(Thoughts about) FOSS Solutions for Geospatial Information Processing in Environmental Science and Engineering

Ari Jolma
Professor, Aalto University, Finland

Outline

- Environmental science and engineering
- Free and open source software stack
- Geoinformatica
- Web-based tools

Environmental Science and Engineering: Tasks, goals and topics

- Understanding
- Managing
- Developing solutions for
- Studying
- Planning
- Engineering

- Civil infrastructure
- Environmental change
- Sustainability
- Risks

Environmental Science and Engineering: Subjects of Information

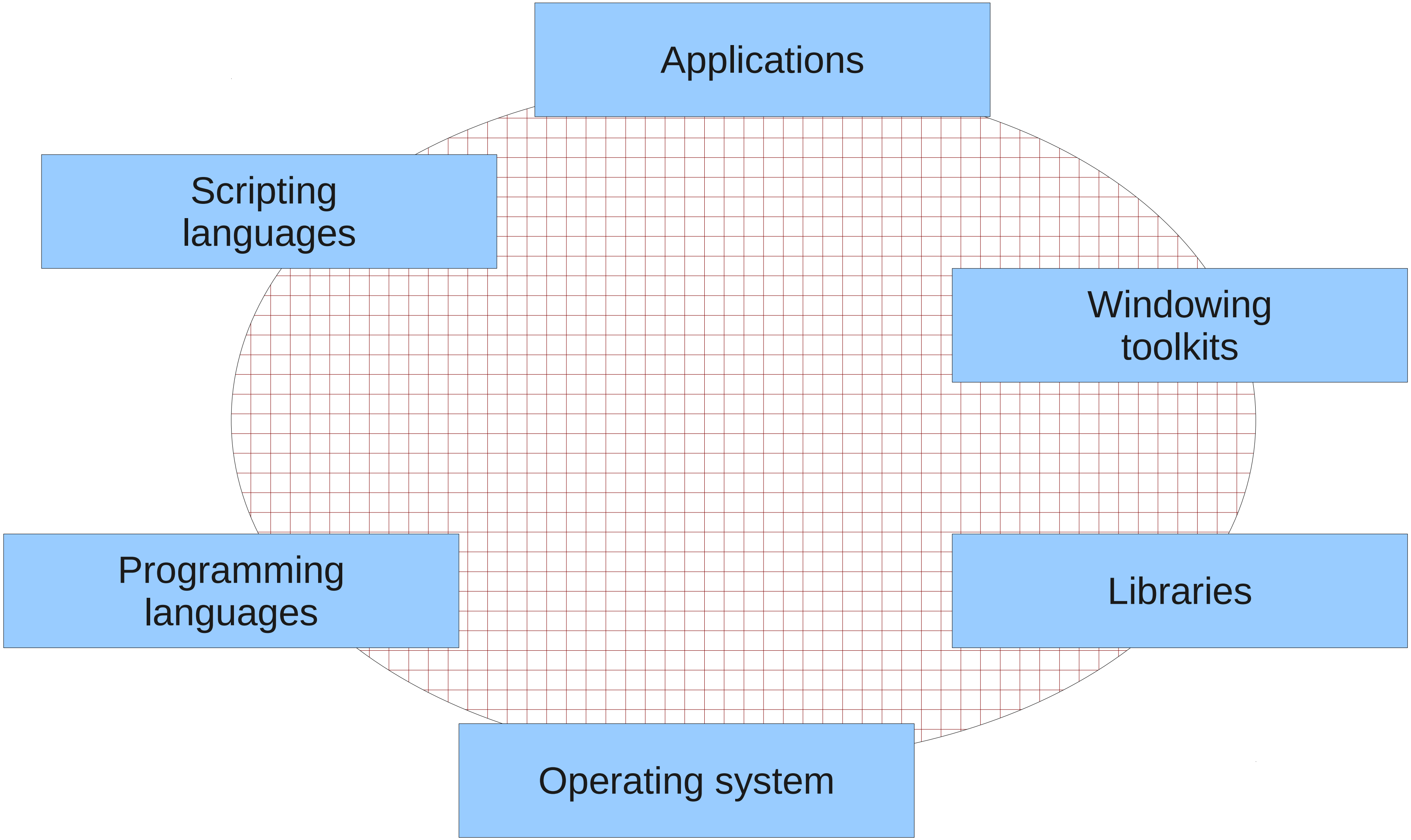
- Processes
  - For example: p in river systems, coastal p, catchment p
  - Physical and biological
  - Positive and negative
- Plans
  - Background materials: assessments, questionnaires, interviews
  - Ideas, drafts
- Engineering structures building and maintenance
  - Metadata, life-cycle
  - Various stake-holders and contractors

The four problems of information system development

- The presentation problem
  - Map
- The interaction problem
  - Display, mouse, keyboard
- The modeling problem
  - Data models and algorithms
- The development problem
  - Software tools

- Processes, plans and life-cycles
- Risks, sustainability criteria

The software stack



Operating system

- The main choices: Linux-based, Win32-based, Darwin-based
- Which distribution?
- For software development there are several choices, including Gnu for Windows
- Portability is useful: It is often easy to deploy to Win32 platform and users may prefer it
- Development (especially integration) is often much easier on Linux-based systems

Programming languages

- C or C++? A part of the modeling community prefers FORTRAN
  - Doesn't really matter if the result is a shared library
- I make an important distinction between programming languages and scripting languages (which are high-level PLs in fact!)
  - In my mind Java attempts to be both
- Some PLs are platform specific or make up their own platform (C#, Java)
  - This is always problematic

Libraries

- Should be easy to compile, focus on one task, have a clean and stable interface, and in general do what they are supposed to do and do it fast
- When within libraries: different data models are a hurdle, but not a show-stopper
  - Also, things may get complicated when you get closer to OS (memory, error handling, threads,...)
    - An occasional library developer should not have to deal with these – one solution is to develop within an existing library (GDAL for example) or use a utility library (GLib for example)
- Occasionally we need to use proprietary ones
- A general principle of mine is to always use libraries through a scripting language foreign function interface

libral

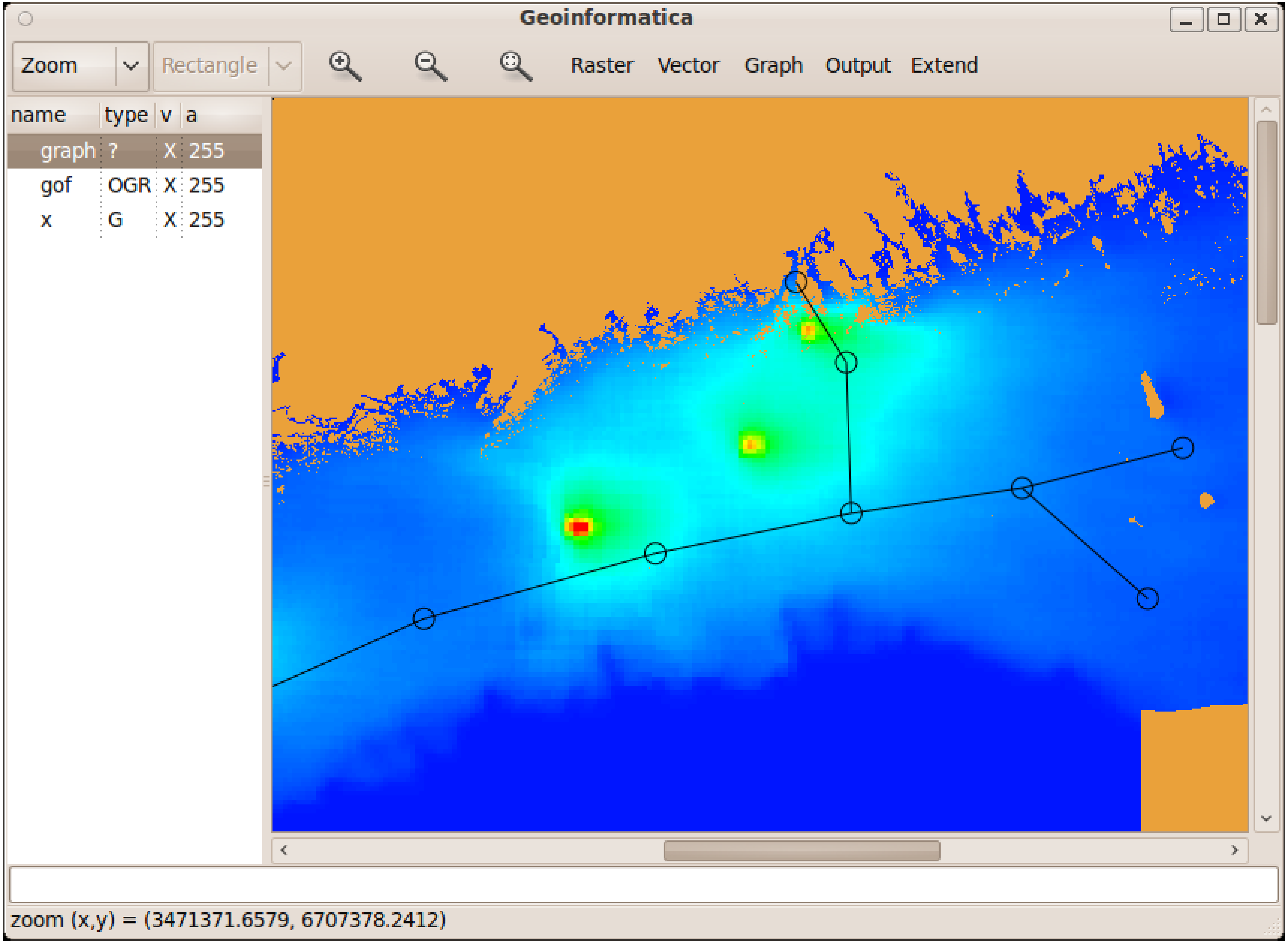
- A C library I've been developing (and using as a research/learning tool...) for raster algebra
  - Simple in-memory rasters → really fast algorithms
  - A back-end for Perl extension for raster algebra
  - Code for rendering rasters and vector data (coming from OGR) on GDK-pixbufs and/or Cairo surfaces has crept in
  - Interoperates with GDAL rasters (very simple to convert a libral raster to an OGR memory raster)
    - Perl rasters can polymorphically be libral or GDAL rasters → interesting possibilities for raster algebra
  - Interoperates (well, one way currently) with PDL (Perl Data Language) → easy to bring in data supported by PDL

Scripting languages

- Surprisingly many: Perl, Python, Ruby are well-known all-purpose ones, but there are several more specialized ones: R, The-Matlab-like-one-Octave-uses, Postscript, ... and even more more specialized ones: SQL, Glade, The-MILP-language-I-whipped-up, …
  - The concept of little languages or minilanguages is well-known
- Benefits: division of labor, domain-specificness, fewer lines of code
- Problems: complexities of mixtures, debugging, challenges to intellect

Windowing toolkits

- Needed when you are required to deliver an application with a new graphical user interface
- I use GTK+, which is a part of GNOME, the alternative is Qt (used by Quantum GIS for example)
  - GNOME is the default in Linux-based distributions that I use
  - gtk-perl is alive and well


Geoinformatica
Zoom
Rectangle
Raster
Vector
Graph
Output
Extend
name type v a
graph ? X 255
gof OGR X 255
x G X 255
zoom (x,y) = (3471371.6579, 6707378.2412)

Geoinformatica

- A stack of GDAL, libral, Perl, GTK+, gtk-perl
- Statistics of the Perl part (not counting external Perl modules and GDAL Perl):
  - ~19 500 lines of code, of which
  - ~3 900 lines is comments
  - 800 subroutines (= average 20 lines per sub)
  - 19 dialogs (stored in Glade XML files)
  - 40 source code files
  - 5 major classes, 61 in all
  - Start-up time ~3 seconds (the first time, 2nd time is faster)
  - The main program is 250 lines, which sets up a vanilla application

Applications

- A program, which interacts with the user, who wishes to accomplish a task
  - Input-output program
    - A small program written in a scripting language, often by the user or a more generic program that is controlled by switches
    - Task is well-structured
  - Graphical program
    - Task-oriented, with a as-simple-as-possible GUI, or a generic one, with as large set of functionalities as possible
    - Task often not so well-structured, typically providing decision support or a platform for explorative research
    - May also be used for structured tasks

Geoinformatica as a research platform

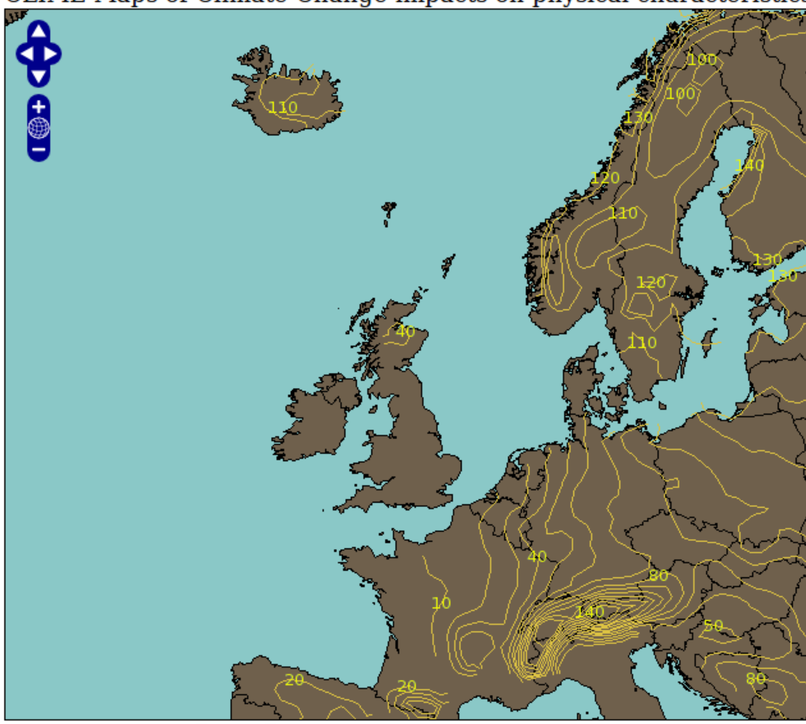
- An optimized set of tools (for myself), with very good basic functionality provided by packages (which are free)
- It is possible to deliver good solutions implemented as good graphical applications
  - Examples: soil database management tool, oil spill risk assessment tool
- Really interesting simulation modeling applications still to be developed
- Looking forward to study planning with complex features and geospatial design (civil engineering, landscape design)

The web

- Remote servers will be increasingly used as data sources and data processing services
- Typically data will enter the system through data access libraries. For example GDAL can already be compiled to access WMS and WCS.
- Scripting languages are powerful tools to implement those services. For example I've implemented a simple WMS server for research purposes with Perl equipped with appropriate modules (there is no WMS module as such) with < 300 lines of code.

Deploying information about environmental change using the Web

CLIME Maps of Climate Change impacts on physical characteristics of lakes



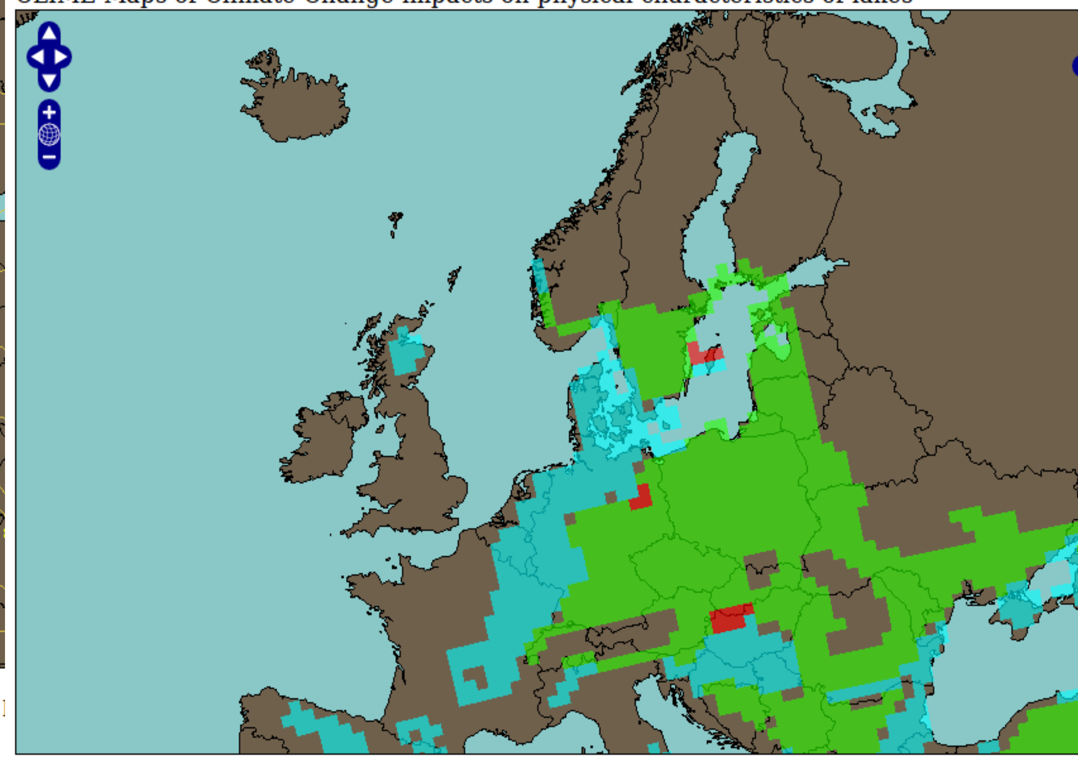
IPCC Scenario: A2 B2
Lake type: small and shallow small and deep large and shallow

- Regions whose lakes may start experiencing ice-free winters more...
- Shortening of the duration of ice cover [days] more...
- Becoming earlier of the timing of ice-off [days] more...
- Increase in summer temperature of lake surface water [degrees C] more...

The climate variable map page depicts the input data used for these projection

Regional climate model data courtesy of the Prudence project.

CLIME Maps of Climate Change impacts on physical characteristics of lakes



IPCC Scenario: A2 B2
Lake type: small and shallow small and deep large and shallow large and deep more...

- Regions whose lakes may start experiencing ice-free winters more...
- Shortening of the duration of ice cover [days] more...
- Becoming earlier of the timing of ice-off [days] more...
- Increase in summer temperature of lake surface water [degrees C] more...

The climate variable map page depicts the input data used for these projections.

Regional climate model data courtesy of the Prudence project.

More interaction ...

CLIME lake analyst for the selected lake/location

Location is [53° 33′ 12″ N, 14° 44′ 36″ W]
The model is SMHI HC (grid is SMHI)
closest CLIME lake is 'Müggelsee'
closest GLRIP lake is 'Nehmitzsee' (ice duration data from 29 years)
closest lake with a Wikipedia entry that has been added to the database is [Dąbie Lake]

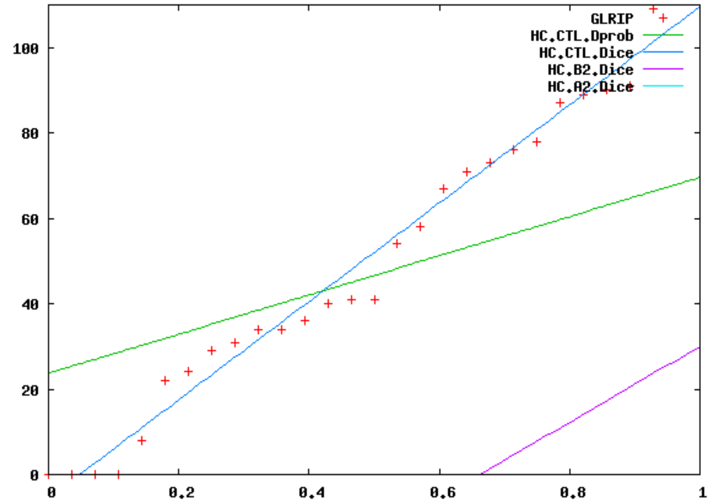


Figure The number of below-zero days and duration of ice cover vs. its cumulative probability. The unit of x-axis is probability and that of the y-axis is days. The climate data i from cell (41,51) of the grid.

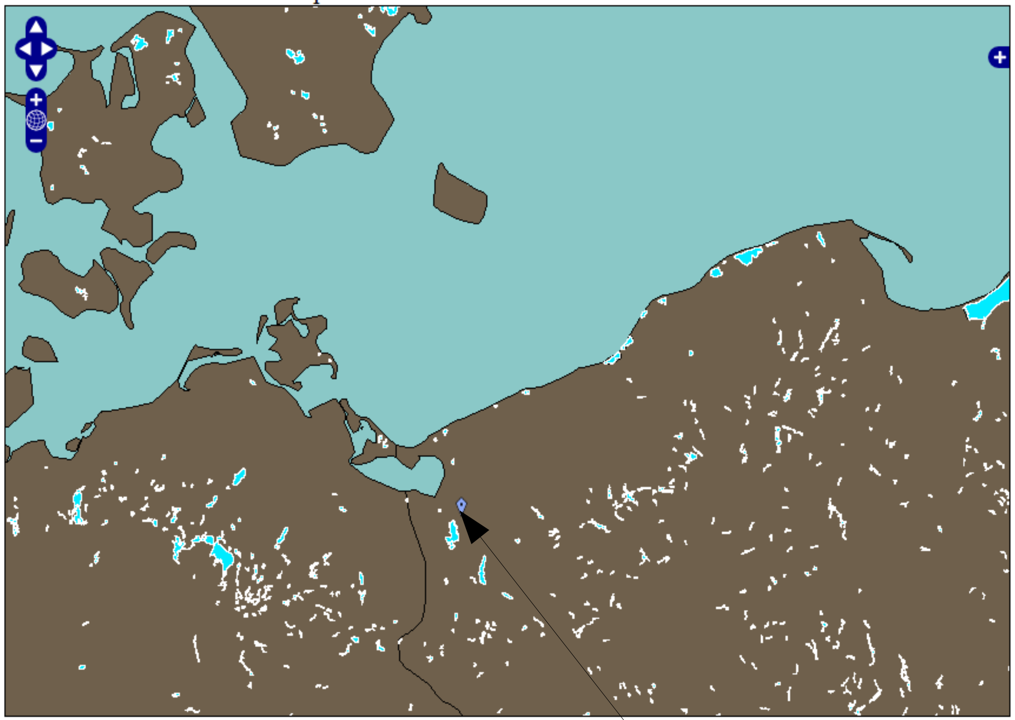
c = D^-_{prob}/D_{ice} = 2.50840014634213

df = how many days to have ice cover = 25.9039369835667

change c and df

use c and df from closest GLRIP lake

CLIME Climate Variable Maps



Model	Scenario	Variable
○ HadRM	○ A2	○ D^-_{prob}
○ SMHI MPI	○ B2	○ theta
◉ SMHI HC	◉ Control	○ T_summer

[Back to the impacts map page.]

A single click on the map opens the ice cover duration analysis page.

Regional climate model data courtesy of the [Prudence project].

Selected location

The software stack used for the climate change on lake ice study

Presentation
Interaction
Modeling
Development

- OpenLayers viewer
- WMS server (DIY)
- Analytical tools (DIY, Gnuplot)
- Data management (PostGIS)
- Geospatial computations (GEOS, GDAL)
- Spatial extrapolation of the geophysical model (GDAL+Perl)
- The geophysical model (Octave)

Thank You!

ari.jolma@tkk.fi