



Geo-X training course

Hands-on 4

Please use the following citation for:

- **This exercise**

Miguel Sanchez-Lopez, Michaela Rychetska, 2026, Hands-on 4: Running the Model – Geo-X Training Course (Version 1.0.). Climate Compatible Growth. DOI: [to be assigned].

- **Geo-X software**

Sanchez-Lopez, M., Naqvi, S., Leonard, A., Schirren, L., & Hirmer, S. (2026). Geo-X: Generalised geospatial software for production of green commodities. *SoftwareX*, 34, 102712. <https://doi.org/10.1016/j.softx.2026.102712>

Learning outcomes

By the end of this exercise, you will be able to:

- Understand how Snakemake orchestrates the Geo-X workflow and what each rule does.
- Download Ghana's weather data and run a first real optimisation.
- Locate and read the output files Geo-X produces.
- Recognise and resolve a common Snakemake error.
- Design and run a second scenario by editing config.yaml.

This exercise is split into two parts. Weather data downloads can take anywhere from a few minutes to a few hours, depending on your connection, so Part I gets that download started while we cover the concepts behind it. Part II picks up once the download has finished

Part I – Setup and weather data

Goal: understand how Snakemake works, confirm your inputs are in place, and start Ghana's weather data download.

Activity 1 – Snakemake, the workflow engine

Without Snakemake, you would need to run each script yourself, in the right order, every time: download weather data, build the network, solve it, export results. Change one input and you would have to remember to rerun everything downstream, which makes results hard to reproduce.

With Snakemake, you ask for a final output and it works out the rest: which rules need to run, in what order, and which ones can be skipped because nothing changed.

- **Order resolved automatically**, based on which rule needs which file.
- **Only reruns what changed**, so iterating on scenarios is fast.
- **Same command every time**, fully reproducible.

Three commands resolve the full, seven-rule Geo-X chain:

- `snakemake -j N run_weather` downloads weather data (rule 1).
- `snakemake -j N optimise_all` runs rules 2 to 5: transport costs, water costs, plant optimisation, and total costs.
- `snakemake -j N map_all` runs rules 2 to 7: everything in `optimise_all`, plus cost components and spatial maps.

Note: you can also run an individual rule by name, which is useful for debugging a single step without rerunning the whole chain.

Activity 2 – Confirm your parameter files are in place

In Hands-on 3, you reviewed Ghana's parameter files in the shared, read-only `parameters/GA/` reference folders. Geo-X, however, reads from your own fork, not from that shared copy, so before running anything you need your own set of files in place.



Try it:

- On your ghana-h2-nh3 branch, copy the hydrogen parameter files from parameters/GA/hydrogen/ into your own Geo-X fork's parameters/GA/hydrogen/ folder.
- Confirm data/GA/hex_final_GA.geojson is still in place from Hands-on 3.
- Commit the addition: `git add .` then `git commit -m "Add Ghana hydrogen parameter files"`.

Note: we will only need the ammonia parameter files later, in Activity 7, so there is no need to copy those yet.

Activity 3 – Download Ghana's weather data

Geo-X needs historical solar and wind data to simulate generation. This comes from the ERA5 reanalysis dataset, downloaded via the Climate Data Store (CDS) API through Atlite, and saved as a cutout file.

Try it:

- Check whether the cutout already exists: look inside your cutouts/ folder for a file named GA_2023.nc (or whichever weather year you set in Hands-on 3). If it is already there, skip ahead to Part II.
- Otherwise, start the download: `snakemake -j 4 run_weather`

Note: runtime varies widely depending on Ghana's size and your internet connection, and can take anywhere from a few minutes to a few hours. This is a natural stopping point: once the download is running, or has finished, you are ready for Part II. The same cutout file can be reused for every future run on Ghana for this weather year, so you only need to do this once.

Part II – Running the model and reading results

Before starting, make sure your cutouts/GA_2023.nc file from Part I has finished downloading.

Activity 4 – Run your first optimisation, starting small

The heaviest rule in the chain is `optimize_plant`, which can take from minutes to hours depending on the country's size, the number of demand centres, and the commodity. Ghana's case is small (177 hexagons, one demand centre), so a full run should be manageable, but it is still good practice to start small.

Try it:

- In `config.yaml`, set `freq: '4H'` and `restrict_hexagons: True`. This restricts results to hexagons containing demand centres, setting everything else to null, which keeps a first run fast.
- Run the optimisation: `snakemake -j 4 optimise_all`
- This produces total costs per hexagon at `results/hex_total_cost_GA_2023_hydrogen.geojson`.

Note: if Snakemake stops with an error, read it from the top: which rule failed, what type of error it is, and which file it points to. For example, a `MissingInputException` naming `data/GA/hex_final_GA.geojson` means Geo-X could not find that file, usually because it was not copied into the right folder, or was misnamed.

Activity 5 – Explore your output files

Geo-X writes its outputs into two folders at the root of your cloned repository: `results/` for the GeoJSON cost files, and `plots/` for PNG maps, generated in Activity 6.

Try it:

- Locate `results/hex_total_cost_GA_2023_hydrogen.geojson`, one row per hexagon and demand centre combination.



- Open it by dragging the file into QGIS. Hexagons should render immediately, with cost attributes available for styling and filtering, no code required.
- If you prefer Python, a quick look with geopandas: `df = gpd.read_file("results/hex_total_cost_GA_2023_hydrogen.geojson")`

Note: `hex_cost_components_GA_2023_hydrogen.geojson` sits alongside it, with the same hexagons broken down by cost component. We'll use this in detail in Class 5.

Activity 6 – Generate cost maps

Try it:

- Run: `snakemake -j 4 map_all`. This reruns only what is needed beyond `optimise_all`, adding the cost component breakdown and spatial maps.
- Locate your maps at `plots/GA_2023_hydrogen/`, one PNG per cost component (total delivered cost, transport, electrolyser, solar, and more).

Note: we will interpret what these maps are telling us, and what drives the spatial patterns, in Class 5.

Activity 7 – Design and run a second scenario

Changing a single line in `config.yaml` and rerunning gives you a second, comparable scenario. Since Snakemake only reruns what changed, this is fast. We'll compare `plant_type: 'hydrogen'` against `plant_type: 'ammonia'`, reusing the ammonia demand data from Hands-on 3.

Try it:

- Commit your hydrogen results first: `git add .` then `git commit -m "Hydrogen scenario results"`.
- Copy the ammonia parameter files from `parameters/GA/ammonia/` into your fork's `parameters/GA/ammonia/` folder, the same way you did for hydrogen in Activity 2.
- Edit `config.yaml`: change `plant_type: 'hydrogen'` to `plant_type: 'ammonia'`. Leave everything else as it is, so the two scenarios are comparable.



- Rerun: `snakemake -j 4 map_all`. This produces a new set of results and plots, without repeating the weather download.
- Commit your ammonia results: `git add .` then `git commit -m "Ammonia scenario results"`, then `git push origin ghana-h2-nh3`.

Note: this same one-line-change approach works for other comparisons too, such as toggling `pipeline_construction`, changing the renewable generator mix, or moving the demand location. We'll explore some of these as advanced options in Class 6.

Wrap-up

You have completed this exercise. Before moving on, check that:

- You can explain what Snakemake does and why it is used.
- Your hydrogen parameter files are in your own fork, alongside `hex_final_GA.geojson`.
- `cutouts/GA_2023.nc` has downloaded successfully.
- `optimise_all` completed and produced results in `results/`.
- `map_all` completed and produced maps in `plots/`.
- You have run and committed both a hydrogen and an ammonia scenario.

Well done! You can now run Geo-X end to end. In Class 5, you will learn how to interpret these cost maps and results in depth.