CloudLab-SciPy2025

This repository contains examples of how to use Dataplug for managing large-scale data stored in the cloud, and how to scale up processing using Lithops for seamless serverless execution.

Installation

First, install the required libraries:

Install Dataplug

```
pip install git+https://github.com/CLOUDLAB-URV/dataplug
```

Install Lithops

```
pip install lithops
```

You may also need to configure your cloud backend (e.g., AWS, IBM, Azure) using the Lithops config guide.

Example 1 – Using Dataplug Locally

The notebook `dataplug_example.ipynb` shows how to:

1. Load a FASTA file directly from an S3 bucket using `CloudObject.from_s3`.
2. Inspect the number of sequences and total size.
3. Preprocess the file by splitting it into chunks.
4. Partition the data into slices for parallel or sequential processing.

Run the notebook

```
jupyter notebook dataplug.ipynb
```

Example 2 – Scalable Processing with Dataplug + Lithops

The second notebook `dataplug_lithops.ipynb` shows how to use **the exact same code** to process the data **on the fly in Lithops**, without local resource limits.

It demonstrates how to:

- Partition a FASTA file into slices using `co.partition(...)`
- Define a processing function for each partition (`process_fasta_partition`)
- Use `lithops.FunctionExecutor` to execute processing in parallel

Run the notebook

```
jupyter notebook dataplug_lithops.ipynb
```

> ☑ Thanks to the native integration of Dataplug with Lithops, you can scale your code effortlessly — no changes in logic required!

What You Need

- Access to an S3-compatible storage (AWS S3, MinIO, etc.)
- Proper cloud credentials (can be set with `aws configure` or via environment variables)
- Python 3.10 or higher

About

This code is part of the **CloudLab-SciPy2025** tutorial series for scientific computing in the cloud.