

Why flat?
Why code?



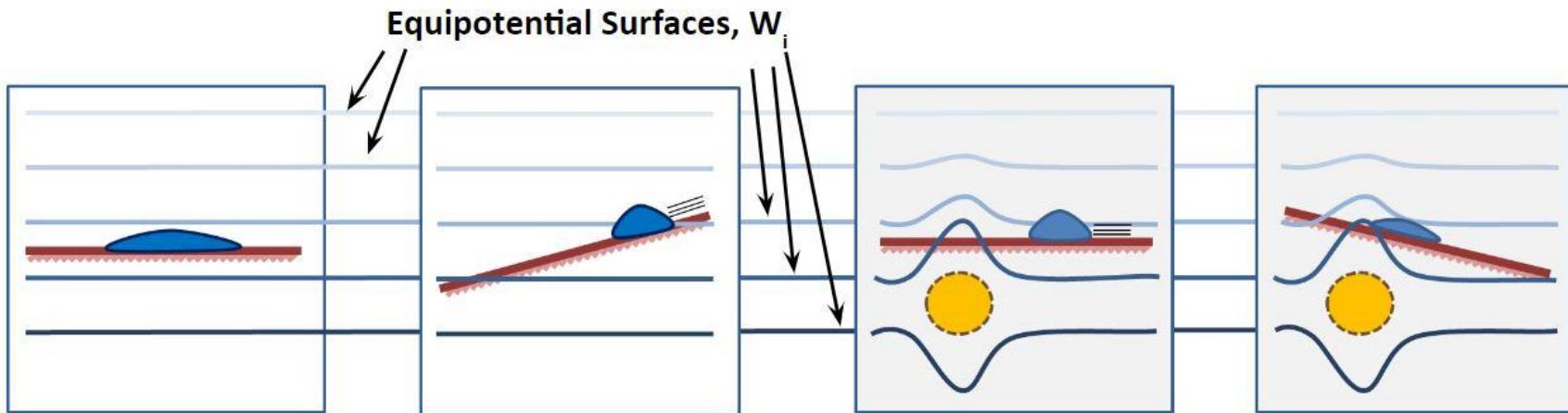
Ing. Gianfranco Di Pietro
gianfrancodp.github.io

@gianfrancodp





Why Flat?



Imagine pouring water onto a dry lake bed. If everything is level it will simply pool up. It would “like” to get to lower potential (darker lines), but the lake bed is in the way.

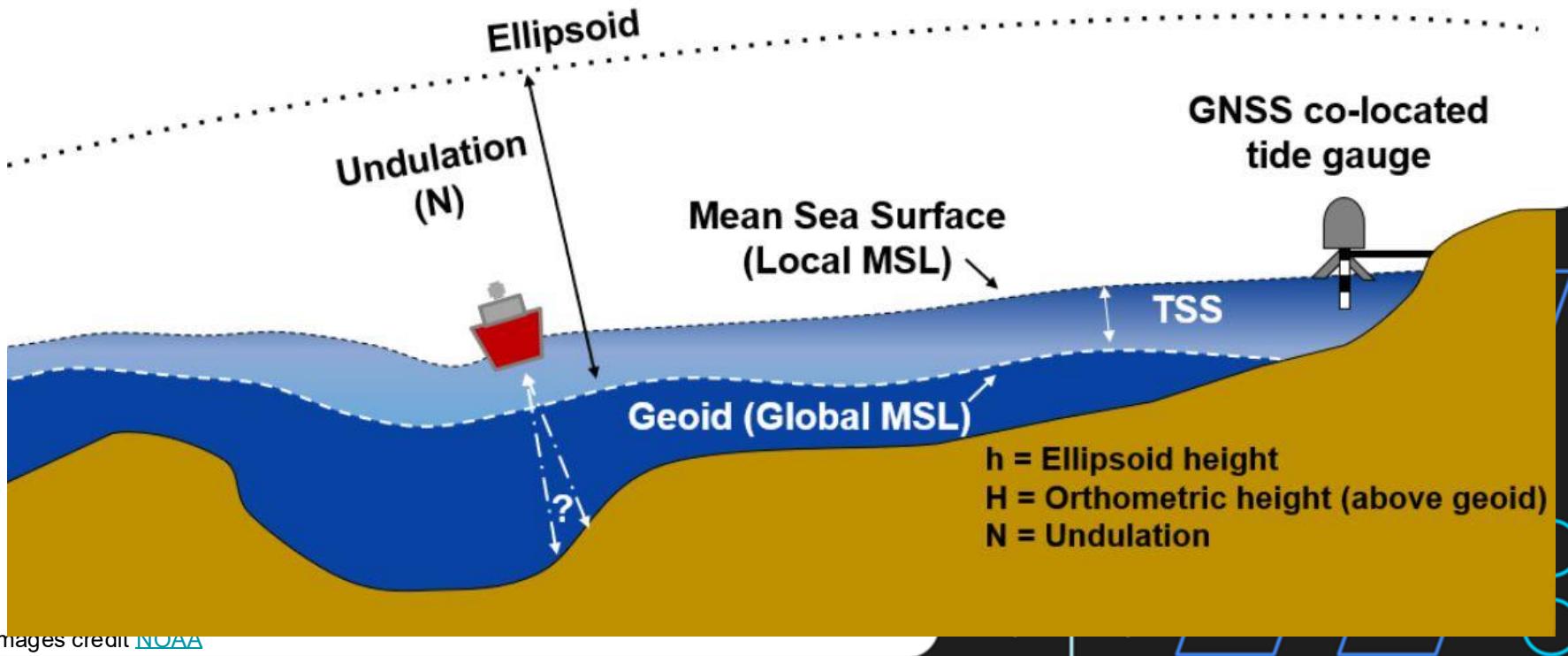
If the lake bed is tilted (it's now the side of a hill), the water can move towards lower potential (towards darker lines). Water appears to flow because of a change in height

If the lake bed appears flat, but there is a **mass anomaly** under one end, the resulting geopotential difference causes water to flow to lower potential (again, from light to dark lines). It appears to flow because of gravity.

Exaggerated (!), but you can imagine a scenario where the height difference and gravity attraction “conspire” to keep the water still on the side of a hill!



Why Flat?

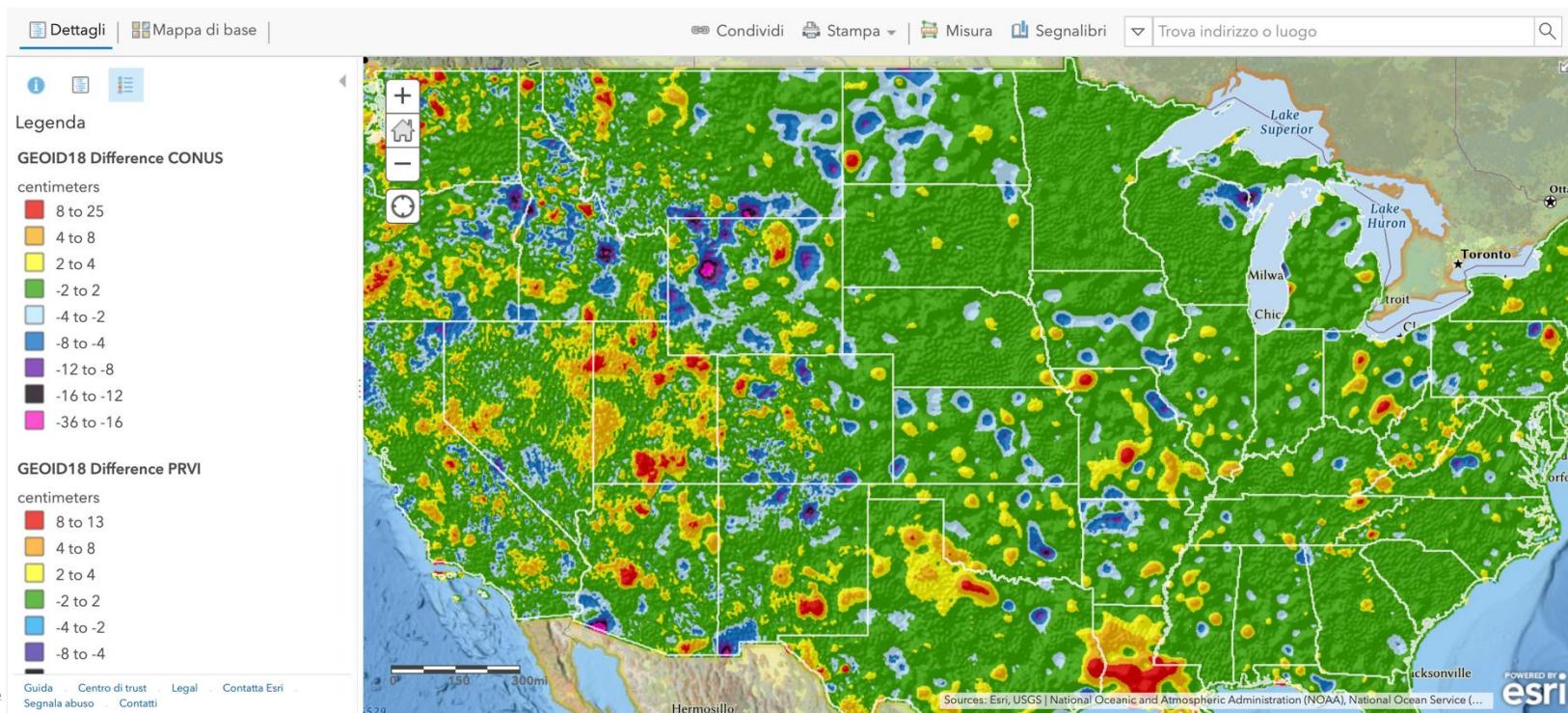


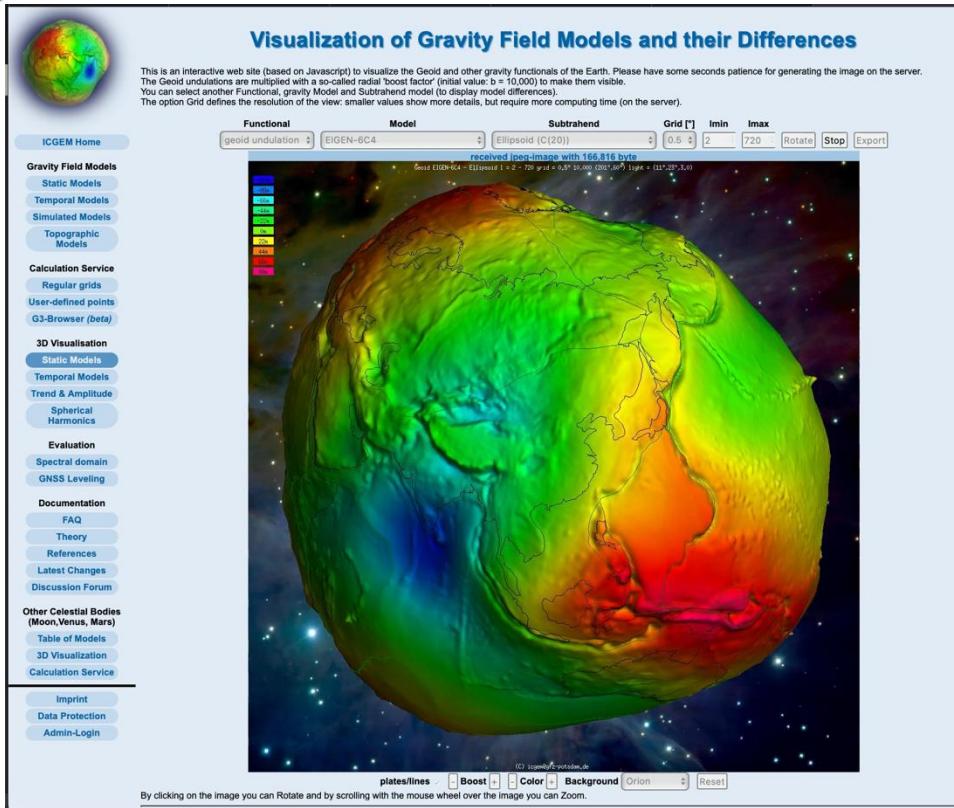


Why Flat?

Home ▾ GEOID18 Web Map

Apri nel map viewer Modifica mappa Accedi

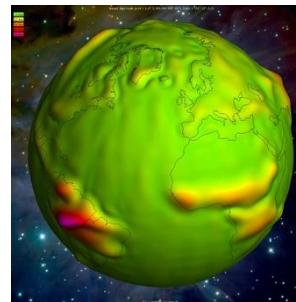




Images credit [International Centre for Global Earth Models \(ICGEM\)](#)

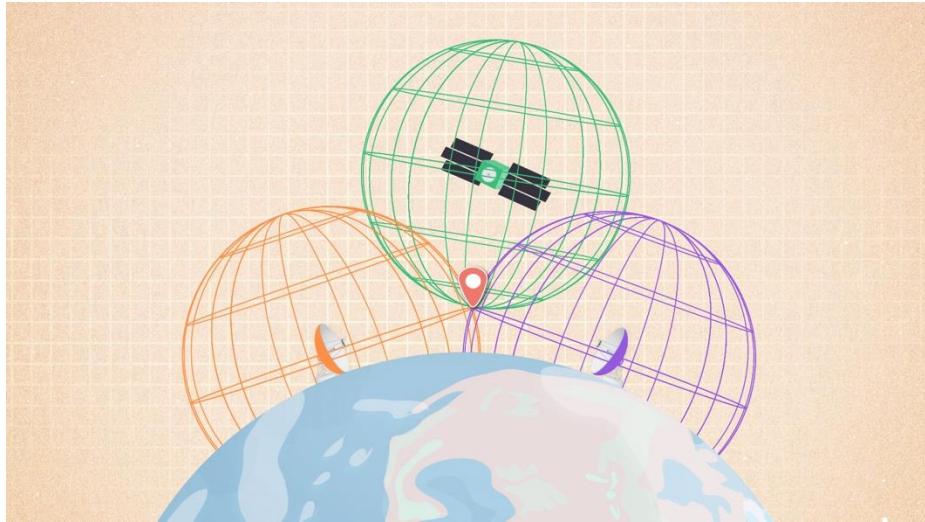
Why Flat?

- Timeseries of Gravity anomalies
- Geoid data as anomalies from simplified geometries (ellipsoid)



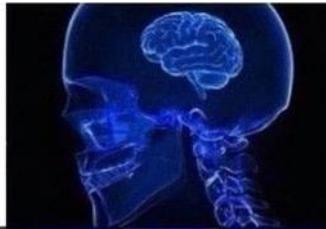
WGS84 and Global Positioning System

Lat/Lon value is angles not meters!



Credit: NASA/JPL-Caltech

**2D
REPRESENTATION
OF THE EARTH**



**SPHEROID
APPROXIMATION**



**ELLIPSOID
APPROXIMATION**

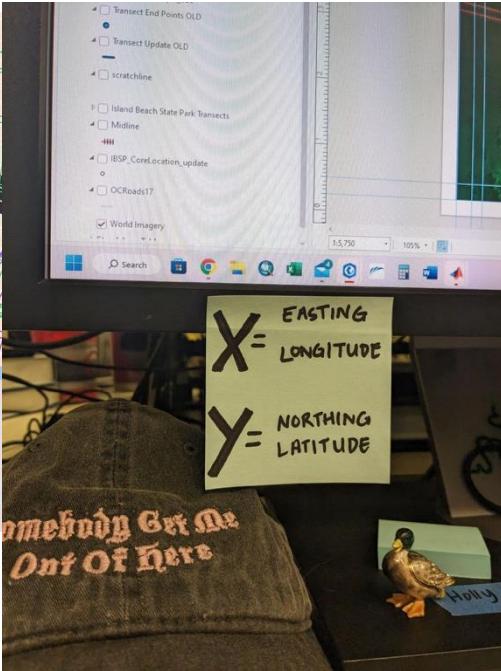


**GRAVITATIONAL
ANOMALY
MODEL (GEOID)**



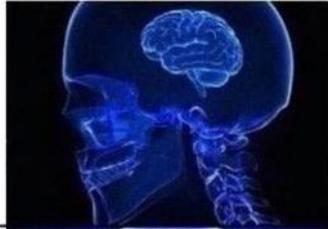
WGS84 and Global Positioning System

Lat/Lon value is angles not meters!



Credit: NASA/JPL-Caltech

2D REPRESENTATION OF THE EARTH



SPHEROID APPROXIMATION



ELLIPSOID APPROXIMATION



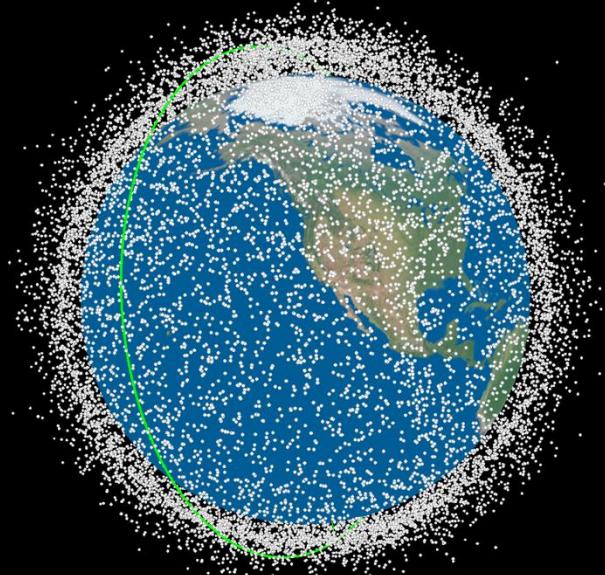
GRAVITATIONAL ANOMALY MODEL (GEOID)





Satellite Tracker 3D

– Name or NORAD ID
DOVE 3



SURCAL 159 #2872

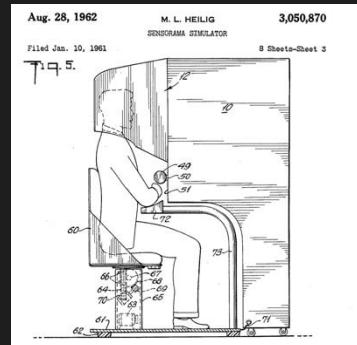
i

Speed: 26,612 km/h
Height: 919 km
Latitude: -62.44°
Longitude: -108.72°



A screenshot of a web-based story map. The background is a high-resolution 3D satellite view of a dense urban area with buildings, roads, and green spaces. Overlaid on the top left is a white rectangular box containing text and a button. At the bottom, there are three logos: CARTO, Google Maps Platform, and deck.gl.

3D in geospatial, una sfida aperta



 Google Developer Group
Gela



I soldati dell'esercito americano si ammalano durante il test delle cuffie basate su HoloLens di Microsoft

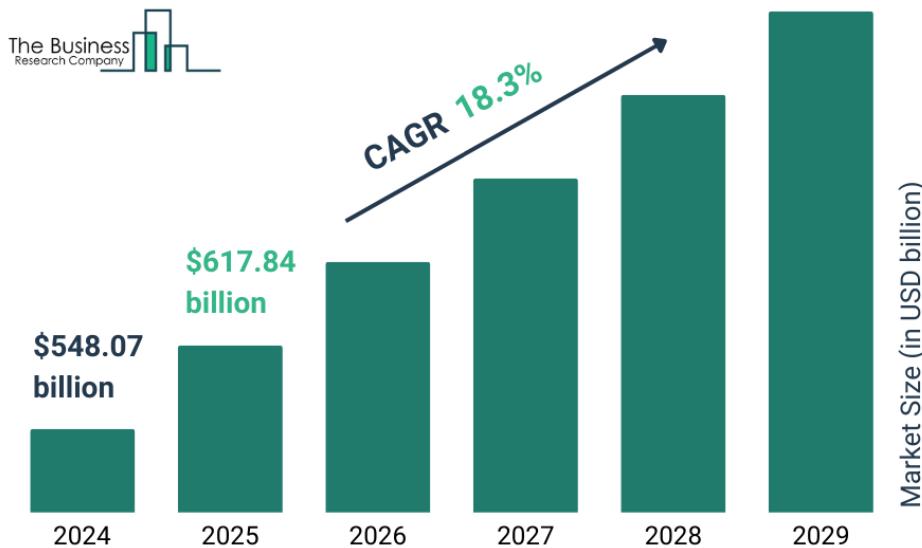


2025 compound
annual growth
rate

12.7%

Con una crescita
stimata del 18.3 nei
prossimi anni

Geospatial Solutions Global Market Report 2025



source: <https://www.thebusinessresearchcompany.com/>

Why Code?



1

Scripting, analysis and report

2

Deploy services (back-end)

3

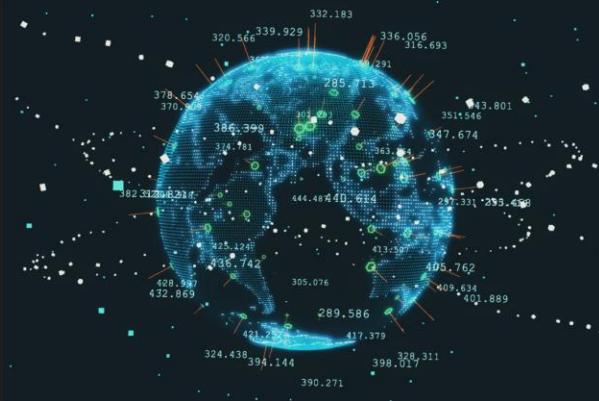
Data management

4

Data viz and sharing (front-end)



Why Code?



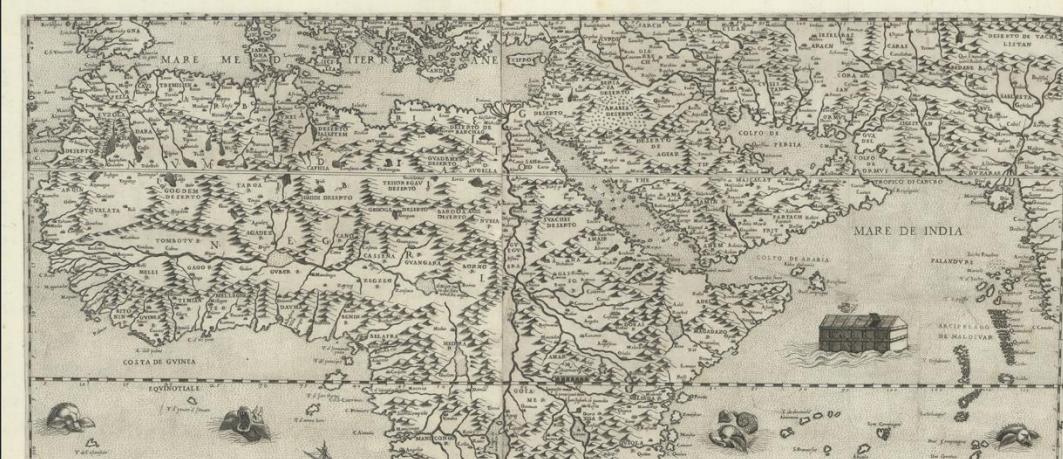
- 1 Computational thinking.
 - 2 Data thinking
 - 3 Data science thinking
 - 4 Geospatial thinking



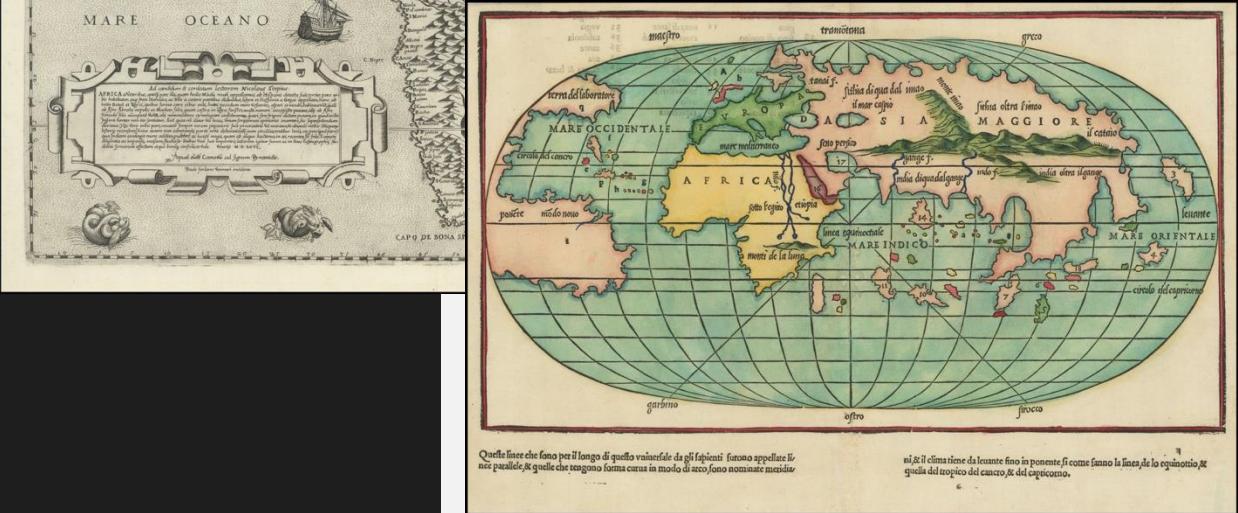
Geospatial technology is...
and is not...

@gianfrancodp





A little history...



A little history...



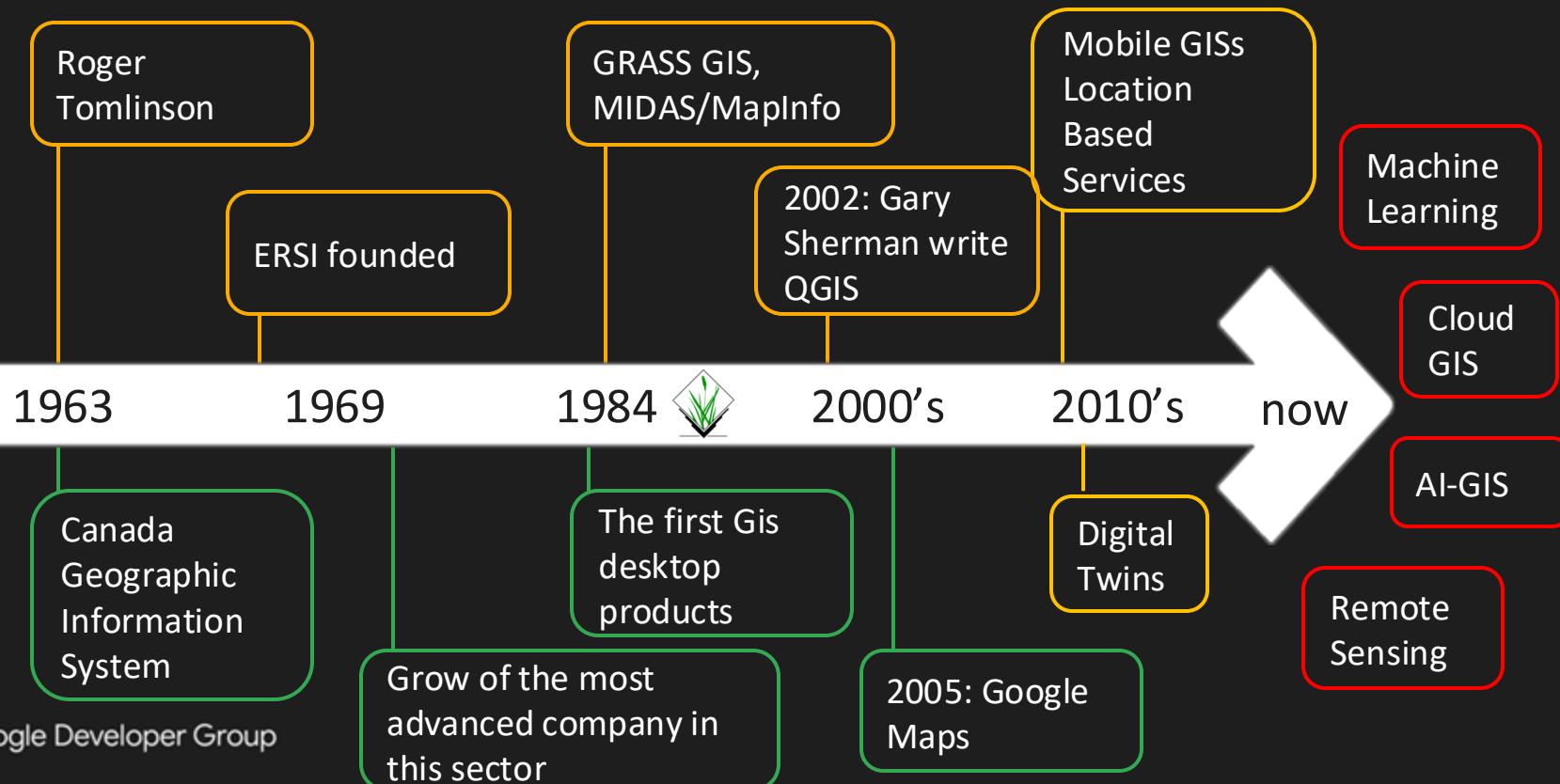
Google Developer Group
Gela

British Paté [Making Maps], 1961

A little history...



Milestone of GIS (Geographical Information Systems)





Three Case History

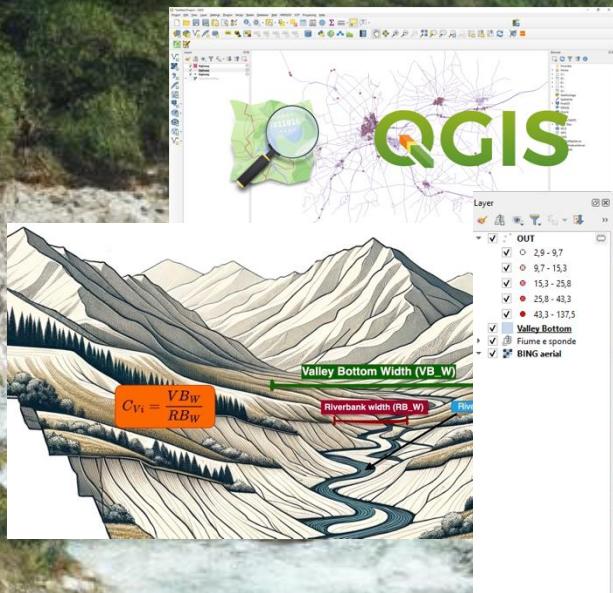
I wisely started
with a map

franz tolkaen.



Case #1: NO-CODE approach

Qgis Riverbanks Tools



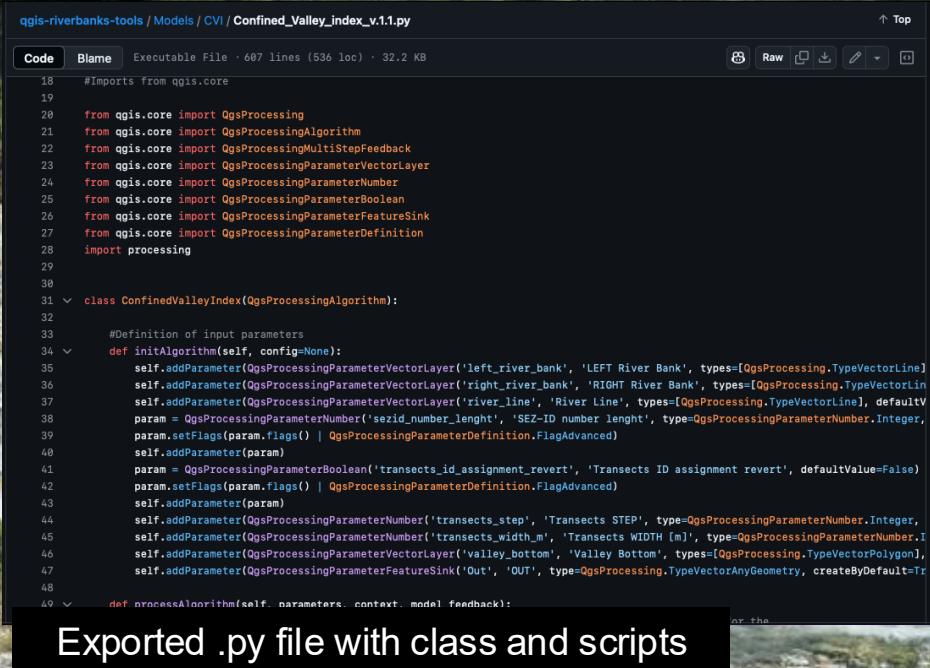
toolbox for river analysis
with Qgis and Python



<https://github.com/gianfrancodp/qgis-riverbanks-tools>

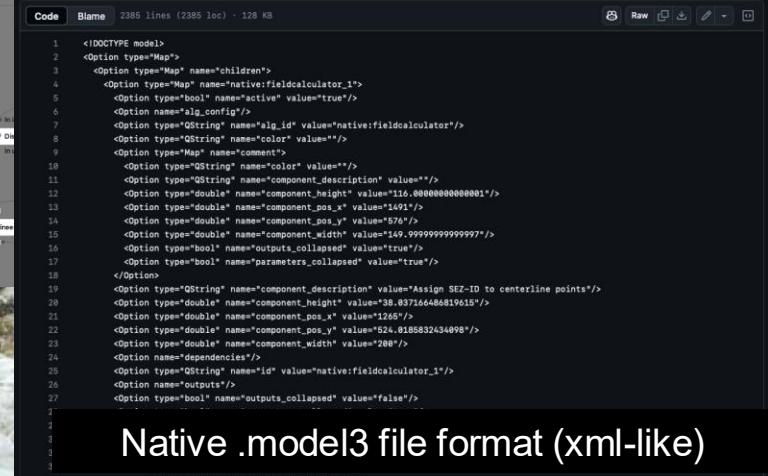
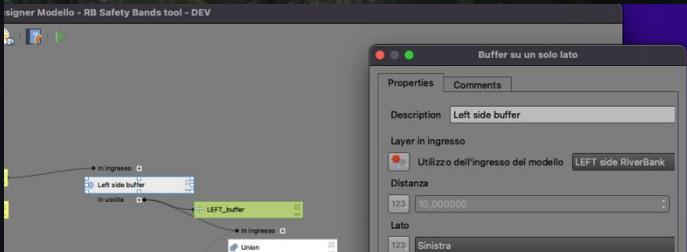
Case #1: NO-CODE approach

Qgis Riverbanks Tools



```
qgis-riverbanks-tools / Models / CVI / Confined_Valley_index_v1.1.py
Code Blame Executable File · 687 lines (586 loc) · 32.2 KB
18 #Imports from qgis.core
19
20 from qgis.core import QgsProcessing
21 from qgis.core import QgsProcessingAlgorithm
22 from qgis.core import QgsProcessingMultiStepFeedback
23 from qgis.core import QgsProcessingParameterVectorLayer
24 from qgis.core import QgsProcessingParameterNumber
25 from qgis.core import QgsProcessingParameterBoolean
26 from qgis.core import QgsProcessingParameterFeatureSink
27 from qgis.core import QgsProcessingParameterDefinition
28 import processing
29
30
31 class ConfinedValleyIndex(QgsProcessingAlgorithm):
32
33     #Definition of input parameters
34     def initAlgorithm(self, config=None):
35         self.addParameter(QgsProcessingParameterVectorLayer('left_river_bank', 'LEFT River Bank', types=[QgsProcessing.TypeVectorLine])
36         self.addParameter(QgsProcessingParameterVectorLayer('right_river_bank', 'RIGHT River Bank', types=[QgsProcessing.TypeVectorLine])
37         self.addParameter(QgsProcessingParameterVectorLayer('river_line', 'River Line', types=[QgsProcessing.TypeVectorLine], defaultValue=None)
38         param = QgsProcessingParameterNumber('sezid_number_length', 'SEZ-ID number lenght', type=QgsProcessingParameterNumber.Integer,
39         param.setFlags(param.flags() | QgsProcessingParameterDefinition.FlagAdvanced)
40         self.addParameter(param)
41         param = QgsProcessingParameterBoolean('transects_id_assignment_revert', 'Transects ID assignment revert', defaultValue=False)
42         param.setFlags(param.flags() | QgsProcessingParameterDefinition.FlagAdvanced)
43         self.addParameter(param)
44         self.addParameter(QgsProcessingParameterNumber('transects_step', 'Transects STEP', type=QgsProcessingParameterNumber.Integer,
45         self.addParameter(QgsProcessingParameterNumber('transects_width_m', 'Transects WIDTH [m]', type=QgsProcessingParameterNumber.I
46         self.addParameter(QgsProcessingParameterVectorLayer('valley_bottom', 'Valley Bottom', types=[QgsProcessing.TypeVectorPolygon],
47         self.addParameter(QgsProcessingParameterFeatureSink('OUT', 'OUT', type=QgsProcessing.TypeVectorAnyGeometry, createByDefault=True)
48
49 def processAlgorithm(self, parameters, context, model_feedback):
50     pass
```

Exported .py file with class and scripts

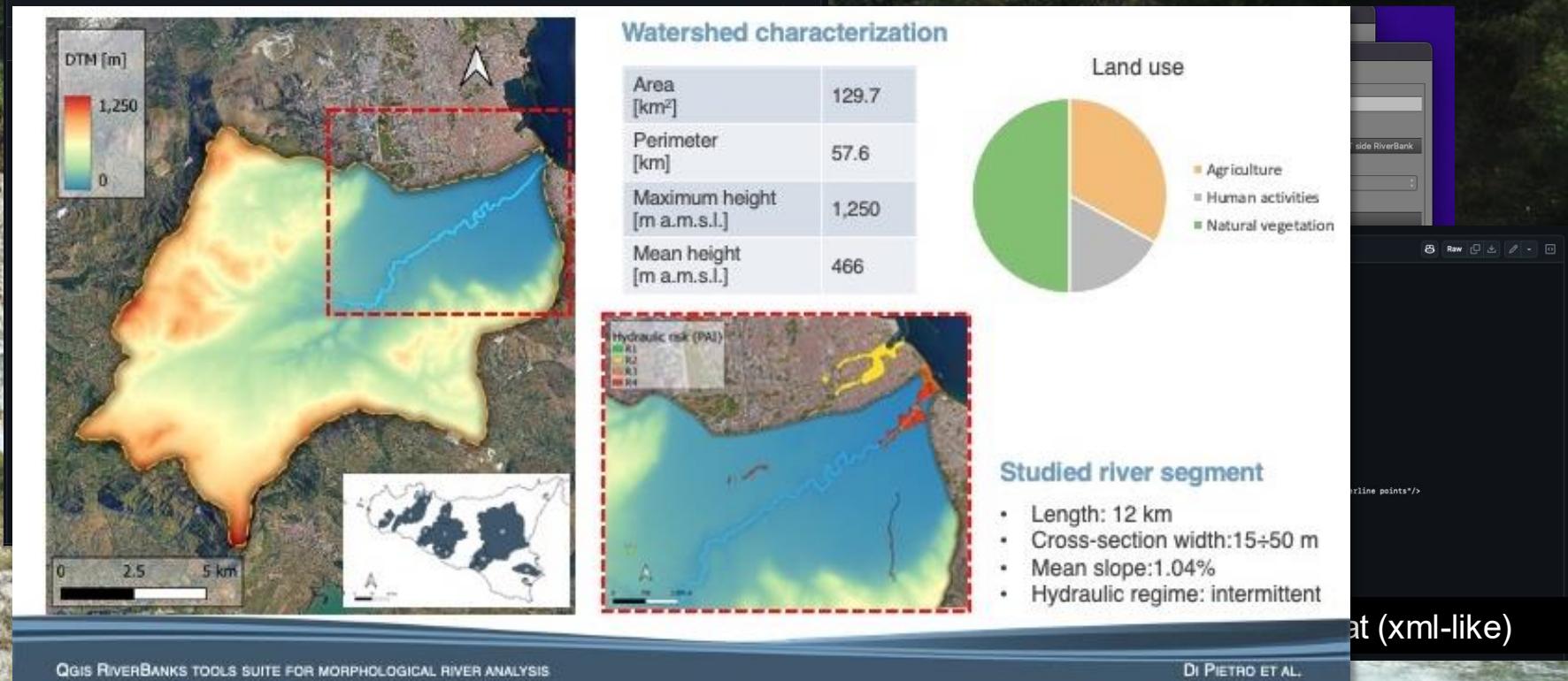


```
signer Modello - RB Safety Bands tool - DEV
Buffer su un solo lato
Properties Comments
Description Left side buffer
Layer in ingresso Utilizzo dell'ingresso del modello LEFT side RiverBank
Distanza 123 10.000000
Lato Sinistra
Code Blame 2385 lines (2385 loc) · 128 KB
<!DOCTYPE model>
<Option type="Map">
<Option type="Map" name="children">
<Option type="Map" name="native:fieldcalculator_1">
<Option type="Bool" name="active" value="true"/>
<Option name="alg_config"/>
<Option type="QString" name="alg_id" value="native:fieldcalculator"/>
<Option type="QString" name="color" value="" />
<Option type="Map" name="comment"/>
<Option type="QString" name="color" value="" />
<Option type="QString" name="component_description" value="" />
<Option type="Double" name="component_height" value="116.00000000000001" />
<Option type="Double" name="component_pos_x" value="1491" />
<Option type="Double" name="component_pos_y" value="576" />
<Option type="Double" name="component_width" value="149.9999999999997" />
<Option type="Bool" name="outputs_collapsed" value="true" />
<Option type="Bool" name="parameters_collapsed" value="true" />
</Option>
<Option type="QString" name="component_description" value="Assign SEZ-ID to centerline points" />
<Option type="Double" name="component_height" value="38.83716686191618" />
<Option type="Double" name="component_pos_x" value="1265" />
<Option type="Double" name="component_pos_y" value="524.185832434098" />
<Option type="Double" name="component_width" value="208" />
<Option name="depends" />
<Option type="QString" name="id" value="nativefieldcalculator_1" />
<Option name="outputs" />
<Option type="Bool" name="outputs_collapsed" value="false" />
<Option type="Text" name="source" value="nativefieldcalculator_1" />
```

Native .model3 file format (xml-like)

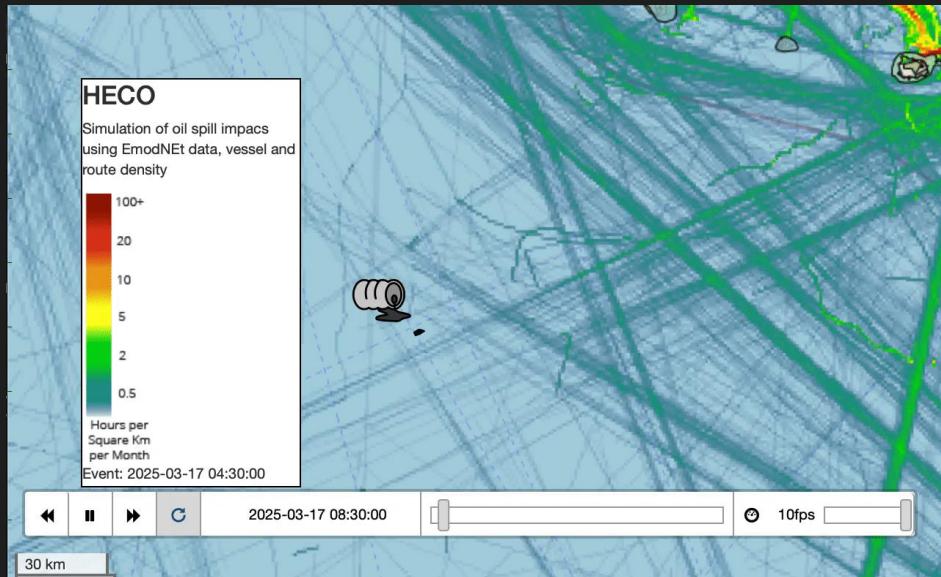
Case #1: NO-CODE approach

Qgis Riverbanks Tools



Case #2 – Hardcoded geoprocessing script

HECO: Here Comes the Oil! – proof of concept



Highlights

- 📍 A quick tool for simulating surface oil spill using 2D surface simplified LDPM model.
- ➡️ EMODnet data and real-time sea current forecasting used as forcings
- ⚠️ Identifies at-risk marine areas and vessel routes
- 🌐 Interactive web-GIS environment for instant visualization
- 🚢 Supports decision-making in emergency response

HECO: Here Comes the Oil! – Proof of concept



<https://github.com/SeaQuestTeam/HECO>

heco.py

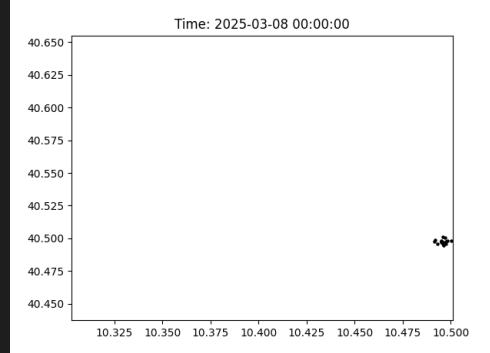
HECO / heco / heco.py

gianfrancodp updated function name

Code Blame 879 lines (681 loc) · 32.2 KB

```
1  ...
2 This module contains the functions developed for the HECO Proof of Concept
3 author: Gianfranco Di Pietro ~ PhD student at University of Catania
4 contributors: Martina Stagnitti, Massimiliano Marino, Elisa Castro, Sofia Nasca
5 supervisor: Rosaria Ester Musumeci
6 ...
7
8 import matplotlib.pyplot as plt
9 import matplotlib.colors as mcolors
10 import matplotlib.colorbar as mcb
11 from matplotlib.patches import FancyBboxPatch
12 from matplotlib.animation import FuncAnimation, PillowWriter
13 import pandas as pd
14 import numpy as np
15 import os
16 import cartopy.crs as ccrs
17 import cartopy.io.img_tiles as cimgt
18 import xarray as xr
19 import ssl
20 import ipywidgets as widgets
21 from IPython.display import display
22 import pyproj
23 from glob import glob
24 from geopandas import GeoDataFrame
25 from shapely.geometry import Point, LineString
26 import yaml
```

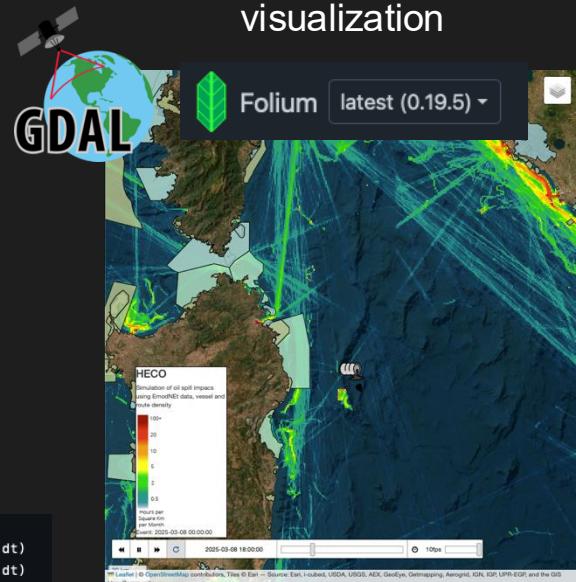
Lagrangian Dispersion
Particle Model



```
472     # Compute the new position of the oil spill
473     x_new = x + u * dt + np.random.normal(0,1)* np.sqrt(2 * D * dt)
474     y_new = y + v * dt + np.random.normal(0,1)* np.sqrt(2 * D * dt)
```



Data web-gis
visualization



Google Developer Group
Gela

Case #2 – Hardcoded geoprocessing script

HECO: Here Comes the Oil! – Proof of concept



<https://github.com/SeaQuestTeam/HECO>

Deploy in EDITO's cloud containers

EDITO Datalab

My Services

Access your running services

Services are supposed to be shut down as soon as you stop using them.

Refresh + New service Delete all Events

Running services

HECO-test

Service: Jupyter-pyton-ocean-science Started: yesterday

Open

HECO.ipynb

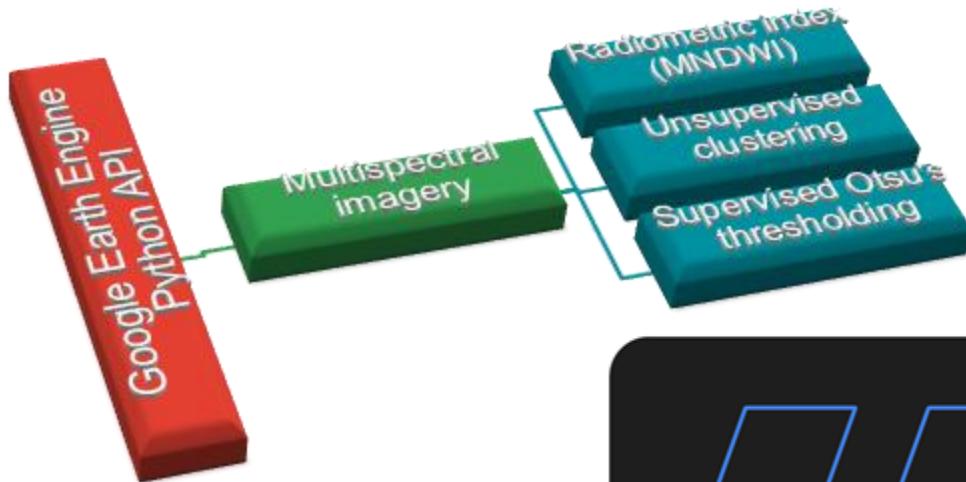
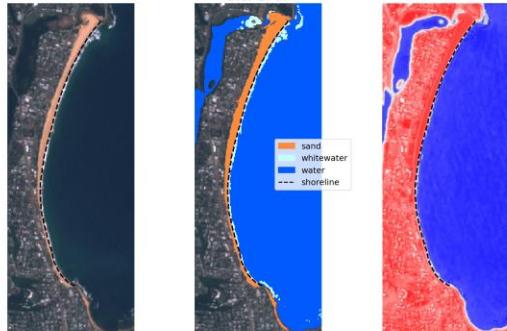
--> This Notebook will be used for the hackaton Deliverable

HEMODnet
OSL4.0
OPEN SEA LAB HACKATON 2020

HECO

HEre Comes the Oil

| / HECO / heco / | | |
|-----------------------|--|------------|
| Name | | Modified |
| heco_map.html | | 2 min. ago |
| heco_results.csv | | 2 min. ago |
| heco_results.geojson | | 2 min. ago |
| heco_results.gif | | 2 min. ago |
| HEKO_TEST.nc | | next yr. |
| heco-polygons.geojson | | 2 min. ago |
| HECO.ipynb | | 2 min. ago |



Satellite Derived Shoreline

ML-Segmentation from Google Earth Engine datasets





1. Registrare un Progetto Google cloud
2. Abilitare Python API
3. Importare in python/javascript: ee (aka. Earth Engine)

API Reference

Overview

Client Libraries

- JavaScript/Python
- ee.Algorithms
- ee.Array
- ee.Blob
- ee.Classifier
- ee.Clusterer
- ee.ConfusionMatrix
- ee.Date
- ee.DateRange
- ee.Dictionary
- ee.ErrorMargin
- ee.Feature
- ee.FeatureCollection
- ee.Filter
- ee.Geometry
- ee.Image
- ee.Image**
- abs
- acos
- add
- addBands
- and
- arrayAccum

ee.Image



Send feedback

An object to represent an Earth Engine image. This constructor accepts a variety of arguments:

- A string: an EarthEngine asset id,
- A string and a number: an EarthEngine asset id and version,
- A number or ee.Array: creates a constant image,
- A list: creates an image out of each list element and combines them into a single image,
- An ee.Image: returns the argument,
- Nothing: results in an empty transparent image.

Usage

`ee.Image(args)`

Returns

Image

Argument

Type

Details

`args` *ImageList, optional*

Constructor argument.

Examples

EE contain more than 40 years of satellite and scientific data

Earth Engine Data Catalog

Earth Engine's public data catalog includes a variety of standard Earth science raster datasets. You can import these datasets into your script environment with a single click. You can also upload your own raster data or vector data for private use or sharing in your scripts.

Looking for another dataset not in Earth Engine yet? Let us know by [suggesting a dataset](#).

[Filter list of datasets](#)

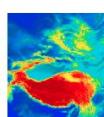
2000 Greenland Mosaic - Greenland Ice Mapping Project (GIMP)



This dataset provides a complete 15 m resolution image mosaic of the Greenland ice sheet. It was derived from Landsat 7 ETM+ and RADARSAT-1 SAR imagery from the year 2000. The mosaic creation methods include a combination of image cloud masking, pan sharpening, image sampling and resizing.

arctic gimp greenland imagery nasa polar
aster elevation
elevation-topography
emissivity geophysical
infrared jpg

AG100: ASTER Global Emissivity Dataset 100-meter V003



The Advanced Spaceborne Thermal Emission and Reflection Radiometer Global Emissivity Database (ASTER-GED) was developed by the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL), California Institute of Technology. This product includes the ASTER Global Emissivity Database (AG100). This product includes the ASTER Global Emissivity Database (AG100).

ahn dem elevation
elevation-dem
elevation-topography
geophysical lidar
geophysical jpg

AHN Netherlands 0.5m DEM, Interpolated



The AHN DEM is a 0.5 m DEM covering the Netherlands. It was derived from aerial photography in the spring between 2007 and 2012. It contains ground level samples and samples taken above ground (such as buildings, bridges, trees etc.) removed. This version is ...

ahn dem elevation
elevation-dem
elevation-topography
geophysical lidar
geophysical jpg

AHN Netherlands 0.5m DEM, Non-Interpolated



The AHN DEM is a 0.5 m DEM covering the Netherlands. It was derived from aerial photography in the spring between 2007 and 2012. This version contains both ground level samples and samples taken above ground level (such as buildings, bridges, trees etc.) removed. This version is ...

ahn dem elevation
elevation-dem
elevation-topography
geophysical lidar
geophysical jpg

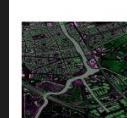
AHN Netherlands: 0.5m DEM, Raw Samples



The AHN DEM is a 0.5 m DEM covering the Netherlands. It was derived from aerial photography in the spring between 2007 and 2012. This version contains both ground level samples and samples taken above ground level (such as buildings, bridges, trees etc.) removed. This version is ...

ahn dem elevation
elevation-dem
elevation-topography
geophysical lidar
geophysical jpg

AHN: Netherlands AHN 0.5m



The Actual Hoogtebestand Nederland (AHN) is a dataset with digital elevation data for the whole of the Netherlands. Elevation

AHN4: Netherlands AHN 0.5m



The Actual Hoogtebestand Nederland (AHN) is a dataset with digital elevation data for the whole of the Netherlands. Elevation

ALOS DSM: Global 30m v3.2



Starting from the night of January 1st, 2004, based on the requirement of the Japanese Ministry of Land, Infrastructure, Transport and Tourism (MLIT), the Japan Aerospace Exploration Agency (JAXA) implemented the Advanced Land Observing Satellite (ALOS) with the PALSAR-2 StripMap Level 2.1

ALOS-2 PALSAR-2 StripMap Level 2.1



This dataset is contains orthorectified images from the Advanced Land Observing Satellite (ALOS) with the PALSAR-2 sensor on board. The PALSAR-2 sensor is a dual polarization L-band Synthetic Aperture Radar (SAR) instrument.

ALOS/AVNIR-2 ORI

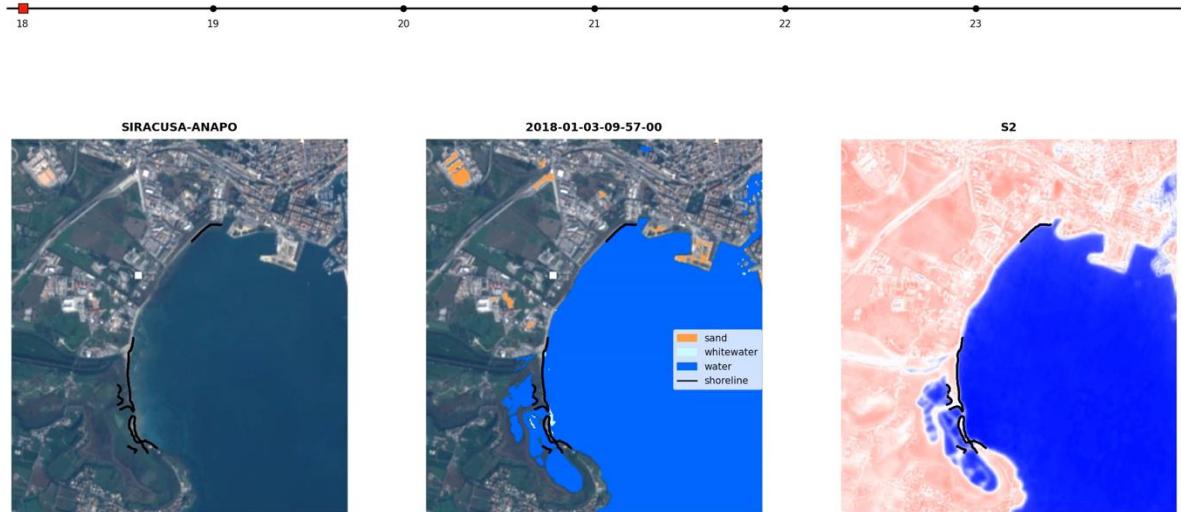


This dataset is contains orthorectified images from the Advanced Land Observing Satellite (ALOS) with the AVNIR-2 sensor on board. The AVNIR-2 sensor is a visible and near-infrared (VNIR) imager.

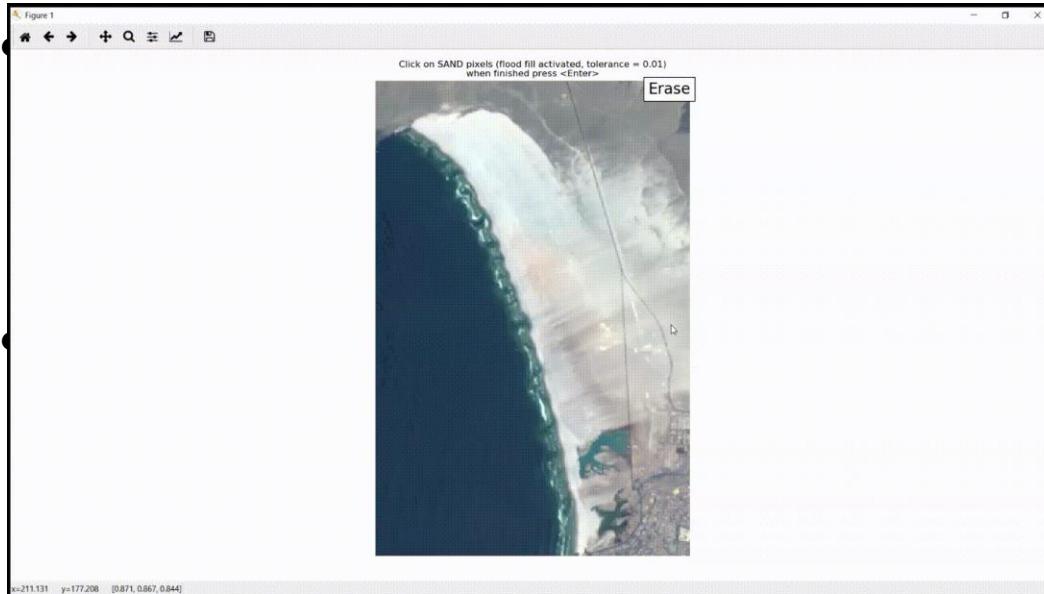
CASE #3 – Machine Learning for Remote Sensing

Machine learning in Python: The power of scikitlearn and matplotlib

- Classification:
Identifying which category an object belongs to
- Used pre-trained neural network pretrained classifier with *joblib* function

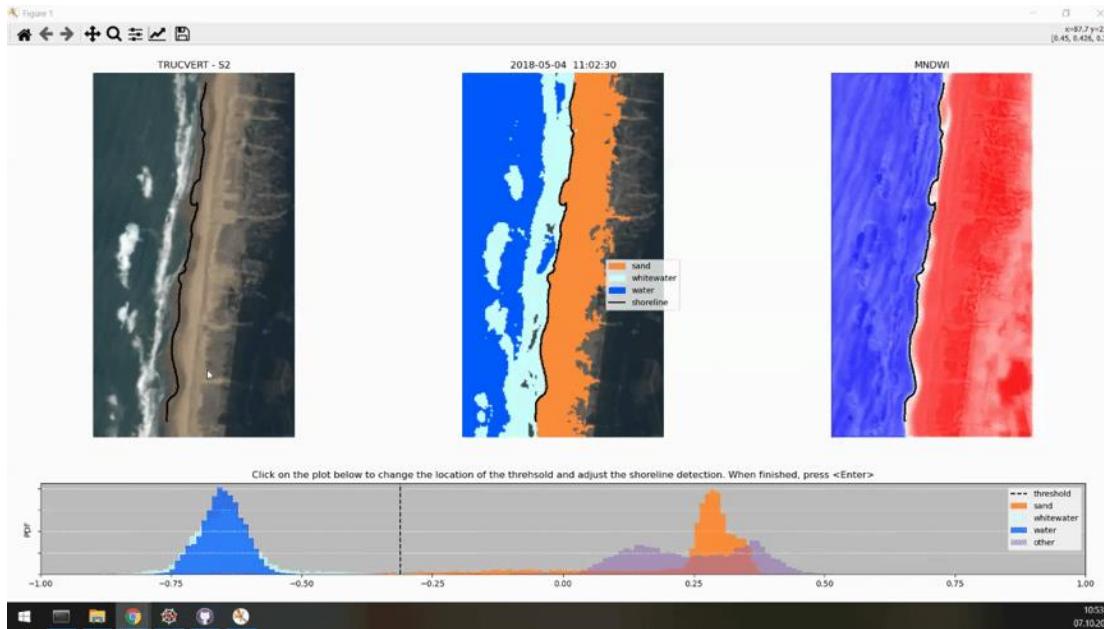


Machine learning in Python: The power of scikitlearn and matplotlib



- Train new classifier with manual labeling and matplotlib inside a jupyter notebook.
- Supervised threshold adjusting for better segmentation

Machine learning in Python: The power of scikitlearn and matplotlib



- Train new classifier with manual labeling and matplotlib inside a jupyter notebook.
- Supervised threshold adjusting for better segmentation

 matplotlib



Grazie per
l'attenzione



@gianfrancodp

