Sustainable and Innovative Approaches for Environmental Safety
Summer School – 11-13 July 2022 – Tbilisi, Georgia

# Spatial and spatio-temporal Environmental Modeling: Equipping your open source Toolbox

Marco Ciolfi

Research Institute on Terrestrial Ecosystems of the National Research Council

# CNR – IRET

## Institute of Research on Terrestrial Ecosystems of the National Research Council of Italy

Nearly 200 employees in seven offices with multiple lab facilities (Porano, Roma, Firenze, Pisa, Sassari, Lecce, Napoli) following several research lines: Molecular biology and genetics; IRMS: stable isotopes mass spectrometry; GIS, remote sensing and geospatial modeling; Landscape dynamics; Ecophysiology, groundwater ecology and soil dynamics; Plant and animal diversity.

Porano headquarters – a 1700 villa and park.

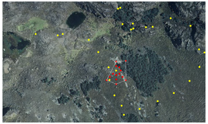
Consiglio Nazionale delle Ricerche

# CNR – IRET – geoSpatiaLab

## Activities

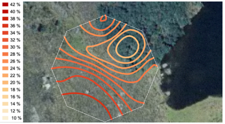
The lecturer is mainly involved in the geoSpatiaLab and IRMS activities in Porano headquarters.

**Sampling**



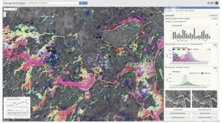
Pattern design and field survey support.

**Isoscapes**



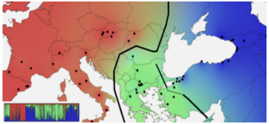

Stable isotopes ratios spatial distribution and time dynamics.

**Dynamical Modeling**



Time series analysis, PDE, AI algorithms.

**Landscape Genetics**



Plant populations genotype distribution and spatial evolution.

The **geoSpatiaLab** group is involved in most number-crunching activities of the institute. A special experience has been developed in the field of isoscapes (i.e. **iso**topic land**scapes**), both in spatial-only and spatio-temporal modeling, in conjunction with the stable isotopes lab.

# CNR – IRET – geoSpatiaLab & IRMS

## Equipment

### geoSpatiaLab

The **geoSpatiaLab** group is equipped with a totally standard bunch of desk/laptops and a dedicated workstation (Apple MacPro Xeon hexacore, 64GB ram).
Other than in-house software development, various scientific packages are exploited:

- Octave, R / RStudio, SPSS (licensed),
- QGIS, QField, GRASS, ArcGIS (licensed),
- SNAP, Google Earth Engine (free?),
- PostgreSQL, MySQL, Spyder, Eclipse.

### IRMS

The IRET stable isotopes lab is equipped with three mass spectrometers for H, C, N, O and S analysis, especially tuned to plant physiology:

- **Isoprime** continuous flow with Eurovector PyrOH and Gilson Miltiprep,
- **Isotech Isochrome** continuous flow with Carlo Erba NA1500 and GV dilutor,
- **Isotech SIRA10** dual inlet continuous flow with HPLC interface.

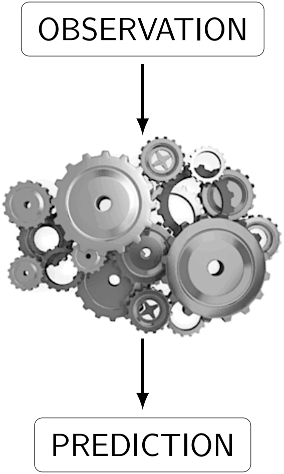
# Modeling

**modeling** (modelling) / mod·el·ing | 'mäd$^{ə}$lŋ — noun:

- ▷ the work of a fashion model
- ▷ the act of designing or imitating forms
- ▷ **the devising or use of logical and mathematical techniques to assist calculations and predictions**
- ▷ the activity of making three-dimensional models
- ▷ the action of a person that serves as a prototype for an artist . . .

from Merriam-Webster dictionary

i.e. **modeling –** a reliable, <u>documented</u>, replicable [in a finite number of calculable steps] procedure for inferring quantitative predictions from a set of input data.




The general model of modeling.

# Modeling in Environmental Sciences

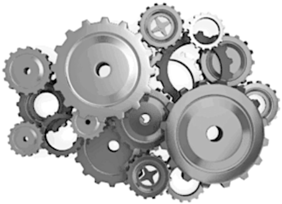
Environment-related modeling includes:

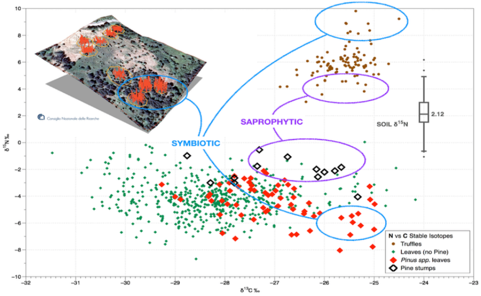
- ▷ spatial (georeferenced) data,
- ▷ time series / historical records,
- ▷ inhomogeneous spot observations,
- ▷ disputed or weak interpretations.

The expected outputs are often required as maps and forecast/hindcast series, or both.
In order to cope with such diverse *ingredients*, researchers usually gather a bunch of software:

- ▷ Spreadsheets (never the wisest choice),
- ▷ Computational and statistical packages,
- ▷ Relational database management systems,
- ▷ GIS and remote sensing tools.

Depending on each researcher's taste and skills, ad-hoc software development can be included as a part of the modeling process.






Modeling results are often presented as graphs.

# The Open Approach to Modeling

## It is not a Good vs Evil Clash

Adopting an **open** approach for data and procedures management is in strict accordance with the acknowledged **scientific good practices**: open source means that everyone knows what a procedure or program actually does, implying **replicable algorithms**. Open data publishing ensures a **fair data availability policy**, while safeguarding the authors's intellectual rights.

Exploiting open products in one's workflow implies an **ethical commitment** towards the **open disclosure** of methods and results. It does not imply open publishing author's fees, though!

An open approach does not completely forbid the use of proprietary software, but its use should be restricted to ancillary and cosmetic tasks, as data preprocessing or graphs embellishment.

Last but not least: open software comes for free, without license management fusses.

# The Open Marketplace: Spreadsheet, CSV & Databases

## Spreadsheet

| ID | SPECIES | SHIFT | UTMX | UTMY | ELEV | DAY | D13C | D15N | LAI |
|---|---|---|---|---|---|---|---|---|---|
| T1-P1 | PIN | 6W 1SW | 738525 | 4746921 | 626 | 64 | -27.67 | -5.41 | 0.05 |
| T1-P4 | PIN | 7SE 2S | 738534 | 4746916 | 625 | 64 | -28.10 | -5.38 | 0.04 |
| T1-Q3 | QRC | 7W 1SW | 738524 | 4746921 | 626 | 64 | -30.50 | -3.39 | 0.01 |
| T1-Q4 | QRC | 8W | 738522 | 4746921 | 627 | 64 | -29.83 | -2.33 | 0.01 |
| T1-Q5 | QRC | 10W 2SW | 738520 | 4746920 | 627 | 64 | -30.04 | -3.24 | 0.01 |
| T8-Q4 | QRC | 7.5E 3.5NE | 738397 | 4747047 | 656 | 30 | -27.38 | -4.52 | 0.01 |
| T8-Q7 | QRC | 1.5S | 738389 | 4747043 | 658 | 30 | -29.88 | -4.40 | 0.01 |
| T9-Q3 | QRC | 8W 1NW | 738378 | 4746959 | 655 | 19 | -28.10 | -3.28 | 0.01 |
| T9-Q4 | QRC | 12W 0.5SW | 738375 | 4746957 | 656 | 19 | -30.18 | -4.63 | 0.01 |
| T9-S1 | SOR | 14.5SE 1.5S | 738396 | 4746947 | 651 | 19 | -28.46 | -2.05 | 0.01 |

The preferred choice for simple calculations.

- ✗ Random data type behaviour, especially concerning dates and times.
- ✓ Simple calculations are quick and easy.
- ✗ Quick often rhymes with dirty.
- ✗ There is no logical connection between sheets, nor object-oriented tables.
- ✓ Open alternatives exist, e.g. Gnumeric, OpenOffice and LibreOffice.

## CSV and RDBMS

```
ID,SPECIES,SHIFT,UTMX,UTMY,ELEV,DAY,D13C,D15N,LAI
T1-P1,PIN,6W 1SW,738525,4746921,626,64,-27.67,-5.41,0.05
T1-P4,PIN,7SE 2S,738534,4746916,625,64,-28.10,-5.38,0.04
T1-Q3,QRC,7W 1SW,738524,4746921,626,64,-30.50,-3.39,0.01
T1-Q4,QRC,8W,738522,4746921,627,64,-29.83,-2.33,0.01
T1-Q5,QRC,10W 2SW,738520,4746920,627,64,-30.04,-3.24,0.01
T8-Q4,QRC,7.5E 3.5NE,738397,4747047,656,30,-27.38,-4.52,0.01
T8-Q7,QRC,1.5S,738389,4747043,658,30,-29.88,-4.40,0.01
T9-Q3,QRC,8W 1NW,738378,4746959,655,19,-28.10,-3.28,0.01
T9-Q4,QRC,12W 0.5SW,738375,4746957,656,19,-30.18,-4.63,0.01
T9-S1,SOR,14.5SE 1.5S,738396,4746947,651,19,-28.46,-2.05,0.01
```

**Comma Separated Values** standard is the true *lingua franca* of data storage.

- ✓ It is the simplest exchange format.
- ✓ It is extremely easy to read and write.

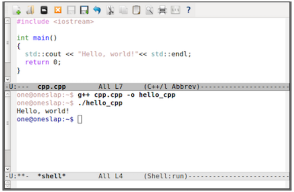
For big and complex data consider a **Relational Database Management System**.

- ✓ It accommodates logical relationships.
- ✓ Each value belongs to a codified *data type*.
- ✗ Not so easy to master.

# The Open Marketplace: Editors

Data editing is of paramount importance in modeling. Files for modeling can be roughly categorized as **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchage) and **binary**. Structured data should be stored in a **relational database**.

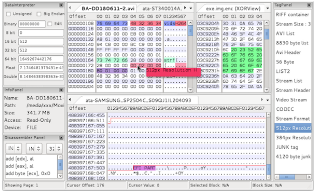
## ASCII




ASCII editors exist since the dawn of computer epoch, with plenty of open alternatives.
A popular choice is Emacs from the GNU Foundation.

✓ **It is perhaps the most important tool for modeling.**

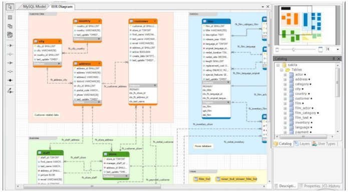
## HEX



Editing binary files happens rarely, if ever. However, be prepared with a suitable tool.
An open editor is wxHexEditor.
The GNU alternative is poke.

✗ Avoid binary editing.
✓ Handy to peep file headers.

## RDBMS



A few open/free options exist:

- ▷ Oracle – great but bulky.
- ▷ MySQL – a common choice.
  - ✓ robust and documented.
- ▷ PostgreSQL – truly open.
  - ✓ excellent for geodata.

# The Open Marketplace: Geographical Information Systems

Environmental modeling often includes a geographical component. Spatial datasets are not the easiest kind of data to be dealt with, they deserve a GIS tool for proper treatment.

The first open alternative has long been the Geographic Resources Analysis Support System (GRASS GIS). ✗ Not too easy to use.

The Open Source Geospatial Foundation provides a Geospatial Data Abstraction Library (GDAL). This library is of paramount importance for spatial data management.

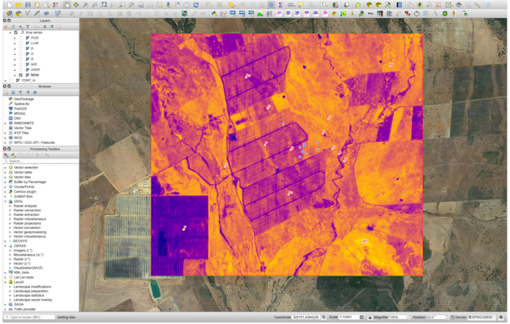
✓ Cross platform. ✗ very hard to master.

The R project for statistical computing also offers some GIS capabilities.

✓ State of the art geostatistical algorithms.

✗ R lacks a graphical GIS interface.

**QGIS**

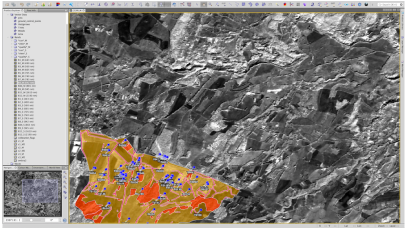


The software of choice is QGIS. ✓ It also serves as an interface to GRASS GIS and GDAL tools.

An excellent proprietary alternative is ArcGIS. It is fast and reliable but with a cumbersome license management. ✗ MS Windows only. ✗ $$$

# The Open Marketplace: Remote sensing

## SNAP

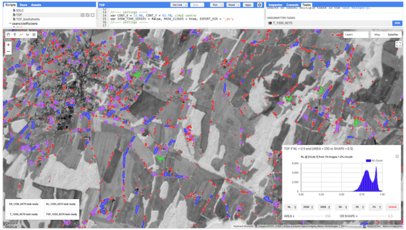


The European Copernicus programme, other than Sentinel satellites data, provides an open software for any sort of image processing: SNAP, the **S**e**N**tinel **A**pplication **P**latform.

- ✓ Finest control on image processing.
- ✗ Hard science: steep learning curve.
- ✗ Limited GIS capabilities.

MODIS and USGS Landsat are also available.

## Google Earth Engine

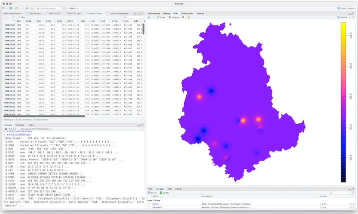


GEE is a shifting paradigm in satellite image processing. Endless storage at unprecedented computing speed. ✗ Not open!

- ✓ It is free, registering as developer.
- ✓ images stay on Google servers.
- ✓ Unbelievably fast for time series.
- ✗ Some advanced functions ask for cash.
- ✗ Programming is not as easy as it seems.
- ✓ On-server advanced zonal statistics.

# The Open Marketplace: Number Crunching

## R / R Studio



The R project for statistical computing of the R Foundation is **the** choice in many fields of Environmental modeling.

- ✓ Countless libraries for all statistical needs.
- ✓ Works with spatial data, but it's not a GIS.
- ✗ Purists' R interaction is a bit awkward.

R Studio is an open cozy alternative offering integrated interface and package management.

## GNU Octave


GNU Octave is the computing environment of the GNU Foundation, closely resembling the MathWorks proprietary MATLAB environment.

Octave offers many sets of packaged functions through Octave Forge.

- ✓ Well-reputed as a modeling language.
- ✓ Widely used, well documented, libraries.
- ✓ Understands spatial datasets.

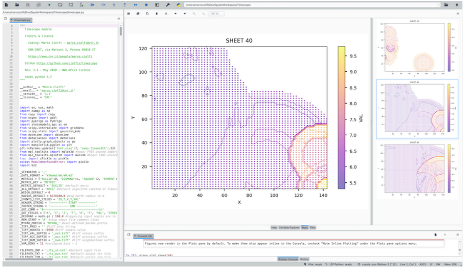
# The Open Marketplace: Programming

As a matter of fact, ecological modeling means computing. Sometimes this can be done in R or Octave, or even in a spreadsheet, but most often it requires the development of ad-hoc procedures using a programming language.

The choice of a particular language is a matter of taste and needs. Assessed choices include:

- **Fortran, C, C++** Enduring classics, these languages require specific training. It is easy to find published models which require end-user compilation, if not some editing.
- **Python** Perhaps the most widely used nowadays scientific computing language, it offers prepackaged functions for everything.
- **Javascript** Very popular in client/server programming. Among other web services, it is also the language of Google Earth Engine.

Newbies should consider starting a programming career with Python. ✓ Smooth learning curve.




The Spyder python IDE.

Anaconda offers a cross-platform **Integrated Development Environment** for scientific computing in python, including the Spyder development tool and the Jupyter notebook.

✓ Handy package management system.

# From Tools to Toolbox

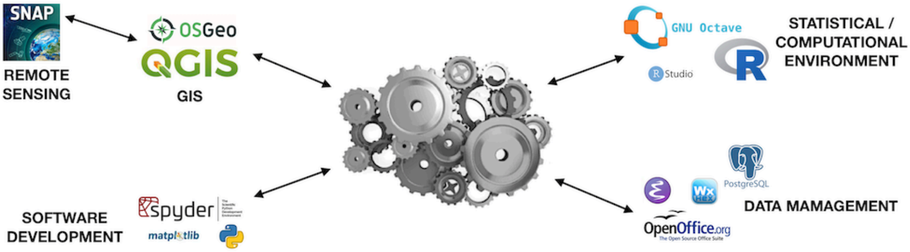

**So the shopping is done – for free. Now what?**

We have loaded our computers with tons of excellent software, the newest, most-performant instruments, each one with its own tastes, especially with regards to data formatting and storage. We must now assemble our brand-new downloads into a coherent toolbox.

A general note: keep your software updated – bug fixing occurs almost daily.

# How to Assemble a Modeling Environment

## From Tools to Toolbox




### Find a common data standard.

There is a simple solution for interfacing the diverse software packages: finding a common data format (i.e. as common as possible). The suggestion is to abandon the seemingly comfortable spreadsheet way of organising data towards the old-fashioned **CSV** (**C**omma **S**eparated **V**alues) standard. For really large and involved data consider a relational database, sticking to the **geoTiff** format for all georeferenced image needs. ✗ Raw binaries is hard stuff!

# Structuring your Data – Objectwise Thinking

## From Tools to Toolbox

Follow a logical structure for your data. Each measurement record needs an **ID**, i.e. its label that must be **unique** and **not-null**. Design your data following an **entity-relationship** model, as if they were tables in a relational database, establishing **primary** and **foreign keys**. Use a file, or a table, for each logical object. Mind **metadata** and explain what is expected to be found in each field, including units.

**Leaves – a very bad table**

| Label | Coordinates | Tree | $\delta^{13}$C ‰ | $\delta^{15}$N |
|---|---|---|---|---|
| leaf 1 | 12.34, -38.9 | the first Pine | -24.21 | 3.27 |
| 2 | 38.85 W, 12.36 N | spruce no.4 | -26.11 | 4.00 |
| L.4/3 | (562374, 1564232) | tallest pine | -25.25 | 3.21 |
| maple 23 | 2 m south of maple 1 | Maple 2 | -24.57 | 2.84 |

This table is a collection of bad practices:

- ✗ Non-consistent labels.
- ✗ Coordinates are expressed in a variety of formats.
- ✗ Leaves and trees properties are intermixed.
- ✗ Trees descriptions are inconsistent.
- ✗ $\delta^{13}$C **‰** isn't wrong, but units belong to metadata.

**Leaves**

| ID | Tree | $\delta^{13}$C | $\delta^{15}$N | ... |
|---|---|---|---|---|
| **L001** | **M1** | -25.34 | 4.31 | ... |
| **L002** | **M1** | -25.34 | 3.45 | ... |
| **L003** | **M1** | -26.34 | 3.98 | ... |
| **L004** | **M1** | -24.34 | 4.65 | ... |

**Trees**

| ID | Lat | Lon | Elev | Species |
|---|---|---|---|---|
| **F1** | 42.45 | 12.43 | 1820 | Fagus sylvatica |
| **M1** | 42.23 | 12.22 | 1413 | Acer opalus |
| **M2** | 42.87 | 12.54 | 1320 | Acer opalus |
| **N1** | 42.42 | 12.01 | 1008 | Juglans regia |

✓ Clean, structured data with **primary** and **foreign** keys.

# Tricks of the Trade

## From Tools to Toolbox

Keeping **clean datasets** is of the utmost importance for modeling. A few suggestions, especially about primary and foreign keys and about date/time management.

- ▷ Be consistent with the units. Specify once and for all your conventions and stick to them.
- ▷ Avoid spreadsheets since they try to interpretate your data types.

### Keys

- ▷ Keys are essential for database integration, do not miss a consistent ID labelling convention: before or later you will need a RDMBS.
- ▷ Whenever possible, use fixed length strings.
- ▷ Use numerical IDs only if you have a really large number of records.
- ▷ Avoid spaces, use the **underscore** instead.
- ▷ Do not begin the ID with a number.
- ▷ Build your IDs such that they represent a little abstract of the record, i.e. **L2T005D42** could be the second leaf of the fifth tree, sampled on day 42 of data collection.

### Dates


- ▷ There is only one reliable data format: **YYYY-MM-DD**. Period.
- ▷ Spreadsheets are particularly prone to fancy date/time interpretations.
- ▷ **Unix time** (the number of seconds elapsed since 1 January 1970) is perfectly fine, but hard to be read by humans.

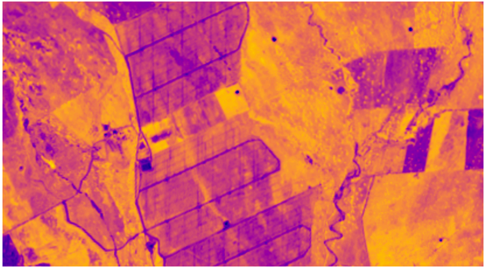
# Working with Images

## From Tools to Toolbox

Digital images come in a variety of formats. Only **tiff** and **geotiff** images are to be considered, in fact:

- **tiff** stands for *tagged image file format*, a versatile standard that includes **binary data** as well as descriptive **tags**.
- **tiff** is typed: Int16, Uint16, Float32 . . .
- **geotiff** is a tiff-derived standard for geodata.
- **jpeg**, i.e. *joint photographic experts group*, stores lightweight but ✗ lossy images.
- **png** – *portable network graphics* is a versatile, web-oriented indexed format, not suitable for geodata. ✓ **png** can be georeferenced.

Satellite imagery is generally distributed as multiband **geotiff** files. Different bands can have different data type and resolution.



The **gdalinfo** command reads and formats the **geotiff** tags:

```
Driver:  GTiff/GeoTIFF
Files:  NIR_ts.tif NIR_ts.tif.ovr NIR_ts.tif.aux.xml
Size is 417, 373
PROJCRS["WGS 84 / UTM zone 30N",
   BASEGEOGCRS[DATUM["World Geodetic System 1984",
      ELLIPSOID["WGS 84",6378137,298.257223563,
         LENGTHUNIT["metre",1]]],
      PRIMEM["Greenwich",0,
         ANGLEUNIT["degree",0.0174532925199433]],
      ID["EPSG",4326]],
  ... ...
```

# Tools vs Data

## Setup for Modeling

Don't be overwhelmed by the huge amount of available tools. Being free does not mean that you have to use them all at the same time.

A **model** is a complex entity, so it is important to have a clear vision of

- The input data: **observations**, literature data and network-availabe datasets.
- The modeling **tools**, but set them aside for now.
- The **expected output**, including technical stuff as file types, etc.
- A model **validation procedure**.

Keep data and tools separate; don't be mislead by your own taste towards a single tool.

# All Starts with Data . . .

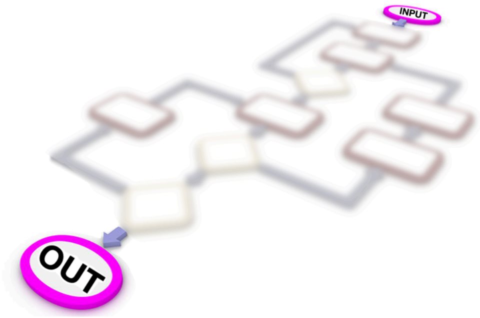
## Setup for Modeling

At first, keep your modeling toolbox closed, focusing only on input and output data.

Choose a known data format, and stick to it:

- **CSV** files for observarions records – easy to convert into R's **data frames**.
- An established **database** if the data structural complexity is demanding.
- An R **.RData** object to keep data tidy.
- **tiff** files for images and **geotiff** for spatial **raster** data; **geodatabase** and **shapefile** for spatial vector data; raw **binary** files if need be.
- **mesh** multidimensional numerical data, very popular with meteorology and time series.

Figure out the storage needs of input and expected output data, checking your hardware.



# . . . The Algorithm Follows . . .

## Setup for Modeling

At this stage, try to translate your model in a symbolic language, be it an old style flowchart, a pseudo-code listing or just a paper sketch.

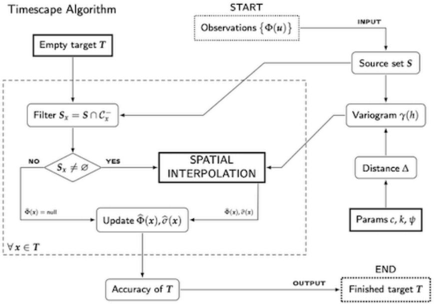
**It's not yet time for choosing the tools!**

The resulting algorithm should be:

- ▷ As simple as possible,
- ▷ **Calculable, in a finite number of steps**,
- ▷ Readable, try asking a peer's opinion,
- ▷ Programming language independent,
- ▷ **Tool independent** – nonetheless, at this stage, begin thinking about your tools.

At this stage, also think about how to **validate** your model output, what kind of **test**, etc.

| | for each $x$ in $T$ do | dashed box of figure 8 |
|---|---|---|
| 1 | $C_x^- \leftarrow u \in S \mid \Delta(u,x) < \infty$ | eq.7 eq.9 fig.3 |
| 2 | $S_x \leftarrow S \cap C_x^-$ | filtering – eq.11 fig.7 |
| 3 | $\eta(x) \leftarrow \lvert S_x \rvert$ | eq.15 |
| 4 | if $S_x = \varnothing \Longrightarrow \widehat{\Phi}(x) \leftarrow$ null | |
| | else $\widehat{\Phi}(x) \leftarrow \sum_{S_x} W(u...)\, \Phi(u)$ | estimate – eq.14 |
| | if **interp.** allows $\widehat{\sigma}(x) \leftarrow \sqrt{\text{var}}$ | Kriging… |
| | else $\widehat{\sigma}(x) \leftarrow$ null | IDW… |
| 5 | update $x$: $[\ \widehat{\Phi}(x),\ \widehat{\sigma}(x),\ \eta(x)\ ]$ | |



# . . . Then it's Time to Pick the Tool(s)

## Setup for Modeling

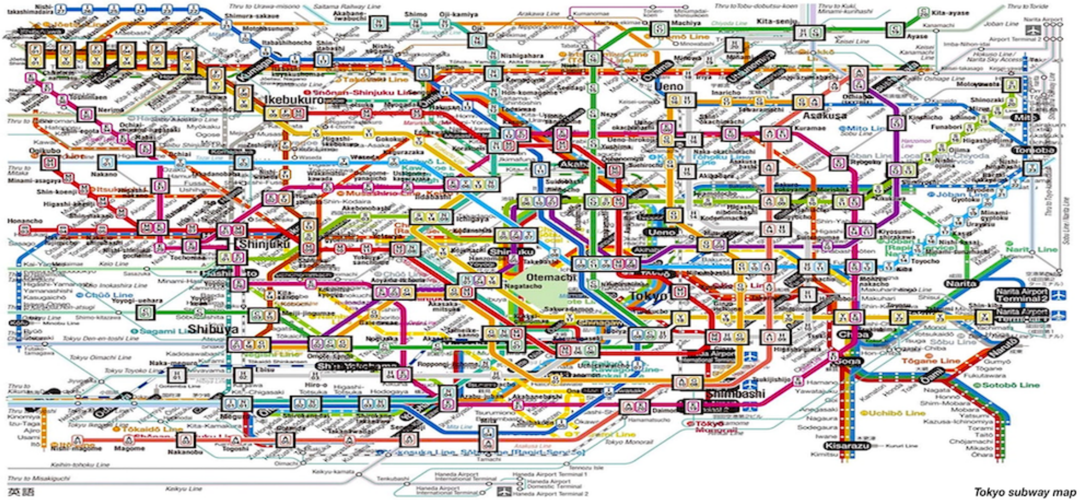
Every **model** should be **centred about a main tool**, depending on the data types and the kind of expected calculations involved:

- ▷ **R**, maybe **R Studio**, if statistics and functional programming are dominant.
- ▷ GNU **Octave** for higher maths algorithms.
- ▷ **QGIS** if the focus is on spatial data; it copes with mesh data too.
- ▷ A dedicated programming **IDE** for true craftsmanship; avoid command line tools.

✓ If your datasets are stored into a database, each of the mentioned tools can access them easily. RDMBs can act as a data hub.

✗ Use **spreadsheets** as the last resort, paying attention to bad typization and random editing.

# But don't do this!




**A good recipe should be as simple as possible, based on the fewest ingredients.**

# A Spatial Modeling Environment

## Setup for Modeling

Environmental modeling is more and more centred on spatial data and remote sensing.

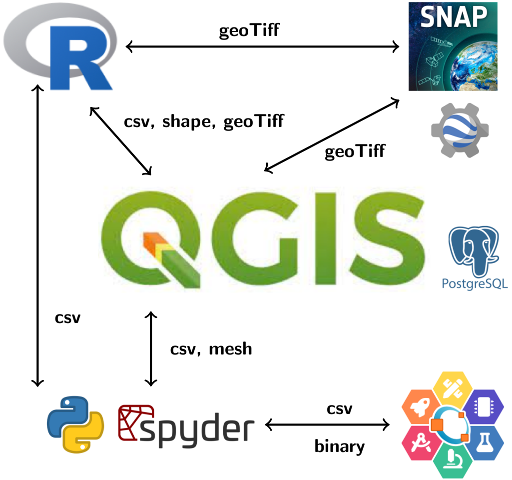
In such a context **QGIS** is the most obvious choice as the preferred tool.

- ✓ QGIS talks with **R** and many other tools,
- ✓ it connects with **RDBMs**,
- ✓ it is based on **python**.

Remote sensing data can be retrieved through **SNAP** and **Google Earth Engine**.

It is of the utmost importance to be able to **exchange data** among the components of the modeling environment: **use suitable formats**.

- ✗ Avoid **mixing spatial reference systems**. It could be done, but it **calls for disaster**.
- ✗ Spatial datasets can be bulky.




A QGIS-centric modeling environment. The main file exchange formats are suggested. ✓ Good old **csv** rules!

# Case Study: Mycorrhiza - Host Tree Interaction

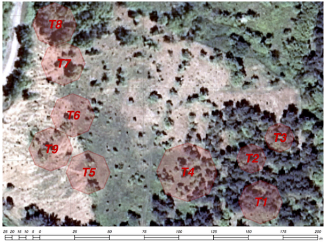
This case study shows how a modeling tool was developed starting from an actual research.

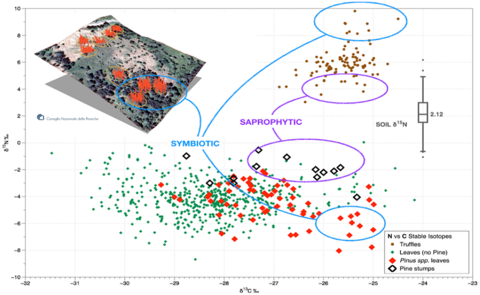
**scope** Finding the mycorrhiza–host tree relationships from isotopical measurements in a thinned pine forest. *Tuber aestivum* (edible black truffles) vs pines, oaks, maples... possibly **symbiotically or saprophytically connected**.

**input** A georeferenced dataset of $\delta^{13}$C and $\delta^{15}$N measurements of fungi, leaves and soil, along all the truffles ripening season, showing a **spatial and temporal entangled variability**.

**output** the expected output was a map of supposed symbiotic host trees represented as a **probability of symbiosis**, a real number ranging from **0 to 1**.

**validation** field checking.



# Case Study Background: Isotopic Fractionation

We adopted an extremely simplified fungi–tree relationship representation.

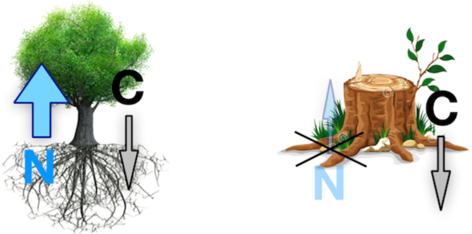
In the **symbiotic** case, mycorrhiza gets photoassimilates from trees: **carbon stuff up to down**, while trees get nitrogen compounds from the soil via the intermediation of fungi: **nitrogen stuff down to up**, this is typical of living healthy plants.

In the **saprophytic** case, mycorrhiza exploits whatever (the remnants of) trees can offer in terms of carbon, offering nothing in return.

In order to find the kind of relationship, we needed the spatial distribution maps of soil, fungi and leaves $\delta^{13}$C and $\delta^{15}$N.

We needed a tool to evaluate **ISOSCAPES**.
[doi:10.1007/978-90-481-3354-3]

Every chemical reaction is known to prefer the lighter isotope over the heavier one(s). This is known as **isotopic fractionation**.




Left, symbiosis – in our interaction representation we were seeking for high $^{15}$N (i.e. very selective) and low $^{13}$C fractionation. [doi:10.1016/j.fbr.2012.01.001]

Right, saprophytic behaviour – the fungi $\delta^{15}$N is expected to be roughly equal to soil's, while fungi $\delta^{13}$C is expected to be like host's one (most often stumps).

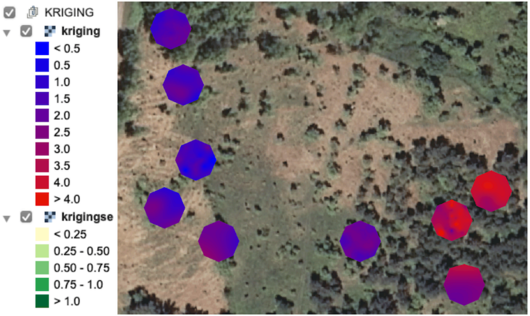
Fractionation is generally expressed as $\Delta$ (big delta) i.e. as a difference of relative concentrations $\delta$, in ‰ units.

# Case Study: Ordinary Geostatistics for Soil and Leaves

We evaluated ordinary (spatial only) isoscapes for **soil**, **fungi**, end each tree species **leaves**, to find the tentative mycorrhiza–host matching.

Ordinary Kriging was perfectly fine for soil and leaves, because soil has an extremely slow changing isotopic composition, while leaves'values are the result of an integrated chemistry all over the growing season.

We did not succeed in evaluating fungi isoscapes, in particular we repeatedly obtained an **inconsistent fungi N isoscape**, no matter the spatial interpolation algorithm used.
The reason is that the truffles grow in a few days, rapidly fractionating, so their $\delta^{15}$N grows over the collection time. **We had to consider an entangled space-time variability**.

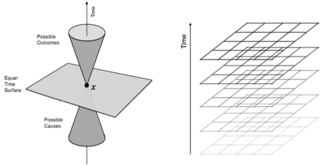



Soil N isoscape. The values represent $\delta^{15}$N ‰.

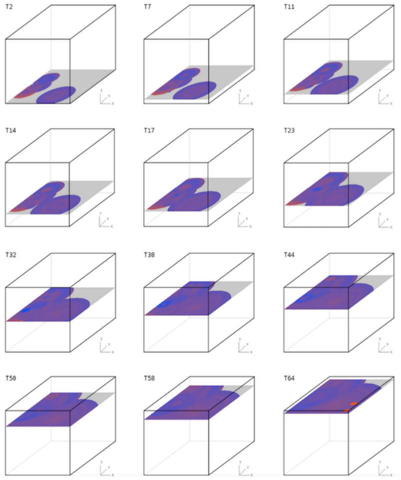
Needless to say, all the ordinary geostatistical interpolations have been performed with open tools (R, QGIS).

# Case Study: New Geostatistics for Mycorrhiza

From relativistic physics, we borrowed a **causal spatiotemporal structure** known as the Minkowski 4-dimensional space, defining a novel **spatiotemporal distance** to obtain a sheaf of spatial surfaces, indexed by time. We called such a structure a **TIMESCAPE**, i.e. a temporally changing landscape.

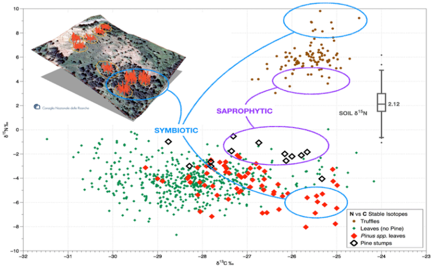



This new procedure had to be implemented from scratch, requiring the **development of specific software**: in fact, a new tool.




Mycorrhiza N timescape. Time increases vertically.

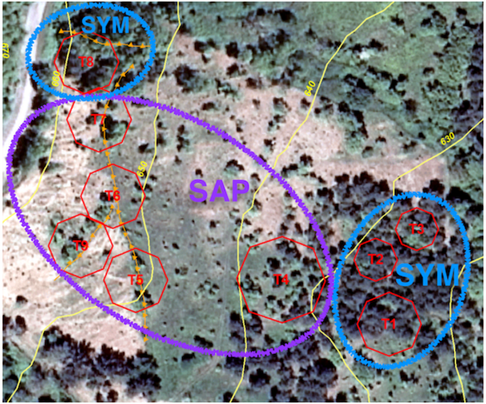
# Case Study: Finding the Match




Leaves, stumps, fungi and soil isotopic composition, represented in an abstract space ($\delta^{15}$N vs $\delta^{13}$C coord). **$\Delta^{13}$C fractionation** is measured horizontally, **$\Delta^{15}$N** fractionation is measured vertically.

Fungal nitrogen is on average 2‰ heavier than soil's.

Statistical matching techniques single out a **group** of possibly **symbiotic** and possibly **saprophytic** relationships between mycorrhiza and host trees.




The map resulting from statistical matching. We found **SYM** – two groups of living pines in symbiotic relation, **SAP** – a group of old pine stumps saprophytically consumed by mycorrhiza.

No clue of other host tree species was found.

# Case Study: a New Idea becomes a New Tool

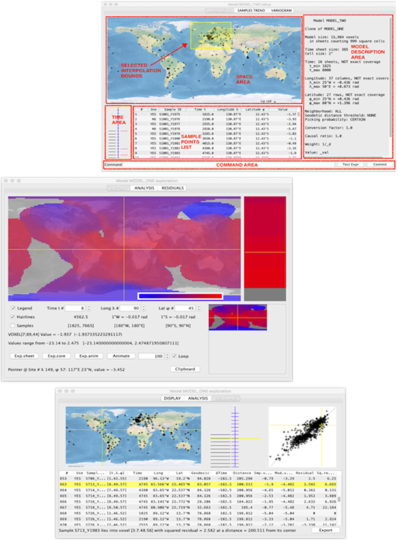
While working on the development of our **Timescape interpolation algorithm**, we realised that such procedure could have had a vastly larger base of potential users. In fact, any space- and time-related value (technically, any **random real scalar field**) could be modelled.

So we started developing a **new modeling tool**, nicknamed **Timescape**, following the open source licensing practices.

The software has been published under GNU/GPL3

- A **Python** version, as a standard module [doi:10.3390/earth3010017]
- A **Java** version, requiring a dedicated database for data storage [doi:10.30441/smart-elab.v10i0.201]

The original Timescape algorithm has been generalized in order to include **seasonality**. A few ancillary functions have been included as well (input/residuals analysis etc).

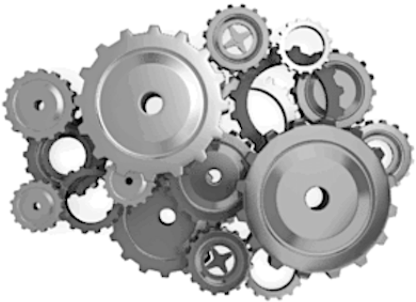



Input, output and validation windows from the Java version. [doi:10.30441/smart-elab.v11i0.202]

# Concluding Remarks – i.e. Modeling Cheat Sheet

A few random tips to keep your toolbox tidy:

- ✓ Use standard data formats: *csv*, *tiff*, *RData*, etc.
- ✓ Try to learn some (new) programming language.
- ✓ Keep data and procedures distinct.
- ✓ Don't be afraid of including other's (open) tools.
- ✓ Don't be afraid of publishing your tools.
- ✓ Check your algorithm: is it computable?
- ✓ Always use platform-independent tools.
- ✓ Design a clean ER data organization.
- ✓ Consider a RDBMS for complex datasets.
- ✓ Always estimate your data storage needs.
- ✓ Keep software regularly updated.



- ✗ Check carefully the software licenses.
- ✗ Do not stick to an operating system.
- ✗ Stay away from spreadsheets.

## Thank you for your kind attention


Sustainable and Innovative Approaches for Environmental Safety
Summer School – 11-13 July 2022 – Tbilisi, Georgia

# Spatial and spatio-temporal Environmental Modeling: Equipping your open source Toolbox

Marco Ciolfi – marco.ciolfi@cnr.it

Research Institute on Terrestrial Ecosystems of the National Research Council
12.1001 E 42.6818 N – Via Marconi 2, Porano 05010 TR, Italy