

Fort Collins Science Center

User Documentation for the Software for Assisted Habitat Modeling (SAHM) package in VisTrails.

By Colin Talbert, Drew Ignizio, Nick Young

Report Series XXXX–XXXX

U.S. Department of the Interior

U.S. Geological Survey

U.S. Department of the Interior

KEN SALAZAR, Secretary

U.S. Geological Survey

Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia: 201x  
Revised and reprinted: 201x

For more information on the USGS—the Federal source for science about the Earth,  
its natural and living resources, natural hazards, and the environment—visit   
<http://www.usgs.gov> or call 1–888–ASK–USGS

For an overview of USGS information products, including maps, imagery, and publications,  
visit <http://www.usgs.gov/pubprod>

To order this and other USGS information products, visit <http://store.usgs.gov>

Suggested citation:  
Author1, F.N., Author2, Firstname, 2001, Title of the publication: Place of publication   
(unless it is a corporate entity), Publisher, number or volume, page numbers; information   
on how to obtain if it’s not from the group above.

Any use of trade, product, or firm names is for descriptive purposes only and does not imply   
endorsement by the U.S. Government.

Although this report is in the public domain, permission must be secured from the individual   
copyright owners to reproduce any copyrighted material contained within this report.

Contents

Installing SAHM and VisTrails 6

Simple installation 6

Supported Operating Systems 6

Setting up 6

Pointing to location of Data and Dependencies 6

Expert installation 6

Configuration 6

Using SAHM and VisTrails 6

Basic Opperation 6

The Session Folder 6

Data Inputs 6

Field Data 6

Template Layer 6

Covariates 6

Data Preparation 6

Field Data Query 6

Field Data Aggregate and Weight 6

Project Aggregate Resample Clip (PARC) 6

Raster Format Converter 6

Projection Layers 6

Data Manipulation 6

Merged Dataset Builder 6

Test Training Split 6

Covariate Correlation and Selection 6

Modeling 6

GLM 6

MARS 6

BoostedRegressionTree 6

RandomForest 6

Maxent 6

Apply Model 6

Viewing Output 6

Spreadsheet 6

SAHM Model Output Viewer Cell 6

SAHM Spatial Output Viewer Cell 6

MISC 6

Spreadsheet 6

References Cited 6

Glossary 6

Figures

1. Figure caption. 6

2. Figure caption with (*A*) multiple parts (*B*) that are divided by the (*C*) MultipartFigCap style. 6

Tables

1. This is a table title. 6

User Documentation for the Software for Automated Assisted Habitat Modeling (SAHM) package in VisTrails.

By Colin Talbert, Drew Ignizio, Nick Young

# Introduction

The Software for Assisted Habitat Modeling (SAHM) has been created to both expedite habitat modeling and help maintain a record of the various input data, pre- and post- processing steps and modeling options incorporated in the construction of a species distribution model. The four main advantages to using the combined VisTrail:SAHM package for species distribution modeling are:

1. formalization and tractable recording of the entire modeling process
2. easier collaboration through a common modeling framework
3. a user-friendly graphical interface to manage file input, model runs, and output
4. extensibility to incorporate future and additional modeling routines and tools.

This user guide provides detailed instructions on how to install the SAHM package within VisTrails and then presents information on the use of the package. A step-by-step tutorial to create cheatgrass habitat suitability maps for Rocky Mountain National Park, USA, is provided in appendix A.

**Installing SAHM and VisTrails**

**Windows Installation**

The following detailed steps can be used to install and setup the VisTrails and the SAHM package on Windows. The various components are all freely available but require separate downloads and installations. These include VisTrails 2.0, R (developed under version 2.14.0), Maxent (developed under version 3.3.3k), GDAL (developed under version 1.8.0), and the GDAL Python bindings for Python 2.7. While not terribly difficult each component must be installed correctly and configured for the individual computer. The most difficult step in this process is getting the GDAL Python bindings installed into the Python installation that comes with VisTrails. Previous installations of any of the above components or PyQt might cause compatibility problems with VisTrails SAHM. These issues can be difficult to troubleshoot and fix. Particular care should be taken to ensure that the versions of each component are compatible. For example if you install the 64-bit version of VisTrails you should also install the 64-bit version of the GDAL Python bindings.

**You must have Administrative Rights on the computer you are trying to install on.**

**1) First obtain and install VisTrails,** which is available from <http://www.vistrails.org/index.php/Downloads>. Choose the most current release which is currently 2.0-alpha. The bundle is available for both 32-bit and 64-bit machines (to view the specs of a computer to determine which version you should get, go to Control Panel>System and Security>System). Install this application on your system using the provided installers. You will need administrative rights to complete this installation. Unlike some installations which will prompt a user to login as an administrator during the installation, a user should be logged in with admin privileges before attempting the installation, or else it will fail. Complete instructions and documentation are available from their site. Note that for the purpose of these instructions I installed my VisTrails to C:\VisTrails so subsequent instructions will be relative to this location.

**2) Next download and install GDAL.**

32-bit and 64 bit versions are available here:

http://www.gisinternals.com/sdk/

A user should choose the 32-bit or 64-bit version to match their system architecture. We tested these instructions with the 'release-1600' version.

Specifically we used “Generic Installer for the GDAL Core Components”:

<http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-x64-gdal-1-8-mapserver-6-0\gdal-18-1600-x64-core.msi> for 64-bit architecture

and

<http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-gdal-1-8-mapserver-6-0\gdal-18-1600-core.msi> for 32-bit architecture.

When it finishes downloading double click the msi and choose the 'complete installation' option.

The final step for GDAL is to add its location to your Path variable and add a couple of required Environmental (System) Variables to your system. On your machine’s system, navigate to Control Panel > System and Security > Advanced System Settings (on the left) > Environmental Variables



Click on the Environment Variables tab and add a new system variable called 'GDAL\_DATA' that points to the 'gdal-data' folder within the GDAL install location. Note this location will be specific to where you installed GDAL and might be C:\Program Files (x86)\GDAL\gdal-data if you installed the 32-bit version on a 64-bit machine.



Next add a new system variable called 'PROJ\_LIB' that points to the 'projlib' folder in the GDAL install location:



Now click on and edit the 'Path' variable. Append the path to the root GDAL folder to the variable value for the 'Path' variable. Include a semicolon at the end, with no spaces, 

when done modifying the 'Path' System Variable:



**3) Next download and install the GDAL bindings for Python.**

These are available from the same location we obtained the GDAL core from, <http://www.gisinternals.com/sdk/> Once again match the 32-bit or 64-bit architecture of your system and be sure to grab the version for Python 2.7.

Specifically we used “Installer for the GDAL python bindings (requires to install the GDAL core)”:

<http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-x64-gdal-1-8-mapserver-6-0\GDAL-1.8.1.win-amd64-py2.7.msi> for 64-bit architecture

and

<http://www.gisinternals.com/sdk/Download.aspx?file=release-1600-gdal-1-8-mapserver-6-0\GDAL-1.8.1.win32-py2.7.msi> for 32-bit architecture.

Once it finishes downloading double click to install. For the installation of these bindings you must choose ‘Python from another location’ and then point to the python installation used by VisTrails. In our case I used C:\VisTrails\vistrails\Python27\_64 Note that if you've installed the 32-bit version of VisTrails this will be C:\VisTrails\vistrails\Python27 When providing the location after specifying "Python from another location," a user should be providing the location in which the python.exe program resides. We want to provide the location where the bindings should be installed, rather than pointing to the full path of the actual python.exe file.



The final thing to do is copy the DLL files from the GDAL install location to the VisTrails folder in the VisTrails installation.

-In GDAL install location (C:\Program Files\GDAL), sort all files by type. Select all the DLL files (these might be labeled 'Application extension') and copy them. Paste the files here: C:\VisTrails\vistrails



**4) Once VisTrails is installed, download the SAHM package.**

This is available from <http://www.fort.usgs.gov/xxxxxxxxxxxxxxxxxx>. Unzip this into the VisTrails packages directory which is at C:\VisTrails\vistrails\packages.

**5) Download and install the external software R**

R is available from <http://www.r-project.org/>. If you have a previously installed version of R you might be OK with it but we have only tested our code with version 2.14.0 for Windows. All required libraries will be programmatically downloaded and installed assuming you have internet connectivity at the time you first run an R module. The following libraries are loaded as needed PresenceAbsence, rgdal, sp, (survival except random Forest), raster, tcltk2, foreign, ade4, mda, randomForest, tools, and lattice.

**6) Download and install the external software Maxent**

The Maxent package is available for download from: <http://www.cs.princeton.edu/~schapire/maxent/>

**7) Configure the SAHM package to know where the above packages are installed.**

Once all the dependencies are installed we just need to let the SAHM package know where the various applications are installed. To do this open the VisTrails application and select Edit -> Preferences -> Module Packages (tab). Select and enable sahm in the list of Disabled packages. 

Click Configure.



By double clicking in the 'Value' area you can change the path to point to the location where various components were installed. In our case we used:

**gdal\_path** = C:\Program Files

note: this is the folder that contains the GDAL folder

**r\_path**  = C:\Program Files\R\R-2.14.0\bin

note: this is the bin folder within your R installation

**maxent\_path** = C:\Maxent

note: this is the folder that contains the maxent.jar file

**Additional configuration:**

While you are setting the above you could also change the following configuration variables:

**output\_directory** = The default location you want SAHM to store your session folders. **verbose**  = Whether to print out all output messages to console screen. This should

stay the default value of True

**Using SAHM and VisTrails**

**Basic Operation.**

VisTrails operation is covered in detail in their own documentation and much of the content won’t be repeated here. The documentation is available at <http://www.vistrails.org/usersguide/VisTrails_Documentation/VisTrails_Documentation.html>

When getting started with the SAHM VisTrails package it is helpful to refer to some previously used workflows to see how modules are commonly connected. To help with this we have provided some template workflows (.vt) that demonstrate the order that components generally go together. These templates are available in the package directory in a folder called templates (ToDo. Design and test these templates). For any module that you are unfamiliar with you can refer to documentation from within VisTrails by clicking on the small triangle in the upper right of the widget and selecting ‘View Documentation’. When connecting modules make sure that the input and output ports are connected to the appropriate location.

The VisTrails application consists of at least two main windows, the GUI and a secondary console window. SAHM displays various messages and warnings to this window. When running SAHM it is generally helpful to periodically check this console window for progress and warning information. If an error is encountered during execution the module that had the error will turn red and an error message will be shown when the module is hovered over. Additonal information might be shown in the console window.

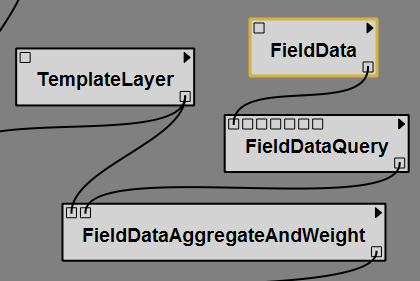
**The session folder.**

All of the outputs produced during a single session of SAHM in VisTrails are saved to a single directory which we refer to as a ‘session folder’. When you first start up SAHM in VisTrails a new uniquely named folder is created in the directory specified in the SAHM configuration. If you would like to continue using a folder from a previous session of change where outputs are being saved to you can select Packages -> SAHM -> Change Session Folder



Within the session folder individual outputs are generally uniquely, sequentially named so that subsequent outputs do not overwrite previous outputs. For example the the first output from the Field Data Aggregate and Weight module will be called FDAW\_1.csv. The next time this module is run the output will be FDAW\_2.csv. In the case of modules that produce several related outputs the group will be placed in a folder named with the same unique sequential convention. The exceptions to this pattern are the Covariate Corelation and Selection module which will have the name from its selectionName parameter PARC module which will be send its output to a folder named according to the template layer used.

**Data Inputs**

****

**Field Data**

The fundamental input to a modeling session is the field data file. At a minimum this file contains rows with the X ans Y coordinates and the recorded detection of a species at that point. It is possible to supply a file that contains additional inputs and use the Field Data Query Module to subset/reformat the output (see Field Data Query Module documentation for more information).

The only Method that must be specified is the value (Path) which is the location of the csv file. The name parameter is an artifact of VisTrails and is not used.

The input data for this module must be in the form a .csv file that follows one of two formats:

Format 1

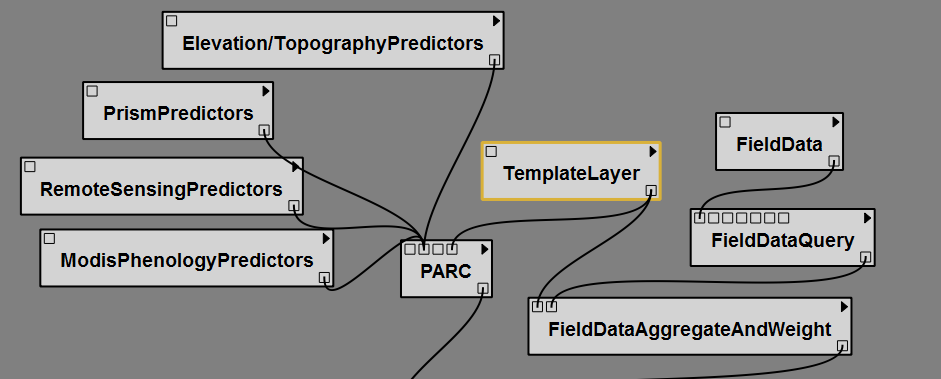
A .csv file with the following column headings, in order: “X,” “Y,” and “responseBinary”.

In this case, the “X” field should be populated with the horizontal (e.g., longitudinal, easting, X, etc ) positional data for a sample point. The “Y” field should be populated with the vertical (e.g., latitudinal, northing, Y, etc) data for a sample point. These values must be in the same coordinate system/units as the template layer used in the workflow. The column “responseBinary” should be populated with either a ‘0’ (indicating absence at the point) or a ‘1’ (indicating presence at the point). Absence points are optional for Maxent.

Format 2

A .csv file with the following column headings, in order: “X,” “Y,” and “responseCount”.

In this case, the “X” field should be populated with the horizontal (e.g., longitudinal) positional data for a sample point. The “Y” field should be populated with the vertical (e.g., latitudinal) data for a sample point. These values must be in the same coordinate system/units as the template layer used in the workflow. The column “responseCount” should be populated with either a ‘-9999’ (indicating that the point is a background point) or a numerical value (either ‘0’ or a positive integer) indicating the number of incidences of the phenomenon recorded at that point.

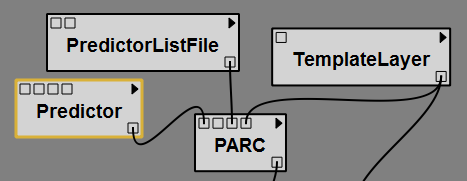


**Template Layer**

The second fundamental input in an analysis is the template layer. It is used to define the extent and resolution that will be used in all subsequent analyses. The TemplateLayer is a raster data layer with a defined coordinate system, a known cell size, and an extent that defines the study area. The data type and values in this raster are not important. All additional raster layers used in the analysis will be resampled and reprojected as needed to match the template, snapped to the template, and clipped to have an extent that matches the template. Users should ensure that additional covariates considered in the analysis have complete coverage of the template layer used.

**Covariates**

The SAHM package provides several ways of specifying which covariates or predictors will be included in subsequent analyasis.



**Predictor**

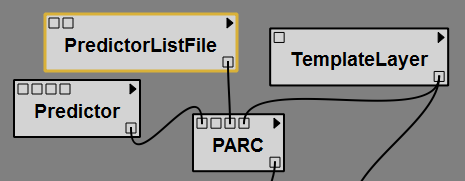
The Predictor module allows a user to select a single raster layer for consideration in the modeled analysis. Four parameters must be specified by the user:

1. Aggregation Method: The aggregation method to be used in the event that the raster layer must be up-scaled to match the template layer (e.g., generalizing a 10 m input layer to a 100 m output layer). Care should be taken to ensure that the aggregation method that best preserves the integrity of the data is used.

2. Resample Method: The resample method employed to interpolate new cell values when transforming the raster layer to the coordinate space or cell size of the template layer.

3. Categorical (Boolean): Checking this box indicates that the data contained in the raster layer is categorical (e.g., landcover categories). Leaving this box unchecked indicates that the data contained in the raster is continuous (e.g., a DEM layer). This distinction is important in determining an appropriate resampling method.

4. File Path: The location of the raster file. A user can navigate to the location on their file system. When a user is selecting an ESRI grid raster, the user should navigate to the ‘hdr.adf’ file contained within the grid folder.



**PredictorListFile**

The PredictorListFile module allows a user to load a .csv file containing a list of rasters for consideration in the modeled analysis. The .csv file should contain a header row and four columns containing the following information, in order, for each input raster.

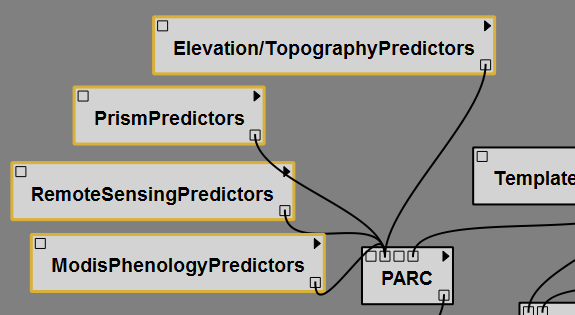
Column 1: The full file path to the input raster layer including the drive .

Column 2: A binary value indicating whether the input layer is categorical or not. A value of "0" indicates that an input raster is non-categorical data (continuous), while a value of "1" indicates that an input raster is categorical data.

Column 3: The resampling method employed to interpolate new cell values when transforming the raster layer to the coordinate space or cell size of the template layer, if necessary. The resampling type should be specified using one of the following values: "nearestneighbor," "bilinear," "cubic," or "lanczos."

Column 4: The aggregation method to be used in the event that the raster layer must be up-scaled to match the template layer (e.g., generalizing a 10 m input layer to a 100 m output layer). Care should be taken to ensure that the aggregation method that best preserves the integrity of the data is used. The aggregation should be specified using one of the following values: "Min," "Mean," "Max," "Majority," or "None."

In formatting the list of predictor files, the titles assigned to each of the columns are unimportant as the module retrieves the information based on the order of the values in the .csv file (the ordering of the information and the permissible values in the file however, are strictly enforced). The module also anticipates a header row and will ignore the first row in the .csv file.



**Individual Predictors selector widgets.**

The SAHM package allows users to set up dynamic Predictor selection widget which allow users to easily select from commonly used predictors. Examples might include Daymet, PRISM, MODIS, etc. For this to work the layers must exist on the local file system and the user must configure a csv file which contains the locations and parameters for each. This csv is located in the SAHM package and is called layers.csv. The first column (FileName) contains the full path to each individual file. The second column (LayerName) is a short descriptive name for this layer. The third column (Group) specifies which widget this layer will be found in. All of the distinct entries in the Group column will generate individual widgets with similar data. In our case we found it convenient to separate climate layers from phenology layers, from topology layers. The fourth column (Source) provideds a means of grouping layers within an individual layer. Each unique entry within a widgets layers will generate a collapsible group of similar layers. The fifth and final column (Categorical) contains a ‘Y’, ‘N’ Boolean value indicating whether this layer is categorical. If it is categorical Resampling will default to ‘nearest neighbor’ and Aggregation will default to ‘majority’. If it is not categorical Resampling will default to ‘bilinear’ and Aggregation will default to ‘mean’. See the original layers.csv for an example of how to set up this file.

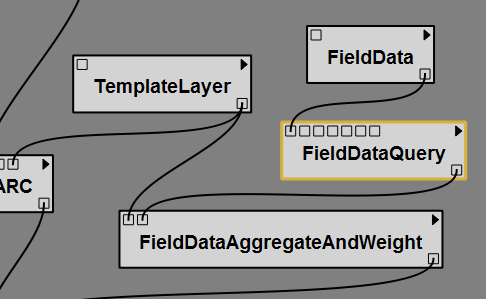
The resulting widgets derived from this file are named the unique Group column name followed by ‘Predictors’. For example you might end up with a ClimatePredictors widget and a PrismPredictors widget. To select individual layers from each widget click the small triangle in the widget’s upper left and select ‘Edit Configuration’ in the dropdown. A box will pop up which contains the layers managed by this widget. Checking the boxes at the left will include that layer in the model. Optionally you can also change the categorical, resampling and aggregation parameters used in subsequent processing.



Within the widget convenience functions to select all layers or switch the selection are provided via buttons at the bottom. The Query box can be used to easily add/remove multible layers. To use enter text in the Query box that is unique to the layers that you want to include or remove. The full paths to the layers will be individually searched for the provided text. For example I might want to include all layers that have MonthlyPrecip in the path. **Note that the query string is case sensitive.**

When the final selection is made click OK to continue.

**Data Preparation**

****

**Field Data Query**

Often raw field data come to us in a format that contains more information than we need to include in any single model. This can take the form of additional columns that contain extraneous information, additional columns that contain occurance data for additional species, or rows that from a time period, collection method, or species that we are not interested in modeling. The Field Data Query module contains functionality to reformat this output into the format used by the SAHM package.

At a minimum you must specify the columns that contain the X, Y, and response. Columns can be specified with either a positional argument (1, 2, 3, etc) if you want to select the first, second, third etc column. Note these numbers start from 1. Alternatively you can select a column based on name by entering the text of the column name found in the header. Finally you must select the response type of either Presence(Absence) or Count. If none is selected it will default to Presence(Absence).

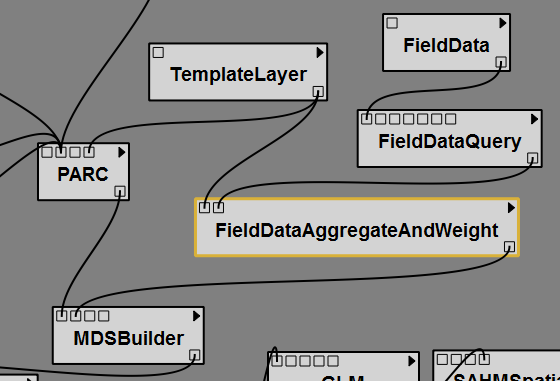
If you would also like to select by rows you can use the Query and Query\_column methods. The Query\_column is the column that contains the data you would like to query on. It will be selected similarly to how columns are selected for X, Y, and response. For the Query column you can either enter a single value or enter an equality statement with x used as a placeholder to represent the values in the query column.

*For example:*

*x < 2005 (would return values less than 2005)*

*x == 2000 or x == 2009 (would return 2000 or 2009)*

*The syntax is python in case you want to create an involved query.*



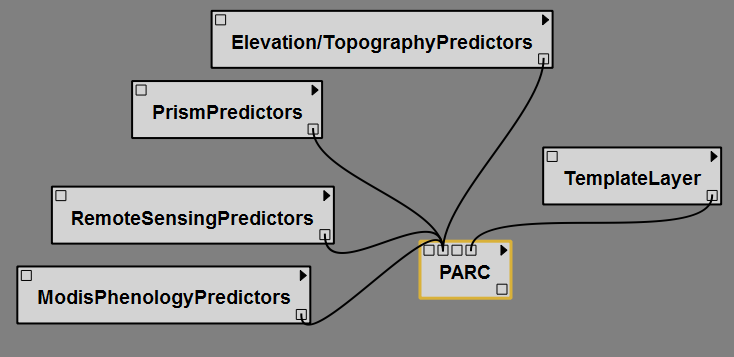
**Field Data Aggregate and Weight**

In many instances data collected in the field can be redundant, both spatially and temporally. When running species distribution models (SDMs) such as those contained in SAHM, spatial issues need to be addressed in order to avoid introduction of pseudo-replication. For instance, considering multiple field data observations which are all spatially located in the same modeled pixel will generate replicate values or redundant information. When running a model, this redundancy causes pseudo-replication and can negatively influence model development. The FieldDataAggregateAndWeight tool helps aggregate field data locations so only one field data observation is represented per pixel. Similarly, field data is often collected temporally through time, where observations of the same location are often recorded annually.

Currently two parameters can be specified by the user:

1. Field Data: The file with the X, Y, Response data. This can be specified here or come from the FieldDataQuery or FieldData modules

4. Template Layer: Raster used to determine cell size and extent. This can be specified here or come from the TemplateLayer module.

**Project, Aggregate, Resample, Clip (PARC)**

The Projection, Aggregation, Resampling, and Clipping (PARC) module is a powerful utility that automates the preparation steps required for using raster layers in most geospatial modeling packages. In order to successfully consider multiple environmental predictors in raster format, each layer must have coincident cells (pixels) of the same size, have the same coordinate system (and projection, if applicable), and the same geographic extent. The PARC module ensures that all of these conditions are met for the input layers by transforming and or reprojecting each raster to match the coordinate system of the template layer. This process usually involves aggregation (necessary when an input raster layer must be up-scaled to match the template layer— e.g., generalizing a 10 m input layer to a 100 m output layer), and or resampling (necessary for interpolating new cell values when transforming the raster layer to the coordinate space or cell size of the template layer). Lastly, each raster predictor layer is clipped to match the extent of the template layer.

The settings used during these processing steps follow a particular set of decision rules designed to preserve the integrity of data as much as possible. However, it is important for a user to understand how these processing steps may modify the data inputs. For additional information about the PARC module, please see the extended help and documentation for the SAHM package.

The PARC module accepts four kinds of inputs:

1. Predictor List: A user should not have to populate this field. This field is populated at execution time with selections made from the pre-loaded .csv file of raster predictor layer inputs (specified during the SAHM install) and connected to the PARC module in the visual display.

2. ListRasterWithPARCInfoCSV: The .csv file list corresponds to the Predictor List File element and allows a user to load a .csv file containing a list of rasters for consideration in the modeled analysis. For additional information, please see the documentation for the Predictor List File element.

3. Predictor: The predictor input allows a user to select a single raster predictor layer to be considered in the analysis. It is recommended that a user add this input as a separate element in the visual display (and then link it to the PARC module) so that the aggregation and resampling settings can be established. The PARC module can accept multiple predictor elements. For additional information, please see the documentation for the Predictor element.

4. Template Layer: The template layer is a raster data layer with a defined coordinate system, a known cell size, and an extent that defines the study area. This raster layer serves as the template for all the other inputs in the analysis. All additional raster layers used in the analysis will be resampled and reprojected as needed to match the template, snapped to the template, and clipped to have an extent that matches the template. Users should ensure that any additional layers considered in the analysis have coverage within the extent of the template layer. The template layer is a required input for the PARC module.

Additional Methods:

1: multipleCores: If checked individual layers will be processed in parallel on each of the available cores on a machine. This can lock up a machine until processing is finished but processing times are much, much faster.

2: ignoreNonOverlap: If checked the requirement that every covariate completely covers the template is ignored. The output extent will be the intersection of the covariates extents. The output will be snapped to the template raster. Use this with caution.

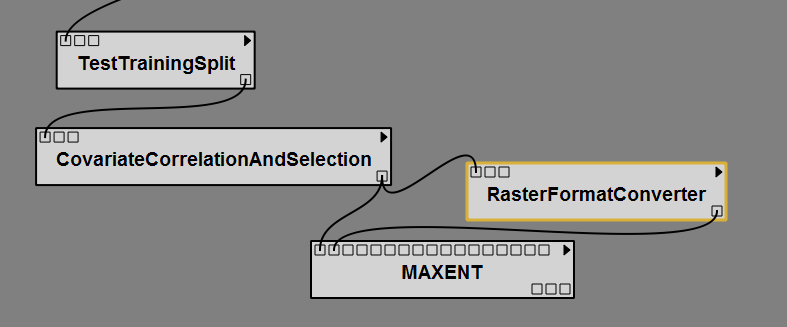
**Typical connections:**

**How PARC works:**

PARC uses a combinataion of GDAL (Which used Proj.4), numpy, and python code to accomplish its task. The processing steps that PARC uses are fairly complex and geospatial implications should be considered fully. The default transformation chosen by GDAL will be used in the reprojection but in some cases this is not optimal and can introduce significant errors. In particular be careful with switching datums, using unusual projections or dataums, using projections or datums far outside of their intended area of accuracy. When in doubt you might want to perform reprojections in an external GIS package. See <http://www.remotesensing.org/geotiff/proj_list/> for more information.

For each raster that is going to transformed into the projection, extent, and cell size of the template raster the following steps are performed:

1. The ratio of source cell size to template cell size is determined. If the source raster cell dimension is greater that 0.5 of the template raster cell dimension no aggregation will be performed. Otherwise aggregation will be performed if specified. The values used in the comparison are the units in the template projection.
2. The source raster is projected into the template projection and extent. If no aggregation is to be used the output cell size will be identical to the template cell size. If aggregation is to be performed the cell size is calculated to be the value closest to the original source cell size that results in template cells containing even numbers of source cells. For example if aggregating from 30m to 100m the value would be 25m. These new pixels will be aligned with the origin of the template such that a specified number of whole pixels fall within each template pixel. The source pixel type and nodata value will be maintained.
3. If required aggregation is performed on the resulting layer. To aggregate, all of the reprojected source pixels that fall within a given template pixel are considered using the specified algorithm minimum, maximum, mean or majority and the resulting value is written to the output raster.



**Raster Format Converter**

The RasterFormatConverter module allows a user to easily convert .tif raster layers into a different raster format for use and display in other software packages. The module accepts as an input either a list of rasters from a Merged Dataset File (MDS) in the workflow or the location of a directory containing multiple raster files. All outputs will be sent to a folder named "ConvertedRasters" (followed by an underscore and a number corresponding to the run sequence of the module) within the user’s current VisTrail’s session folder. Typically this module will be used within a workflow to convert the geotiff format used by the rest of the modules to the ascii format needed by Maxent.

Three parameters can be specified by the user:

1. Format: The format corresponds to the desired raster output format. The following output file formats are supported: Arc/Info ASCII Grid, ESRI BIL, ERDAS Imagine, and JPEG.

To specify the desired output, users should enter the values shown below.

For an ASCII (.asc) output, enter: "asc"

For an ESRI BIL output, enter: "bil"

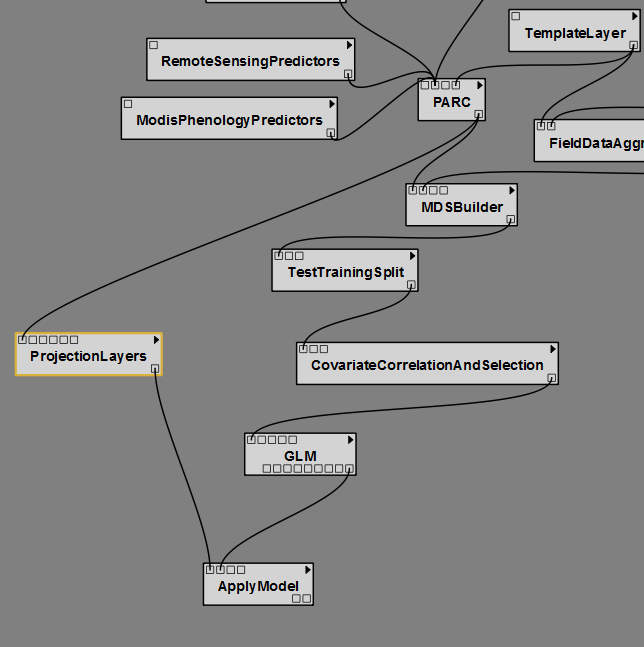
For an Erdas Imagine (.img) output, enter: "img"

For a JPEG (.jpg) output, enter: "jpg"

If no value is entered by the user, the module will default to an ASCII (.asc) output format.

2. Input Directory: The input directory allows a user to point to an entire folder as an input to the RasterFormatConverter. The contents of the specified folder will be checked for raster files and all the raster files contained within the directory will be converted to the format specified in the "Format" dialogue box. The module will identify and convert files of the following raster types: .bil, .img, .tif, .jpg, and .asc.

3. Input Merged Dataset (MDS): The input merged dataset allows a user to specify a .csv file created in the VisTrails workflow (containing a list of .tif raster files) as an input to the raster converter. All of the files listed in the MDS will be converted to the raster format specified in the "Format" dialogue box.



**Projection Layers**

Note: as of June 2011, this module offers some functionality that is only available to users running the SAHM package within the USGS Fort Collins Science Center (FORT).

The projection layers module provides the option to prepare a separate set of predictor layers so that the results of a model developed from one set of environmental predictors can be projected onto a new modeled space. This second set of environmental predictors (corresponding to the "projection target") most often contains the same environmental predictors but represents data captured at a different temporal or spatial location. For example, a user could generate a model predicting habitat suitability using recorded presence points and certain environmental predictors such as elevation, landcover, and proximity to water in one geographic location. Based on the training from this information, the modeled results could be generated for (or "projected to") a new location based on the range of values seen in elevation, landcover, and proximity to water in the second geographic area. Similarly, modeling predicted results through time is also possible. A model trained using field data and a set of predictor layers representative of one time period could be projected onto the same geographical area using a new set of predictor layers corresponding to the same predictors but representing data from a different time period (e.g., different climate data).

The output of this module is subsequently used as the projection target in the ApplyModel module.

(As part of the process of preparing the layers for modeling, the ProjectionLayers module runs the PARC module internally on the inputs. Outputs from the ProjectionLayers module will possess matching coordinate systems, cell sizes, and extents and do not need to be run through PARC before being used downstream in the workflow.)

Six parameters can be set by the user:

1. Directory Crosswalk CSV: This is a .csv file containing two columns designating the layers that should be swapped out in the projected model. The first column contains a list of the full paths to the predictor layers used to develop the original model that will be replaced in the projection process. The second column contains the full paths to the new predictor layers that will substitute the respective layers used in the original model. Each original layer in the first column should be paired with its replacement in the second column (e.g., Column 1 = C:\ModelLayers\Precipitation1980.tif, Column 2 = C:\ModelLayers\Precipitation2000.tif). In the case of any file used to develop the first model that is not expressly listed in the Directory Crosswalk CSV with a replacement, the original file will be reused in the new model projection. The module anticipates a header row in this .csv file (thus, the first row of data will be ignored).

2. File List CSV: This is a .csv file containing the list of predictor files used to develop the first model. Effectively, this file will be updated based on the information provided in the directory crosswalk .csv and used as the input to the training process for the projected model. The output of the PARC module from the first model iteration should be used as the input to this parameter.

3. Model (available only to users at the FORT): This parameter allows VisTrail users running the SAHM package on site at the USGS Science Center in Fort Collins (FORT) to specify one of three models to use for the projected model run ("CCCMA, " "CSIRO, " or "hadcm3").

4. Scenario (available only to users at the FORT): This parameter allows VisTrail users running the SAHM package on site at the USGS Science Center in Fort Collins (FORT) to specify one of two scenarios for the projected model run ("A2a" or "B2b").

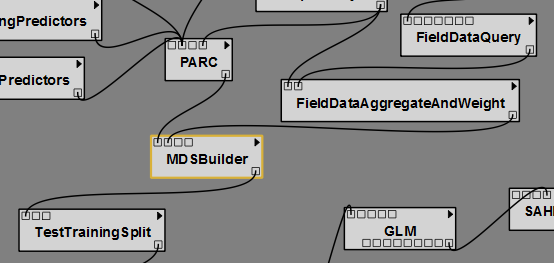
5. Template: This parameter allows a user to specify the new template layer to be used in the projected model run. The template layer is a raster data layer with a defined coordinate system, a known cell size, and an extent that defines the (new) study area. This raster layer serves as the template for all the other inputs in the analysis. All additional raster layers used in the analysis will be resampled and reprojected as needed to match the template, snapped to the template, and clipped to have an extent that matches the template. Users should ensure that all the layers used for the projected analysis have coverage within the extent of the template layer.

6. Year (available only to users at the FORT): This parameter allows VisTrail users running the SAHM package on site at the USGS Science Center in Fort Collins (FORT) to specify one of three years to use for the projected model run ("2020," "2050," or "2080").

**Pre-modeling Data Manipulation**

Much of the remaining workflow uses a file format we devised called Merged Dataset or MDS. This format combines the X, Y, response columns of the field data file with the pixel values extracted from the individual covariate rasters at each point. In this regard it is similar to the Samples With Data (SWD) format used by Maxent. The MDS format uses three headerlines to record and transmit additional pieces of information as well. The first header line contains the shortnames for each covariate. Additionally for categorical covariates the work \_categorical is appended to the end of the name. This will indicate to subsequent models how to treat this variable. The second line in an MDS file contains zeros and ones which indicate whether to include or remove this covariate from subsequent models. The third line contains a file path to the location of the raster used to extract values.

In addition to the above items the MDS file stores a few bits of information about its provenance. In cell 1, 2 is the template used for the field data aggregate and weight. In cell 2,2 is the field data file used. Cell 2, 3 contains the name of any selection applied to the MDS. Cell 1,3 contains the raster used for the PARC operation (this should always be identical to cell 1, 2). Cell 2,3 contains the folder that PARC output was saved to.

****

**Merged Dataset Builder.**

The Merged Data Set (MDS) Builder module is a utility that extracts the values of each predictor layer to the point locations included in the field data set. The module produces a .csv file that contains the x and y locations of the sample points and a column indicating whether each point represents a presence recording, an absence recording, a presence count, or a background point. Following these first three columns, each environmental predictor layer is appended as a column with row entries representing the value present in the raster layer at each field sample point. There are a total of three header rows in the output .csv of the MDSBuilder. The first row contains the columns "x," "y," "ResponseBinary" or "ResponseCount," and the names of each of the raster predictor files that were passed to the MDS Builder. The second row contains a binary value indicating whether the column should be included when the model is finally applied; these values are later modified during the Covariate Correlation and Selection process that takes place downstream in the workflow. The final header row contains the full path on the file system to each of the raster predictor files.

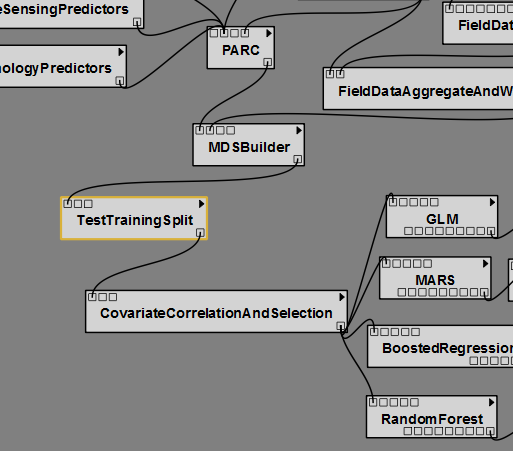
The MDSBuilder accepts four inputs from the user:

1. Rasters with PARC Info .csv File: The raster layers listed in this file will have their values extracted to the points in the field data .csv file. This parameter should be supplied by connecting the output of a PARC module within the workflow to the MDSBuilder module. (In order to properly extract the values of each predictor layer to the field data points, all the layers must have matching coordinate systems and cell sizes; outputs from the PARC module will have had the prerequisite processing).

2. Background Point Count: This is an optional value that applies only to workflows that employ the Maxent modeling package. The dialogue box provides the option of specifying a number of background points to be randomly scattered throughout the study area (the extent of the template layer) to capture a more complete sample of the range of values present for each predictor layer. These points will be added to the field data .csv file with a value of "-999" denoting them as background points.

3. Background Probability Surface: This is an optional parameter that applies only to workflows that employ the Maxent modeling package. In some analyses, it may be appropriate to spatially limit background points to a particular subset of the study area (e.g., islands within a study area polygon, particular regions within a study area polygon, or a region determined by the known bias present in the field data). Specifying a background probability surface raster allows a user to control where random points will be scattered within the extent of the study area. The raster layer specified by a user should have the same projection and extent as the template layer and contain values ranging from 0 to 1. These values represent the probability that a randomly generated point will be retained should it fall within a particular cell. That is, randomly generated points will not be generated in any part of the probability grid with a value of "0," while all points falling in an area with a value of "1" will be retained. A point falling in an area with a value of "0.5" will be kept as a background point 50% of the time.

4. Field Data: The field data input corresponds to a .csv file containing presence/absence points or count data recorded across a landscape for the phenomenon being modeled (e.g., plant sightings, evidence of animal presence, etc.). This input file must be in a particular format, and in most cases, a user should populate this field by connecting a FieldData element to the MDSBuilder in the visual display within VisTrails. Please see the documentiondocumentation for the FieldData module for more details.



**Test Training Split.**

The TestTrainingSplit module provides the opportunity to establish specific settings for how field data will be used in the modeling process. Three parameters can be set by the user:

1. Ratio of Presence/Absence Points:

This field is populated with a number corresponding to the desired proportion of presence and absence points to be used in the analysis. If populated, this entry should be a number greater than zero. (A value of ‘1’ will result in an equal number of both presence and absence points being used, a value of ‘2’ indicates that twice as many presence points will be used, a value of ‘0.5’ indicates that twice as many absence points will be used, etc.). All field data points with a value equal to or greater than 1 are interpreted as presence points. Although the original field data is unmodified, this option will reduce the sample size as the merged dataset containing sample points will have points deleted from it to achieve the specified ratio. A warning will be generated if more than 50% of either the presence or absence points will be deleted based on the ratio specified by the user. Background points are ignored by this module (they are read in and written out, but not assigned to either the test or training split).

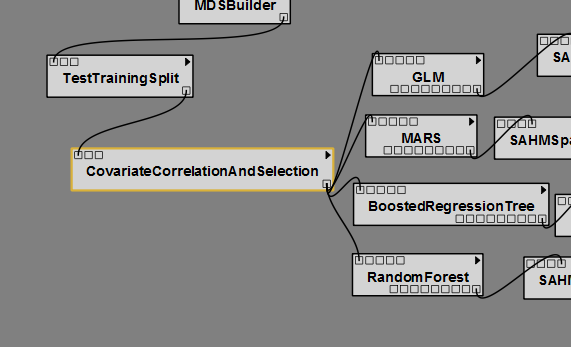
When left empty, this field will default to ‘null’ and the model will use the existing presence/absence ratio present in the field data.

2. Input Merged Data Set (MDS):

This is the input data set consisting of locational data for each sample point, the values of each predictor variable at those points, and if established, a field denoting the weight that will be assigned to each point in modeling. This input is usually provided by the upstream steps that precede the Test Training Split module. Any value entered here (e.g., specifying another existing MDS on the file system) will override the input specified by a model connection in the visual display.

3. Training Proportion:

This is the proportion of the sample points that will be used to train the model, relative to the total number of points. Entered values should be greater than 0 but less than 1. For example, a value of ‘0.9’ will result in 90% of the sample points being used to train the model, with 10% of the sample being held out to test the model’s performance. Choosing an appropriate training proportion can depend on various factors, such as the total number of sample points available.



**CovariateCorrelationAndSelection**

The CovariateCorrelationAndSelection view provides a breakpoint in the modeling workflow for the user to assess how well each variable explains the distribution of the sampled data points and to remove any variables that may exhibit high correlation with others.

The display shows the 10 most correlated variables of those selected. These variables are displayed on the diagonal and their respective graphical display and correlation with other variables can be found by locating the row/column intersection between each (above and below the diagonal). The column heading over each variable displays the number of other variables with which the environmental predictor is correlated. The user defined “Threshold” option allows a user to specify the degree of correlation required between two variables for them to be counted in this tally.

A user is provided with the opportunity to select a new set of the environmental predictor variables and “Update” the Covariate Correlation screen to investigate the relationships among the new variables selected. The options are provided to include or exclude the presence/count points, the absence points (when applicable), and the background points in this correlation test.

Variables with a high degree of correlation with other variables should generally be unchecked in their respective radio buttons, and will be excluded from subsequent analysis steps in the model workflow.

Multiple iterations can be run at this screen, allowing the user to investigate the relationships among the environmental predictor variables and choose the most appropriate set to be used in the subsequent modeling. When the desired set of variables has been chosen, the “OK” button is selected and processing will resume in the VisTrails workflow.

Three parameters can/must be specified by the user:

**ShowGUI** – This Boolean indicates whether to stop execution and display the GUI for user interaction. In somecases such as exploration you might want to make a selection in a previous run and then change this to false so that the selection will be apply to subsequent runs without interrupting execution.

**inputMDS** - The file to select from. If this file contains unselected layers (0 in the second header line) these will initially appear deselected in the GUI.

**selectionName** – This serves two purposes. First to uniquely identify a given selection. This unique name is used to determine if a selection has been previously made, to apply for example. And secondly to provide something that can be changed to trigger VisTrails to rerun this module even if nothing upstream has changed.

**Modeling**

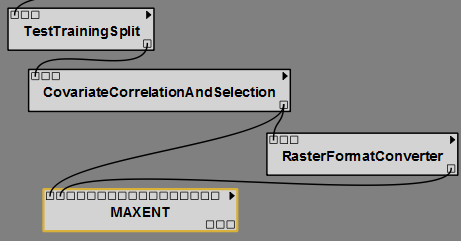
Marian to write

**GLM**

**MARS**

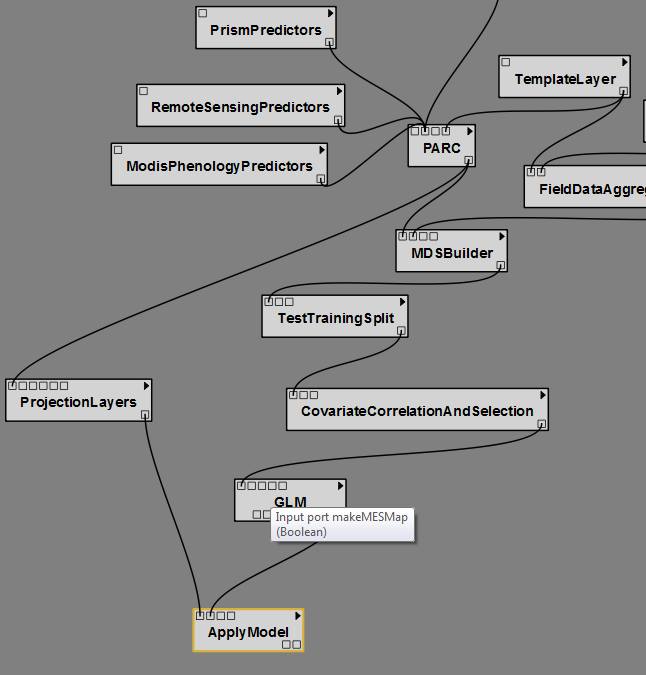
**BoostedRegressionTree**

**RandomForest**

****

**Maxent**

Catherine to write.



**Apply Model**

Marian to write

**Viewing Output**

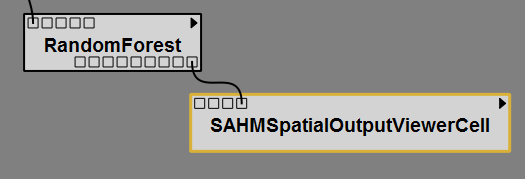
**Spreadsheet**

VisTrails has a spreadsheet to view individual outputs side by side. Besides the built in cells for viewing text, webpages, etc SAHM has two specialized cells for viewing model outputs. The output from several model runs can be viewed side by side in individual spreadsheet cells.



**SAHM Model Output Viewer Cell**

This widget is for viewing the textual and graph output from R. It contains tabs for the model output report, the AUC plot, and more

****

**SAHM Spatial Output Viewer Cell**

The SAHM Spatial Output Viewer Cell provides a convenient means for viewing the numerous spatial outputs produced by individual model runs as well as the input presense and absence points and background points if applicable. The spatial viewer displays the outputs in an interactive Matplotlib chart which functions much like a full GIS. ****

Attached to each cell is a toolbar that allows changing of the displayed layer and the overlaid points

**Spatial Output Viewer Toolbar:**

When panning you can zoom by holding a right click and moving the mouse up for zoom out or down for zoom in.

When using the Zoom rectangle a right click rectangle selection will zoom in, a left will zoom out.

Reference, 2000, National water quality inventory—1998 report, U.S. Environmental Protection Agency Report EPA–841–F–00–006, 45 p.

# Glossary

Glossary Term Glossary definition.

Appendix

# Additional character style examples

Thes style for superscript is Superscript.

The style for subscript is Subscript.

The style for italic superscript is SuperEmphasis

The style for italic subscript is SubEmphasis

#### The style for italic Arial Narrow is Emphasis UC

## The style for italic Arial Narrow Bold is EmphasisStrongUC

#### The style for Arial Narrow superscript is SuperEmphasisUC.

#### The style for Arial Narrow subscript is SubEmphasisUC.

The style for a URL is <http://www.usgs.gov> (Hyperlink)