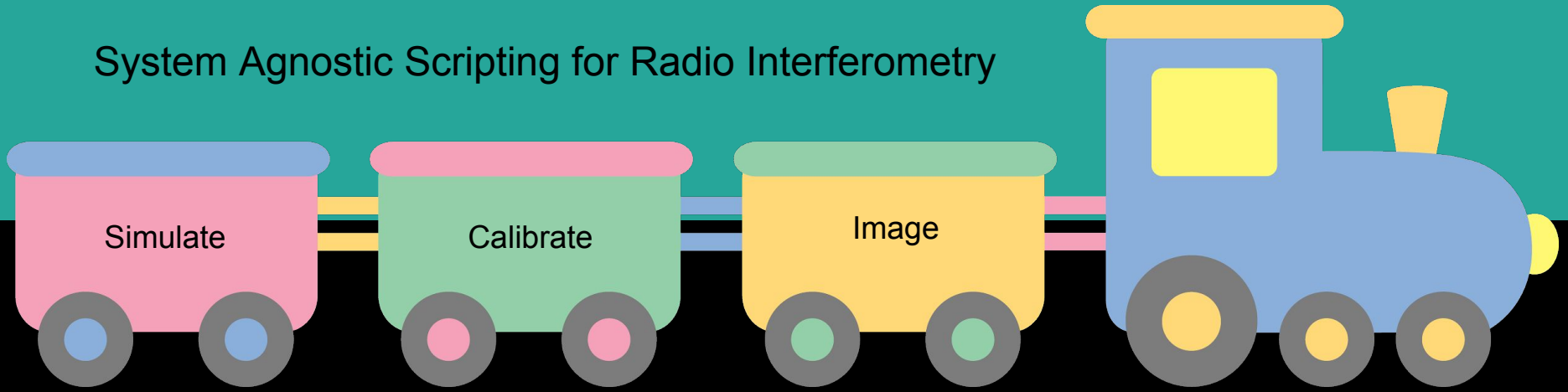
System Agnostic Scripting for Radio Interferometry



stimela

Sphesihle Makhathini
SKA SA, Rhodes University
smakhathini@ska.a.c.za

RHODES UNIVERSITY
Where leaders learn

SKA

The data is bigly

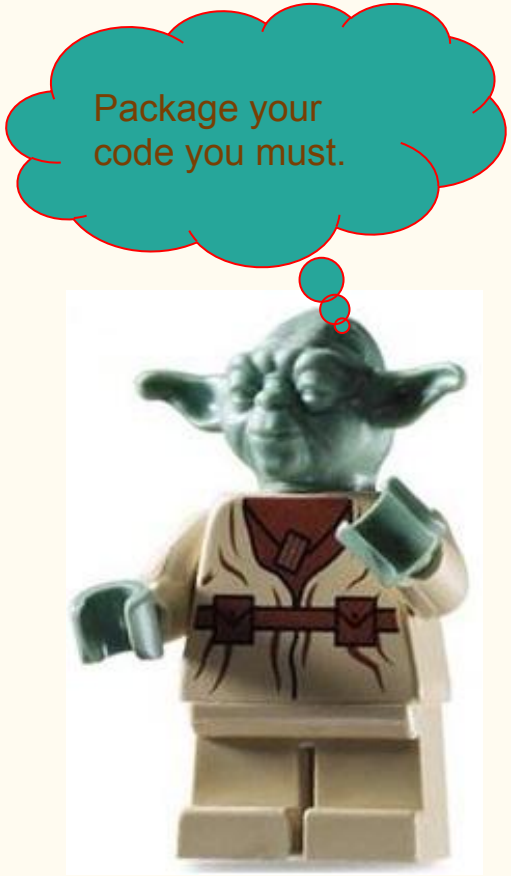
- More data than we can process, with a lot more on the way
- Instruments and sky more complex
- Data reduction, analysis and pipelining tools increasingly more complex
  a. Poorly documented
  b. Unstable
  c. Difficult to build/install
- Astronomers write bad code (#NotAllAstromers)
  a. It needs to work for this case, so I can get my PhD/paper result
- Increasingly difficult to reproduce science data products

<u>To meet expectations of modern instruments</u>

- Flexible, easy to use, robust, scalable, easily portable workflow management tools

Packaging

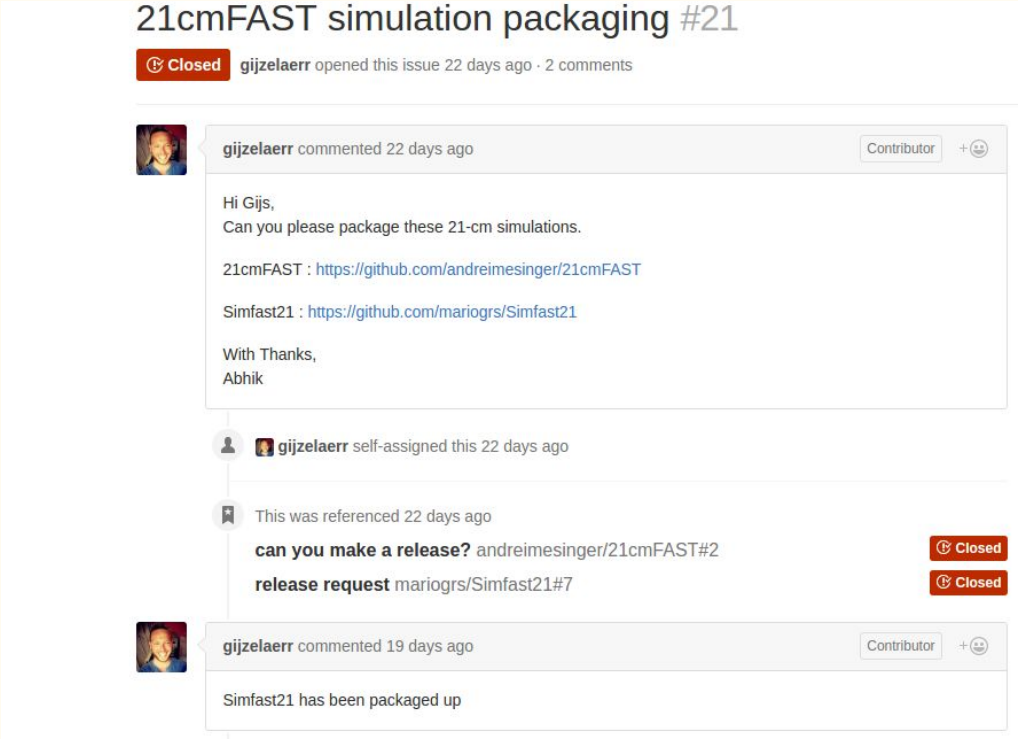
Packaging software is vital to producing reproducible science data products



- Manage dependencies
- Proper versioning of software
- Forces basic level of standardization of how you write software
- Lowers burden on user
- Many tools to help you do it (even automate it)

➔ Launchpad : https://launchpad.net
➔ Python Packaging Index (pip) : https://pypi.python.org/pypi
➔ **Kernsuite (For radio astronomy packages)** : http://kernsuite.info

Contact Kernsuite



21cmFAST simulation packaging #21

Closed **gijzelaerr** opened this issue 22 days ago · 2 comments

gijzelaerr commented 22 days ago — Contributor

Hi Gijs,
Can you please package these 21-cm simulations.

21cmFAST : https://github.com/andreimesinger/21cmFAST

Simfast21 : https://github.com/mariogrs/Simfast21

With Thanks,
Abhik

gijzelaerr self-assigned this 22 days ago

This was referenced 22 days ago

can you make a release? andreimesinger/21cmFAST#2 — Closed

release request mariogrs/Simfast21#7 — Closed

gijzelaerr commented 19 days ago — Contributor

Simfast21 has been packaged up

```
$ apt-get install 21cmfast simfast21
```

Ensure that the code will at least install correctly; with proper management of deps.

Only supports Ubuntu LTS distros

https://github.com/kernsuite/packaging

Container Technology

- OS level virtualization
- Build self contained computing environments for applications
- Minimal interaction with host system (only share Linux Kernel)
- Lightweight, and very little overhead
- Presents an elegant and simple solution to most problems with RI software applications.
- Build. Ship. Run

coreos.com/rkt

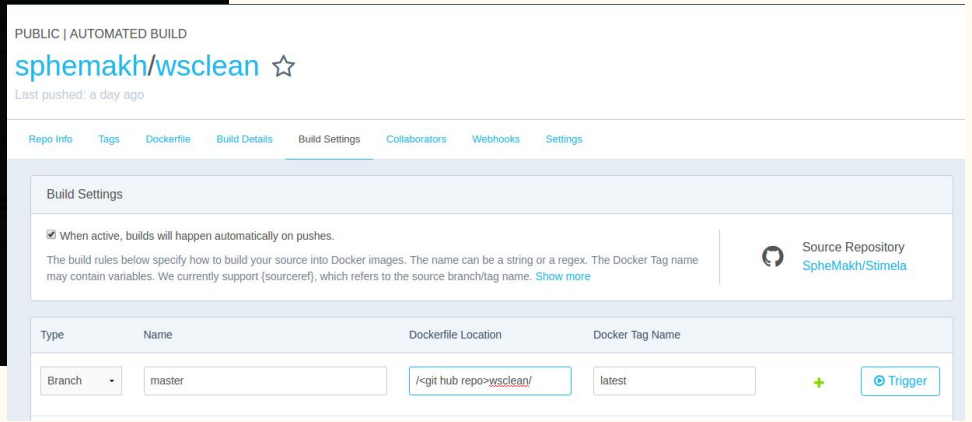
singularity.lbl.gov

docker.com

Docker

Pull from docker hub (https://hub.docker.com)

```
1  FROM ubuntu:16.04
2
3  # Adding the kernsuite PPA
4  RUN apt-get update
5  RUN apt-get install software-properties-common -y
6  RUN apt-add-repository -s ppa:kernsuite/kern-1
7  RUN apt-add-repository multiverse
8
9  # Installing wsclean
10 RUN apt-get update && \
11     apt-get install -y \
12         wsclean
```




```
$ docker build -t wsclean <location of Dockerfile>
$ docker run wsclean /usr/bin/wsclean <args>
```

Commit changes and push new image

```
$ docker commit <container name/ID> <image name>
$ docker push <new image name>
```

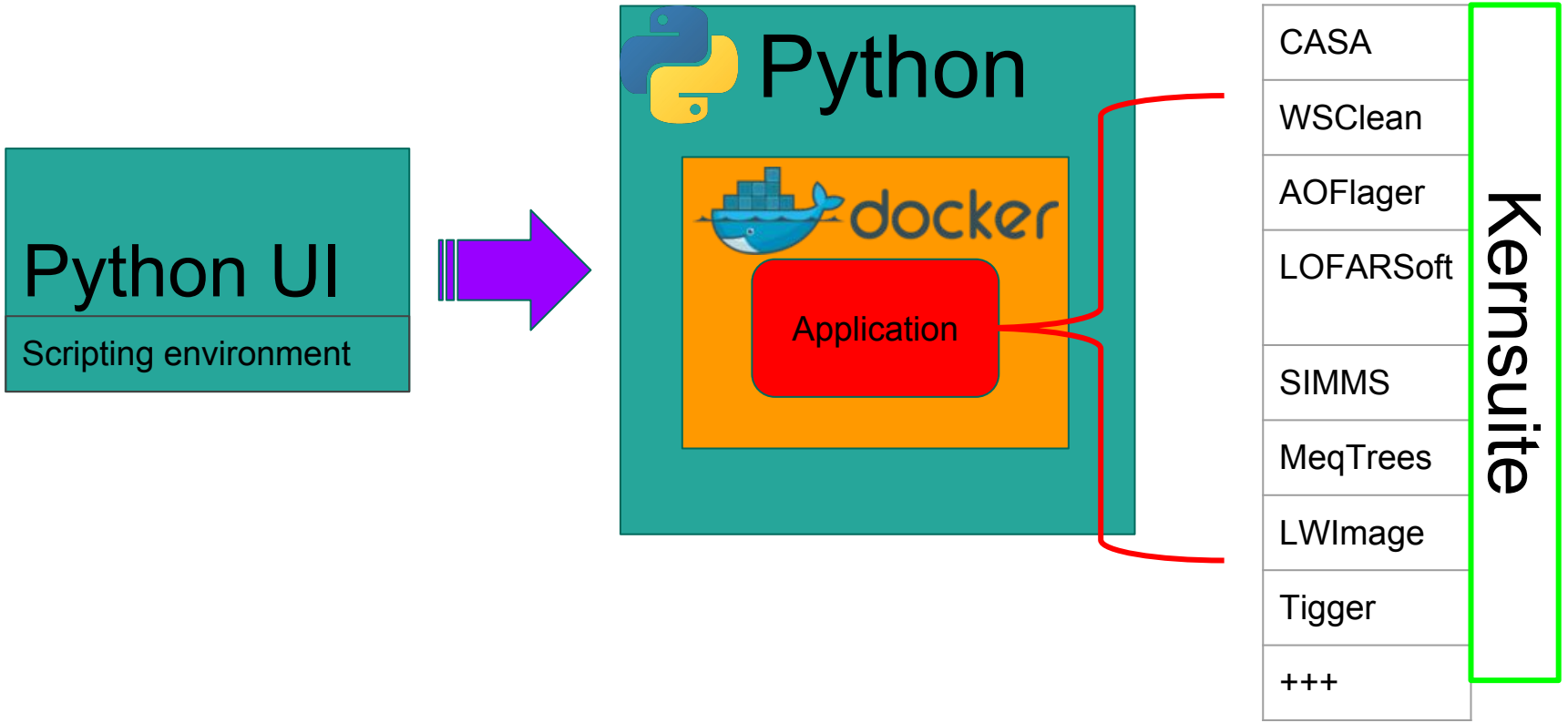
Stimela

- Combines Docker & Kernsuite to create a platform agnostic scripting framework.
- Provides easy access to legacy and novel applications
- Simple Python scripting environment
- Modular, Portable, Robust

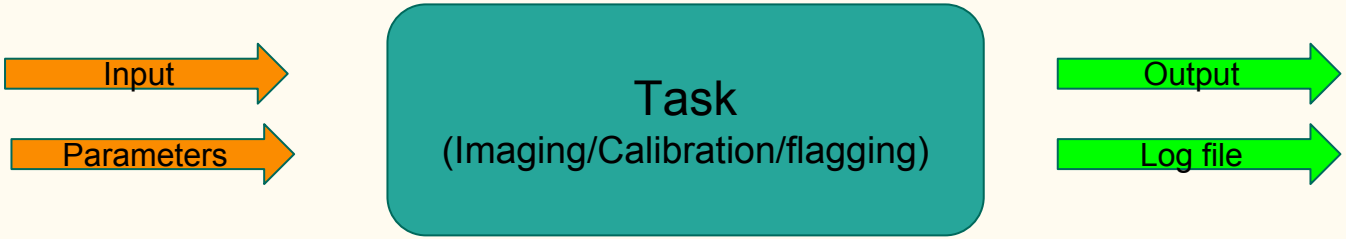
https://github.com/SpheMakh/Stimela
https://github.com/SpheMakh/Stimela/wiki

```
$ pip install stimela
```

Architecture



Workflow of a stimela module (a.k.a cab)




```
{
    "task": "casa_uvsub",
    "base": "stimela/casa",
    "tag": "0.2.5",
    "description": "Subtract/add model from/to the corrected visibility data.",
    "prefix": "",
    "binary": "uvsub",
    "msdir": true,
    "parameters": [
        {
            "info": "Name of input visibility file (MS",
            "name": "msname",
            "io": "msfile",
            "default": null,
            "dtype": "file",
            "mapping": "vis"
        },
        {
            "info": "reverse the operation (add rather than subtract)",
            "dtype": "bool",
            "default": false,
            "name": "reverse"
        }
    ]
}
```

Example

```
1  import stimela
2
3  # recipe I/O flow
4  INPUT = "input"
5  OUTPUT = "output"
6  MSDIR = "msdir"
7
8
9  MS = "measurement_set_name"
10 LSM = "sky_model_name"
11
12 # Start recipe instance
13 recipe = stimela.Recipe("Example Stimela Calibration Recipe", ms_dir=MSDIR)
```

Example

```
# Add task to the recipe
recipe.add("cab/calibrator",  "Gjones_cal_1", # Label of container that will be created from the Docker image
{ # Parameters to be parsed to task
    "msname"    :   MS,
    "skymodel"  :   LSM,
    "Gjones"    :   True,
    "Gjones_solution_intervals" :   [1,1],
},
           input=INPUT, output=OUTPUT, # Task I/O
           label="Direction independent calibration", # Its good to log what you doing
          )
```

Example

```
27  recipe.add("cab/wsclean", "make_image", {
28      "msname"        :   MS,
29      "prefix"    :  "example_stimela_calibration"
30      "npix"          :   4096,
31      "cellsize"      :   2, # in arcsec
32      "weight"        :   "briggs 0" ,
33  },
34
35              input=INPUT, output=OUTPUT,
36              label="Image visibility data",
37              )
```

```
recipe.run()
# recipe.run([1,2])
# recipe.run([1])
# recipe.run([2])
# recipe.run([2,1])
```

```
$ stimela run <recipe name>
```

VerMeerKAT

End-to-end MeerKAT data reduction pipeline

- Full treatment of DDEs
- Portable, configurable
- Can be easily deployed on any system (with Docker)

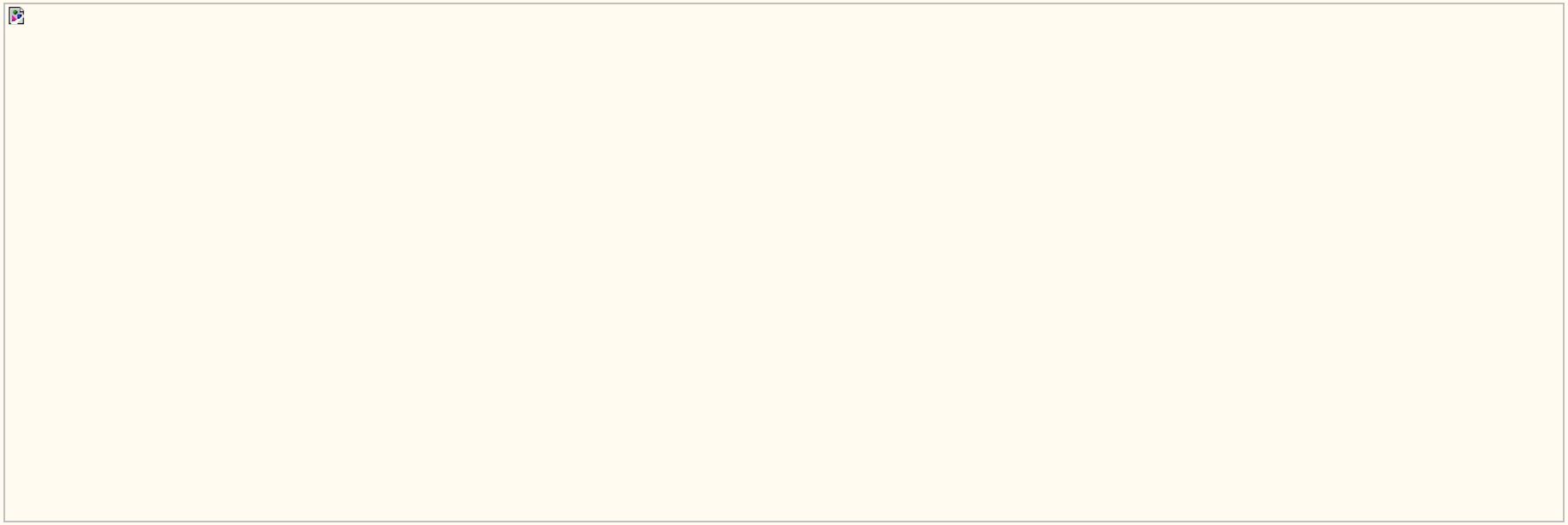
```
---
general:
    solr-url: "http://192.168.1.50:8983/solr/kat_core"
    msdir : 'msdir'
    input : 'input'
    output : 'output'
    fov : 1.5
    sampling : 0.1 # PSF sampling (<1)
    prefix : 'vermeerkat-pipeline' # change me, please!

h5toms:
    full_pol: true

rfimask:
    rfi-mask-file: "rfi_mask.pickle"

autoflag:
    column: "DATA"
    strategy-file: "29Dec2016_firstpass_strategy.rfis"
```

Install and run

```
$ pip install git+https://github.com/ska-sa/vermeerkat
$ vermeerkat -f <observation id>.h5 -c <config file>
```

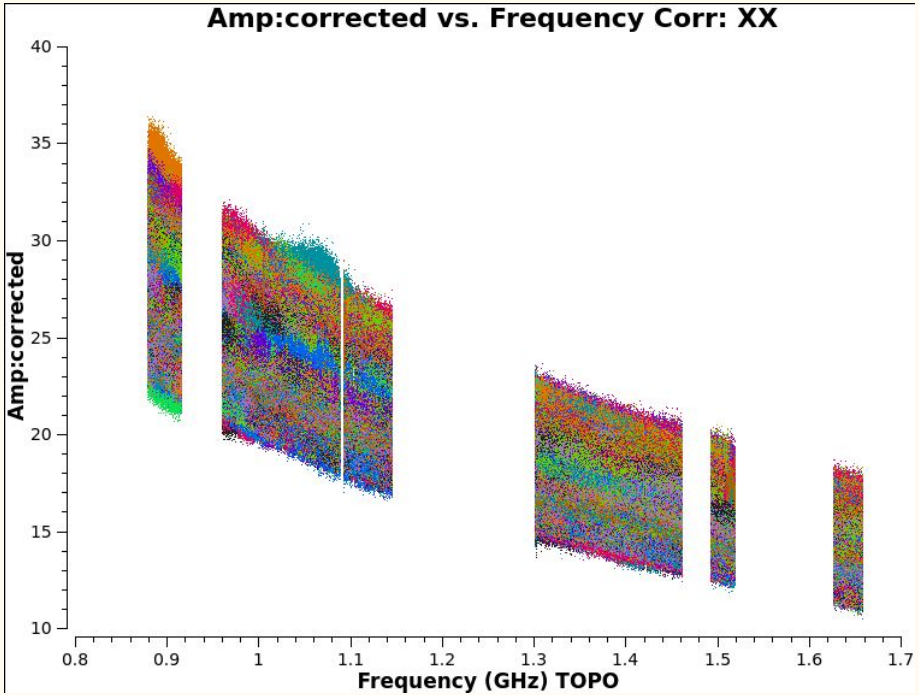
https://github.com/ska-sa/vermeerkat (private for now)

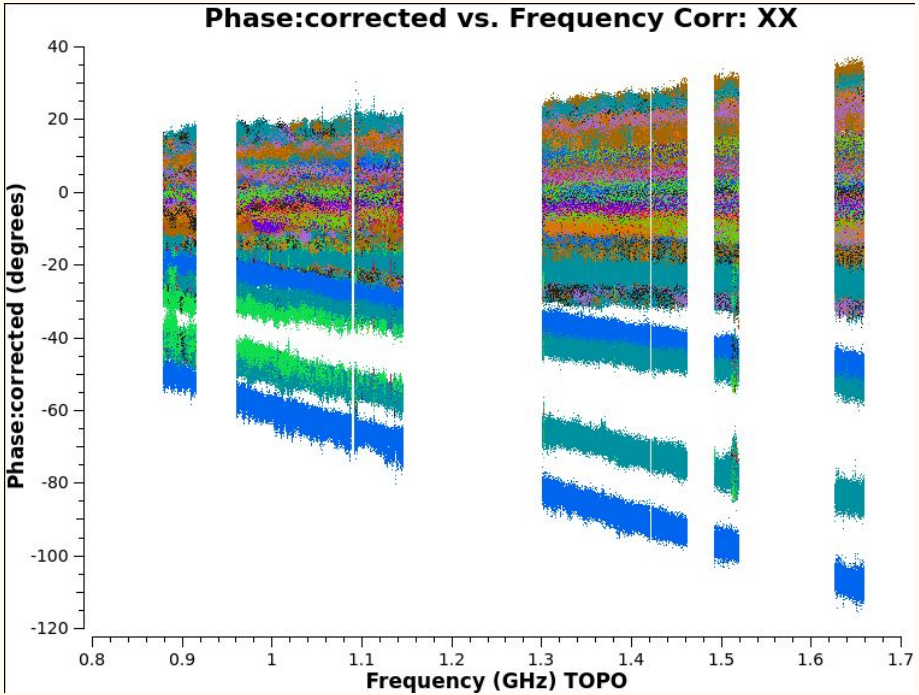
VerMeerKAT continuum pipeline



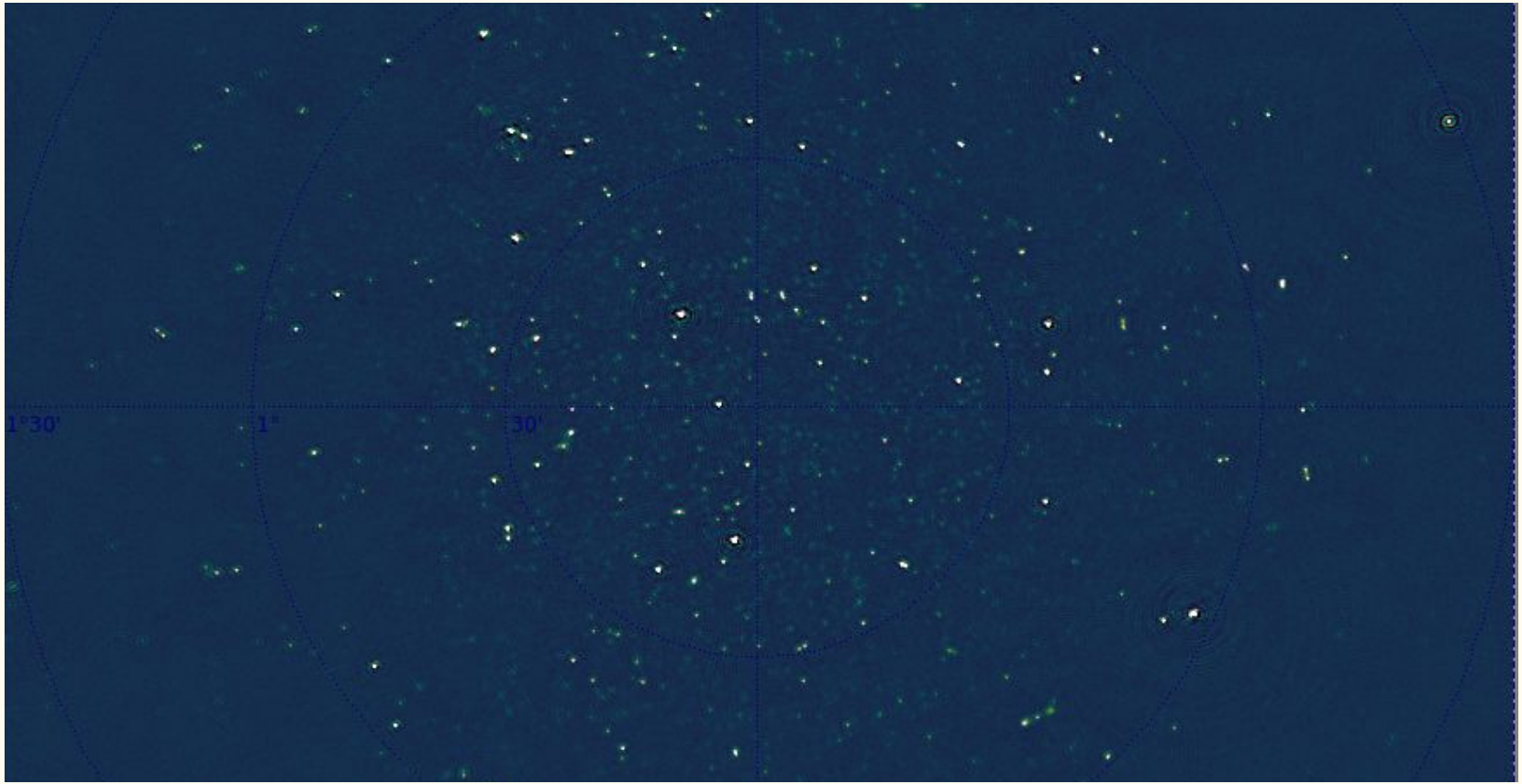
https://github.com/ska-sa/vermeerkat (private for now)

DEEP2


Amp:corrected vs. Frequency Corr: XX
Amp:corrected
Frequency (GHz) TOPO
40
35
30
25
20
15
10
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

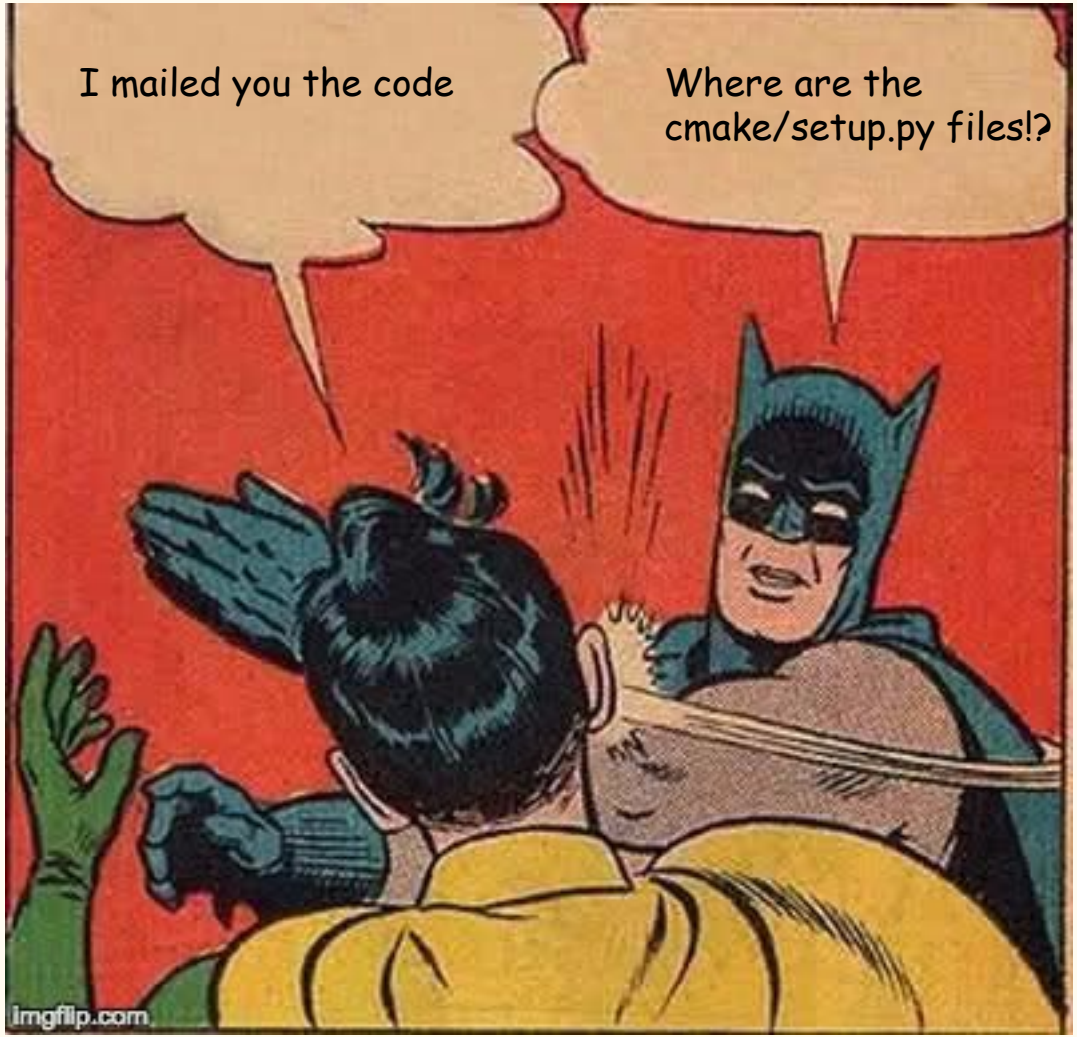

Phase:corrected vs. Frequency Corr: XX
Phase:corrected (degrees)
Frequency (GHz) TOPO
40
20
0
-20
-40
-60
-80
-100
-120
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7

DEEP2



Conclusion

- Document your code
- Package it
- Try out container technology
- We have the tools, expertise and resources to make robust, modular and scalable data reduction/simulation pipelines.


I mailed you the code
Where are the cmake/setup.py files!?
imgflip.com