Karlsruhe Institute of Technology

CO2 CCON Collaborative Carbon Column Observing Network

PROFFAST 2.0 and PROFFASTpylot

EM27/SUN Telecon, 25 January 2022

Lena Feld, Benedikt Herkommer, Darko Dubravica, Frank Hase

Outline

- PROFFAST v 2.0 (Frank Hase)
- PROFFASTpylot (Lena Feld and Benedikt Herkommer)
  - Motivation
    New features
  - How to install
    How to run
- Closing words and time for questions

Motivation

- Simplified Usage
  - One combined input file
  - Start PROFFASTpylot once
    and run all parts of PROFFAST automatically

- Benefits
  - Avoid careless mistakes
  - Reduce workload when processing data

New Features

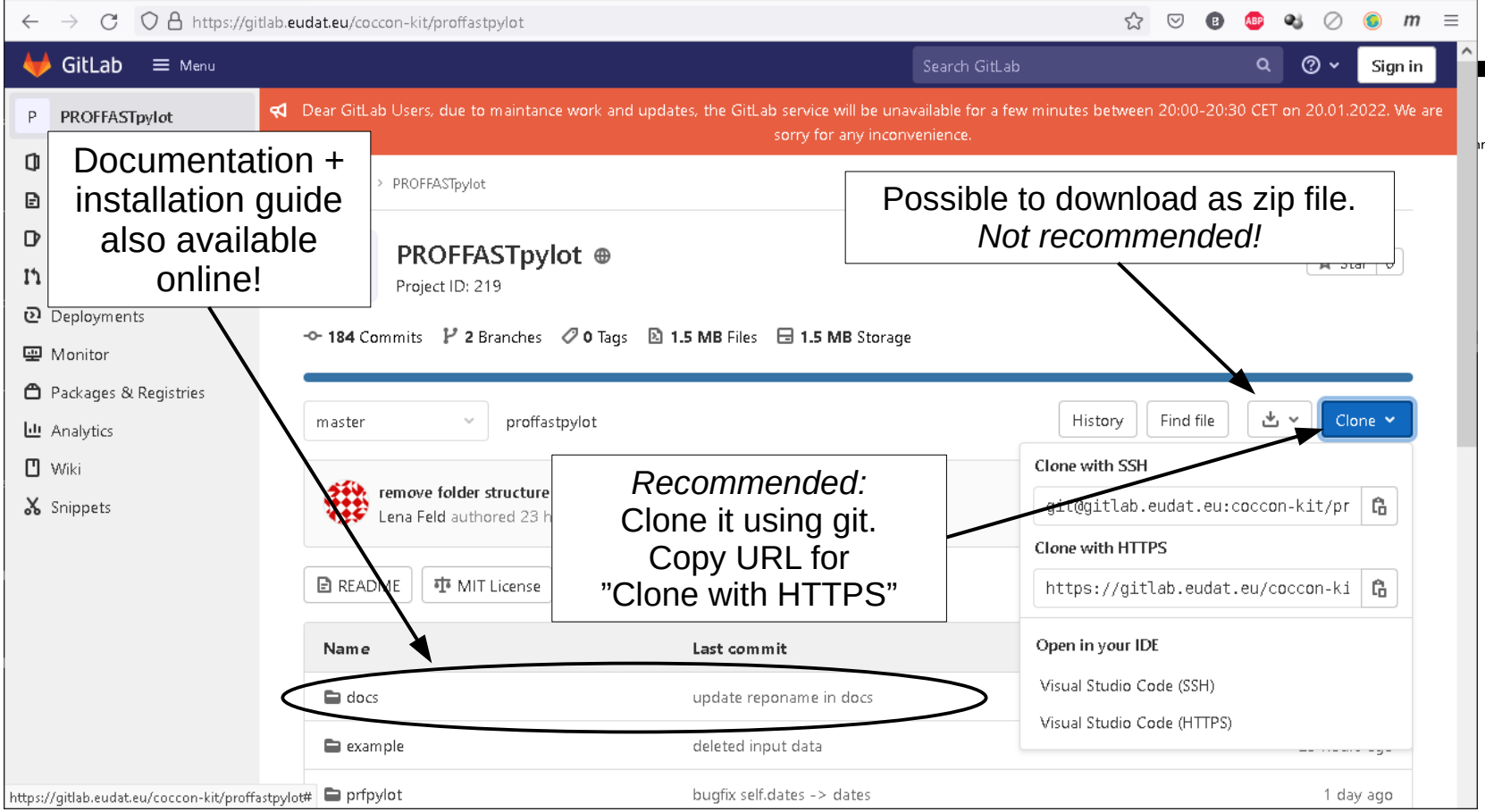
- Support of GGG2020 map files
- Select start- and end date of processing
- Identifies days with corrupt interferograms
- ILS values are read automatically
- Usable with Windows and Linux

User friendly additions

- Paths for input data and results can be chosen freely
- Merged result file with local time and UTC
- Possibility to store frequently occurring coordinates in file
- Option to start with spectra
- Source code can be extended to read different pressure input files

Installation: Where to find PROFFASTpylot

- Available at https://gitlab.eudat.eu/coccon-kit/proffastpylot



- Available at https://gitlab.eudat.eu/coccon-kit/proffastpylot

Install with Git (recommended)

1) If you do not have Git. Download and more information: https://git-scm.com/
2) Navigate to a folder where you want to install PROFFASTpylot
3) Open Git-Bash
4) Enter “git clone https://gitlab.eudat.eu/coccon-kit/proffastpylot.git”

Institute of Meteorology and Climate Research
COllaborative **C**arbon **C**olumn **O**bserving **N**etwork

Insta

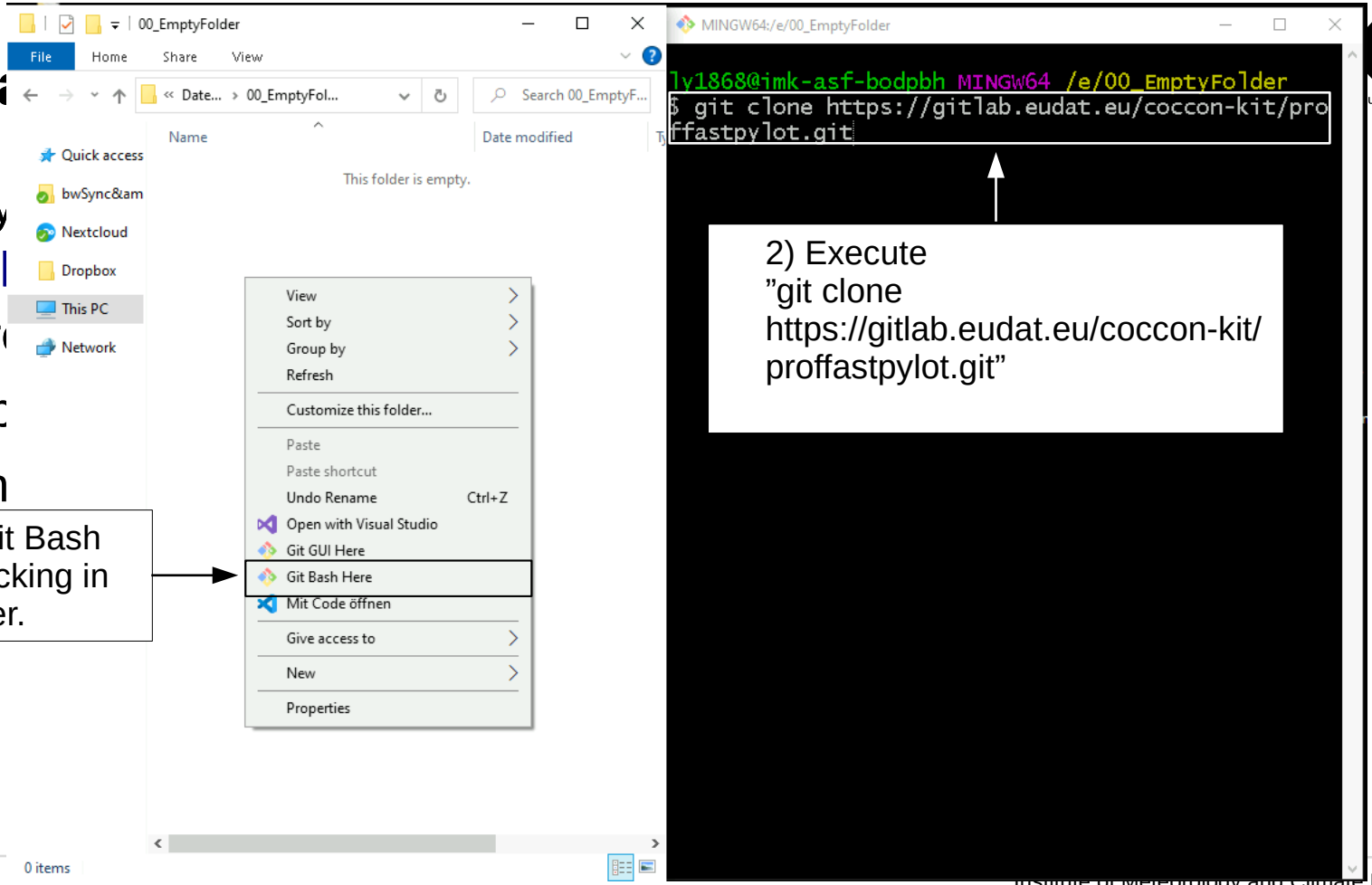
1) If y
htt

2) Cr

3) Op

4) En

1) Open Git Bash by right-clicking in the explorer.



2) Execute
”git clone https://gitlab.eudat.eu/coccon-kit/proffastpylot.git”

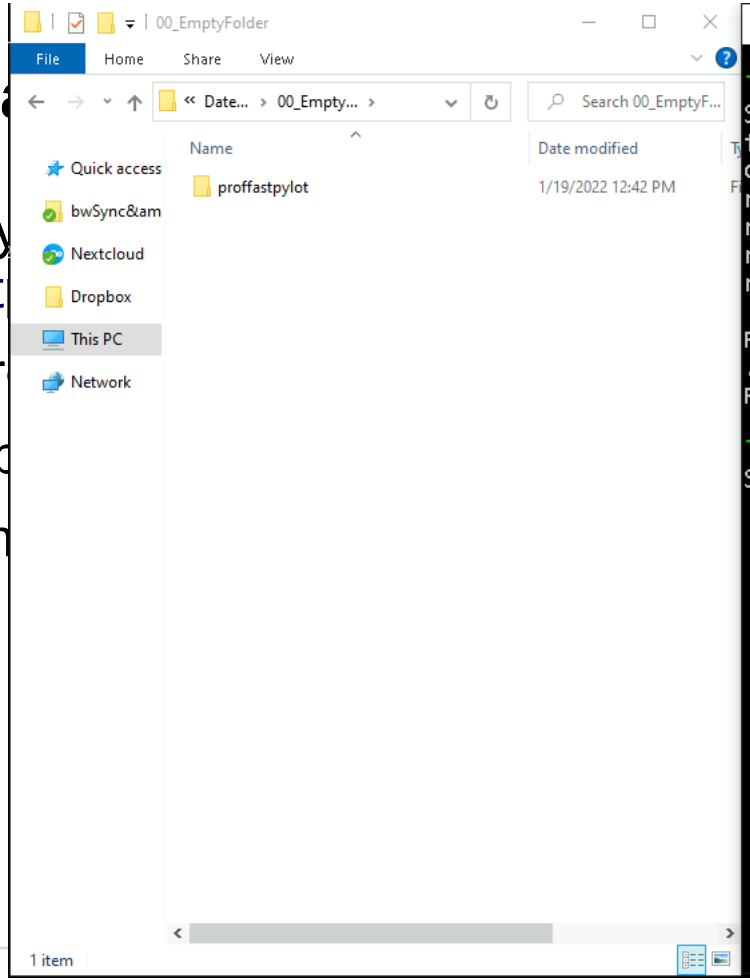
Insta

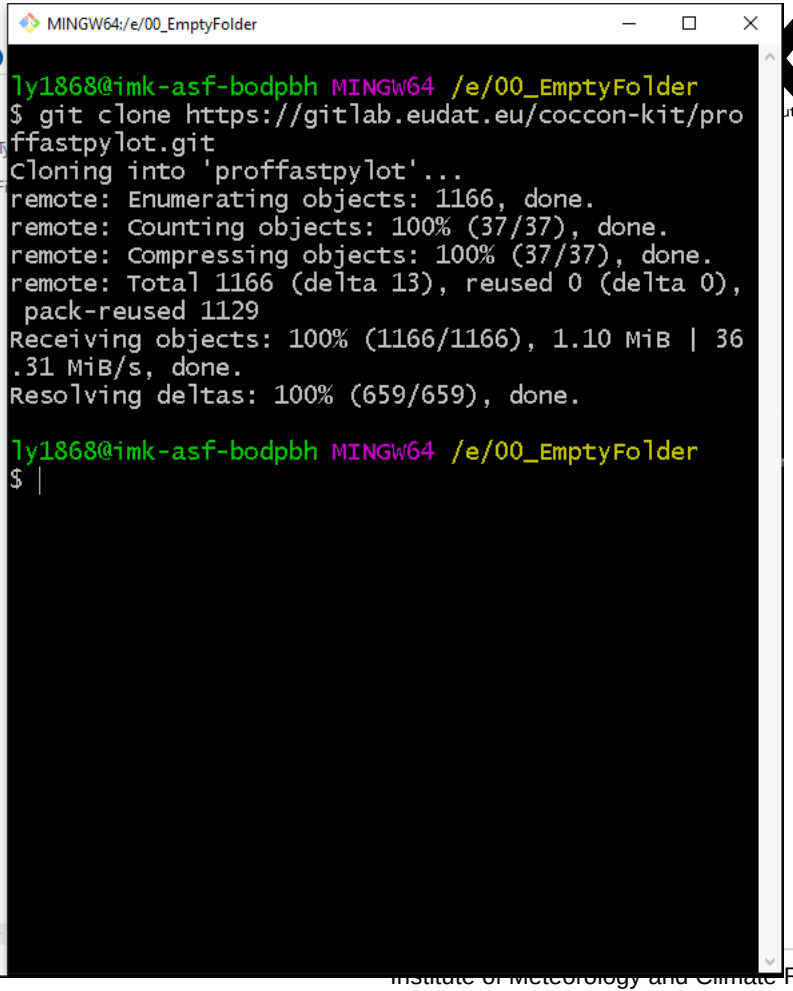
1) If y
htt

2) Cr

3) Op

4) En

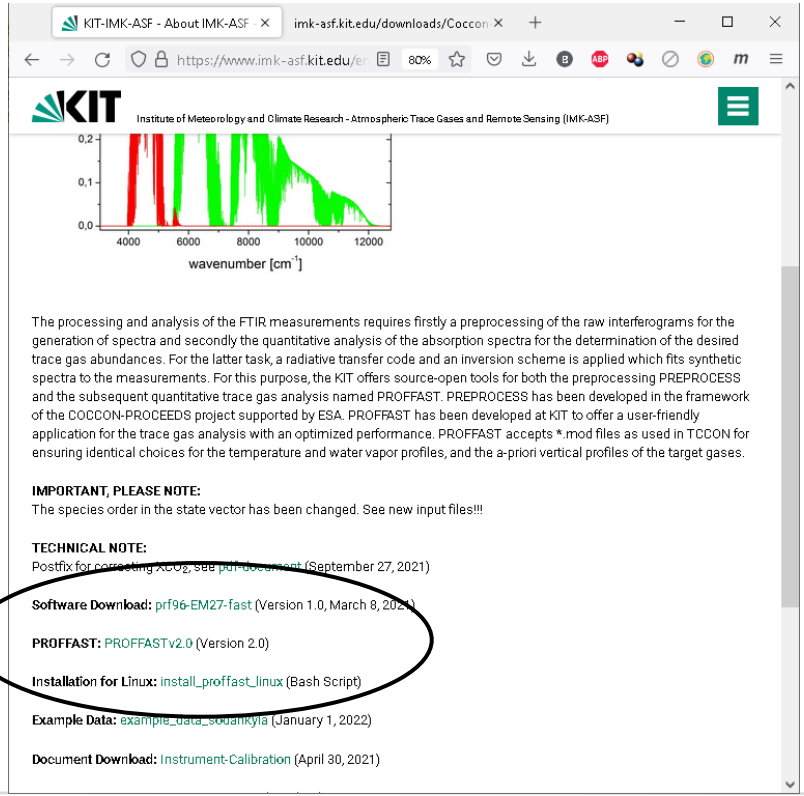


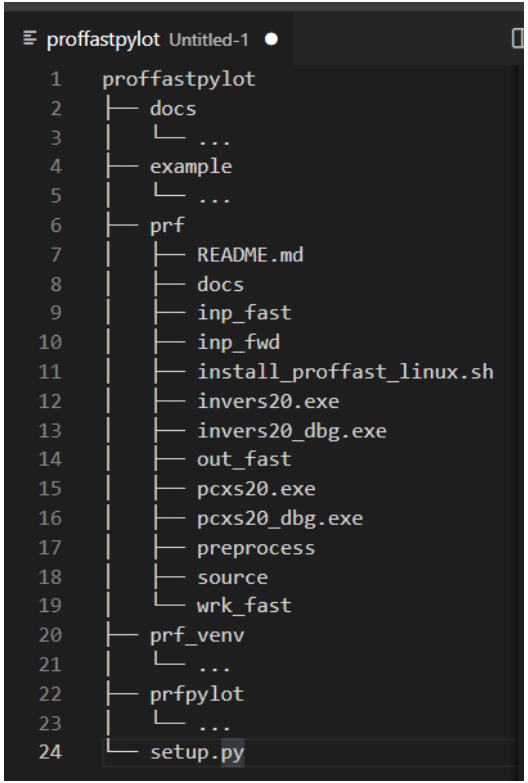


Installation: Download PROFFAST v 2.0

- Go to https://www.imk-asf.kit.edu/english/3225.php
- Download PROFFASTv2.0
- Extract zip file and rename folder to prf
- Move it to `\proffastpylot\` such that the path to e.g. invers20.exe is `\proffastpylot\prf\invers20.exe`

Installation: Download PROFFAST v 2.0





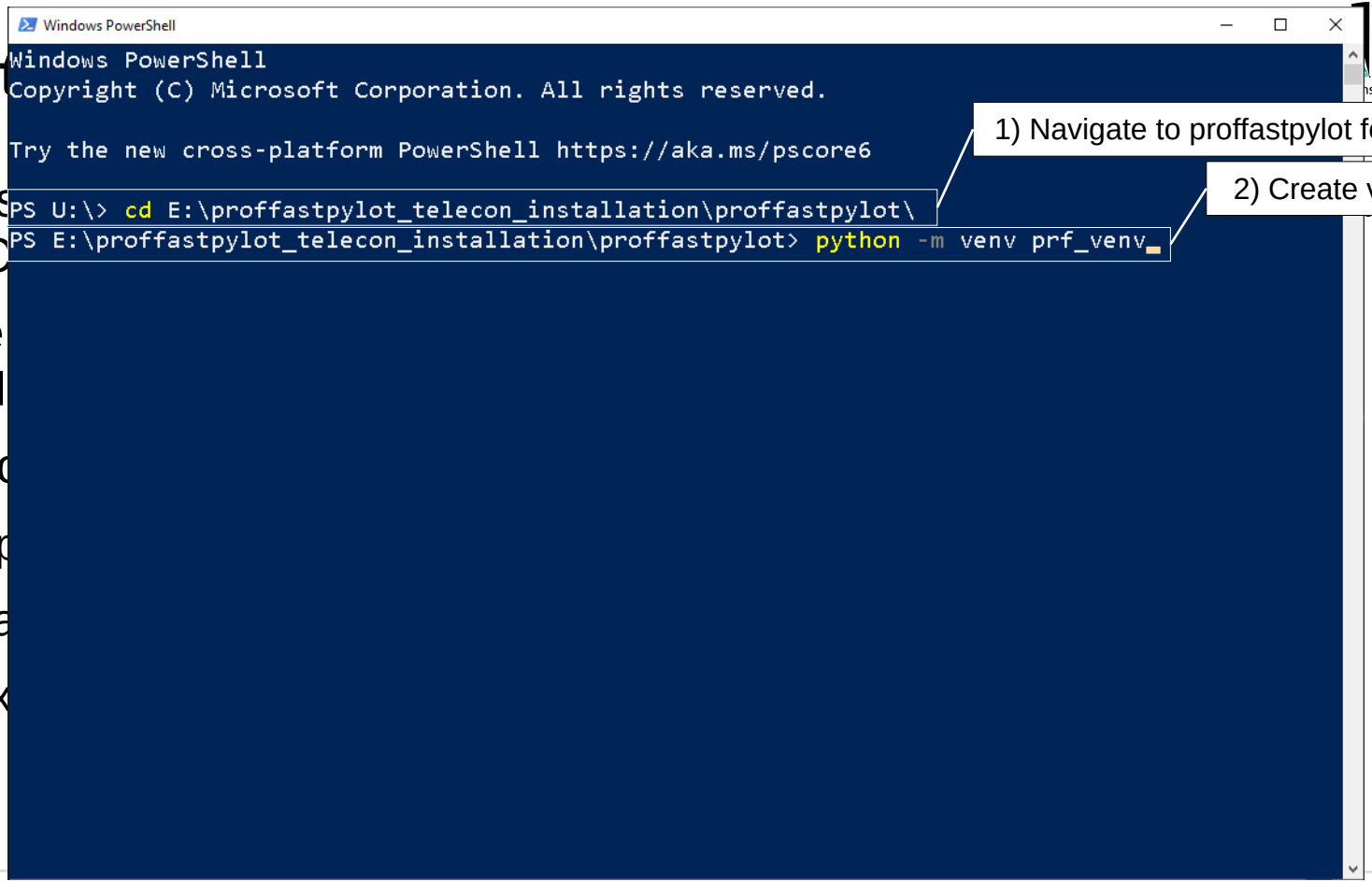
Create a python virtual environment

- We strongly recommend to use a virtual environment (venv) to run the PROFFASTpylot
- A venv is an isolated python environment
  → No interference with other python packages
- For creation:
  - Open a PowerShell/cmd
  - Navigate to the folder “`proffastpylot`”
  - Execute “`python -m venv prf_venv`”

Institute of Meteorology and Climate Research
COllaborative **C**arbon **C**olumn **O**bserving **N**etwork

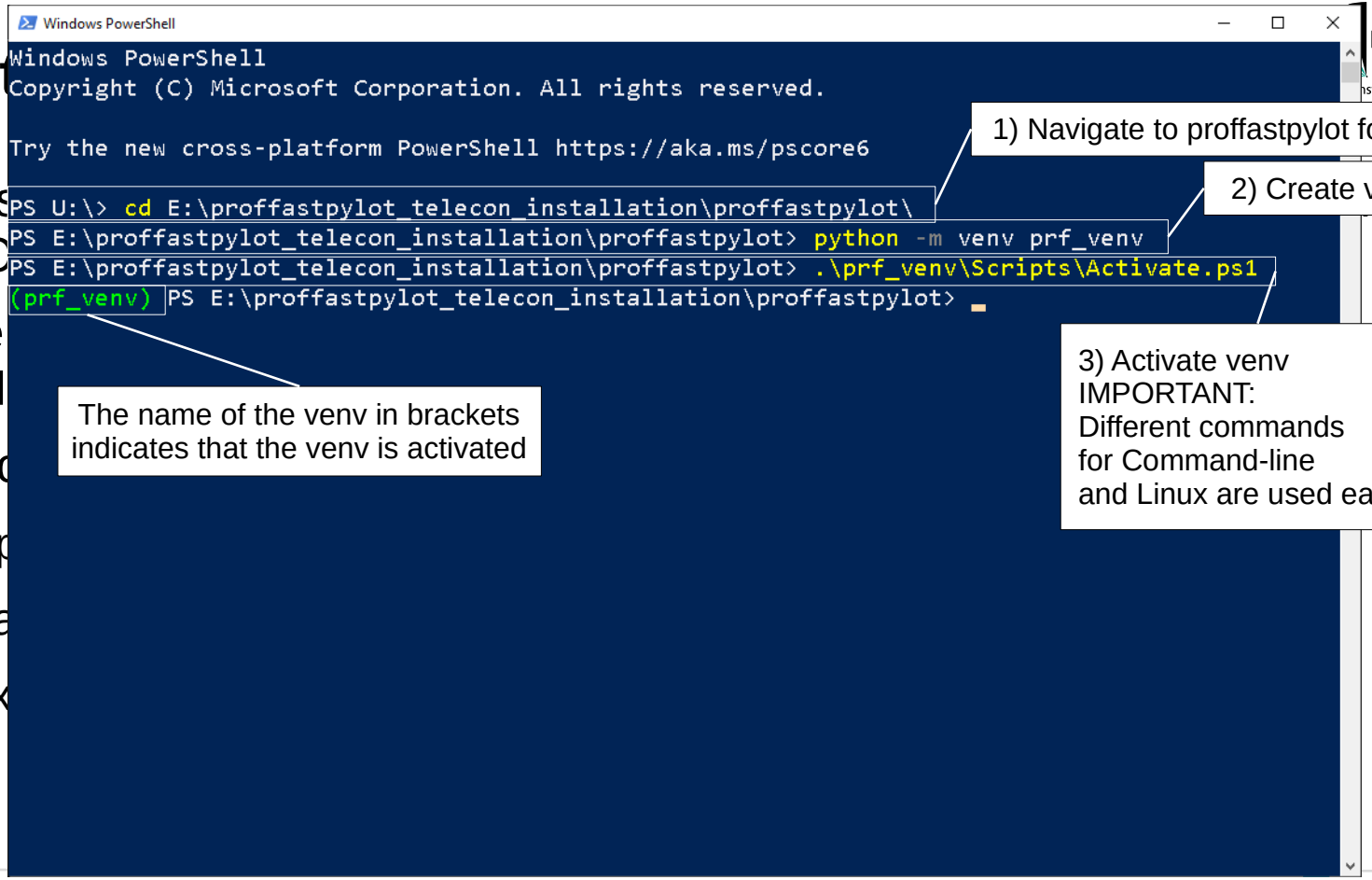
Creat

- We s
  PRO
- A ve
  → N
- For c
  - Op
  - Na
  - Ex



Creat

- We s
  PRC
- A ve
  → N
- For c
  - Op
  - Na
  - Ex

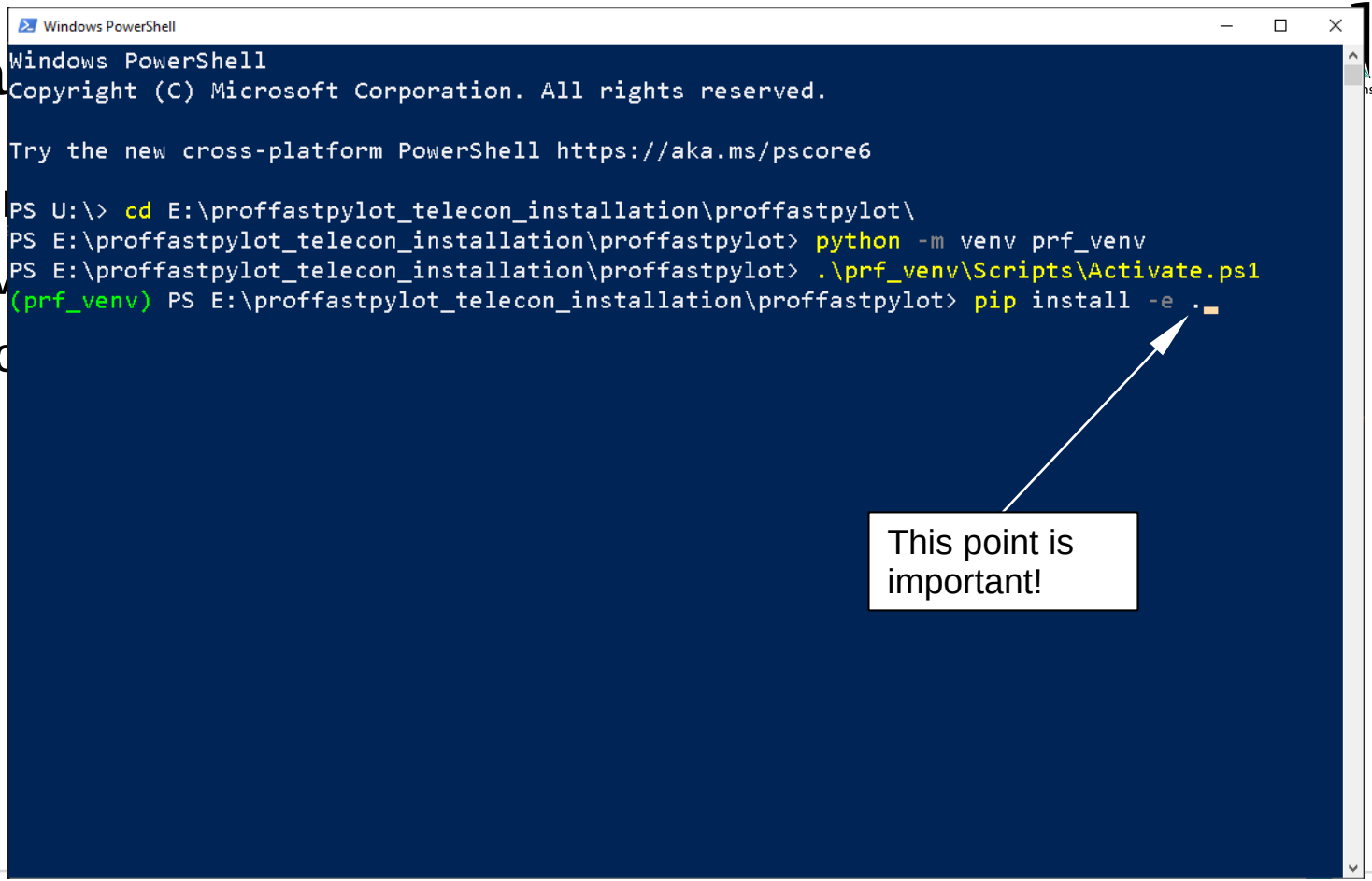


Installation of PROFFASTpylot

- Open a console and navigate to the `proffastpylot` folder
- Activate the virtual environment
- Execute `pip install -e .`

Insta

- Oper
- Activ
- Exec



Insta

- Oper
- Activ
- Exec

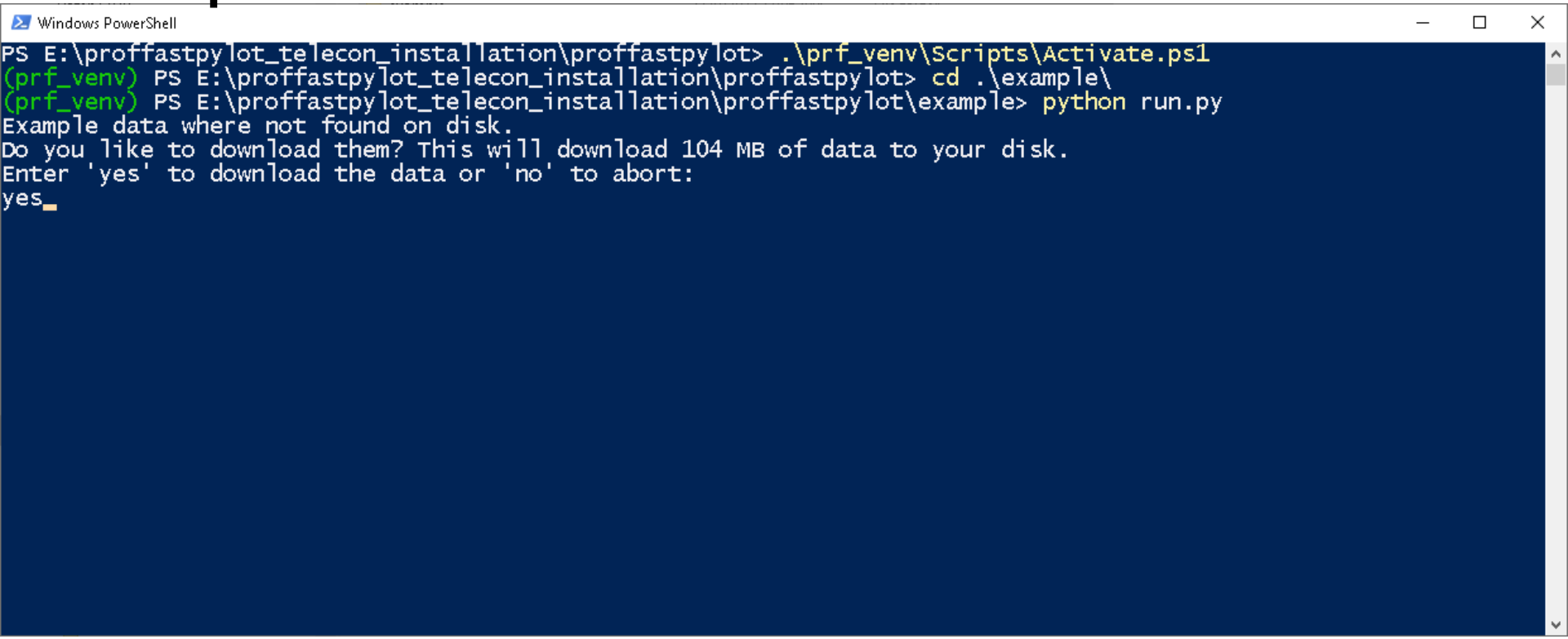
Windows PowerShell

```
  Using cached pandas-1.3.5-cp38-cp38-win_amd64.whl (10.2 MB)
Collecting PyYAML
  Using cached PyYAML-6.0-cp38-cp38-win_amd64.whl (155 kB)
Collecting timezonefinder
  Using cached timezonefinder-5.2.0-py36.py37.py38-none-any.whl (43.0 MB)
Collecting pytz
  Using cached pytz-2021.3-py2.py3-none-any.whl (503 kB)
Collecting fortranformat
  Using cached fortranformat-1.1.1.tar.gz (22 kB)
Collecting python-dateutil>=2.7.3
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting numpy>=1.17.3; platform_machine != "aarch64" and platform_machine != "arm64" and
python_version < "3.10"
  Using cached numpy-1.22.1-cp38-cp38-win_amd64.whl (14.7 MB)
Collecting six>=1.5
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Using legacy setup.py install for fortranformat, since package 'wheel' is not installed.
Installing collected packages: wheel, six, python-dateutil, numpy, pytz, pandas, PyYAML, tim
ezonefinder, fortranformat, prfPylot
    Running setup.py install for fortranformat ... done
  Running setup.py develop for prfPylot
Successfully installed PyYAML-6.0 fortranformat-1.1.1 numpy-1.22.1 pandas-1.3.5 prfPylot pyt
hon-dateutil-2.8.2 pytz-2021.3 six-1.16.0 timezonefinder-5.2.0 wheel-0.37.1
WARNING: You are using pip version 20.1.1; however, version 21.3.1 is available.
You should consider upgrading via the 'e:\proffastpylot_telecon_installation\proffastpylot\p
rf_venv\scripts\python.exe -m pip install --upgrade pip' command.
(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot>
```

Test of the Installation: Execute the Sodankyla-Example

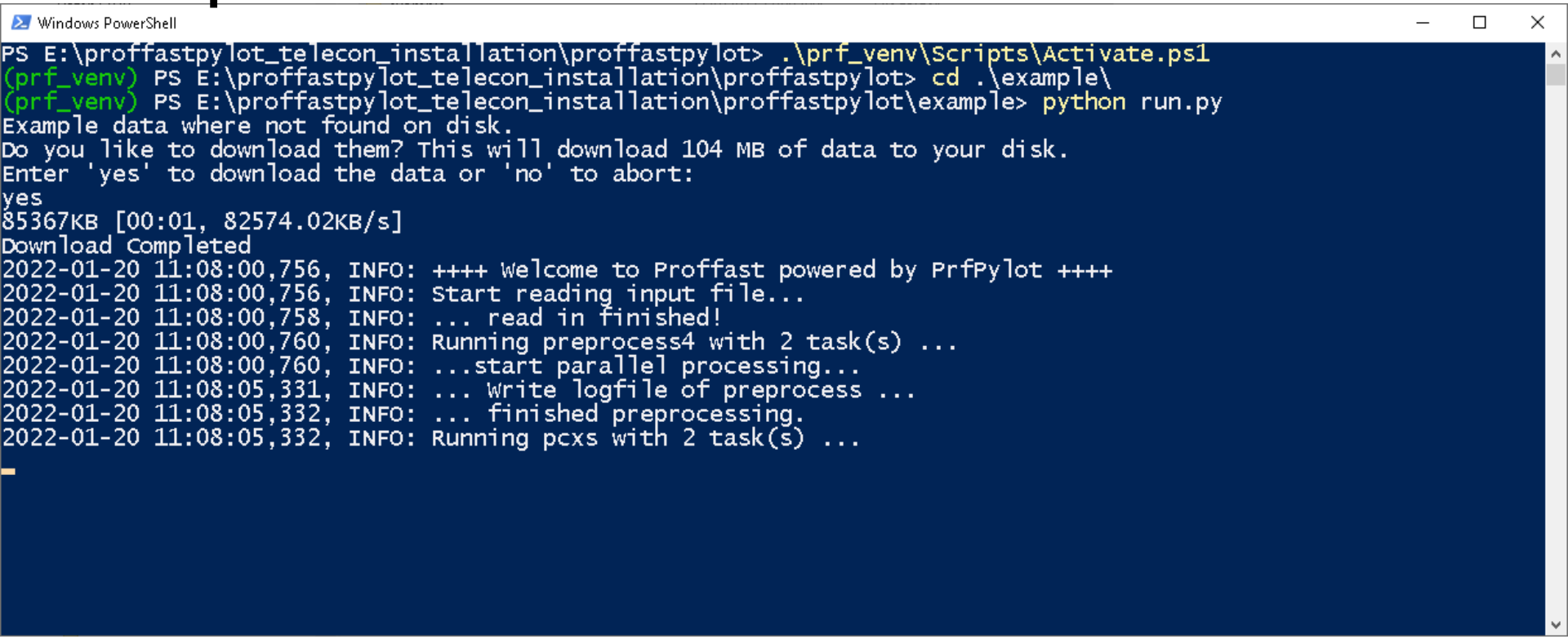
- The PROFFASTpylot comes with an out-of-the-box example which can be used for testing.
- It can be executed by running `run.py` in `proffastpylot\example`
  - Downloads the example interferograms and input data automatically
  - Automatically generates an input file with the required settings
  - Automatically generates folders for the output data
- Do not forget to activate your virtual environment

Test of Installation: Execute Sodankyla-Example



Test of Installation: Execute Sodankyla-Example

Karlsruhe Institute of Technology

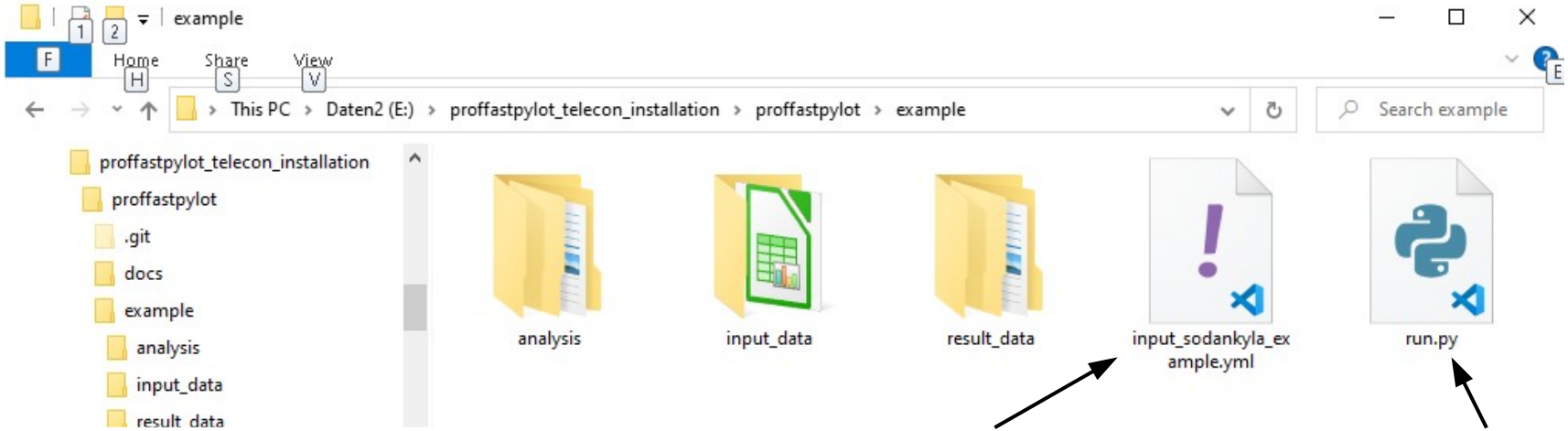


Test of Installation: Execute Sodankyla-Example

Windows PowerShell

```
Do you like to download them? This will download 104 MB of data to your disk.
Enter 'yes' to download the data or 'no' to abort:
yes
85367KB [00:01, 82574.02KB/s]
Download Completed
2022-01-20 11:08:00,756, INFO: ++++ Welcome to Proffast powered by PrfPylot ++++
2022-01-20 11:08:00,756, INFO: Start reading input file...
2022-01-20 11:08:00,758, INFO: ... read in finished!
2022-01-20 11:08:00,760, INFO: Running preprocess4 with 2 task(s) ...
2022-01-20 11:08:00,760, INFO: ...start parallel processing...
2022-01-20 11:08:05,331, INFO: ... Write logfile of preprocess ...
2022-01-20 11:08:05,332, INFO: ... finished preprocessing.
2022-01-20 11:08:05,332, INFO: Running pcxs with 2 task(s) ...
2022-01-20 11:10:28,581, INFO: ... Write logfile of pcsx ...
2022-01-20 11:10:28,583, INFO: Running inv with 2 task(s) ...
2022-01-20 11:10:30,544, INFO: ... Write logfile of inv ...
2022-01-20 11:10:30,545, INFO: Move results to final output folder
2022-01-20 11:10:30,722, INFO: Sucessfully wrote the combined invparams to e:\proffastpylot_telecon_installation\proffastpylot\example\result_data\results_sodankyla\Sodankyla_SN039_20170608_20170609\combined_invparms_Sodankyla_StartStopDates20170608_20170609.csv
2022-01-20 11:10:30,722, INFO: Delete abscos-bin files
2022-01-20 11:10:30,758, INFO: Move input files
2022-01-20 11:10:30,763, INFO: Moved the general logfile to the result/logfiles folder
(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot\example>
```

How to Run: Sodankyla Example



Three different data folders:

1) `input_data`: contains e.g. interferograms
2) `analysis`: contains spectra generated by preprocess
3) `result_data`: contains the results and logs

For each site and spectrometer one input file is needed, containing ALL settings and information about the side.

To run a site one single python script is needed

run.py

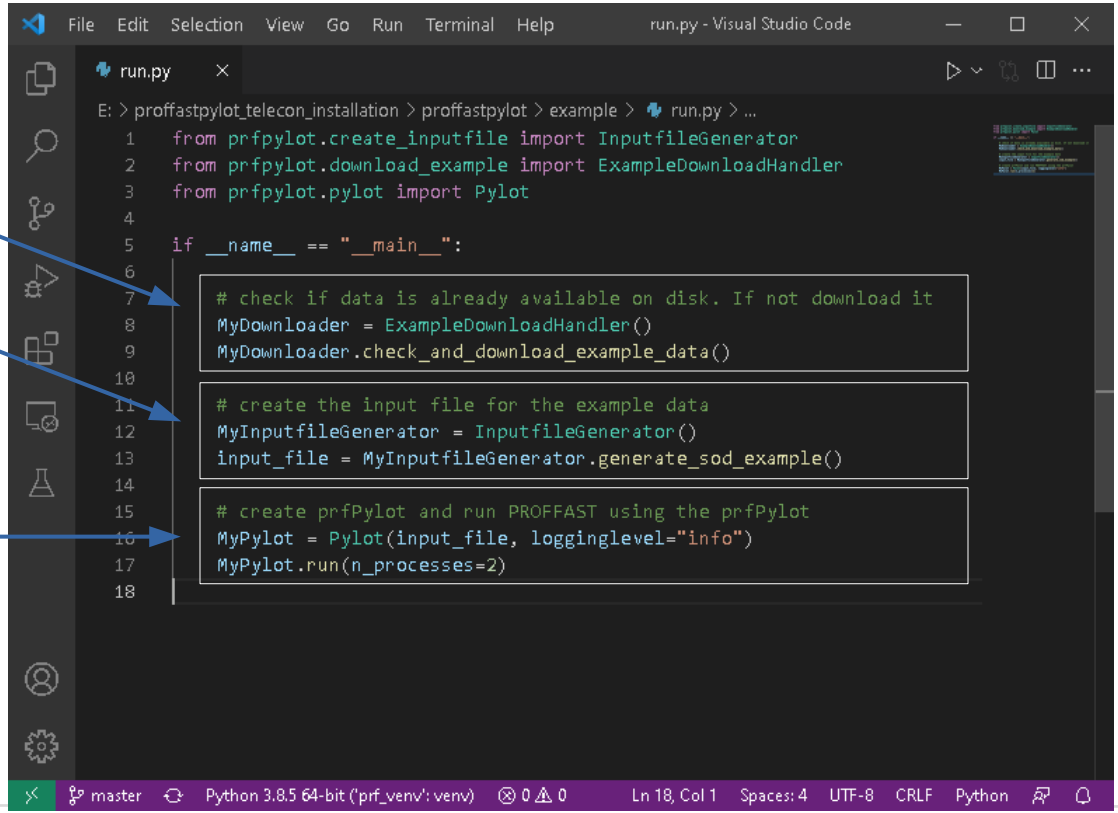
Only 17 lines of code to execute the whole example!

Only needed in Example: Download the interferograms and input data

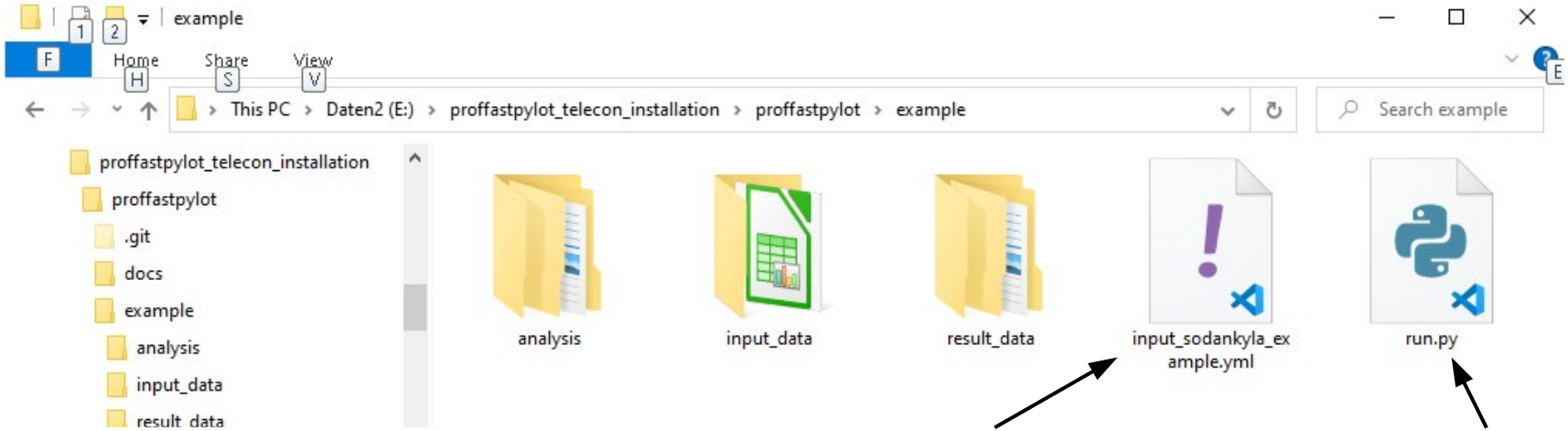
Input file generator: Here in the special use case of generating the example input-file. It also provides a interactive wizard to generate input files.

Pylot class:

- Working horse of **PROFFASTpylot**
- `input_file` contains the path to `sondankyla_example_input.yml`
- run method does all the work for a standard evaluation of EM27/SUN data (supporting multiprocessing!)



How to Run: Sodankyla Example



Three different data folders:

1) `input_data`: contains e.g. interferograms
2) `analysis`: contains spectra generated by preprocess
3) `result_data`: contains the results and logs

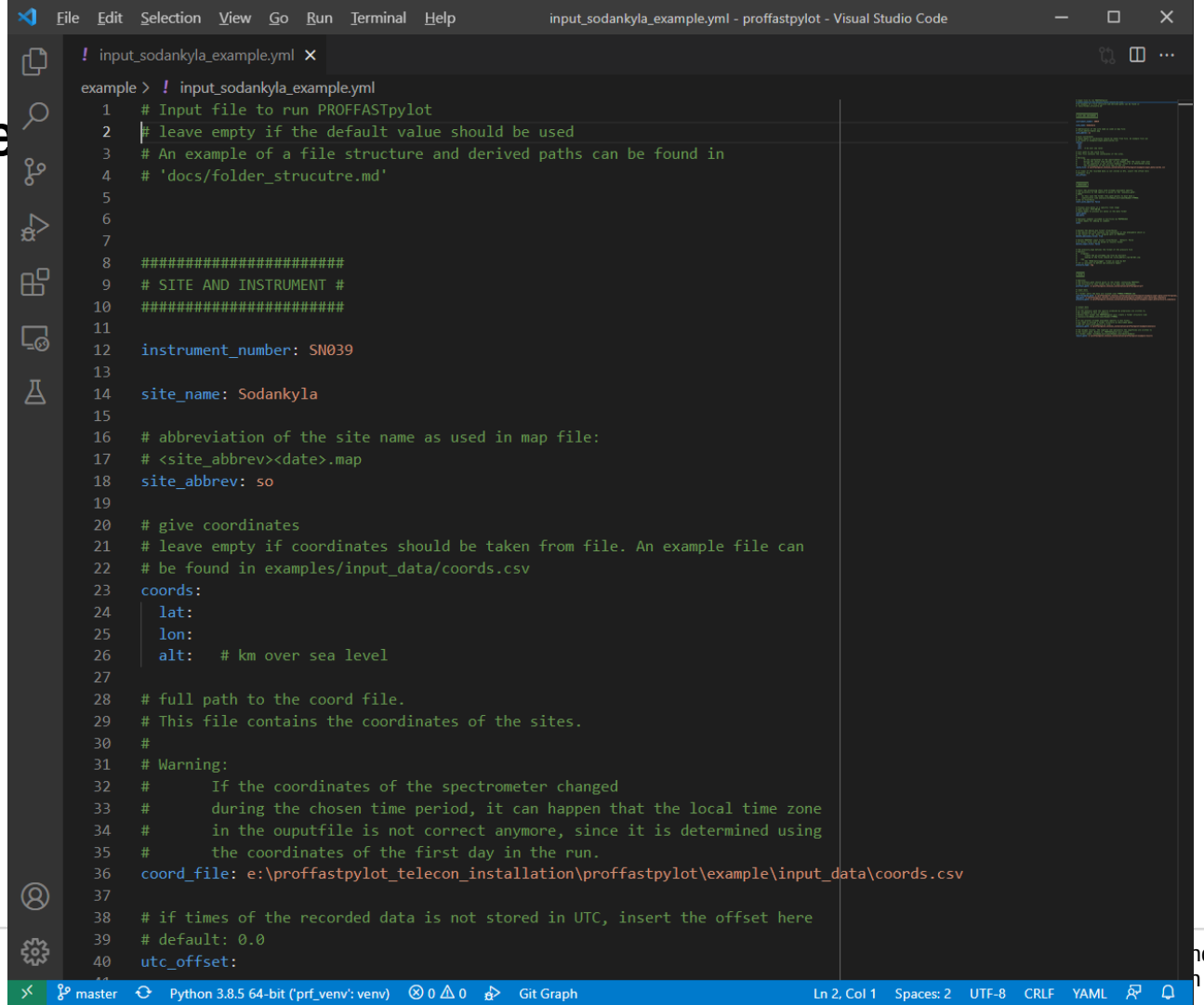
For each site and spectrometer one input file is needed, containing ALL settings and information about the side.

To run a site one single python script is needed

input_sodankyla_example.yml

- Contains all information and settings to one site and one spectrometer
- 3 Blocks:
  - Site and Instrument
  - Behavior of PROFFASTpylot
  - Path

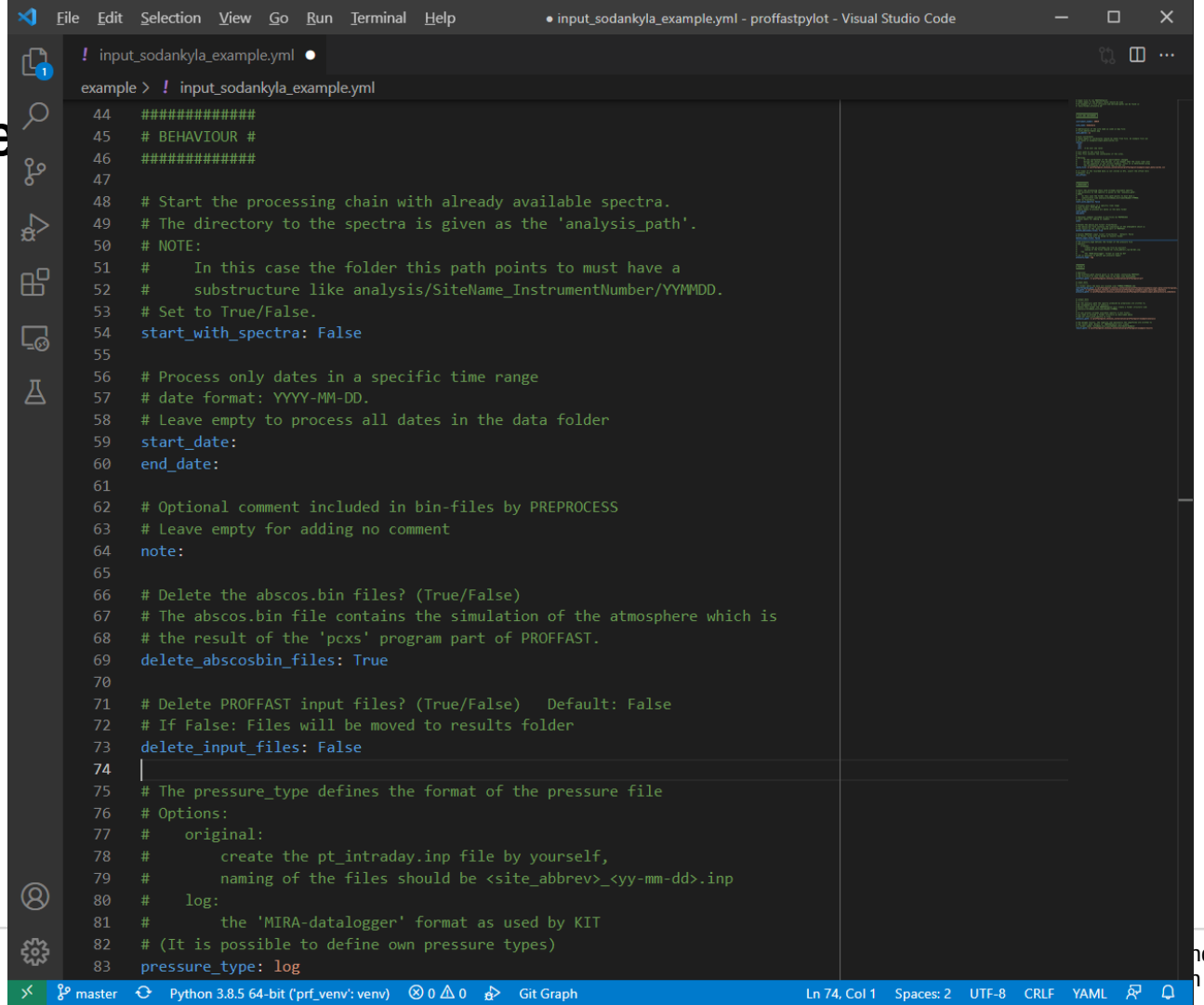
Inputfile



Karlsruhe Institute of Technology

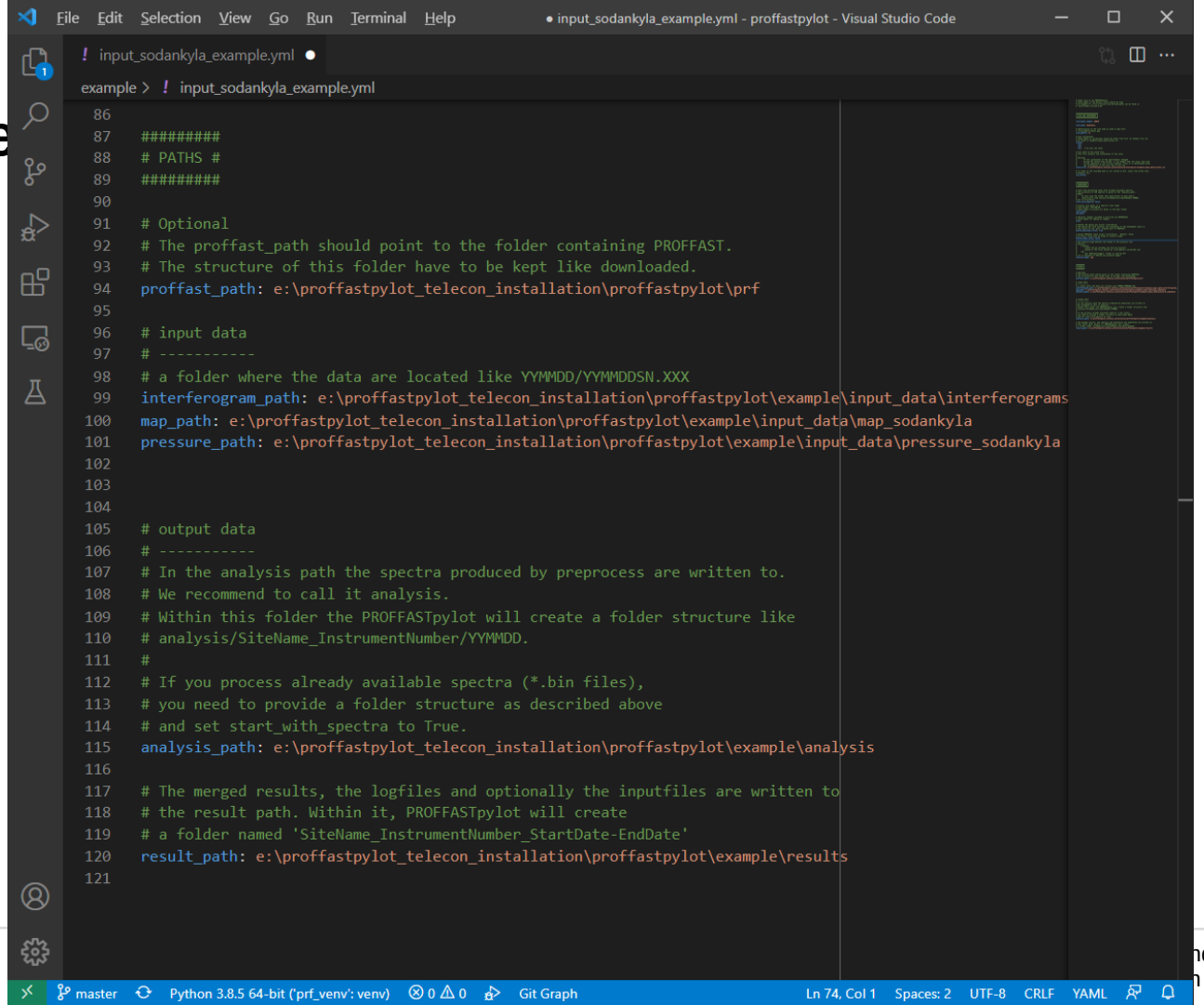
nd Climate Research
n **O**bserving **N**etwork

Inputfile



Karlsruhe Institute of Technology

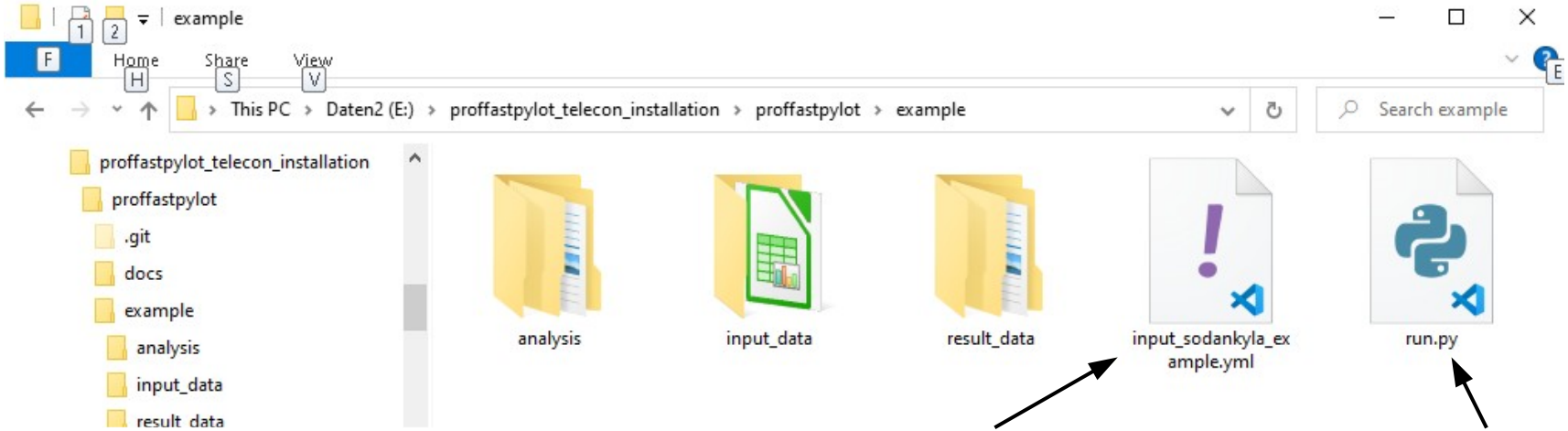
Inputfile



Karlsruhe Institute of Technology

nd Climate Research
n **O**bserving **N**etwork

How to Run: Sodankyla Example



Three different data folders:

1) `input_data`: contains e.g. interferograms
2) `analysis`: contains spectra generated by preprocess
3) `result_data`: contains the results and logs

For each site and spectrometer one input file is needed, containing ALL settings and information about the side.

To run a site one single python script is needed

How to run: General usage of PROFFASTpylot

- For the input file of an arbitrary site:
  - Modify the example manually
  - Use the InputfileGenerator

How to run: General usage of PROFFASTpylot

Windows PowerShell

```
(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot> cd .\prfpylot\
(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot\prfpylot> ls


    Directory: E:\proffastpylot_telecon_installation\proffastpylot\prfpylot


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/19/2022   5:31 PM                templates
d-----         1/20/2022   8:14 AM                __pycache__
-a----         1/19/2022  12:42 PM          15753 create_inputfile.py
-a----         1/19/2022   5:31 PM           2533 download_example.py
-a----         1/19/2022   5:31 PM           9470 filemover.py
-a----         1/19/2022  12:42 PM           5846 ILSList.csv
-a----         1/19/2022  12:42 PM          25559 prepare.py
-a----         1/19/2022  12:42 PM           2966 pressure.py
-a----         1/19/2022   5:31 PM          15886 pylot.py
-a----         1/19/2022  12:42 PM              0 __init__.py


(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot\prfpylot> python .\create_inputfile.py
```

How to run: General usage of PROFFASTpylot

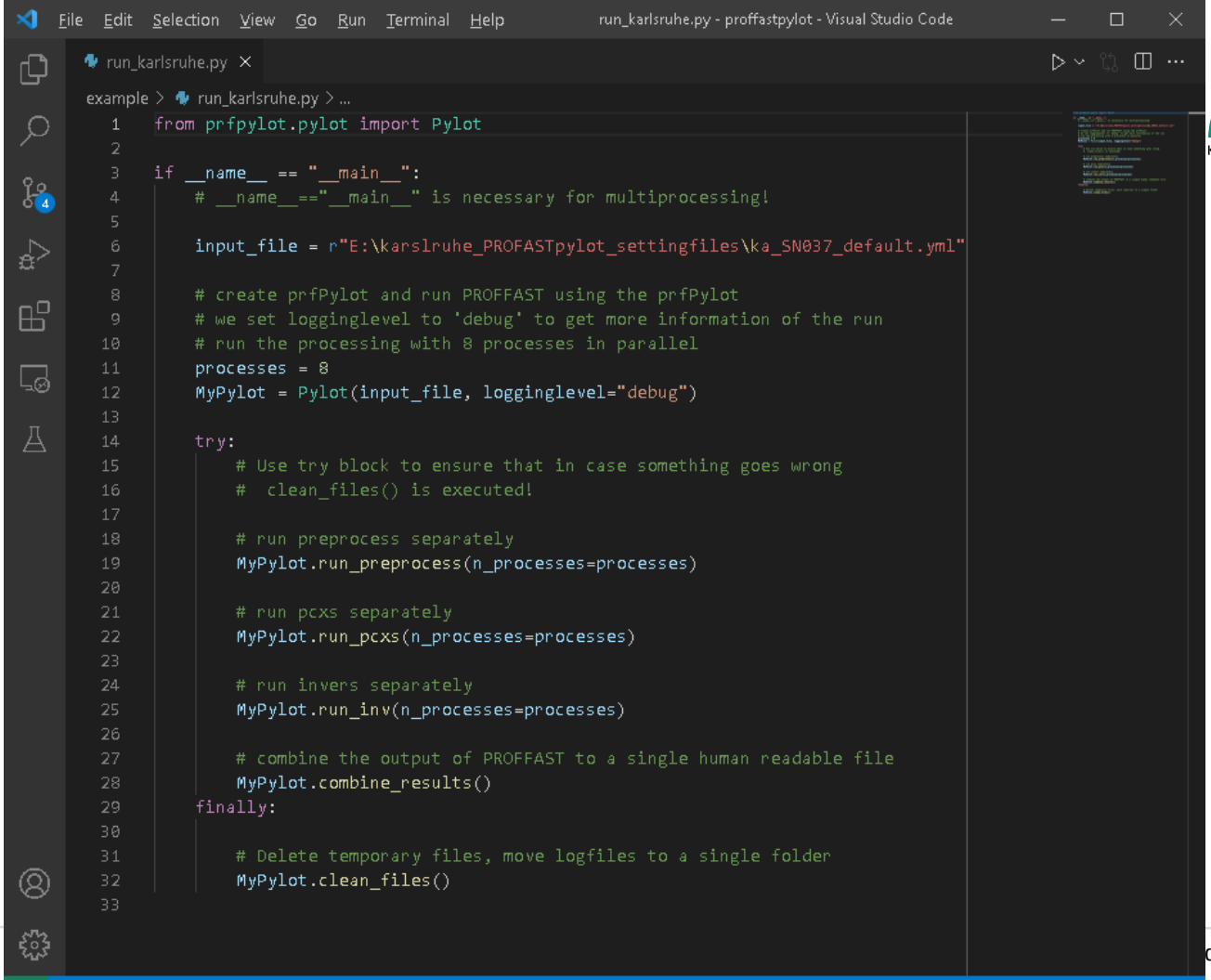
Windows PowerShell

```
(prf_venv) PS E:\proffastpylot_telecon_installation\proffastpylot\prfpylot> python .\create_inputfile.py
################################################################################
#                    Welcome to the inputfile generator.                       #
#                    This tool is part of the prfPylot.                        #
################################################################################


This tool helps you to create the inputfile of your PrfPylot to run the files
of one device at one location.
Please generate a unique inputfile (manually or by using this tool) for each
instrument and each location. Please note, that files like pressure and *.map
files can be shared among various intruments at the same site.

To start the configuration press enter
############ Data paths ############

Please give the path to the PROFFAST executable.
In this folder the PROFFAST executable have to be located.
As the automatic detected default value the following path is used:
E:\proffastpylot_telecon_installation\proffastpylot\prf
To use this path do not enter anything and press enter:
```

How to run: General usage of PROFFASTpylot

- For the input file of an arbitrary site:
  - Modify the example manually
  - Use the InputfileGenerator

- Run PROFFASTpylot
  - Execute the `run()` method of the Pylot class
  - Or: Execute the single steps separately



Karlsruhe Institute of Technology

d Climate Research
Observing **N**etwork

How to run: General usage of PROFFASTpylot

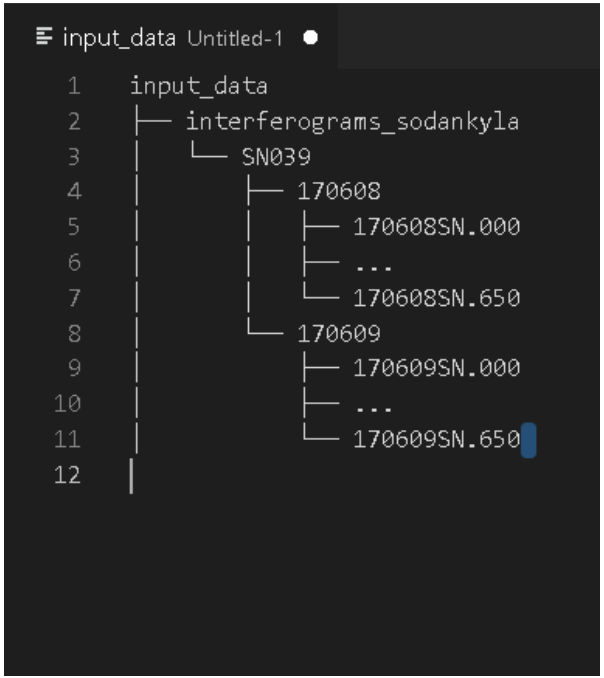
- For an arbitrary side Input file:
  - Modify the example manually
  - Use the InputfileGenerator

- Run PROFFASTpylot
  - Execute the `run` – method of Pylot
  - Execute the single steps separately

Institute of Meteorology and Climate Research
COllaborative **C**arbon **C**olumn **O**bserving **N**etwork

How to run: File structure

- Your interferograms must be saved in folders named `YYMMDD`
- We recommend using the following structure `interferograms_<site>\<instrument_number>\YYMMDD`

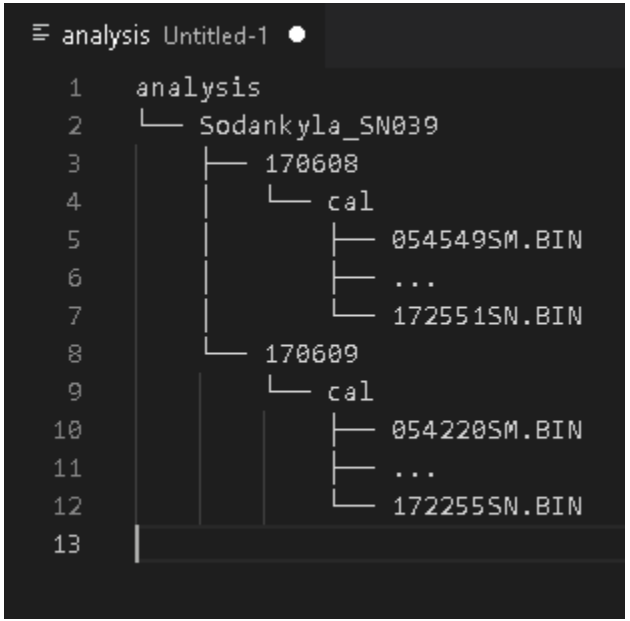


How to run: Starting from spectra

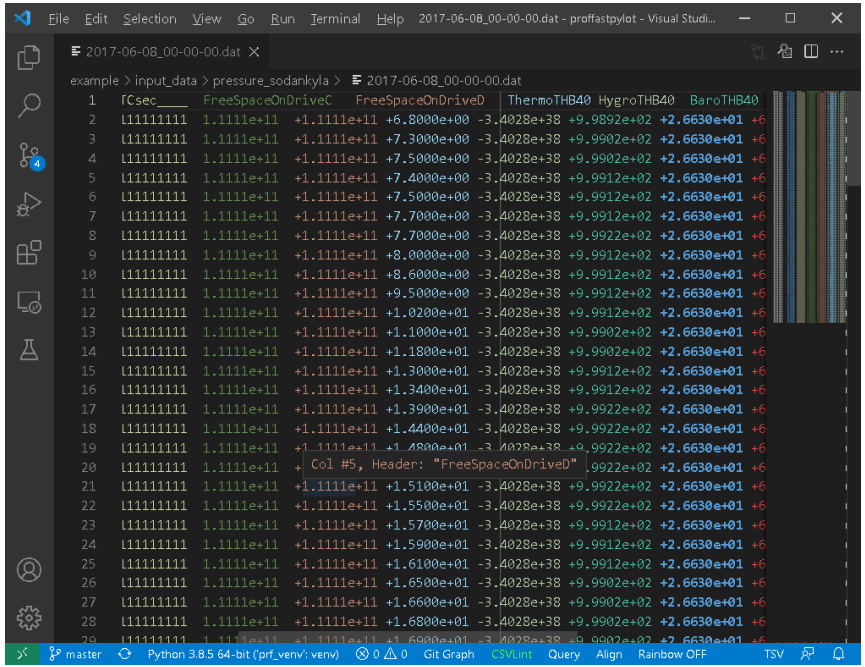
- PROFFASTpylot gives the possibility to process spectra directly
- The analysis path must have a structure as shown on the right hand side:

  `analysis\<site>_<instrument_number>\YYMMDD\cal`

- In ‘normal’ processing mode PROFFASTpylot automatically generates this structure of the `analysis` folder



How to run: Add custom pressure type



- Beyond the scope of this presentation
- Procedure will be changed
- Explained in `docs/pressure_input.md`

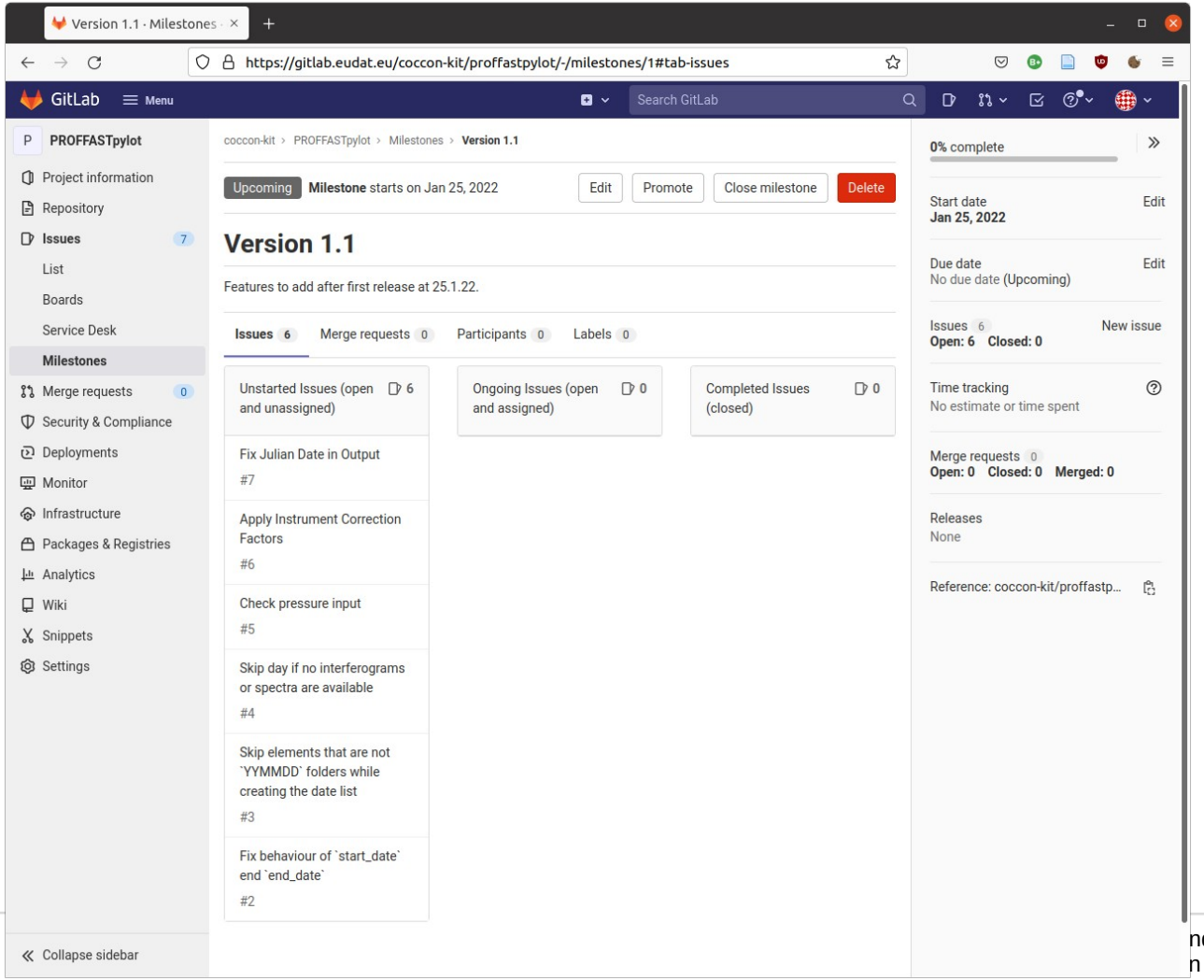
Prerequisites on Hardware

- Free space on Hard-drive: Per day approximately 2,4 GB
  - 2 GB of Interferograms
  - 100 MB of spectra
  - 300 MB of `abscos.bin`-file
- RAM: `pcxs20.exe` needs the most memory: Approximately 700 MB per instance.

Closing words

- First operational Version

Closing


Version 1.1 · Milestones
https://gitlab.eudat.eu/coccon-kit/proffastpylot/-/milestones/1#tab-issues
GitLab
Menu
Search GitLab
PROFFASTpylot
Project information
Repository
Issues 7
List
Boards
Service Desk
Milestones
Merge requests 0
Security & Compliance
Deployments
Monitor
Infrastructure
Packages & Registries
Analytics
Wiki
Snippets
Settings
Collapse sidebar
coccon-kit › PROFFASTpylot › Milestones › Version 1.1
Upcoming Milestone starts on Jan 25, 2022
Edit
Promote
Close milestone
Delete
Version 1.1
Features to add after first release at 25.1.22.
Issues 6
Merge requests 0
Participants 0
Labels 0
Unstarted Issues (open and unassigned) 6
Ongoing Issues (open and assigned) 0
Completed Issues (closed) 0
Fix Julian Date in Output
#7
Apply Instrument Correction Factors
#6
Check pressure input
#5
Skip day if no interferograms or spectra are available
#4
Skip elements that are not `YYMMDD` folders while creating the date list
#3
Fix behaviour of `start_date` end `end_date`
#2
0% complete
Start date
Jan 25, 2022
Edit
Due date
No due date (Upcoming)
Edit
Issues 6
Open: 6 Closed: 0
New issue
Time tracking
No estimate or time spent
Merge requests 0
Open: 0 Closed: 0 Merged: 0
Releases
None
Reference: coccon-kit/proffastp...

Closing words

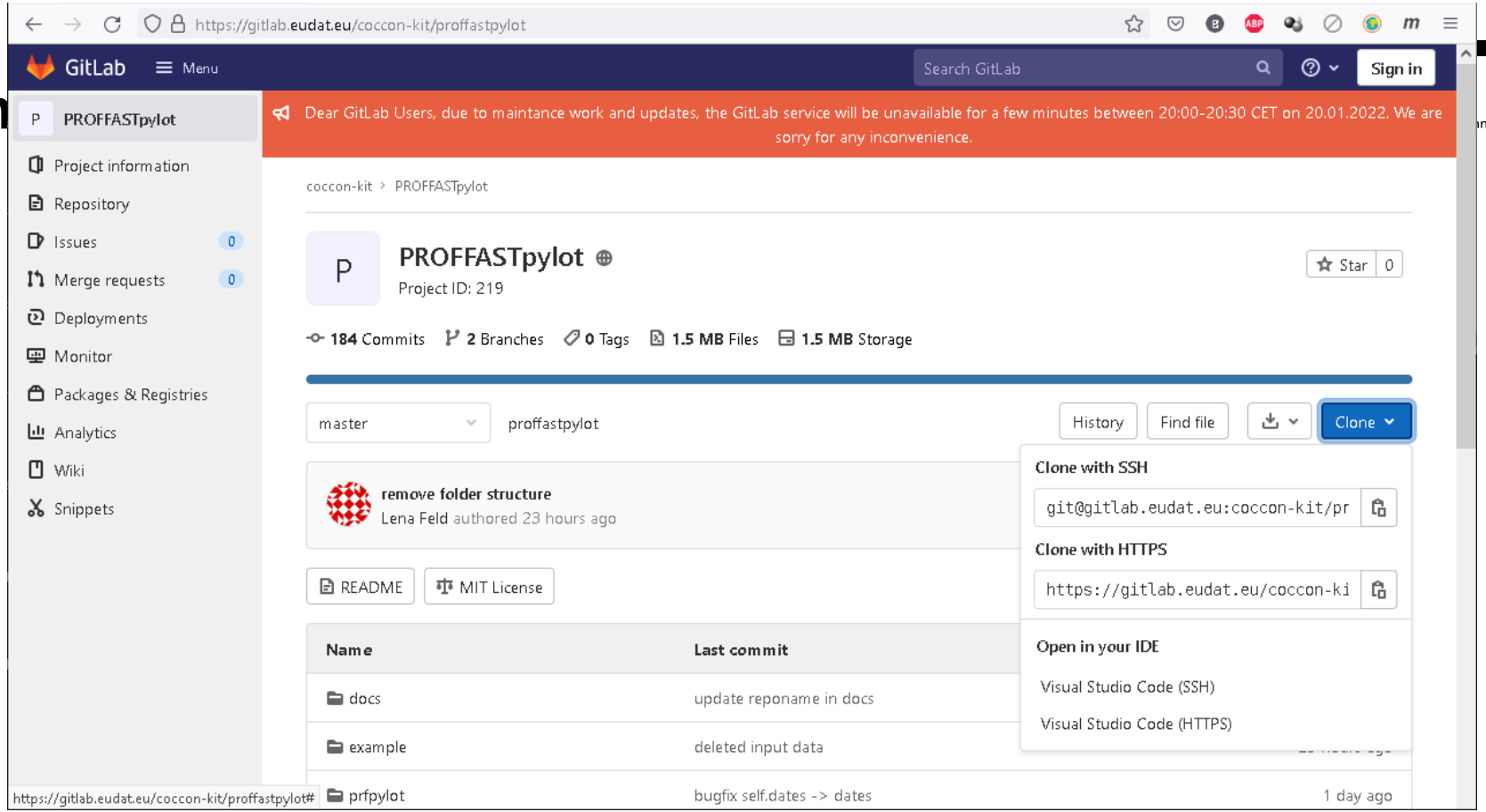
- First operational Version
- If you have **ideas or suggestions** or you find a **bug**
  please contact us (benedikt.herkommer@kit.edu, lena.feld@kit.edu)!
- We can offer an installation workshop

Disclaimer

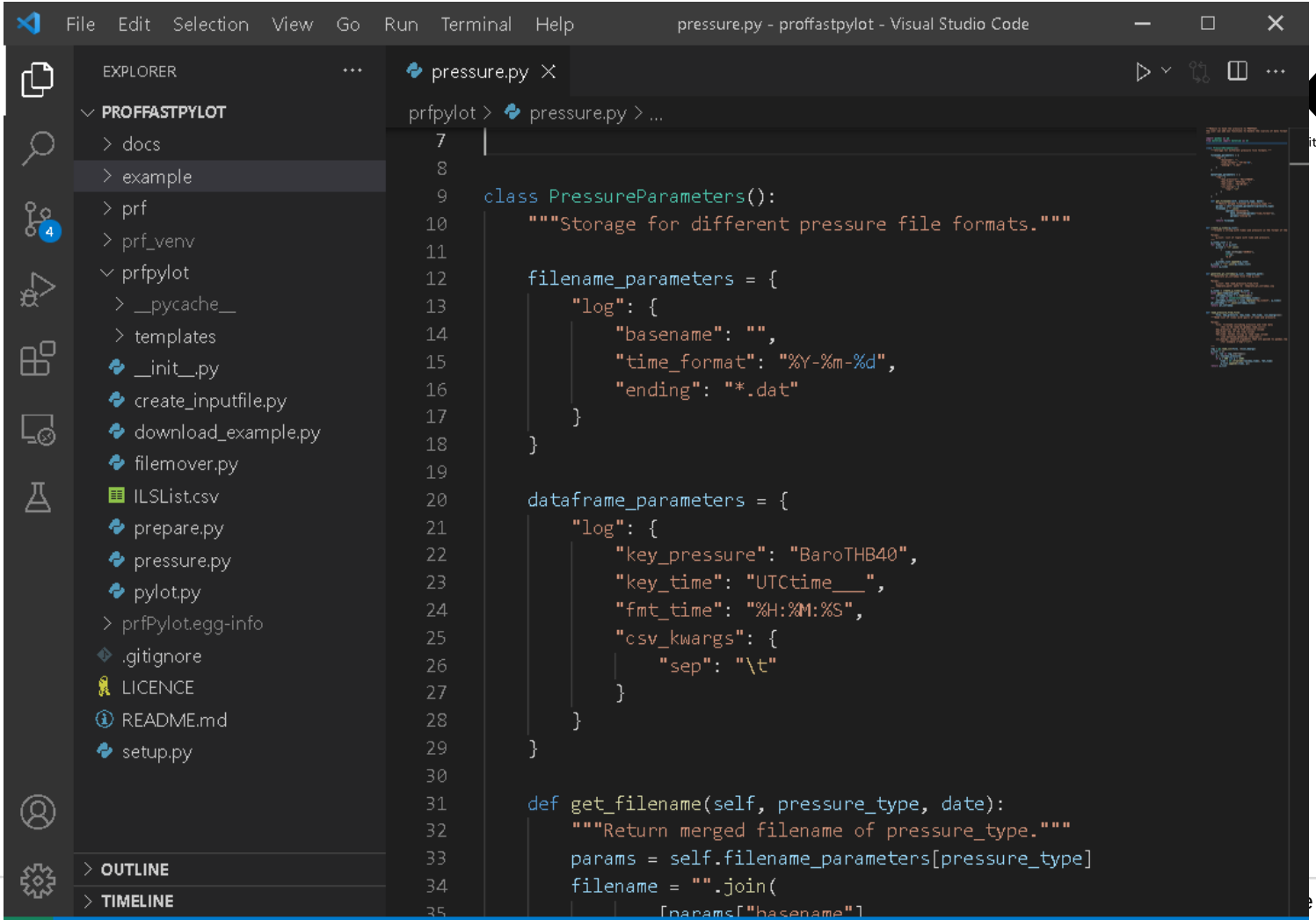
- Preliminary correction factors → not use for publications yet.

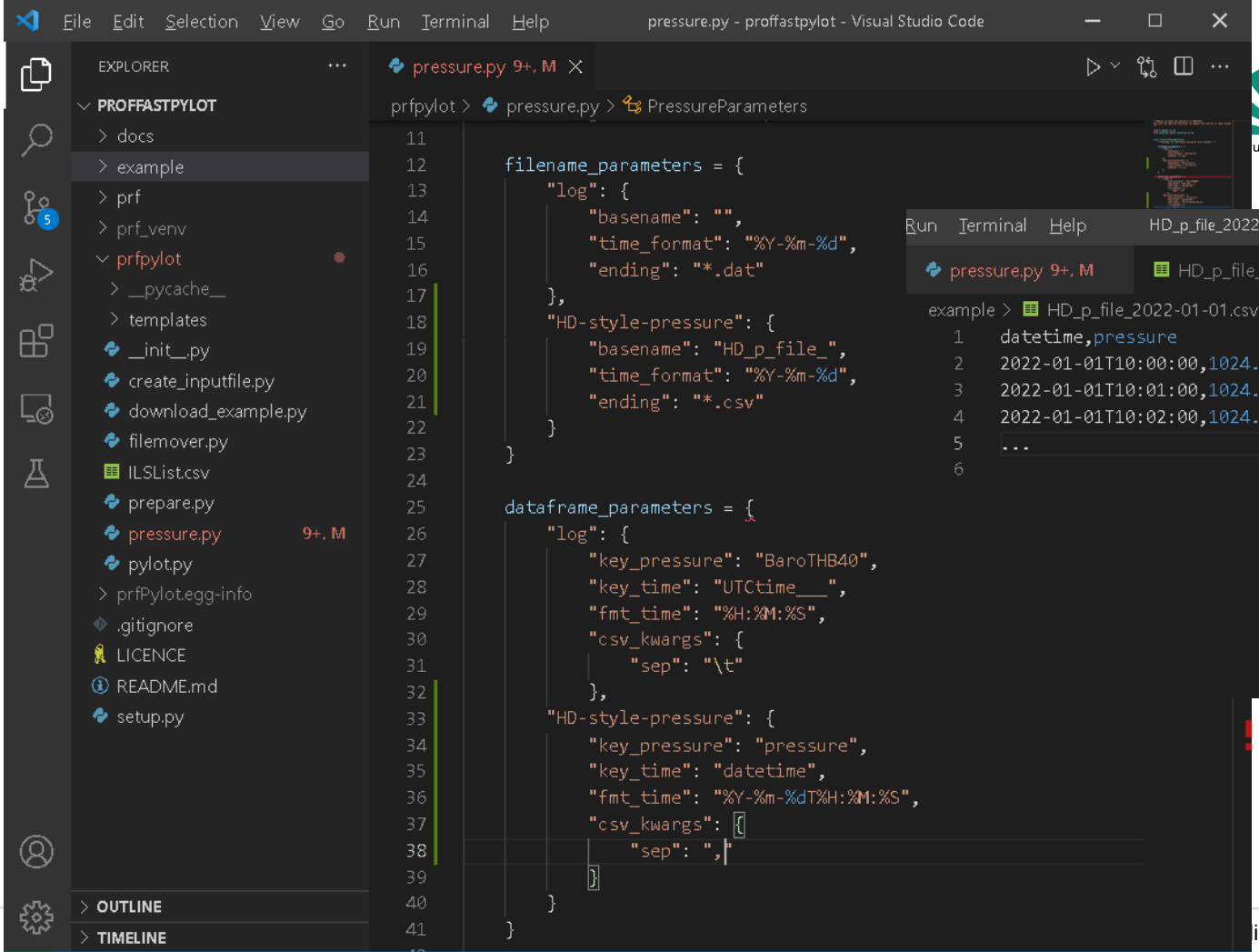
Backup

In



- Available at https://gitlab.eudat.eu/coccon-kit/proffastpylot


File Edit Selection View Go Run Terminal Help
pressure.py - proffastpylot - Visual Studio Code
EXPLORER
PROFFASTPYLOT
docs
example
prf
prf_venv
prfpylot
__pycache__
templates
__init__.py
create_inputfile.py
download_example.py
filemover.py
ILSList.csv
prepare.py
pressure.py
pylot.py
prfPylot.egg-info
.gitignore
LICENCE
README.md
setup.py
OUTLINE
TIMELINE
pressure.py
prfpylot > pressure.py > ...
class PressureParameters():
"""Storage for different pressure file formats."""
filename_parameters = {
"log": {
"basename": "",
"time_format": "%Y-%m-%d",
"ending": "*.dat"
}
}
dataframe_parameters = {
"log": {
"key_pressure": "BaroTHB40",
"key_time": "UTCtime___",
"fmt_time": "%H:%M:%S",
"csv_kwargs": {
"sep": "\t"
}
}
}
def get_filename(self, pressure_type, date):
"""Return merged filename of pressure_type."""
params = self.filename_parameters[pressure_type]
filename = "".join(
master
Python 3.8.5 64-bit ('prf_venv': venv)
0 0
Git Graph
Ln 7, Col 1
Spaces: 4
UTF-8
CRLF
Python


File Edit Selection View Go Run Terminal Help
pressure.py - proffastpylot - Visual Studio Code
EXPLORER
PROFFASTPYLOT
docs
example
prf
prf_venv
prfpylot
__pycache__
templates
__init__.py
create_inputfile.py
download_example.py
filemover.py
ILSList.csv
prepare.py
pressure.py 9+, M
pylot.py
prfPylot.egg-info
.gitignore
LICENCE
README.md
setup.py
OUTLINE
TIMELINE
pressure.py 9+, M
prfpylot > pressure.py > PressureParameters
filename_parameters = {
"log": {
"basename": "",
"time_format": "%Y-%m-%d",
"ending": "*.dat"
},
"HD-style-pressure": {
"basename": "HD_p_file_",
"time_format": "%Y-%m-%d",
"ending": "*.csv"
}
}
dataframe_parameters = {
"log": {
"key_pressure": "BaroTHB40",
"key_time": "UTCtime___",
"fmt_time": "%H:%M:%S",
"csv_kwargs": {
"sep": "\t"
},
"HD-style-pressure": {
"key_pressure": "pressure",
"key_time": "datetime",
"fmt_time": "%Y-%m-%dT%H:%M:%S",
"csv_kwargs": {
"sep": ","
}
}
}
Run Terminal Help
HD_p_file_2022-01-01.csv - proffastpylot
pressure.py 9+, M
HD_p_file_2022-01-01.csv
example > HD_p_file_2022-01-01.csv
datetime,pressure
2022-01-01T10:00:00,1024.00
2022-01-01T10:01:00,1024.34
2022-01-01T10:02:00,1024.15
...
master* Python 3.8.5 64-bit ('prf_venv': venv) 25 3 Git Graph Ln 38, Col 26 Spaces: 4 UTF-8 CRLF Python