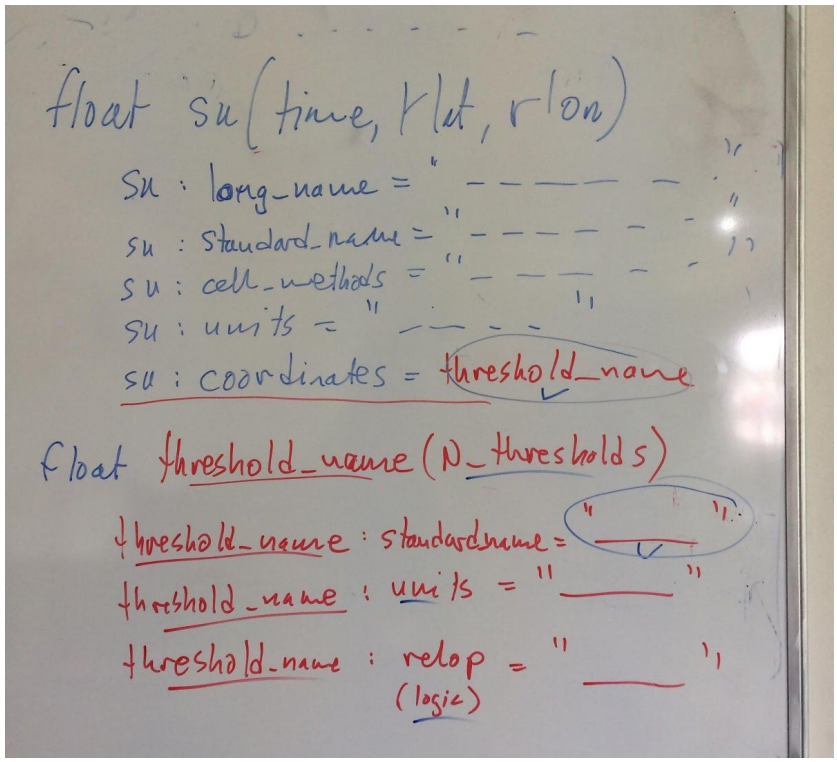
CLIPC ICCLIM 2nd coding sprint, Cerfacs 28 June -- 1 July 2016

Participants: Christian Pagé, Cerfacs, Martin Evaldsson, SMHI, Lars Bärring, SMHI, Milka Radojevic, Cerfacs (Thursday-Friday)

AGENDA --- MINUTES

1) Discussion/implementation for writing netcdf metadata from Json file definition
--- Tech discussion, select subset of indices to focus on in a first round
GOAL: Agree on way forward (see also point 6 and 8)
OUTCOME:

- ***DONE:*** Code for copying global metadata from input file to output file (works for FD, expected to work for all indices). For FD output variable metadata is read from json file and copied to output file (specific for each index).
- ***DONE:*** Implement handling of alternative variable names, in addition to the CMIP5/CORDEX ones (eg. for *tasmin*, also tasminAdjust, tn, tmin,mint)
- ***ONGOING:*** Save all input global and variable metadata in a character array variable (from point 1b).
- ***ONGOING:*** Use variable attribute json file input to verify input metadata.
- ***ONGOING:*** implement output of threshold variable (and metadata) for simple (constant) thresholds. Template for threshold variable is,



If N_thresholds > 1 then SU needs to be 4-d (time, rlat, rlon, N_thresholds). Details to be worked out.

1b) Provenance of global metadata
--- Suggestion to store all relevant input file metadata in a variable
GOAL: Discuss and agree on way forward
OUTCOME:

- Code reviewed and implementation will be done as part of work on global metadata under point 1.
- ***DONE***

2) Finalize testing on 360_day branch and describe the merge/release procedure we use for ICCLIM

--- Review and bring up to master

GOAL: Master brought up-to-date

OUTCOME:

- Testing done.
- One bugfix: user-specified date range not valid in 360_day calendar. We print a warning and truncate to day=30.
- Changes pushed and committed to 360_day branch.
- Fixed some opendap test cases.
- Confirmed and validated results on a regular calendar against master branch results.
- Merged and tagged/release 4.2.2
- Edited version number in files before commit
    - setup.py
    - ./icclim/icclim_doc/source/installation.rst
    - ./icclim/icclim_doc/source/conf.py
    - ./icclim/__init.py__
- Merge procedure (need to push changes to branch before doing it):
    - git checkout master
    - git merge “origin/360_day”
    - git commit -a
    - git push
    - Then do a release and tag on github
    - Documentation is updated automatically on readthedocs
- Github issue 12 closed.
- ***DONE***

3) Test cases (I)

There is a set of test cases committed in the 360_day branch as of yesterday, I've (Martin) basically tried to decouple the test cases from any actual code only using a simple wrapper script to execute each test case. For a full test suite a framework, such as (my favorite) the robot framework is needed to orchestrate and document the execution of all the test cases in one go.

--- discuss and review

--- regression tests ?

GOAL: Agree on test cases and framework

OUTCOME:

- Martin to look into the robot framework for top level testing (any support for this on github?)
- ***ONGOING***

4) Bug fix opendap chunking (also documented as issue by Maarten at KNMI)

See suggested solution in commit 1723977dc, related to github Issue 19

GOAL: Bug solved and issue closed

OUTCOME:

- Validated.
- Included in 360_day branch.
- Github issue 19 closed.
- ***DONE***

4b) Currently the OpenDAP time subsetting is not implemented.
GOAL: implement this
OUTCOME:

- It turned out that in fact OpenDAP subsetting works, and that is was a problem (mis-specification) in the test case. The test case was updated.
- ***DONE***

5) Question from Igor Stepanov (KNMI)
See attachment (index is 0 when all missing data: e.g. RR1, RX1day, R95p, PRCPTOT, ID, TX90p, FD)
--- update code
GOAL: Code updated and issue closed
OUTCOME:

- Christian to work on this
- ***ONGOING***

6) Solution to avoid hard coded variable names (tasmin/tasmax) in source code
--- Review and update code if necessary (cf. point 1 re. “known_variables”)
GOAL: Solve issue (from Maarten?) --- may depend on point 1
OUTCOME:

- Solved issue for hardcoded tasmin and tasmax in 360_days branch. Committed and pushed changes.
- ***DONE***

7) PEP8 code layout
Branch to follow Python standard code format (white spaces etc)
GOAL: Discuss and agree on style
OUTCOME:

- In principle the code should follow PEP8 code style standard, but be pragmatic and introduce this successively and update code style when updating the code
- For future coding we should aim to follow PEP8
- ***AGREED***

8) Discuss/implement Udunits (cfunits) support in ICCLIM for transparent handling of units and unit conversion
--- Will come up naturally when discussing point 1
GOAL: Depends on point 1; use cfunits, or find alternative strategy for handling units
OUTCOME:

- Cfunits will be used to handle units in input files. Special care is needed to handle model output (such as CMIP5 and CORDEX precipitation (pr) files) as they are providing `precipitation flux (kg m-2 s-1)`, which has to be converted to `lwe_precipitation_rate` (`m s-1`) because the units are formally incompatible.
- User-defined constants (and otherwise additional input) will be given together with a unit string that is compatible with cfunits.
- The output file will have human-readable units, eg. mm/day, degree Celsius, etc. (compliant with cfunits). Possible exception is “simple statistics” where the user might specify the units, or the output file should have the same units as the input file.
- In addition to the CF-compliant variable attribute “units”, the output file will have an additional variable attribute tentatively called “legend_units” that is defined in the index-definition json file. These may not always be compliant with cfunits, eg. “degree-days” instead of “degree_Celsius days”. ***DONE, see point 1***.

- ***ONGOING*:** Review code and update where necessary.
  Christian to check that internal calculations are in SI units and put appropriate comments in the code. Martin to implement cfunits at relevant places in the code.
    - Unit conversion should be done by a function call in icclim.py 4.2.2 line 408: currently has: var_units = getattr(inc.variables[v],'units') and below are a few conversion statements to remove in favor of cfunits calls.
    - Unit conversion currently is using var_scale and var_add to specify conversion factors, and they are applied after reading the data later. It is not using scale_factor and add_offset in the variable metadata directly, as library netcdf4-python is already doing it by default.
    - In the code, all temperatures are in Kelvin, except for user-defined thresholds which are expected in Celsius… That should internally all be converted to Kelvin at the beginning of the code, not within functions. Many index functions expect the threshold to be in Celsius… to be fixed.
    - Precipitation is converted to mm/day before calling the index calculation functions. I suggest to stick to mm/day...
    - Units of output variable is done on icclim.py 4.2.2 line 640 block: eval('set_longname_units.' + indice_name + '_setvarattr(ind)'). This is where we should set proper human-readable unit according to json file info and cfunits conversion.

9) Test cases II (to add)

The recommended set of minimal tests from our earlier discussion, not all are implemented,

su -- a simple temperature index
dtr -- and index involving two variables (hardcoded var names --- see point 10))
prcptot -- a precip index involving "wet days"
tx95t -- calculation of an annual percentile value (not implemented yet --- find alternative, or implement?)
tx90p -- proportion (%) days exceeding the reference 90 percentile

--- discuss and review
GOAL: Agree on a strategy/way forward
OUTCOME:

- ***ONGOING:*** Add a test case for high demand input data. Test is ongoing (Martin)

10) Performance of bootstrapping - I've got some measurements using lprof (https://github.com/rkern/line_profiler), probably similar to what you've already noted, that we could discuss
--- Not top prio, come up with a strategy and way forward, wait for Milka
GOAL: Discuss and agree on strategy.
OUTCOME:

- ***DONE:*** The discussion resulted in the following strategy:
- ***ONGOING:*** Implement a general option to allow all percentile-based thresholds to be calculated with or without bootstrapping.
- ***ONGOING:*** In relation to point 10b, implement the following functionality:
    - If the input_threshold_file argument is given then import (anch check) threshold data from this source.
    - If the output_threshold_file argument is given then calculate a thresholds for a range of percentile values based on the input data (argument for reference period is needed) and stored in the threshold file. Thresholds values for the specific percentile (of the requested index) is used in the calculations and stored in the index output file.
    - Either the input_threshold file argument or the output_threshold_file argument can be given, not both. If neither is given, only the relevant threshold is calculated and stored in the output index file.

- ***ONGOING:*** The performance issue is likely not directly related to the bootstrapping, but to the multi-level loop for selection of days to select data from. The probable critical lines have been identified. This issue will be revisited once the remaining work under this point have been concluded.

10b) Allow percentile thresholds to be imported rather than calculated from the same dataset, and implement saving the percentile thresholds in separate ('simple') netCDF files for later use.
GOAL: Discuss and come up with a strategy, possibly implement?
OUTCOME:

- ***DONE:*** Analyse which indices do use the same percentile threshold? Can the threshold variable be reused? --- YES
- ***ONGOING:*** Lars work on what metadata are needed if external threshold data is used?
- ***ONGOING:*** The CF convention does currently, CF-1.6 and CF-1.7 (draft), not allow this kind of spatially and seasonally varying thresholds. Lars to explore various alternatives, including organising a dedicated workshop.

11) To what extent rely on external software for verification (R-package)?
GOAL: Discuss
OUTCOME:

- Not needed at this stage. Relevant test cases will be developed (point 3 and 9). May come up at a later stage in relation to point 15.
- ***DONE***

13) Bootstrapping option for wsdi, csdi type indices ???
GOAL: Look into the code and see if/where a general option for using or bypassing bootstrapping when calculating percentile thresholds. The option will be set by the user.
OUTCOME:

- Refer to point 10 and 10b.
- ***DONE***

14) Generate file name from global metadata (2nd prio….)
GOAL: Would be nice, but is not priority now.
OUTCOME:

- ***Revisit at later stage.***

15) Initiate joint activity with rclimdex/climpact developers for cross-validation of software development
GOAL: Discuss and agree on strategy.
OUTCOME:

- Good idea but we need to make more progress before this becomes useful.
- ***Revisit at later stage.***

Version 4.2.3 of icclim released on <http://icclim.readthedocs.io/en/latest/>