What's happening in harp

Andrew Singleton MET Norway

Brief Overview

harp is a set of R packages for reading, analysing and visualizing NWP data

- **harpIO** for reading and writing data
- **harpPoint** for analysing point data (verification)
- **harpSpatial** for analysing spatial data (verification)
- **harpVis** for visualization
- **harp** attaches everything

Initial thinking

- Point data and spatial data are very different...
  - Point data are stored as vectors
  - Spatial data are stored as (multi dimensional) arrays
- It makes sense to have separate packages

New thinking

- Point data and spatial data can exist as columns in a data frame
  - Point data as a vector
  - Spatial data as a list of 2d «geofields»
    - «geolist» class with many basic mathematical operations
  - All data frame operations can be done on spatial data
    - Mutating
    - Filtering by other columns
    - Grouping and summarizing
    - Pivoting

Refactored packages

- Many methods for point data now work spatial data as well so it no longer makes sense to have them in separate packages
  - **harpCore** - class definitions and methods, common operations and math
  - **harpIO** - as before
  - **harpVerif** - point and spatial verification
  - **harpVis** - as before (web app part may be abstracted out into a separate package)
  - **harp** - attaches everything + wrapper scripts read › operation › (write)
    - "automated " verification
    - Score cards

Upcoming changes with user impact

- read_<...> functions have new date_times argument
- The Old Way

```
read_forecast(
  start_date = 2022040400,
  end_date   = 2022040818,
  …,
  by = "6h"
)
```

Upcoming changes with user impact

- read_<...> functions have new date_times argument
- The Old Way

```
read_forecast(
  2022040400,
  2022040818,
  …,
  by = "6h"
)
```

Upcoming changes with user impact

- read_<...> functions have new date_times argument
- The New Way

```
read_forecast(
  date_times = seq_dates(2022040400, 2022040818, by = "6h")
  …,
)
```

Upcoming changes with user impact

- read_<...> functions have new date_times argument
- The New Way

```
read_forecast(
  seq_dates(2022040400, 2022040818, by = "6h")
  …,
)
```

Upcoming changes with user impact

- read_<...> functions have new date_times argument
- The New Way

```
read_forecast(
  seq_dates(2022040400, 2022040818, by = "6h")
  …,
)
```

- Allows for specification of irregularly spaced date-times
- No longer need to supply the same date-time twice for only one date-time

Upcoming changes with user impact

New parameter definitions (*not linked to FA yet)

```
t2m = list(
  description = "Air temperature at 2m above the ground",
  min = 223,
  max = 333,
  grib = list(
    name       = c("2t", "t"),
    level_type = c("heightAboveGround", "surface"),
    level      = 2
  ),
  netcdf = list(
    name = "air_temperature_2m"
  ),
  v = list(
    harp_param  = "T2m",
    name        = "TT",
    param_units = "K",
    type        = "SYNOP"
  ),
  wrf = list(
    name = "T2"
  ),
  fa = list(
    name  = pad_string("CLSTEMPERATURE", 16),
    units = "K"
  ),
  obsoul = list(
    name      = 39,
    units     = "K",
    harp_name = "T2m"
  )
)
```

Upcoming changes with user impact

New parameter definitions with possibility to apply a function at read time

```
ws10m = list(
  description = paste(
    "Wind speed at 10m above the ground calculated",
    "from U and V winds at 10m above the ground"
  ),
  .
  .
  .
  netcdf = list(
    name = list(u = "x_wind_10m", v = "y_wind_10m")
  ),
  .
  .
  .
  func = function(u, v) sqrt(u ^ 2 + v ^ 2)
)
```

Upcoming changes with user impact

New read functions

- **`read_obs()`** - replaces **`read_obs_convert()`** and more consistent with **`read_forecast()`**
- **`read_obs_file()`** - reads a single observations file (like **`read_grid()`** but for observations)
- **`read_analysis()`** - reads gridded analyses

Upcoming changes with user impact

New transformations

`"xsection"`	Vertical cross section between two user defined points (in long lat)
`"subgrid"`	Subset of gridded from user defined (i, j) locations of corners
*`"zoom"`	Zoom into a grid from user defined grid centre (long, lat) and radius of grid squares
*`"upscale"`	Upscale the grid by a user defined integer factor

Upcoming changes with user impact

Sub hourly lead times

- Specify lead times in different units of time in read functions
  - *Only tested for netcdf so far

Upcoming changes with user impact

*Faster score computation

- Huge speed up for bootstrapping

*Verification of cyclic variables (e.g. wind direction)

- Not entirely clear what to do for some scores

Upcoming changes with user impact

Compatibility with ggplot2 for 2d fields

- New geoms for «geolist» columns
  - **`geom_georaster()`** with data thinning for faster plotting
  - **`geom_geocontour()`**
  - **`*geom_geovector()`**
- New coordinates - will automatically add map background / overlay
  - **`**coord_geo()`**
- New plot theme for gridded data
  - **`theme_harp_map()`**

Upcoming changes with user impact

Compatibility with ggplot2 for 2d fields

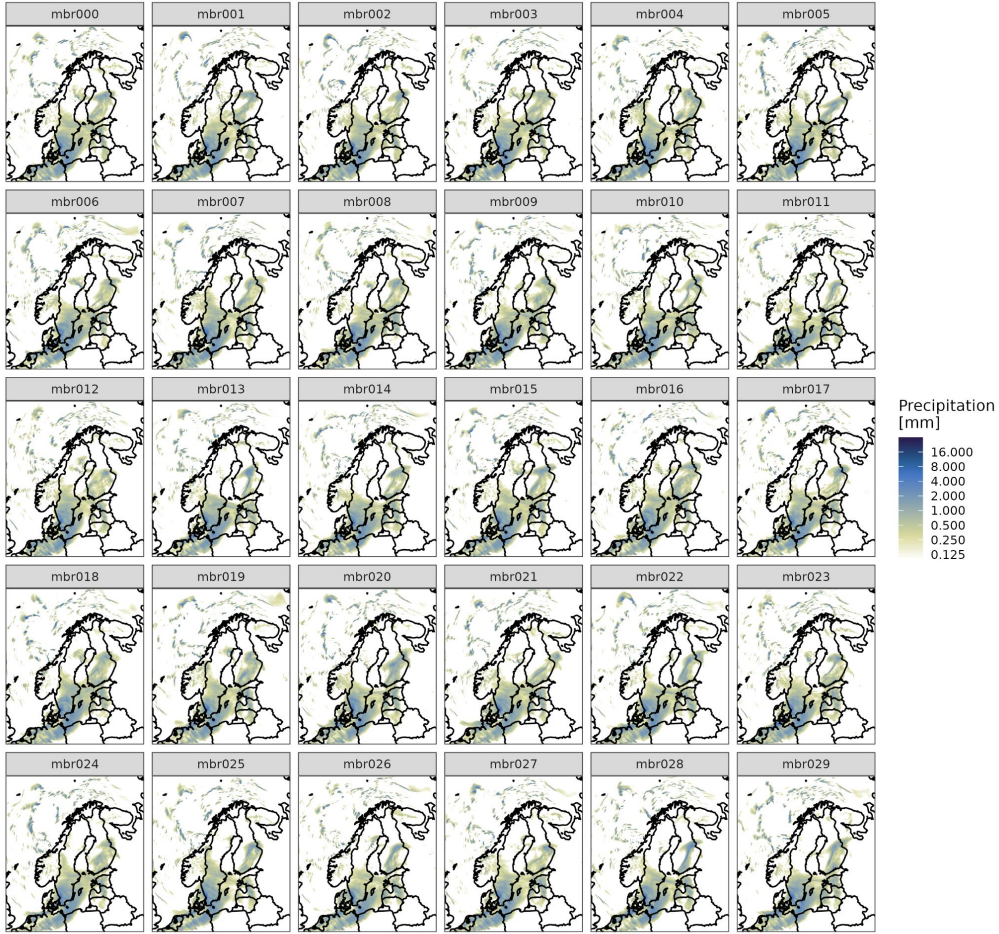
- New colour scales
  - `*scale_fill_precip()`
  - `*scale_fill_temp()`
  - `*scale_fill_wind()`

- *Data thinning to speed up raster plotting

Upcoming changes with user impact

Postage stamp plot example

```
ggplot(meps) +
  geom_georaster(aes(geofield = forecast)) +
  scale_fill_precip() +
  facet_wrap(vars(member)) +
  coord_geo(add_map = TRUE) +
  theme_harp_map() +
  labs(fill = "Precipitation\n[mm]")
```


mbr000
mbr001
mbr002
mbr003
mbr004
mbr005
mbr006
mbr007
mbr008
mbr009
mbr010
mbr011
mbr012
mbr013
mbr014
mbr015
mbr016
mbr017
mbr018
mbr019
mbr020
mbr021
mbr022
mbr023
mbr024
mbr025
mbr026
mbr027
mbr028
mbr029
Precipitation [mm]
16.000
8.000
4.000
2.000
1.000
0.500
0.250
0.125

Upcoming changes with user impact

Better handling of groups in web app

- Group dropdowns are automatically generated from the data
- Selection from each group should now influence what can be chosen from other groups
- *Different x-axis data (e.g. valid date-time / lead time) to be detected and a radio button created

*Automatic vertical profile plotting in web app

- When vertical profile verification is detected plots will automatically switch to having vertical level in the y axis and facetting by lead time / valid date-time

Time scale of new releases

- Aiming for end May 2022
  - Only items marked with an asterisk are not fully implemented in development versions
  - All packages will be tagged and become more stable
  - Development versions will be easily installable / forkable

Longer term plans

- Fractions skill score for ensembles (already in testing)
- Fractions skill score using point observations and gridded forecasts (following UK Met Office)
- **harpSat** package for dealing with satellite data***
  - Resample swath data to model grid
  - Resample gridded data to swath
- On the fly scores using web app
- Wind into / along cross sections***
- Cross sections for long-lat projections following great circles***
- Spatial verification coding weeks in Autumn

(Default verification set for ACCORD)

***Seeking collaborators