Automating your analysis using SAGA GIS

Johan Van de Wauw

System for Automated Geoscientific Analyses

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

- CARBIOCLICAL – 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.
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…
Front Ends | Graphical User Interface

- Manager
- Map View
- Histogram
- Print Layout
- Properties
- Notifications
- Attributes
- Scatterplot
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.
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.
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
- Via QGIS Processing

http://www.zoo-project.org/docs/kernel/sagagis.html
**Supported Platforms**

- MS Windows
- Linux
- FreeBSD
  - Maintainer: Rainer Hurling, NW-FVA
- Mac OS X
  - Maintainer: Wanted!
"I spend a lot of time on this task. I should write a program automating it!"

**Theory:**
- **Work on original task**
- **Writing code**
- **Automation takes over**
- **Free time**

**Reality:**
- **Writing code**
- **Debugging**
- **Rethinking**
- **Ongoing development**
- **No time for original task anymore**

Source: xkcd (http://imgs.xkcd.com/comics/automation.png)
Changes and New Features in the GUI

SAGA User Group Association

FOSDEM 2016

SAGA 2.2.3 | News & Outlook

### Manager

Load Tool Library
Find and Run Tool

### Run Tool

#### Options

**Search for...**
- Name
- Description
- Case Sensitive

#### Search Result: 'slope'

- 05: Direct neighbours - slope and aspect
  - Aspect-Slope Grid
  - DTM Filter (slope-based)
  - Downslope Area [interactive]
  - Downslope Distance Gradient
  - Relative Heights and Slope Positions
  - Slope Length
  - Slope Limited Flow Accumulation
  - Slope, Aspect, Curvature

#### Manager

- Tree
- Thumbnails
- Maps
  - 01. Altitude
  - 01. Channel Network
  - 01. Altitude

### Map

- 01. Altitude
- 09. Classification
- 03. Catchment Area
- 04. Landsat Band 4
- 06. Landsat Band 6.1

### Map Features

- Meters
  - 0
  - 1600
  - 3200
  - 4800
  - 6400
  - 8000
  - 9500

- Coordinate System
  - X560759.67417
  - Y5715226.856

### Ready

- Graticule

- Apply
- Restore
- Load
- Save
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

```
a.wmo_id=b.wmo_id AND a.wmo_mod=b.wmo_mod AND b.year=1950
```
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

3D Globe Viewer

TIN Viewer

3D Shapes 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
Ways of Automation with SAGA

System for Automated Geoscientific Analyses

Michael Bock, Olaf Conrad, Volker Wichmann
TWI | Topographic Wetness Index

- TWI = \log(\frac{A_s}{\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
Sink Removal → Slope → Flow Accumulation → $\ln(A_s / \tan(S))$ → Flow Accumulation

$A_s$
TWI | Apply Tool Chain to Another Data Set

The image shows a user interface for managing and executing geoscientific tool chains. The interface includes a tool chain manager with options for different tool libraries such as Climate, Garden, Grid, Imagery, Import/Export, Landscape Evolution Modelling, Projection, Reports, Shapes, Simulation, Simulation - QM of ESP, Spatial and Geostatistics, TIN, Table, Terrain Analysis, Tool Chains, Terrain Analysis Tool Chains, Contour Lines from Points, Gridding of Points, Local Climate Zone Classification, Object Based Image Segmentation, Regression Kriging, Sieve and Clump, Temperature Downscaling, and TWI. There are also sections for data objects, settings, and properties with options for grids and results. The interface appears to be part of a software tool for automated geoscientific analyses, specifically SAGA Ways of Automation, as indicated by the text at the top of the page.
TWI | Edit Tool Chain

SAGA Ways of Automation

Manager

- Tool Libraries
  - Climate
  - Garden
  - Garden - Dynamics
  - Grid
  - Imagery
  - Import/Export
  - Landscape Evolution Modelling
  - Projection
  - Reports
  - Shapes
  - Simulation
  - Simulation - QM of ESP
  - Spatial and Geostatistics
  - TIN
  - Table
  - Terrain Analysis
  - Tool Chains
    - Terrain Analysis
    - Tool Chains
      - Tool Chain
        - Contouring
        - Griding
        - Local GIS
        - Objects
        - Regressions
        - Sieve analysis
        - Temperature
        - TWI
        - Visualization

Tools

- Edit
- Execute
- Save to Script File
- Save to Clipboard
- Reload
- TWI

<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>
      <input varname="tool_05__ELRVATION" type="grid">
        <name>Elevation</name>
      </input>
      <input varname="tool_04__DEM" type="grid">
        <name>DEM</name>
      </input>
      <input varname="tool_02__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" name="Slope, As">
        <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">tool_03__ELRVATION</input>
      </tool>
      <tool id="tool_04" library="ta_preprocessor" module="2" name="Sink Res">
        <output id="DEM_PREPROC">tool_04__DEM_PREPROC</output>
        <option id="METHOD">1</option>
        <option id="THRESHOLD">FALSE</option>
        <option id="THRESHRIGHT">100.000000</option>
    </tools>
</xml>
### Tool Chains | Main Structure

**<toolchain>**

- **Attributes**
  - saga-version

- **Content**
  - `<group>`
  - `<identifier>`
  - `<name>`
  - `<author>`
  - `<description>`
  - `<menu>`
  - `<parameters>`
  - `<tools>`

```xml
<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">
          <tools>
            <tool id="tool_05" library="ta_morphometry" module="0" name=
            <tool id="tool_04" library="ta_preprocessor" module="2" name=
            <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>
  </parameters>
</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>

<parameters>

<option varname="GRID_SYSTEM" type="grid_system">
    <name>Grid System</name>
</option>

<option varname="FILE_TRAINING" type="file">
    <name>Training Areas</name>
    <filter>RML/RMZ Files|*.xml;*.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>RML 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>RML 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" parent="GRID_SYSTEM">
    <name>LCZC (Filtered)</name>
</output>
</parameters>
<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

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>grid</td>
<td></td>
</tr>
<tr>
<td>table</td>
<td></td>
</tr>
<tr>
<td>shapes</td>
<td></td>
</tr>
<tr>
<td>tin</td>
<td></td>
</tr>
<tr>
<td>points</td>
<td></td>
</tr>
<tr>
<td>grid_list</td>
<td></td>
</tr>
<tr>
<td>table_list</td>
<td></td>
</tr>
<tr>
<td>shapes_list</td>
<td></td>
</tr>
<tr>
<td>tin_list</td>
<td></td>
</tr>
<tr>
<td>points_list</td>
<td></td>
</tr>
</tbody>
</table>
<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

---

<table>
<thead>
<tr>
<th>Type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>double</td>
<td></td>
</tr>
<tr>
<td>degree</td>
<td></td>
</tr>
<tr>
<td>range</td>
<td></td>
</tr>
<tr>
<td>choice</td>
<td></td>
</tr>
<tr>
<td>text</td>
<td></td>
</tr>
<tr>
<td>long_text</td>
<td></td>
</tr>
<tr>
<td>file</td>
<td></td>
</tr>
<tr>
<td>table_field</td>
<td></td>
</tr>
<tr>
<td>table_fields</td>
<td></td>
</tr>
<tr>
<td>grid_system</td>
<td></td>
</tr>
</tbody>
</table>
Tool Chains | The Tools Section

<tools>
  ● Content
    ● <tool>
    ● <condition>
  ● <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)
**TWI | One Single Input instead of Three**

```xml
<parameters>
  <input varname="tool_05_ELEVATION" type="grid">
    <name>Elevation</name>
  </input>
  <input varname="DEM" type="grid">
    <name>Elevation</name>
  </input>
  <input varname="tool_02_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">
    <input 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" tool_05_ELEVATION output=
  </tool>
  <tool id="tool_04" library="ta_preprocessor" module="2">
    <input id="DEM_PREPROC" tool_04DEM_PREPROC output=
    <option id="METHOD" 1</option>
    <option id="UNIT_SLOPE" 0</option>
    <option id="UNIT_ASPECT" 0</option>
    <option id="THRESHHEIGHT" 100.000000 output=
  </tool>
</tools>
```

One single input grid named 'DEM'
TWI | Using One Single Grid System for Input and Output

One single grid system as parent for all grids
TWI | Reload Changed Tool Chain
OBIA | Object Based Image Analysis

<tool library="imagery_segmentation" module="2" name="Seed Generation">
<tool library="imagery_segmentation" module="3" name="Simple Region Growing">
<tool library="grid_filter" module="6" name="Majority Filter">
<tool library="shapes_grid" module="6" name="Vectorising Grid Classes">
<tool library="shapes_grid" module="2" name="Grid Statistics for Polygons">
<tool library="table_calculus" module="14" name="Cluster Analysis">
<tool library="shapes_polygons" module="5" name="Polygon Dissolve">
<tool library="shapes_polygons" module="10" name="Polygon Parts to Separate Polygons">
</tool>

Object Based Image Segmentation

- Data Objects
  - Grid System
    - Features: 28.5; 580x510y; 561749.25x5'
  - Shapes
    - Objects: 7 objects (Landsat Band 1)

- Options
  - Band Width: 2
  - Generalization: 1
  - Number of Clusters: 12
  - Normalize: [ ]
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

```xml
<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>
</tool library="grid filter">```
```
Thank you for your attention
Many thanks for your attention

www.saga-gis.org
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
### Marine Data Literacy

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

<table>
<thead>
<tr>
<th>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.</th>
<th>General information:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <a href="#">Saga Homepage</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Saga Forum on Sourceforge</a></td>
</tr>
<tr>
<td></td>
<td>Saga installation files.</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Saga Files on Sourceforge</a></td>
</tr>
<tr>
<td></td>
<td>• The ZIP version (not the unexplained EXE setup version) should be copied to a convenient location and unzipped to C:\</td>
</tr>
<tr>
<td></td>
<td>• Run by clicking on <code>saga_gui.exe</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Windows 32 or 64. Create a shortcut to the executable <code>saga_gui.exe</code> to run the program</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Saga's Tutorials Collection</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Australia-Indonesia Training in Saga for Resource Management with Imagery</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Rohan Fisher's Saga Tutorials</a> (in English and Indonesian)</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">1.3 Running 32-Bit Saga on a Mac with WINE</a> - Provided by a student</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">Saga Wiki on Sourceforge</a> for Linux information</td>
</tr>
<tr>
<td></td>
<td>“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 <a href="http://sourceforge.net/p/saga-gis/discussion/790705/thread/b11de126/">http://sourceforge.net/p/saga-gis/discussion/790705/thread/b11de126/</a> Have a look at <a href="http://www.wxwidgets.org">http://www.wxwidgets.org</a> for background information about the wxWidgets project.”</td>
</tr>
<tr>
<td></td>
<td>• <a href="#">DOMINOC925</a> - 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</td>
</tr>
</tbody>
</table>
Simple method to count trees using Saga GS

1. Load the DSM and DTN datasets
2. Calculate the canopy heights
3. Smooth the canopy heights
4. Segment the canopy heights
5. Count the number of segments with canopy heights above a certain value

Load the source datasets:
1. Start Saga GIS
2. Load and display the digital surface model (DSM) grid file, e.g., c:rtdata/dsm.asc
3. Load and display the digital terrain model (DTM) grid file, e.g., c:rtdata/dtm.asc

Satellite Image display and analysis with a focus on Nusa Tenggara Timur.
Penelitian dan analisis citra satelit dengan focus terhadap Nusa Tenggara Timur

The use of satellite data for mapping and monitoring is an important tool assisting effective and timely natural resource management. Furthermore, the interpretation of these data is often key to understanding the mapped landscape, observed changes, and for deriving useful management outcomes. Currently, most satellite based statements of natural resources in Eastern Indonesia are conducted by non-local. However, with evolving technologies and changing research methods, new opportunities are arising for the wider use of satellite technology. This tutorial has been created as part of ongoing collaborative engagement between Charles Darwin University (Darwin, Australia) and Nusa Cendrawasih University (Jayapura, Indonesia) and has been funded by the Australia Indonesia Institute.

Pemetaan dan monitoring dengan data citra satelit adalah alat-alat yang penting untuk pengelolaan sumber daya alam yang efektif dan tepat waktu. Salahsatu pemetaan lahan lahat dalam penanaman pemetaan data satelit, termasuk untuk pendeteksi status kondisi lahan, pemanfaatan penggunaan lahan dan perubahan-perubahan yang ditemukan dan memudahkan pengambilan keputusan yang tepat.

Tutorial ini adalah bagian dari Kolaborasi lebih luas, antara Charles Darwin University (Darwin, Australia) dengan Universitas Nusa Cendrawasih (Jayapura, Indonesia), yang memperluas analisis data citra satelit. Dana dari Australia Indonesia Institute.