0.350    assim    global
  NOAA18    8       6211     0.011     0.279    0.279   -0.315    0.396   -0.326    0.400    assim    global
  NOAA18    9       6511     0.025     0.200    0.202   -0.663    0.254   -0.688    0.450    assim    global
  NOAA18   10       6511     0.052     0.232    0.237   -0.461    0.280   -0.513    0.450    assim    global
  NOAA18   11       6511     0.073     0.254    0.264   -0.379    0.297   -0.451    0.600    assim    global
  NOAA18   12       6506     0.065     0.374    0.379    0.027    0.421   -0.038    0.850    assim    global
  NOAA18   13       6470     0.047     0.518    0.520    0.298    0.589    0.250    1.500    assim    global
  NOAA18   14       6511     0.044     0.846    0.847    0.776    0.921    0.733    3.000    assim    global
  NOAA18   15       2206     2.429     3.299    4.097    2.429    3.299    0.000    1.930    monitr   global
----------------------------------------------------------------------------
```

### Detailed Description

By comparing raw innovations to 6-hour forecasts and computing the spatial standard deviation four times a day, we can construct a time series for the control run and each experiment, for every radiance observation type and channel. Each observation type and channel, including unassimilated channels that were monitored, will show either statistical improvement or degradation, or be neutral. The aggregation of these results are in accord with traditional forecast diagnostics that require multiple months of simulated global NWP; however, the fit to observations diagnostic requires only a few weeks of simulation. In addition, it has also been shown to be more sensitive than traditional diagnostics. Up until now, we have lacked this vital tool.

The tool consists of six routines written in Python 3. The main program is in fit2obs.py, and is where the bulk of the data handling is done. Fit2obs.py reads in the radiance statistics from either the *_ar_1_* files or the *_fit2obs_* files for a set of date-time groups for two runs, an experiment and a control. For each sensor and channel, a paired difference t-test is run on the log-transformed raw standard deviations, accounting for serial correlation in time. (Statistics routines are in fit2obs_stats.py). This information is passed to fit2obs_plots.py, which produces a set of graphics that are part of the final output. Fit2obs_scorecard.py also produces a scorecard (described below). Input arguments are handled by fit2obs_arguments.py, which includes an optional gui interface (requires the gooey package).

Instrument_info.py defines two dictionaries. The first, sat_sen, is a copy of sat_sen from radiometer_define.py, with dummy entries for CrIS, AIRS, and IASI added for convenience. The second, rad_chan, consists of channel subsets corresponding to the species detected (T, q, ozone) and the approximate vertical location in the atmosphere. The channel subsets are used by fit2obs_plots.py to make plots containing their particular subset of channels for each instrument, and to aggregate statistics into a pandas dataframe.

Fit2obs_plots.py takes a dictionary created by fit2obs.py, either a comparison of an experiment and a control, which is its primary use, or just the control. Graphics options include plot type (bar or line), orientation (vertical or horizontal), image format (png or pdf). Plots can be generated for all channels, or only for assimilated channels. The confidence level defaults to 95%, but can be specified on the command line. When given an experiment and a control dictionary and graphical options, it produces a set of images containing the ratios of raw innovation standard deviations with confidence intervals for each channel. If the confidence interval does not include 1.0, then it is plotted in red (magenta) when the mean ratio is greater than 1.0, indicating that the fit to the assimilated (monitored) channel is better in the experiment than the control. If the mean ratio is less than 1.0 and the confidence interval does not included 1.0, the plot is navy (cyan) for assimilated (monitored) channels. Confidence intervals that include 1.0 are plotted in black (gray) for assimilated (monitored) channels, indicating that the experiment and control fits to this channel are not statistically significant. Fit2obs_plots also generates a set of images containing the mean absolute bias differences with confidence intervals and the same color scheme as the standard deviation ratio plots. Options for plotting bias-corrected standard deviations and absolute differences are also available.

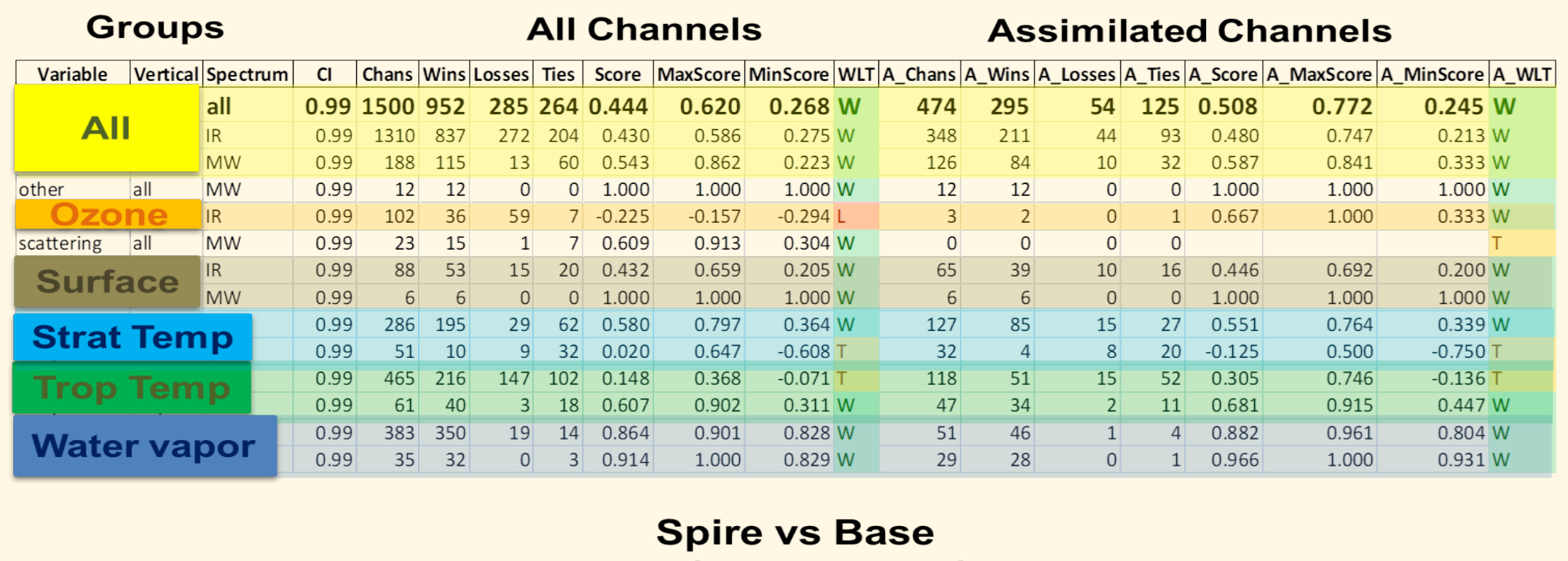
Fit2obs_scorecard.py produces an Excel spreadsheet with summary statistics of wins, losses, and ties for all channels, and for common subsets (e.g. microwave temperature channels in the stratosphere for all instruments). For groups of channels, we chose a conservative criterion for declaring a group win (win = experiment fits the observations in the group better than the control): If number of individual channel wins exceeds the sum of losses and ties, we declare a group win; if number of individual losses exceed the sum of wins and ties, we declare a group loss; otherwise a group tie.

## Scorecard

- A **scorecard** can provide **summary guidance** as to whether enough evidence has been gathered to favor the experiment over the control or v.v., or whether not enough data has yet been evaluated to decide between them
- As the **time series** of ratios becomes **longer**, **confidence intervals** get **shorter**, and observations that were neutral will **eventually*** show either a **win** or a **loss**
- We can do this for individual observations, physically sensible groups of observations, and all observations aggregated together
- For groups of observations, we chose a conservative criterion for declaring a **group win**: If number of individual wins exceeds the sum of losses and ties, we declare a **group win**; if number of individual losses exceed the sum of wins and ties, we declare a **group loss**; otherwise a **group tie**

* Assumes statistical stationarity, which may not be true. Some side effects may occur. Consult your doctor(ate) before using the scorecard.

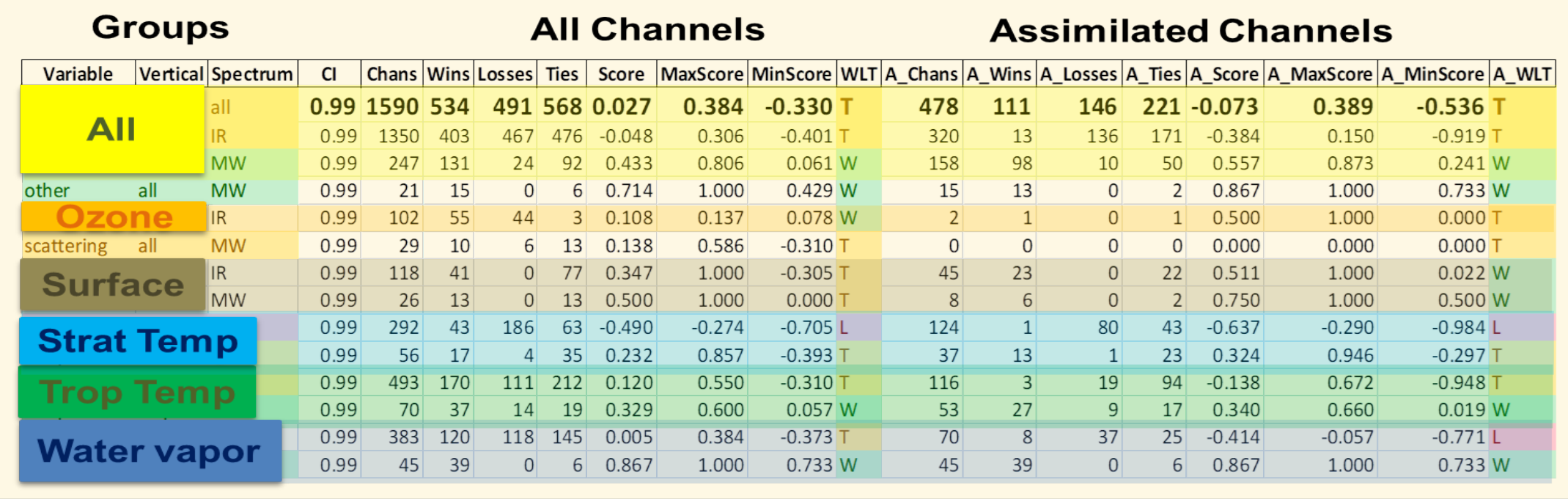
**Cpl_jac_pb0 vs Cntrl**
**Jun 1 – Aug 31, 2016**



Groups | All Channels | Assimilated Channels

| Variable | Vertical | Spectrum | CI | Chans | Wins | Losses | Ties | Score | MaxScore | MinScore | WLT | A_Chans | A_Wins | A_Losses | A_Ties | A_Score | A_MaxScore | A_MinScore | A_WLT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All | | all | 0.99 | 1500 | 952 | 285 | 264 | 0.444 | 0.620 | 0.268 | W | 474 | 295 | 54 | 125 | 0.508 | 0.772 | 0.245 | W |
| | | IR | 0.99 | 1310 | 837 | 272 | 204 | 0.430 | 0.586 | 0.275 | W | 348 | 211 | 44 | 93 | 0.480 | 0.747 | 0.213 | W |
| | | MW | 0.99 | 188 | 115 | 13 | 60 | 0.543 | 0.862 | 0.223 | W | 126 | 84 | 10 | 32 | 0.587 | 0.841 | 0.333 | W |
| other | all | MW | 0.99 | 12 | 12 | 0 | 0 | 1.000 | 1.000 | 1.000 | W | 12 | 12 | 0 | 0 | 1.000 | 1.000 | 1.000 | W |
| Ozone | | IR | 0.99 | 102 | 36 | 59 | 7 | -0.225 | -0.157 | -0.294 | L | 3 | 2 | 0 | 1 | 0.667 | 1.000 | 0.333 | W |
| scattering | all | MW | 0.99 | 23 | 15 | 1 | 7 | 0.609 | 0.913 | 0.304 | W | 0 | 0 | 0 | 0 | | | | T |
| Surface | | IR | 0.99 | 88 | 53 | 15 | 20 | 0.432 | 0.659 | 0.205 | W | 65 | 39 | 10 | 16 | 0.446 | 0.692 | 0.200 | W |
| | | MW | 0.99 | 6 | 6 | 0 | 0 | 1.000 | 1.000 | 1.000 | W | 6 | 6 | 0 | 0 | 1.000 | 1.000 | 1.000 | W |
| Strat Temp | | | 0.99 | 286 | 195 | 29 | 62 | 0.580 | 0.797 | 0.364 | W | 127 | 85 | 15 | 27 | 0.551 | 0.764 | 0.339 | W |
| | | | 0.99 | 51 | 10 | 9 | 32 | 0.020 | 0.647 | -0.608 | T | 32 | 4 | 8 | 20 | -0.125 | 0.500 | -0.750 | T |
| Trop Temp | | | 0.99 | 465 | 216 | 147 | 102 | 0.148 | 0.368 | -0.071 | T | 118 | 51 | 15 | 52 | 0.305 | 0.746 | -0.136 | T |
| | | | 0.99 | 61 | 40 | 3 | 18 | 0.607 | 0.902 | 0.311 | W | 47 | 34 | 2 | 11 | 0.681 | 0.915 | 0.447 | W |
| Water vapor | | | 0.99 | 383 | 350 | 19 | 14 | 0.864 | 0.901 | 0.828 | W | 51 | 46 | 1 | 4 | 0.882 | 0.961 | 0.804 | W |
| | | | 0.99 | 35 | 32 | 0 | 3 | 0.914 | 1.000 | 0.829 | W | 29 | 28 | 0 | 1 | 0.966 | 1.000 | 0.931 | W |

**Spire vs Base**
**Jan 1 – Jul 25, 2019**



Groups | All Channels | Assimilated Channels

| Variable | Vertical | Spectrum | CI | Chans | Wins | Losses | Ties | Score | MaxScore | MinScore | WLT | A_Chans | A_Wins | A_Losses | A_Ties | A_Score | A_MaxScore | A_MinScore | A_WLT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All | | all | 0.99 | 1590 | 534 | 491 | 568 | 0.027 | 0.384 | -0.330 | T | 478 | 111 | 146 | 221 | -0.073 | 0.389 | -0.536 | T |
| | | IR | 0.99 | 1350 | 403 | 467 | 476 | -0.048 | 0.306 | -0.401 | T | 320 | 13 | 136 | 171 | -0.384 | 0.150 | -0.919 | T |
| | | MW | 0.99 | 247 | 131 | 24 | 92 | 0.433 | 0.806 | 0.061 | W | 158 | 98 | 10 | 50 | 0.557 | 0.873 | 0.241 | W |
| other | all | MW | 0.99 | 21 | 15 | 0 | 6 | 0.714 | 1.000 | 0.429 | W | 15 | 13 | 0 | 2 | 0.867 | 1.000 | 0.733 | W |
| Ozone | | IR | 0.99 | 102 | 55 | 44 | 3 | 0.108 | 0.137 | 0.078 | W | 2 | 1 | 0 | 1 | 0.500 | 1.000 | 0.000 | T |
| scattering | all | MW | 0.99 | 29 | 10 | 6 | 13 | 0.138 | 0.586 | -0.310 | T | 0 | 0 | 0 | 0 | 0.000 | 0.000 | 0.000 | T |
| Surface | | IR | 0.99 | 118 | 41 | 0 | 77 | 0.347 | 1.000 | -0.305 | T | 45 | 23 | 0 | 22 | 0.511 | 1.000 | 0.022 | W |
| | | MW | 0.99 | 26 | 13 | 0 | 13 | 0.500 | 1.000 | 0.000 | T | 8 | 6 | 0 | 2 | 0.750 | 1.000 | 0.500 | W |
| Strat Temp | | | 0.99 | 292 | 43 | 186 | 63 | -0.490 | -0.274 | -0.705 | L | 124 | 1 | 80 | 43 | -0.637 | -0.290 | -0.984 | L |
| | | | 0.99 | 56 | 17 | 4 | 35 | 0.232 | 0.857 | -0.393 | T | 37 | 13 | 1 | 23 | 0.324 | 0.946 | -0.297 | T |
| Trop Temp | | | 0.99 | 493 | 170 | 111 | 212 | 0.120 | 0.550 | -0.310 | T | 116 | 3 | 19 | 94 | -0.138 | 0.672 | -0.948 | T |
| | | | 0.99 | 70 | 37 | 14 | 19 | 0.329 | 0.600 | 0.057 | W | 53 | 27 | 9 | 17 | 0.340 | 0.660 | 0.019 | W |
| Water vapor | | | 0.99 | 383 | 120 | 118 | 145 | 0.005 | 0.384 | -0.373 | T | 70 | 8 | 37 | 25 | -0.414 | -0.057 | -0.771 | L |
| | | | 0.99 | 45 | 39 | 0 | 6 | 0.867 | 1.000 | 0.733 | W | 45 | 39 | 0 | 6 | 0.867 | 1.000 | 0.733 | W |

## Results, Conclusions, and Future Work

- Evaluating the background fit to observations is a useful **verification** tool in **observation space** that **achieves** statistical **significance much faster** than traditional verification methods in model space
- Substantial savings in **computer time** and **developer time** can be realized
- As it uses innovations that are already produced in the normal data assimilation process, there is **little overhead needed to compute** and evaluate their statistics
- A lightweight Python implementation **runs in minutes on a laptop for months of data** (satellite radiances only at present)
- Output plots for each satellite, instrument, and user-defined channel group yield useful **diagnostic** information to the researcher
- Output **scorecard** is a useful **summary and decision aid**

- Evaluation of reliability and fidelity to traditional forecast scores
- Fits to bias-corrected innovations, absolute bias
- Add other space-based observations such as AMVs, scatterometers, GPS-RO, Lidar, future sensors
- Regional statistics and scorecards to match traditional verification, or for limited area models
- Longer lead times (e.g. fit to 5-day forecasts), which will project onto model error
- OSE and OSSE applications, e.g. if new observations fit poorly, there may be a problem with the new observations or the forward model rather than the forecast model

*Poster presented on Nov 4, 2019, at the 22nd International TOVS Study Conference (ITSC) in Saint-Sauveur, Quebec, Canada*