HIRLAM – ALADIN verification project

A. Deckmyn, E. v.d. Plas, A. Singleton, X. Yang, C. Zingerle,

Outline

- Development of HARP
- Input of data
- Verification routines
- Results
  - Database files
  - Plots
- Outlook

Starting point

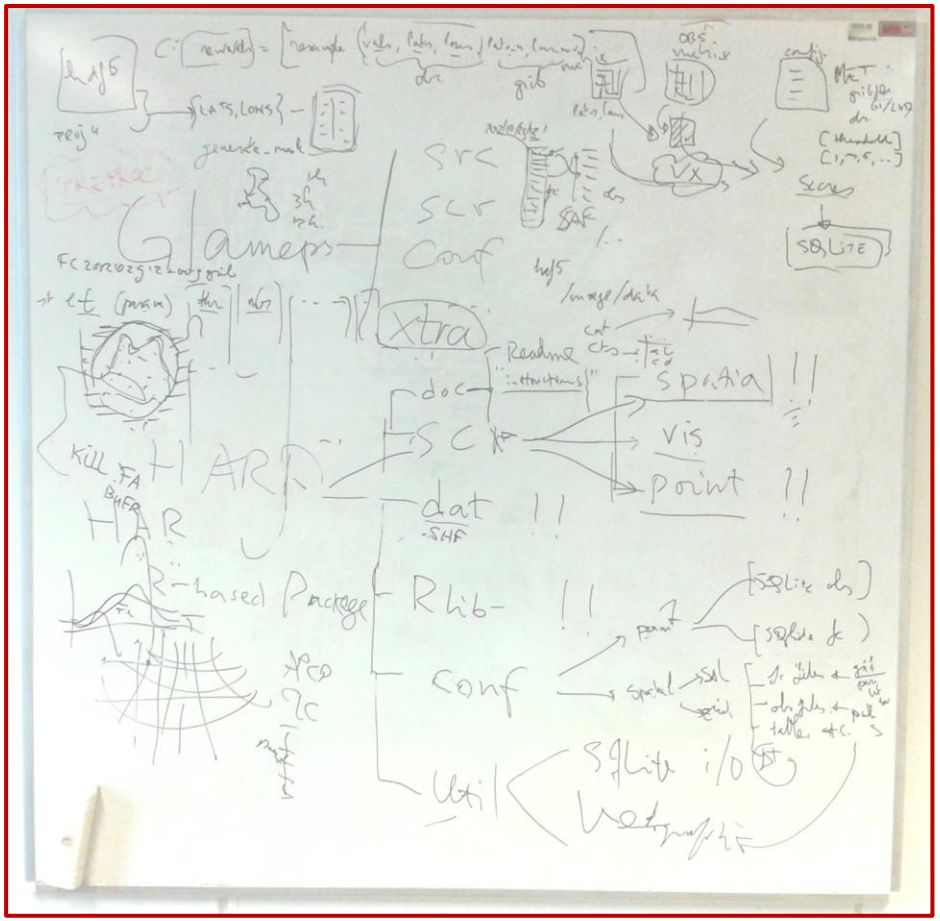
Aladin and Hirlam have established monitoring systems and verification tools for point verification

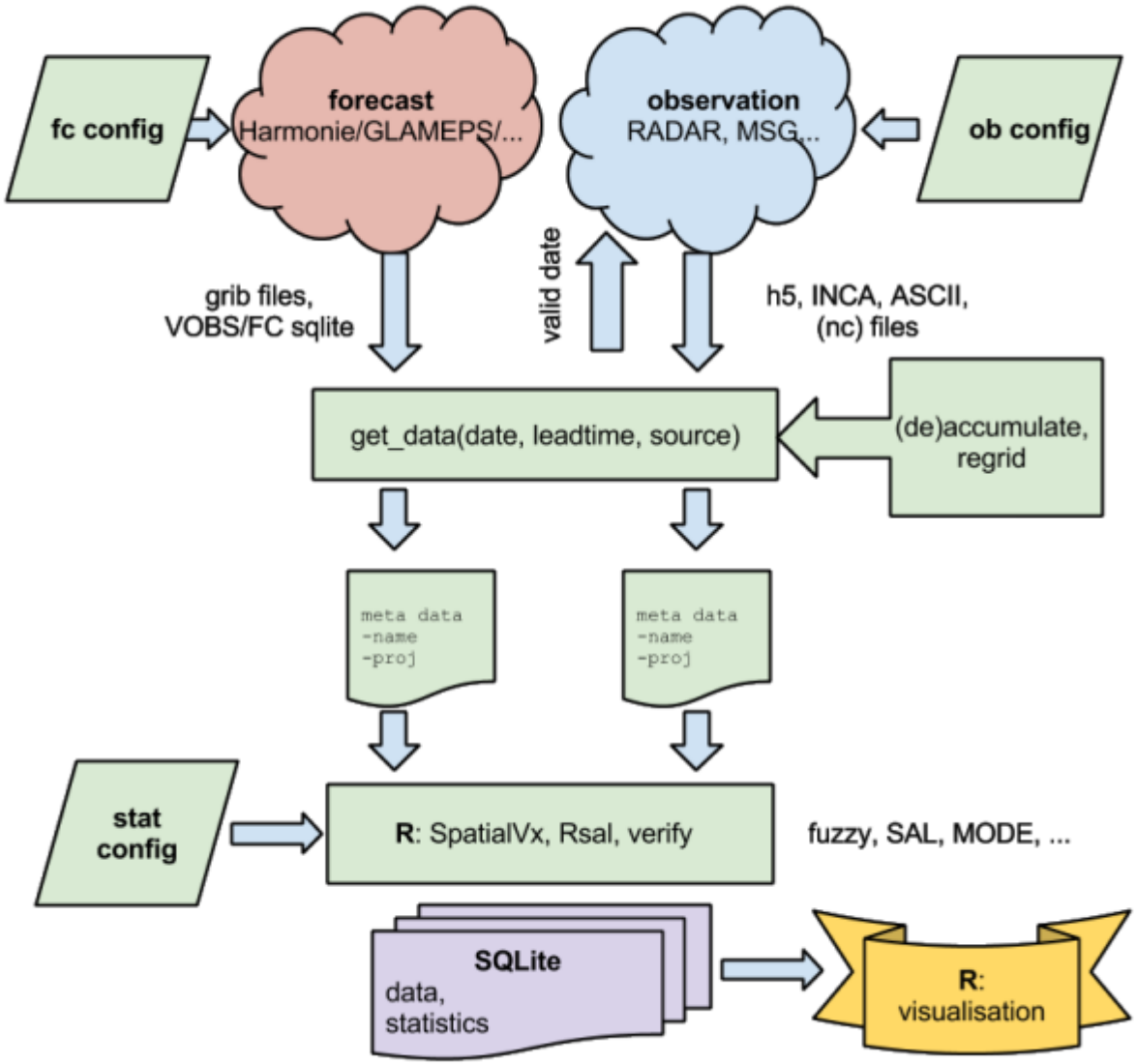
- Work together on EPS and spatial verification tools
- Build a toolbox available to the members
  - Based on R (a lot of verification packages are available
  - Common I/O of data and results
- Local adaptions of input is necessary

Hirlam – **A**ladin **R**-**P**ackage

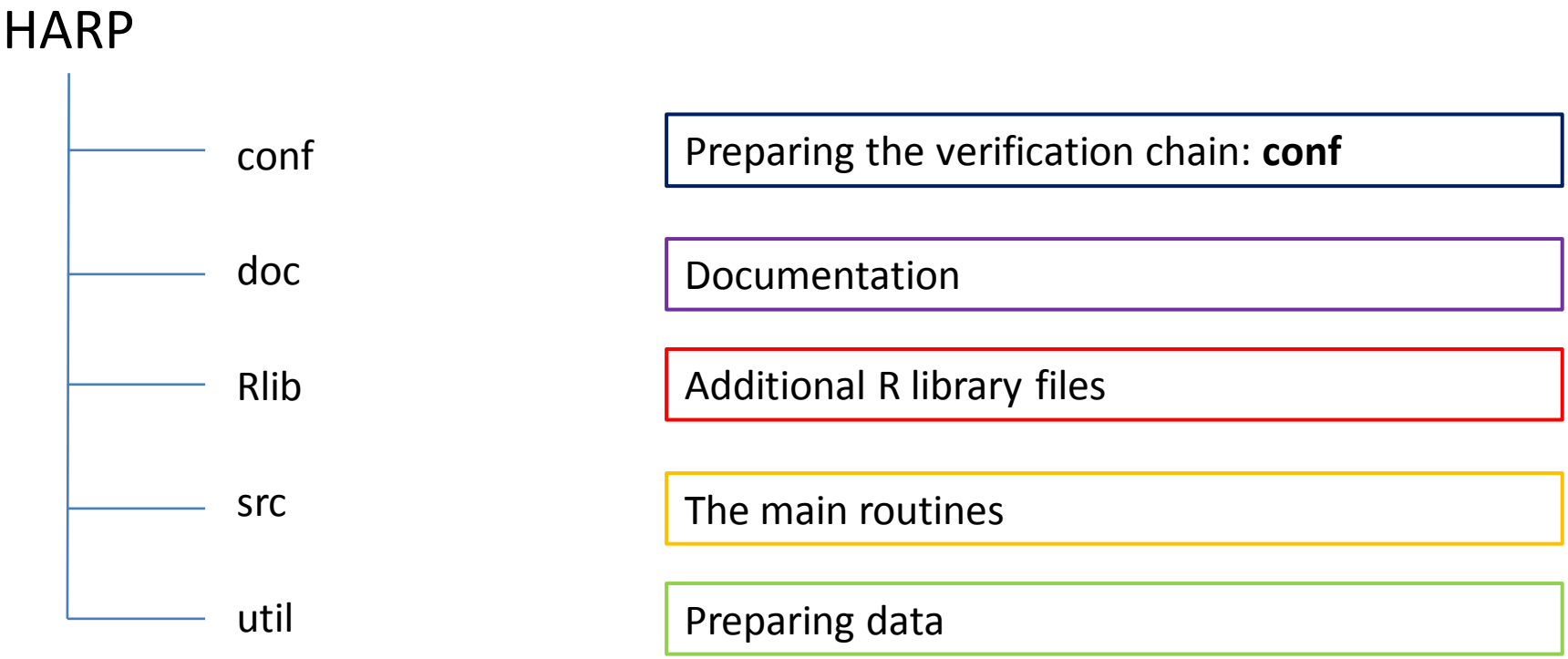
Which is not just an R-package

- Provide tools for a number of verification issues (e.g. work with spatial and point data, ensemble, …)
- Utilities for preparation of input data
- Working examples and data
- Documentation
- Open to extensions, new scores, data sources, …



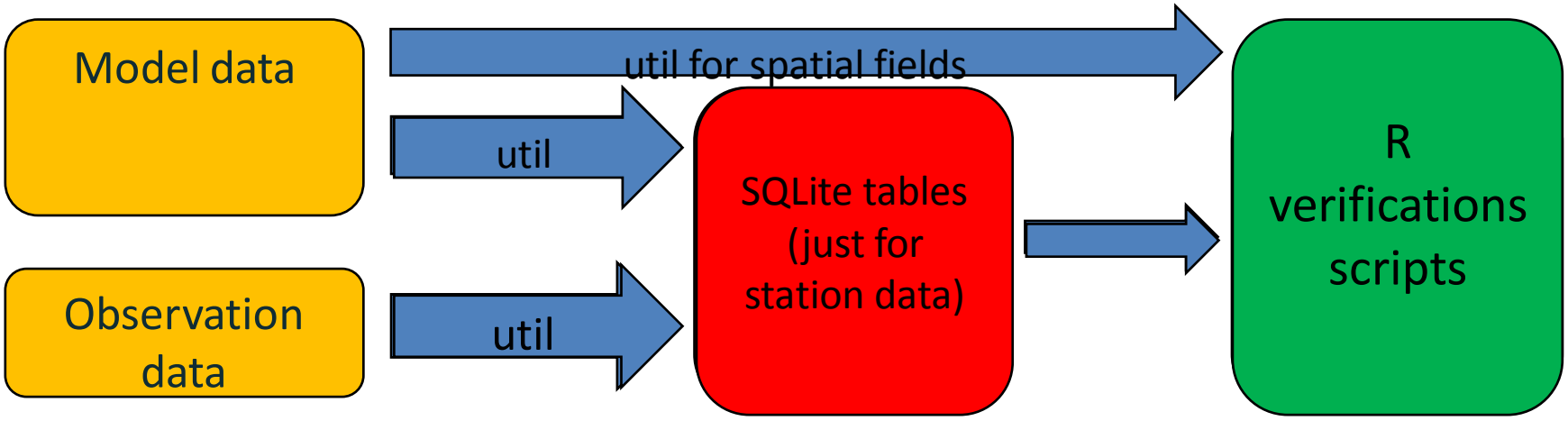

fc config
forecast
Harmonie/GLAMEPS/...
observation
RADAR, MSG...
ob config
grib files,
VOBS/FC sqlite
valid date
h5, INCA, ASCII,
(nc) files
get_data(date, leadtime, source)
(de)accumulate,
regrid
meta data
-name
-proj
meta data
-name
-proj
stat
config
R: SpatialVx, Rsal, verify
fuzzy, SAL, MODE, ...
SQLite
data,
statistics
R:
visualisation

HARP structure



util

To extract data from local environment and plug them to the interface



- Interpolation of model data to station position -> SQLite table
- Extraction of observations from VOBS -> SQLite table
- parsing of VFLD files -> SQLite table
- reading and plug of several spatial fields (Grib, FA, hdf5 ascii, bin)

- not restricted to SQLite – any other database can be used with changes in the setup

SQLite input

Station table:

Information about observation stations used in the process of verification.

Station id	Lat	Lon	Elevation	Land type
sid (int)	lat (float)	lon (float)	elev (float,m)	ltype (int?)

Forecast data tables:

Data table for each parameter / monthly

Point forecast data extracted from the model concatenated in monthly SQLite files.

-> interpolated to station position as from station tables as a view

id	fc.date	Verif.date	Lead time	System time update time	lat	lon	Member 1	Member 2
sid (int)	fcdate (int)	validdate (int)	leadtime (int)	lastmod (int)	lat (float)	lon (float)	m1name (char)	m2name (char)

SQLite input

SYNOP

Holds information from SYNOP or other surface station (by month)

-> extracted from local database (e.g. as with VOBS files)

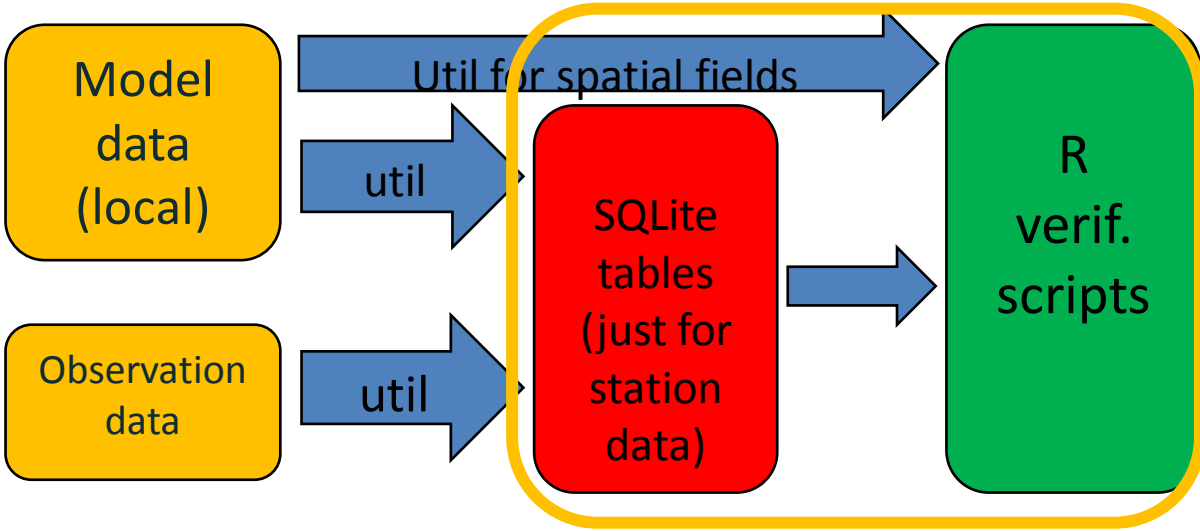
Station id	Verif.date	T2m	Wind direction	...
sid (int)	validdate (int)	t2m (float)	dd (float)	...

SQLite SNYOP input

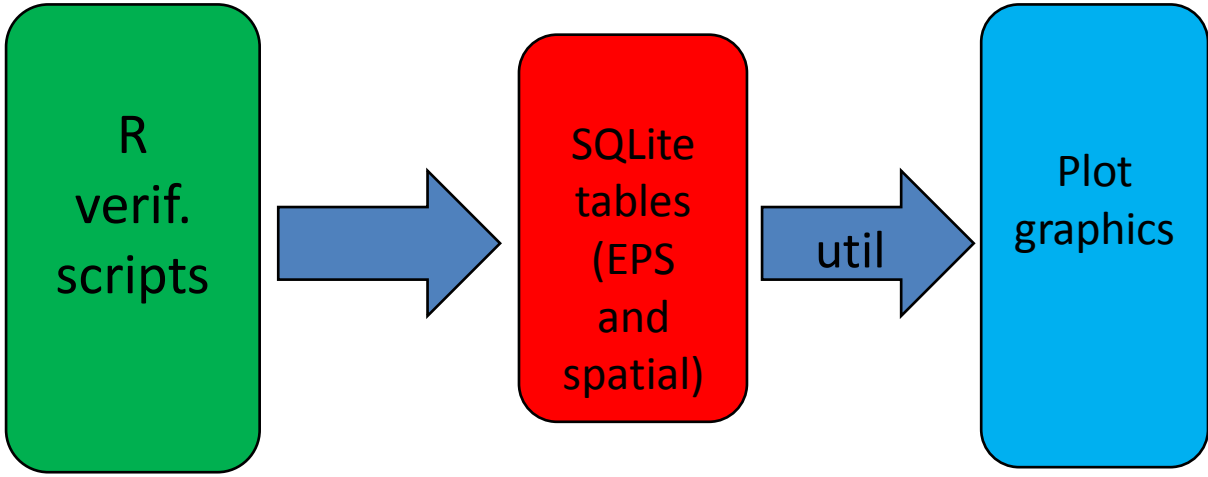
validdate	SID	lon	lat	elev	CCtot	D10m	S10m	T2m	RH2m	Pmsl	AccPcp...	Q2m	vis	Td2m	Tmax	T
1364774400	1001	-8.67	70.92	9	6	340	11	261.9	76.4	1013.4		0.001214	2500	258.6		
1364774400	1002	16.25	80.05	8		160	5	261.2		1001.3						
1364774400	1003	15.5	77	11	3					1000.5			12000			
1364774400	1006	22.82	78.25	14		230	5	254.8	84.2	1002.5		0.000750...		252.8		
1364774400	1008	15.47	78.25	26	1	120	5	257.6	74.4	1001.7		0.000842...	55000	254.1		
1364774400	1009	25	80.65	5				259.3	83.3	1003.2		0.001082		257.1		
1364774400	1010	16.12	69.3	13	6	310	11	272.3	80	1007.1		0.00284	50000	269.3		
1364774400	1011	31.5	80.07	10		60	16	264		1001.7						
1364774400	1015	17.83	69.6	14		190	11	272.2	83.7					269.8		
1364774400	1016	28.88	78.92	14		100	7	265.2	93.9	999.4		0.001964		264.4		
1364774400	1017	14.78	77.68	6		60	5	253.7		1001.9						
1364774400	1020	16.53	76.47	10		60	10	259.3		1000.8						
1364774400	1023	18.53	69.05	76		250	1	267.9	94.8	1007.3		0.002441	1800	267.2		
1364774400	1025	18.9	69.67	8		190	6	270.2	92.1	1005.8		0.0028	7000	269.1		
1364774400	1026	18.93	69.65	114	8	200	4	269.7	94.2	1006.1		0.002794	15000	268.9		
1364774400	1027	18.9	69.65	20				270.6	93.5					269.7		
1364774400	1028	19	74.52	16	2	150	3	261.1	83.6	1002.9		0.001258	30000	258.9		
1364774400	1033	19.5	70.23	24		230	4	271	86.1	1004.7		0.002787		269		
1364774400	1037	20.25	69.38	4		90	2	266.1	92.5	1005.8		0.002059		265.1		
1364774400	1043	22.82	70.4	10		260	4	271.1								
1364774400	1044	22.13	70.48	6		240	7	271.4	92.2	1003.3		0.00307		270.3		
1364774400	1046	20.95	69.78	4		290	2	269.2	74.8	1005.2		0.00211		265.4		
1364774400	1049	23.35	69.97	2		120	3	264	81.3	1004.3		0.001538		261.4		
1364774400	1052	23.67	70.67	79		230	6	270.3	96.3	1002		0.002988		269.8		
1364774400	1055	23.97	71.08	14		250	13	273	64.8	1000.1		0.002439		267.2		
1364774400	1058	23.53	69.57	381		130	2	255.5	83.5	1007.4		0.000826...		253.4		

R-tools

- Read from SQLite data files or from spatial fields
- Calculate scores
- Write to SQlite results files



- Read from SQLite results files
- Plot



R-tools

So far:

EPS

- Prototype version providing basic scores
- Plotting routines
- Reading / writing to SQLite

Spatial

- Prototype version for FSS and SAL
- Development version for plotting routines
- SQLite output, reading from several radar formats, INCA,

SQLite results

Aggregated results

• Overall scores for each model and region (one table per sub-domain). Different tables for scores with different dimensions.

Ensemble / Model	Lead time	Score 1	Score2	….	...
mname (character)	Leadtime(int)	(float)	(float)		
…					

Disaggregated results

• Scores are calculated per validation time and lead time. It is for some scores possible to aggregate overall scores from this tables.

• Enable calculation of scores for changing regions, domains, lead-times, periods ...

• Scores can be extracted directly via query for selected stations, lead times ...

Station id	Ensemble / Model	fc.date	Lead.time	Score 1	Score 2
sid (int)	mname (character)	fcdate (integer)	leadtime (integer)	(float)	(float)

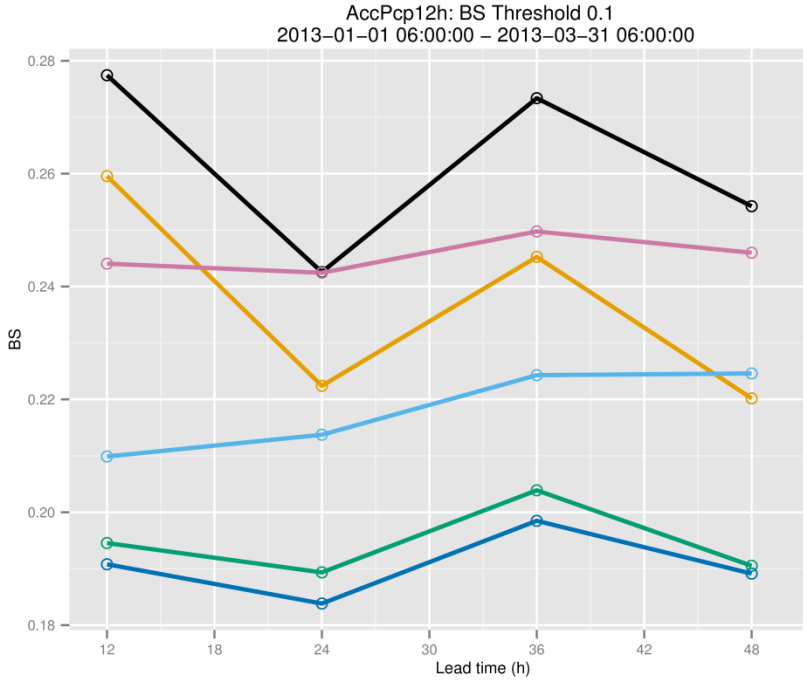
SQLite results (spatial)

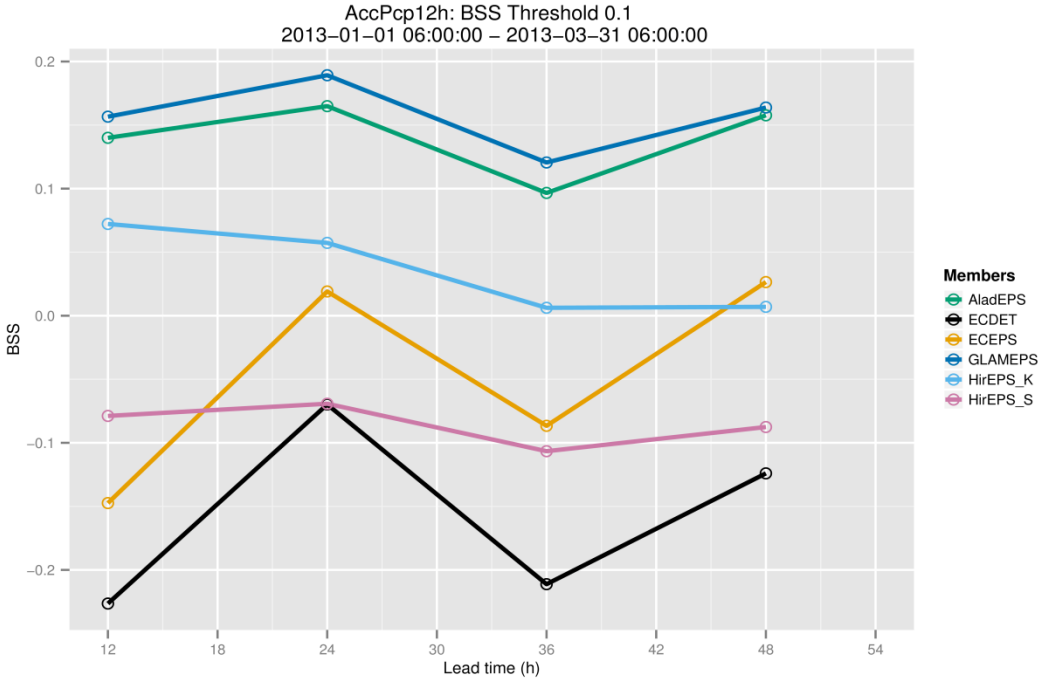
SELECT * FROM stats

SQL ausführen | Aktionen | Letzter Fehler: not an error

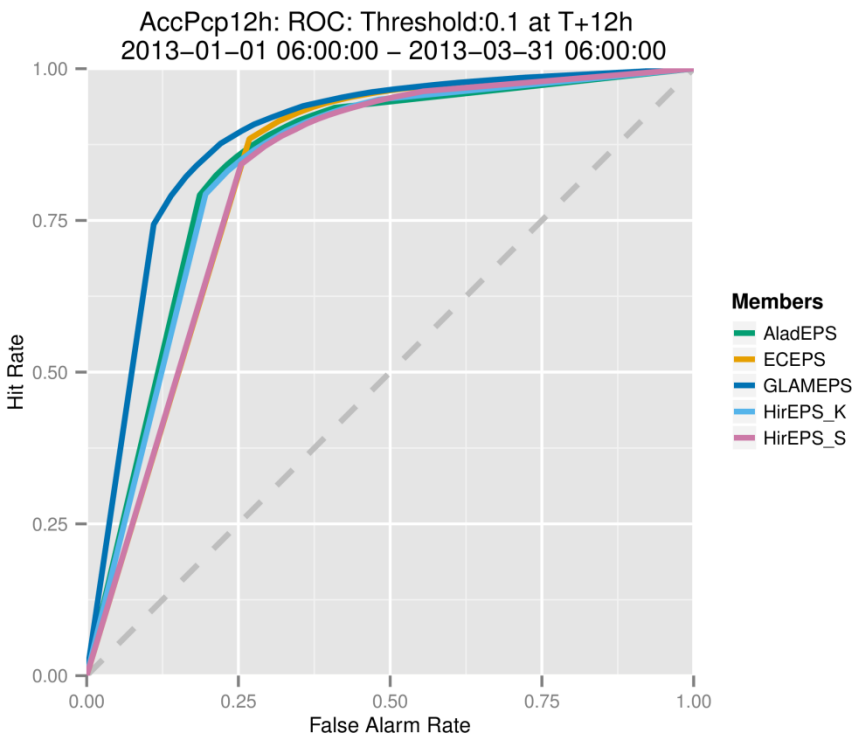
date	leadtime	threshold	nbpts	baserate	bias	mse	ets	hk	fss
1359687600	3	0.1	1	0	0.353755555555...	68106604.91330...	-0.023911938248...	0	0.270578438280
1359687600	3	0.3	1	0	0.353755555555...	68106604.91330...	-0.014775234334...	0	0.212940651494
1359687600	3	1	1	0	0.353755555555...	68106604.91330...	-0.000036047075...	0	0.054626028760
1359687600	3	3	1	0	0.353755555555...	68106604.91330...	0.000042915672...	0	0.011700053182
1359687600	3	0.1	3	0	0.353755555555...	68106604.91330...	-0.022594020707...	0	0.302009742117
1359687600	3	0.3	3	0	0.353755555555...	68106604.91330...	-0.015235583898...	0	0.240330675946
1359687600	3	1	3	0	0.353755555555...	68106604.91330...	0.000415387331...	0	0.062704143351
1359687600	3	3	3	0	0.353755555555...	68106604.91330...	0.000552570178...	0	0.011337717059
1359687600	3	0.1	15	0	0.353755555555...	68106604.91330...	0.016668765676...	0	0.384897665169
1359687600	3	0.3	15	0	0.353755555555...	68106604.91330...	0.019608897972...	0	0.319181716818
1359687600	3	1	15	0	0.353755555555...	68106604.91330...	0.029103585754...	0	0.111844038780
1359687600	3	3	15	0	0.353755555555...	68106604.91330...	0.009354066884...	0	0.011379605227
1359687600	3	0.1	49	0	0.353755555555...	68106604.91330...	0.068793494005...	0	0.500830525043
1359687600	3	0.3	49	0	0.353755555555...	68106604.91330...	0.074501185120...	0	0.452073514141
1359687600	3	1	49	0	0.353755555555...	68106604.91330...	0.078996886154...	0	0.226470424844
1359687600	3	3	49	0	0.353755555555...	68106604.91330...	0.042014584666...	0	0.014063126024
1359687600	3	0.1	225	0	0.353755555555...	68106604.91330...	0.111040061223...	0	0.766232840350
1359687600	3	0.3	225	0	0.353755555555...	68106604.91330...	0.097623249380...	0	0.772002174414
1359687600	3	1	225	0	0.353755555555...	68106604.91330...	0.091339020558...	0	0.701200059533
1359687600	3	3	225	0	0.353755555555...	68106604.91330...	0.110636712777...	0	0.116465620451
1359698400	6	0.1	1	0	0.515333333333...	68107318.64632...	-0.137276808472...	0	0.605442549847
1359698400	6	0.3	1	0	0.515333333333...	68107318.64632...	-0.122692190793...	0	0.611260053619
1359698400	6	1	1	0	0.515333333333...	68107318.64632...	-0.037991482582...	0	0.374994345682
1359698400	6	3	1	0	0.515333333333...	68107318.64632...	0.000187509404...	0	0.017862549197
1359698400	6	0.1	3	0	0.515333333333...	68107318.64632...	-0.116006077444...	0	0.648591387783
1359698400	6	0.3	3	0	0.515333333333...	68107318.64632...	-0.108707937711...	0	0.661047043609

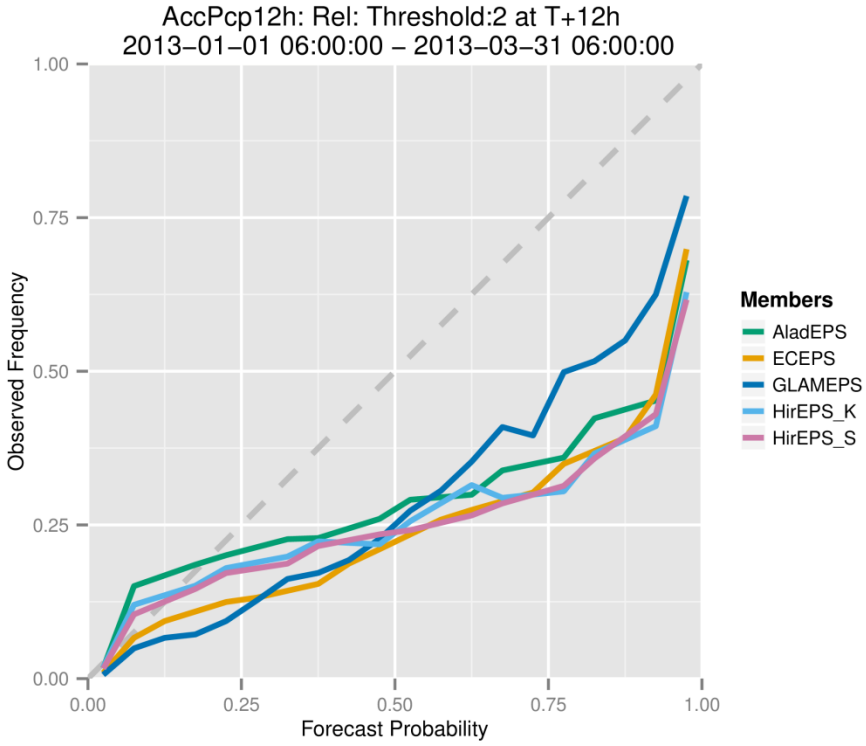
visualization



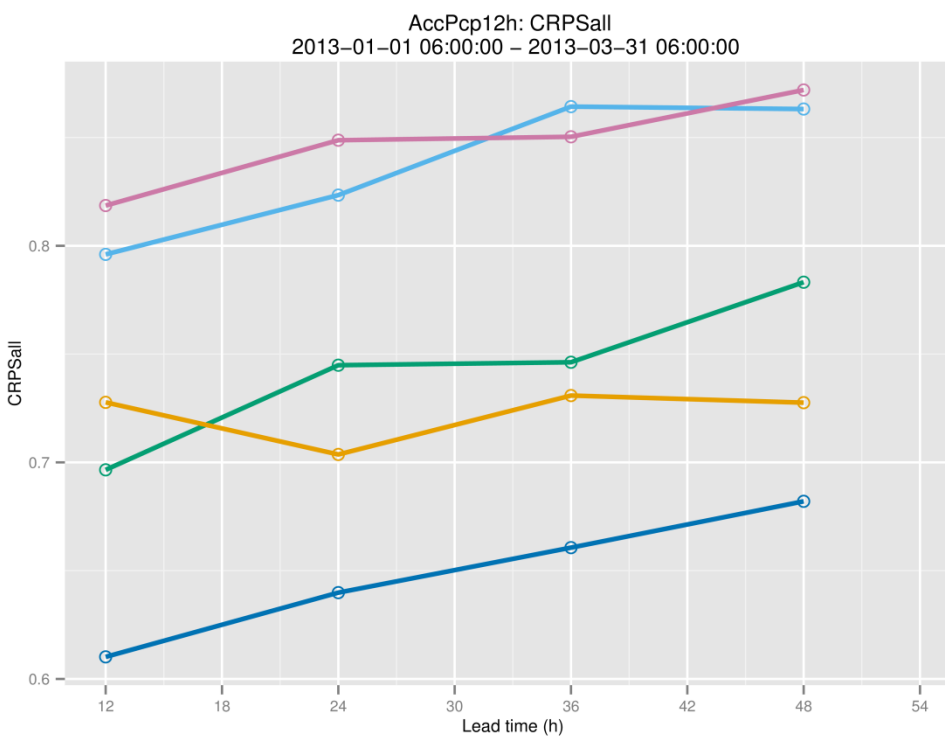


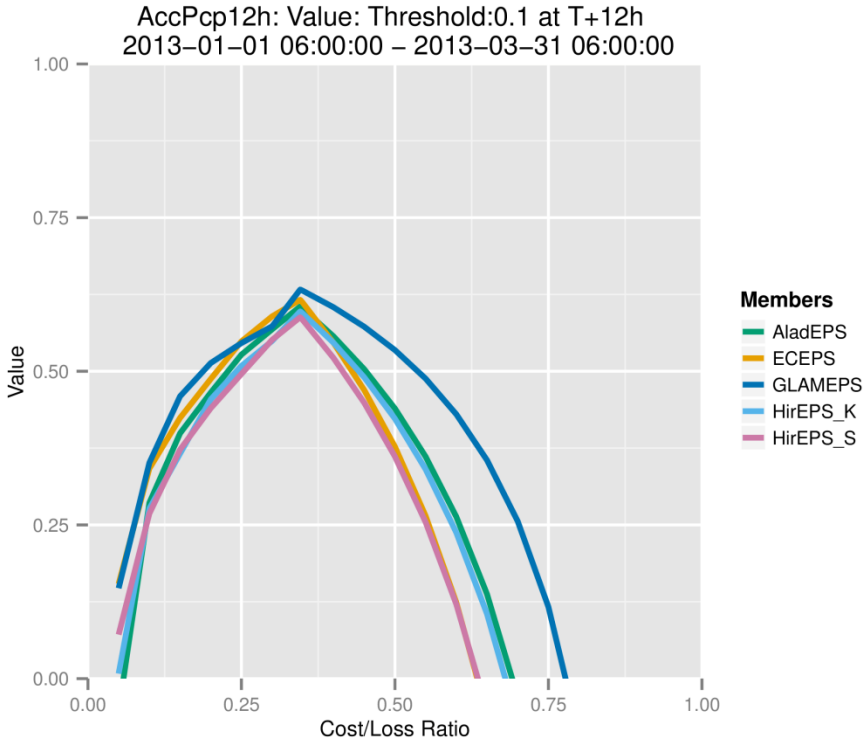
visualization



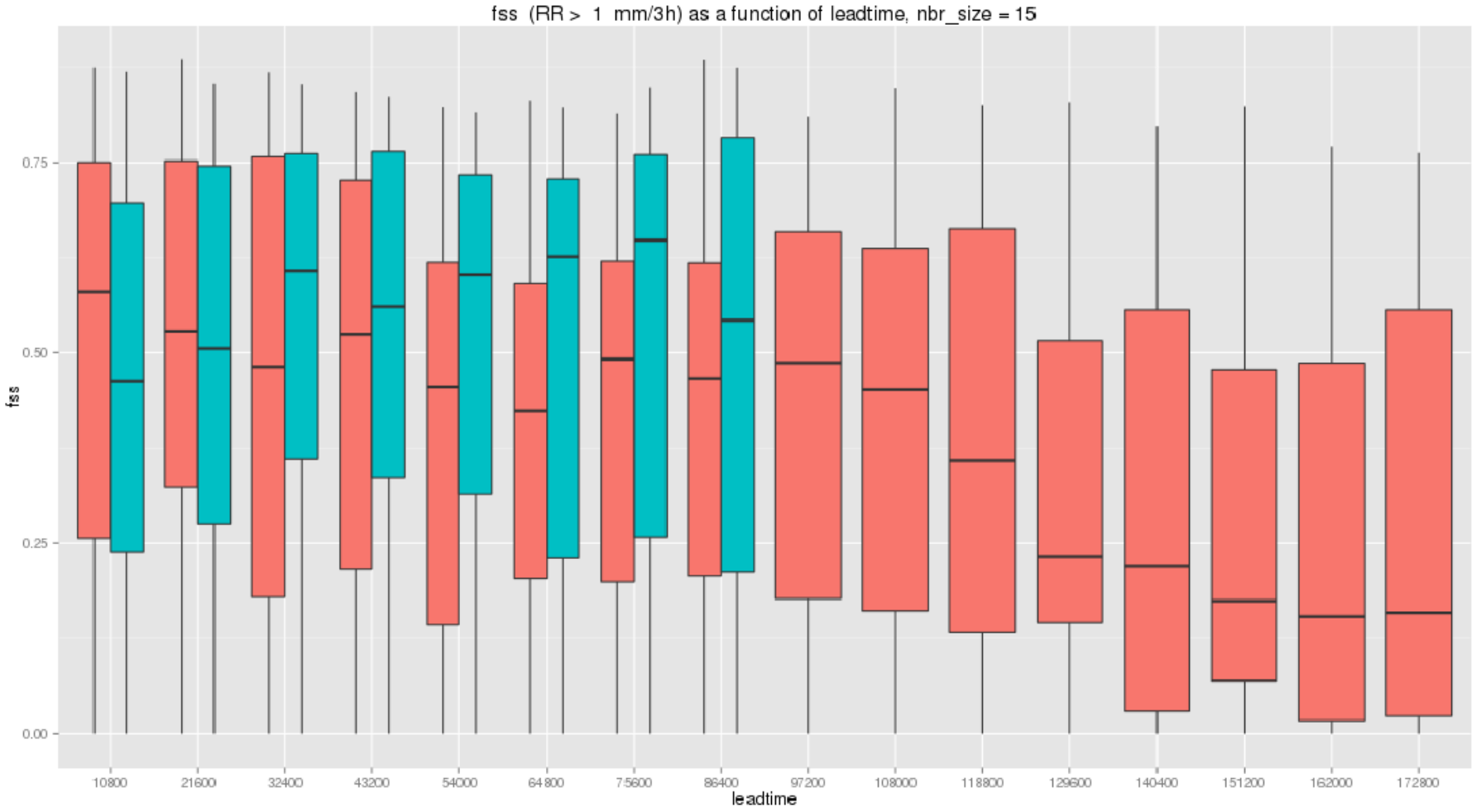


visualization





visulization



What we achieved

- Structure of framework
  - Git repository for Harp and Harp_sample (example data)
- General database structure for input and output data
- Prototype version for EPS verification
  - Simple to implement locally
- Possibility to run spatial verification FSS
- Production of graphics is externalized
  - Verification results stored in database
  - Vizualisation scripts for plotting

Plans

- Move from prototype to version 1.0
  - EPS:
    - Implementing reading and writing from SQLite files
    - Cosmetics to the plotting tools
    - Example data (Harp_sample)
  - Spatial verification
    - Util‘s for different input data types
    - Develop output format for verification results
    - Plotting routines
  - Documentation
- Explore options for user interface
  - Results to web-platform

THANKS