SMHI Gridded Climatology

Appendix C: Documentation of scripts

1 Fetching of observations

Observations need to be collected from different sources. All scripts are independent of each other and can run in parallel.

Note that local observations are also needed from MET Norway and FMI via personal communication.

1.1 Fetching from MORA

Directory:
`/nobackup/smhid17/proj/sik/Refdata_Scripts/Observation_Preprocessing/Fetching_MORA/`

Script:
`Get_Local_Obs_monthly_v1.1.py`

reads the station list from an external file:
`Klimat.txt`

Execution:
```
python Get_Local_Obs_monthly_v1.1.py
```

The script was executed on the elin4-server. It needs to be investigated if it possible to perform the fetching on Bi.

1.2 Fetching from MARS

Observations from ECMWF's archiving system are fetched for different sources depending on the period following the advice by ECMWF.
The scripts were executed on ECMWF's HPC system on ecgate. It needs to be checked if scripts can be executed on Bi.

Directory:
`/nobackup/smhid17/proj/sik/Refdata_Scripts/Observation_Preprocessing/Fetching_MARS`

Scripts:
```
Referens_observations_OD_DCDA_YYYY_Local_LatLon
Referens_observations_OD_e4_YYYY_Local_LatLon
Referens_observations_OD_OPER_YYYY1_Local_LatLon
Referens_observations_OD_OPER_YYYY_Local_LatLon
```

Additional input file with filter rules:
`bufr_filter_rules_LocalLatLon`

Execution as batch-job on ecgate, e.g.:
```
sbatch Referens_observations_OD_DCDA_YYYY_Local_LatLon
```

2 Preparation of observations

2.1 BUFR to ASCII conversion

The BUFR output created by "Fetching from MARS" described above needs to be converted to ASCII for input to TITAN processing. The scripts used for this are located in this folder:

`/nobackup/smhid17/proj/sik/Refdata_Scripts/Observation_Preprocessing/BUFR2ASCII`

There is one master script each for each MARS data stream, `E4_oper, OD_dcda, OD_oper`, respectively. These master scripts call the python script `bufr2titan_month_extract.py` for the actual BUFR2ASCII conversion. The python script requires that the `eccodes` python library is installed. For the Referensdata project this was solved by a local python3 installation on bi under directory

`/nobackup/smhid17/users/sm_psamu/ANACONDA/anaconda3/envs/gridpp/`

The scripts described here are not prepared to be executed as they are since paths to BUFR files and output paths need modifications. The ASCII output prepared by these scripts is now available here

`/nobackup/smhid17/proj/sik/refdata/observations/BUFR_EXTRACT`

2.2 SYNOP observations

The final setup of SYNOP observations from Sweden, Finland and Norway is the result of an iterative procedure where problematic deliveries, for different reasons, were replaced by new ones. Here we do not document this iterative procedure but simply refer to the final versions used for the Referensdata project. The final versions are available here

`/nobackup/smhid17/proj/sik/refdata/observations`

More specifically in the subfolders `Finland, MORA_1.1` (for Sweden) and `Norway`

2.3 TITAN quality control

The scripts for the TITAN quality control are in the folder

`/nobackup/smhid17/proj/sik/Refdata_Scripts/TITAN_quality_control`

The scripts assume that all observations needed are available as ASCII files and that the data in these files are organised in a precise way. All observation input data used for the Referensdata project and prepared in the format for TITAN quality control is stored in the folder

`/nobackup/smhid17/proj/sik/refdata/observations`

More specifically in the subfolders `BUFR_EXTRACT, Finland, MORA_1.1, Norway`

The master script for TITAN quality control is

`master_TITAN.sh`

This master script is submitted as a sbatch job and is prepared to be executed in this specific environment. The master script and its underscripts call

```
TITAN_t2m_td2m_rr_sd.sh
TITAN_tn_tx.sh
mandtg
rearrange_TITAN_output.sh
TITAN_R_scripts/TITAN/titan_mod.R
TITAN_R_scripts/TITAN_new_buddy/titan.R
```

Here, all the `*.sh` scripts are constructed for the purpose of this project and `mandtg` is a HARMONIE-AROME system script for managing conversion of date and time. The R-scripts under `TITAN_R_scripts` are based on MetNorway GitHub releases of TITAN in repository https://github.com/metno/TITAN.git. `TITAN/titan_mod.R` is a slightly modified version of commit 1b920c7e80eca in the master branch. The modification just adds additional output in the

log file. `TITAN_new_buddy/titan.R` is based on commit 6e52095a38ce4da in the new_buddy branch. This version includes buddy check modification recommended by Cristian Lussana (MetNorway) applied for snow depth (sd) and precipitation (rr).

A Digital Elevation Map (DEM) is used for the TITAN processing. The DEM used and needed is provided as NetCDF file in the TITAN quality control folder. This DEM is produced by a combination of SURFEX and python processing and the scripts used for that are available here:

```
/nobackup/smhid17/proj/sik/Refdata_Scripts/TITAN_quality_control/DigitalElevationMap
```

The python script is needed to convert SURFEX NetCDF output to the input format required by TITAN.

3 Downscaling the first guess

The scripts for downscaling the first guess are in the folder

```
/nobackup/smhid17/proj/sik/Refdata_Scripts/Downscaling_first_guess
```

There is one script for setting up the geometry of the SWECLIM area and then three script for each of the entities to be analysed. One for collecting data during the period for which UERRA and MEPS data overlap, a second one for estimating the coefficients for the downscaling based on these data and a third one for applying the downscaling to the UERRA files. There is one exception; the downscaling of the daily minimum and maximum temperatures are done based on the coefficients for t2m:

```
klimat_create_phys.py

klimat_collect_arome_aladin_t2m.py
klimat_estimate_downscale_params_t2m.py
klimat_downscale_t2m.py
klimat_downscale_tnx.py

limat_collect_arome_aladin_rh2m.py
klimat_estimate_downscale_params_d2m.py
klimat_downscale_d2m.py

klimat_collect_arome_aladin_rr24.py
klimat_estimate_downscale_params_rr24.py
klimat_downscale_rr24.py

klimat_collect_arome_aladin_snd.py
klimat_estimate_downscale_params_snd.py
klimat_downscale_snd.py
```

Note that the dew point temperature is not available from the NWP models so it has to be diagnosed from t2m and rh2m The downscaling is applied to td2m diagnosed from UERRA data for t2m and rh2m.

There are also a number of additional scripts for managing the submission of downscaling jobs to the slurm que system on the HPC at NSC:

```
klimat_submit_ds_t2m.sh (calls klimat_run_ds_t2m.sh)
klimat_submit_ds_tnx.sh (calls klimat_run_ds_tnx.sh)
klimat_submit_ds_d2m.sh (calls klimat_run_ds_d2m.sh)
klimat_submit_ds_rr24.sh (calls klimat_run_ds_rr24.sh)
klimat_submit_ds_snd.sh (calls klimat_run_ds_snd.sh)
```

4 Preparing and performing the gridpp analysis

Scripts for preparing and performing the gridpp analyses are located here:

```
/nobackup/smhid17/proj/sik/Refdata_Scripts/Gridpp_analysis
```

There are three scripts for each of the entities - one for optimizing the gridpp parameters (per decade, season and possible also time of day), one for collecting the optimization results and

finding expressions for yearly interpolation between these results and finally one for executing the analysis based on these parameters. There are two exceptions; the parameters for t2m are also used for tn and tx and the “analysis” of rh2m is actually a diagnose of rh2m from the analyses of t2m and td2m:

```
klimat_opt_gridpp_t2m.py
klimat_collect_gridpp_opt_t2m.py
klimat_an_t2m.py
klimat_an_tnx.py

klimat_opt_gridpp_d2m.py
klimat_collect_gridpp_opt_d2m.py
klimat_an_d2m.py
klimat_an_rh2m.py

klimat_opt_gridpp_rr24.py
klimat_collect_gridpp_opt_rr24.py
klimat_an_rr24.py

klimat_opt_gridpp_snd.py
klimat_collect_gridpp_opt_snd.py
klimat_an_snd.py
```

There are also a number of additional scripts for managing the submission of the analysis jobs to the slurm que system on the HPC at NSC:

```
klimat_submit_an_t2m.sh (calls klimat_run_an_t2m.sh)
klimat_submit_an_tnx.sh (calls klimat_run_an_tnx.sh)
klimat_submit_an_d2m.sh (calls klimat_run_an_d2m.sh)
klimat_submit_an_rr24.sh (calls klimat_run_an_rr24.sh)
klimat_submit_an_snd.sh (calls klimat_run_an_snd.sh)
```

5 Converting GRIB to CF compliant NetCDF

The scripts for converting the GRIB files output from the gridpp analysis to CF (Climate and Forecast) compliant NetCDF files are found in this folder:

```
/nobackup/smhid17/proj/sik/Refdata_Scripts/GRIB_to_NetCDF
```

There is one generic script for converting GRIB to NetCDF and then one script for each parameter that set the some of the metadata according to CF standards. Finally there is one script for post processing these data and output the final dataset with the proper names and time averages:

```
klimat_grb2nc.sh

klimat_grb2nc_tas.sh
klimat_grb2nc_tnx.sh
klimat_grb2nc_hurs.sh
klimat_grb2nc_rr.sh
klimat_grb2nc_snd.sh

reformat_data.py
```