

SENTINEL-3 OLCI L2 BIOGEOCHEMISTRY DATA ACCESS PROGRAM WITH PYTHON

EDWARDS TAUFIQURRAHMAN

2024-08-01

INTRODUCTION

Sentinel-3 is part of Copernicus Program, Europe's ambitious remote sensing program aimed to "providing accurate, timely and easily accessible information to improve the management of the environment, understand and mitigate the effects of climate change and ensure civil security,"¹

Sentinel-3 Ocean and Land Colour Instrument (OLCI) collect and provides data to monitor ocean and land geophysical parameters, for example sea surface Chlorophyll-a concentration.

¹<https://sentiwiki.copernicus.eu/web/copernicus-programme>

SENTINEL-3 OLCI USAGE

Applications	OLCI Geophysical Products						
	Rxxx	PAR	T865	A865	IWV	OTCI	GI-FAPAR RC681 RC865
Mapping and monitoring of Land Use and Cover (LUC)						✓	✓
Athmospheric Monitoring			✓	✓	✓		
Climate Change Monitoring	✓	✓	✓	✓	✓	✓	✓

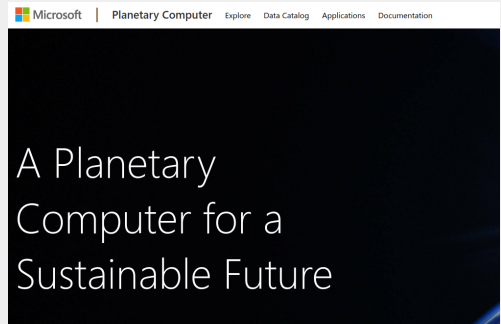
Figure 1: Sentinel-3 OLCI products and application
(<https://sentiwiki.copernicus.eu/web/olci-applications>)

SENTINEL-3 DATA ACCESS

Data access provided by several organizations. Copernicus' WEkEO and Microsoft's Planetary Computer to name two of them.



(a) WEkEO



(b) Planetary Computer

Figure 2: Two example of Sentinel-3 data provider: WEkEO and Planetary Computer

TUTORIAL ON ACCESSING SENTINEL-3 OLCI

TUTORIAL ON ACCESSING SENTINEL-3 OLCI

USING GITHUB CODESPACE

Python and Github/Jupyterhub can be used to access and open satellite datasets.



(a) Python



(b) JupyterHub



(c) Github Codespaces

Figure 3: Python, Jupyter and Github make it easier to use satellite dataset.

i Note

Python is “easy” and reliable. Cloud infrastructure such as Github Codespace and JupyterHub make it even more easier.

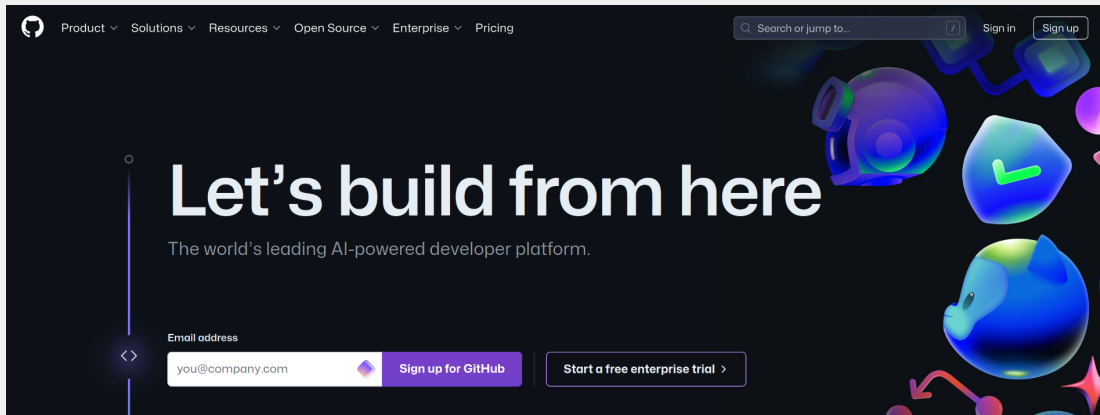
🔥 Caution

This tutorial will guide you using **Github Codespaces**

STEP 1: CREATE GITHUB ACCOUNT

💡 Steps

Easy. Just go to <https://github.com/signup>



STEP 2: MAKE A COPY OF THE CODE REPOSITORY I

Steps

1. Log-in to your Github & go to `https://github.com/eds-ocean/imber-sen3`,
2. Click **Fork**,
3. Continue as example on the next slide.

STEP 2: MAKE A COPY OF THE CODE REPOSITORY II

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk ().*

Owner *



ed10302007

Repository name *

imber-sen3

✔ imber-sen3 is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Sentinel-3 OLCI Biogeochemistry Data Acquisition Program

☒ Copy the `main` branch only

Contribute back to eds-ocean/imber-sen3 by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

STEP 3: CREATE GITHUB CODESPACE I

💡 Steps

1. On the forked repository, click **Code**, then select tab **Codespaces**,
2. Click the three dots (...) and select **Manage Codespaces**,
3. Click down arrow beside **Create codespace on main** and then select **Configure and create codespace**,
4. Click **Configure and create codespace** and fill parameter like in the example.

(see next slides)

STEP 3: CREATE GITHUB CODESPACE II

The screenshot shows the GitHub interface for a repository named 'imber-sen3' (Public), which is forked from 'eds-ocean/imber-sen3'. The 'main' branch is selected, and it is noted that the branch is up to date with the upstream. The repository contains several files and folders, including '.resources', 'Code', 'Settings', '.gitignore', 'LICENSE', and 'README.md'. The 'Codespaces' tab is active, displaying a message: 'No codespaces. You don't have any codespaces with this repository checked out.' A green button 'Create codespace on main' is visible, along with a link to 'Learn more about codespaces...'. A dropdown menu is open, showing options: '+ New with options...', 'Configure dev container', 'Set up prebuilds', 'Manage codespaces', 'Share a deep link', and 'What are codespaces?'. The 'Manage codespaces' option is highlighted. The bottom of the page shows the 'README' and 'License' tabs, and the URL '02007/imber-sen3/codespaces' is visible in the address bar.

STEP 3: CREATE GITHUB CODESPACE III

The screenshot displays the GitHub Codespaces interface. At the top, the heading "Your codespaces" is visible. To its right are two buttons: "Go to docs" and "Create codespace on main". The "Create codespace on main" button is highlighted in green and has a dropdown arrow. A dropdown menu is open below this button, showing a checked option "Create codespace on main" with specifications "2-core • 8GB RAM • 32GB". Below this is an option "Configure and create codespace" with the subtext "Show advanced options before launching codespace".

Below the buttons, a blue banner states: "Codespace usage for this repository is paid for by ed10302007."

The main section is titled "Getting started with GitHub Codespaces" and contains three cards:

- Learn core concepts**: New to Codespaces? [Start here](#). Learn the core concepts and how to get started.
- Configure and manage**: Learn more about features like [secret management](#) and [port forwarding](#).
- Develop locally**: Access codespaces from within [Visual Studio Code](#) or [JetBrains](#).

STEP 3: CREATE GITHUB CODESPACE IV

Create codespace for ed10302007/imber-sen3

Branch
This branch will be checked out on creation

main ▾

Region
Your codespace will run in the selected region

Europe West ▾

Machine type
Resources for your codespace

4-core ▾

Create codespace

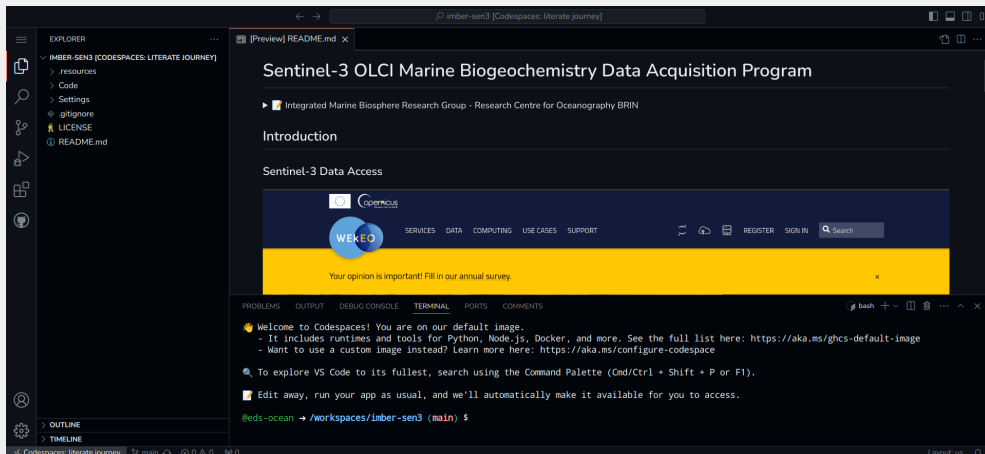
TUTORIAL ON ACCESSING SENTINEL-3 OLCI

RUNNING THE CODE

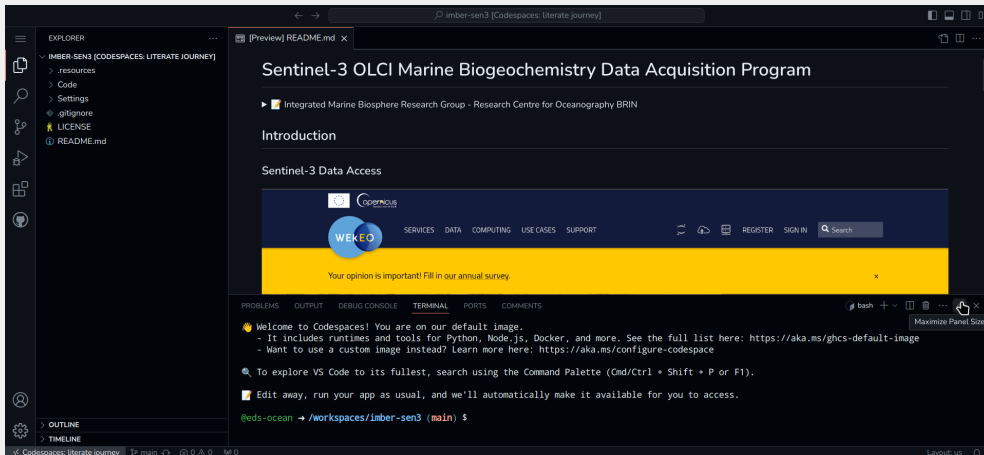
Github Codespace is an environment to run code from various programming language. By default, it opened in VS-Code web and screen will open the Readme file in a Jupyter Notebook page. It also provide Terminal to work on Linux command line environment.

We will just use the Terminal in this tutorial. Of course you can always try to use the Notebook to run the code.

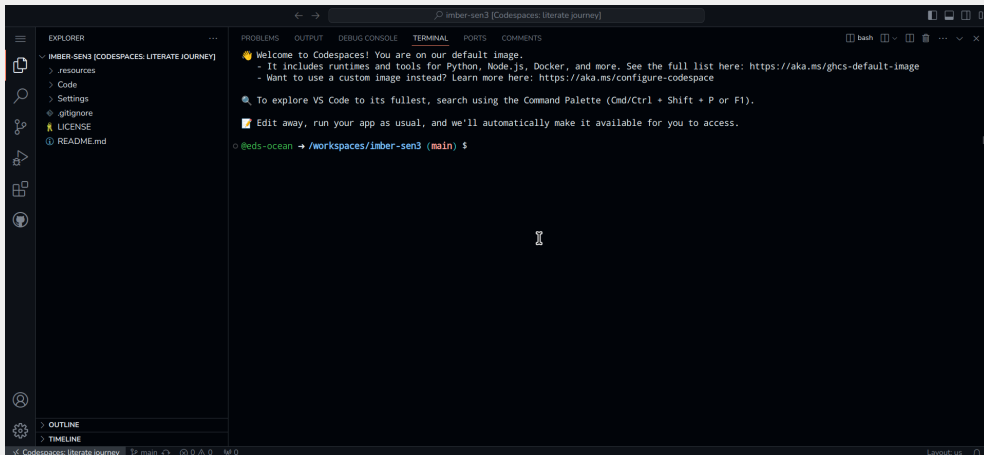
Default codespace screen



Click on the arrow to maximize terminal



Maximized terminal screen



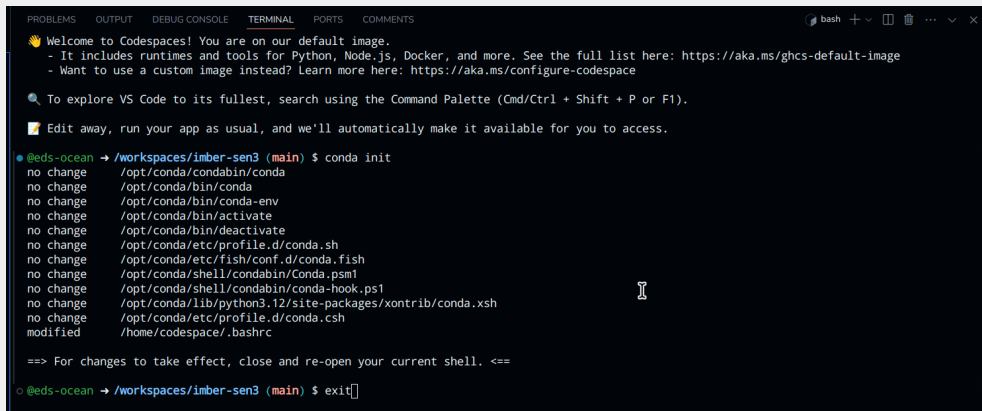
INSTALL CONDA ENVIRONMENT I

Steps

1. Initiate conda,
2. Restart terminal,
3. Install conda environment from yml file.

INSTALL CONDA ENVIRONMENT II

Type `conda init` to initiate Conda. Wait until finished and then type `exit` to close the Terminal



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Welcome to Codespaces! You are on our default image.
- It includes runtimes and tools for Python, Node.js, Docker, and more. See the full list here: https://aka.ms/ghcs-default-image
- Want to use a custom image instead? Learn more here: https://aka.ms/configure-codespace

To explore VS Code to its fullest, search using the Command Palette (Cmd/Ctrl + Shift + P or F1).

Edit away, run your app as usual, and we'll automatically make it available for you to access.

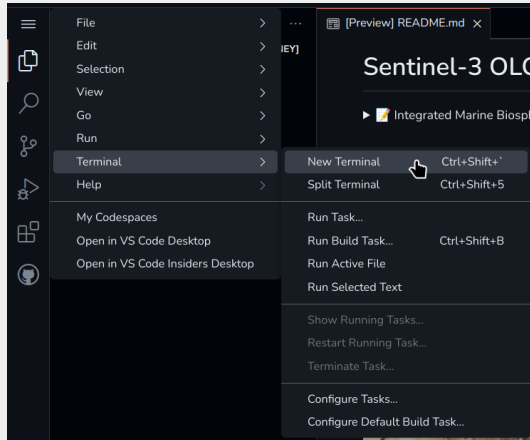
@eds-ocean → /workspaces/imber-sen3 (main) $ conda init
no change /opt/conda/condabin/conda
no change /opt/conda/bin/conda
no change /opt/conda/bin/conda-env
no change /opt/conda/bin/activate
no change /opt/conda/bin/deactivate
no change /opt/conda/etc/profile.d/conda.sh
no change /opt/conda/etc/fish/conf.d/conda.fish
no change /opt/conda/shell/condabin/Conda.psm1
no change /opt/conda/shell/condabin/conda-hook.ps1
no change /opt/conda/lib/python3.12/site-packages/xontrib/conda.xsh
no change /opt/conda/etc/profile.d/conda.csh
modified /home/codespace/.bashrc

==> For changes to take effect, close and re-open your current shell. <==

@eds-ocean → /workspaces/imber-sen3 (main) $ exit
```

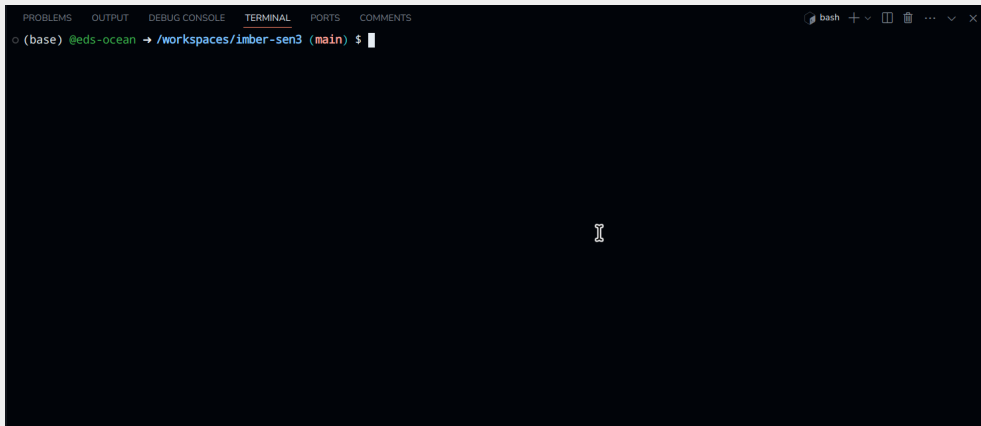
INSTALL CONDA ENVIRONMENT III

Open new terminal by clicking three-lines > Terminal > New Terminal



INSTALL CONDA ENVIRONMENT IV

Newly opened terminal. Note that it now have (base) on left side.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS bash + - [ ] ... - x
o (base) @eds-ocean -> /workspaces/imber-sen3 (main) $
```


INSTALL CONDA ENVIRONMENT V

Step

1. Now type code below,

```
conda env create -f Settings/sen3_pc.yml
```

2. Wait until finished, then type,

```
conda activate sen3_pc
```

RUN THE CODE

 Step

Type:

```
python Code/mode_1_planetarycomputer.py
```