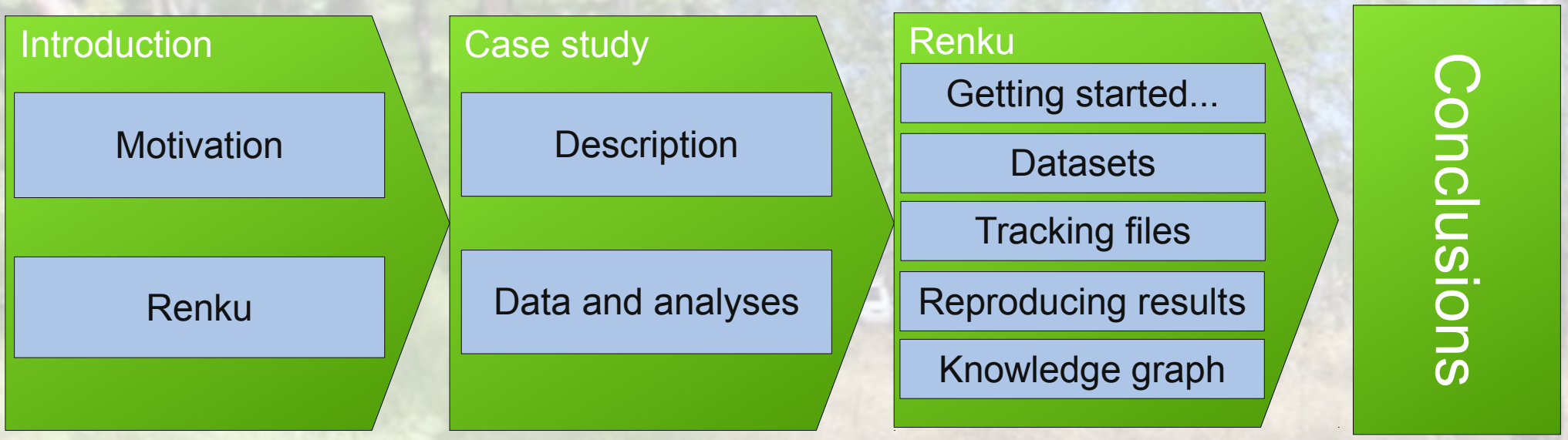
A repeatable and reproducible modelling workflow using the **Vegetation Optimality Model** and **RENKU**

R.C. Nijzink[1], C. Ramakrishnan[2], R. Roskar[2], S. Schymanski[1]

[1] Luxembourg Institute of Science and Technology, Belvaux, Luxembourg,
[4] Swiss Data Science Center, Zurich, Switzerland

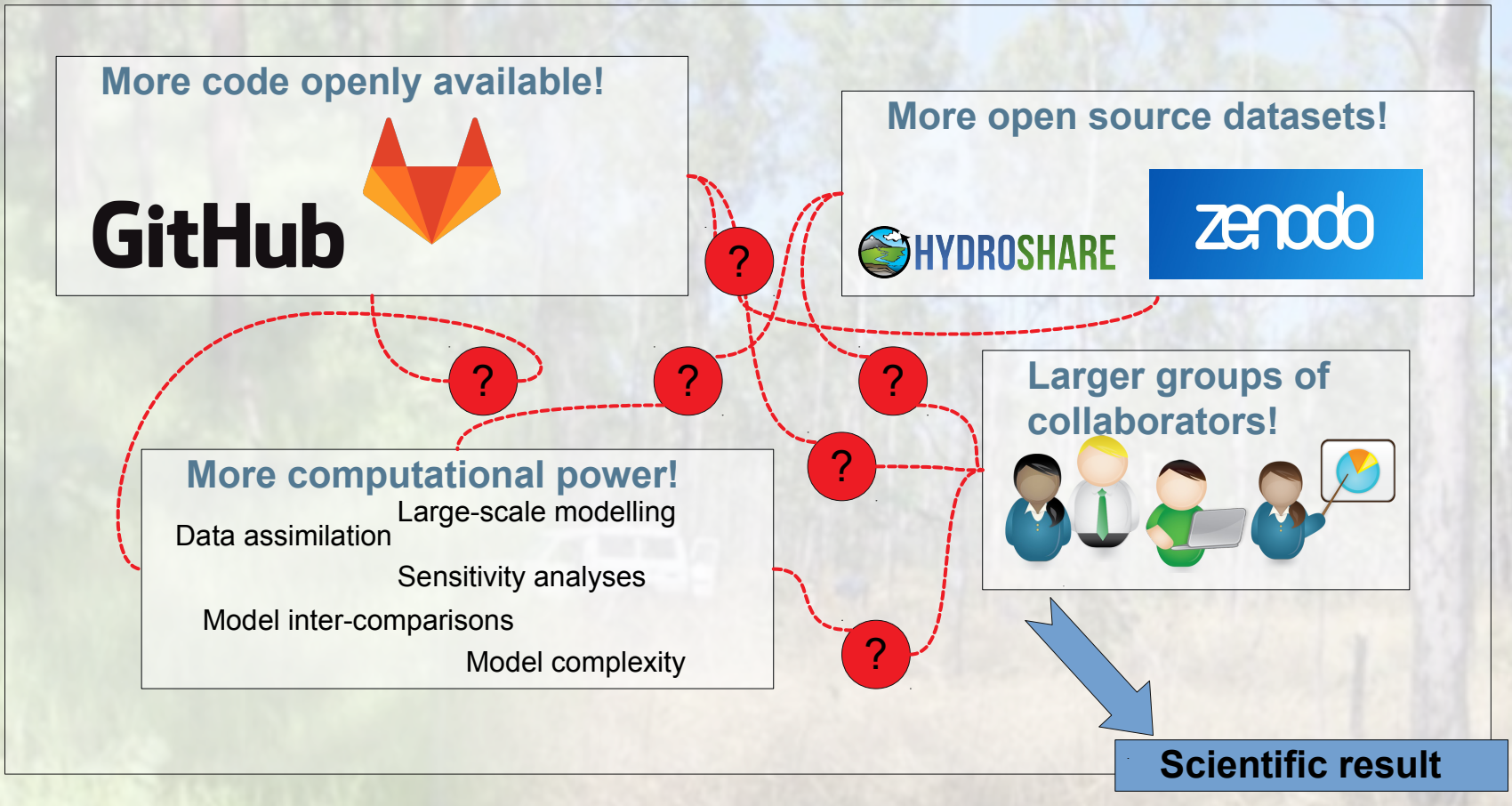


Luxembourg National Research Fund
LUXEMBOURG INSTITUTE OF SCIENCE AND TECHNOLOGY
LIST

Supported by the Luxembourg National Research Fund (FNR) ATTRACT programme (A16/SR/11254288)

Next

MOTIVATION



How to ensure reproducibility and repeatability?

RENKU

RENKU 連句

Environment for collaborative, reproducible data science

Concepts

- Tracking of scientific steps to create data lineage, i.e. a knowledge graph
- Updating of out-dated results
- Tool to re-use or re-run analyses
- Sharing of analyses

Features

RENKU is based on :

- Gitlab
- JupyterHub
- Kubernetes
- Keycloak
- Common Workflow Language

Usage

- Command Line Interface
- Web platform

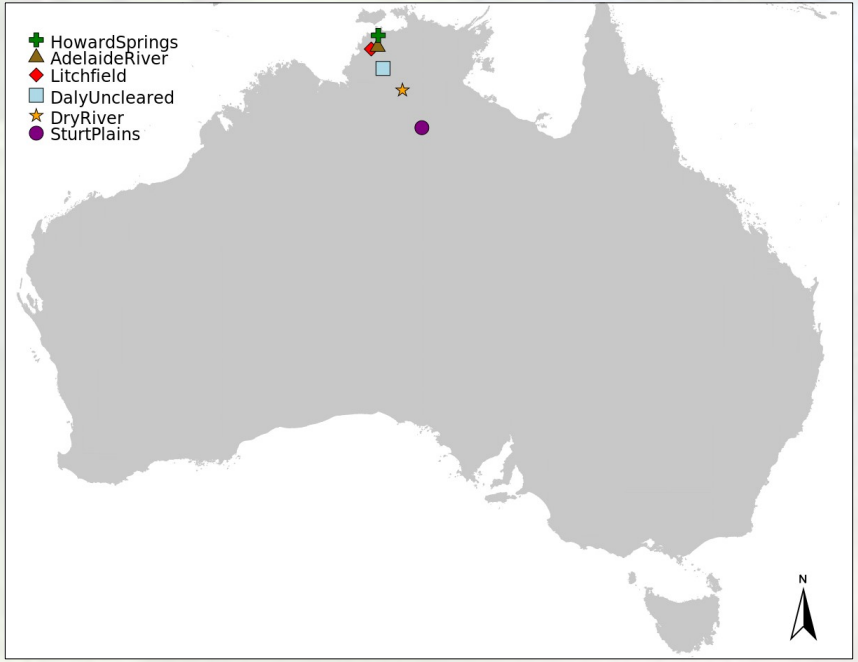
Previous Home Next

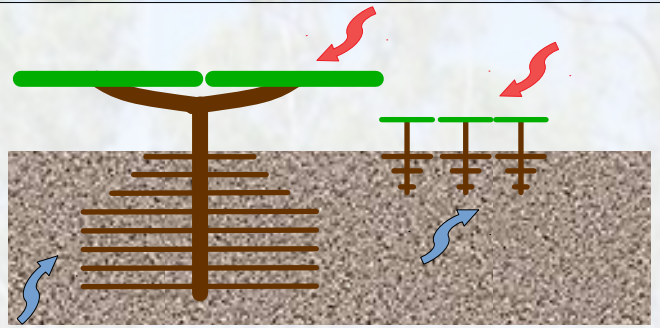
More info

CASE STUDY - DESCRIPTION

Study sites

Six sites with a strong precipitation gradient along the ***North Australian Tropical Transect***.





Vegetation Optimality Model

Optimizes vegetation properties to maximize the ***Net Carbon Profit***, i.e. the difference of carbon uptake by photosynthesis and carbon costs of the system.

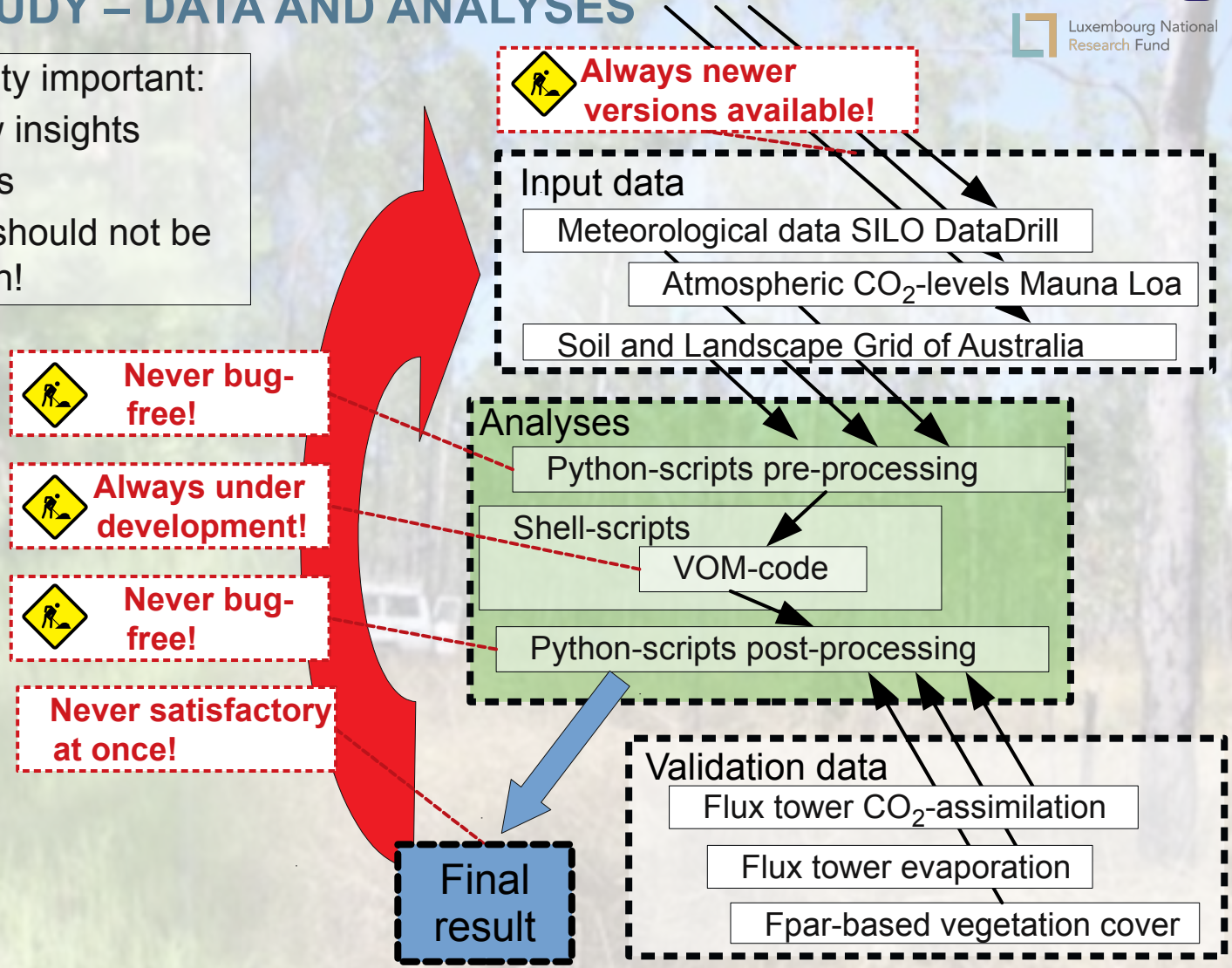
Research Question

Does maximization of Net Carbon Profit explain vegetation behaviour in savanna sites along a precipitation gradient?

See session HS10.1 General Ecohydrology, Thursday 7th of May 8.30 – 10.15 for the scientific results!

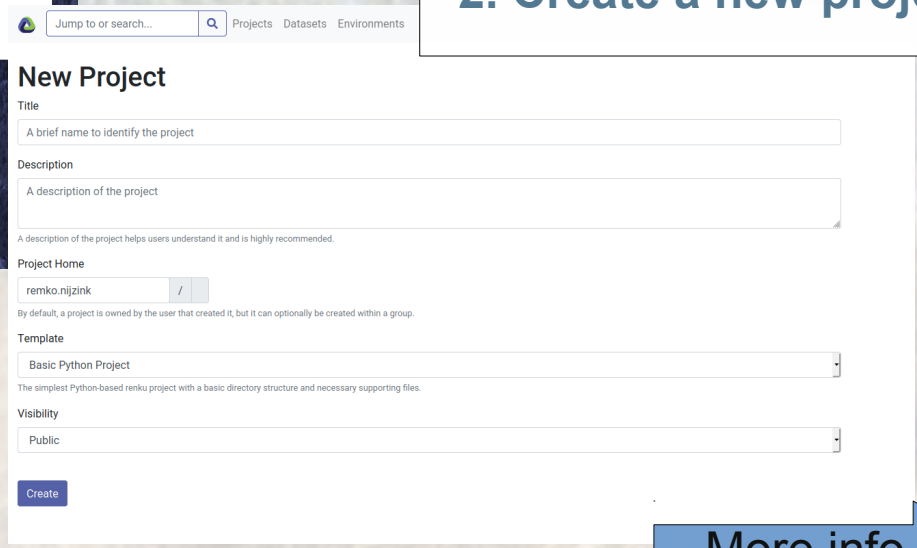
CASE STUDY – DATA AND ANALYSES

- Repeatability important:
  - New insights
  - Bugs
- "Failures" should not be be forgotten!



GETTING STARTED...

1. Register at renkulab.io



2. Create a new project

- Gitlab-repository
- Platform for collaboration and sharing of data and code

Previous Home Next

More info

GETTING STARTED...

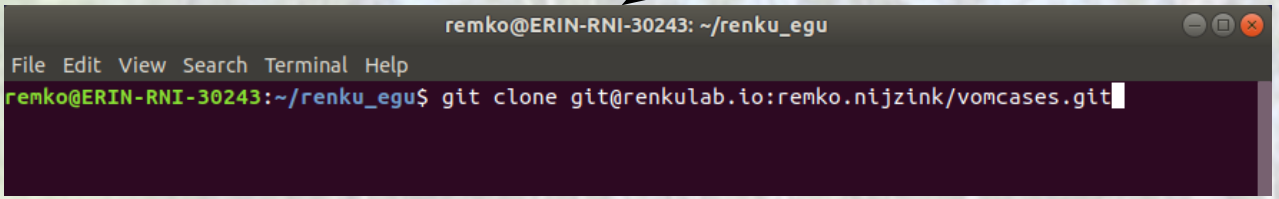
Working with RENKU:

- Local through command line
- Online through JupyterLab

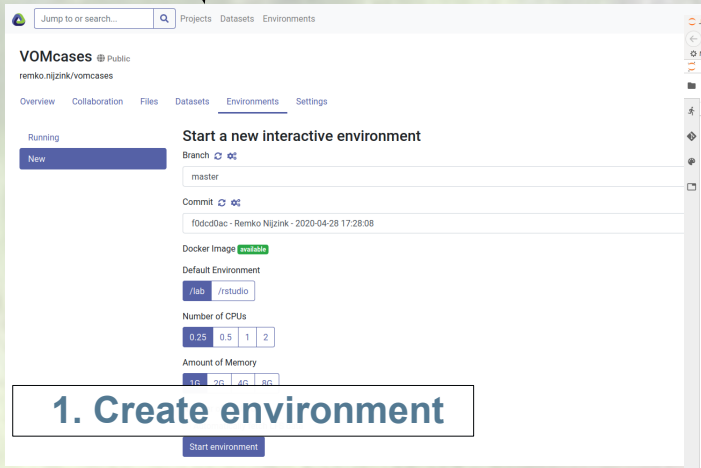
➔ No local software needed!

➔ Easy for collaboration!

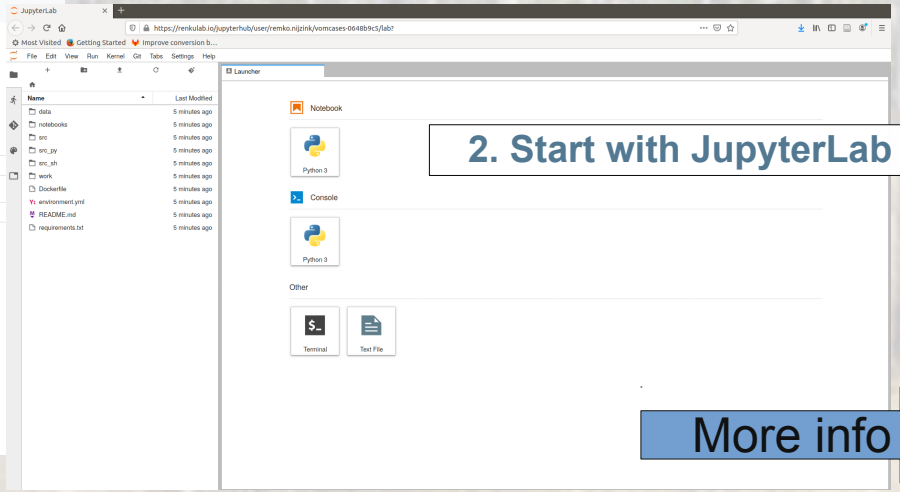
Clone the project locally



Use RENKU on JupyterLab



1. Create environment



2. Start with JupyterLab

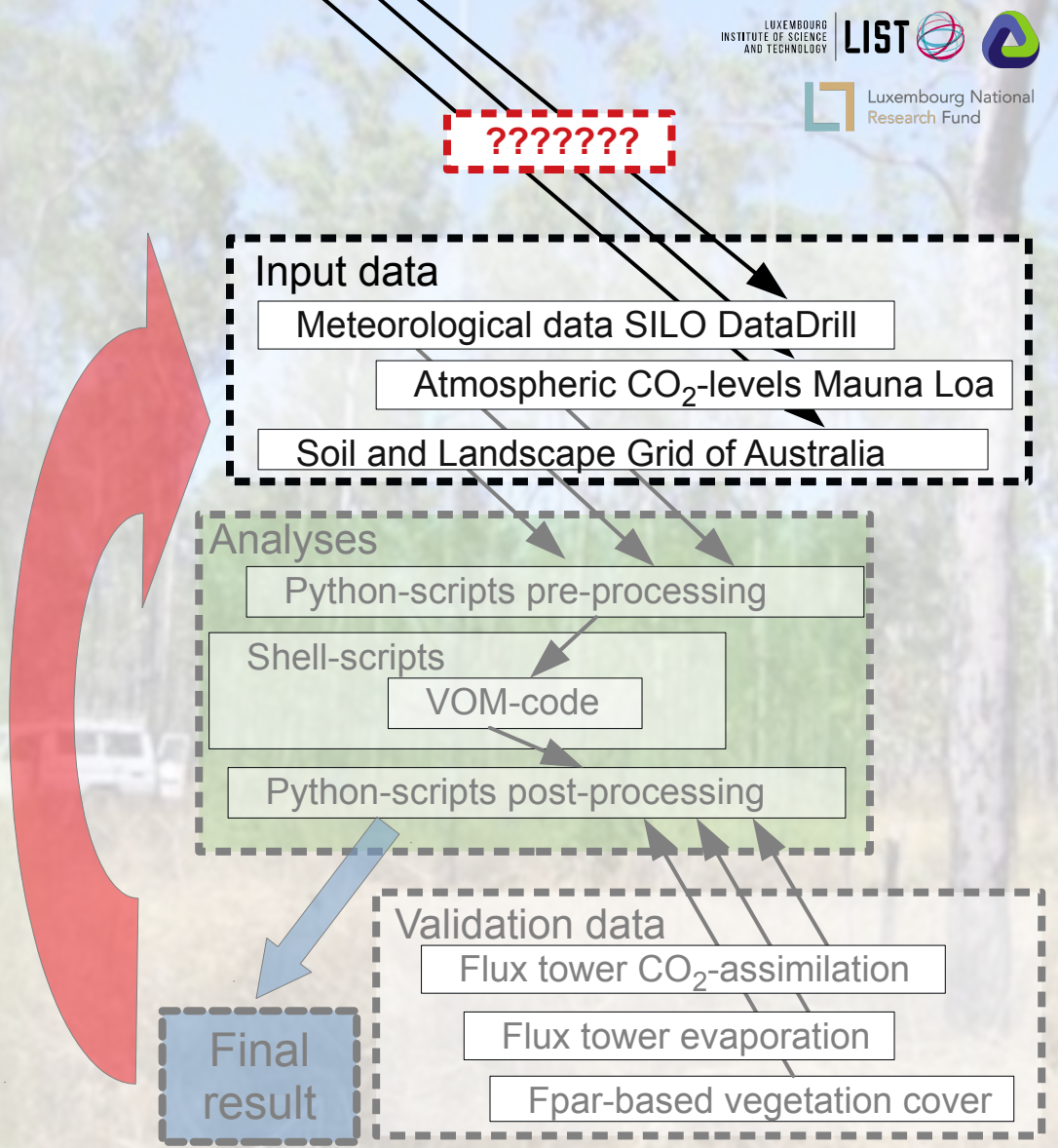
More info

DATASETS

Challenges

- Different versions of data
- Data continuously updated
- Increasing size
- Different ways of distributing data:
  - Repositories
  - Servers
  - Email
  - ….

How to link final outputs to the ***original, unchanged and permanent*** data source?



DATASETS

Solving it with RENKU

1. Create a dataset in the repository

- Adds metadata
- Automatically under git large file storage (git-lfs)

2. Add data with RENKU

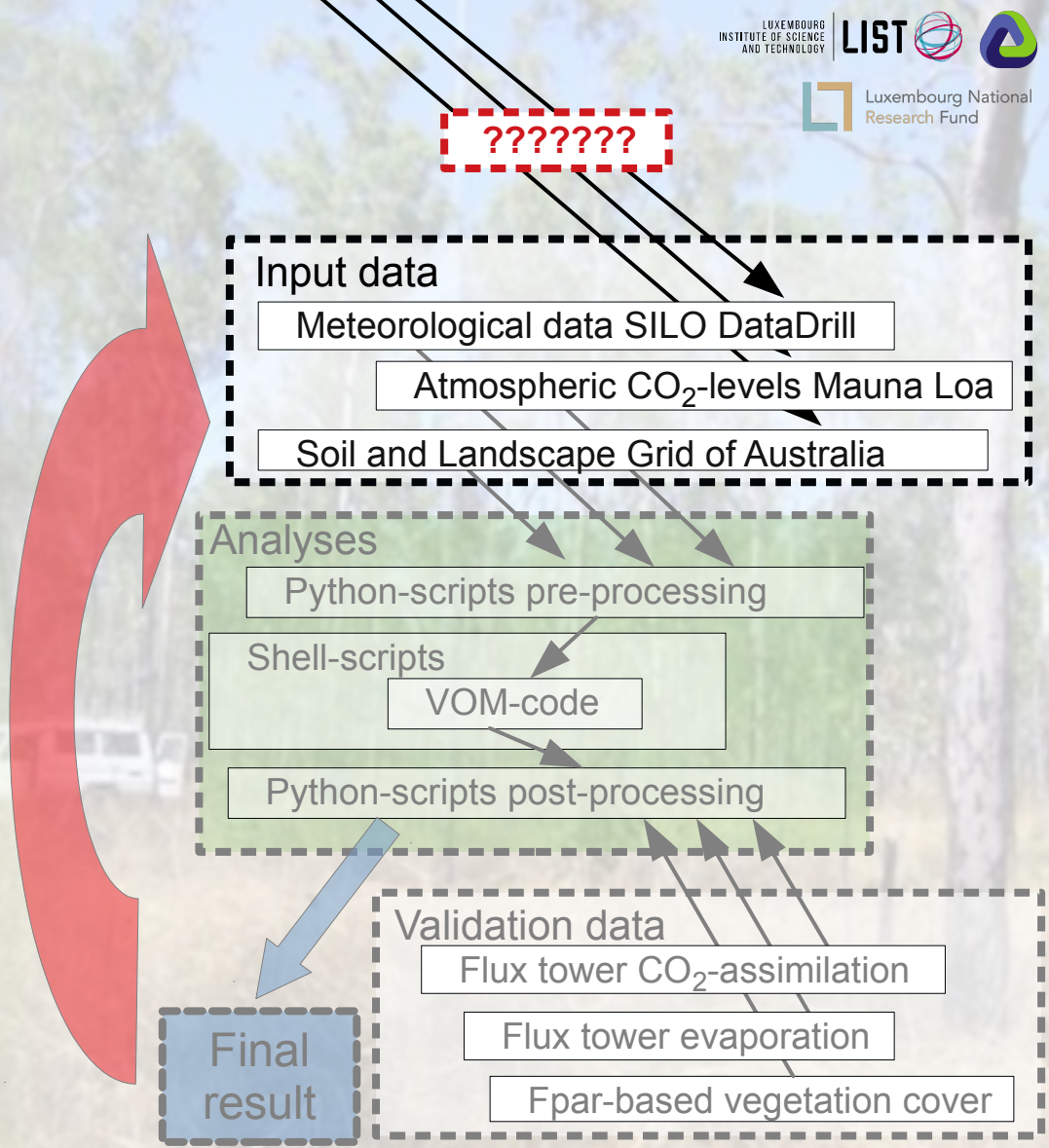
- Directly from source
  ➔ e.g. DOI, hyperlink
- Metadata saved in repository!

More info

How to link final outputs to the ***original, unchanged and permanent*** data source?

Introduction
Case study
Renku
Conclusions

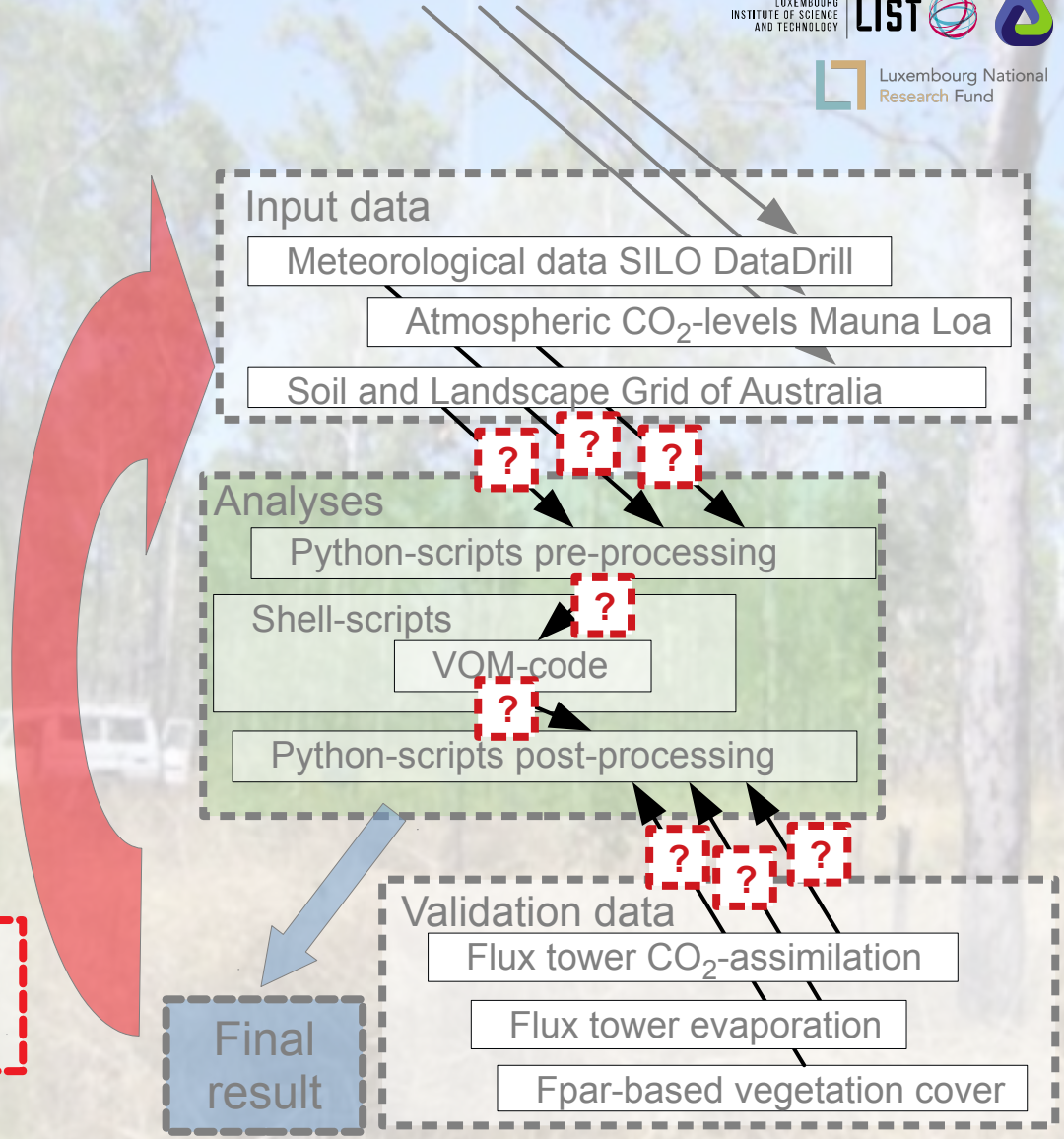
Previous Home Next



TRACKING FILES

Challenges

- Different versions of data, code and scripts
- Linkage between files unclear
- Complex computations with many options, flags, settings etc.



How to track how final outputs are ***exactly*** created with which tools, data, and settings?

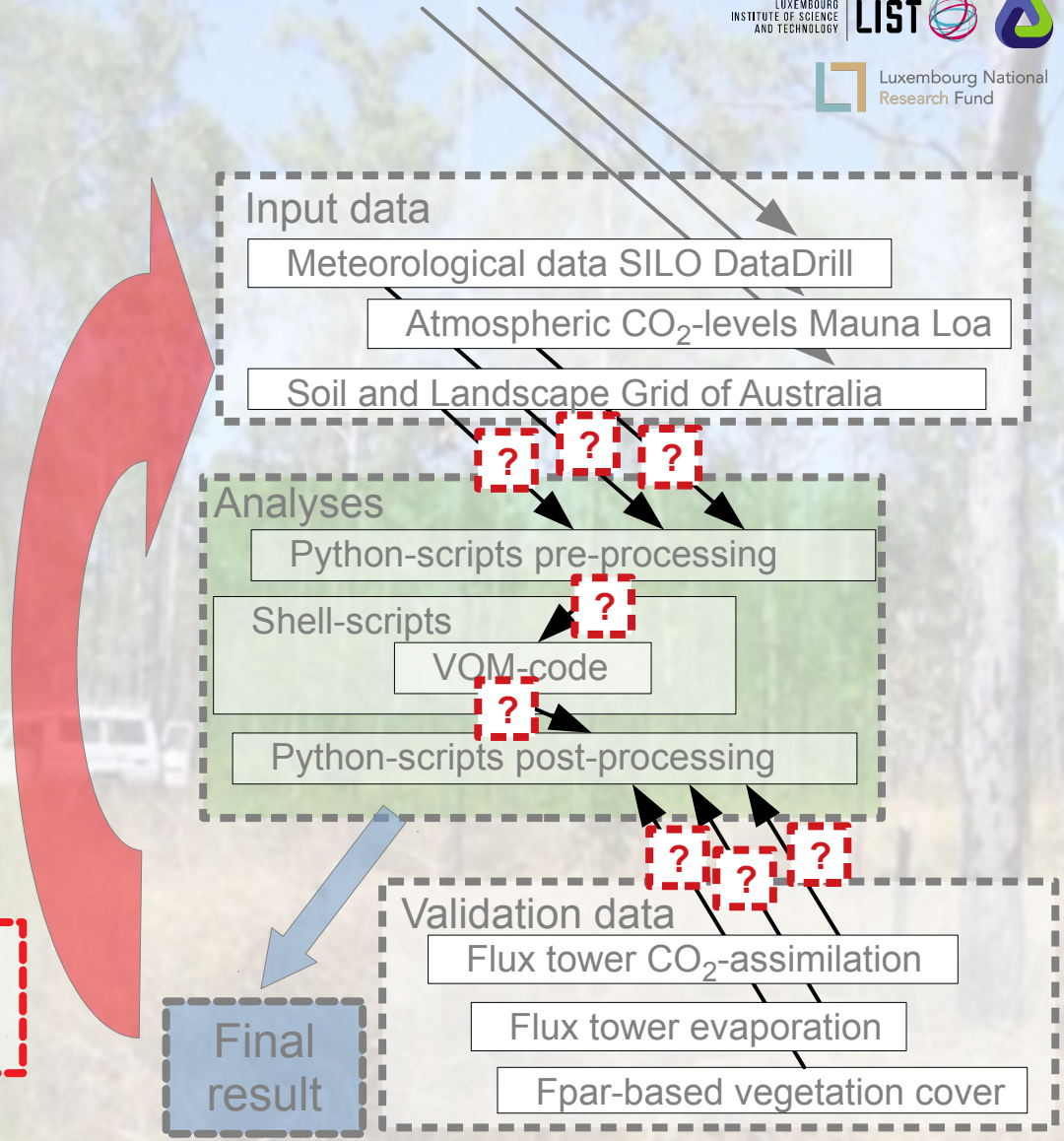
TRACKING FILES

Solving it with RENKU

Renku run

- Tracks inputs and outputs
- Detects arguments
- Git-history shows how the file was created!
- Helpful for reproducibility and repeatability!

More info

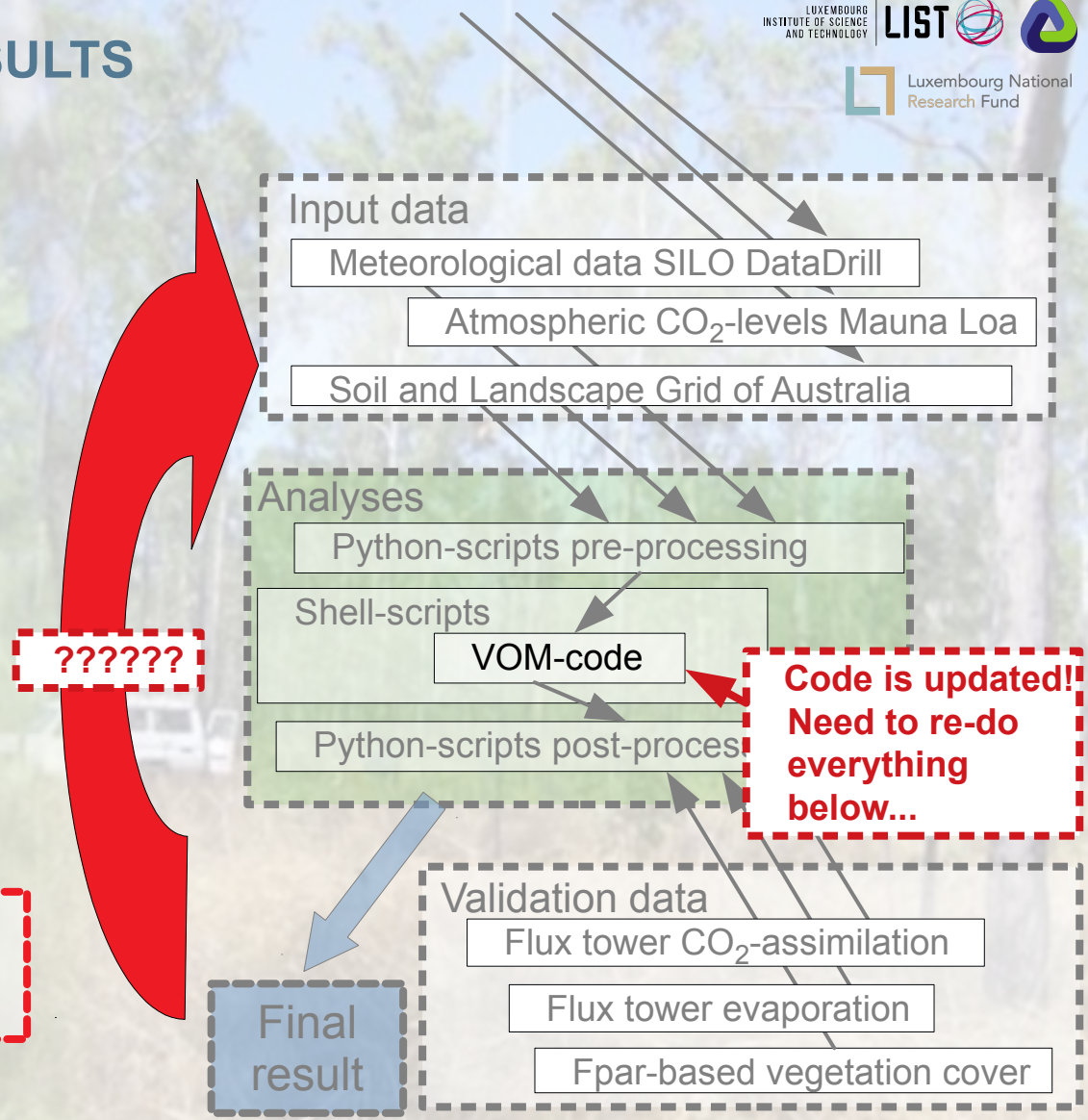


How to track how final outputs are ***exactly*** created with which tools, data, and settings?

REPRODUCING RESULTS

Challenges

- Different versions of data, code and scripts
- Linkage between files unclear
- Complex computations with many options, flags, settings etc.



How to easily update results in ***exactly*** the same way as the outdated results were created?

REPRODUCING RESULTS

Solving it with RENKU

Renku status

- Detects changes
- Shows if files are up-to-date

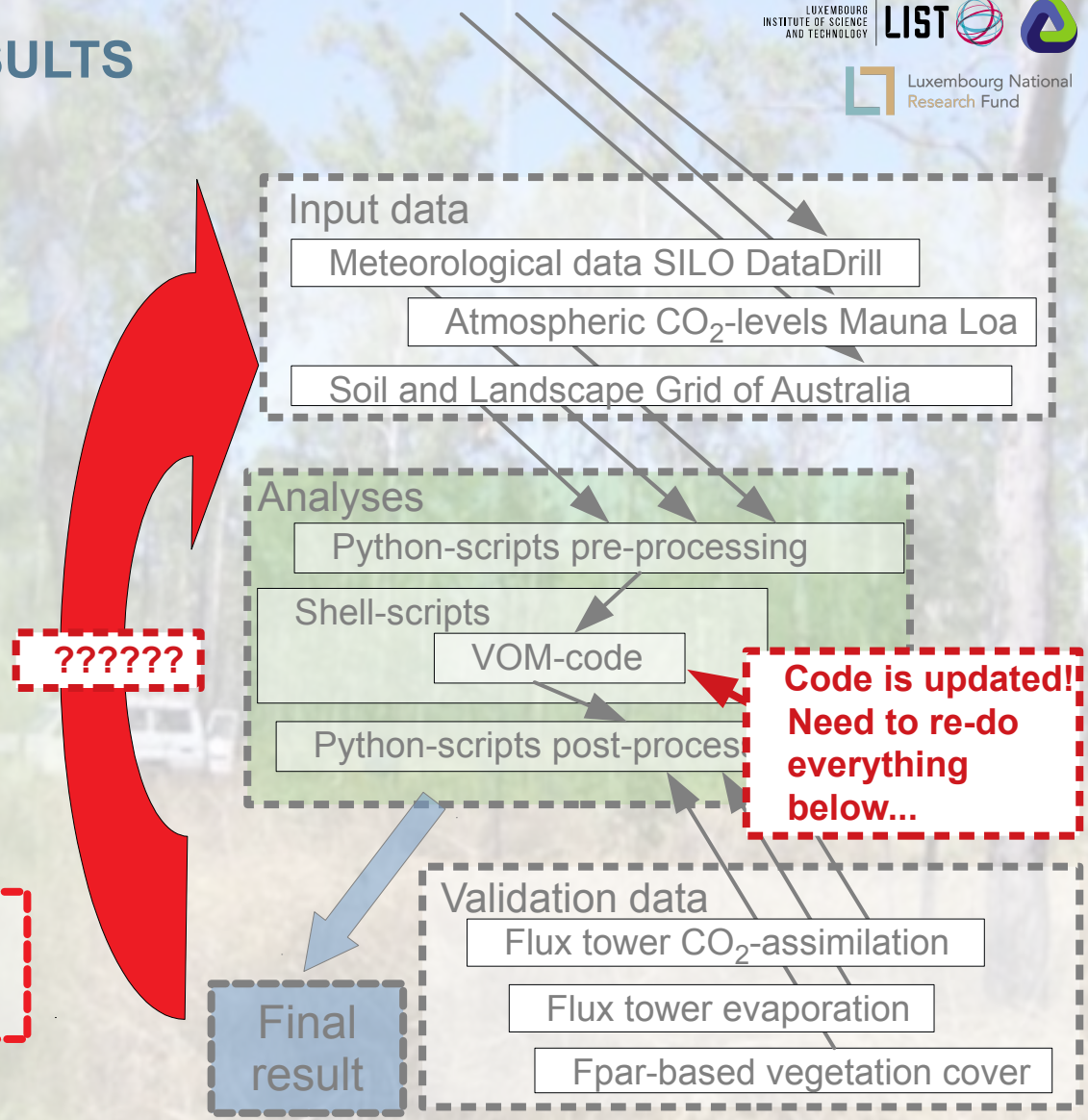
More info

Renku update

- Repeats analysis
- Updates all outputs based on newest inputs

More info

How to easily update results in ***exactly*** the same way as the outdated results were created?

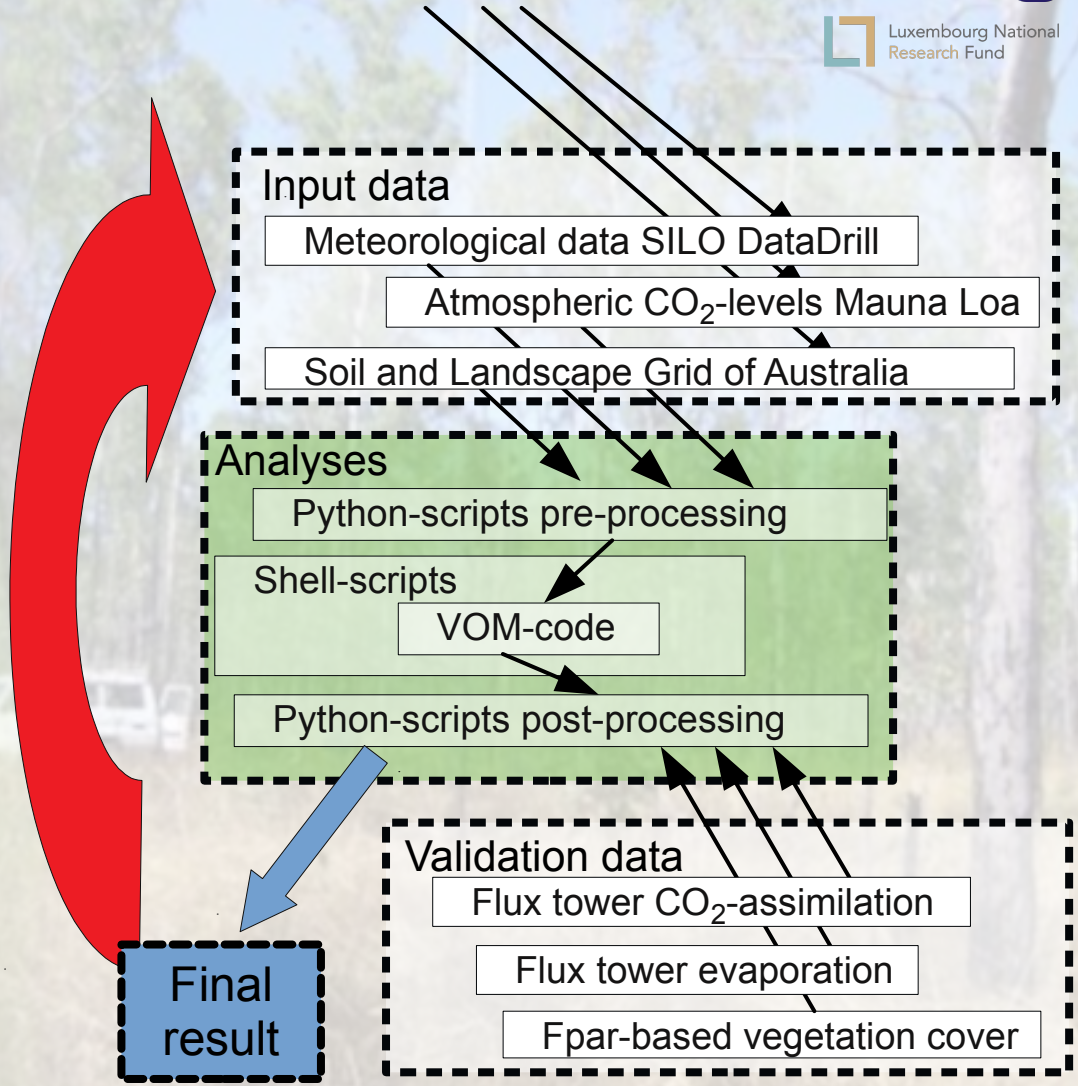


KNOWLEDGE GRAPH

Challenges

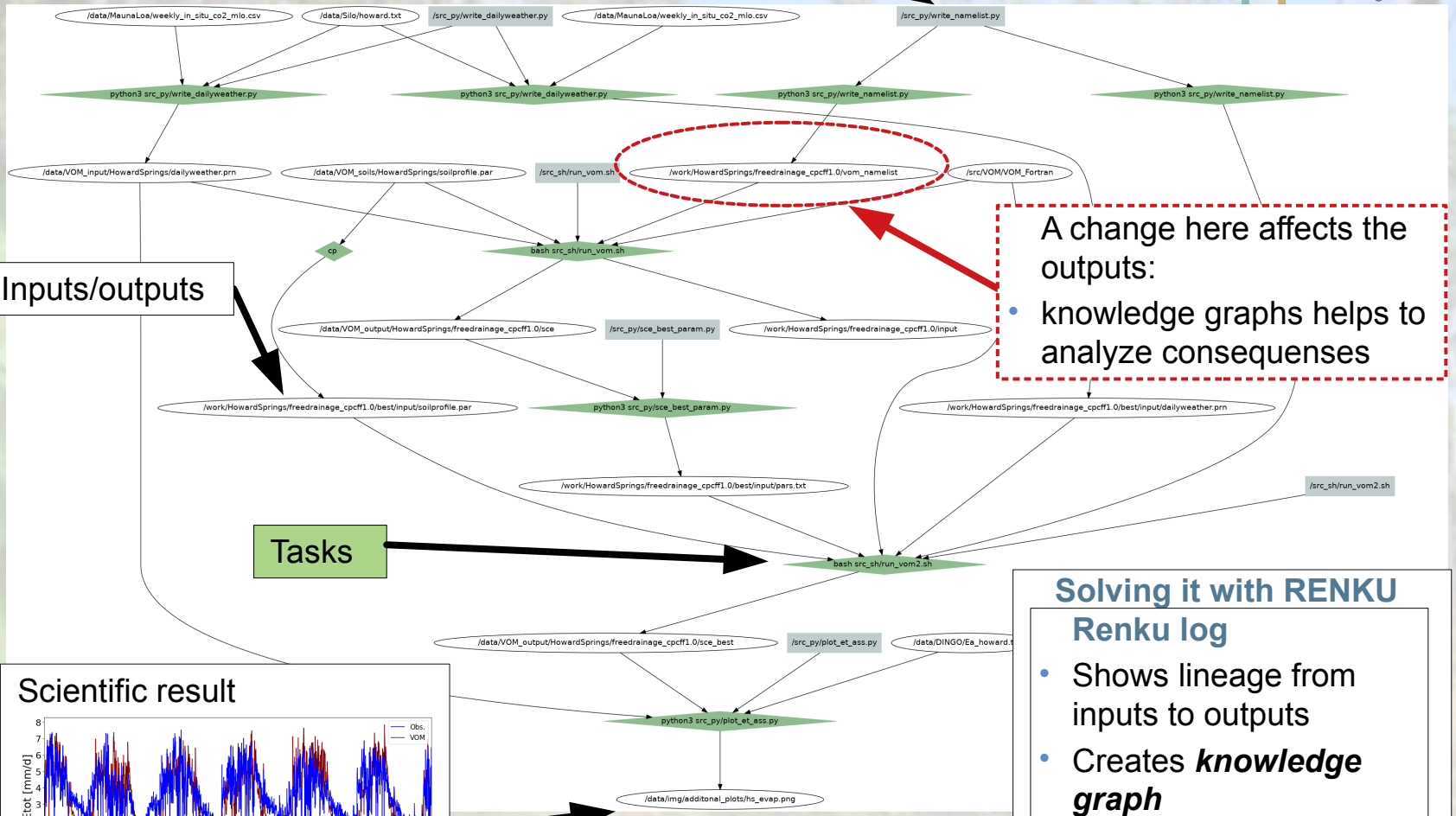
- Lineage from inputs to outputs often not clear
- Different versions of code: what was used?
- Complex computations with many options, flags, settings
- Not clear which files are affected when inputs change

How to see ***exactly and completely*** how the results are produced and affected by changes?



KNOWLEDGE GRAPH

Scripts/code



A change here affects the outputs:

- knowledge graphs helps to analyze consequenses

Inputs/outputs

Tasks

Solving it with RENKU

Renku log

- Shows lineage from inputs to outputs
- Creates ***knowledge graph***

More info

Scientific result

More info

Conclusions

- Renkulab.io provides a platform for collaborative science by sharing code, data and workflows.

  See here →

- The Renku Jupyterlab allows working on a collaborator's project without installing software.

  See here →

- Renku provides systematic way to store metadata of datasets and link to the original sources

  See here →

- Data lineage can be preserved from original data source until final results.

  See here →

- Renku can repeat analyses in order to update results, based on the latest inputs.

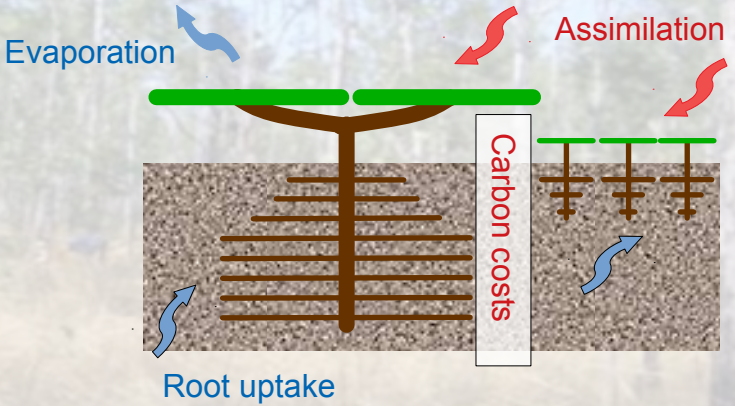
  See here →

- Knowledge graph shows full data lineage for reproducibility and repeatability, and assessment of the influence of changes.

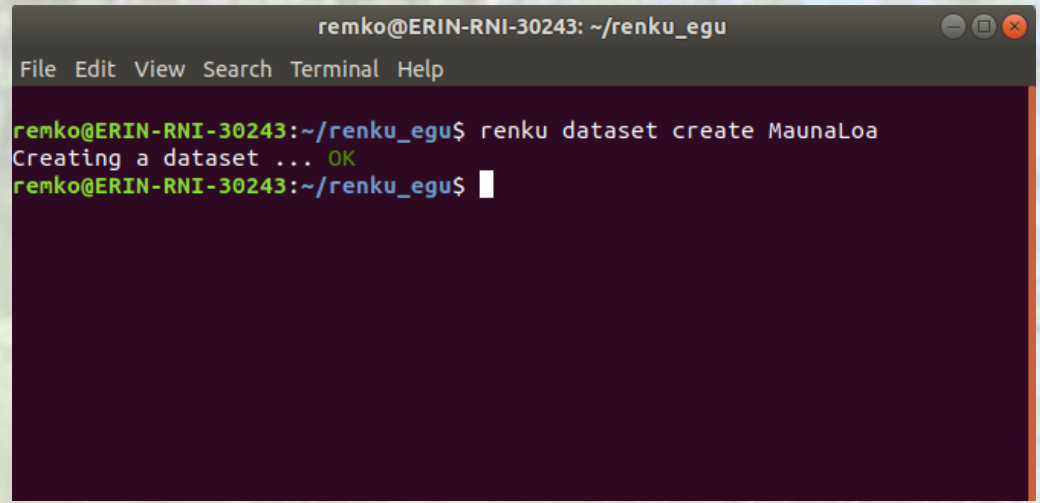
  See here →

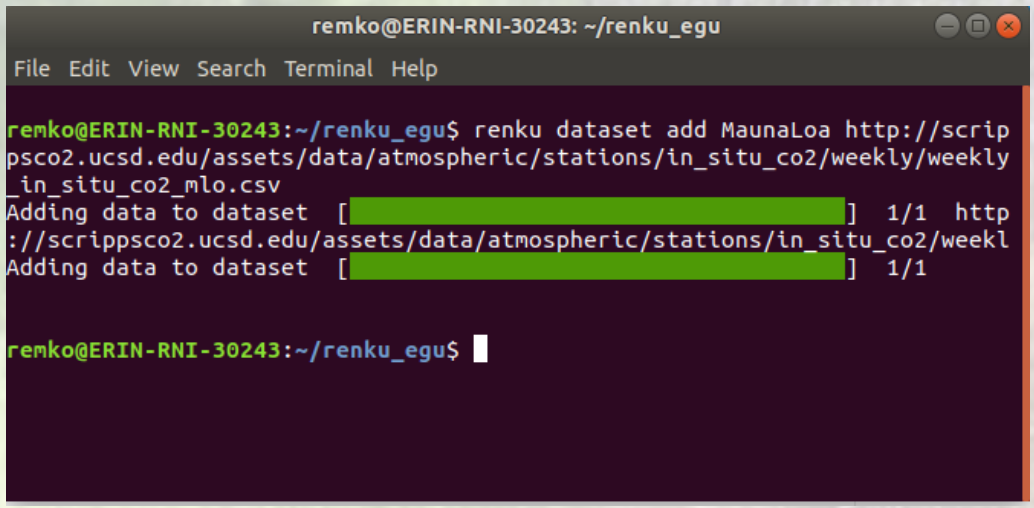
Want to get started? Click here for a tutorial.

APPENDIX



DATASETS





1. Create a dataset in the repository

- Adds metadata
- Automatically under git large file storage (git-lfs)

More info

2. Add data

- From online sources
- Import with DOI
- Local files

More info

Back

TRACKING FILES


```
remko@ERIN-RNI-30243: ~/renku_egu
File  Edit  View  Search  Terminal  Help
remko@ERIN-RNI-30243:~/renku_egu$ renku run bash run_vom.sh ../../data/VOM_input/HowardSprin
gs/dailyweather.prn vom_namelist ../../src/VOM/VOM_Fortran/VOM-code/*
```

```
remko@ERIN-RNI-30243: ~/renku_egu
File  Edit  View  Search  Terminal  Help
remko@ERIN-RNI-30243:~/renku_egu$ git log data/VOM_input/dailyweather.prn
commit d59d4b48b8efef892bdbcb6418fd7ac06bfc8ef1 (HEAD -> master)
Author: Remko Nijzink <remko.nijzink@list.lu>
Date:   Thu Apr 30 13:27:38 2020 +0200

    renku update

commit 57dd7eed60a257219286ca137140f0210ad463c3
Author: Remko Nijzink <remko.nijzink@list.lu>
Date:   Thu Apr 30 13:27:38 2020 +0200

    renku: automatic removal of unchanged files

commit d61513ecbe45d7878c7c2de0b6797734b1fe930a
Author: Remko Nijzink <remko.nijzink@list.lu>
Date:   Thu Apr 30 13:08:34 2020 +0200

    renku run python3 src_py/write_dailyweather.py -im data/SILO/howard.txt -ic data/MaunaL
oa/weekly_in_situ_co2_mlo.csv -p linear -o data/VOM_input/dailyweather.prn
remko@ERIN-RNI-30243:~/renku_egu$
```

History shows how the file was created
➔ helpful for reproducibility and repeatability!

Renku run

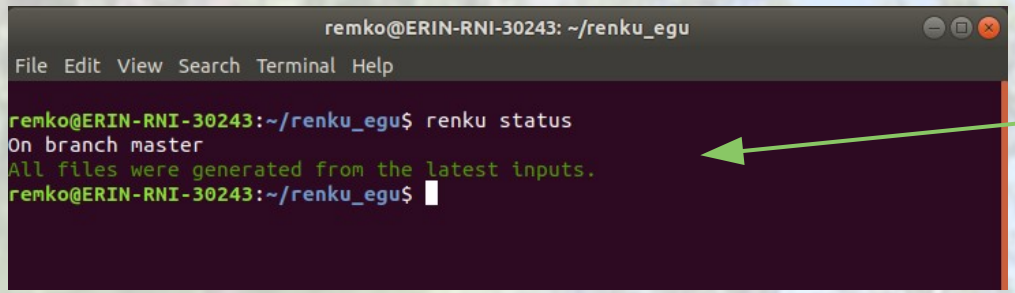
- Tracks inputs and outputs
- Detects arguments
- Adds information as commit to the output

More info

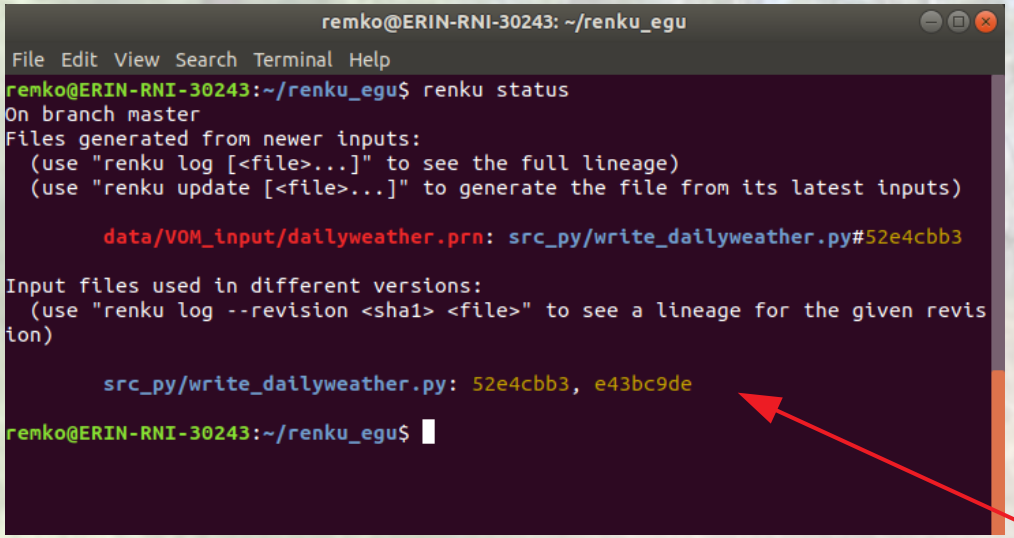
Previous Home Next

Back

REPRODUCING RESULTS



Everything up-to-date!



Renku status

- Detects changes
- Shows if files are up-to-date

More info

Need to update outputs!

Back

REPRODUCING RESULTS


```
remko@ERIN-RNI-30243: ~/renku_egu
File  Edit  View  Search  Terminal  Help

remko@ERIN-RNI-30243:~/renku_egu$ renku update
Resolved '.renku/workflow/21b601c544df4dc8872173d045c3d8d6.cwl' to 'file:///home
/remko/renku_egu/.renku/workflow/21b601c544df4dc8872173d045c3d8d6.cwl'
[workflow ] start
[workflow ] starting step step_1
[step step_1] start
[job step_1] /tmp/tmp4qpm8jfl$ cp \
    /tmp/tmptjlyrl2o/stg2e3a4e46-0fc8-4bc9-af40-55831c9b59ae/evi_adelaide.txt \
    pc_mod.txt
[job step_1] completed success
[step step_1] completed success
[workflow ] completed success
remko@ERIN-RNI-30243:~/renku_egu$
```

Renku update

- Repeats analysis
- Updates all outputs based on newest inputs

More info

KNOWLEDGE GRAPH


```
remko@ERIN-RNI-30243: ~/renku_egu
File  Edit  View  Search  Terminal  Help
remko@ERIN-RNI-30243:~/renku_egu$ renku log data/VOM_input/dailyweather.prn
*     d59d4b48 data/VOM_input/dailyweather.prn
|\            (part of data/VOM_input directory)
* |   d59d4b48 data/VOM_input
| |           (part of data directory)
@ | .d59d4b48 data
 /
*     d61513ec .renku/workflow/7a0f71ede9ff4874beb50b793b93b2f8_python3.cwl
|\            .(part of d59d4b48 .renku/workflow/64bcf91f08194ca581a50c9bc8392590.cwl#steps/step_1
)
| \
| |\
* | |   e43bc9de src_py/write_dailyweather.py
| | |           (part of src_py directory)
@ | |   e43bc9de src_py
 / /
| *   2d5eee3e data/MaunaLoa/weekly_in_situ_co2_mlo.csv
| |           (part of data/MaunaLoa directory)
| *   2d5eee3e data/MaunaLoa
| |           (part of data directory)
| @   2d5eee3e (latest -> d59d4b48) data
*    990ef1c5 data/SILO/howard.txt
|           (part of data/SILO directory)
*    990ef1c5 data/SILO
|           (part of data directory)
@    990ef1c5 (latest -> d59d4b48) data
remko@ERIN-RNI-30243:~/renku_egu$
```

Shows commit hash

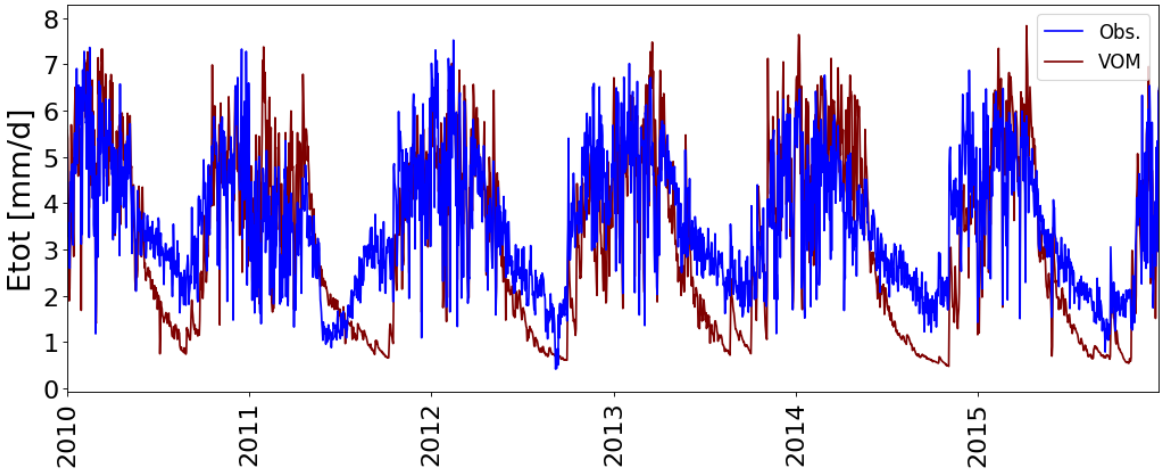
Shows how files are linked

Renku log

- Shows lineage from inputs to outputs
- Creates knowledge graph

More info

More on the knowledge graph...



The VOM-results at Howard Springs (in red), the wettest site along the North-Australian Tropical Transect, show that the VOM under-estimates total evaporation mainly during the dry season (i.e. the lower parts in the time series) in comparison with flux tower observations (blue). This happened similarly at Adelaide River, the second wettest site of the NATT, and this could be explained by the freely draining conditions of the soil column that have been adopted to parameterize the model.

Back to knowlegde graph

Next

Introduction

Case study

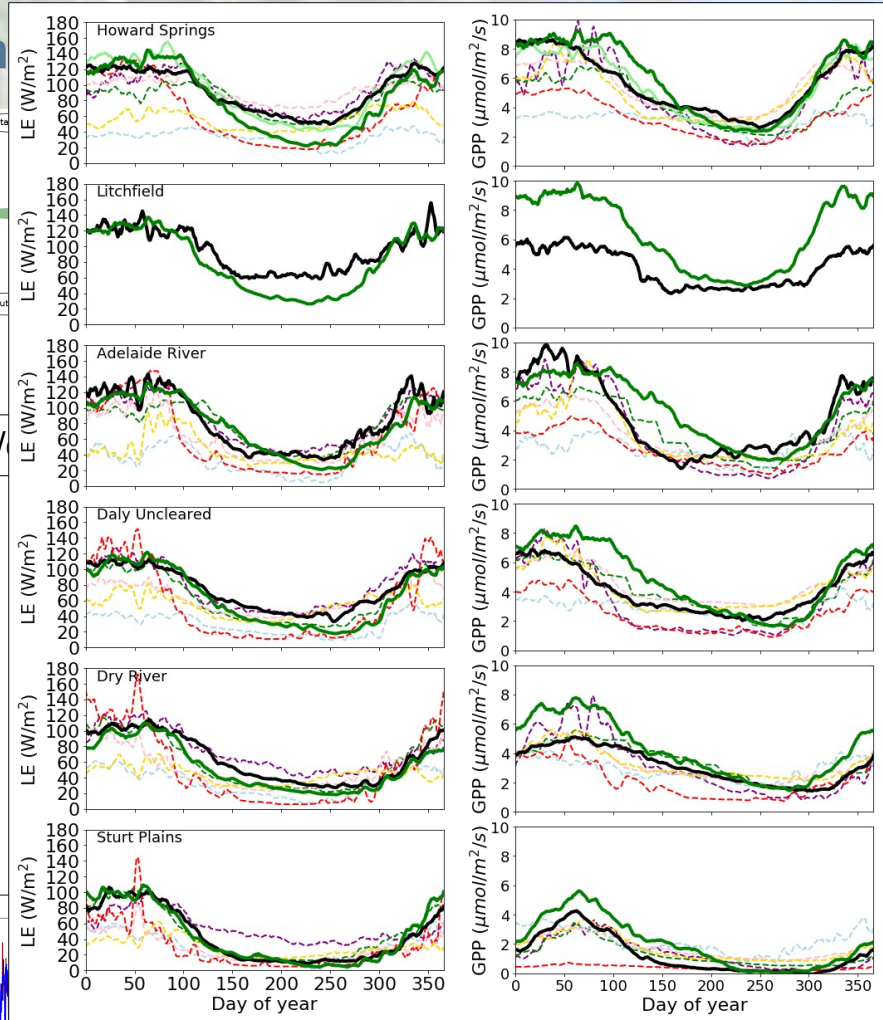
Renku

Conclusions

Previous

Home

Next



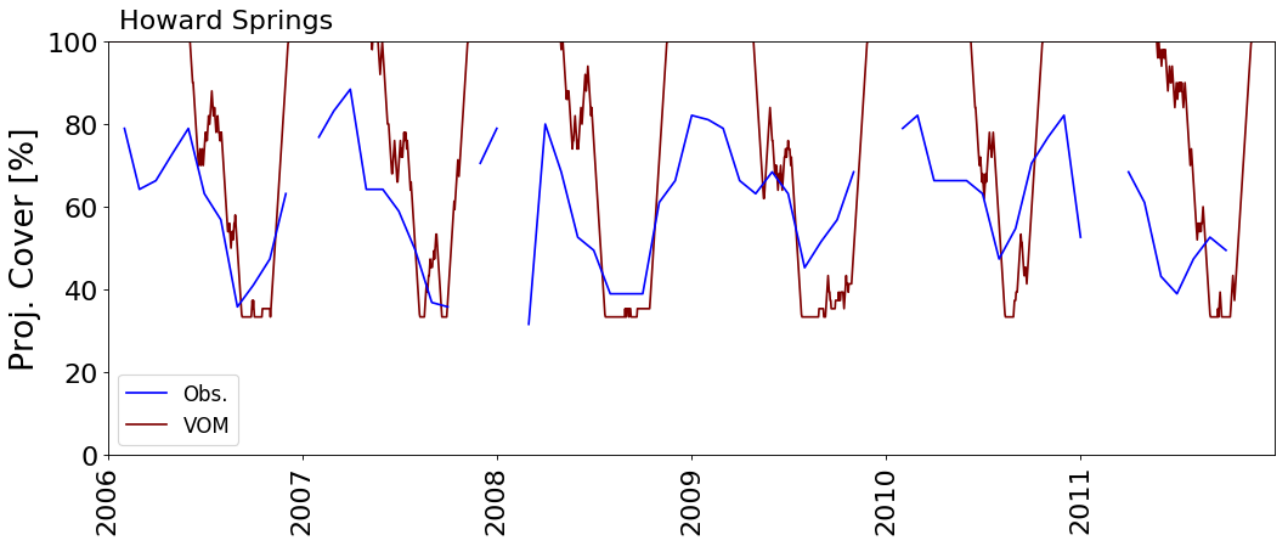
In comparison with other models, the VOM shows a correct seasonal amplitude in most cases and performs equally well or better than these other models. However, an off-set in assimilation can be observed in the transition from the wet to the dry season (days 50-150).

Model data from:
Whitley et al. (2015): Biogeosciences 13

Introduction
Case study
Renku
Conclusions

Previous Home Next

Back
Back to knowlegde graph
Next



The vegetation cover predicted by the VOM (red) shows similar seasonal patterns as remotely sensed fPar-based vegetation cover (blue). However, the vegetation cover predicted by the VOM always reached 100% during the wet season, consistently for all sites, whereas the remotely sensed fPar-based vegetation cover (blue) shows much lower values during the wet season.

Back
Back to knowlegde graph
Next

Previous
Home
Next

See session HS10.1 General Ecohydrology, Thursday 7th of May 8.30 – 10.15 for more scientific results!

Back
Back to knowlegde graph
Close
Previous
Home
Next