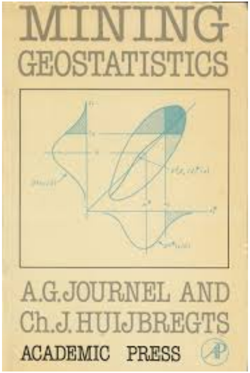
Scalable spatiotemporal geostatistics

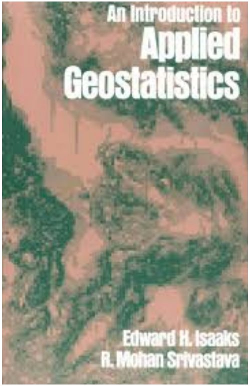
Edzer Pebesma
(with Marius Appel and Meng Lu,
and the sf and units dev teams)

ifgi
Institute for Geoinformatics
University of Münster

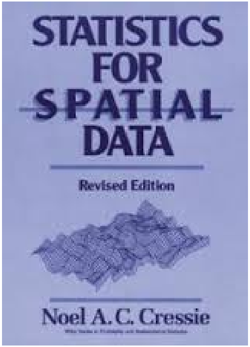
UIBK, Dec 15, 2016

What is geostatistics?





What is spatiotemporal geostatistics?

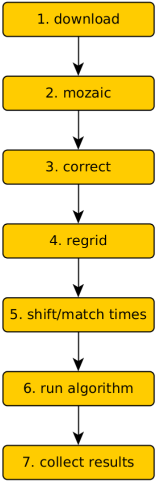


Statistics for SPATIO-TEMPORAL DATA
Noel Cressie · Christopher K. Wikle
WILEY

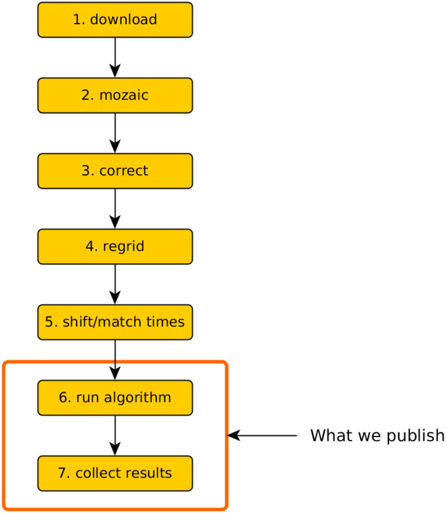
This talk ...

- is not about geostatistics in the narrow sense (interpolation and simulation using random field models)
- is not about statistical inference on large data sets (markov models, sparse matrices, covariance tapering, distributed linear algebra, INLA)
- is about geostatistics in the wider sense (“statistics for geographic or geoscientific data, where location plays an explicit role”)
- deals with analysing large datasets, in particular large EO arrays
- deals the practice of computing, and doing (open) science
- shows activities my group has been involved in over the last 6 months: SciDB4geo, measurement units for R, simple features for R.
- partly grew out of my frustration with the Earth observation research community

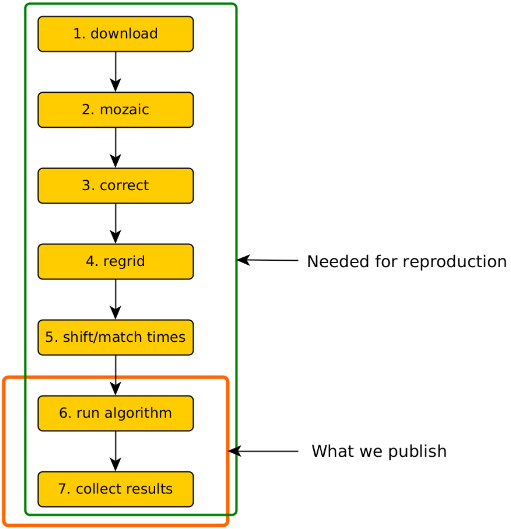
Current Earth Observation Research:



Current Earth Observation Research:



Current Earth Observation Research:



Arrays

(From: Lu, Appel, Pebesma, *Multidimensional arrays for analysing big geoscientific data*, in prep):
Arrays are a mapping from dimensions D to values V:

$$A : D \rightarrow V$$

where

- $D \subset D_1 \times D_2 \times ... \times D_n$ and
- $V \subset V_1 \times V_2 \times ... \times V_m$ and
- individual dimensions D_i are finite and totally ordered.

Dimensions are typically encoded by integers, but an invertible relation with dimension values always exists. Examples:

- space (1, 2, 3-dimensional affine, station or polygon IDs)
- time (linear, cyclic, multi-cyclic)
- wavelength (EM radiation, sound)
- any functional data

Note that

- Regular tables (and `data.frames`) are one-dimensional arrays
- R `arrays` are not arrays: they only have a single value.

Arrays

(From: Lu, Appel, Pebesma, *Multidimensional arrays for analysing big geoscientific data*, in prep):
Arrays are a mapping from dimensions D to values V:

$$A : D \rightarrow V$$

where

- $D \subset D_1 \times D_2 \times ... \times D_n$ and
- $V \subset V_1 \times V_2 \times ... \times V_m$ and
- individual dimensions D_i are finite and totally ordered.

Dimensions are typically encoded by integers, but an invertible relation with dimension values always exists. Examples:

- space (1, 2, 3-dimensional affine, station or polygon IDs)
- time (linear, cyclic, multi-cyclic)
- wavelength (EM radiation, sound)
- any functional data

Note that

- Regular tables (and `data.frames`) are one-dimensional arrays
- R `arrays` are not arrays: they only have a single value.

Arrays

(From: Lu, Appel, Pebesma, *Multidimensional arrays for analysing big geoscientific data*, in prep):

Arrays are a mapping from dimensions D to values V:

$$A : D \to V$$

where

- $D \subset D_1 \times D_2 \times ... \times D_n$ and
- $V \subset V_1 \times V_2 \times ... \times V_m$ and
- individual dimensions D_i are finite and totally ordered.

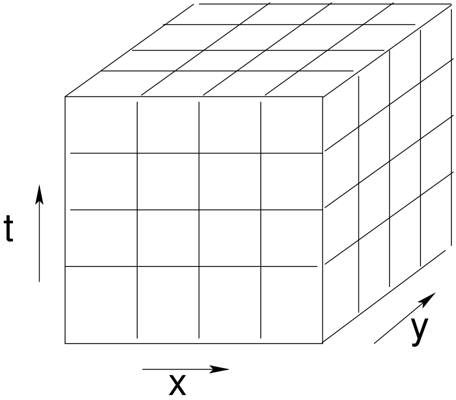
Dimensions are typically encoded by integers, but an invertible relation with dimension values always exists. Examples:

- space (1, 2, 3-dimensional affine, station or polygon IDs)
- time (linear, cyclic, multi-cyclic)
- wavelength (EM radiation, sound)
- any functional data

Note that

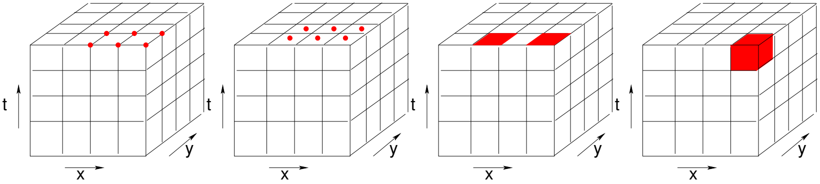
- Regular tables (and `data.frames`) are one-dimensional arrays
- R `arrays` are not arrays: they only have a single value.

```
      time   x    y red green blue
2012-04-06 4.5 48.3  22    41   51
2012-04-24 4.5 48.3  29    44   39
2012-05-09 4.5 48.3  33    29   46
2012-05-17 4.5 48.3  32    64   30
       ... ...  ... ...   ...  ...
```



Array cells, array cell values

- We refer to the array cell, or grid cell, or pixel, as the particular combination of dimension values D', and to the grid cell *value* as the corresponding data record V'
- The array model only specifies scalar dimension values. Whether array cell values V', refers to an area (space), time period (time) or wavelength interval is information that must be "somewhere else", including how this (area/period/interval) is related to the dimension (starting, lower-left corner, grid cell center).
- ISO 8601 has clear semantics on how text representations of date and datetime refer to time intervals, e.g. 2016−12 referring to the full month of December 2016, but e.g. R and most databases have no data types that match these semantics.


t
x
y
t
x
y
t
x
y
t
x
y

example from xts

```
> x = xts(1:3, Sys.Date()+1:3)

> x
           [,1]
2016-12-15    1
2016-12-16    2
2016-12-17    3

> x["2016-12-15 12:00:00"]
     [,1]

> x["2016-12-15"]
           [,1]
2016-12-15    1

> x["2016-12-15 12:00:00::2016-12-16 12:00:00"]
           [,1]
2016-12-16    2
```

SciDB

- SciDB is an array database that implements the most generic array, aiming at scientists: open source core (community edition), scalable, versioning of arrays, several missing value flag options, etc.
- used by data scientists, biostatistians, finance, astrophysicists, ...
- little uptake in Earth observation, so far

Features:

- scalable, shared-nothing architecture
- breaks up arrays in chunks, processed in parallel
- each dimension is represented by an Int64 (pm precision of the Earth)
- only cells with values are stored (sparse array)
- fast and flexible rearrange (swap dims with values and back).
- allows R scripts to run at worker nodes (cf. hadoop/stream)

⇒ useful for Earth observation data (x/y/t/bandw)?

Experiments with BFAST and the Verbesselt group in Wageningen

SciDB

- SciDB is an array database that implements the most generic array, aiming at scientists: open source core (community edition), scalable, versioning of arrays, several missing value flag options, etc.
- used by data scientists, biostatistians, finance, astrophysicists, ...
- little uptake in Earth observation, so far

Features:

- scalable, shared-nothing architecture
- breaks up arrays in chunks, processed in parallel
- each dimension is represented by an Int64 (pm precision of the Earth)
- only cells with values are stored (sparse array)
- fast and flexible rearrange (swap dims with values and back).
- allows R scripts to run at worker nodes (cf. hadoop/stream)

⇒ useful for Earth observation data (x/y/t/bandw)?

Experiments with BFAST and the Verbesselt group in Wageningen

SciDB4geo, SciDB4gdal

SciDB4Geo (Appel et al., 2016)

- is a SciDB plugin that adds operators to SciDB's query language
- sets spatial or temporal reference systems for array dimensions
- can match (join) two different arrays based on intersecting space/time cells
- uses the 2-D affine transformation for space, and GDAL's spatial reference systems

SciDB4GDAL (Appel et al., 2016)

- is a GDAL driver that lets you exchange SciDB4Geo-enabled arrays in SciDB with other file formats supported by GDAL (NetCDF, HDF4/5, GeoTIFF etc)
- needs, obviously, additional information on time when reading Earth observation data

Why GDAL? What is GDAL?

SciDB4geo, SciDB4gdal

SciDB4Geo (Appel et al., 2016)

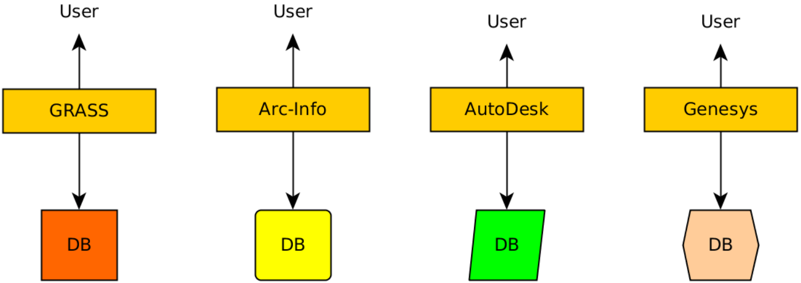
- is a SciDB plugin that adds operators to SciDB's query language
- sets spatial or temporal reference systems for array dimensions
- can match (join) two different arrays based on intersecting space/time cells
- uses the 2-D affine transformation for space, and GDAL's spatial reference systems

SciDB4GDAL (Appel et al., 2016)

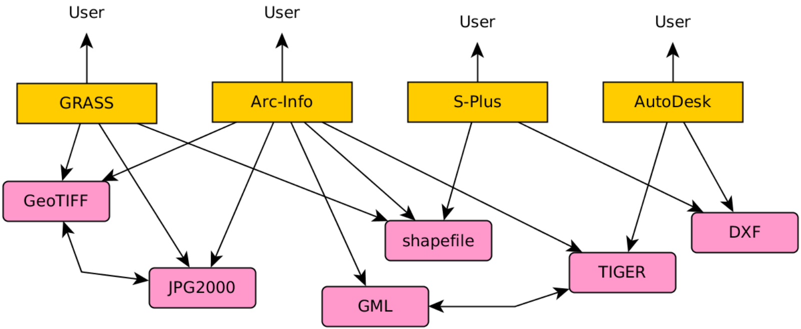
- is a GDAL driver that lets you exchange SciDB4Geo-enabled arrays in SciDB with other file formats supported by GDAL (NetCDF, HDF4/5, GeoTIFF etc)
- needs, obviously, additional information on time when reading Earth observation data

Why GDAL? What is GDAL?

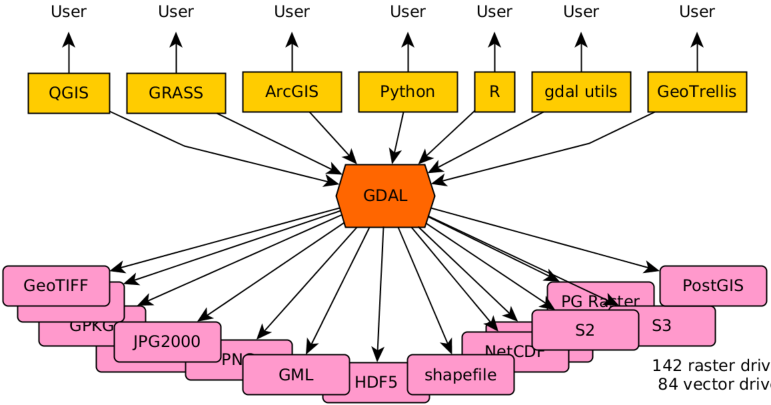
GIS: ’80s


User
User
User
User
GRASS
Arc-Info
AutoDesk
Genesys
DB
DB
DB
DB

File formats: ’90s



GDAL: ’00s



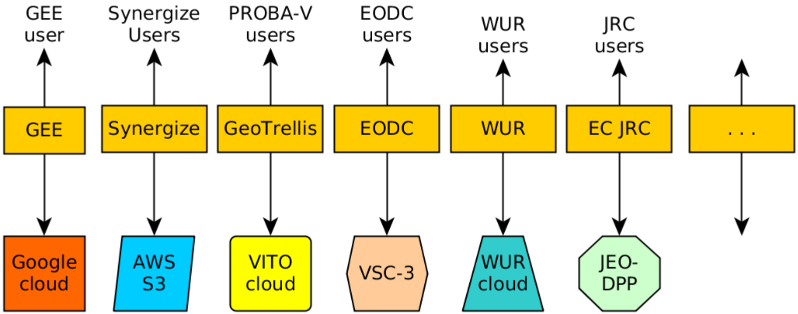
142 raster drivers
84 vector drivers

Looks like GDAL solved it all...?

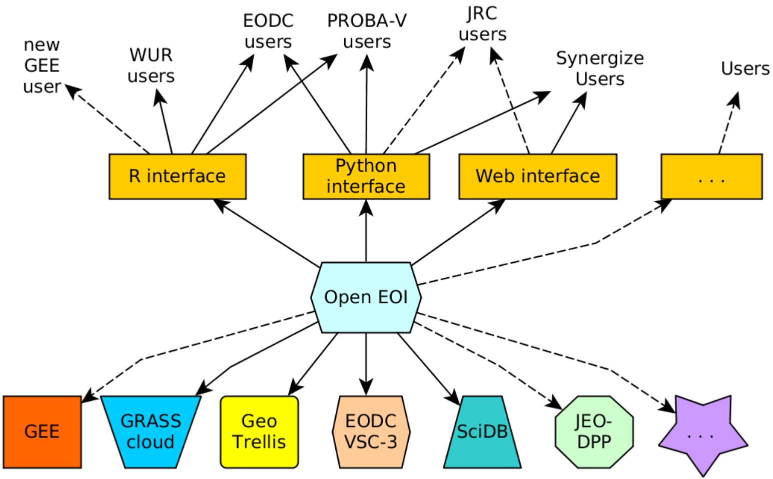
... not for analysing Today's Earth Observation data:

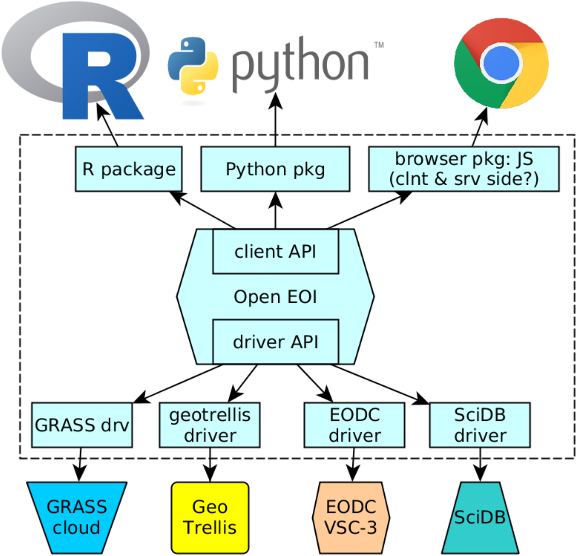
- it is file-based, and works best when files are local
- Earth Observation data nowadays are too large to download and process locally (Tb - Pb)
- In cloud-based solutions, the computations need to be done where the data are
- An interface to a cloud-based solution should provide all analysis operations, or better, accept R or python scripts to be run in a distributed fashion
- Google Earth Engine is the only platform that allows you to work easy and fast on Pb scale EO data, however it limits control over many analysis details, and is not open source

Big EO HPC: ’16


GEE user
Synergize Users
PROBA-V users
EODC users
WUR users
JRC users
GEE
Synergize
GeoTrellis
EODC
WUR
EC JRC
. . .
Google cloud
AWS S3
VITO cloud
VSC-3
WUR cloud
JEO-DPP

OpenEOI: ’20


new GEE user
WUR users
EODC users
PROBA-V users
JRC users
Synergize Users
Users
R interface
Python interface
Web interface
. . .
Open EOI
GEE
GRASS cloud
Geo Trellis
EODC VSC-3
SciDB
JEO-DPP
. . .


python™
R package
Python pkg
browser pkg: JS
(clnt & srv side?)
client API
Open EOI
driver API
GRASS drv
geotrellis
driver
EODC
driver
SciDB
driver
GRASS
cloud
Geo
Trellis
EODC
VSC-3
SciDB

units: Measurement units for R

```
> library(units)
> (a = 1:3 * ud_units$m)
Units: m
[1] 1 2 3
> (b = a + 5 * ud_units$km)
Units: m
[1] 5001 5002 5003
> (c = b / (5 * ud_units$h))
Units: m/h
[1] 1000.2 1000.4 1000.6
> c * 2.5 * ud_units$h
Units: m
[1] 2500.5 2501.0 2501.5
>
> apples = make_unit("apples")
> oranges = make_unit("oranges")
> 5:8 * apples / (c(3,2,5,8) * oranges)
Units: apples/oranges
[1] 1.666667 3.000000 1.400000 1.000000
> 5:8 * apples + (c(3,2,5,8) * oranges)
Error: cannot convert oranges into apples
```

but:

```
> z = zoo(1:3, Sys.Date()+1:3)
> m = ud_units$m
> class(z * m)
[1] "zoo"
Warning message:
Incompatible methods ("Ops.zoo", "Ops.units") for "*"
> class(m * z)
[1] "zoo"
Warning message:
Incompatible methods ("Ops.units", "Ops.zoo") for "*"
```

sf: Simple Features for R

- a standardized way to encode point(s), line(s) and polygon(s) geometries
- widely adopted, e.g. by spatial databases and GeoJSON
- implementation in R, compared to sp: cleaner, leaner, faster, and atomic
- sf geometries are atomic, and go into a list-column of a data.frame
- are considered tidy by Hadley
- integrate nicely with pipes and dplyr workflows
- will soon integrate with ggplot2

How will OpenEO look like, from R?

R `data.frames` will act as proxy to big EO arrays (on disk, or in the cloud):

- columns will contain a smaller set of records (cell values) as a "long table", with dimension values and attribute values;
- an attribute table will know all that is needed about this array (size, properties, dimensions, their reference system, measurement units of dimensions and attribute values)
- compute on this table for testing, defer computation to the cloud back-end for final
- also work proxying data on local disk (now package `raster`)
- also work for data in memory (no proxying: now `sp`)
- "tidy raster / tidy array"
- (solve access to big time series as along the way)

The bigger challenge...

The bigger challenges will be to settle on

- the minimum (and maximum) information EO back ands should provide
- the terminology for operators on EO image arrays

But, at least, we will be able to reproduce and compare complete workflows!

References

1. M. Appel, E. Pebesma, G. Câmara, 2015. **Scalable In-Database Regression Analysis of Large Earth-Observation Datasets.** EO Open Science 2.0 workshop at ESA-ESRIN, Frascati IT, Oct 12-16, 2015.
2. Marius Appel, Florian Lahn, Edzer Pebesma, Wouter Buytaert and Simon Moulds. **Scalable Earth-observation Analytics for Geoscientists: Spacetime Extensions to the Array Database SciDB.** EGU 2016, ESSI3.1; EGU2016-11780.
3. M. Appel, F. Lahn, and E. Pebesma, in prep. **Open and scalable analytics of large Earth observation datasets: from scenes to multidimensional arrays using SciDB and GDAL**
4. Meng Lu, Marius Appel, Edzer Pebesma, in prep. **Multidimensional arrays for analysing big geoscientific data.**
5. Edzer Pebesma, Thomas Mailund, James Hiebert, **Measurement units in R**. The R Journal, accepted.
6. Edzer Pebesma. **Simple Features for R.** `http://github.com/edzer/sfr`