Monitoring long data assimilation time series: a reanalysis perspective with ERA-Interim

Paul Poli, Dick Dee, Paul Berrisford, Sakari Uppala

Introduction: global reanalyses

- **Goal**: Produce datasets based on observations describing the state of the atmosphere, that are consistent: physically, globally, and in time
- **Methodology**: Use a fixed version of a state-of-the-art weather model and data assimilation system (DAS), assimilating as many observations as possible
- **Difficulty**: Besides making sure that no (major) bugs undermine the attempt of using a "fixed version DAS", we have to deal with the irregular variations of the observing system in quantity and in quality, over time and in space

Outline

1. The many dimensions of data assimilation in reanalysis

2. An attempt to get a better grip on the observing system diagnostics: observation statistics database

3. Conclusions and perspectives

Outline

1. The many dimensions of data assimilation in reanalysis

- Current reanalysis at ECMWF: ERA-Interim
- Monitoring of Data Assimilation Performance
- Complexity of the observing system

Current Reanalysis System at ECMWF: ERA-Interim

Now continuing in real-time

	ERA-15	ERA-40	ERA-Interim	ERA-75 (target)
TIME PERIOD	*1979-1993*	*1957-2002*	*from 1989 onwards*	*from 1938 onwards*
USERS	Meteorologists and Atmospheric Scientists			
		Climate Scientists and Wider Earth Science Community		
			Additional "Environmental Users"	
				European Stakeholders **GMES Core & Downstream services**
INPUT DATA ACCESS	Mixed Observational Data Formats in Archive			Unified, Consolidated Database Facility
		Observation Quality Feedback Information		
				Internet Access
GRIDDED PRODUCTS	**Model Fields (GRIB format)**			
				Real-time Product Database **Essential Climate Variables** **Internet Access**
ATMOSPHERE	Assimilation OI 31 levels 150km	Assimilation 3DVAR 60 levels 125km	**Assimilation 4DVAR** **60 levels** **80km**	**Assimilation weak-constraint 4DVAR** **91 levels** **40 km** **Improved Observations**
LAND	Forcing	Model	**Improved Model**	**Improved Model & Assimilation Coupling**
OCEAN & SEA-ICE	SST/ice Forcing	**Improved SST/ice Forcing** **Wave Model**		**Improved SST/ice Coupling**
CHEMISTRY		Forcing	**Improved Forcing**	**Improved Interaction**
IMPACT	Enhance Understanding of Atmospheric Variability, Leading to Improved Models			
		Investigate Past Weather and Climate, Assess Observing System Impact		
			Monitor Near Real-time Climate with Traceability to Input Data	
				Facilitate Environmental Decisions, Enable New Applications of GMES, Assess Regional Climate Change & Risks via Regional Reanalyses, Improve Earth System Modeling, Maximize Benefits from Earth Observation Infrastructure

NWP Changes Affecting Quality: Mitigation in Reanalysis

(usually for the better)

1. **Data**
   - **Observing system (instrumentation – raw data)**
   - **Forcing data: SST, sea-ice, greenhouse gases…**
   - **Data processing**
2. **Data assimilation**
   - **Analysis scheme**
   - **Bias correction**
   - **Data usage: blacklist, thinning, active/passive (!)**
   - **Observation error assignment (!)**
3. **NWP forecast model**
   - **Physics**
   - **Dynamics**
   - **Resolution**
   - **Misc: computer (!), code (!), compiler (!), settings (!)**

Changes that can be minimized in a reanalysis

Requires additional collaboration

Outline

1. The many dimensions of data assimilation in reanalysis

- Current reanalysis at ECMWF: ERA-Interim
- Monitoring of Data Assimilation Performance
- Complexity of the observing system

Data assimilation performance

- How do we **qualify/quantify** it?
  - Extract the "best" information from all observations

    (scientific)

    - Make sure that the minimizations converge!
    - Make sure that the bias correction "works properly"
    - New diagnostics being developed by experts: this workshop!
  - Assimilate what we are supposed to assimilate

    (technical)

    - Keep track of the hundreds of data sources
    - Do not assimilate unwanted data ["blacklist"]
    - Do assimilate wanted data ["whitelist"]

- In reanalysis: we have the same issues, *except*:
  - Over *longer* time periods
  - Covered very quickly, typically 10 days of assimilation per day of run
  - Aim at producing time-consistent products

Dive into the Assimilation Problem: Log!

- **Excerpt from IFS 4DVAR JO table**

Types of information

```
Diagnostic JO-table (JOT) MINIMISATION JOB T0095 NCONF=   131 NSIM4D=     0 NUPTRA=     0
==========================================================================================
       Obstype     1 === SYNOP, Land stations and ships
       --------------------------------------------------
            Codetype    11 === SYNOP Land Manual Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  H2           1470             2005.605696282          1.36    0.113E+00    0.119E+00
                  Z             212             488.6433227499          2.30    0.224E+03    0.448E+02
                  PS          14009             20229.45067233          1.44    0.713E+02    0.535E+02
            Codetype    14 === SYNOP Land Automatic Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  H2           1215             1359.493317157          1.12    0.120E+00    0.108E+00
                  Z              52             247.0854971979          4.75    0.523E+02    0.429E+02
                  PS          12730             25453.43002755          2.00    0.524E+02    0.527E+02
            Codetype    21 === SYNOP-SHIP Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  U            1208             2543.019994507          2.11    0.200E+01    0.112E+01
                  PS           1096             3226.156897906          2.94    0.853E+02    0.600E+02
            Codetype    23 === SYNOP SHRED Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  U               6             12.95046365384          2.16    0.200E+01    0.102E+01
                  PS              5             21.74637926436          4.35    0.853E+02    0.556E+02
            Codetype    24 === SYNOP Automatic SHIP Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  U             828             734.5471588233          0.89    0.200E+01    0.108E+01
                  U10          1130             731.2756952020          0.65    0.200E+01    0.103E+01
                  Z               3             109.4780440042         36.49    0.412E+02    0.299E+02
                  PS           2644             5390.184158827          2.04    0.505E+02    0.610E+02
            Codetype   140 === SYNOP METAR
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  PS          20311             23482.61878113          1.16    0.800E+02    0.558E+02
                          ----------   ---------------------------   --------
      ObsType  1 Total:       56919             86035.68610659          1.51

       Obstype     2 === AIREP, Aircraft data
       --------------------------------------------------
            Codetype   141 === AIREP Aircraft Report
               Variable       DataCount           Jo_Costfunction       JO/n       ObsErr       BgErr
                  U            6176             5428.182041774          0.88    0.326E+01    0.245E+01
                  T            3414             2534.539880515          0.74    0.127E+01    0.714E+00
            Codetype   144 === AMDAR Aircraft Report
```

Data count

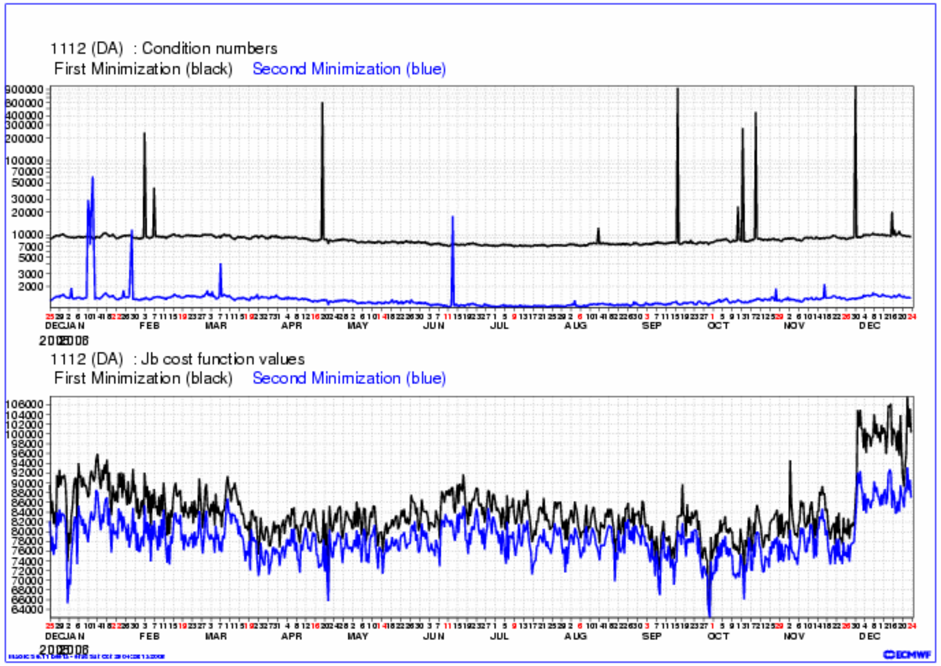
Observational part of the cost function, Assumed observation error stdev., …

Observation type, Observable type, Satellite, Sensor, …

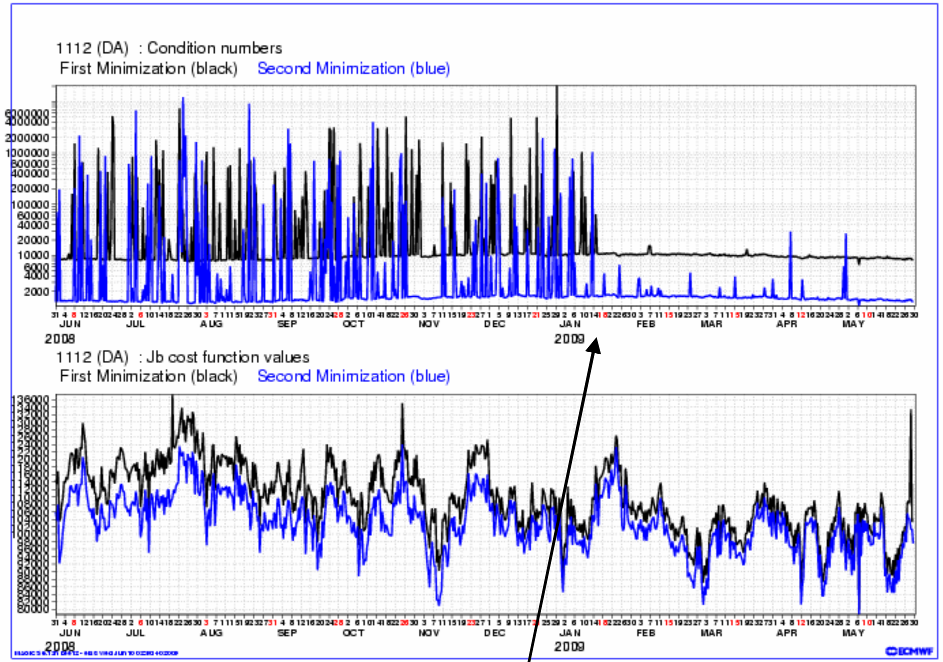
Monitoring of the minimizations in ERA-Interim

Number of GPSRO satellites: 1 (CHAMP) +6 (COSMIC), +1 (GRAS)

2006



2009



Bugfix for GPSRO radio occultation observation operator

Resolved with the help of M. Fisher and S. Healy
[had already been fixed in ECMWF operations]

ECMWF

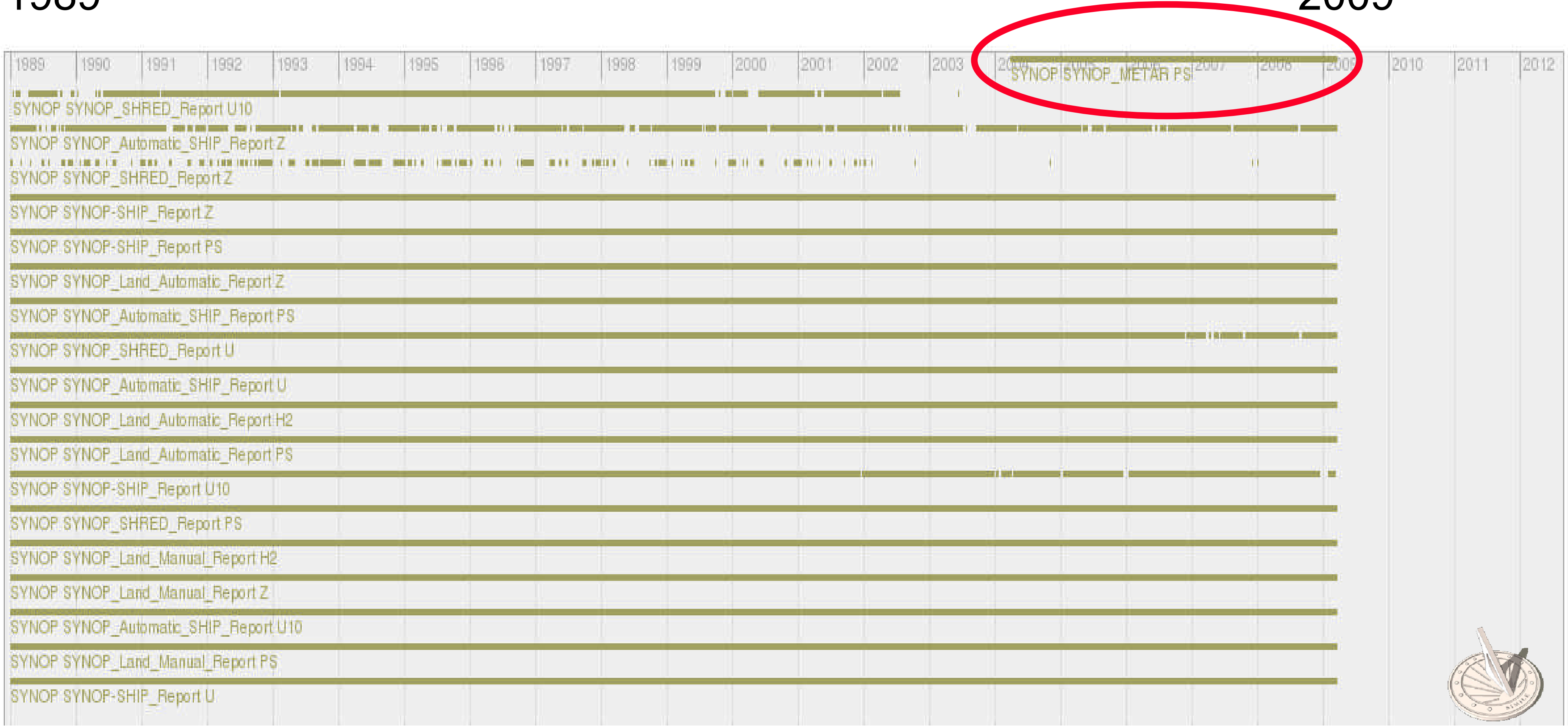
Outline

1. The many dimensions of data assimilation in reanalysis

- Current reanalysis at ECMWF: ERA-Interim
- Monitoring of Data Assimilation Performance
- Complexity of the observing system

Time coverage of in situ surface data

1989 2009

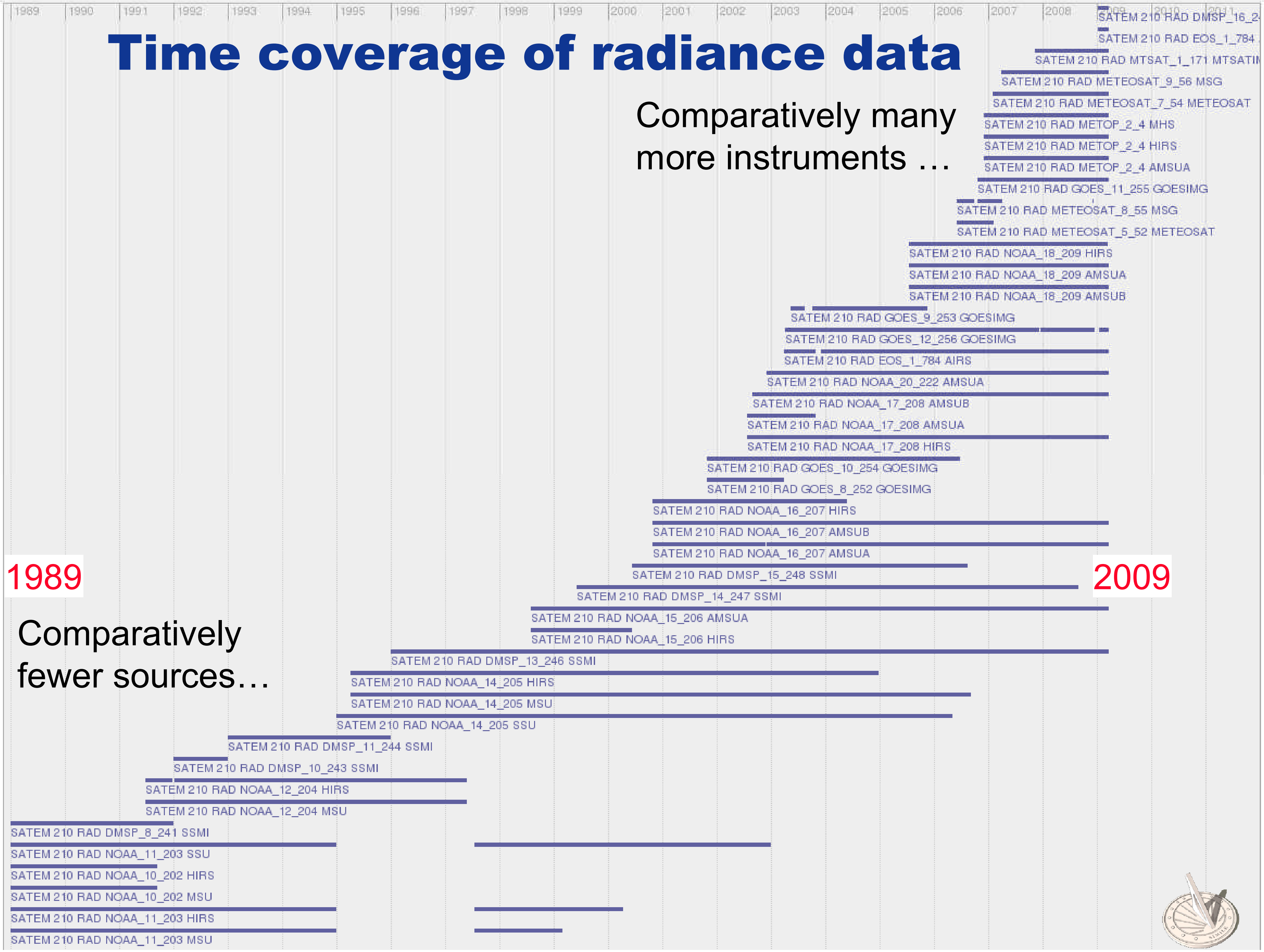


Snapshot of interactive observing system visualization tool built with:

python™ MetPy http://json.org AJAX Google code http://code.google.com/p/simile-widgets/wiki/Timeline

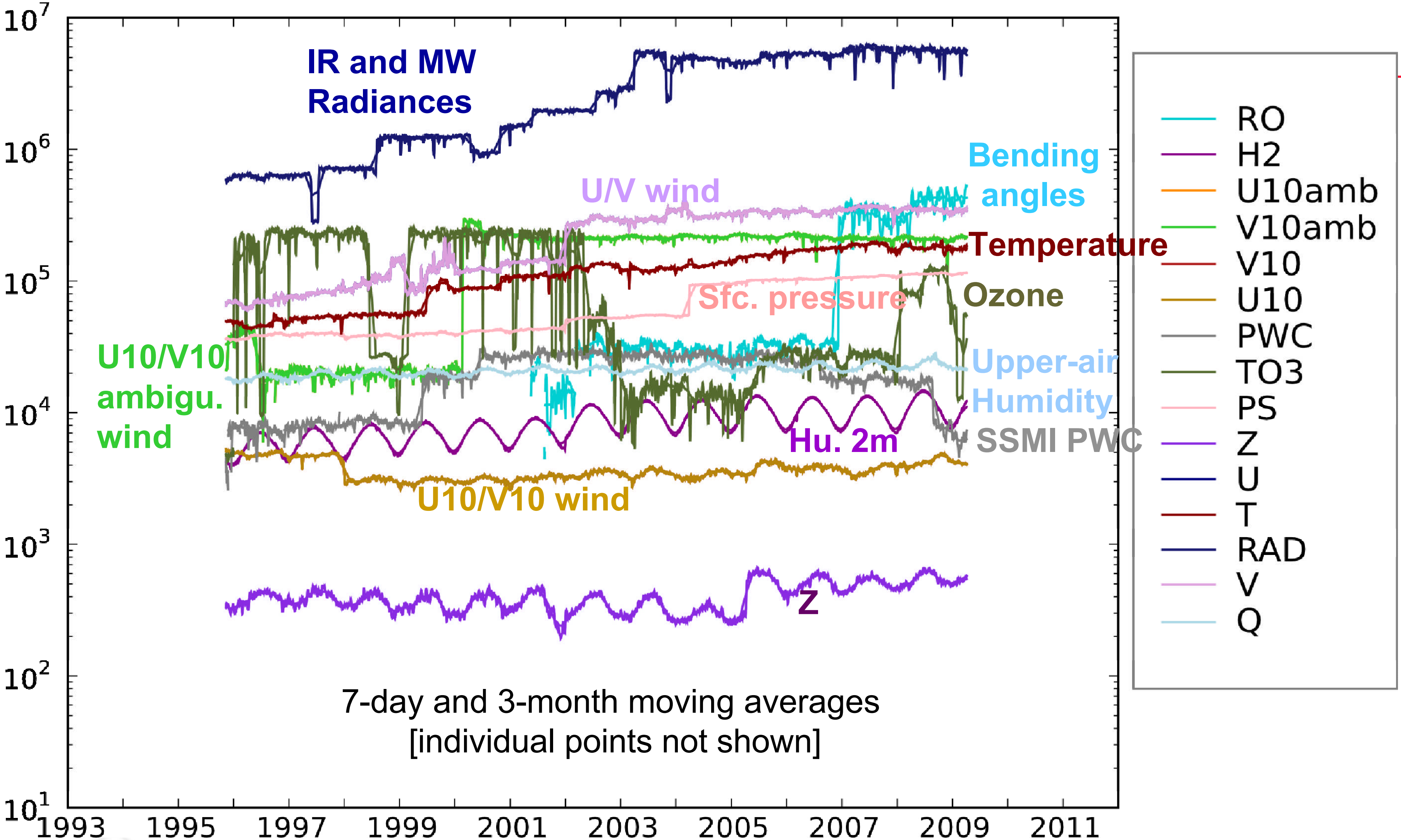
 ECMWF

Time coverage of radiance data



Data counts … by observable type

Used count : **Number of data actively assimilated per day in ERA-Interim 4DVAR**



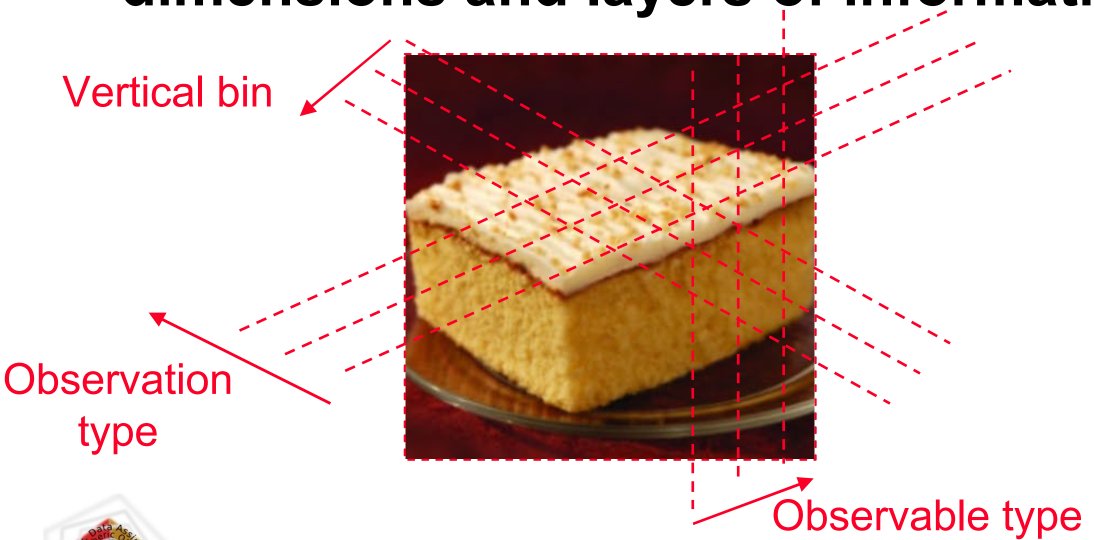
ECMWF

How many more such plots do we have to create and analyze?

- How do we automate their generation?
- How do we automatically trigger alerts?

… related to …

- How can we appropriately "cut through" all the possible dimensions and layers of information?



Outline

1. The many dimensions of the data assimilation in reanalysis

2. An attempt to get a better grip on the data assimilation performance: observation statistics database

3. Conclusions and perspectives

Outline

2. An attempt to get a better grip on the data assimilation performance: observation statistics database

- Generation of long time-series
- Analysis
- Further application

Design considerations

- **Objective:**
  - Create a data supply chain that links as directly as possible the *Observation DataBase (ODB)* to time-series

- **Constraints:**
  - Do not assume any prescribed list of data types
  - Acknowledge the fact that it is virtually impossible to specify *a priori* all the possible *plots* that would span all the dimensions of the observing system; hence: use an input (data)-driven approach instead of an output (plot)-driven approach for the statistics gathering
  - Simply want to specify once and for all what attributes are important to sort/group the observations:
    - For example, Date/Time? Observation type? Assimilation type? Pressure? Altitude? Satellite channel?

Part I: Calculate statistics directly from the ODB in 1 SQL query -- Example for observations on pressure levels

Data count

```
SELECT count(*) as count,
    sum(fg_depar@body) as sumfg_depar, sum((fg_depar@body)*(fg_depar@body)) as s2umfg_depar,
    min(fg_depar@body) as minfg_depar, r                as maxfg_depar, sum(an_depar@body) as
    suman_depar, sum((an_depar@body)*(ar                 s2uman_depar, min(an_depar@body) as
    minan_depar, max(an_depar@body) as maxan_depar, expver@desc as expver, andate@desc as andate,
    antime@desc as antime, obstype@hdr as obstype, codetype@hdr as codetype, varno@body as varno,
    satname_1@hdr as satname_1, satname_2@hdr as satname_2, satname_3@hdr as satname_3, satname_4@hdr as
```

Diagnostics

Sorted by
Experiment ID,
Date,
Time,
Observation type and name (SYNOP, TEMP, …),
Observation sub-type and name (Land Automatic Report, …)
Observable type (Temperature, U-wind, …)
Assimilation type (active, passive, …)
Pressure level bin,
Latitude bin

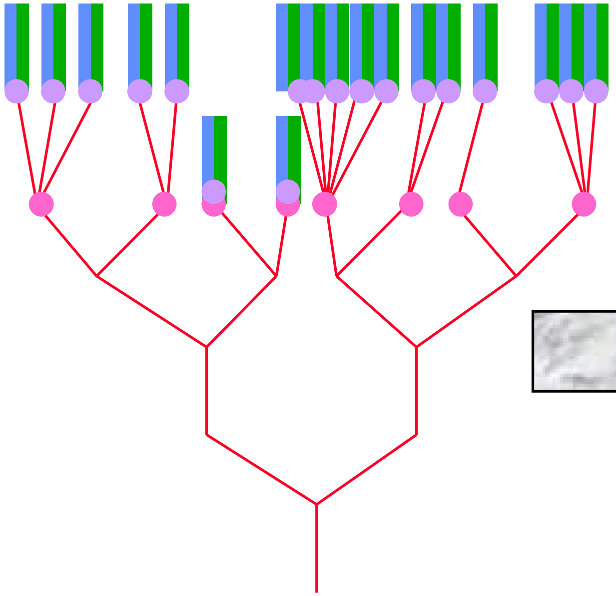
```
FROM desc, hdr, body

WHERE
    n
```

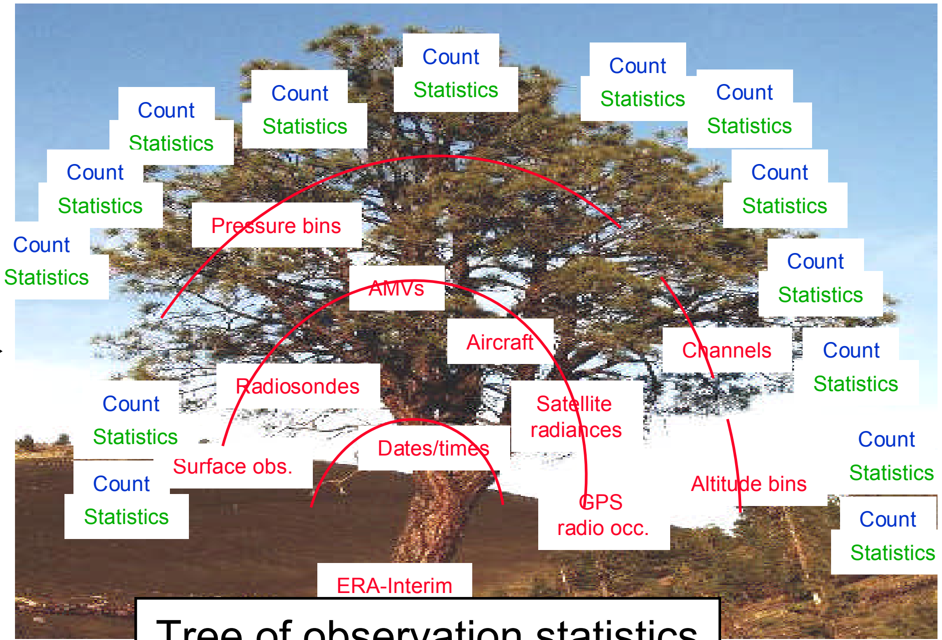
Restrict to observations on pressure levels

Part II: Automate the SQL query generation and build a tree of observation statistics

- How do we make sure we don't forget any query to span the entire observation database?
- How do we write these requests automatically?
- **Solution: tree of requests and conditional keys**



Tree of SQL requests



Tree of observation statistics

Built with: ODB python MetPy json.org

Excerpt of a tree of observation statistics

(direct view of the data structure from the web browser)

```
{
  - where==status_at_body=1 AND (obstype_at_hdr = 7 and codetype_at_hdr = 210) AND (obstype_at_hdr = 7 and codetype_at_hdr = 210): {
    - status@body==1: {
      - varno@body==119: {
        - satname_4@hdr==AMSUA : {
          + press@body==14: { … },
          + press@body==10: { … },
          + press@body==11: { … },
          + press@body==12: { … },
          + press@body==13: { … },
          + press@body==6: { … },
          + press@body==7: { … },
          + press@body==5: { … },
          + press@body==8: { … },
          + press@body==9: { … }
        },
        + satname_4@hdr==SSMI : { … },
        + satname_4@hdr==AIRS : { … },
        + satname_4@hdr==METEOSAT: { … },
        + satname_4@hdr==MSG : { … },
        + satname_4@hdr==HIRS : { … },
        + satname_4@hdr==SSMIS : { … },
        + satname_4@hdr==GOESIMG : { … },
        + satname_4@hdr==MSU : { … },
        + satname_4@hdr==SSU : { … },
        + satname_4@hdr==AMSUB : { … },
        + satname_4@hdr==AMSR-E : { … },
        + satname_4@hdr==MHS : { … },
        + satname_4@hdr==MTSATIMG: { … }
      }
    }
  }
}
```

Observable type 119 (radiances) as found in ODB

Instruments as found in ODB

AMSU-A channels as found in ODB

Part III: Create and populate an observation statistics database

- We insert the "tree" of statistics into … an SQL-type database (PostgreSQL for now), thus effectively stacking several cycles of observation statistics over one another to construct a 20+-year-deep database

- Very good news is…
  - We can apply the same "tree logic" to extract statistics from SQL and have them grouped automatically to generate time-series

- This approach
  - Will still be relevant with the next-generation observation (SQL) database at ECMWF, because <u>it relies exclusively on the SQL engine</u> to calculate the statistics
  - Opens up the possibility to generate <u>quickly</u> and <u>interactively</u> time-series, organized according to a tree definition that can be modified at any time

Outline

2. An attempt to get a better grip on the data assimilation performance: observation statistics database

- Generation of long time-series
- **Analysis**
- Further application

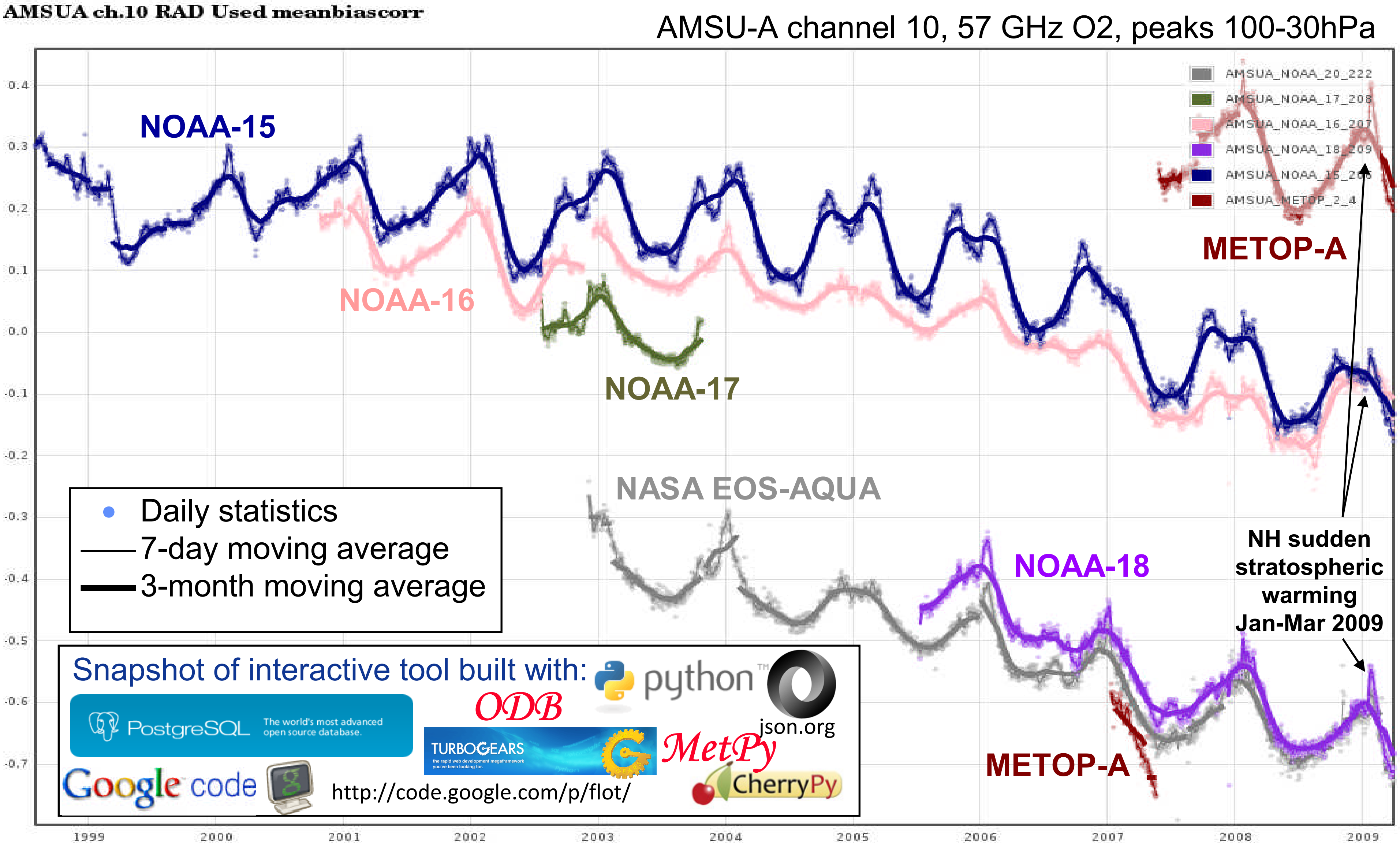
Time-series Investigation

1. Start by plotting the time-series!

2. Most tools / statistical methods available to automatically “process” time-series assume that:
   - The time-series are representative of the same “observable” throughout the time period
   - The data have been “cleaned-up” – there are no outliers …

3. We first have to get a feeling for what may be problematic in our time-series, before passing them on to automatic time-series processing tools

AMSU-A Bias correction

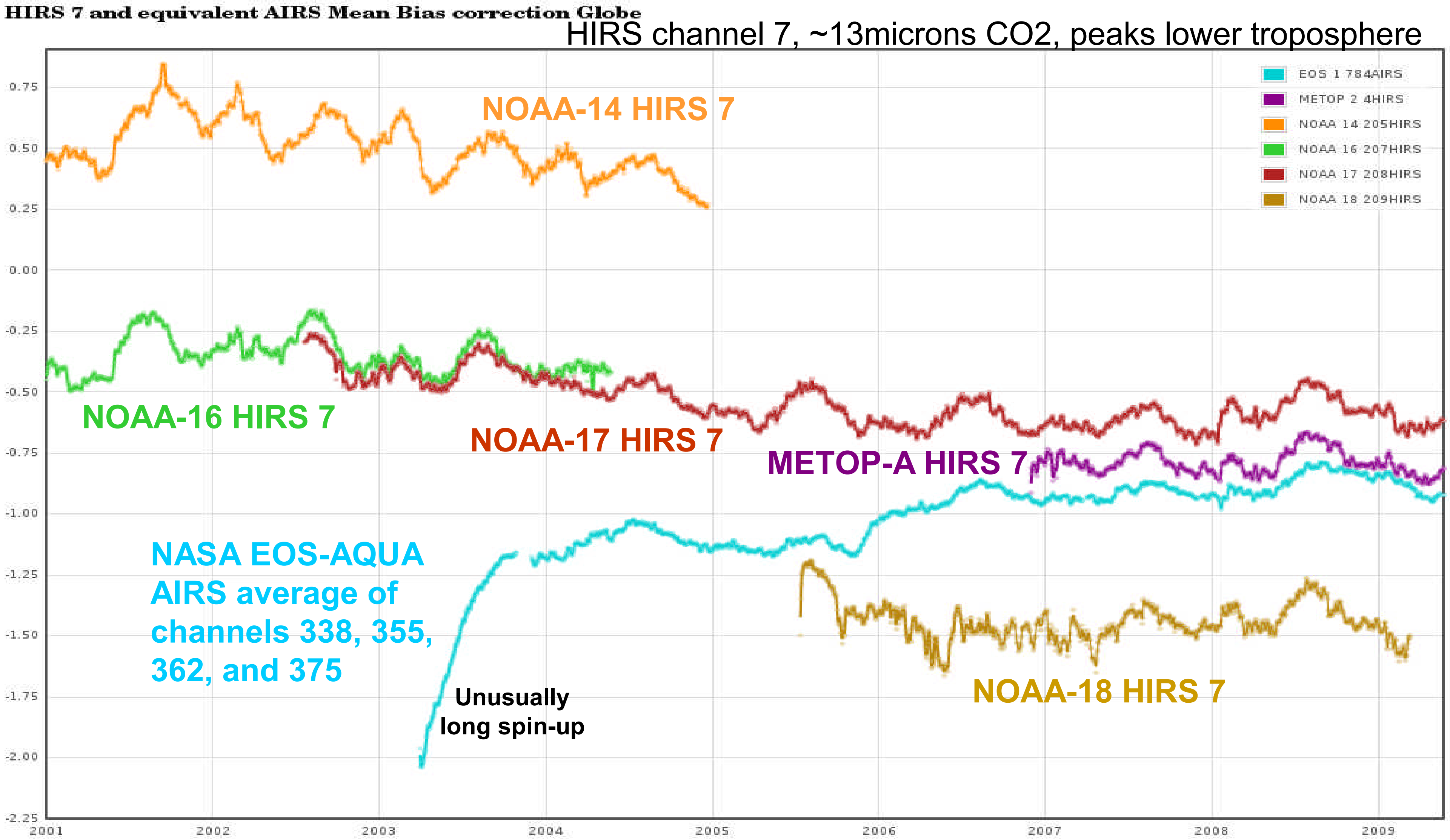


Snapshot of interactive tool built with:

PostgreSQL The world's most advanced open source database. ODB python json.org TURBOGEARS MetPy CherryPy Google code

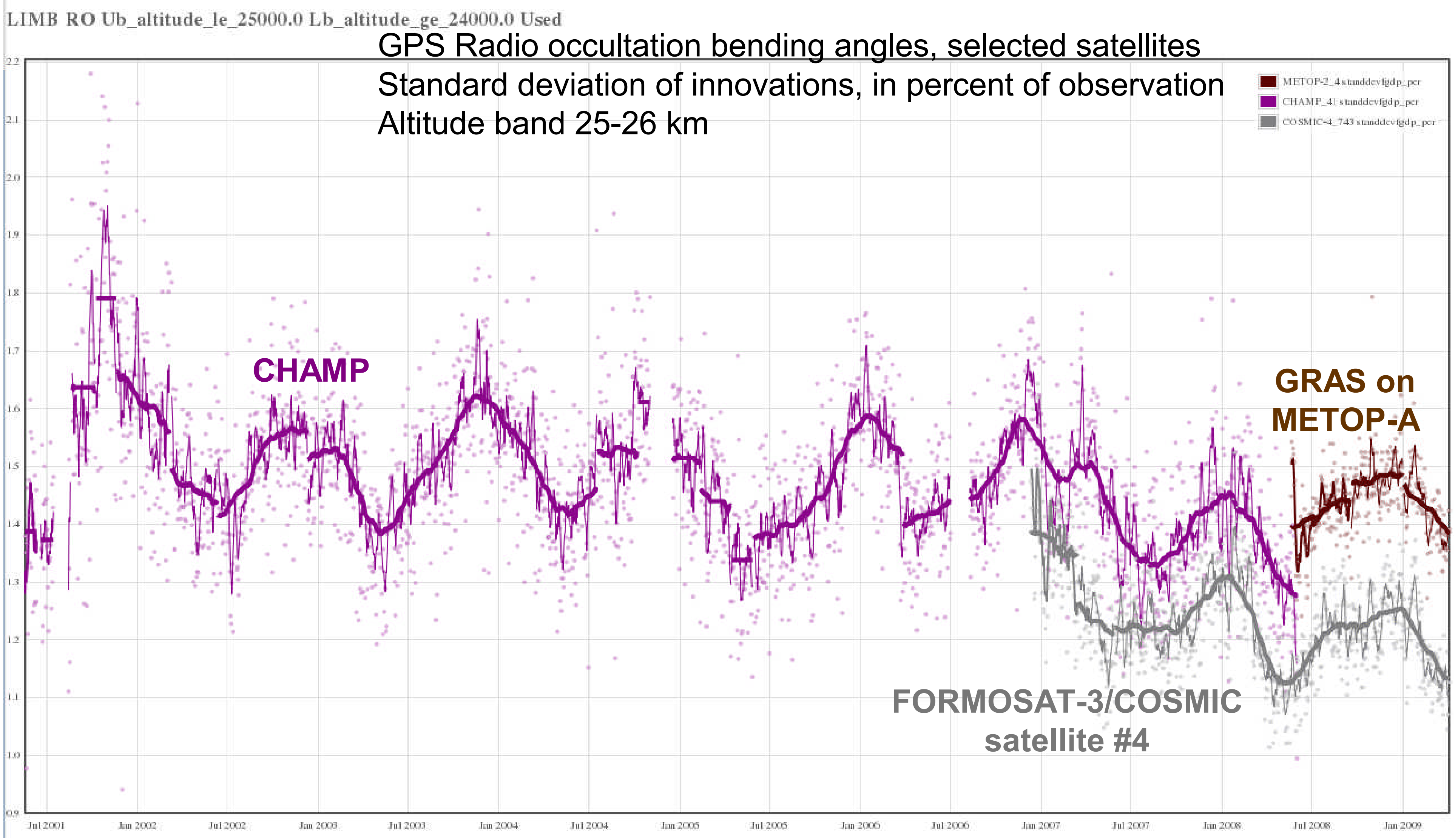
http://code.google.com/p/flot/

Comparison HIRS/AIRS bias corrections

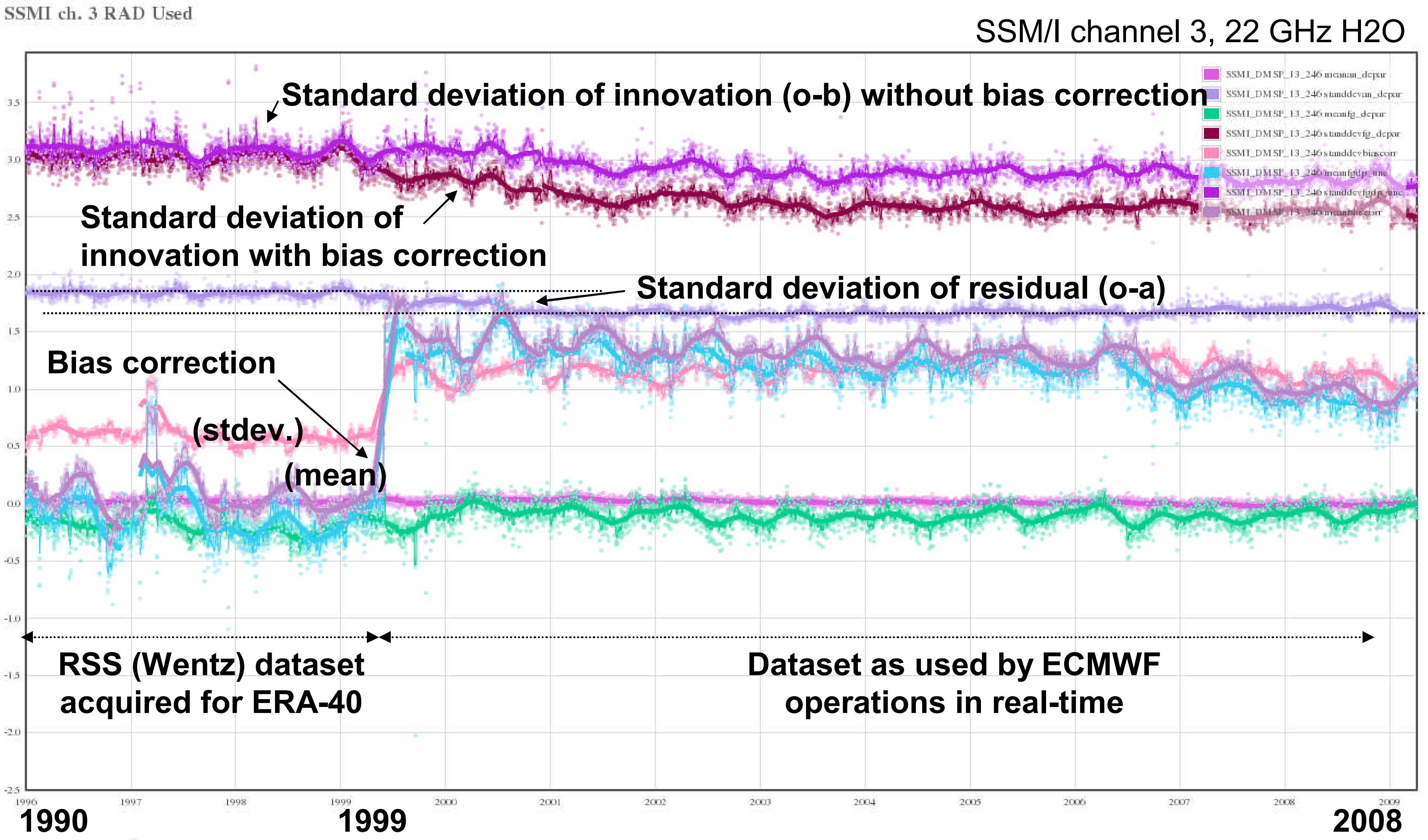


ECMWF

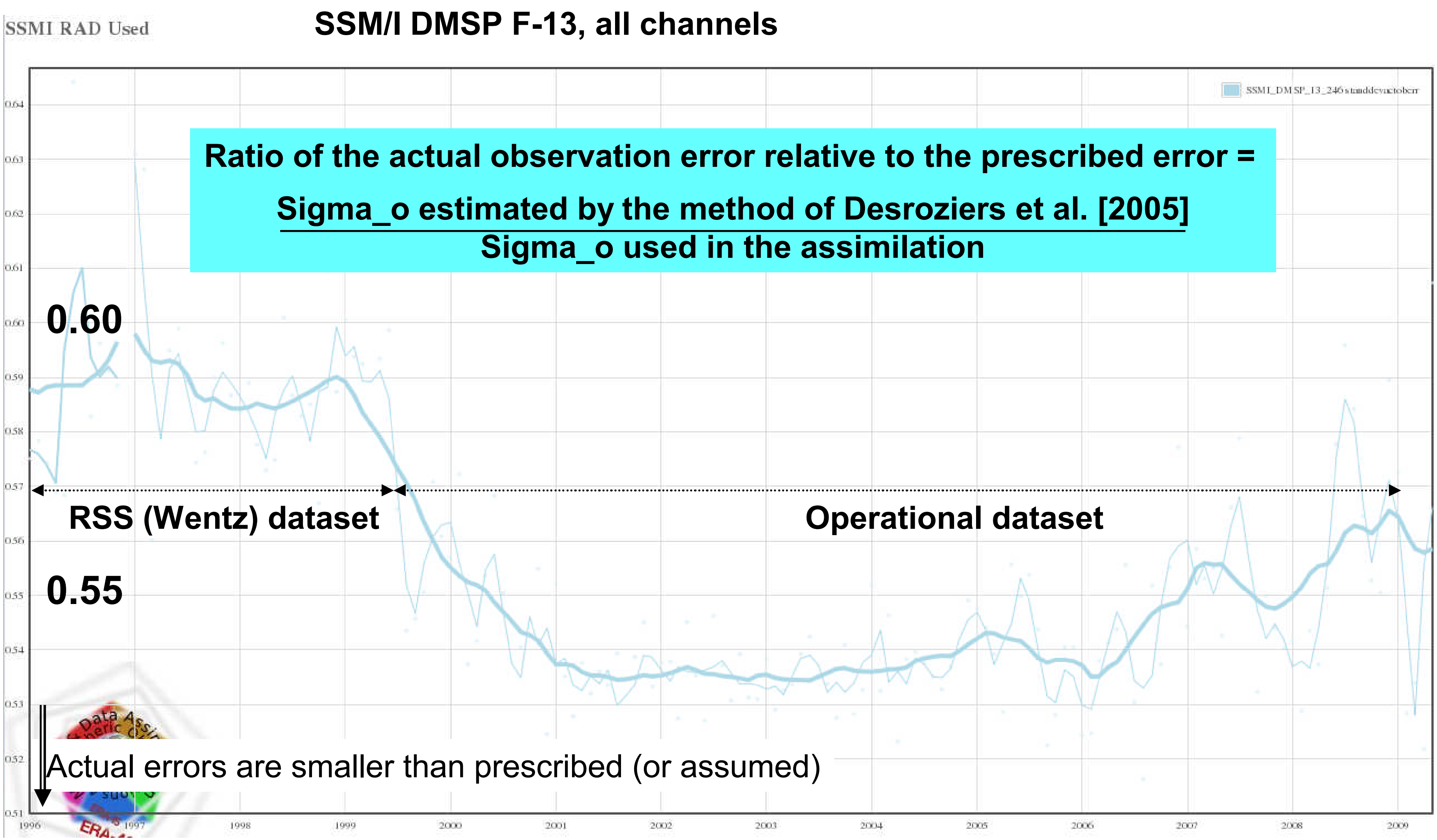
Time-series of GPSRO innovations



SSM/I DMSP F-13 Innovations

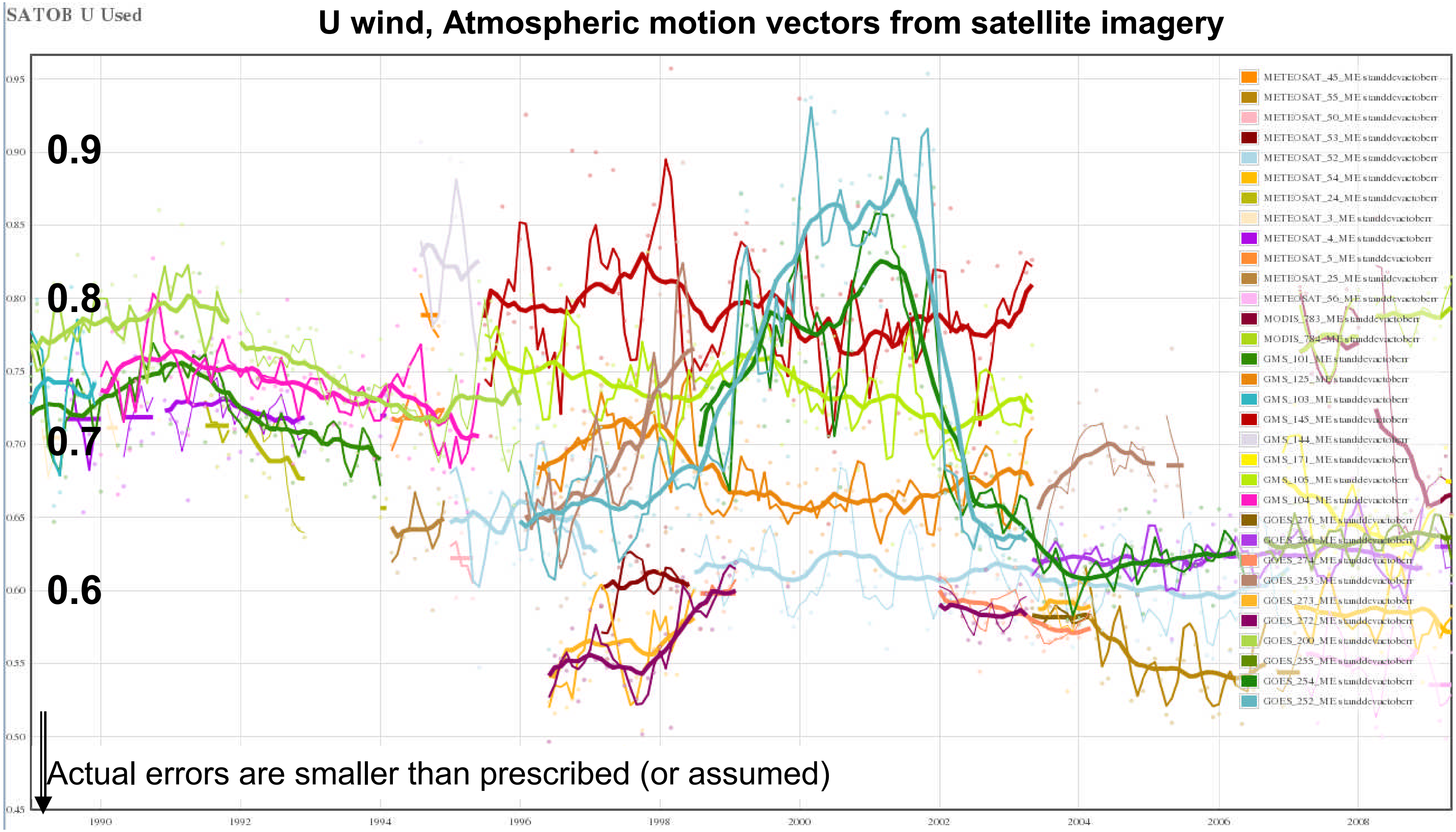


Ratio of (actual over prescribed) sigma_o



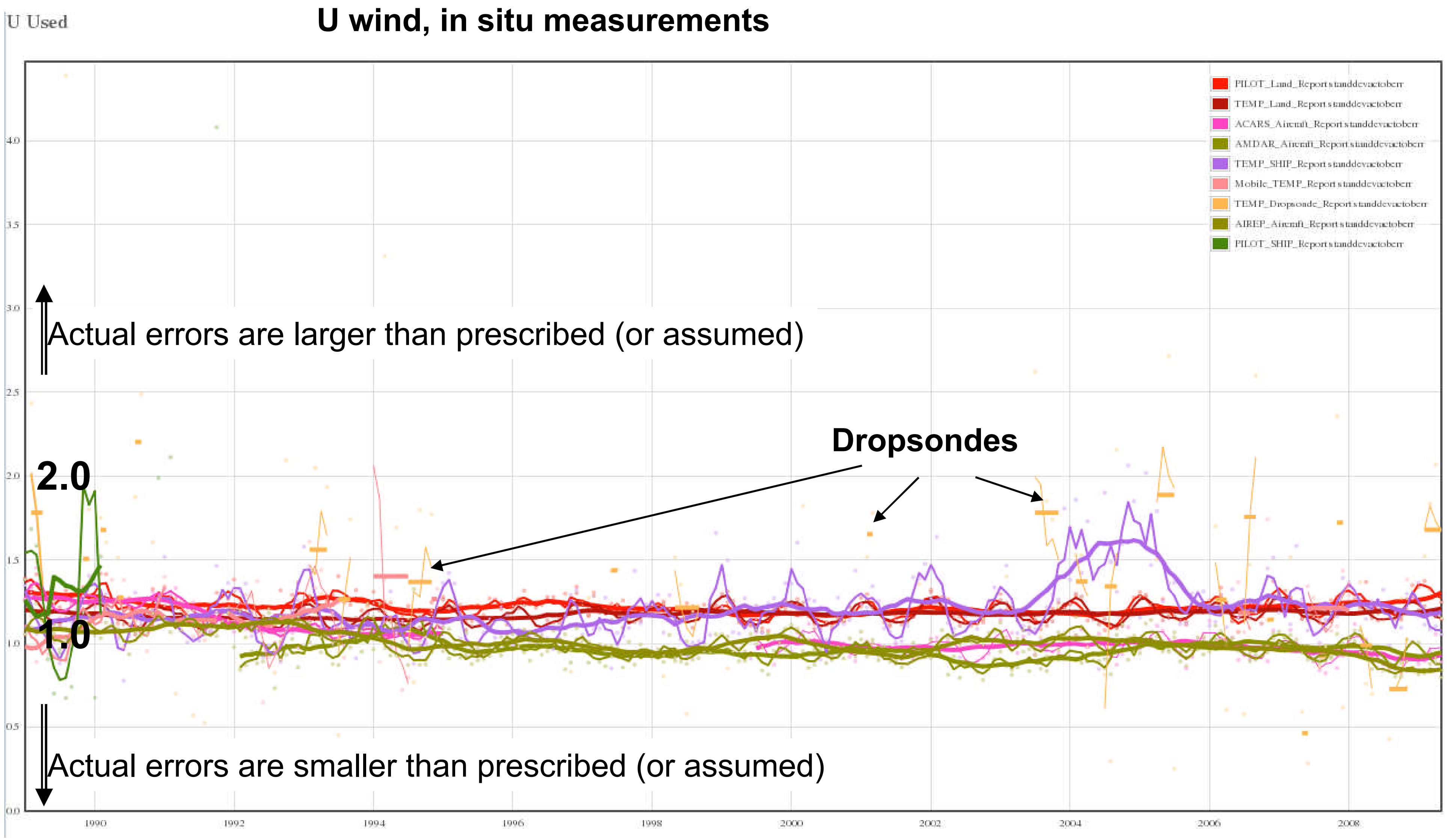
ECMWF

Ratio of (actual over prescribed) sigma_o

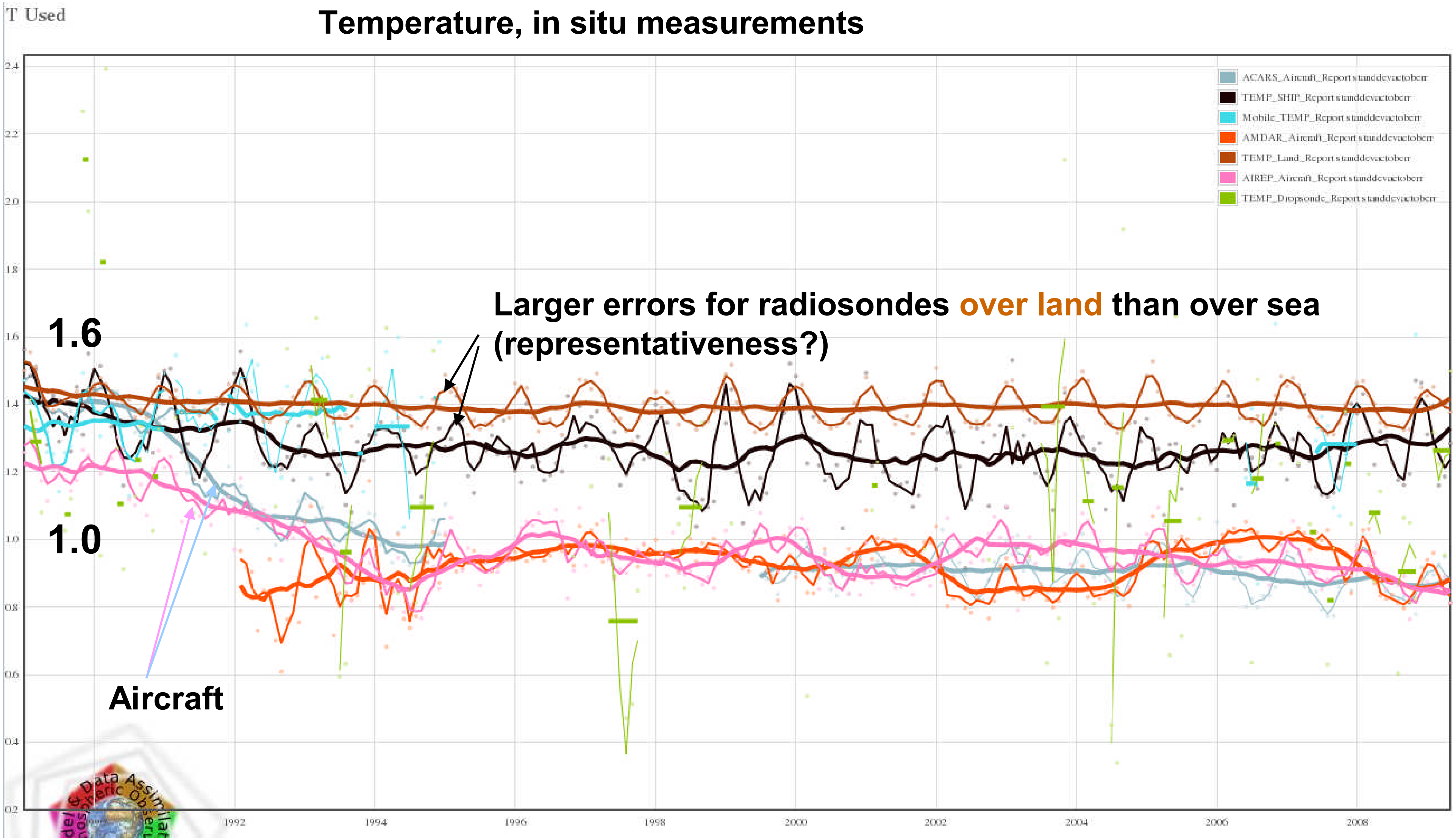


ECMWF

Ratio of (actual over prescribed) sigma_o



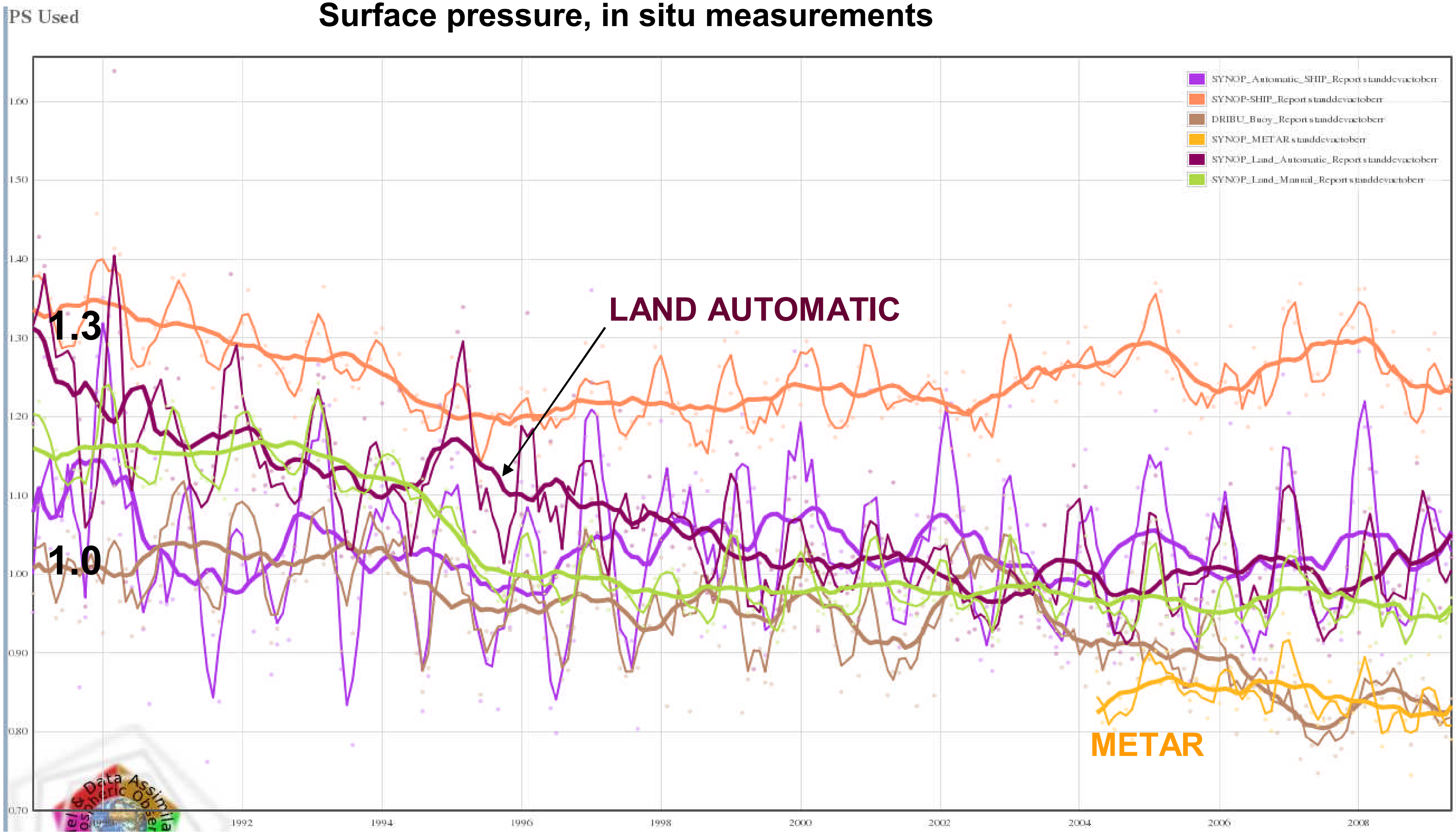
Ratio of (actual over prescribed) sigma_o



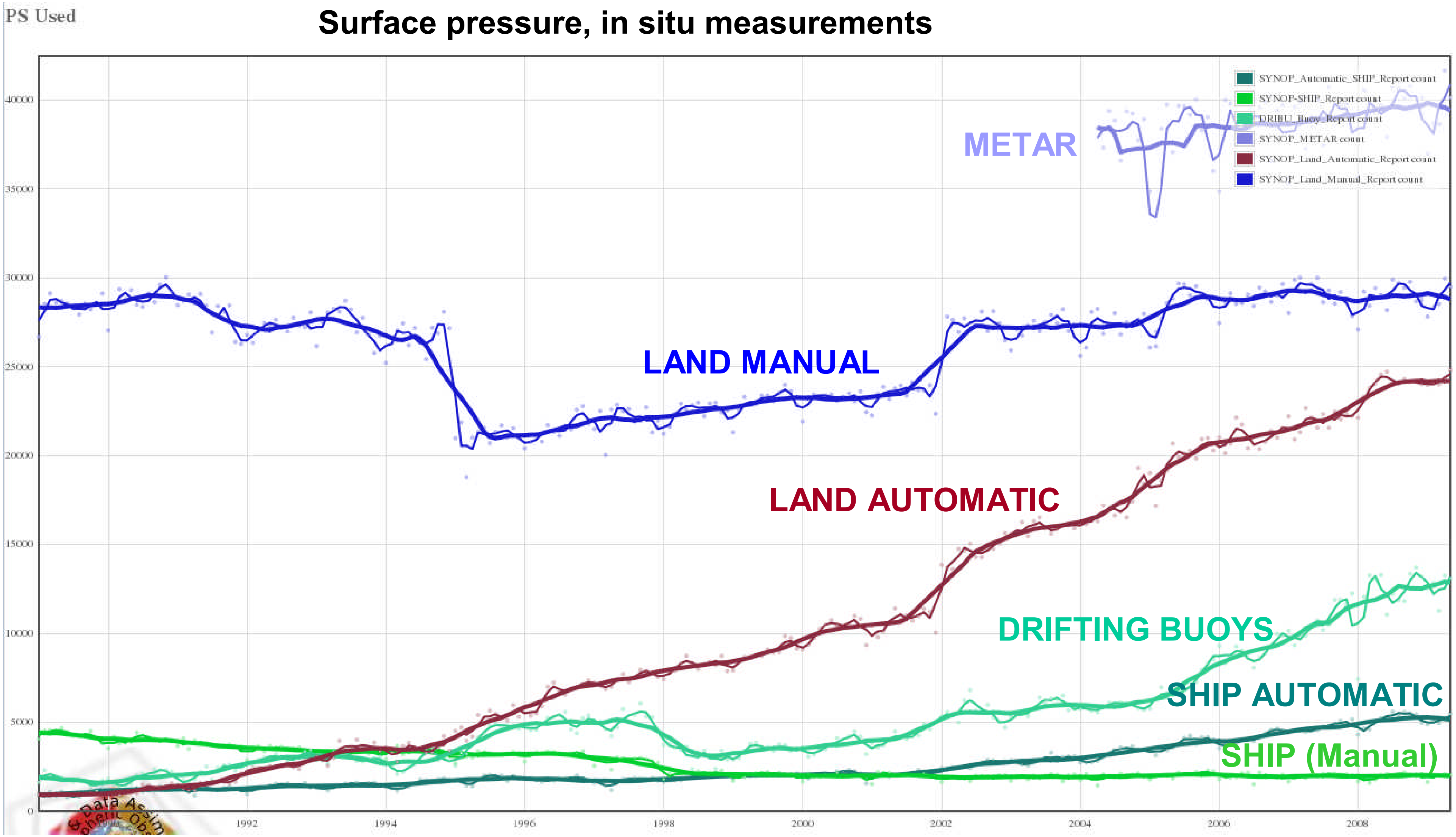
ECMWF

Ratio of (actual over prescribed) sigma_o

Surface pressure, in situ measurements



Count of data assimilated daily in 4DVAR



Outline

2. An attempt to get a better grip on the data assimilation performance: observation statistics database

- Generation of long time-series
- Analysis
- Further application

Time-series: Various Types

- **Physical data:**
  - Observations
- **Process-generated data:**
  - Innovations (O-B), residuals (O-A), bias corrections
  - Very likely more affected by time-correlation than physical data
- **Process control data:**
  - Fit before and after minimization, bias correction…
  - Useful to check that data and products fall within some range

- **Common points in all these time-series:**
  - Aggregate of sensors only valid if the aggregation remains the same
  - Need to consider individual sensors?

How many time-series then...?

Type of "tuple"	Number	Found over
Surface station ⊗ instrument	~15000 stations ⊗ 4 variables = ~25000 tuples	5 years (2004-2009)
Drifting buoy ⊗ instrument	~2000 buoys ⊗ 3 variables = 2500 tuples	5 years (2004-2009)
Radiosonde station ⊗ instrument	1623 stations ⊗ 4 variables = ~5000 tuples	5 years (2004-2009)
Aircraft platform ⊗ instrument	~2500 aircraft ⊗ 3 variables = ~7000 tuples	12 hours in 2009
Satellite ⊗ wind product	79 tuples	20+ years (1989-2009)
Satellite ⊗ radiometer ⊗ channel	28 satellite ⊗ 14 instruments ⊗ 394 channels = 636 tuples	20+ years (1989-2009)
Satellite ⊗ ozone instr.	16 tuples	20+ years (1989-2009)
Satellites with scatter.	3	20+ years (1989-2009)
Satellite with GPSRO	9	20+ years (1989-2009)

Number of time-series

Variability contained in each time-series

Time-series Investigation: Rationale

1. **Describe:** -- Can we detect:
   - Breaks? Seasonality / cycles? Trends? Outliers?
2. **Analyze:** -- Can we explain:
   - The origins of the breaks? The cycles? Are the outliers symptoms of problems in the DAS or simply the results of occasional poor sampling?
3. **Detect:** -- Could we improve:
   - The alarm system to detect problems in the incoming data? Statistical models from long time-series could be used as basis from where to automatically trigger alerts as the screening encounters problematic data – with applications for operational NWP
4. **Control:** -- Check the assimilation performance:
   - 4DVAR, VarBC: “process control” statistics

Conclusions

- **Generating observation-related time-series from a data assimilation system can require significant efforts**
  - <u>Easy approach</u>: long, straightforward scripts and codes that "know" about the data types
  - <u>Simple approach</u>: short, apparently more complex (recursive) scripts and codes that deal with "irregular" structures
  - The differences are not really "interesting" from a scientific point of view <u>if you have somebody else *"doing the plots for you"*</u>… but even then, the resources spent there could probably be better used…
- **An experimental observation statistics database has been constructed from ERA-Interim**
  - Already allowed to find a few points that need improvement in next reanalysis: Detect when the bias spin-up has stabilized, Need to automatically trigger alarms when large changes occur in the observation statistics
  - We are not yet at the point where we can simply call automated methods to detect breaks, trends, cycles etc…
  - Considering sensor-based time-series seems to make more physical sense than aggregate of sensors, whose coverage vary over time

Future Prospects

- To reconstruct our observation statistics database with a finer granularity: (stations, surface type, lat/lon gridding, local time, timeslot…) – quite a few time-series!
  - To start investigating simple, robust methods to "process" the various types of time-series
  - To learn from the current time-series for the design of the observation handling in the next reanalysis

- To investigate how an observation statistics database could help/be implemented very close to the 4DVAR assimilation
  - To store in a unified framework the statistical information that needs propagation in time, e.g. bias correction tables
  - To avoid repeating the monitoring calculations by having them done immediately close to the assimilation
  - To integrate the observation alarm system closer to the assimilation, effectively allowing to use past time-series of observation statistics

Thank you for your attention!

ERA-Interim webpage:
http://www.ecmwf.int/research/era/do/get/index

Technical tools used to construct/serve/display the timeseries information shown in this talk

python · Google code · matplotlib · timeline · TurboGears · json.org · AJAX · CherryPy · PostgreSQL