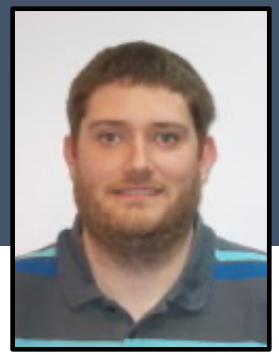
Technical discussion

Hannah Christensen

Massive thanks to Andrew Dawson (ECMWF, previously U. Oxford)



Aims for today

1. Input data and access
2. Scripts to facilitate workflow
   - Produced by Andrew Dawson for a previous project
     - not yet updated for the IFS MUMIP runs
   - High level overview of the two python packages available
   - Discuss workflow using these scripts (as I used them for the previous project)

Input data

- Available data:
  - Indian Ocean region, one 24-hour period (8 timestamps)
  - + a single ‘onecol’ file – all timestamps at one lat-lon point
  - erased all land points, and interpolated from neighbouring sea
    - binary field: “land_erase_flag” identifies these points

- https://mumip.web.ox.ac.uk

Available code

- “scmtiles”: Python software to deploy many independent SCMs over a domain. https://github.com/aopp-pred/scmtiles
- “openifs-scmtiles”: Python software to deploy the OpenIFS SCM using scmtiles. https://github.com/aopp-pred/openifs-scmtiles

- https://mumip.web.ox.ac.uk

scmtiles

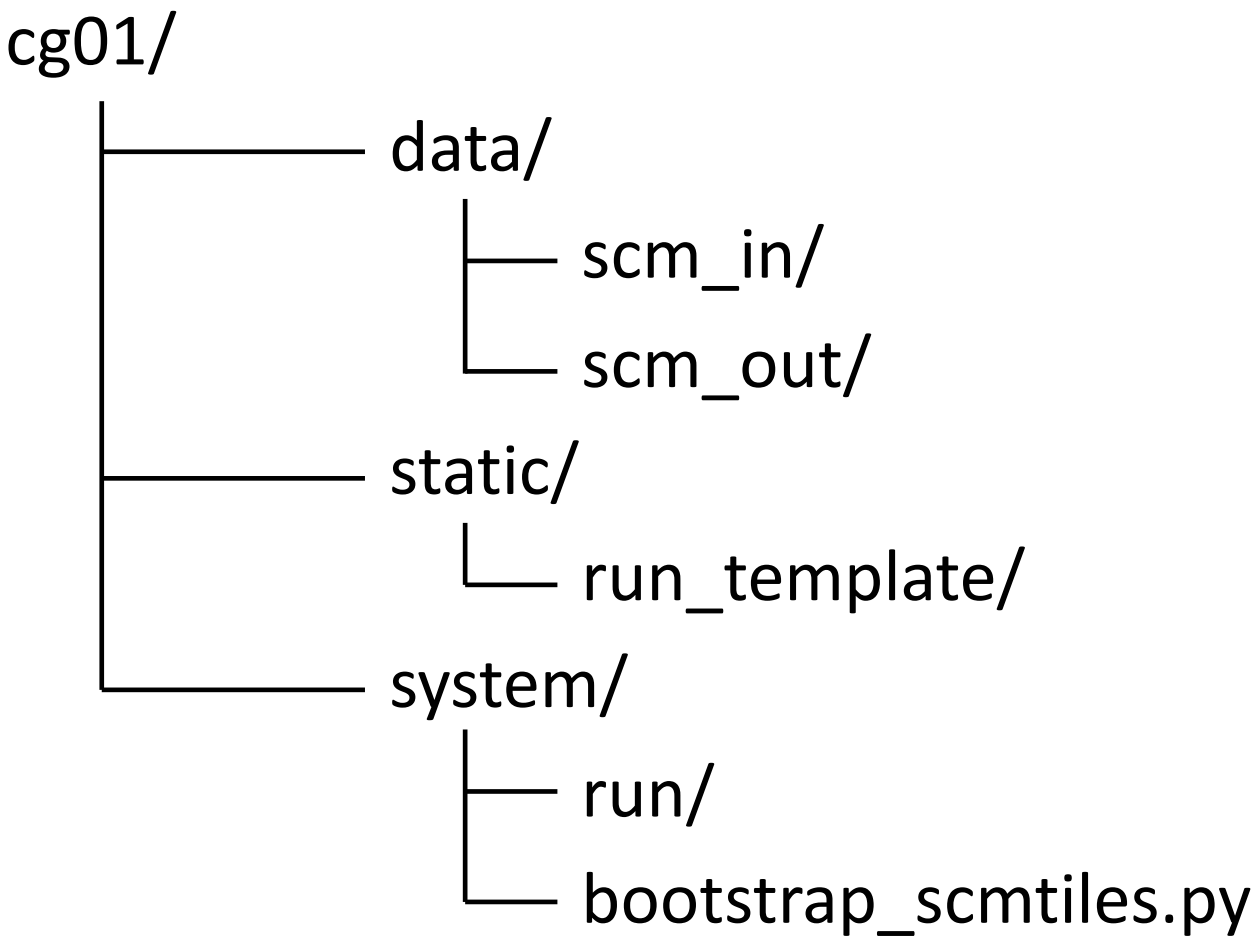
- Framework for running an arbitrary single-column model over a grid.
- It provides high-level task organisation and parallelisation for managing many thousands (or more) of model runs.
- Must write your own runner class, a subclass ofscmtiles.runner.TileRunner that implements the run_cell() method.
- The run_cell() method contains all the specific logic and operations required to run a particular SCM at a single location in space.
- As an example, openifs-scmtiles implements a tile runner for the OpenIFS SCM.

openifs-scmtiles

- Layer on top of scmtiles for running IFS SCM over a grid
- Can be used as an example to help deploy other SCM

- Each IFS SCM run requires its own directory containing:
  - input files and assorted reference files
  - namelist with user defined flags
  - SCM executable
- For each cell, openifs-scmtiles:
  - creates and populates this directory using scmtiles
  - creates the input file, including setting up time variable
  - runs the SCM and archives the results (MPI parallelisation)
- Separate post processing (pp) task:
  - Drops any variables not required
  - Stitches everything together into a single lat-lon output file

My setup 1



To do:

1. Create data directory structure, and move provided files to scm_in/
2. Create static/ directory
   - contains template for SCM directory
3. Create system/ directory
   - Run bootstrap_scmtiles.py in system/ to create system/run/ directory
   - Installs python environment, clones scmtiles and openifs-scmtiles

openifs-scmtiles scripts

	Run the SCMs	**Postprocessing**	**clean up**
User interface	**u_model_pp.sh** **template.cfg**	u_model_pp.sh **u_pp.sh** dropvars.txt	u_model_pp.sh u_pp.sh **u_clean.sh**
Driver (user must set paths)	run.sh	pp.sh	clean.sh
Main	openifs_scm_main.py openifs_scm.py	openifs_pp_main.py	

openifs_scm.py — TileRunner class for OpenIFS SCM

template.cfg — template config file to populate

Running SCM

```
openifs-scmtiles]$ ./u_model_pp.sh run_id 1-10
```

u_model_pp.sh calls **run.sh** to set SCM model runs going
then calls **pp.sh** to do postprocessing

- for time indices 1-10
- run.sh converts time index into date string for config file

run_id will appear in saved config and log files along with time index

One command has launched thousands of SCM to tile the domain, for the first ten initial times!!

Postprocessing and cleaning

```
openifs-scmtiles]$ ./u_pp.sh run_id_2 1-10
openifs-scmtiles]$ ./u_clean.sh run_id_3 1-10
```

Sometimes a SCM tile will not complete

- fails during SCM run? try u_model_pp again
- fails during post processing? u_pp
- fails during clean up? u_clean