

Classe Loader + Masques

Classe Loader

En python, une convention est de commencer le nom d'une classe par une majuscule.

Plusieurs options pour rédiger cette classe s'offre à vous

Le masque en tant qu'attribut

```
class Loader:
    def __init__(self, mask_file) -> None:
        self.mask = xr.open_dataset(mask_file)

    def get_determinist(self, date) -> xr.Dataset:
        ds = xr.open_dataset("arome_forecast_{date.strftime("%Y%m%d%H")}.nc")
        return ds * mask_on_grid(self.mask, "eurw1S100")

    def _get_ensemble_member(self, date, member) -> xr.Dataset:
        file = # Le bon nom !
        ds = xr.open_dataset(file).expand_dims("number")
        return ds * mask_on_grid(self.mask, "eurw1S40")

    def get_ensemble(self, date) -> xr.Dataset:
        ens_list = []
        for mb in range(0, 5):
            # Rajoute la dimension membre afin de pouvoir merger.
            ens_list.append(self._get_ensemble_member(date, mb))
        return xr.merge(ens_list)

    def summarize_info(self, date):
        """
        This function print the determinist and ensemblist forecast which can help to analyse the corresponding date.
        """
```

Autre option

```
class Loader:
    def __init__(self, path, list_path):
        self.path = path
        self.ens_path = list_path

    def get_determ(self, mask_path):
        mask = xr.open_dataset(mask_path)
        return ds * mask_on_grid(self.mask, "eurw1s100")

    def _get_ensemble_member(self, mask_path, member):
        # Lecture du membre et ajout de dimension
        # Multiplication par le bon masque
        pass

    def get_ensemble(self, mask_path):
        ens_list = []
        for mb in range(0,5):
            ens_list(self._get_ensemble_member(mask_path, mb))
        return xr.merge(ens_list)
```

La première option est plutôt tournée vers l'étude d'une région : on ne peut en changer. Si on veut étudier une autre région on doit réinstancier un autre objet. Cependant on peut étudier diverses prévisions sur la même région.

La seconde vers l'exploitation d'une prévision déterministe/ensembliste. Dans ce cas, on peut étudier la situation sur diverses régions facilement.

Utilisation des masques

Exemple pour masquer un dataset

```
import xarray as xr
import matplotlib.pyplot as plt
ds = xr.open_dataset("/home/newton/ienm2021/chabotv/COURS_CS/data/arome_forecast_2024102712.nc")
mask = xr.open_dataset("/home/newton/ienm2021/chabotv/COURS_CS/masks/Sympo/sympo_31.nc")["eurw1s100"]
mask = mask.rename({"latitude_eurw1s100": "latitude", "longitude_eurw1s100": "longitude"})
ds_masked = ds*mask
```

De cet exemple on peut en tirer une fonction "masque sur une grille"

```
def mask_on_grid(mask: xr.Dataset, grid: str="eurw1s100")->xr.DataArray:
    """
    Get the mask for a specific grid.
    Rename dimension as latitude and longitude.
    """
    # 1. Selectionne dans le dataset le masque correspondant à la grille
    # A faire
    # 2. Renommer les coordonnées
    my_mask = mask.rename({f"latitude_{grid}": "latitude", f"longitude_{grid}": "longitude"})
    return my_mask
```