#WebGIS | Final Project

Developing NCMapper for Visualizing NetCDF Data Base on Web Geographic Information System Technique

建置NCMapper:基於網頁地理資訊系統的NetCDF檔案視覺化工具

R12229002 龍威江、41023901L胡舜宇、B10208021 陳冠羽、R13224214王楚鈞

Final Presentation

Introduction

Motivation

目標使用者

大氣科學專業人員

現況問題

NetCDF為常見之大氣科學資料格式。
使用者不易讀取, 難以迅速視覺化。

解決方案

開發便於使用的網頁系統, 實現快速處理與展示NetCDF資料。

Framework

> Interaction between user, frontend and backend.

Front-end

Home Page

NC Mapper **NC Mapper**

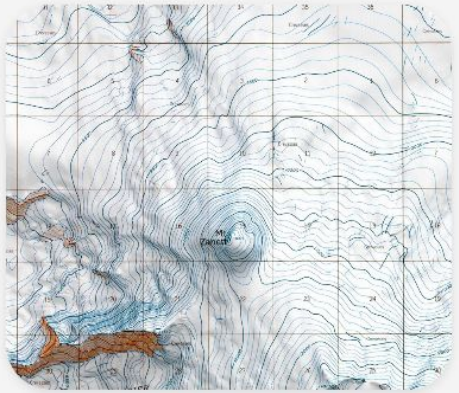
Tutorial Donate Us Contact Us About Us

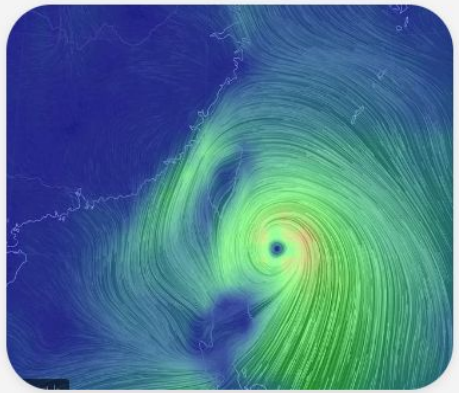
NetCDF to Professional Map

Immediately

Whether you need to quickly view a NetCDF file or create a beautiful scientific map, this tool is designed for you!

Choose file No file chosen Start Mapping





Front-end & Back-end

Data transfer (between pages w/ JS)

```
// 把用戶選項資訊透過 URL傳過去 (Home)
const encodedData = encodeURIComponent(JSON.stringify(selectedVariables));
        window.location.href = `MAP/nc_map_viewer.html?data=${encodedData}`;


// 獲取起始頁面傳輸的資料  (Map)
function getUrlData() {
  const urlParams = new URLSearchParams(window.location.search);
  const encodedData = urlParams.get('data');

  if (encodedData) {
    try {
      const decodedData = decodeURIComponent(encodedData);
      const selectedVariables = JSON.parse(decodedData);
      return selectedVariables;
    } catch (error) {
      console.error('Error parsing URL data:', error);
      return null;
    }
  }
  return null;
}
```

Front-end & Back-end

Data transfer (PHP & Python)

```php
$pythonPath = realpath('../python/envs/ncmapper/python.exe');  // 指定python環境路徑
$scriptPath = realpath('../python/analyze_nc.py');
$dataPath = realpath('../uploads/data.nc');

// 先獲取所有變量
$command = "\"$pythonPath\" \"$scriptPath\" \"$dataPath\" variables 2>&1";
exec($command, $variables);

$heightData = [];
// 對每個變量獲取其高度列表
foreach ($variables as $var) {
    $command = "\"$pythonPath\" \"$scriptPath\" \"$dataPath\" heights \"$var\" 2>&1";
    unset($heights); // 清除前一次的結果
    exec($command, $heights);
    $heightData[$var] = $heights;
}
```

PHP

Front-end & Back-end

Data transfer (PHP & Python)

```php
$command = "\"$pythonPath\" \"$scriptPath\" \"$dataPath\" variables 2>&1";
```

```python
elif len(sys.argv) == 3: # 指令中0為脚本、1為nc檔、2為欲獲得的資料選項
        try:
            nc_file_path = sys.argv[1]
            # 檢查檔案是否存在
            if not os.path.exists(nc_file_path):
                raise FileNotFoundError(f"找不到檔案:{nc_file_path}")
            command = sys.argv[2]

            # Step2: 獲取維度列表
            if command == "dimensions":
                dimensions = coor_name_list_of_nc(nc_file_path)
                for dim in dimensions:
                    print(dim)

            # Step3:變量選擇
            elif command == "variables":
                variables = var_list_of_nc(nc_file_path)
                for var in variables:
                    print(var)
            else:
                print(f"未知的命令:{command}")

        except Exception as e:
            print(f"Error: {str(e)}")
            sys.exit(1)
```

Python

Back-end

Converting NetCDF to GeoTiff

後端
Back-end

NetCDF轉檔存為GeoTiff → 地圖描述物件 → 匯出為JSON

```
class NetCDFDataset(object):
    """
    自定的NetCDF資料物件，用於處理WGS1984(經緯度座標系統)的NetCDF檔案。

    *User-Input*
     - x_name: str
       在NetCDF檔案中代表東西（橫）軸的coordinate之名稱。
     - y_name: str
       在NetCDF檔案中代表南北（縱）軸的coordinate之名稱。
     - z_name: str|None
       在NetCDF檔案中代表高度的coordinate之名稱，若無該coordinate，則設為None。
     - time_name: str|None
       在NetCDF檔案中代表時間的coordinate之名稱，若無該coordinate，則設為None。

    *Attributes*
     - dataset: xarray.Dataset
       NetCDF檔案的資料集

    *Methods*
     - select_to_geotiff(save_at: str, variable: str, time: any, z: any, x_range: list|None=None,
       y_range: list|None=None)->None
       選取指定的變數、時間、層面、緯度、經度的二維資料，並將該資料存成GeoTiff檔案。
    """
```

Back-end

Map Description

後端
Back-end

NetCDF轉檔存為GeoTiff → 地圖描述物件 → 匯出為JSON

```
class MapDescription(object):
```

```
class Remark(Text):
class Renderer(object):
class Raster(Layer):
class CountriesBorderLayer(Layer):
class Layer(object):
class GridLineLayer(Layer):
class ContourLayer(Raster):
class Text(object):
class SubTitle(Text):
class LakesLayer(Layer):
class ColorBar(object):
class FeatureLayer(Layer):
class LayerList(list):
class CoastlineLayer(Layer):
class ShadingLayer(Raster):
class Title(Text):
```

Back-end

Export to JSON

後端
Back-end

NetCDF轉檔 存為GeoTiff → 地圖描述物件 → 匯出為JSON

Map description object have a method to export it as JSON format.

```
class MapDescription(object):
    ...
    def export_to_json_for_ncmapper(self, json_path: str | None = None, include_original_array: bool =
                                    True, include_contour_geojson: bool = True) -> str:
        """
        為ncmaper，將地圖描述文件導出為json文件。

        Args
            - json_path: str|None
                json文件的路徑，若為None，則只返回json字符串，預設為None。
            - include_original_array: bool
                是否將原始陣列值導出，預設為True。
            - include_contour_geojson: bool
                是否將等值線geojson導出，預設為True。
        """
```

Back-end

Export to JSON

後端
Back-end

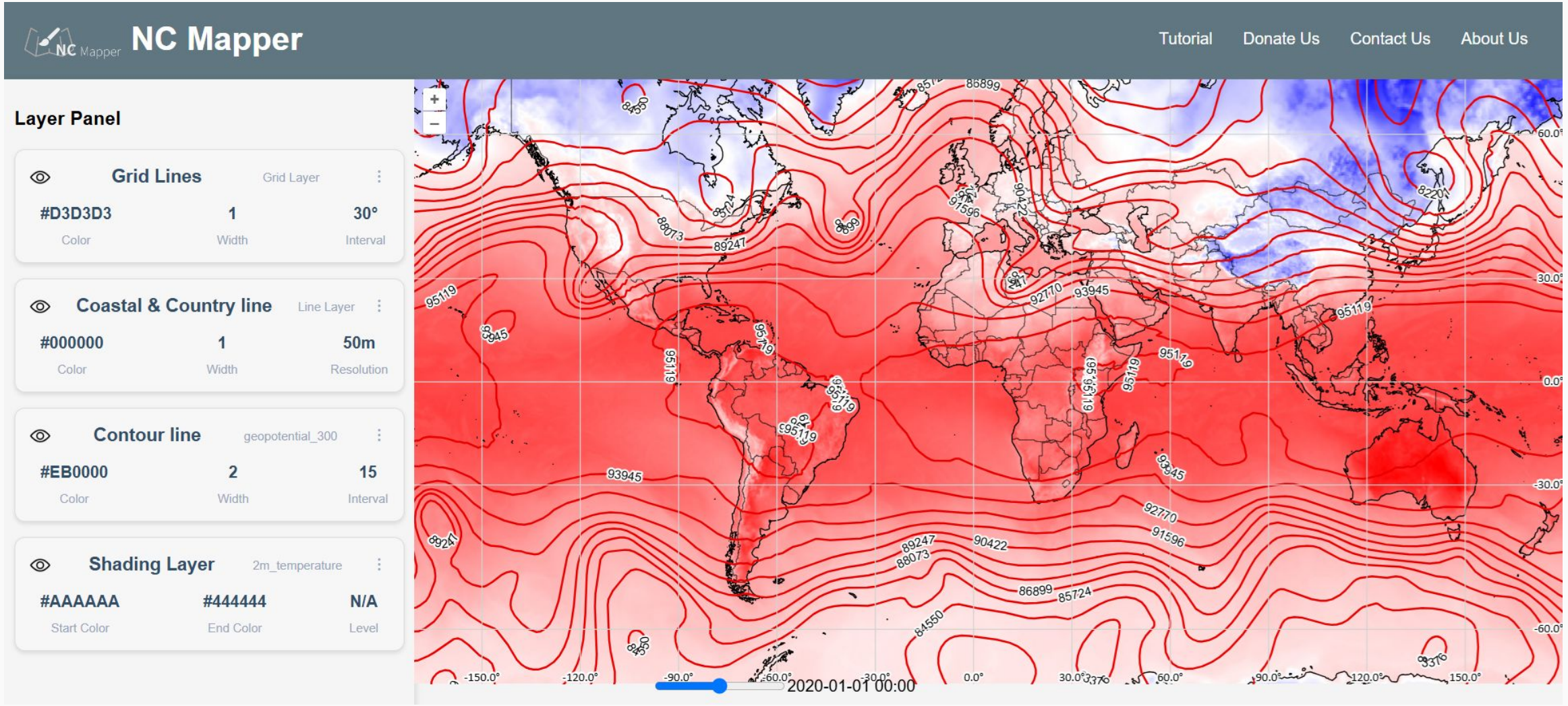
NetCDF轉檔 存為GeoTiff → 地圖描述物件 → 匯出為JSON

Map description object after exported to JSON format.


```
{
        "canvas": {
                "x_left": ...,
                "x_right": ...,
                "y_min": ...,
                "y_max": ...,
                "edge_color": [
                        ...,
                        ...
                        ],
                "edge_width": ...,
                "display_projection_crs": "PROJCRS[...",
                "total_x_range": ...,
                "total_y_range": ...
    },
    "colorbar": {
                "ticks_font_size": ...,
```

Frontend

Editing Page



Framework

> Deep dive into interaction between user, front-end and back-end.