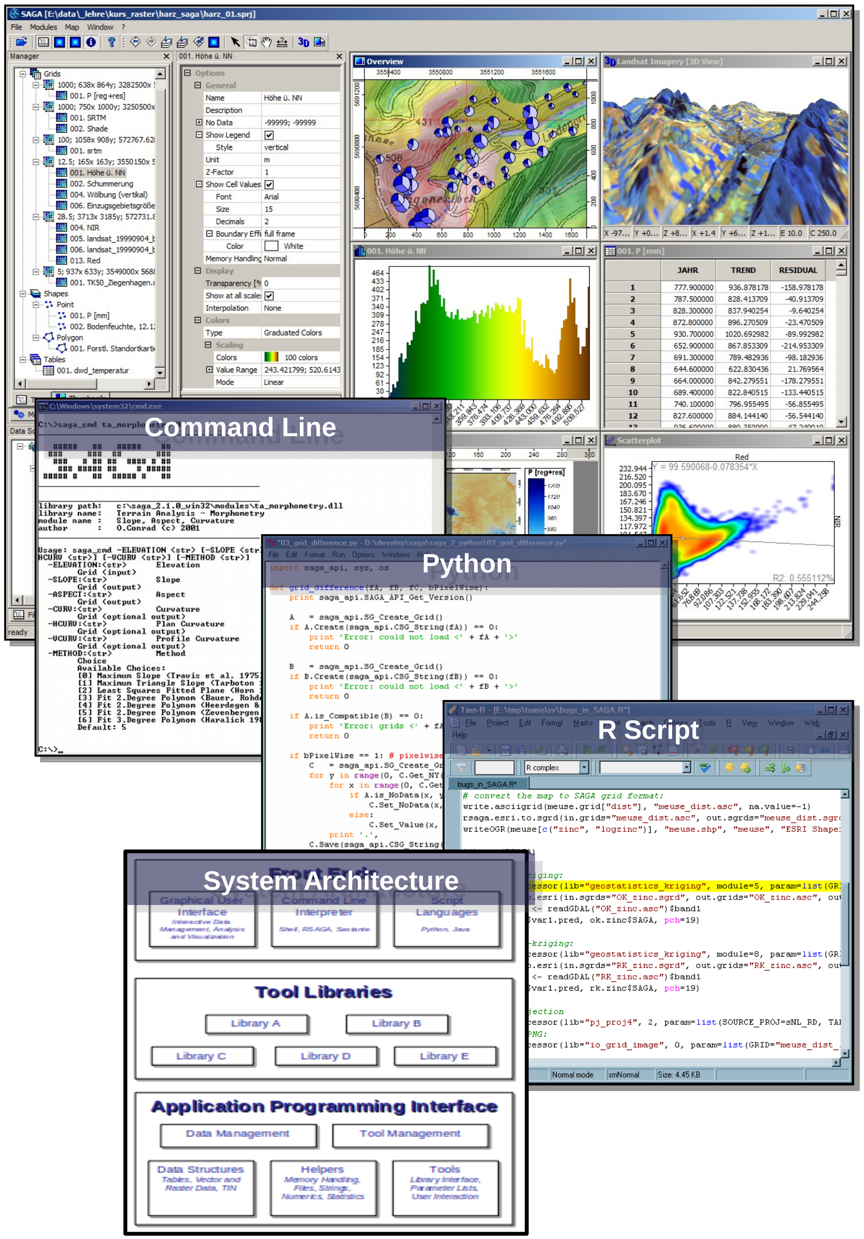



# SAGA

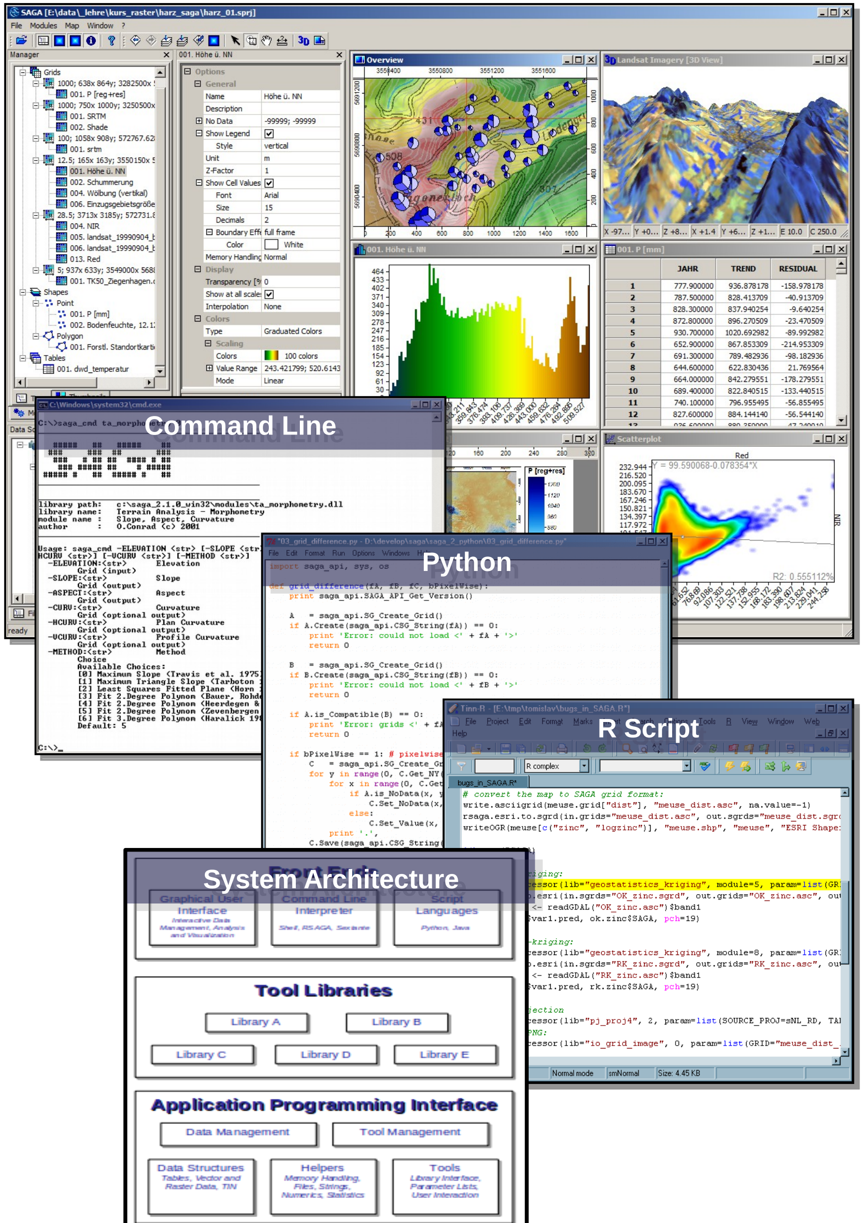
## System for Automated Geoscientific Analyses

# Automating your analysis using SAGA GIS

*Johan Van de Wauw*

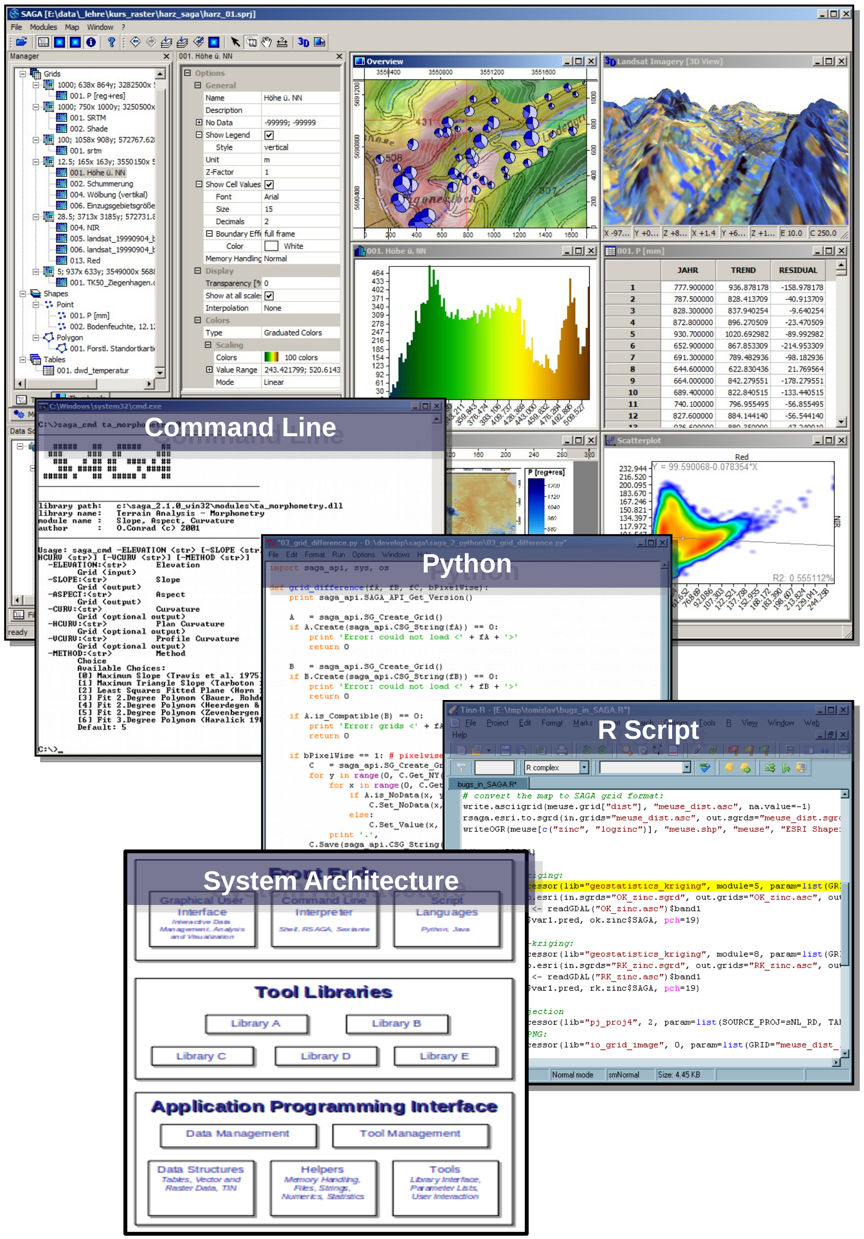
*Slides by:*
*Olaf Conrad, Michael Bock, Volker Wichmann*

# SAGA | System for Automated Geoscientific Analyses




- SAGA is a Geographic Information System **(GIS)** software with strong capabilities for geodata processing and analysis.
- SAGA is programmed in the object oriented C++ language and supports the implementation of new functions with a very efficient Application Programming Interface **(API)**.
- Tools are provided by framework independent Tool Libraries and can be accessed most simply via SAGA's Graphical User Interface **(GUI)** or various scripting environments..

# Key Features




- Object oriented system design
- Modular structure with framework independent tool development
- API with strong support for geodata handling
- GUI for intuitive data management, analysis and visualization
- Far more than 650 free tools
- Runs on Linux as well as on Windows operating systems
- Portable software runs without installation even from USB sticks
- Free and Open Source Software
- 10 years of continuous development

# Drivers of Development

- SAGA's development is mainly driven by the research interests of its inventors and developers
  - Physical Geography, Hamburg University
  - SciLands GmbH, Göttingen
  - Laserdata GmbH, Innsbruck
- SAGA's publication as FOSS lead to several external inputs enriching the spectrum of developments
  - **V. Olaya:** SEXTANTE (Sistema Extremeno de Analisis Territorial), SAGA manual, module & system development
  - **T. Schorr**: GEOSTEP project, Linux, Unicode and 64bit compatibility, SAGA-Python interface
  - **V. Wichmann**: PhD Thesis (rockfall modelling), Laserdata GmbH, support & documentation, module & system development
  - **V. Cimmery**: SAGA 2 User Guide, documentation
  - **A. Brenning**: RSAGA plugin, accessing SAGA modules from R environment
  - **J. Van de Wauw:** Linux support and distribution (Debian/Ubuntu), bug fixes, module development
  - **J. Brunke:** Power User, Environment Agency, County Gifhorn
  - and many other contributions, mainly modules and documentation

# Drivers of Development | Selected Projects,Physical Geography, Hamburg

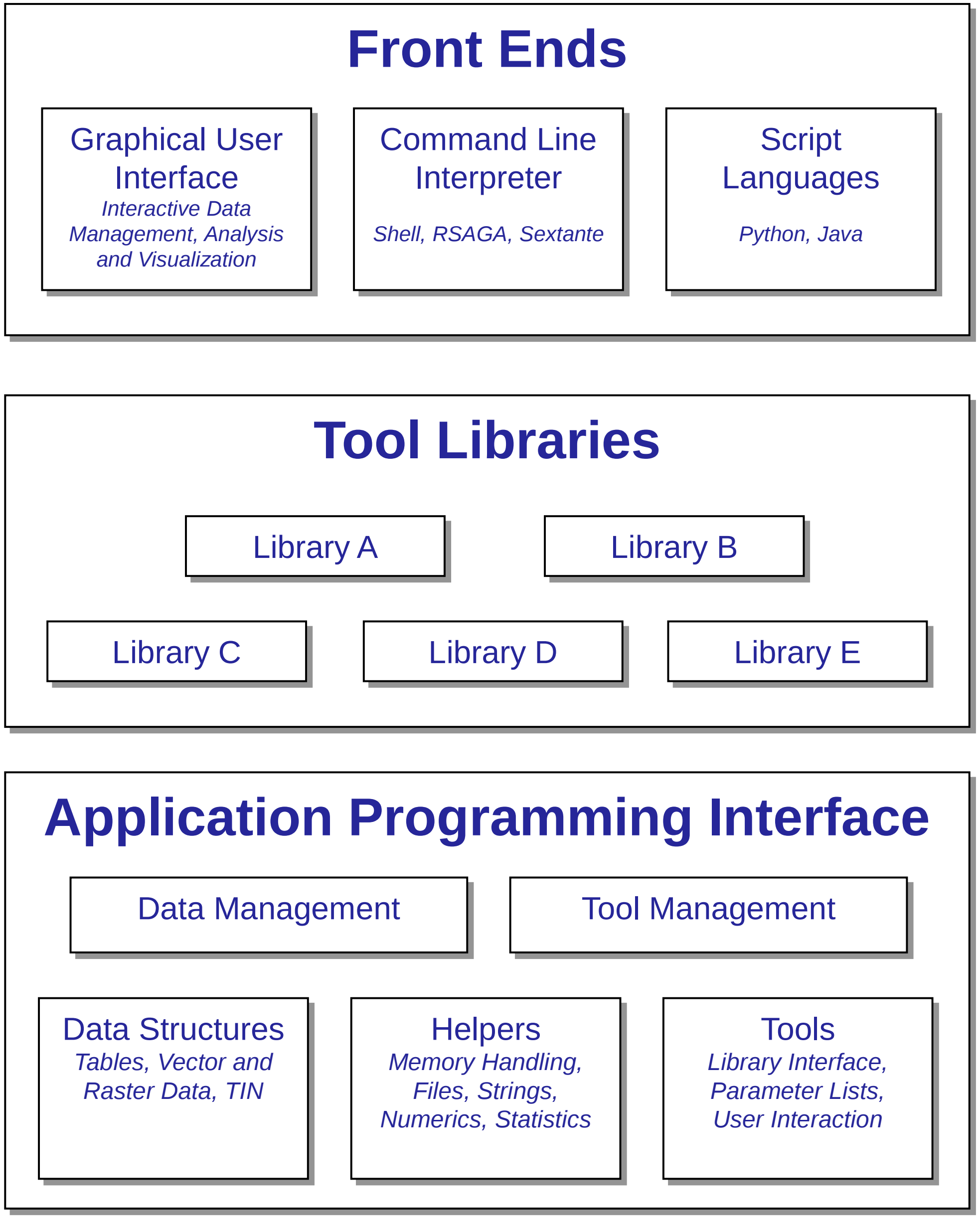
- CARBIOCIAL – Carbon sequestration, biodiversity and social structures in Southern Amazonia: models and implementation of carbon-optimized land management strategies.
- The Future Okavango – Scientific support for sustainable land and resource management in the Okavango basin – GIS-based landscape analyses, environmental modelling, and decision support for integrated resource management.
- CHELSA - Climatologies at High Resolution for the Earth's Land Surface Areas. A research cooperation of Institute of Systematic Botany, University Zürich, Biodiversity, Macroecology & Conservation Biogeography Group, University Göttingen and Physical Geography, University Hamburg.

- SALEM - Development of a Model for the Spatial Prediction of Periglacial Deposits – Funded by the German Federal Institute for Geosciences and Natural Resources (BGR)

- SAGA-REKLIM – Climate Change and Forestry: Researches and developments for a SAGA based problem oriented regionalisation of spatially distributed climate data for Baden-Württemberg".
- Spatial high resolution regionalization of urban climates, integrating statistical-dynamical downscaling and remote sensing techniques. Integrated Climate System Analysis and Prediction (CliSAP).
- Apart from any project: **GIS Training and Education**

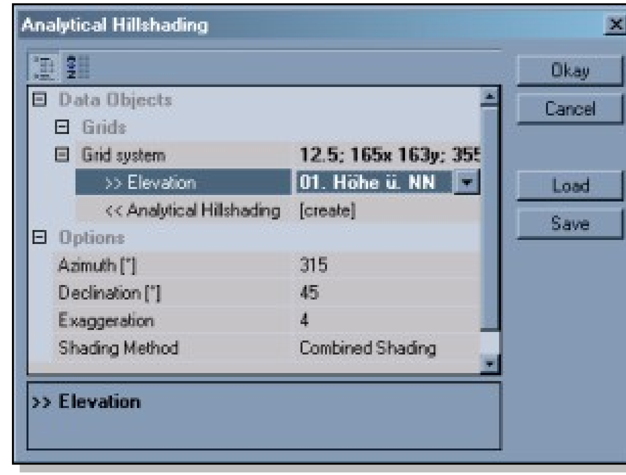
# System Architecture

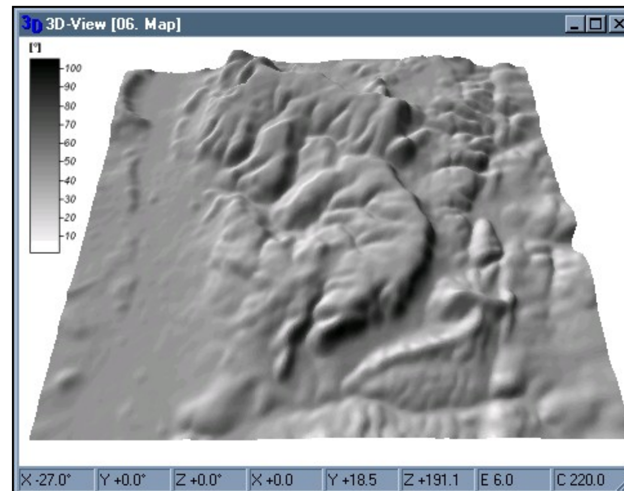
- SAGA's architecture is modular.
- Its base is the **Application Programming Interface** (API), which provides data object models, basic definitions for the programming of scientific modules and numerous helpful classes and functions.
- **Tool libraries** are Dynamic Link Libraries (DLL), or Shared Objects in Linux context, and provide the scientific methods. To access and run the tools you need a front end program.
- A **Graphical User Interface** (**GUI**) and a **Command Line Interpreter** (**CLI**) are the two generic SAGA **front ends**. Alternatively SAGA tools can be used with **scripting**.



# Tool Programming

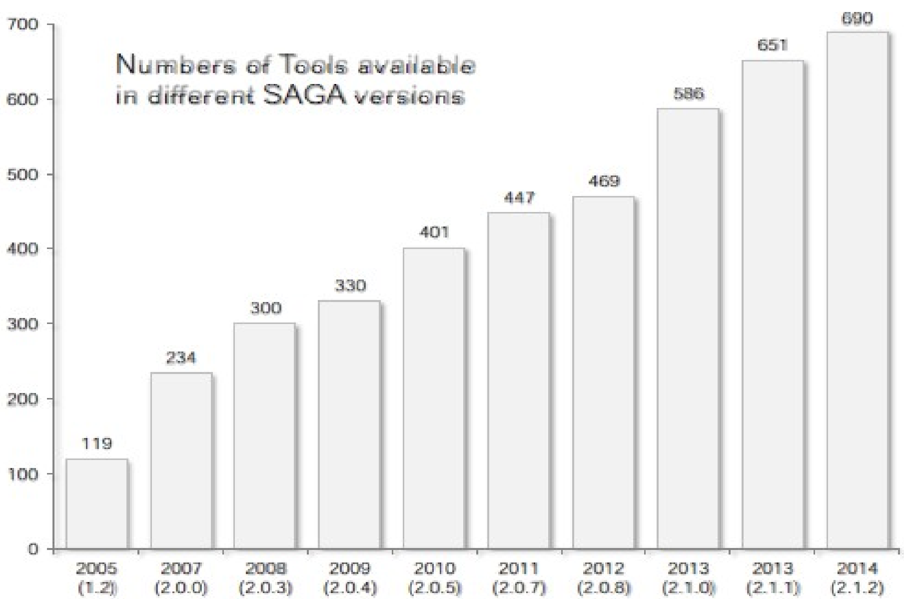
```
CHillshade::CHillshade(void)
{
    Parameters.Add_Grid(
        NULL, "ELEVATION", "Elevation", PARAMETER_INPUT);

    Parameters.Add_Grid(
        NULL, "SHADE"    , "Shade"    , PARAMETER_OUTPUT);
...
}


bool CHillshade::On_Execute(void)
{
    CSG_Grid *pDEM    = Parameters(“ELEVATION“)->asGrid();
    CSG_Grid *pShade  = Parameters(“SHADE“)    ->asGrid();
...
    for(y=0; y<Get_NY(); y++)
    {
        for(x=0; x<Get_NX(); x++)
        {
            if( pDEM->Get_Gradient(x, y, s, a) == false )
            {
                pShade->Set_NoData(x, y);
            }
            else
            {
                d     = acos(sin(s) * sin(Dec) + cos(s) * cos(Dec) * cos(a - Azi));

                pShade->Set_Value(x, y, d);
            }
        }
    }
}
```



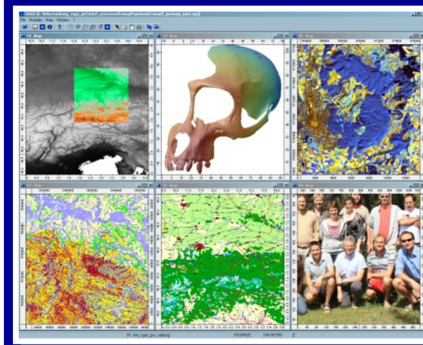



# The SAGA Toolset

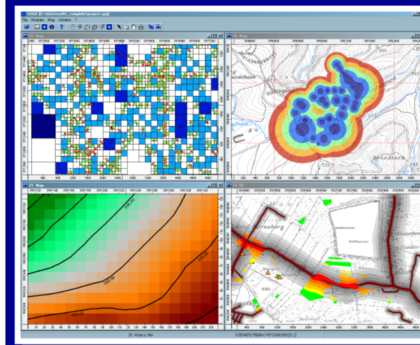
- SAGA offers a comprehensive and growing set of free tools.
  - Data Import & Export
  - Cartographic Projections
  - Numerous Raster & Vector Data Tools
  - Image Processing
  - Terrain Analysis
  - Spatial & Geostatistics
  - and many more…



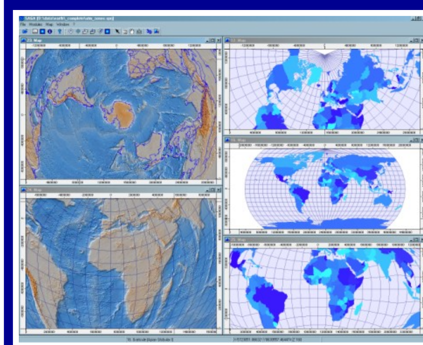

**Data Import & Export**



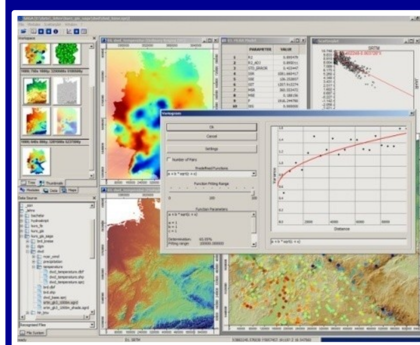
**Vector Tools**



**Projections**



**Terrain Analysis**



**Raster Tools**

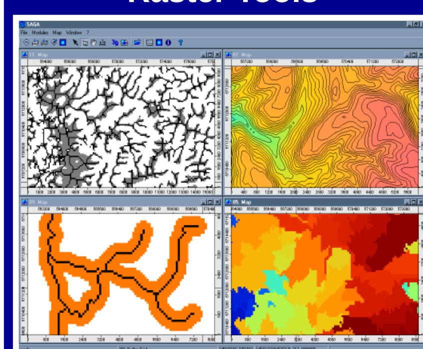
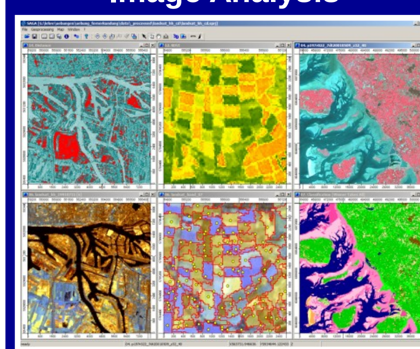
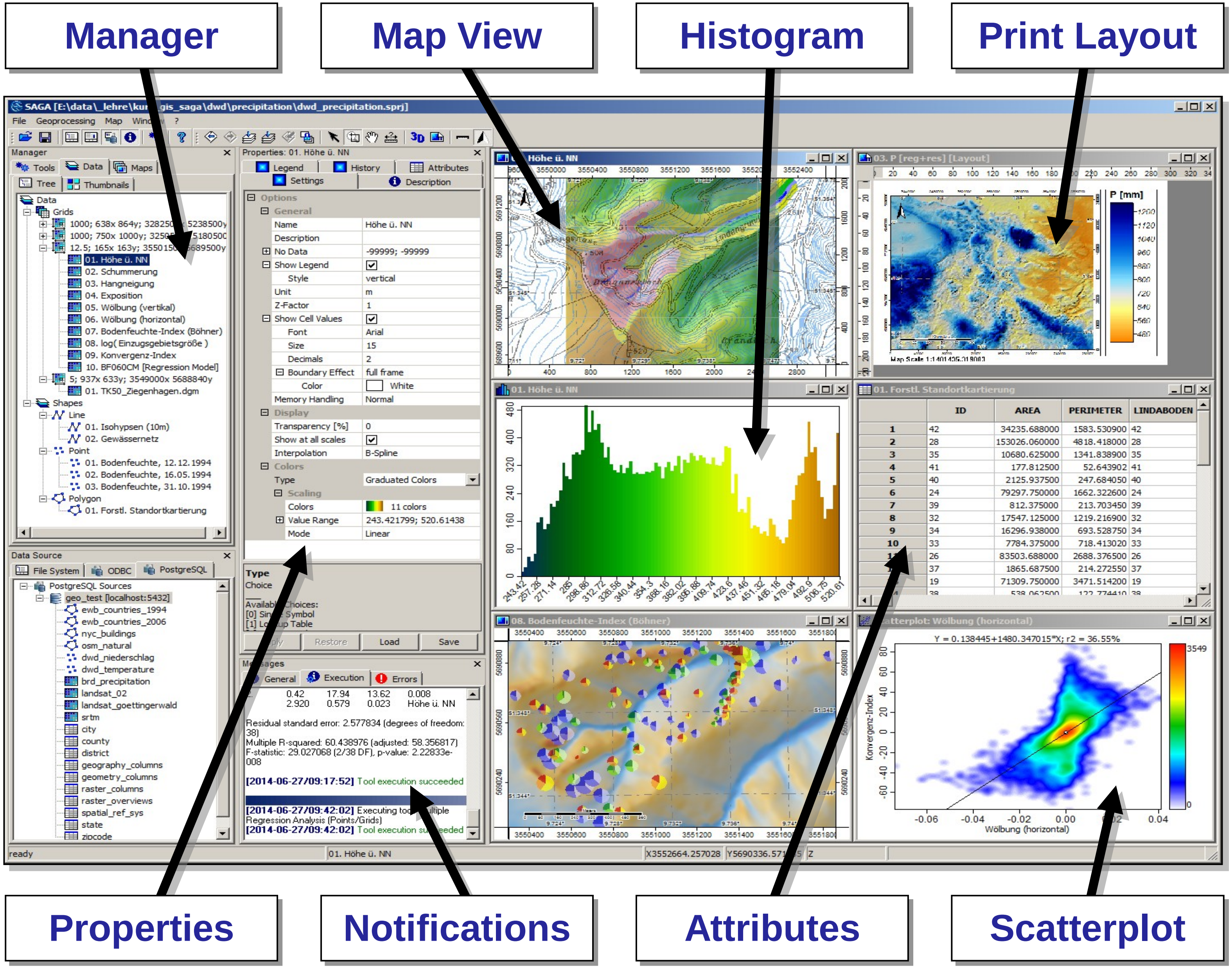


**Image Analysis**

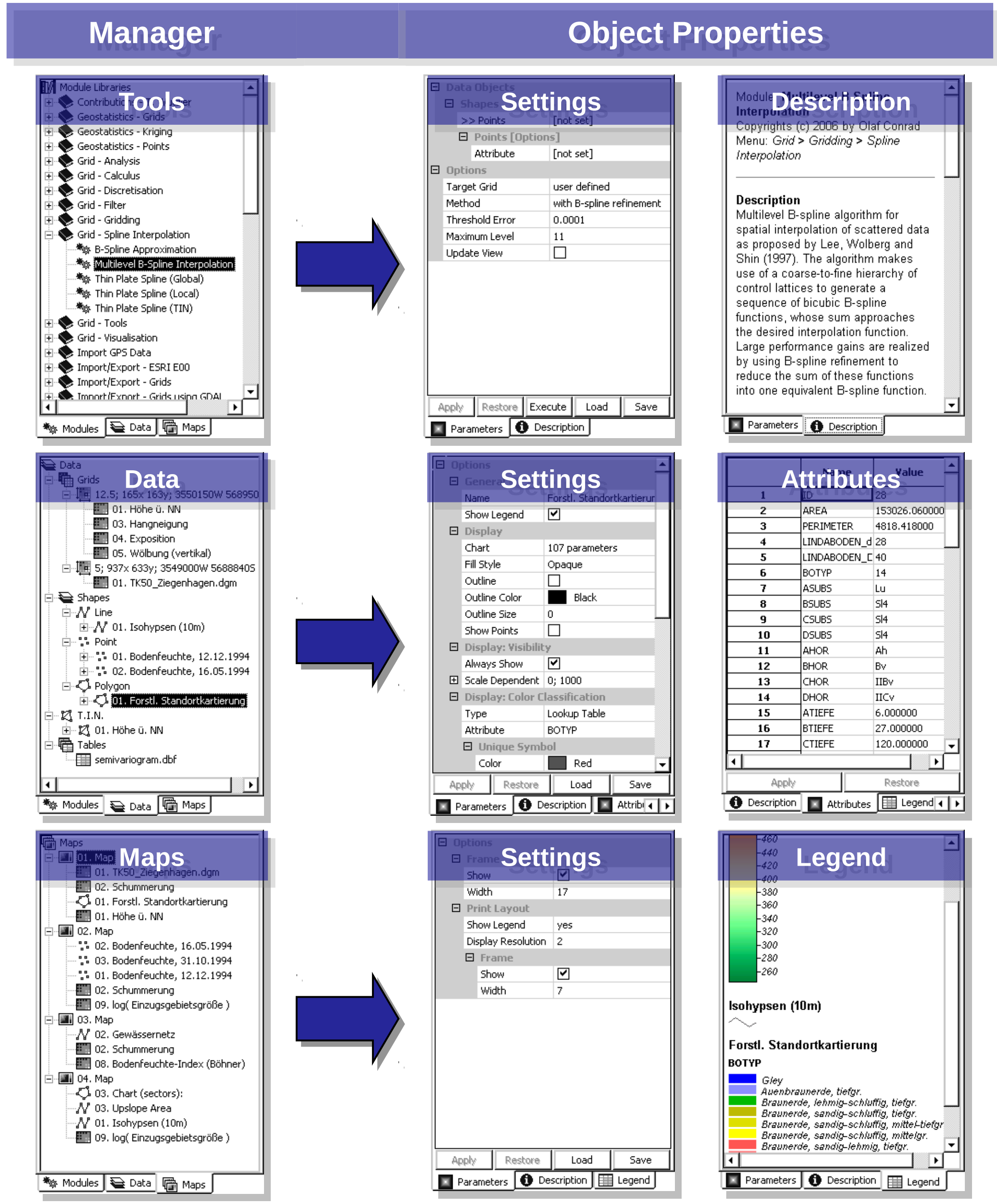


# Front Ends | Graphical User Interface



# Front Ends | Graphical User Interface

- Three Manager Controls
  - Modules, Data, Maps
- Properties depend on the object type selected in the manager control.
  - A settings and a description tab are common to all items.
  - In case of a tool, the settings show the tool's execution parameters. The description gives further information about the tool.
  - In case of a data set, the settings allow to change data set name, memory handling, symbology and other data type specific options. Besides a description a legend and a data set history is added.



# Front Ends | Command Line Interpreter

- The SAGA Command Line Interpreter (CLI) makes it possible to execute SAGA tools from a command line or shell console.
- This is not very user friendly, but allows to write batch/shell scripts for further automation of process flows.

C:\Windows\system32\cmd.exe

```
C:\>saga_cmd ta_morphometry 0
_____________________________________________

   #####   ##   #####    ##
  ###     ###  ##       ###
   ###   # ## ##  #### # ##
    ### ##### ##     # #####
 ##### #   ##  ##### #   ##
_____________________________________________

library path:   c:\saga_2.1.0_win32\modules\ta_morphometry.dll
library name:   Terrain Analysis - Morphometry
module name :   Slope, Aspect, Curvature
author      :   O.Conrad (c) 2001
_____________________________________________

Usage: saga_cmd -ELEVATION <str> [-SLOPE <str>] [-ASPECT <str>] [-CURV <str>] [-
HCURV <str>] [-VCURV <str>] [-METHOD <str>]
  -ELEVATION:<str>       Elevation
        Grid (input)
  -SLOPE:<str>           Slope
        Grid (output)
  -ASPECT:<str>          Aspect
        Grid (output)
  -CURV:<str>            Curvature
        Grid (optional output)
  -HCURV:<str>           Plan Curvature
        Grid (optional output)
  -VCURV:<str>           Profile Curvature
        Grid (optional output)
  -METHOD:<str>          Method
        Choice
        Available Choices:
...
rgess 1996)
```

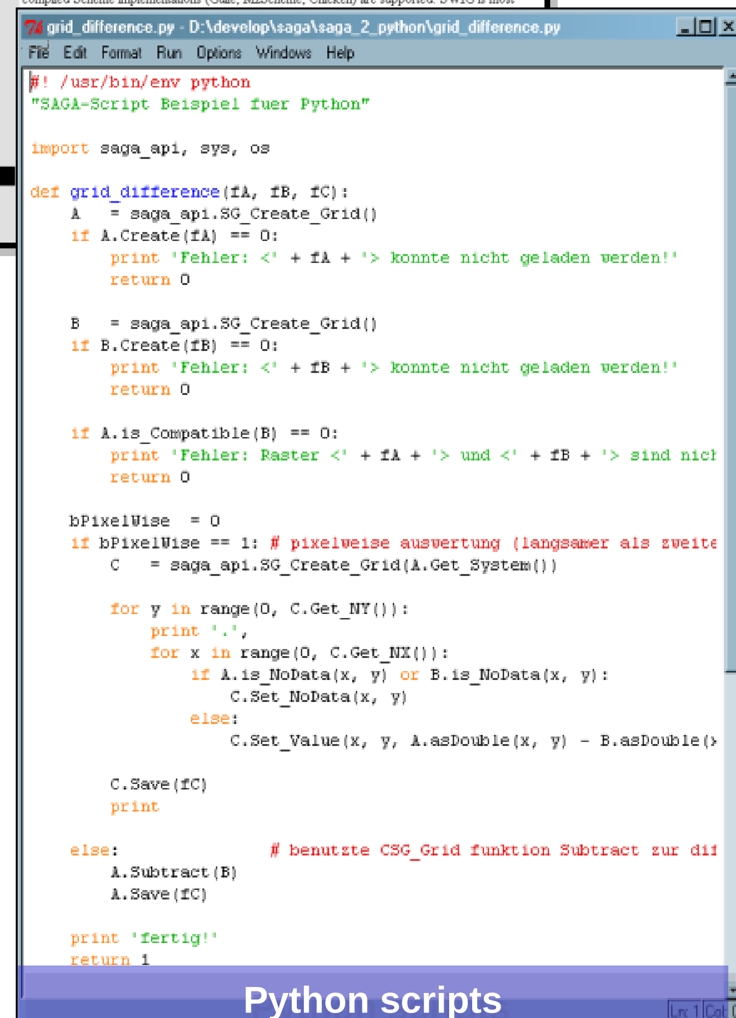
```
SET FLAGS=-f=q
REM SET SAGA=.

IF EXIST srtm.tif GOTO :SRTM
ECHO create a Gaussian landscape
saga_cmd %FLAGS% recreations_fractals 5 -GRID=dem.sgrd -NX=400 -NY=400 -H=0.75
:SRTM
ECHO import and project srtm (geotiff)
saga_cmd %FL...
saga_cmd %FL...

ECHO do some...
saga_cmd %FL...
saga_cmd %FL...
saga_cmd %FL...
saga_cmd %FL...

ECHO run saga...
SET INPUT=de...
saga_cmd %FL...
```

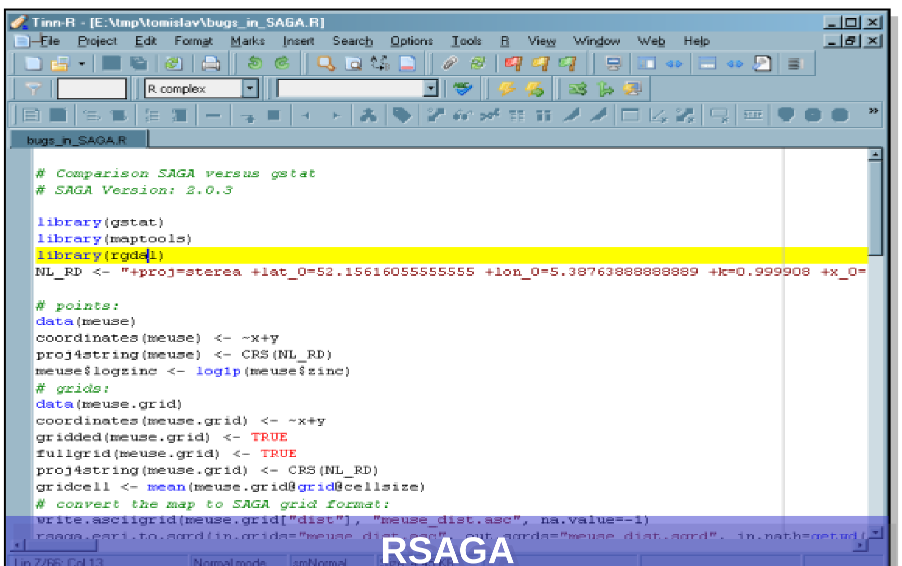
```
REM 'REM' or '#' can be used for comments, 'ECHO' for message output.
REM environment variables can be accessed using the ms-dos/window style

ECHO _____________________________________
ECHO cluster analysis and vectorisation

# cluster analysis
imagery_classification 1 -GRIDS=%INPUT% -CLUSTER=cluster.sgrd -NORMALISE -NCLUSTER=5

# majority filter
grid_filter          6 -INPUT=cluster.sgrd -RADIUS=3

# vectorization
shapes_grid          6 -GRID=cluster.sgrd -POLYGONS=cluster.shp -CLASS_ALL=1

# select cluster class 1
shapes_tools         3 -SHAPES=cluster.shp -FIELD=ID -EXPRESSION="a = 1"

# save selectione
shapes_tools         6 -INPUT=cluster.shp -OUTPUT=cluster_class1.shp
```

```
Usage:

saga_cmd [-h, --help]
saga_cmd [-v, --version]
saga_cmd [-b, --batch]
saga_cmd [-d, --docs]
saga_cmd [-f, --flags][=qrsilpx][-c, --cores][=#] <LIBRARY>
saga_cmd [-f, --flags][=qrsilpx][-c, --cores][=#] <SCRIPT>

[-h], [--help]    : help on usage
[-v], [--version]: print version information
[-b], [--batch]   : create a batch file example
[-d], [--docs]    : create module documentation in current wo...
[-c], [--cores]   : number of physical processors to use for...
[-f], [--flags]   : various flags for general usage [qrsilpx...
  q               : no progress report
  r               : no messages report
  s               : silent mode (no progress and no messages...
  i               : allow user interaction
  l               : load translation dictionary
  p               : load projections dictionary
  x               : use XML markups for synopses and messages

<LIBRARY>         : name of the library
<MODULE>          : either name or index of the module
<OPTIONS>         : module specific options
<SCRIPT>          : saga cmd script file with one or more module calls
```

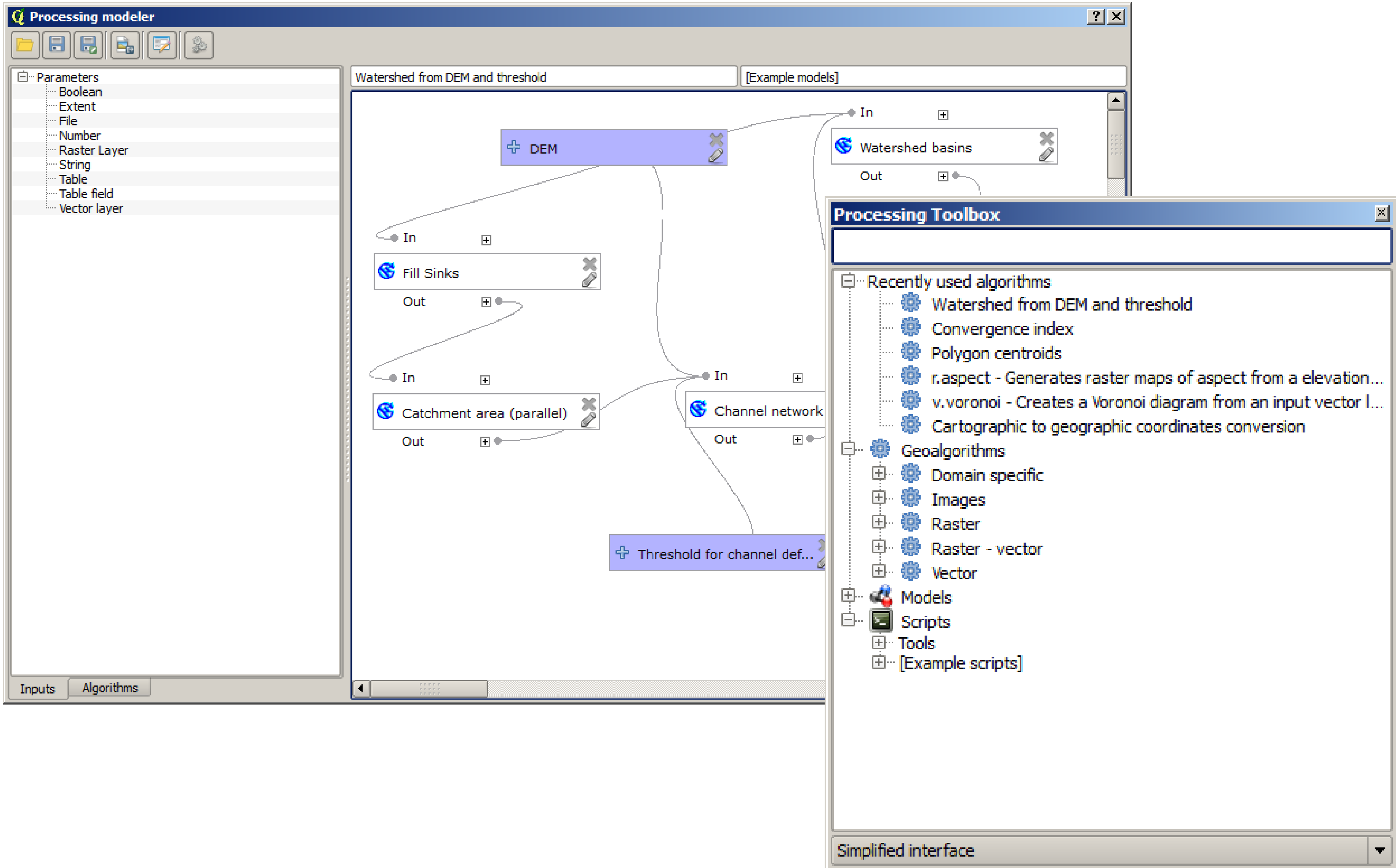
# Front Ends | Scripting

- Via the **SWIG** compiler (Simplified Wrapper and Interface Generator) it is possible to expose the SAGA API as well as SAGA tools to various (script) programming languages, e.g. Python, Java, C#, R.
- The **Python** interface allows to run SAGA modules from (web)server processes and the integration with many other software e.g. ArcGIS.
- SAGA modules can be execute directly from R Scripts via the **RSAGA** interface.

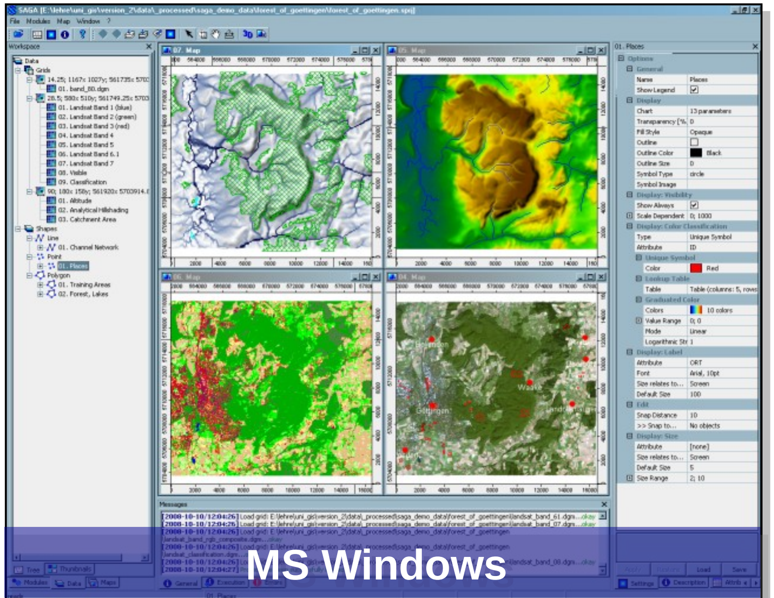



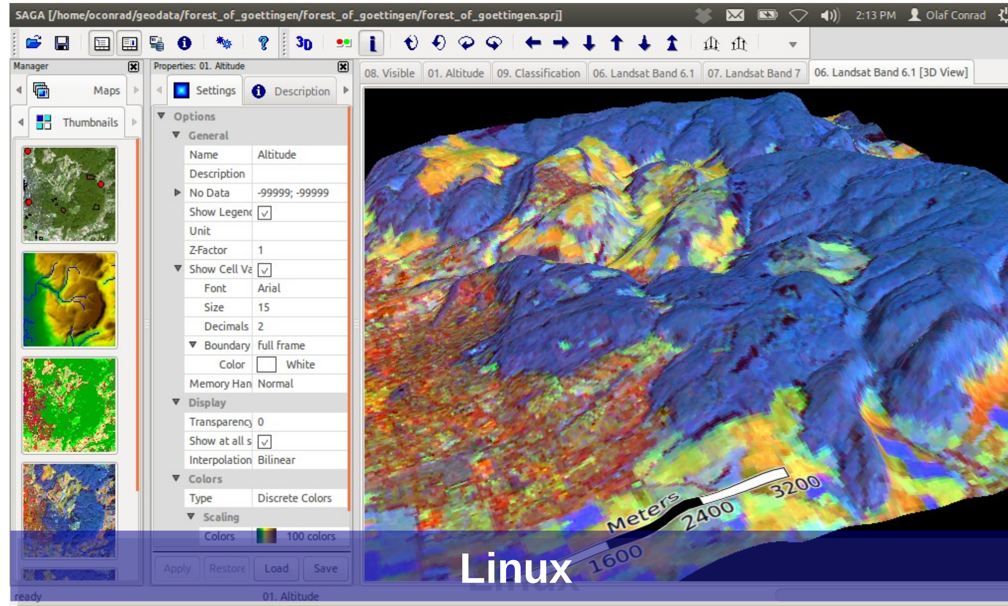



- Via ZOO-WPS http://www.zoo-project.org/docs/kernel/sagagis.html
- Via QGIS Processing

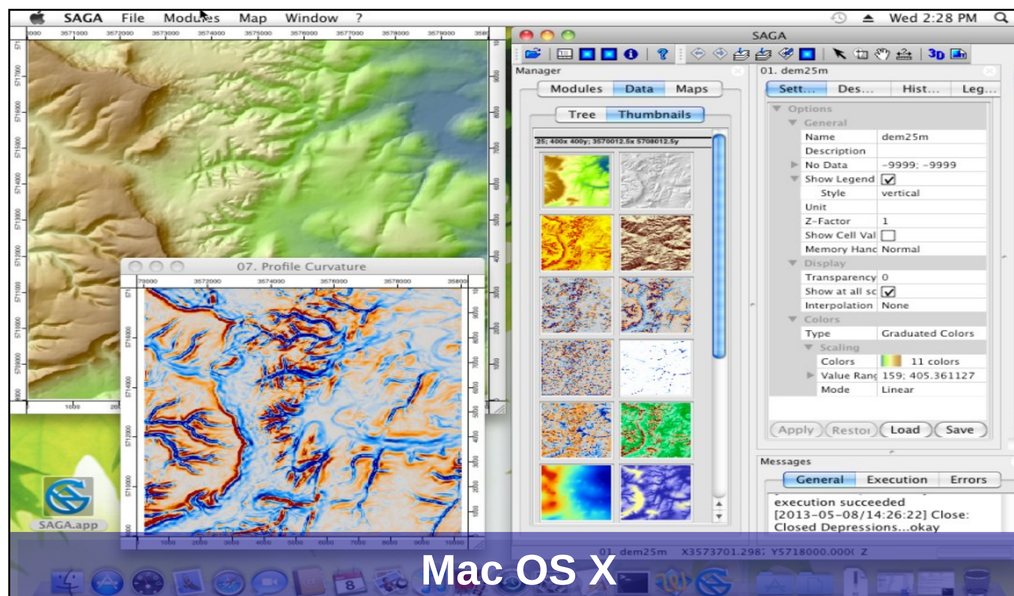


# Supported Platforms

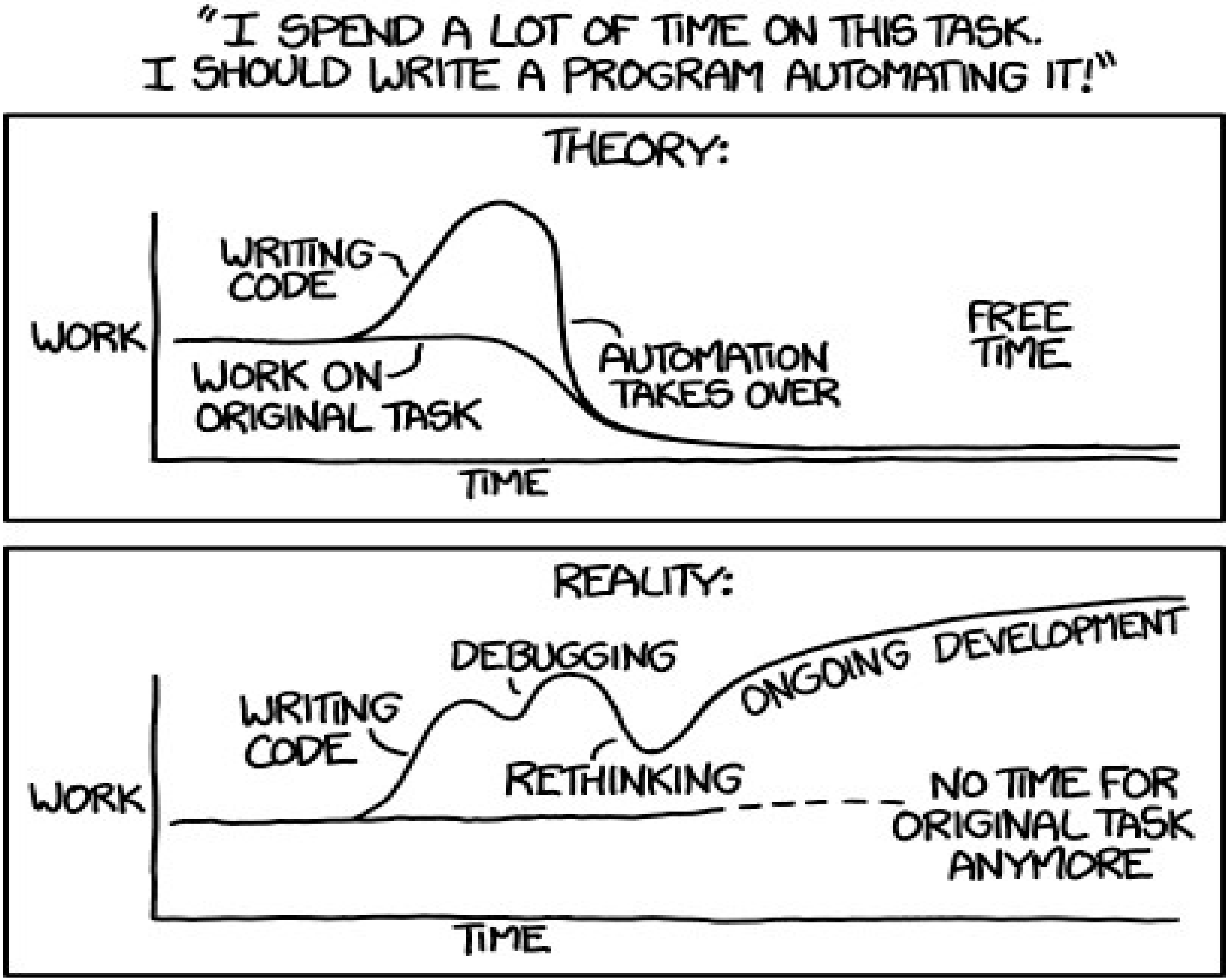



- MS Windows
- Linux
- FreeBSD
  - Maintainer: Rainer Hurling, NW-FVA
- Mac OS X
  - Maintainer: Wanted !

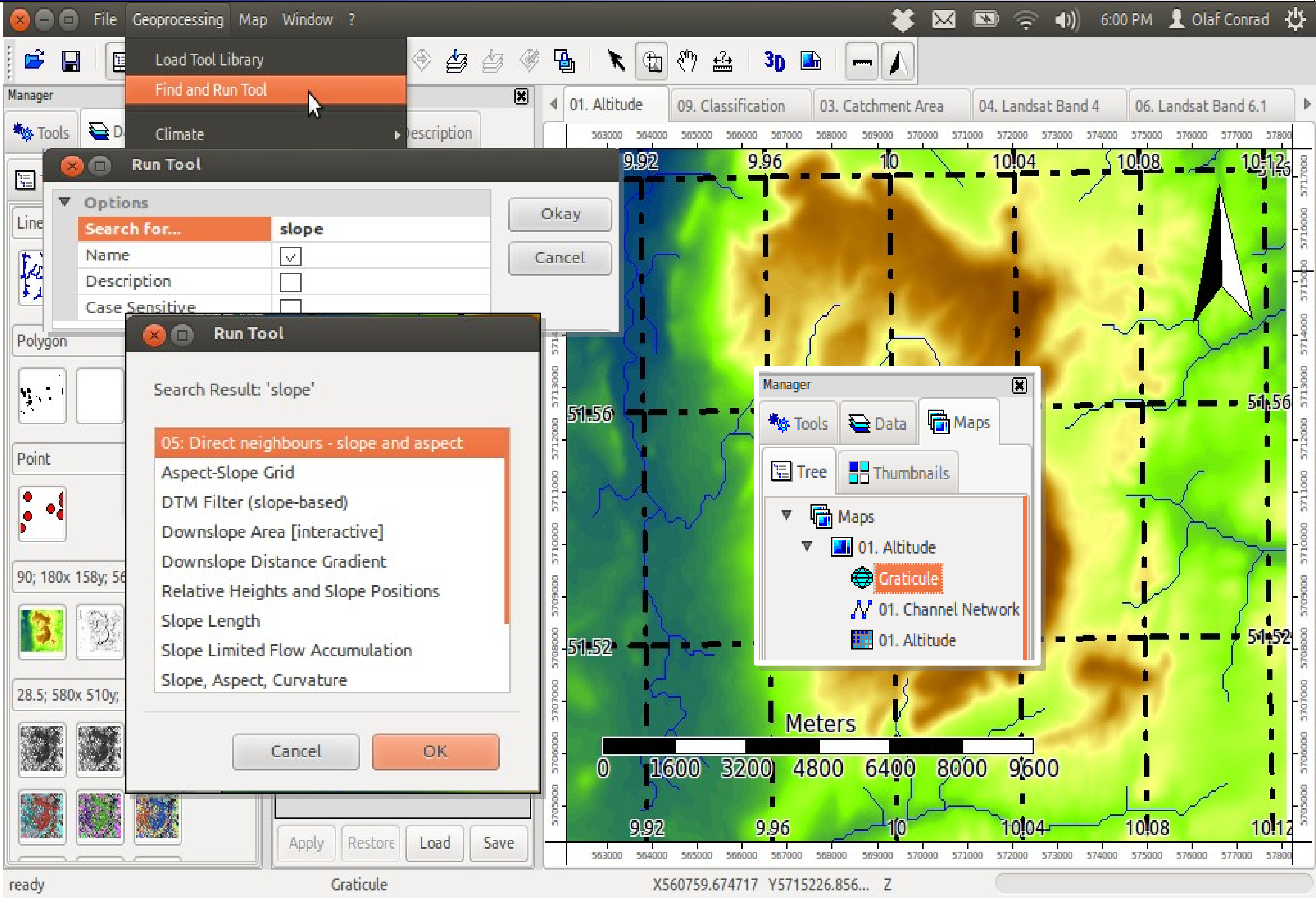


# The SAGA Development



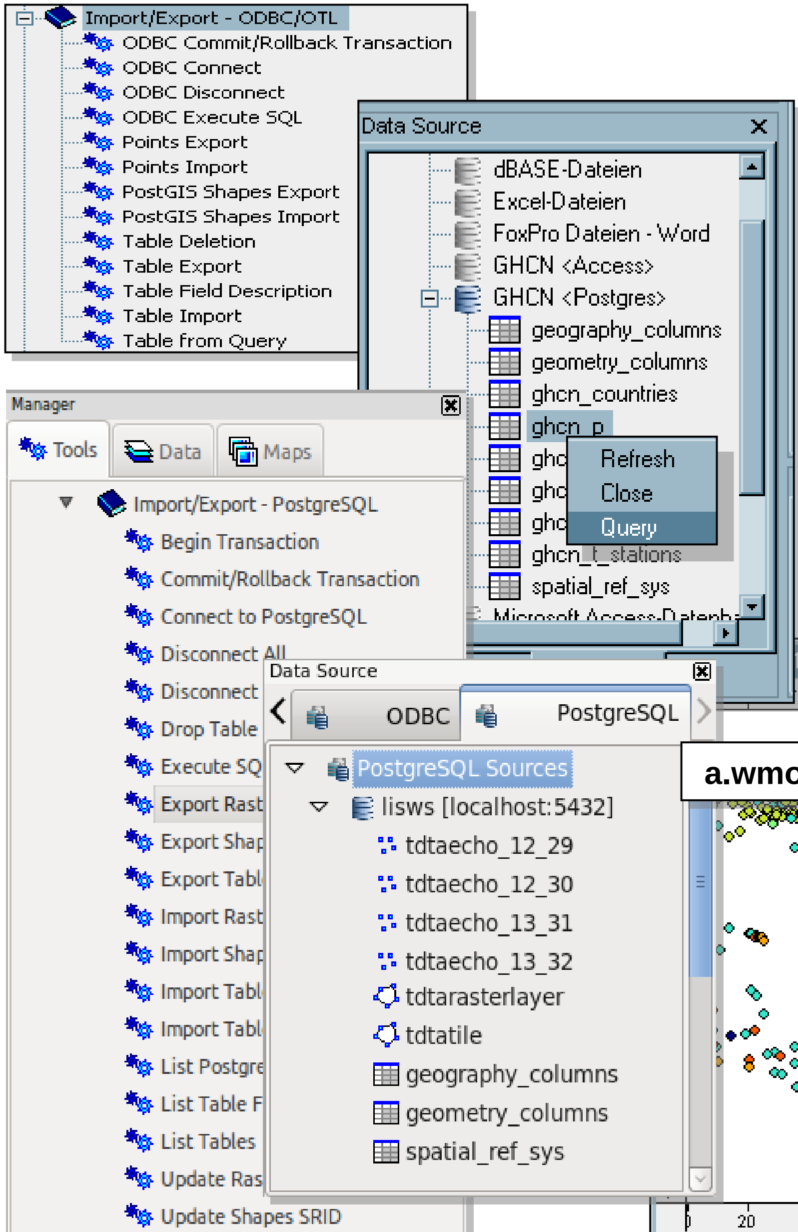

*Source: xkcd (http://imgs.xkcd.com/comics/automation.png)*

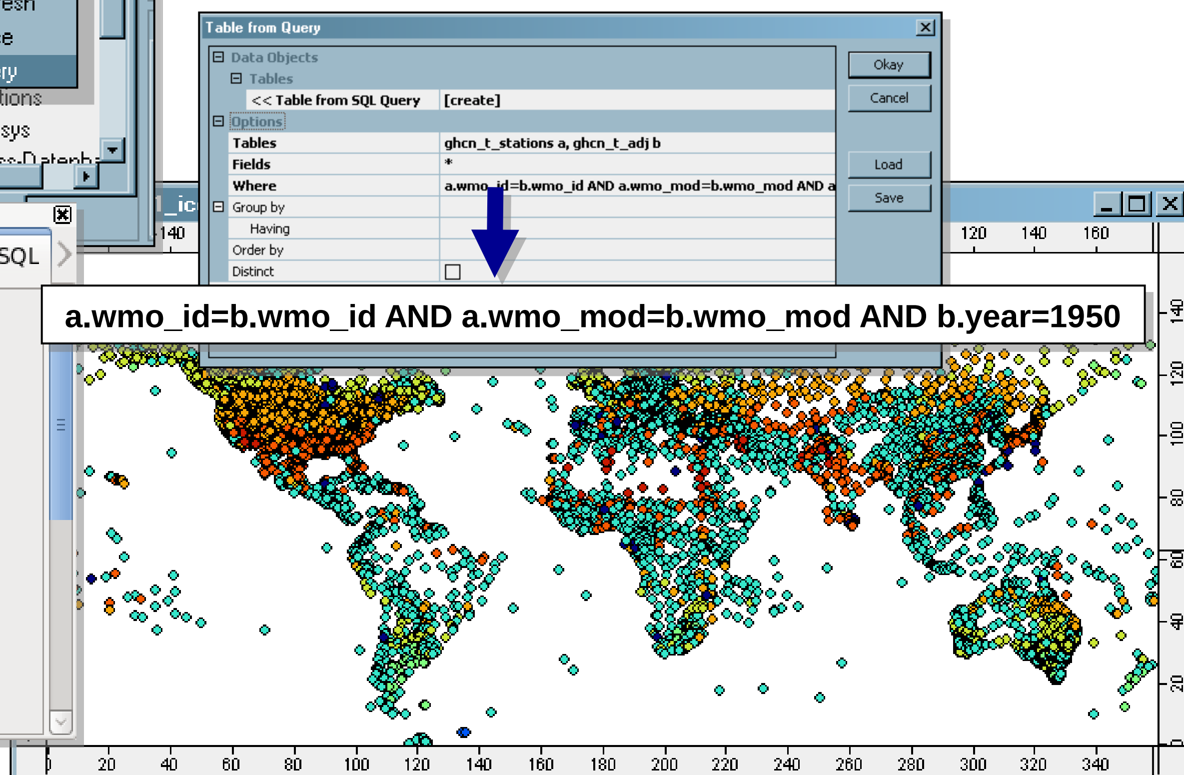
# Changes and New Features in the GUI



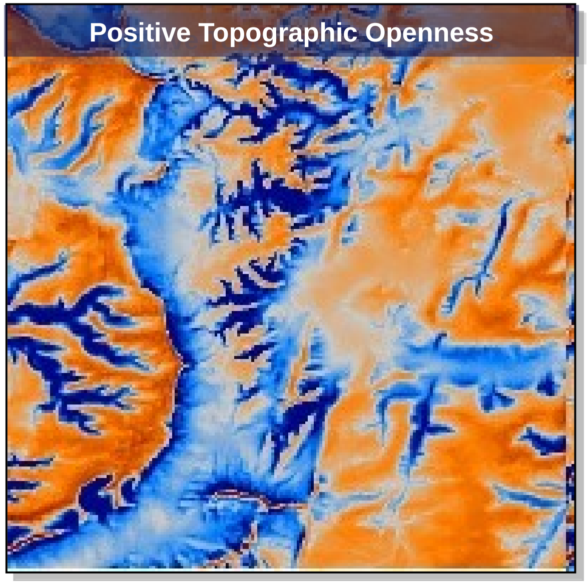
# New Tools | Data Base Integration

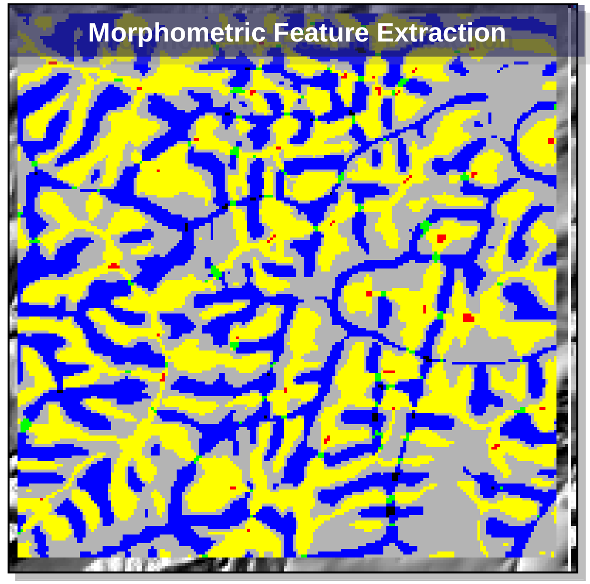
- Database access via Open Data Base Connection (ODBC) interface.
  - SQL – Structured Query Language
  - Problem: binary data types (e.g. BLOBs)
- PostgreSQL + PostGIS
  - Direct linking

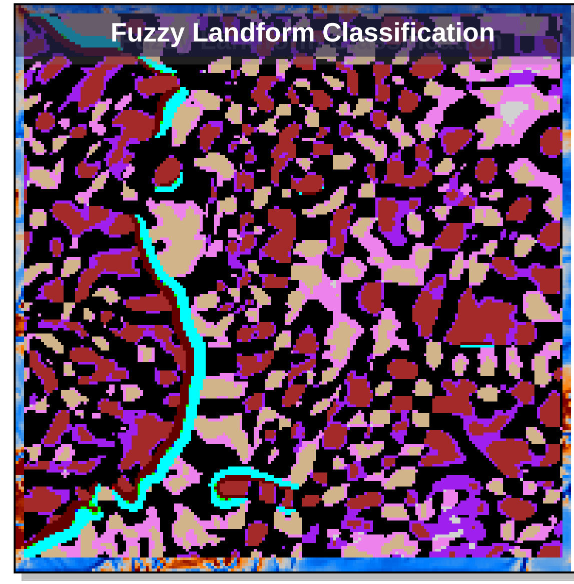


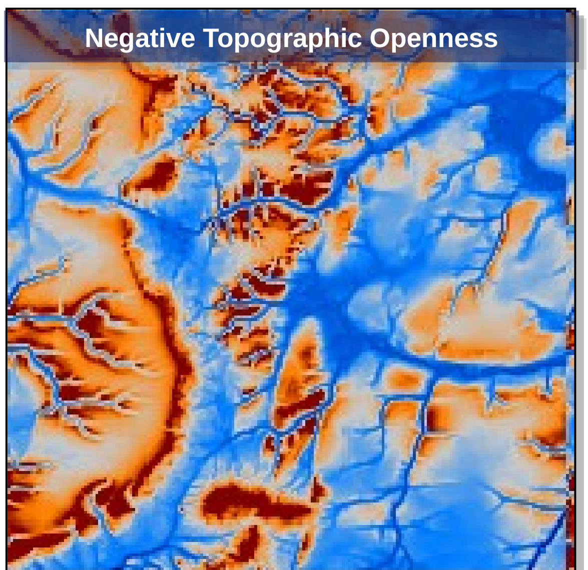



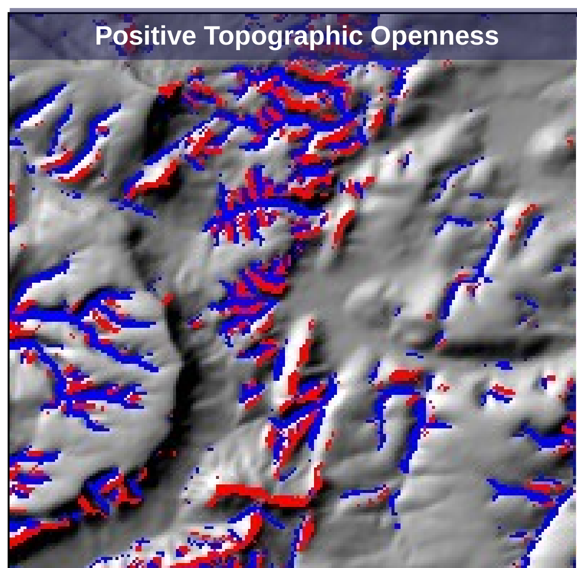
# New Tools | Terrain Analysis & Classification

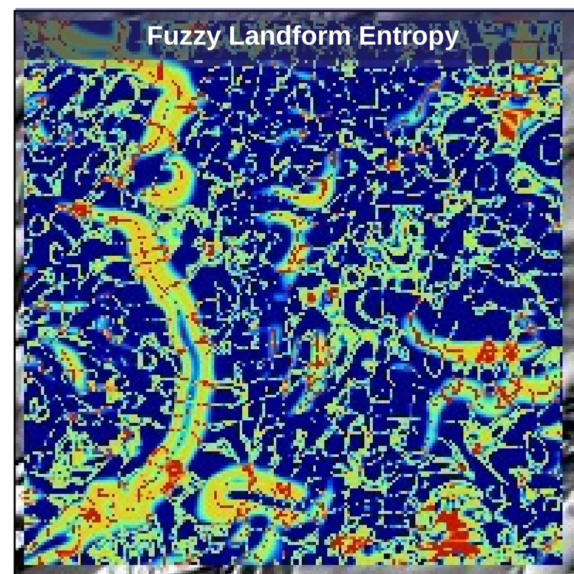

Positive Topographic Openness


Morphometric Feature Extraction


Fuzzy Landform Classification

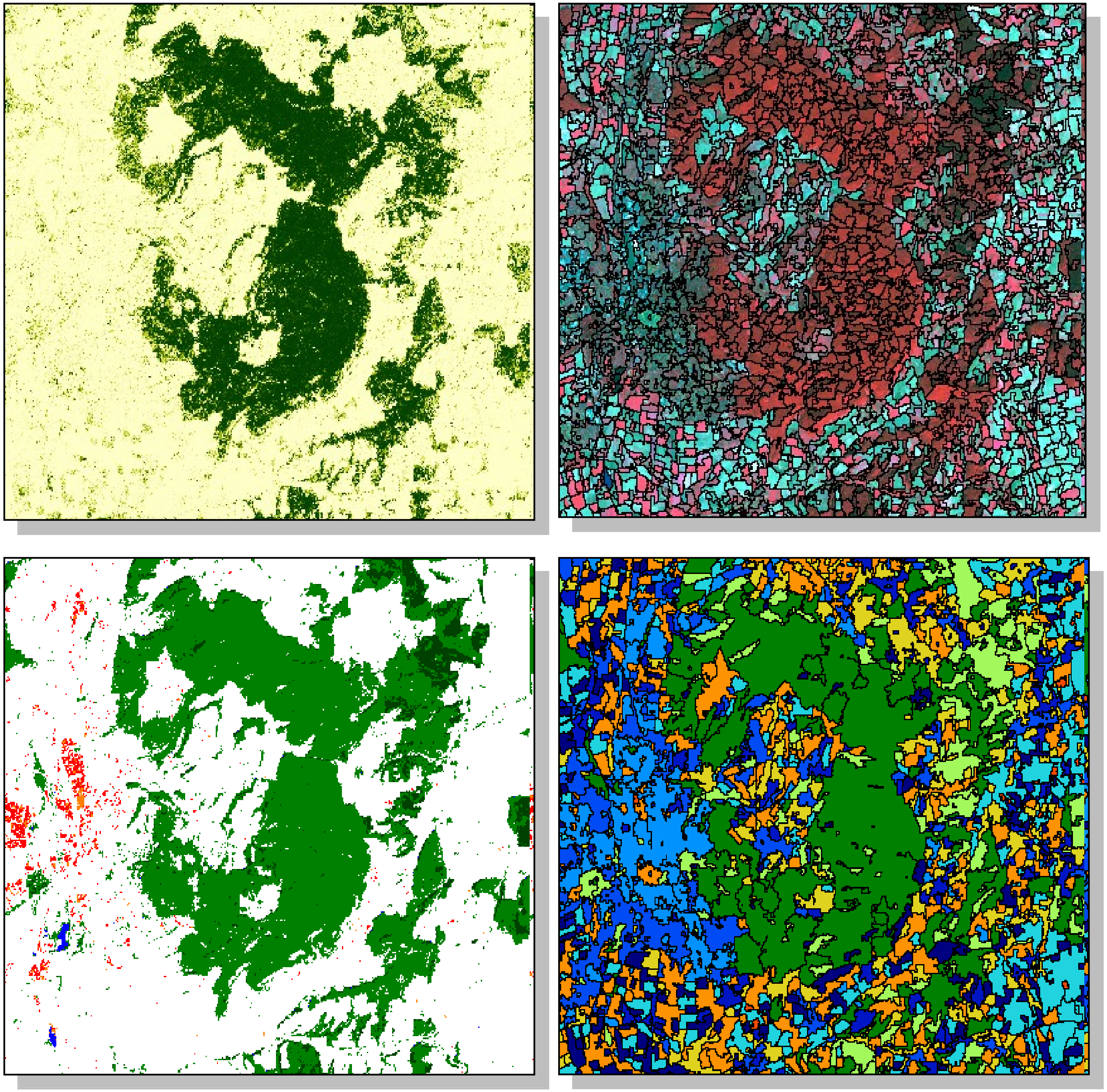

Negative Topographic Openness


Positive Topographic Openness

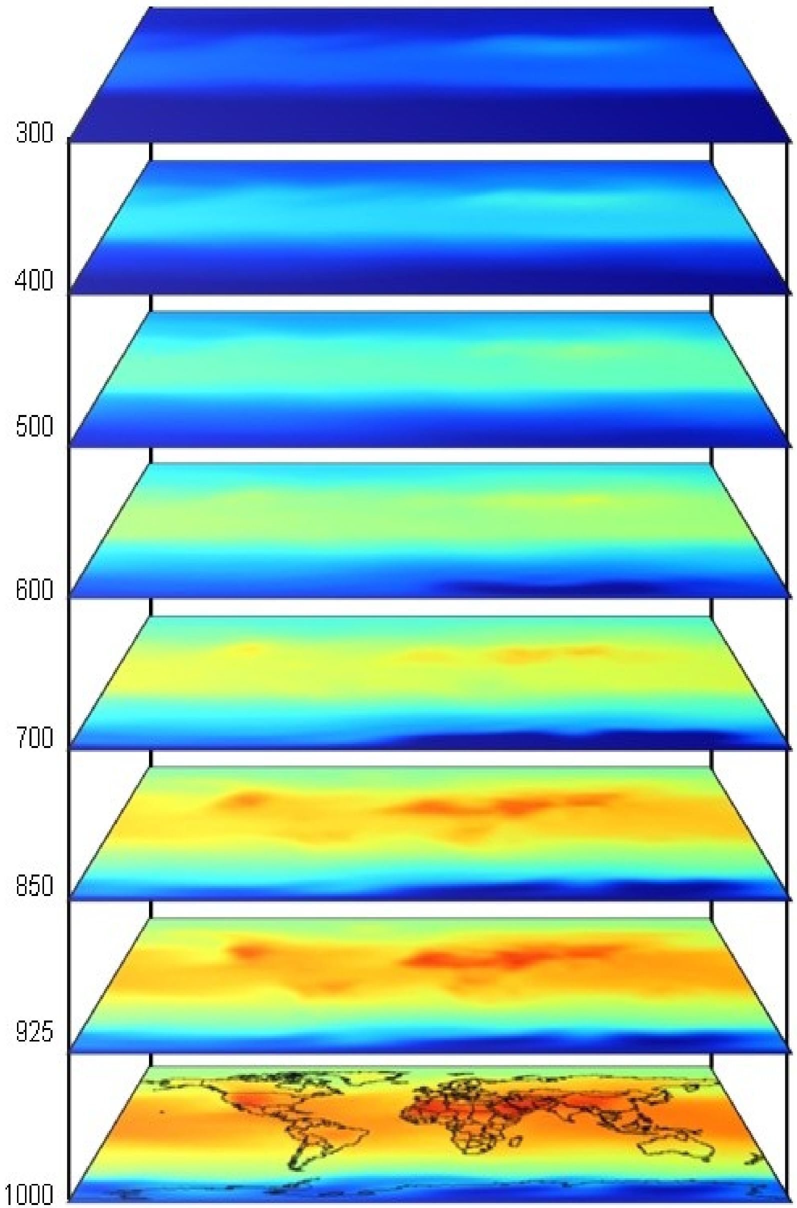

Fuzzy Landform Entropy

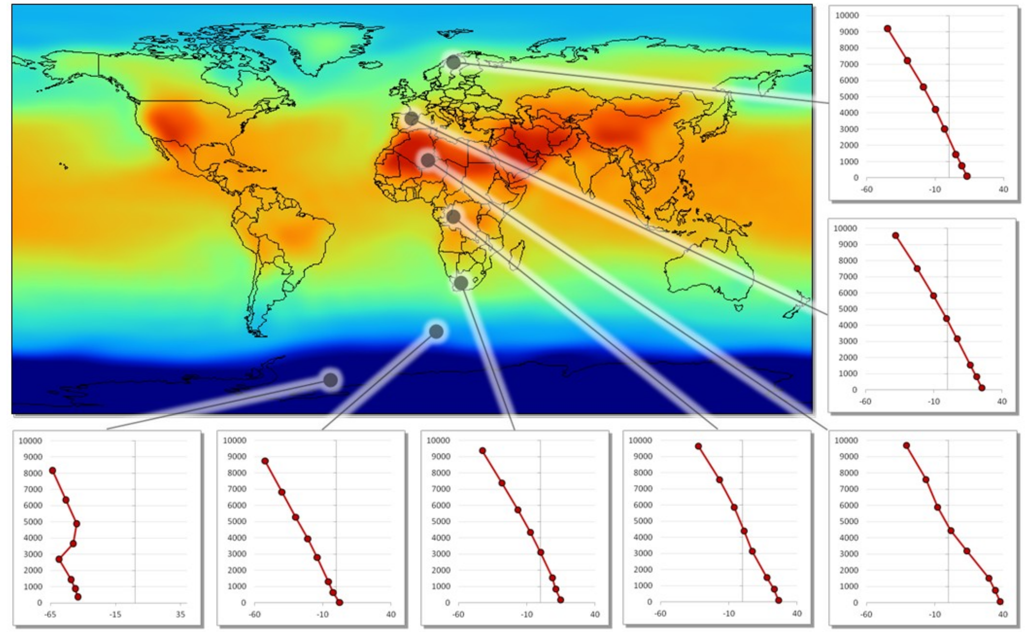
# New Tools | Remote Sensing & Image Analysis

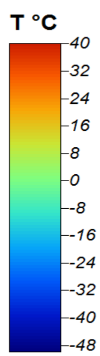
- Landsat Tools
  - Reflectance from metadata
  - Cloud Cover Assessment
- Orthorectification
  - From flight parameters
- Classification Tools
  - Support Vector Machine (SVM)
  - Maximum Entropy
  - Random Forest
  - OBIA



# New Tools | Climate Data & Regionalization

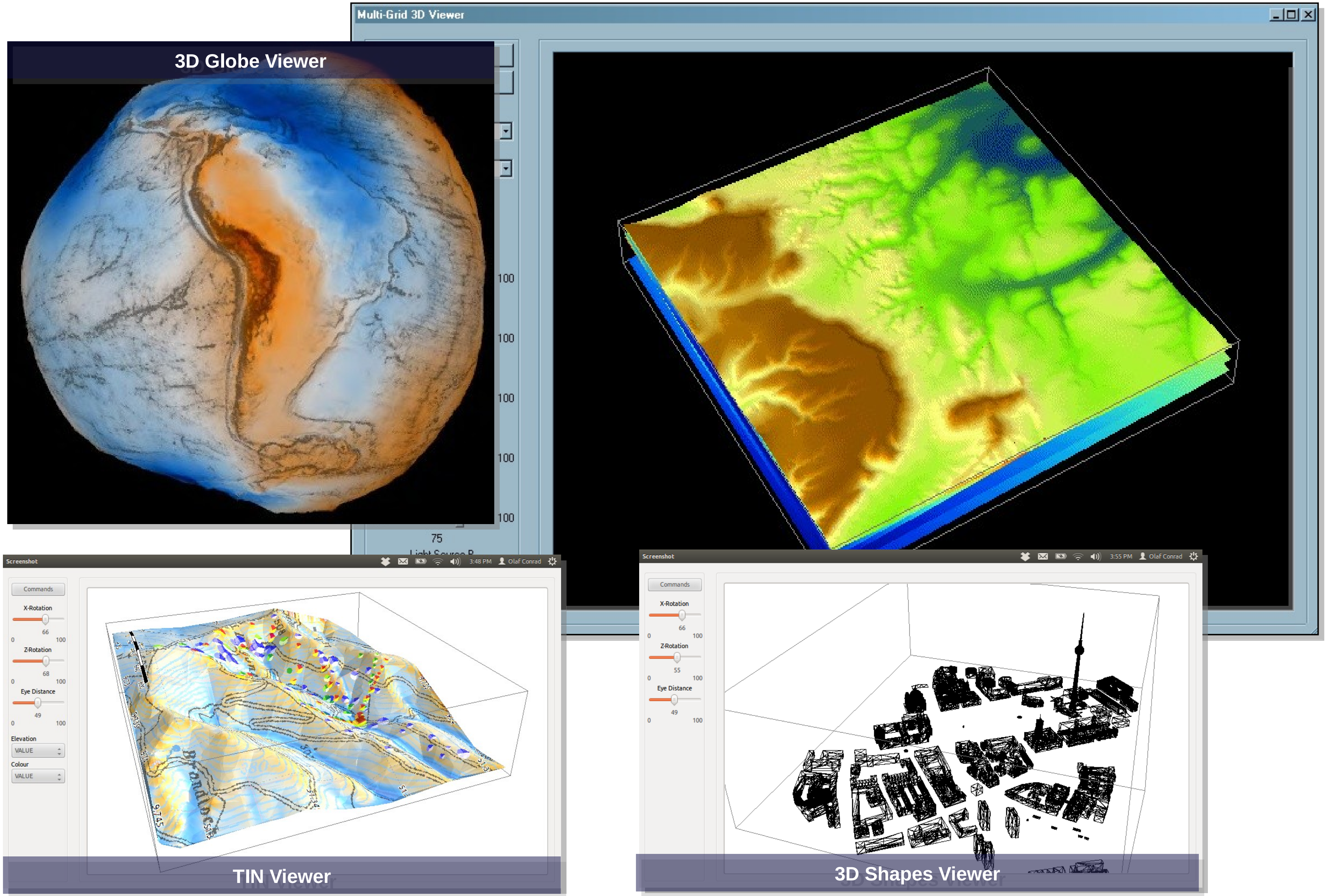




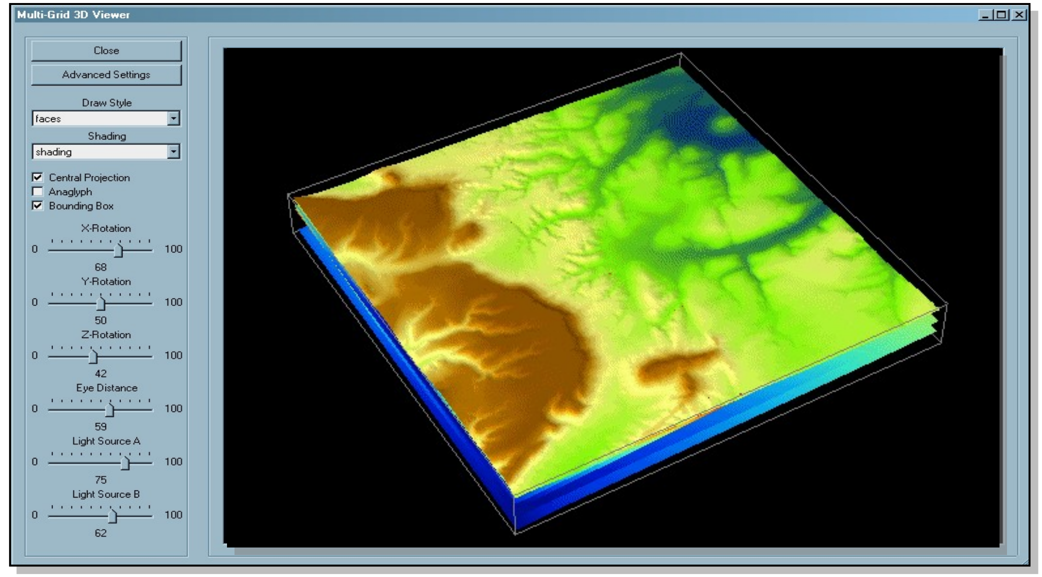

- Improved NetCDF support
- Downscaling tools
  - General and Regional Climate Models (GCM / RCM)

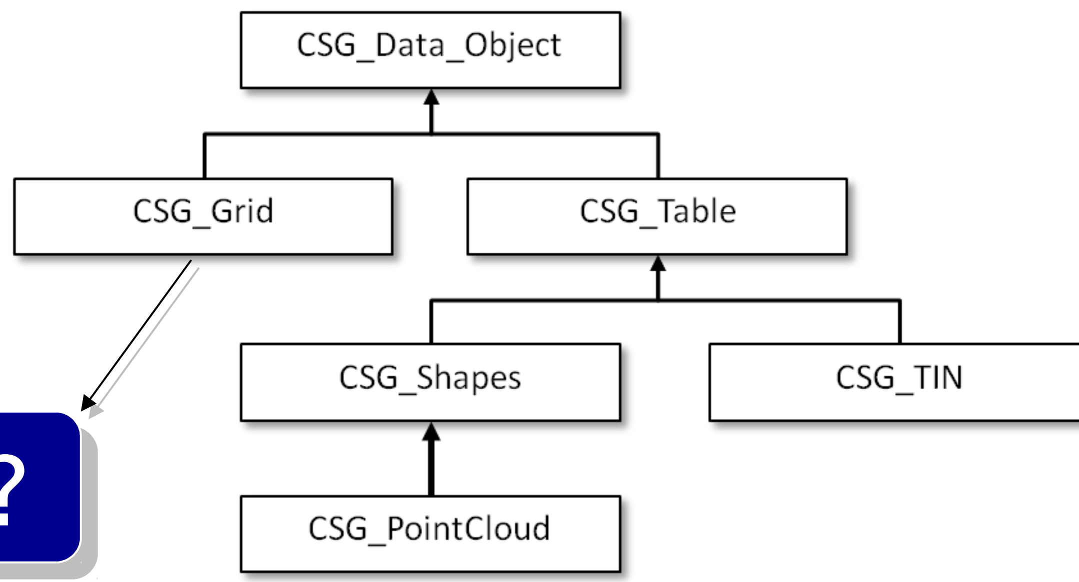
# New Tools | 3D Viewer



# Outlook | Towards Multidimensional Raster

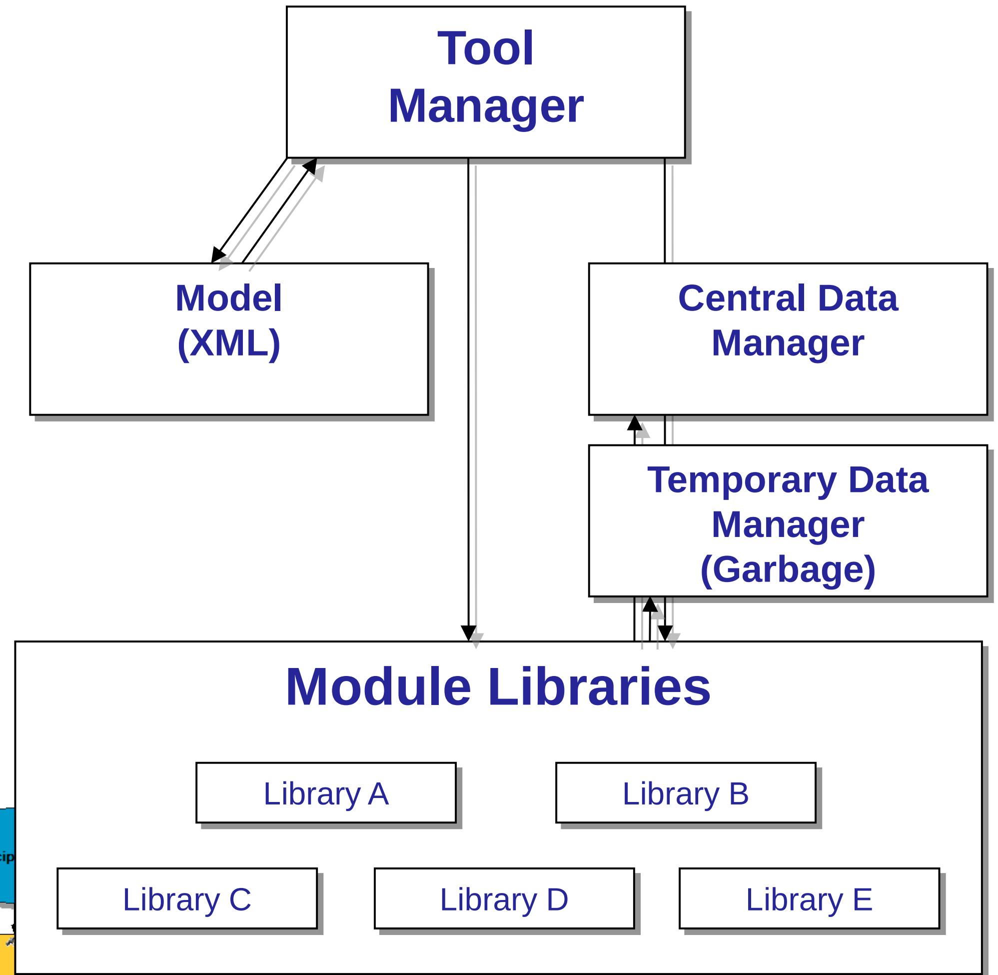
- There is a need for ‚Multi-Raster'
  - Multi-/Hyperspectral Data
    - Remote Sensing
  - Volume Representation
    - Geology, Soils, Atmosphere
  - Time Series
- Points for discussion
  - Class inheritance
  - 3 or more dimensions
  - Visualisation tools
  - How to use with standard grid tools

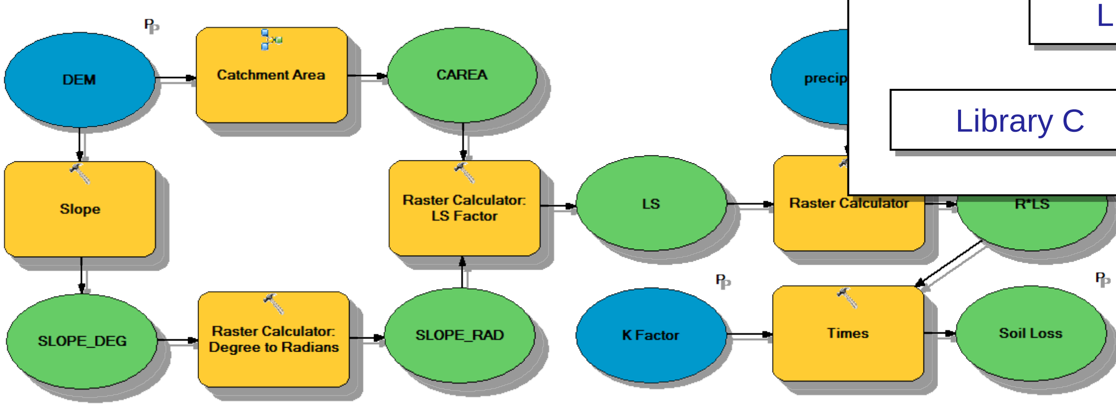


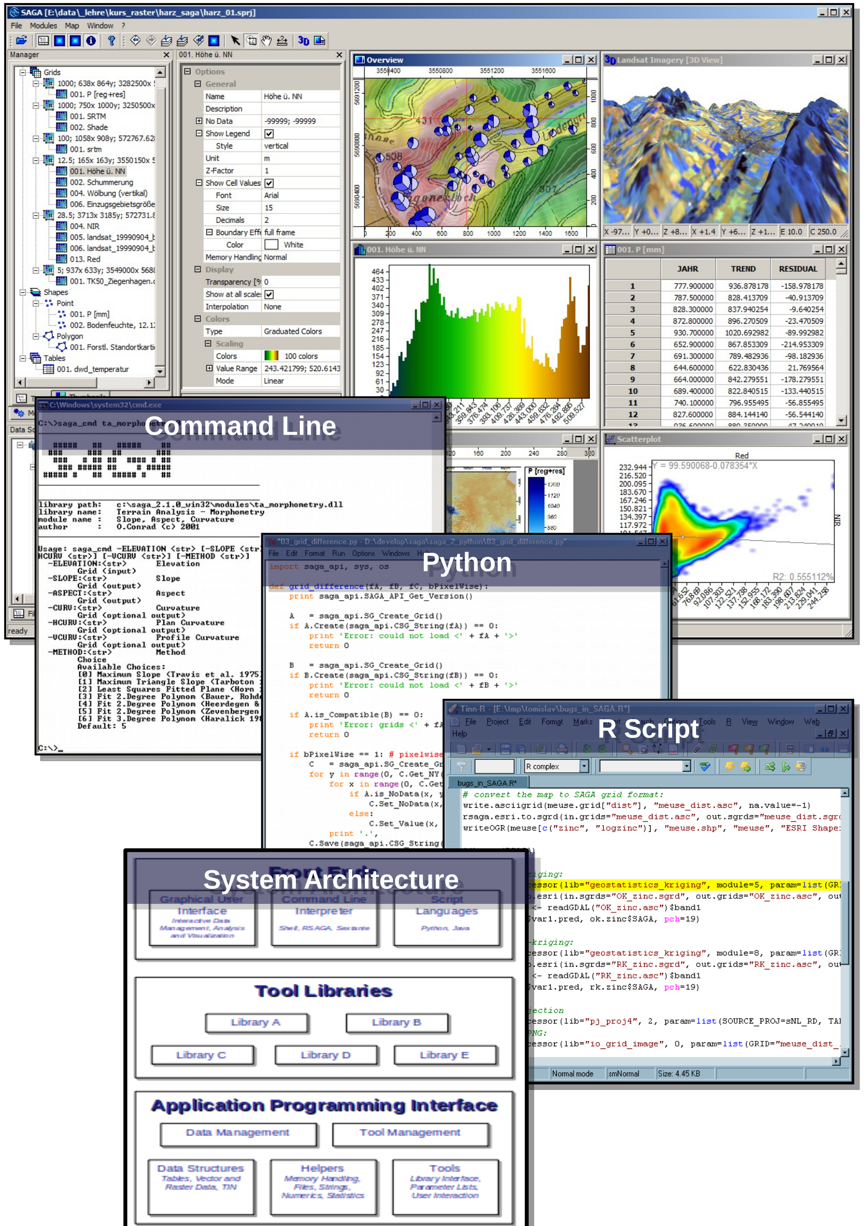
# Outlook | Towards a SAGA Model Builder

- XML based definition
  - Model parameters
  - Which tools to use and how to combine them
- Interpretation by the Module Manager
  - Performs plausibility checks
  - Module execution
  - Temporary data to a garbage collector
- Model building
  - Typing XML code
  - From data set history
  - Visual model designer



# SAGA | System for Automated Geoscientific Analyses



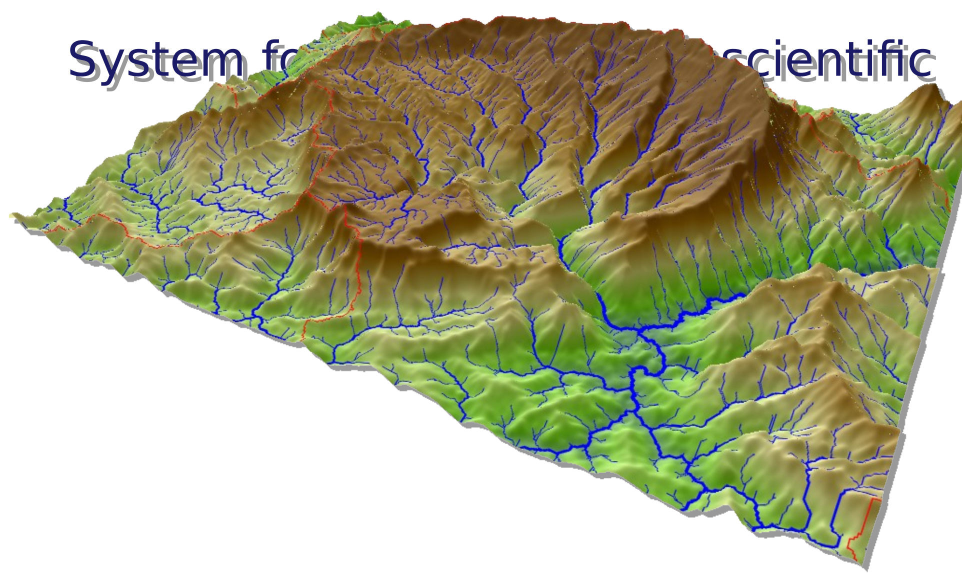

## Ways of Automation

*with*

# SAGA

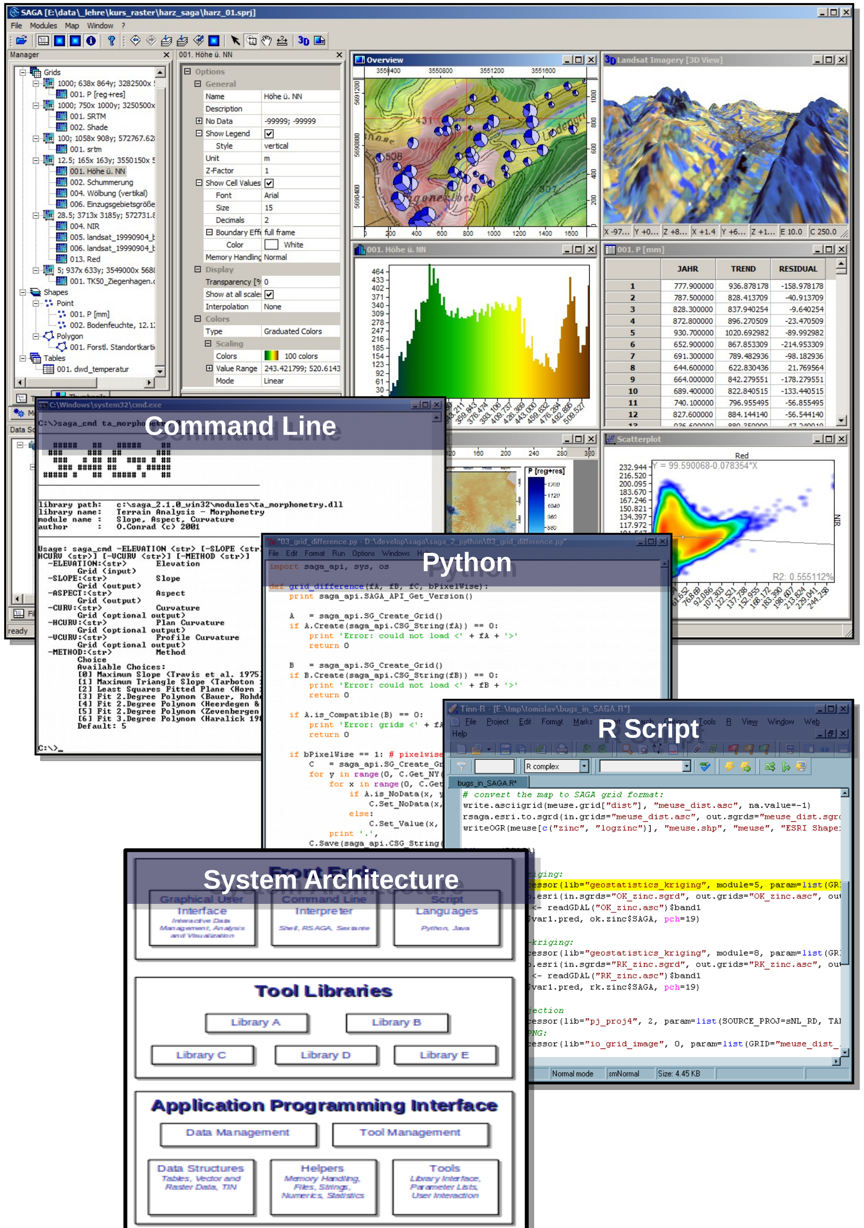
System fo

scientific



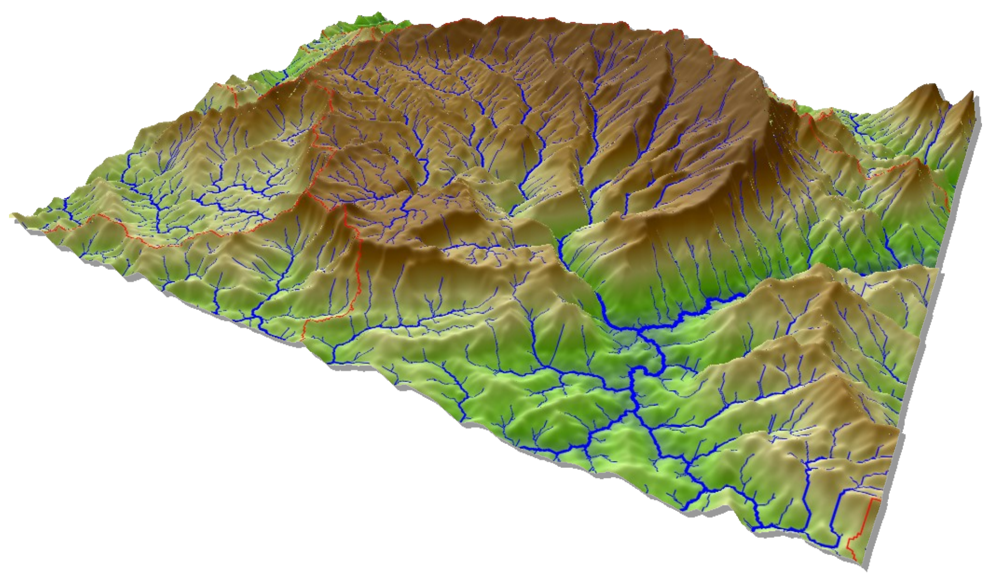
*Michael Bock, Olaf Conrad, Volker Wichmann*

# SAGA | System for Automated Geoscientific Analyses

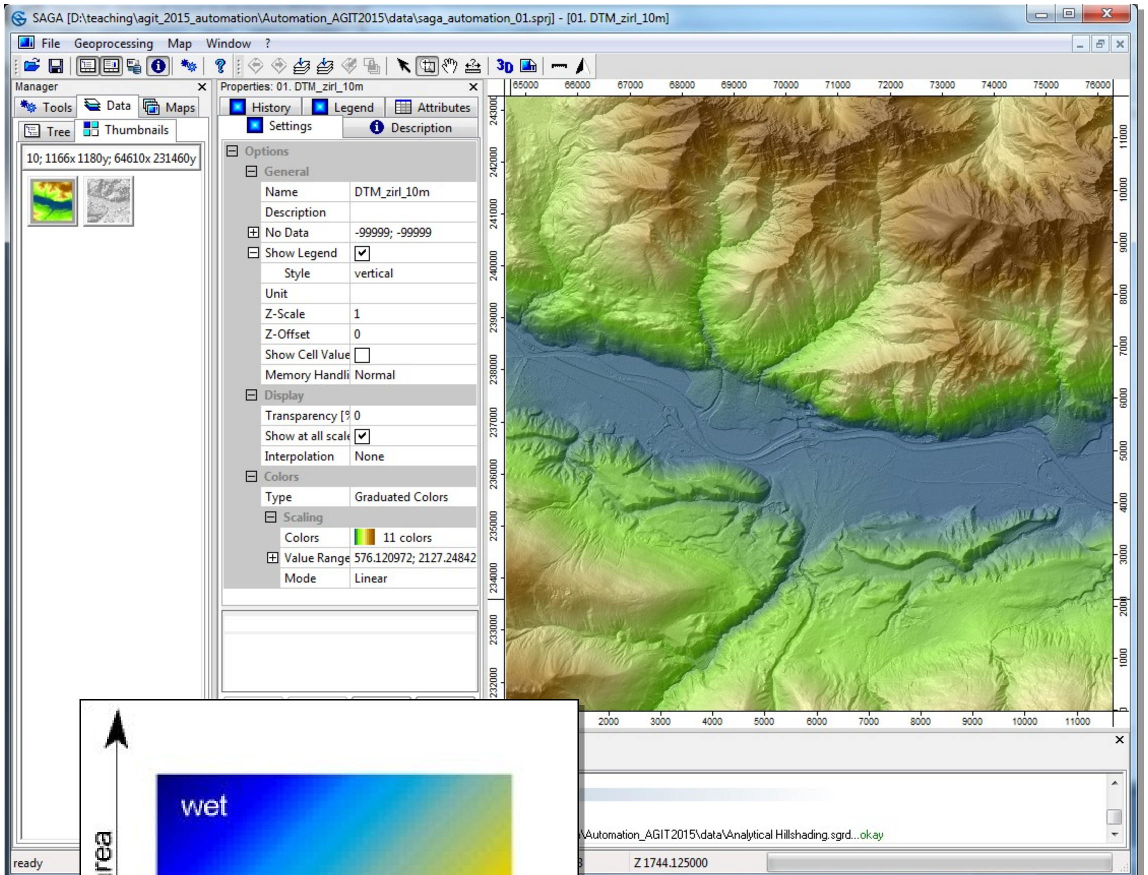


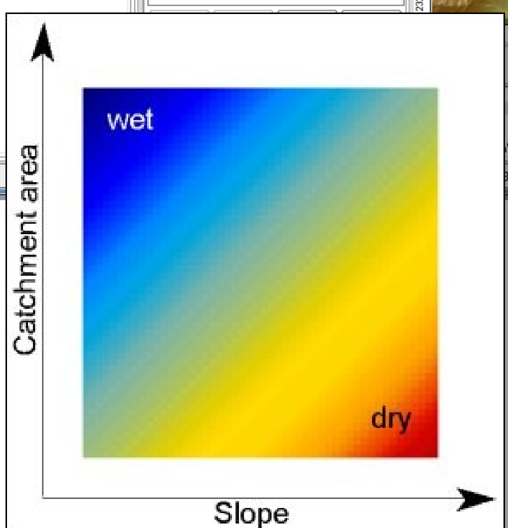

## Ways of Automation

### Tool Chains



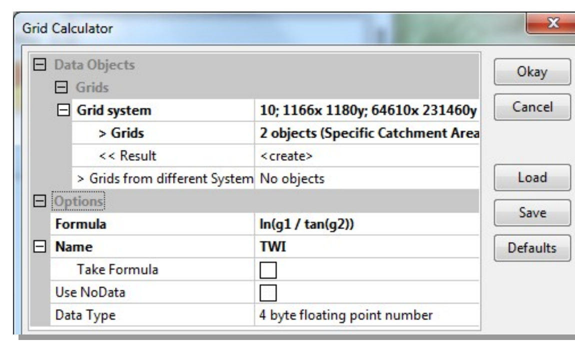
# TWI | Topographic Wetness Index



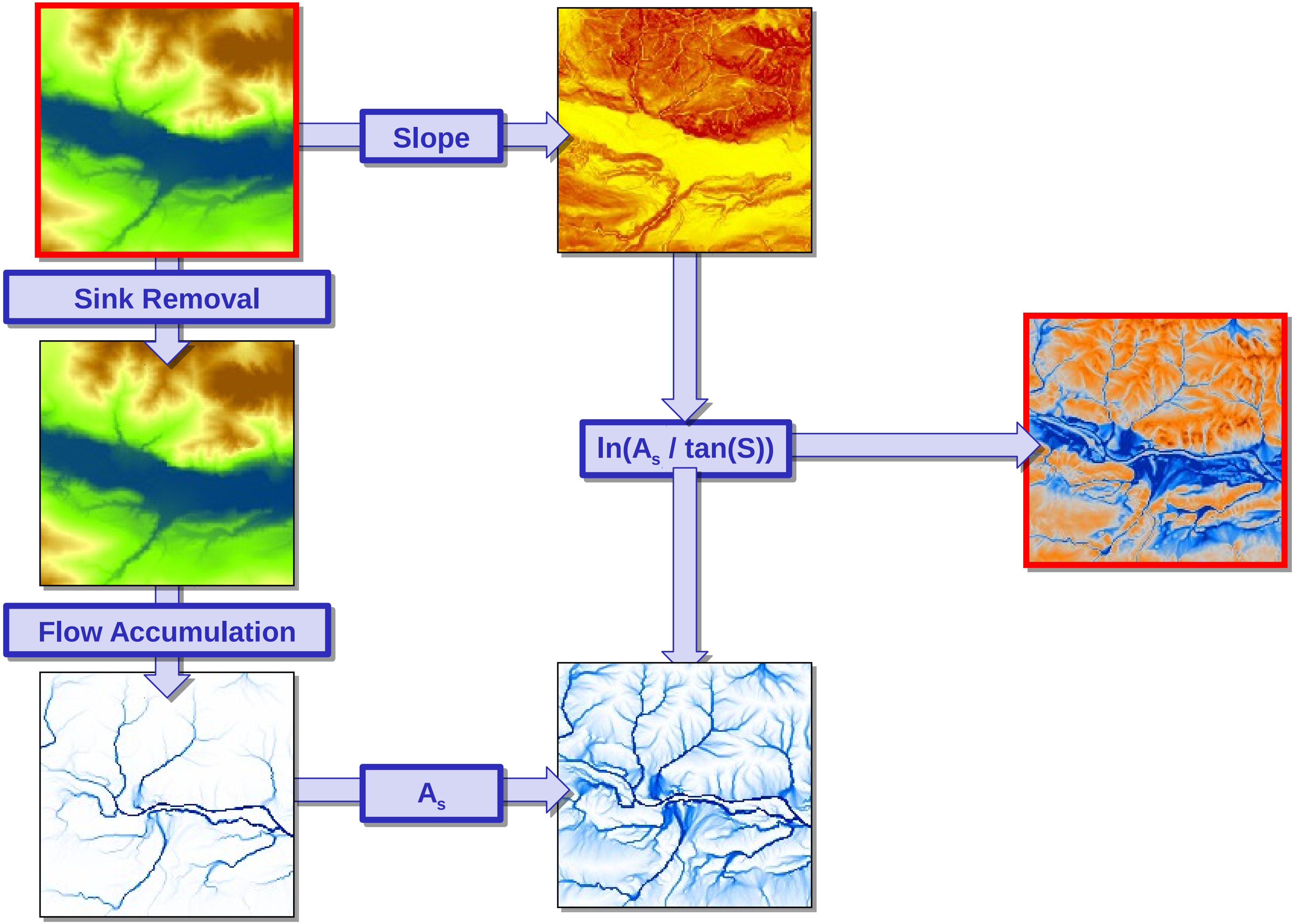




$$W = \ln\left(\frac{A}{\tan(\beta)}\right)$$

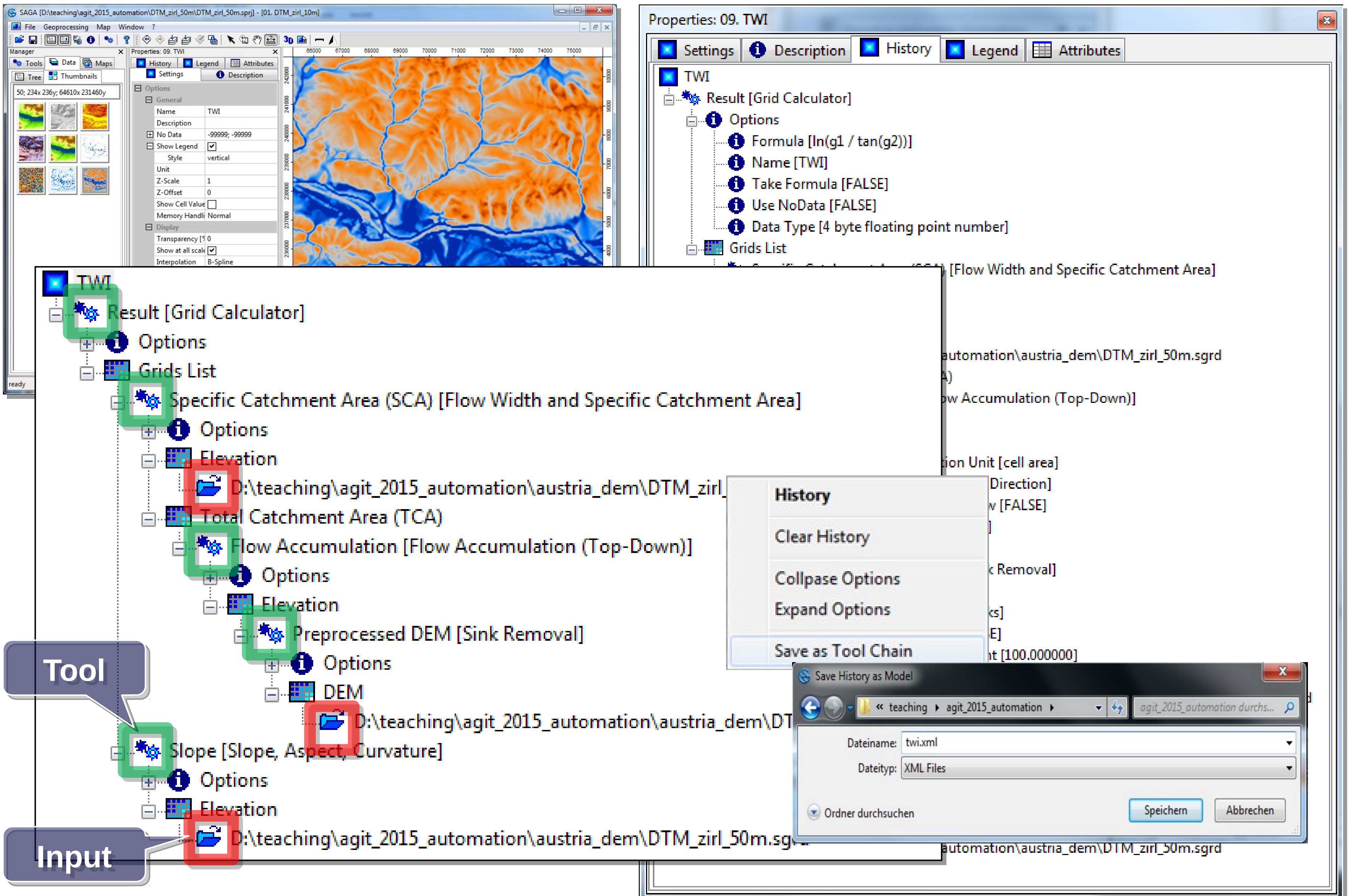
- TWI = log(As / tan(S))
  - As = Specific Catchment Area
  - S = Slope Angle
- Input Data
  - DEM
- Tools
  1. Slope, Aspect, Curvature
  2. Sink Removal
  3. Flow Accumulation
  4. Flow Width and Specific Catchment Area
  5. Grid Calculator



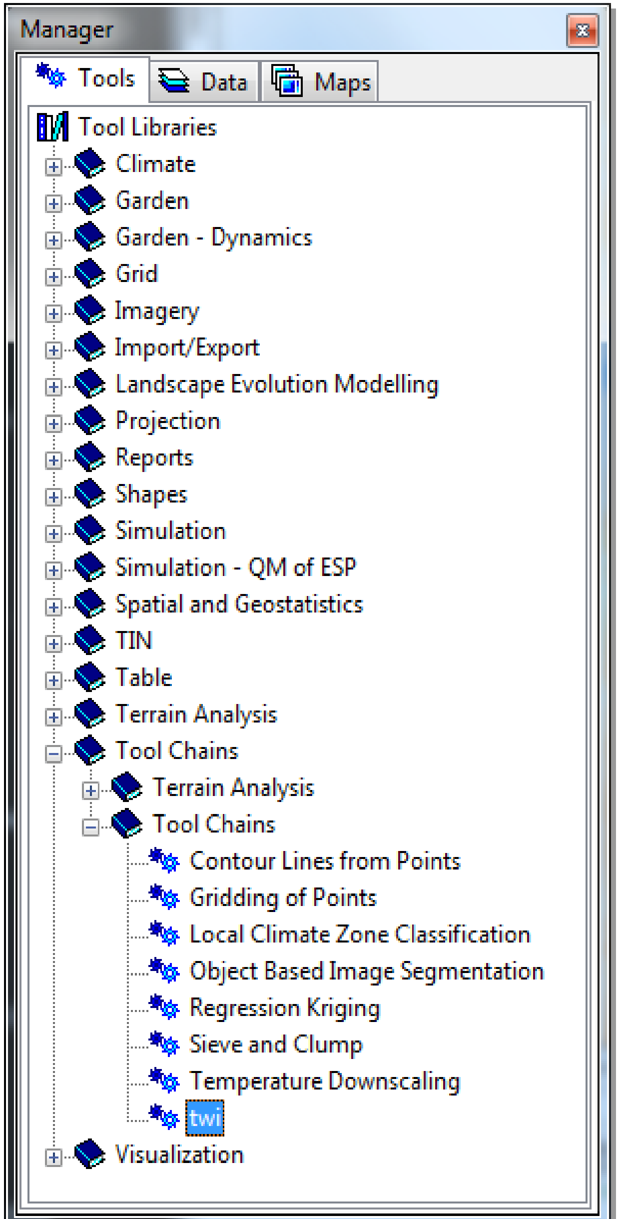
# TWI | Workflow

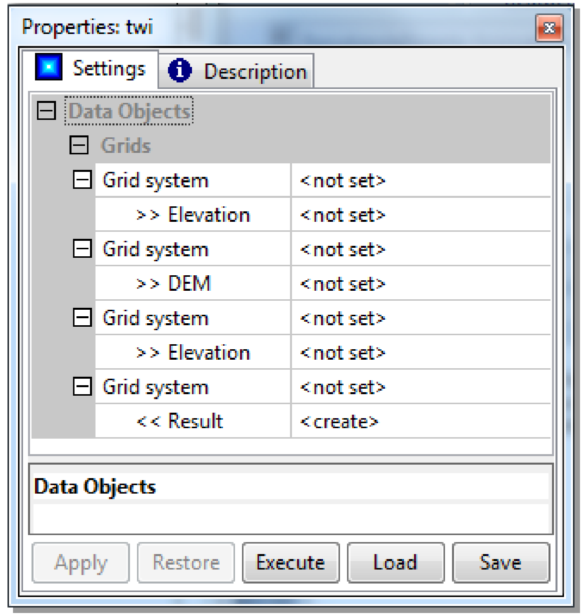


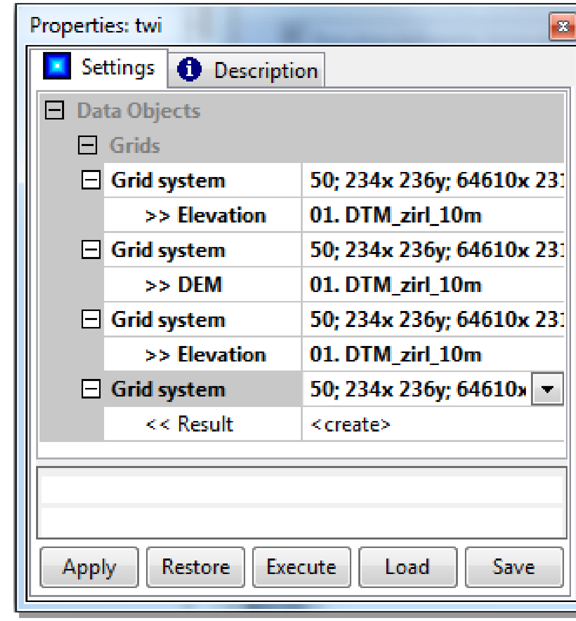
# TWI | Creation History

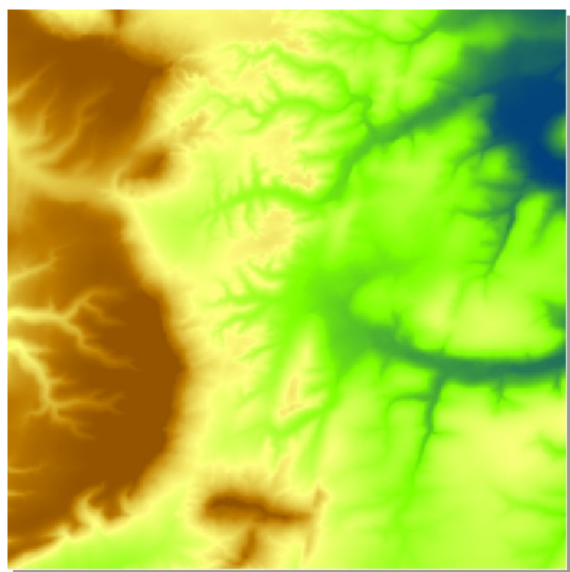


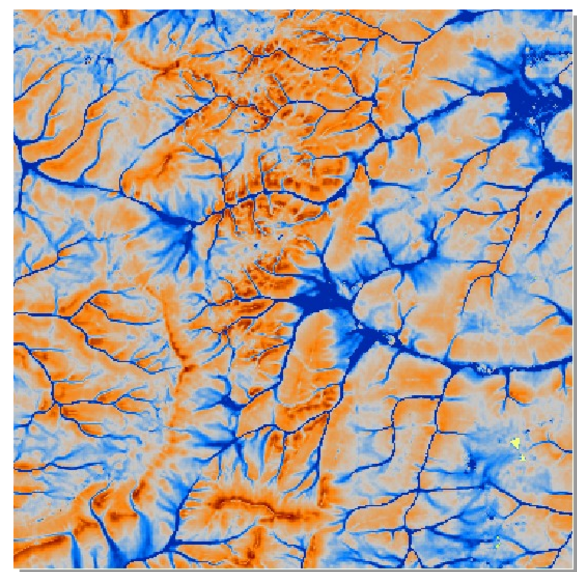
# TWI | Apply Tool Chain to Another Data Set



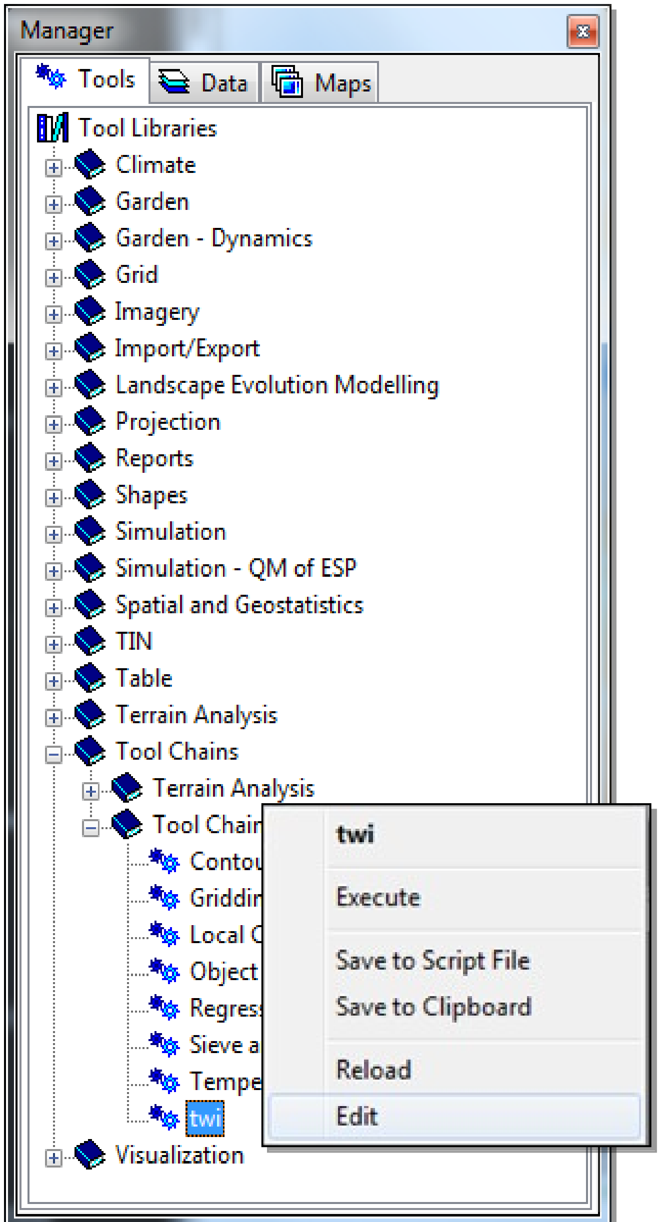



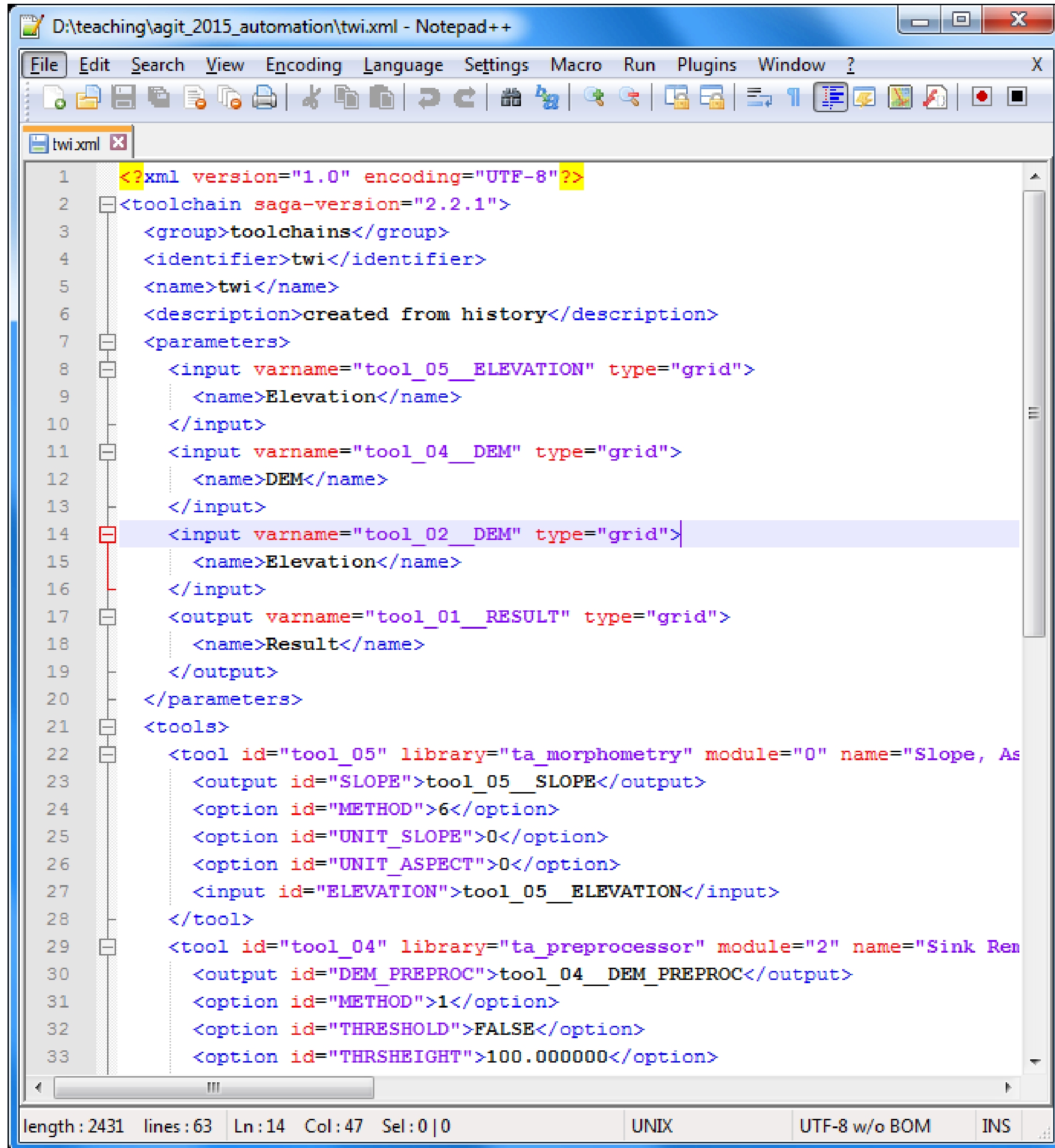





# TWI | Edit Tool Chain



# Tool Chains | Main Structure

## <toolchain>

- Attributes
  - saga-version
- Content
  - <group>
  - <identifier>
  - <name>
  - <author>
  - <description>
  - <menu>
  - <parameters>
  - <tools>

```
<?xml version="1.0" encoding="UTF-8"?>
<toolchain saga-version="2.2.1">
  <group>toolchains</group>
  <identifier>twi</identifier>
  <name>twi</name>
  <description>created from history</description>
  <parameters>
    <option varname="GRID_SYSTEM" type="grid_system">
    <input varname="DEM" type="grid" parent="GRID_SYSTEM">
    <output varname="TWI" type="grid" parent="GRID_SYSTEM">
  </parameters>
  <tools>
    <tool id="tool_05" library="ta_morphometry" module="0" name
    <tool id="tool_04" library="ta_preprocessor" module="2" nar
    <tool id="tool_03" library="ta_hydrology" module="0" name="
    <tool id="tool_02" library="ta_hydrology" module="19" name=
    <tool id="tool_01" library="grid_calculus" module="1" name=
  </tools>
</toolchain>
```

# Tool Chains | General Key Words

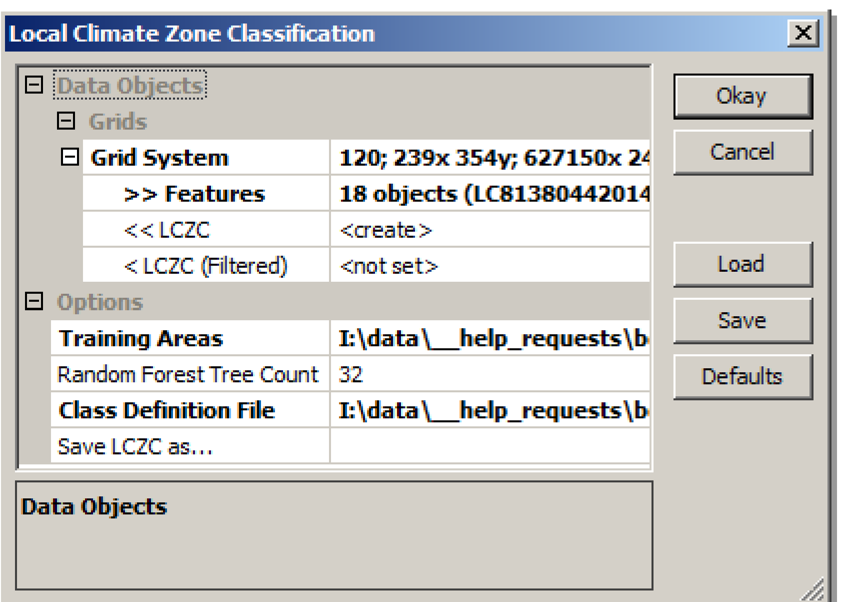
- <group>
  - Content: A category to which the tool belongs to
- <identifier> [obligatory]
  - Content: unique(!) id, used to identify and run the tool from other scripts or tool chains
- <name>
  - Content: human readable name of the tool
- <author>
  - Content: the authors, copyrights, …
- <description>
  - Content: more explicit description.
- <menu>
  - Attributes:
    - absolute=true/false (default=false)
  - Content: menu path of the tool (saga_gui)

# Tool Chains | The Parameters Interface

## <parameters>

## Content

- <input>
- <output>
- <option>




```xml
<parameters>
    <option varname="GRID_SYSTEM" type="grid_system">
        <name>Grid System</name>
    </option>
    <option varname="FILE_TRAINING" type="file">
        <name>Training Areas</name>
        <filter>KML/KMZ Files|*.kml;*.kmz|All Files|*.*</filter>
    </option>
    <option varname="RF_TREE_COUNT" type="integer">
        <name>Random Forest Tree Count</name>
        <value min="1">32</value>
        <description>How many trees to create?</description>
    </option>
    <option varname="FILTER_RADIUS" type="integer">
        <condition type="exists">LCZC_FILTERED</condition>
        <name>Majority Filter Radius</name>
        <value min="1">2</value>
    </option>
    <option varname="FILE_CLASS_DEF" type="file">
        <name>Class Definition File</name>
        <filter>Table Files|*.txt;*.dbf;*csv|All Files|*.*</filter>
    </option>
    <option varname="FILE_LCZC" type="file" save="true">
        <name>Save LCZC as...</name>
        <value></value>
        <filter>KMZ Files|*.kmz|All Files|*.*</filter>
    </option>
    <option varname="FILE_LCZC_FILTERED" type="file" save="true">
        <condition type="exists">LCZC_FILTERED</condition>
        <name>Save LCZC (Filtered) as...</name>
        <value></value>
        <filter>KMZ Files|*.kmz|All Files|*.*</filter>
    </option>
    <input varname="FEATURES" type="grid_list" parent="GRID_SYSTEM">
        <name>Features</name>
    </input>
    <output varname="LCZC" type="grid" parent="GRID_SYSTEM">
        <name>LCZC</name>
    </output>
    <output varname="LCZC_FILTERED" type="grid" optional="true" paren
        <name>LCZC (Filtered)</name>
    </output>
</parameters>
```

# Tool Chains | Input & Output Data

## <input>, <output>

- Attributes
  - varname: unique(!) variable id
  - type: data set type
  - parent: e.g. a single grid system for grids
- Content
  - <condition>
    - conditionally en-/disable the parameter
  - <name>
    - human readable name
  - <description>
    - more explicit parameter description

| Type | |
|---|---|
| grid | |
| table | |
| shapes | |
| tin | |
| points | |
| grid_list | |
| table_list | |
| shapes_list | |
| tin_list | |
| points_list | |

# Tool Chains | Options

## <option>

- Attributes
  - varname: unique(!) variable id
  - type: variable type
  - parent: e.g. a table for table field selection
- Content
  - <condition>
    - conditionally en-/disable the parameter
  - <name>
    - human readable name
  - <description>
    - more explicit parameter description
  - <…>
    - dependent on the option type, various type specific attributes and content keys might apply, e.g.:
    - <value> default value for numeric and text variables
    - <filter> format filter for file selection

| Type | |
|---|---|
| node | |
| boolean | |
| integer | |
| double | |
| degree | |
| range | |
| choice | |
| text | |
| long_text | |
| file | |
| table_field | |
| table_fields | |
| grid_system | |

# Tool Chains | The Tools Section

## <tools>

- Content
  - <tool>
  - <condition>

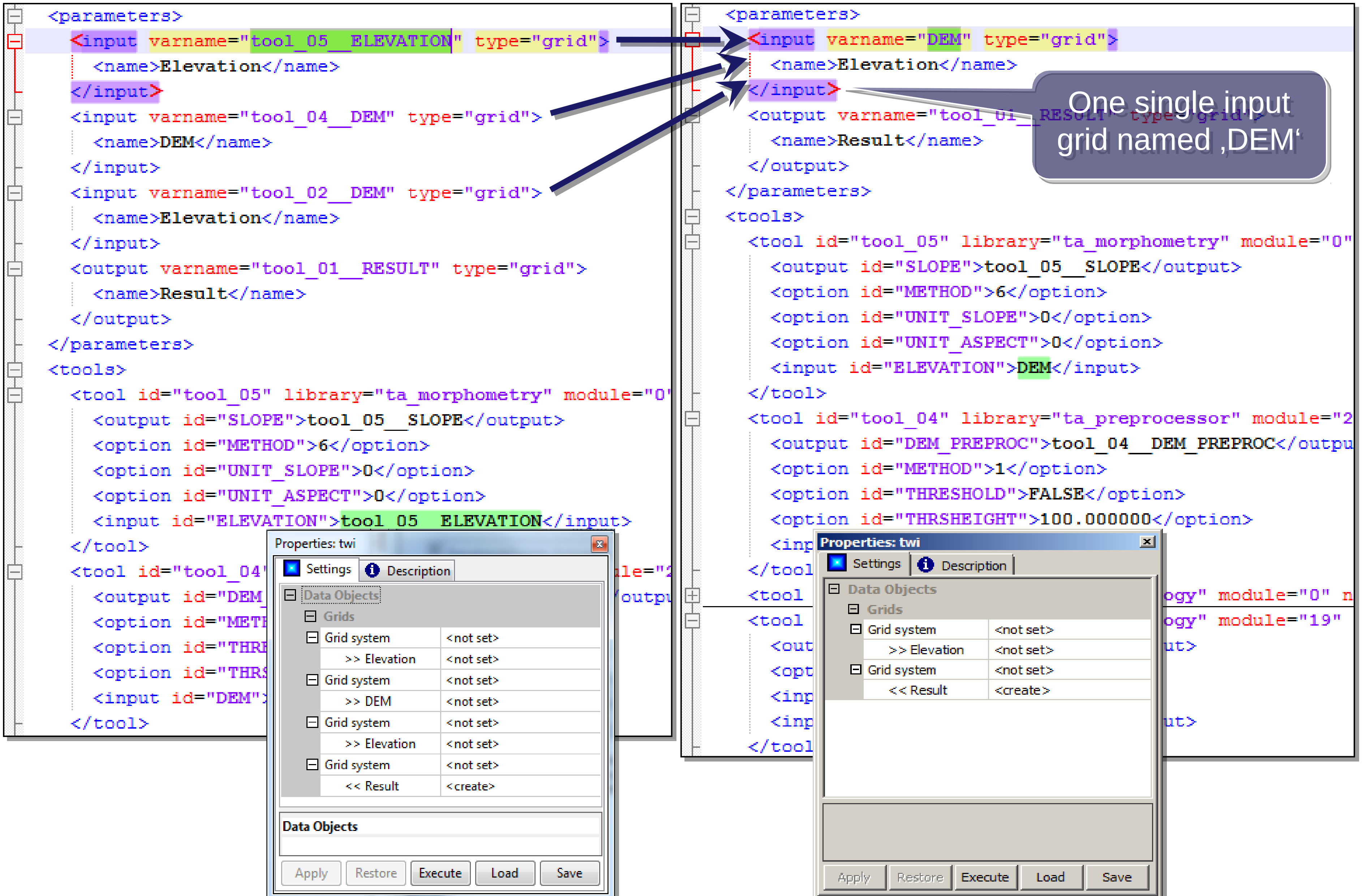
```
<tools>
  <tool library="ta_morphometry" module="0" name="Slope, Aspect, Curvature">
  <tool library="ta_preprocessor" module="2" name="Sink Removal">
  <tool library="ta_hydrology" module="0" name="Flow Accumulation (Top-Down)
  <tool library="ta_hydrology" module="19" name="Flow Width and Specific Cat
  <tool library="grid_calculus" module="1" name="Grid Calculator">
</tools>
```

## <tool>

- Attributes
  - library: tool library of the tool
  - module: tool identifier (unique!)
  - name: unused, good for better reading
- Content
  - <input>/<output>/<option>
    - Attributes
      - id: tool‘s parameter identifier
      - varname=true/false (options only, default=false)
    - Content
      - varname or value (options only)

```
<tool library="ta_morphometry" module="0" name="Slo
  <output id="SLOPE">tool_05__SLOPE</output>
  <input  id="ELEVATION">DEM</input>
  <option id="METHOD">6</option>
  <option id="UNIT_SLOPE">0</option>
  <option id="UNIT_ASPECT">0</option>
</tool>
```

# TWI | One Single Input instead of Three

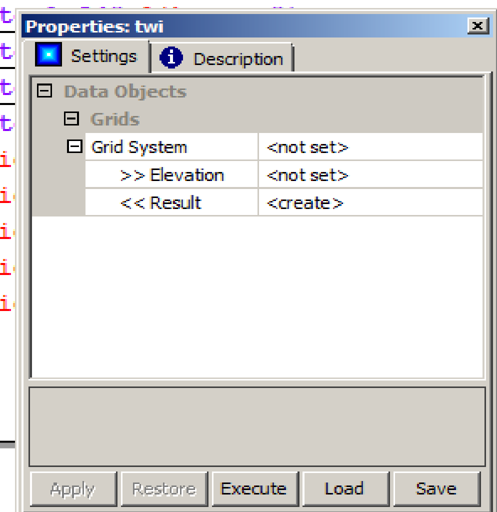


# TWI | Using One Single Grid System for Input and Output

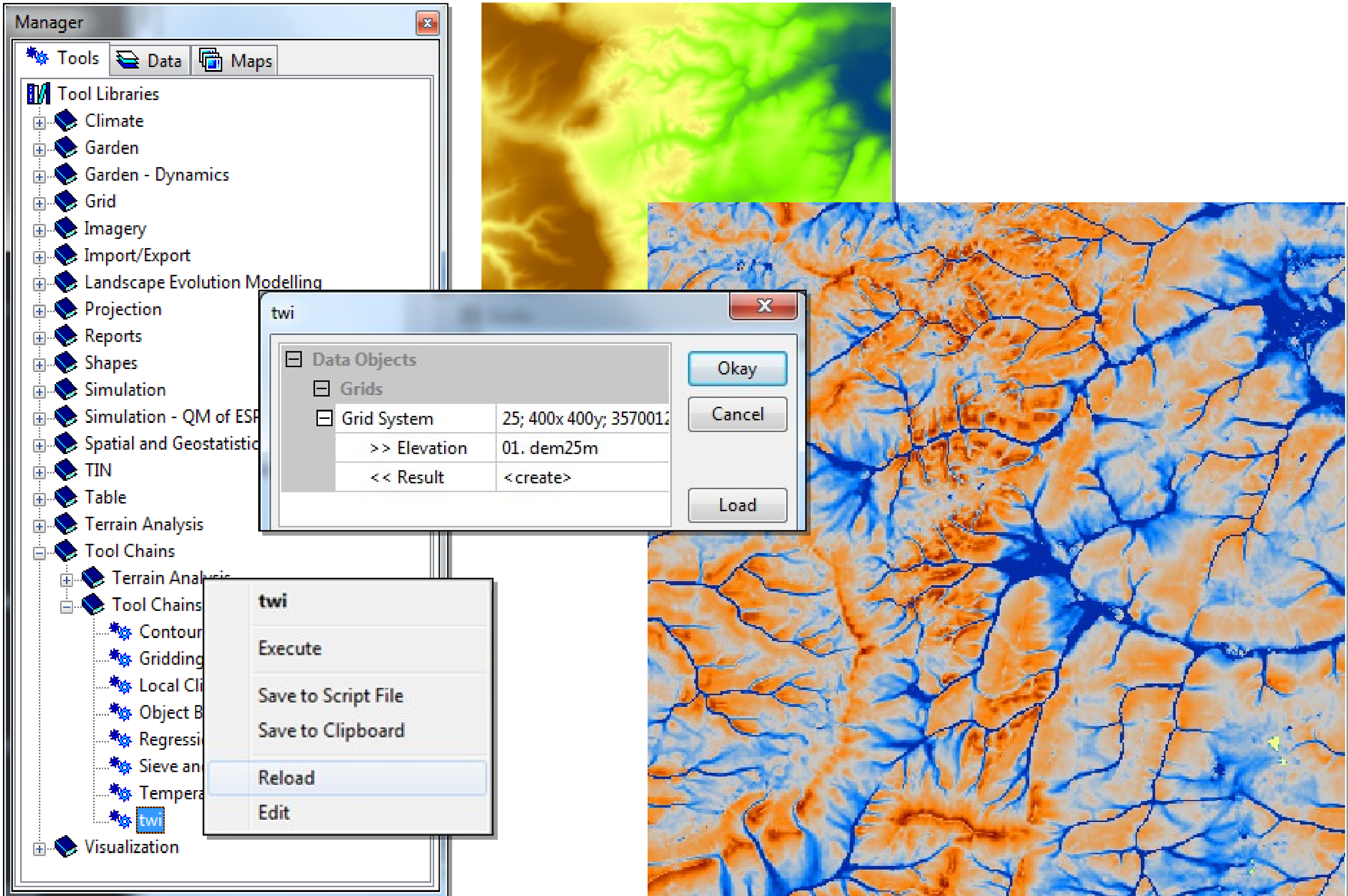
```xml
<parameters>
  <input varname="DEM" type="grid">
    <name>Elevation</name>
  </input>
  <output varname="tool_01__RESULT" type="grid">
    <name>Result</name>
  </output>
</parameters>
<tools>
  <tool id="tool_05" library="ta_morphometry" module="0"
    <output id="SLOPE">tool_05__SLOPE</output>
    <option id="METHOD">6</option>
    <option id="UNIT_SLOPE">0</option>
    <option id="UNIT_ASPECT">0</option>
    <input id="ELEVATION">DEM</input>
  </tool>
  <tool id="tool_04" library="ta_preprocessor" module="2
    <output id="DEM_PREPROC">tool_04__DEM_PREPROC</outpu
    <option id="METHOD">1</option>
    <option id="THRESHOLD">FALSE</option>
    <option id="THRSHEIGHT">100.000000</option>
    <input id="DEM">DEM</input>
  </tool>
  <tool id="tool_03" library="ta_hydrology" module="0" n
  <tool id="tool_02" library="ta_hydrology" module="19"
    <output id="SCA">tool_02__SCA</output>
    <option id="METHOD">2</option>
    <input id="DEM">DEM</input>
    <input id="TCA">tool_03__CAREA</input>
  </tool>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<toolchain saga-version="2.2.1">
  <group>toolchains</group>
  <identifier>twi</identifier>
  <name>twi</name>
  <description>created from history</description>
  <parameters>
    <option varname="GRID_SYSTEM" type="grid_system">
      <name>Grid System</name>
    </option>
    <input varname="DEM" type="grid" parent="GRID_SYSTEM">
      <name>Elevation</name>
    </input>
    <output varname="TWI" type="grid" parent="GRID_SYSTEM">
      <name>Result</name>
    </output>
  </parameters>
  <tools>
    <tool id="tool_05" library="ta_morphometry" module="0" na
    <tool id="t                                  r" module="2" n
    <tool id="t                                    module="0" name
    <tool id="t                                    module="19" nar
    <tool id="t                                    module="1" nar
      <output i
      <option i                                    option>
      <option i
      <input  i
      <input  i                                    >
    </tool>
  </tools>
</toolchain>
```

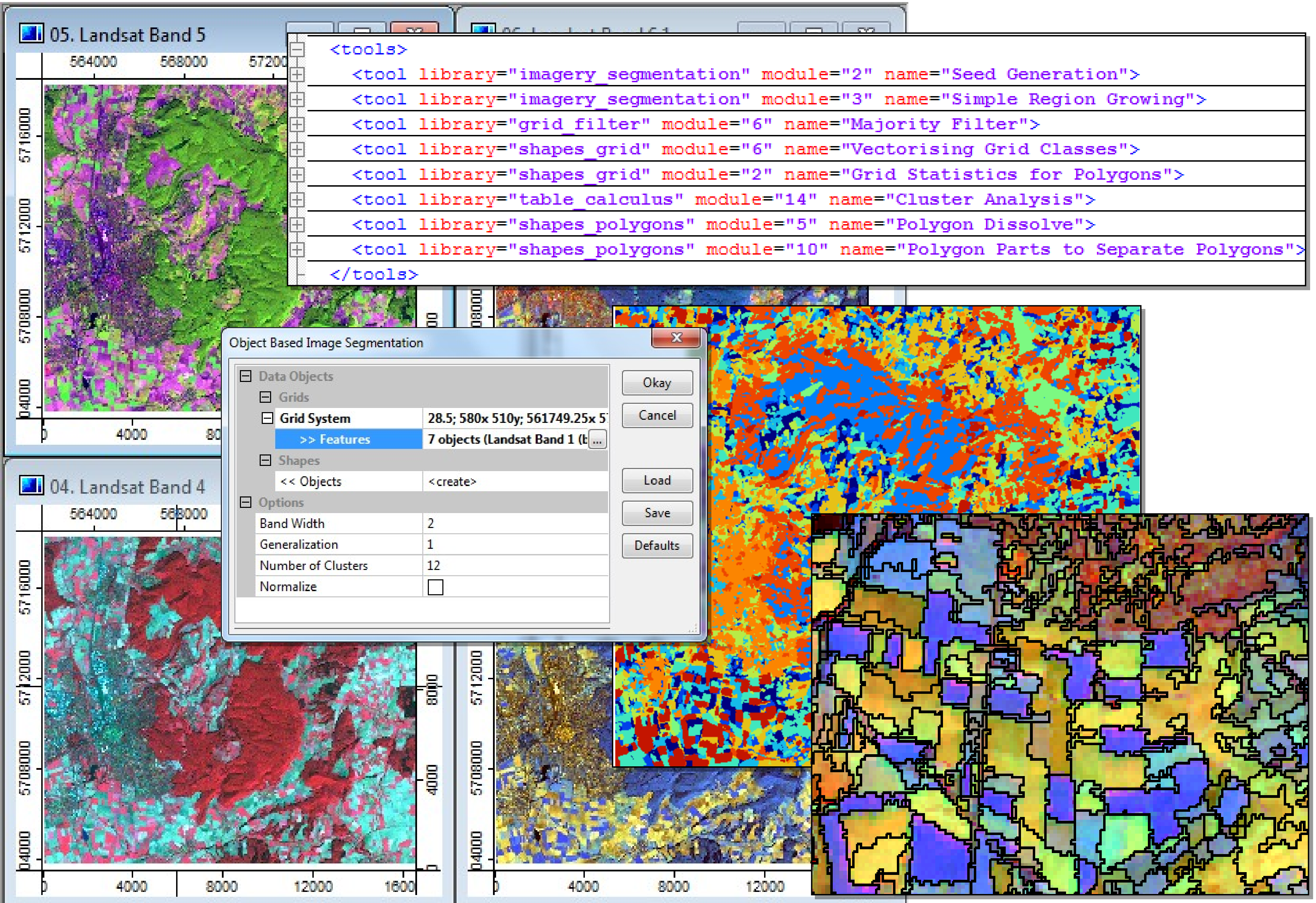
One single grid system as parent for all grids



# TWI | Reload Changed Tool Chain

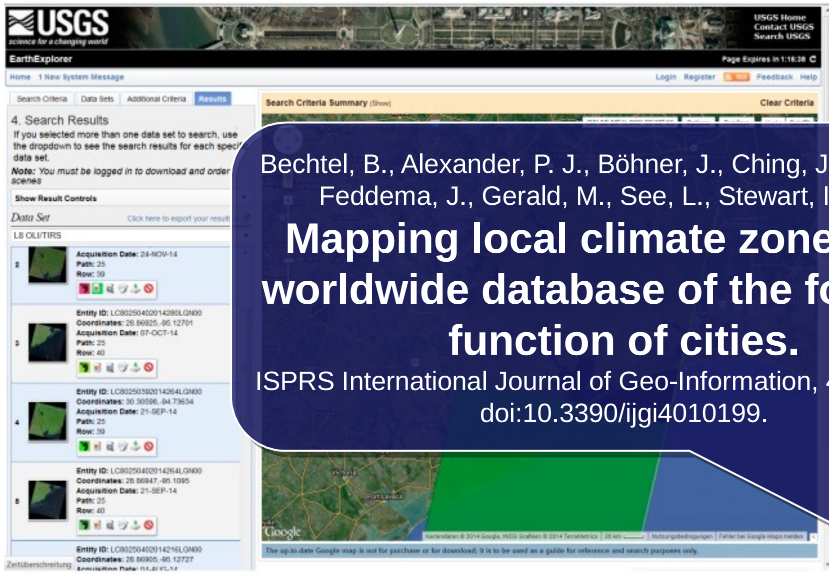


# OBIA | Object Based Image Analysis

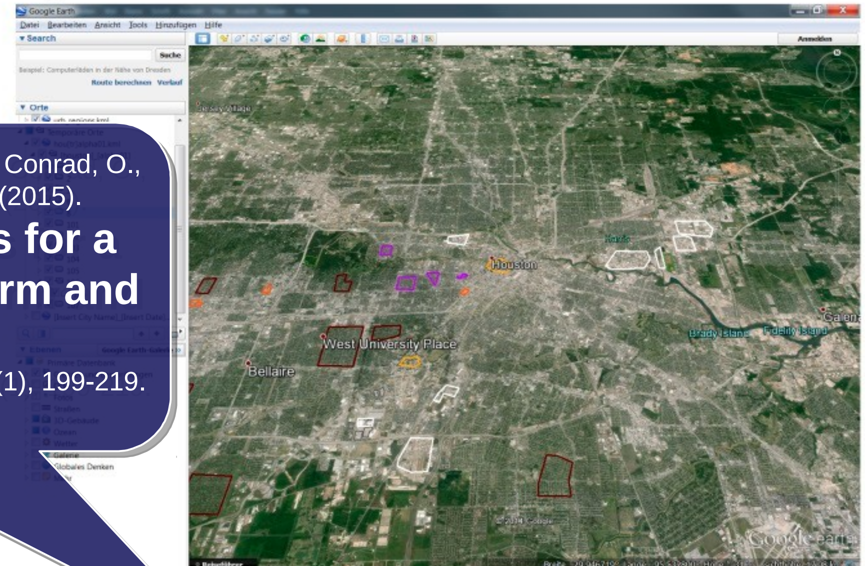


# LCZC | Local Climate Zone Classification

a)



b)

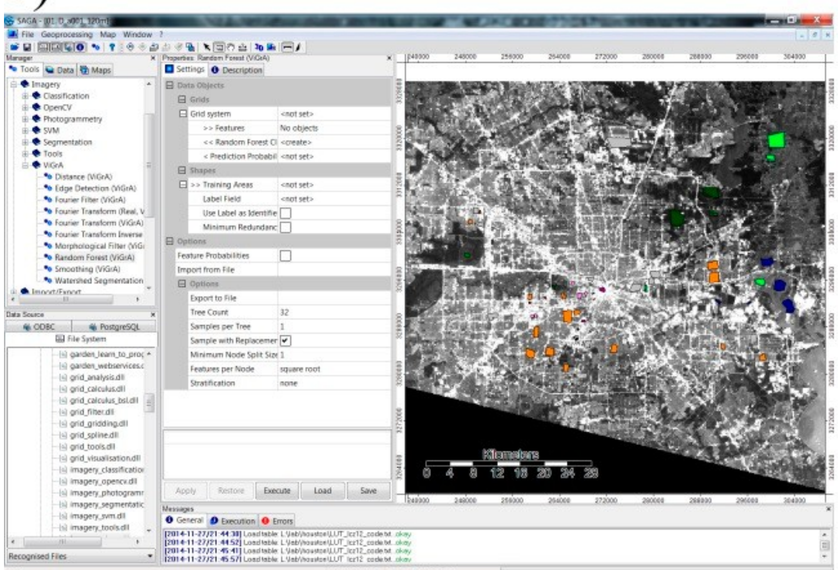


Bechtel, B., Alexander, P. J., Böhner, J., Ching, J., Conrad, O., Feddema, J., Gerald, M., See, L., Stewart, I. (2015).

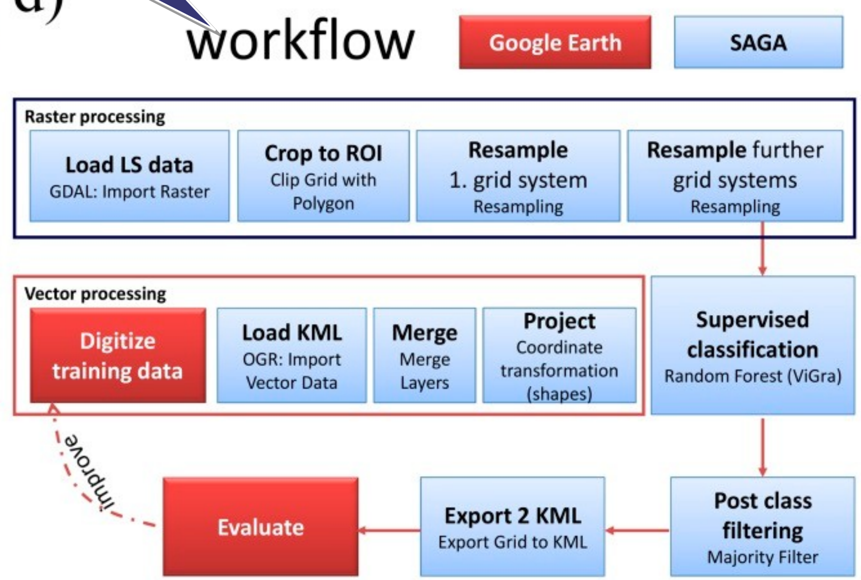
**Mapping local climate zones for a worldwide database of the form and function of cities.**

ISPRS International Journal of Geo-Information, 4(1), 199-219. doi:10.3390/ijgi4010199.

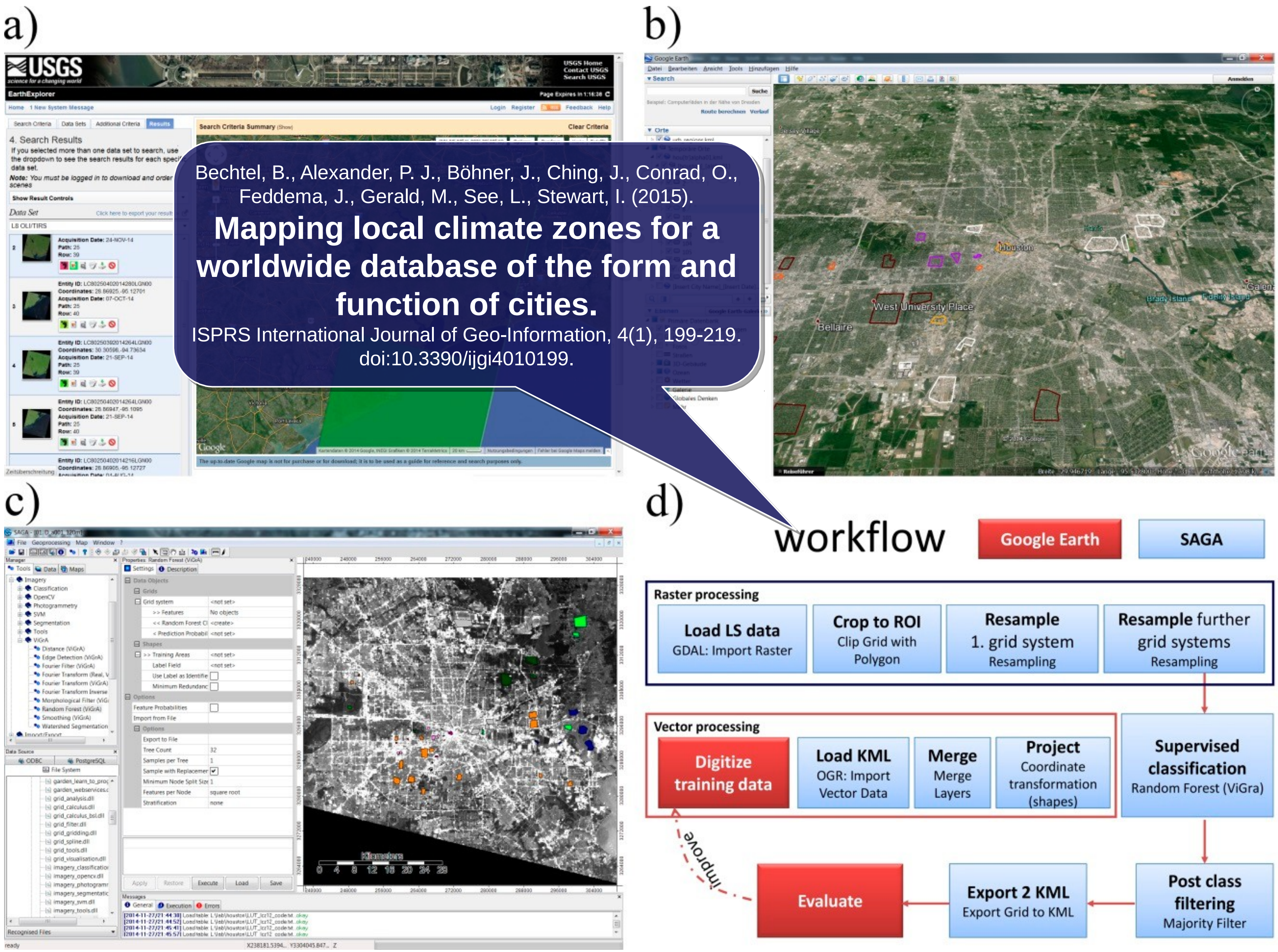
c)



d)



# LCZC | Local Climate Zone Classification



# Tool Chains | Conditional Tool Execution

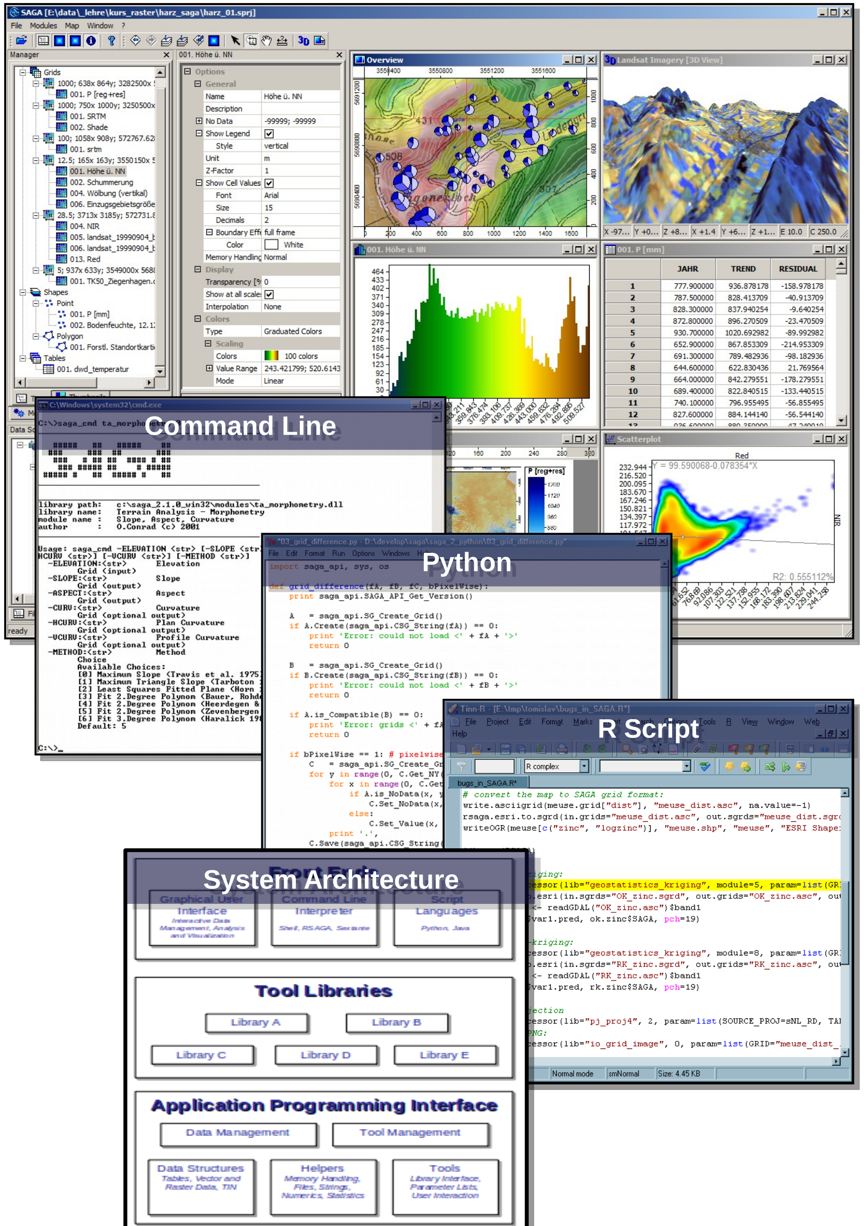
- <condition>
  - type
  - value
  - variable

Look-up table will be loaded and applied, if the user selected a file

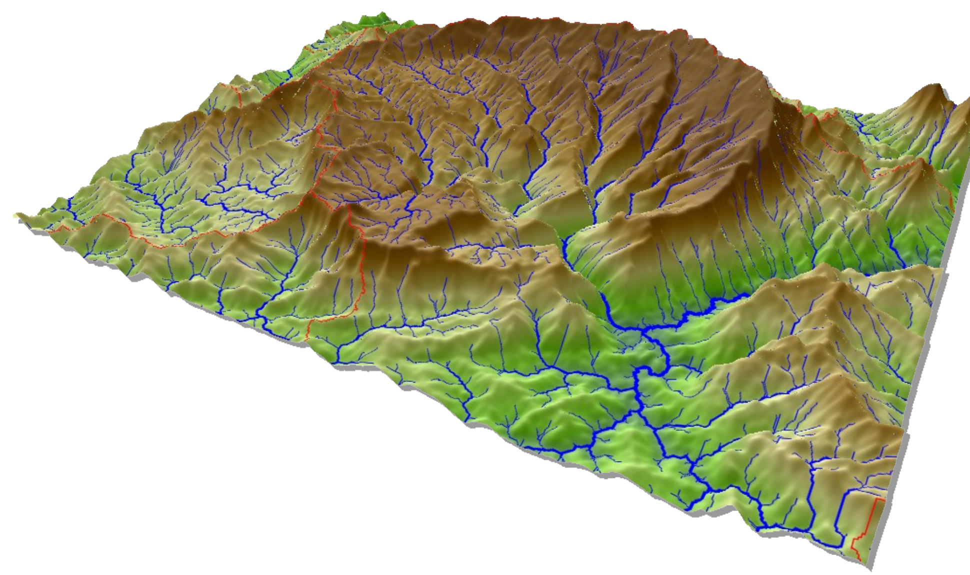
Majority filter will be applied, if the user chose to create the resulting grid

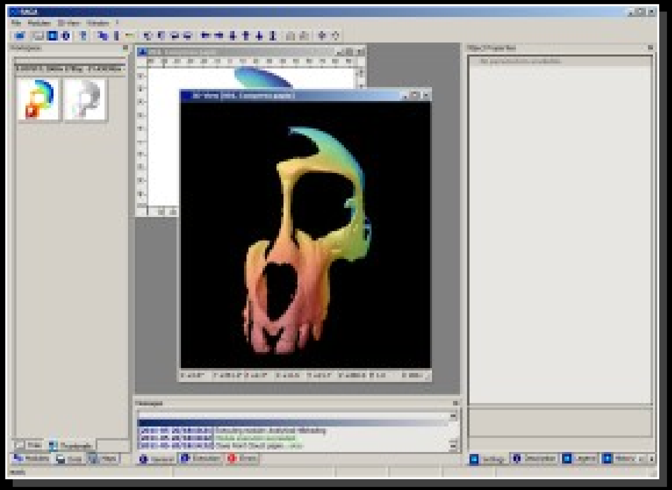
```
<tool library="io gdal" module="3" name="OGR: Import Vector Data">
<tool library="shapes tools" module="2" name="Merge Layers">
<tool library="grid tools" module="32" name="Select Grid from List">
<tool library="pj proj4" module="2" name="Coordinate Transformation (Shapes)">
<tool library="imagery vigra" module="9" name="Random Forest Classification (ViGrA)">
<condition type="not_equal" value="" variable="FILE_CLASS_DEF">
    <tool library="io table" module="1" name="Import Text Table">
    <tool library="grid visualisation" module="10" name="Select Look-up Table for Grid
</condition>
<condition type="not_equal" value="" variable="FILE_LCZC">
    <tool library="io grid image" module="2" name="Export Grid to KML">
</condition>
<condition type="exists" variable="LCZC_FILTERED">
    <tool library="grid filter" module="6" name="Majority Filter">
    <tool library="grid visualisation" module="10" name="Select Look-up Table for Grid
    <condition type="not_equal" value="" variable="FILE_LCZC_FILTERED">
        <tool library="io grid image" module="2" name="Export Grid to KML">
    </condition>
</condition>
```

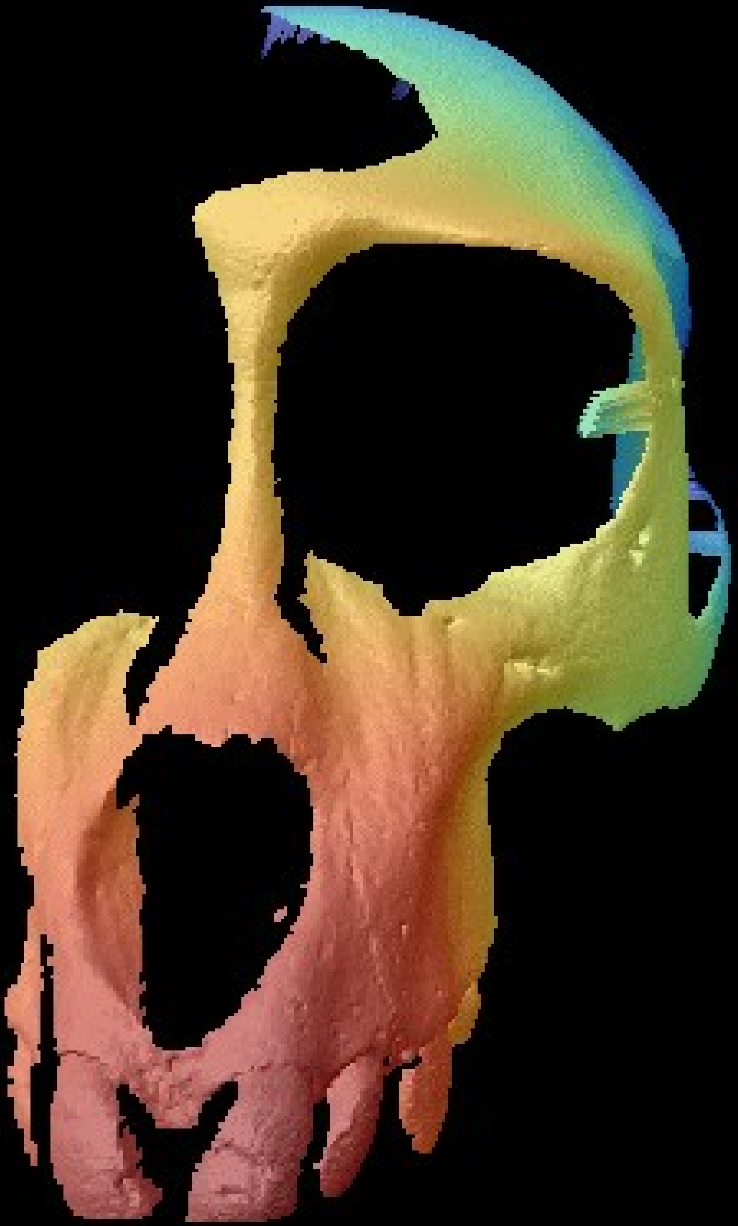
# SAGA | System for Automated Geoscientific Analyses




# Thank you for your attention



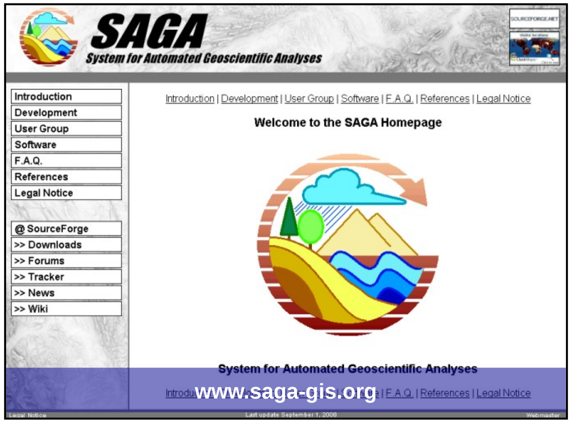




Many thanks
for your attention

**www.saga-gis.org**

# SAGA | Resources





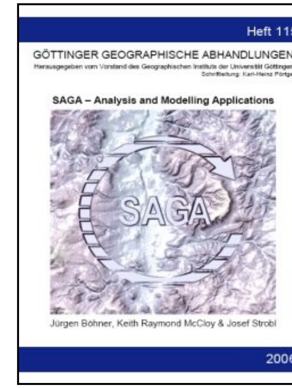

## Explore the world of SAGA GIS

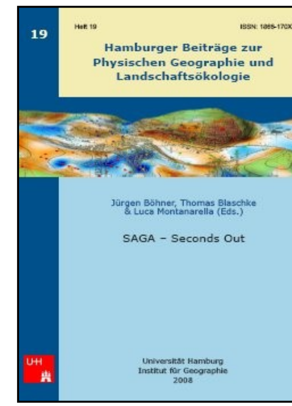
http://www.saga-gis.org

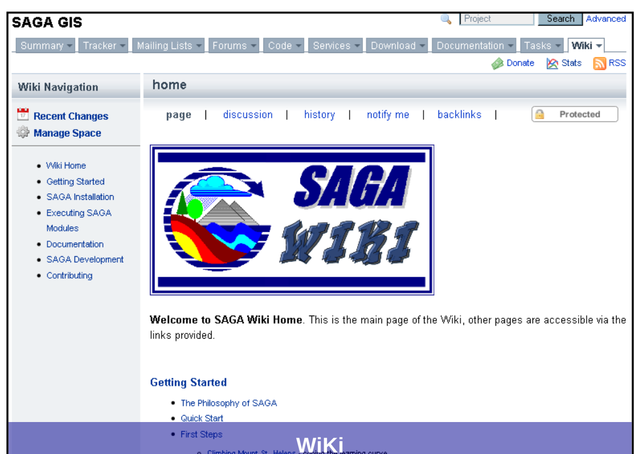
- Basic information
- Comprehensive list of references

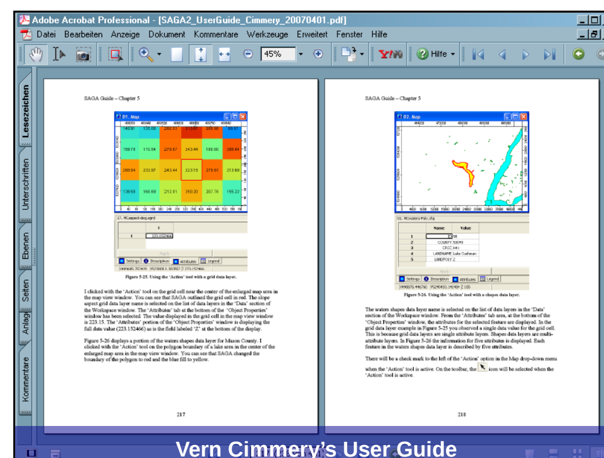
http://sourceforge.net/projects/saga-gis

- SourceForge > host for OSS projects
- Download software, documents, data
- SAGA Wiki
- Bug, Feature Tracker
- User Forum
- User Guide and Manual



# SAGA | Other Information Sources

## Marine Data Literacy

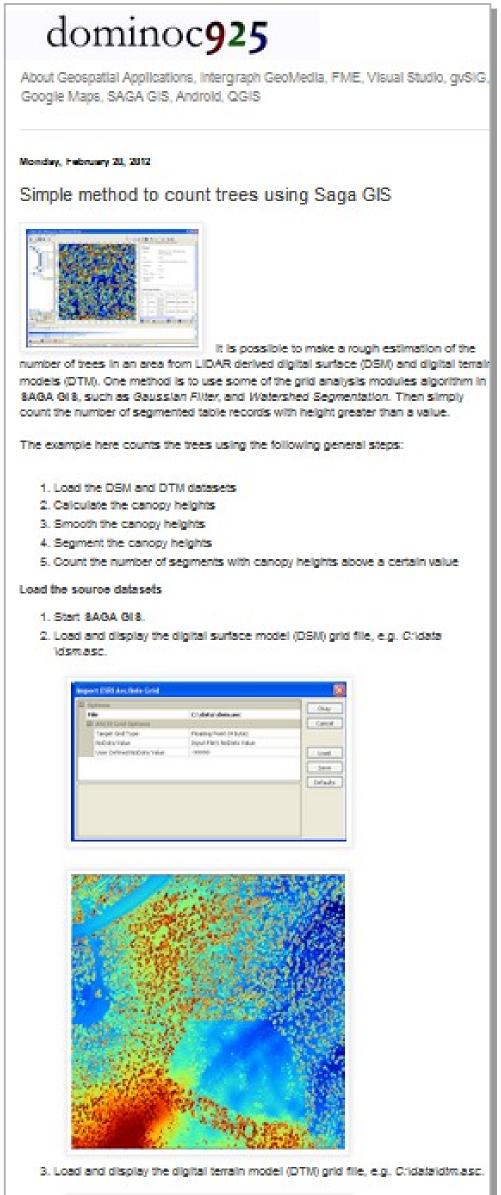
**marinedataliteracy.org**

The Marine Data Literacy Project is an attempt to bring together detailed, profusely illustrated instructions for many specific marine data management and analysis procedures, including basic GIS, ocean station data, satellite imagery, and operational data streams. The exercises are grouped according to an informal typology, but users are advised to simply browse through and see what's available. In general, the entire collection is constructed as a sequence of activities to build a "national marine data resource" for a selected area. Since 2010 this location is the area offshore the Ivory Coast. The exercises are currently used by the UNESCO/IOC/IOCE marine data training program, the Japan Foundation/POGO young scientists training program at Bermuda/BIOS, and the Ghent University-Erasmus Mundus masters degree program. SAGA is extensively used in all "marine GIS" lessons and in lessons dealing with grids, rasters and images. Contributing authors are always welcome, and an HTML exercise template is provided for their use. Intensive use of illustrations, and an absolute adherence to the step-by-step approach for all exercises are the only requisites.

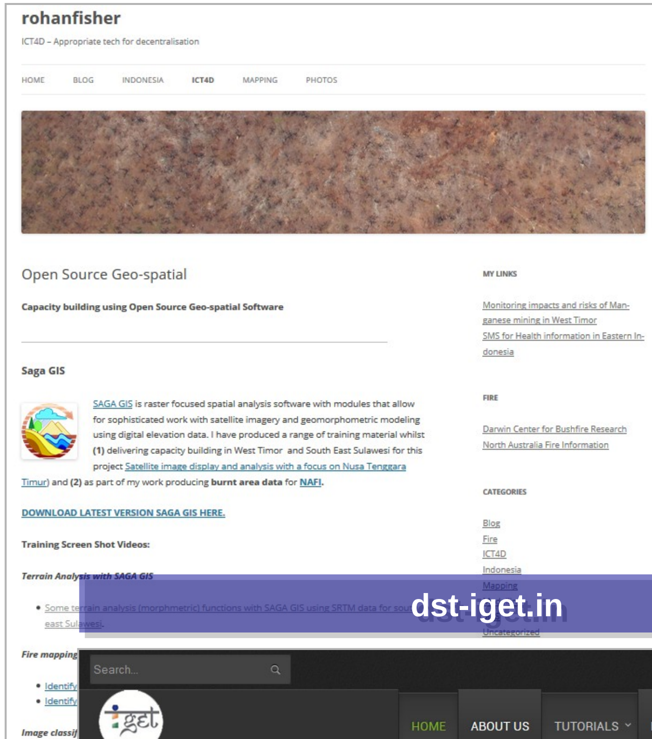
| | | | |
|---|---|---|---|
| **Saga** | This is the general-purpose, "workhorse" program we recommend to all data management students, even if they also use other commercial or public domain GIS solutions. One shortcoming is the minimal documentation. | General information:<br>• Saga Homepage<br>• Saga Forum on Sourceforge<br><br>Saga installation files.<br>• Saga Files on Sourceforge<br>• The ZIP version (not the unexplained EXE setup version) should be copied to a convenient location and unzipped to C:\<br>• Run by clicking on **saga_gui.exe** | • Windows 32 or 64. Create a shortcut to the executable **saga_gui.exe** to run the program<br>• Saga's Tutorials Collection<br>• Australia-Indonesia Training in Saga for Resource Management with Imagery<br>• Rohan Fisher's Saga Tutorials (in English and Indonesian)<br>• 1.3 Running 32-Bit Saga on a Mac with WINE - Provided by a student<br>• Saga Wiki on Sourceforge for Linux information<br>• "Mac users might like to hear that efforts are going on to make SAGA work on MacOS more smoothly. You find a thread regarding the MacOS port in the SAGA User Forum at http://sourceforge.net/p/saga-gis/discussion/790705/thread/b11de126/ Have a look at http://www.wxwidgets.org for background information about the wxWidgets project."<br>• DOMINOC925 - An amazingly good set of illustrated tutorials for Saga and other geospatial software; possibly hundreds of exercises, but not indexed -- use search function to find Saga examples |

# SAGA | More Sources of Information

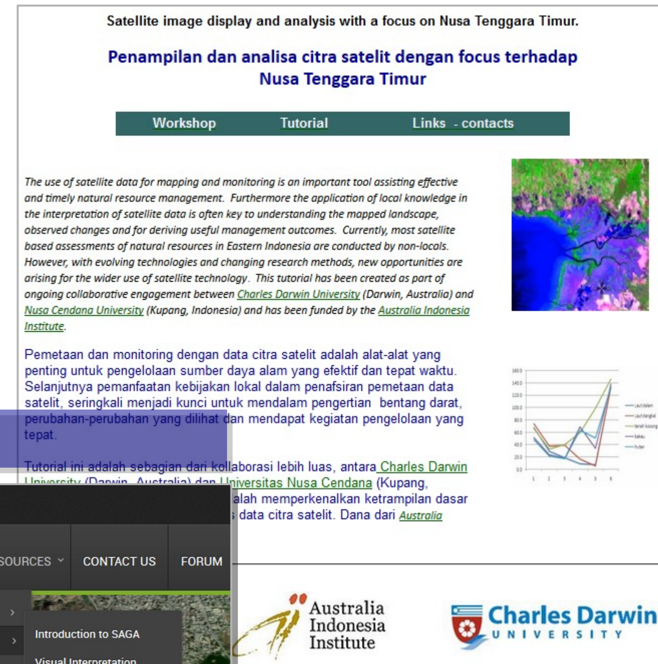
**dominoc925.blogspot.com**




**rohanfisher.wordpress.com/ open-source-geo-spatial**




**www.cdu.edu.au/itl/All-RS/**




**dst-iget.in**