Dan Higgins
August 15, 2005

Environmental Niche Model Workflow

Workflow for Determining Best GARP Rulesets (Single Species)

The basic top-level tasks that must be done to determine the 'best' GARP Rulesets for a single species are indicated in the conceptual workflow of Figure 1. Each Kepler composite actor shown in the workflow is identified by a Roman numeral (e.g. I, II, …).

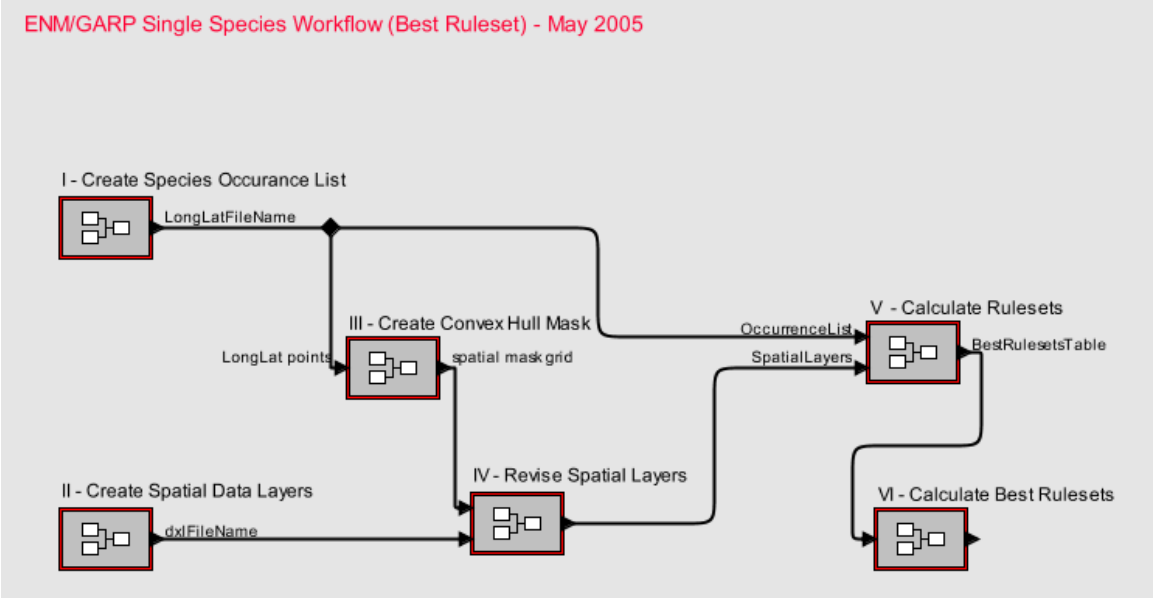


Figure 1 – Top Level Workflow Blocks

I – Create Species Occurrence List

This block will create a file that lists the longitude and latitude of known locations where the species is known to exist. Currently, this data just comes from a search of DigiR for the species name.

How to Search Digir

Searching for species information within Kepler is relatively simple. Just click on the "Data" tab in the left pane the window, enter part of a species name, and (after a delay) a list of results will appear as indicated in Figure 1-1 below.

Note that we just searched for a species name. The ecogrid automatically searches several data sources, including DigiR which has information on museum samples of specimens. If you click on the "`Source`" button on the Data tab shown in the figure, you will see the dialog shown in Figure 1-2. If you remove the checks for the 'KNB Metacat' and 'GEON Search' sources (by clicking on the check box), then only 'KU Digir' sources will be searched.

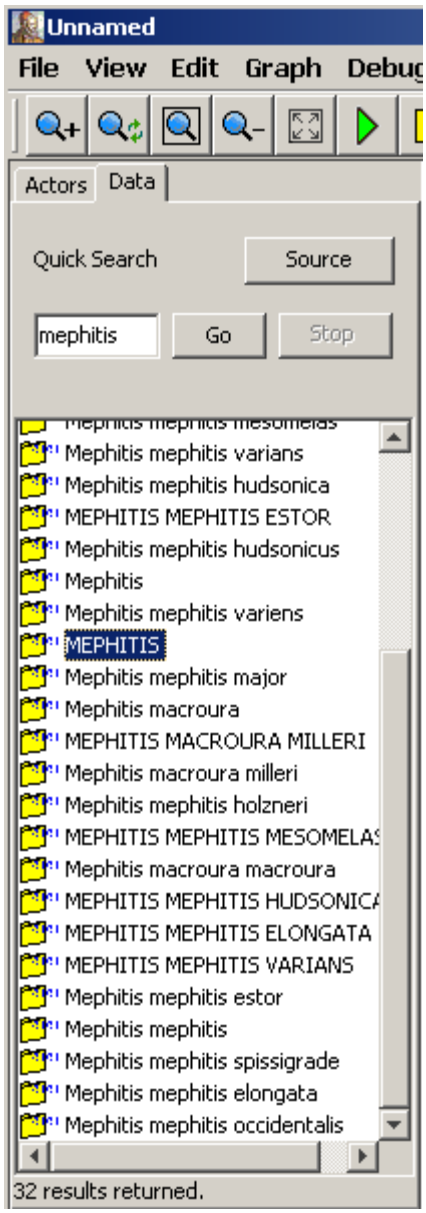


Figure 1-1 Data Search Panel

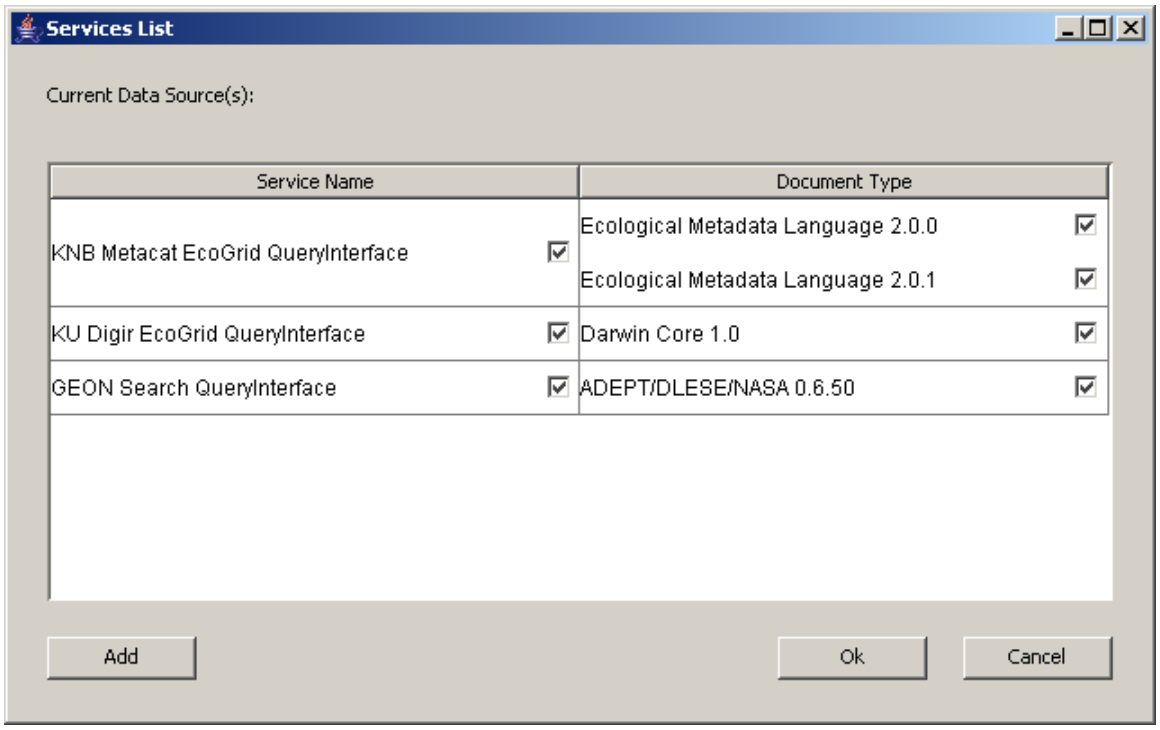


Figure 1-2 Services Dialog (appears when "Source" button is clicked)

If you drag one of the results (say the one labeled "`MEPHTIS`") to the right, an actor like that shown in Figure 1-3 will appear. (Port names have been turned on; normally they will not appear.) (Also, the icon will initially appear in red. Wait until it turns yellow; that indicates that all the associated data has been transferred to the local cache.) The six output ports indicate that there are 6 columns, named as indicated, in the dataset.

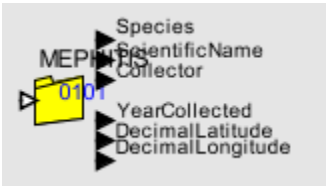


Figure 1-3 Digir datasource actor (with port name display activated)

If you 'right-click' on the actor and choose the 'Configure' menu item, a dialog like that in Figure 1-4 will appear. Note the `outputType` is "`as Field`". This indicates that each of the columns (fields) in the data table is output on a separate output port. We really are more interested in just the longitude and latitude of the specimen, so change the `outputType` to "`As Table`". When you close the dialog, the actor display will change and only 3 output ports appear as indicated in Figure 1.5.

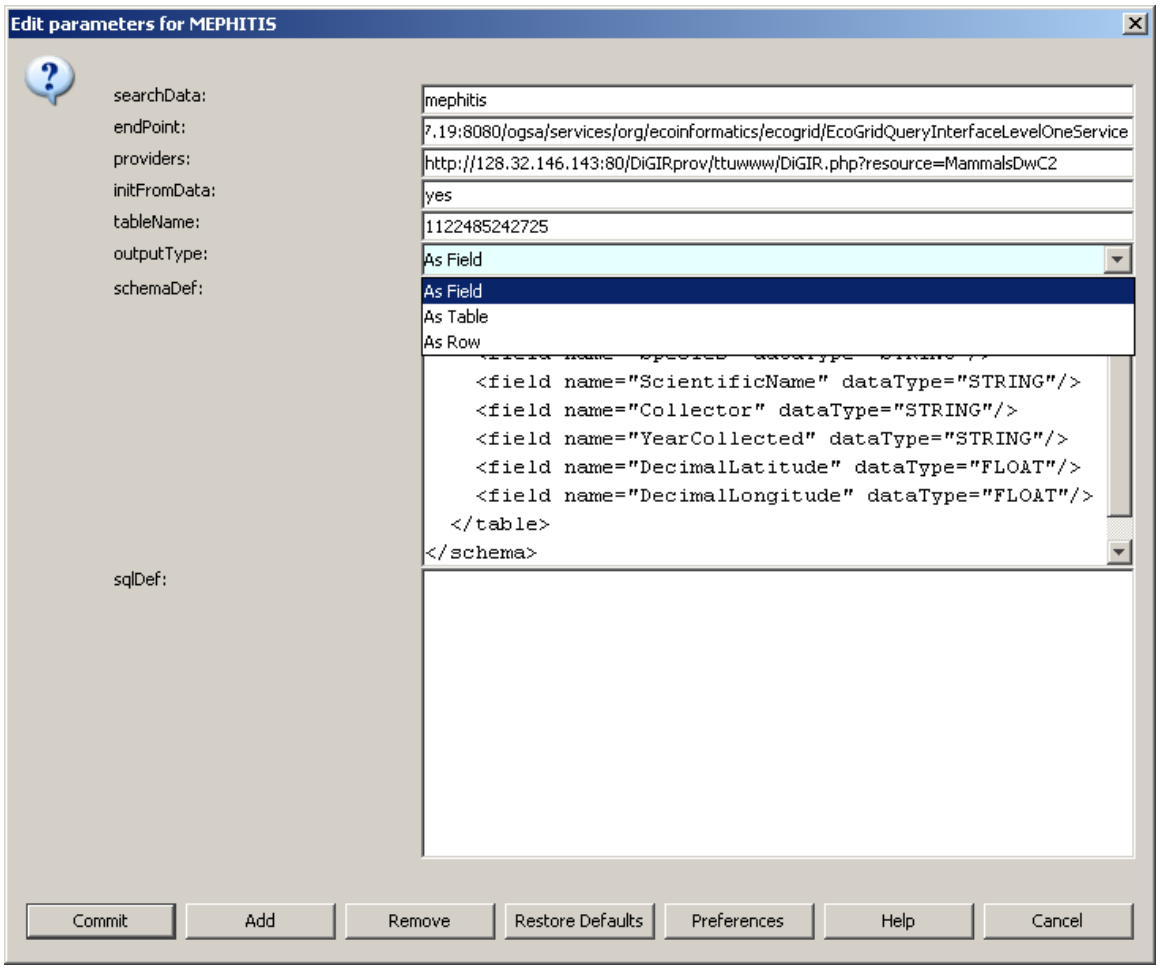


Figure 1-4 Configure Dialog for Digir data source actor

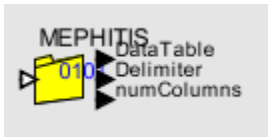


Figure 1-5 Actor when `outputType` is '`as Table`'

You can actually look at the data output as tab delimited text by using the workflow indicated in Figure 1-6. However, we really only want the longitude and latitude information for the GARP/ENM workflow. Thus, select the Digir actor, do a right-click to get the popup menu, and then select "Look Inside". This will bring up the dialog shown in Figure 1-7.

The dialog shown in Figure 1-7 can be used to create a local query that will modify the information returned. Click on 'DecimalLongitudetude' in the top of the dialog and drag it to the bottom. Then do the same for 'DecimalLatitude'. Then set the 'Operator' and 'Criteria' values as indicated in Figure 1-8. (This check eliminates missing/invalid data.) Then close the dialog and accept the changes.

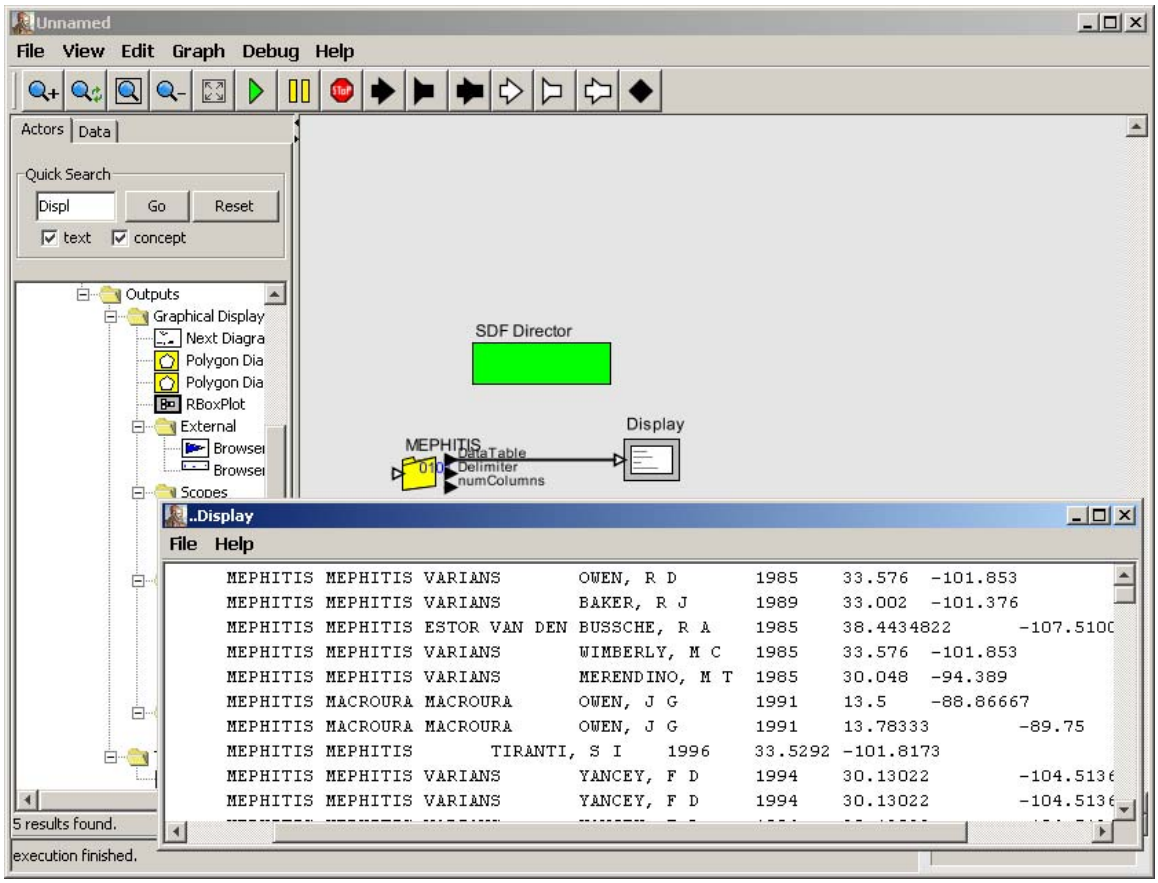


Figure 1-6 Simple Workflow to show Digir data table

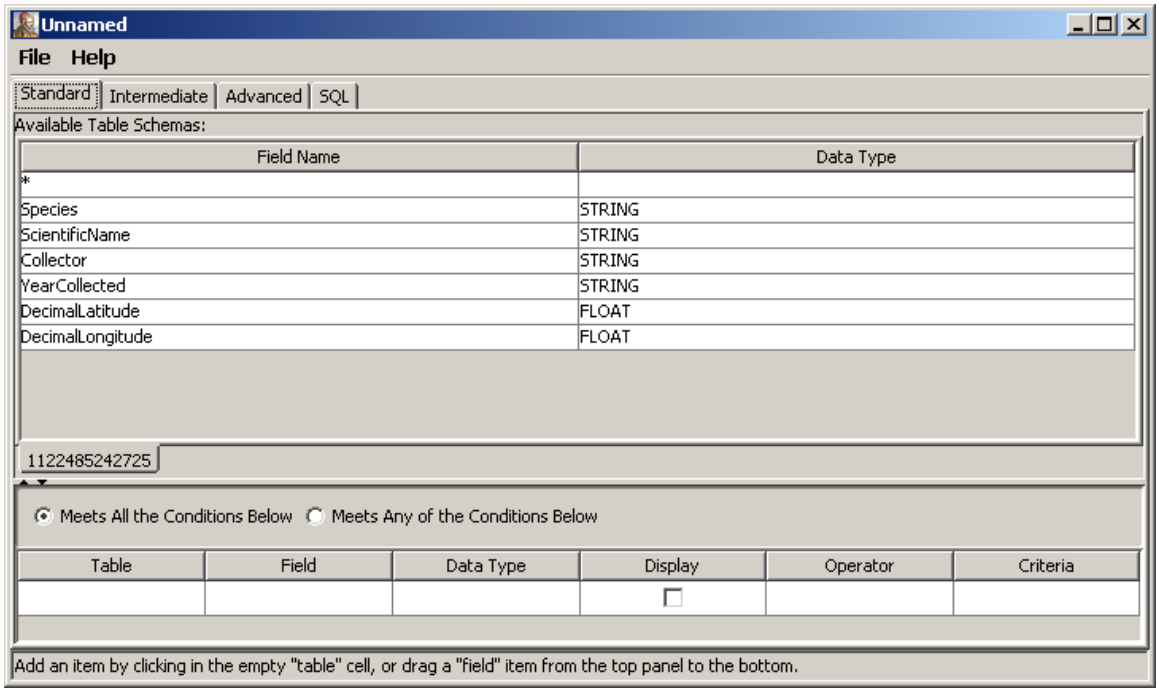


Figure 1-7 Dialog when you “Look Inside” a Digir data source

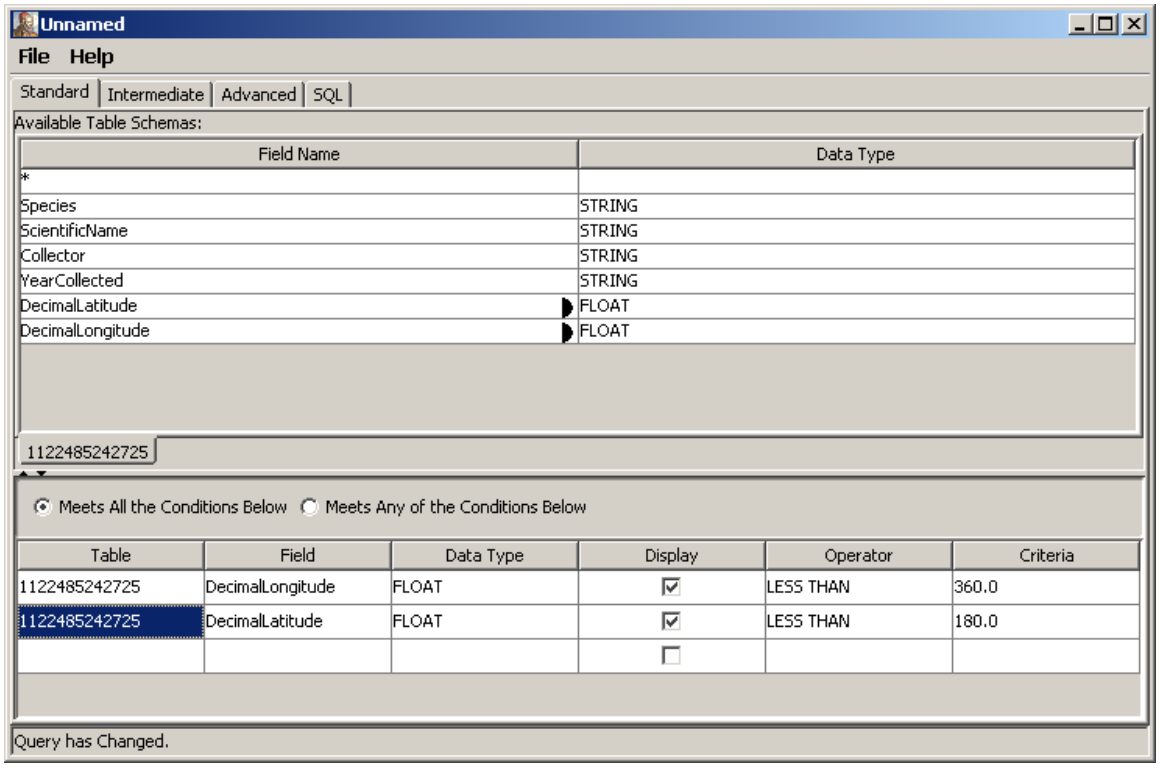


Figure 1-7 A modified table to only return valid longitude and latitude values

If you run a workflow to look at the resulting table after making these changes, you will get a something like that shown in Figure 1-8. Note that now, the resulting table has only longitude and latitude values, which is just what we need for input to GARP. [However, the table is in the form of a single string which needs to be saved as a file for input to GARP.]

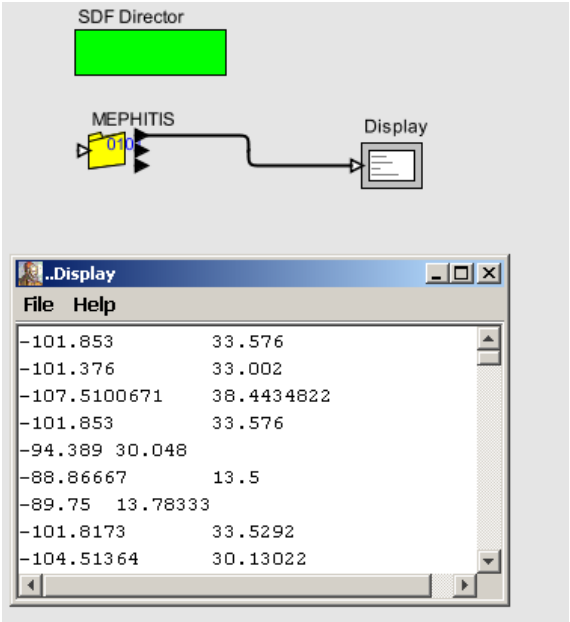


Figure 1-8 Longitude/Latitude Values after query revisions

II – Create Spatial Data Layers

A set of spatial data layers with climate and geographic data must be created for use in the GARP genetic algorithm. Hydro1K and IPCC data sources are currently the sources for this data. All layers to be used are converted to grids with the same resolution and extent.

Note that all the spatial layers used in a GARP calculation are summarized in a '*.dxl' file. This file is an XML document that lists all the spatial layers and a "mask" layer. Any cell in the mask grid with a 'NODATA_value' is ignored, while cells with any other value are processed by GARP. The "ASCToRaw" actor will take an array of spatial layers file names (in *.ASC format), convert the grids to GARP's 'raw' format, and create an output *.dxl summary file for use as GARP input.

Hydro1K Data

HYDRO1k, developed at the U.S. Geological Survey's (USGS) EROS Data Center, is a geographic database providing comprehensive and consistent global coverage of topographically derived data sets. Developed from the USGS' recently released 30 arc-second digital elevation model (DEM) of the world (GTOPO30), HYDRO1k provides a standard suite of geo-referenced data sets (at a resolution of 1 km) that will be of value for all users who need to organize, evaluate, or process hydrologic information on a continental scale. (See http://lpdaac.usgs.gov/gtopo30/hydro/readme.asp)

If you search for 'Hydro1K' in the Kepler Data search window, you will see a result like that shown below with 12 results. You can 'right click' on any of the results and then select 'GetMetadata' to see a detailed description of the information in that package. There are six distinct types of Hydro1K data for both North and South America. Note that the resolution of the data is 30 arc-seconds or approximately 1000 meters (hence the '1K' in the name).

The data types are described in the metadata. A brief summary is:

Aspect - The aspect data set describes the direction of maximum rate of change in the elevations between each cell and its eight neighbors.

CTI - The Compound Topographic Index (CTI), commonly referred to as the Wetness Index, is a function of the upstream contributing area and the slope of the landscape.

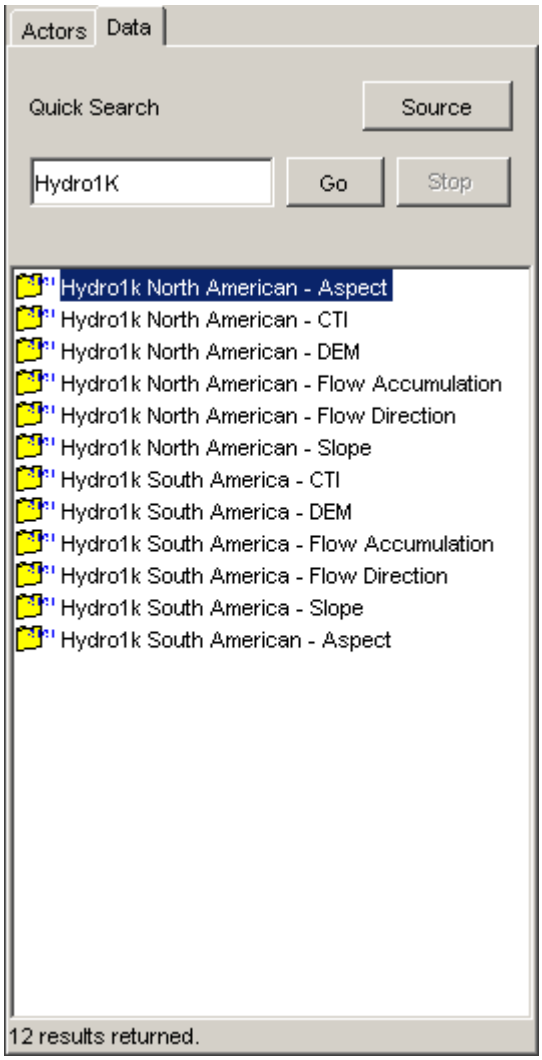
DEM – The Digital Elevation Model forms the basis of all the additional HYDRO1k data sets.

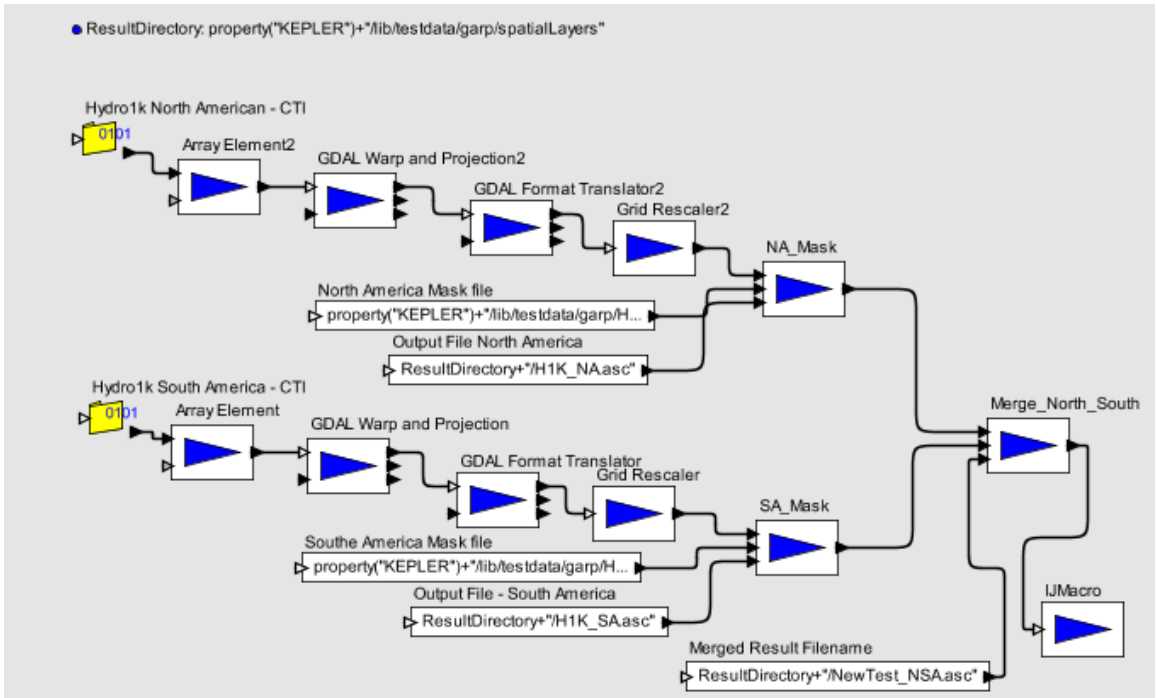
Flow Accumulation - Developed from the flow direction layer, the flow accumulation data set defines the number of cells which flow into each downslope cell.

Slope - The slope data set describes the maximum change in the elevations between each cell and its eight neighbors.

The Hydro1K grids are stored as *.bil files and use a Lambert Azimuthal Equal Area coordinate system. It is thus necessary to read these specialized grid files, re-project the grids to a (longitude, latitude) format, rescale to a different resolution and extent, and save in *.ASC format for use with other spatial layers (e.g the IPCC climate layers). A workflow for carrying out all these operations is shown below.

There is a bit of a problem with the GDAL reprojection to a (longitude, latitude) format as illustrated by the re-projected image of North America shown below. Note the rather strangely shaped black region surrounding the outline of North America. The outer boundary of this region apparently corresponds to the edges of the original grid, and one sees the effects of the projection. Unfortunately, the exterior gray region should have a value of 'NO_DATA' while its actual value is assigned to be '0.0'. We thus need to apply a mask that sets all points outside of North and South America to 'NO_DATA' values.

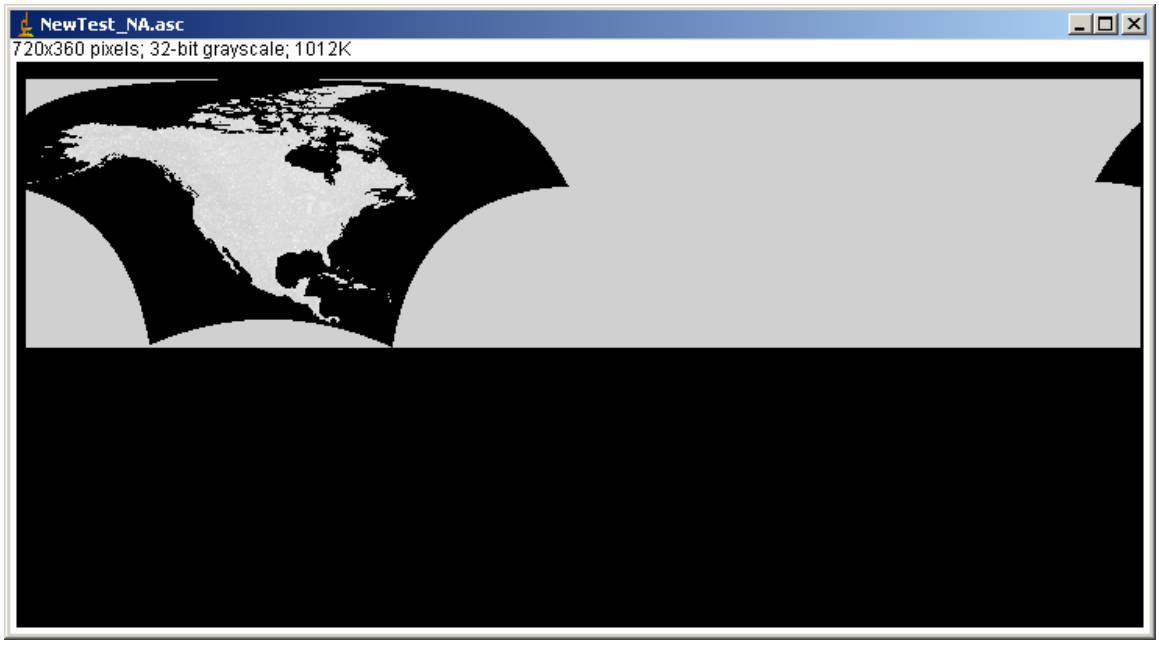




This is an example of how to input Hydro1K data of a specific type (e.g. CTI) from two continents (North and South America) and then combine the results into a single ASC file for the western hemisphere. Note that a projection to (longitude, latitude) format and a rescalng is required.

This workflow may required a long time (~45 minutes) the first time it is opened! There will be a significant delay waiting for the eml2 datasources to turn from red (BUSY) to yellow as data is copied to the local cache.

Dan Higgins - August 17, 2005



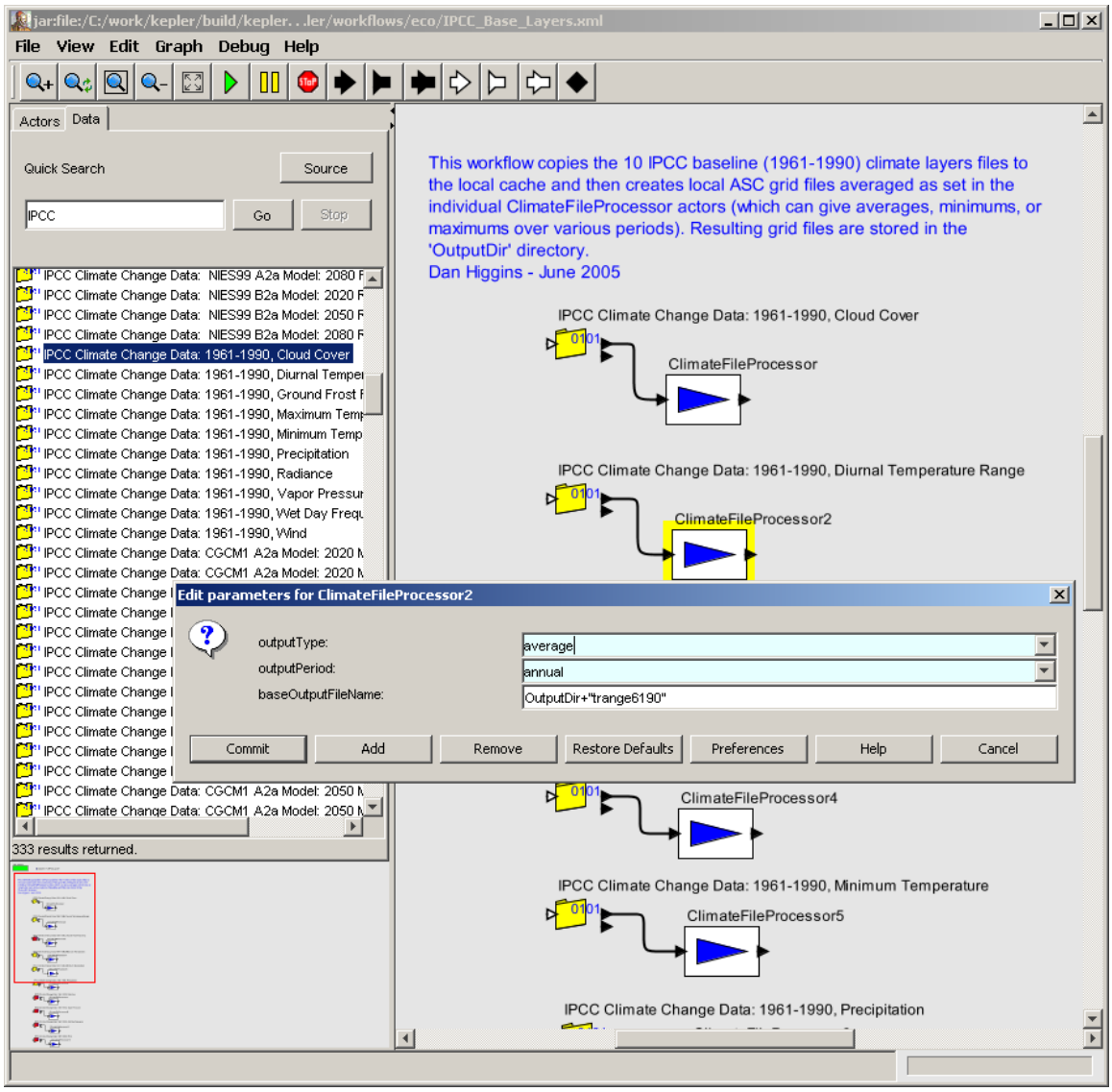
As previously noted, Hydro1K data is organized by continent. Since the current effort is primarily interested in Western Hemisphere species predictions, the results for North and South America are combined rather simply using the MergeGrids actor.

IPCC

IPCC is the acronym for the "Intergovernmental Panel on Climate Change". (See http://www.ipcc.ch/).This group has collected historical climate data and made a number of predictions for future climate values.

If you carry out a Kepler data search for "IPCC", over 300 data sets should be found. Most of these are predicted future climate values using different models and different predictions models. There are 10 baseline datasets which have historical data from 1961 to 1990 for the entire land surface of the earth with a 0.5 degree resolution. The type of data includes:

Cloud Cover
Diurnal Temperature Range
Ground Frost Frequency
Maximum Temperature
Minimum Temperature
Precipitation
Radiance
Vapor Pressure
Wet Day Frequence
Wind



IPCC data is stored is a (longitude, latitude) based grid and separated by month. It is processed into ASC grid layers for use in GARP calculations using the ClimateFileProcessor actor which allows the choice of average, minimum, or maximum values over annual or seasonal periods. The workflow "IPCC_Base_Layers.xml" transfers the 10 baseline layers to the local file system.

III – Create Convex Hull Mask

A convex hull (a 'minimum' bounding polygon) that includes all input locations is calculated, perhaps scaled (i.e. made larger by a factor determined by the user), and then converted to a 'mask' grid indicating the region to be considered in the calculation. This operation limits the calculations to a region enclosing the known locations of the species.

IV – Revise Spatial Layers

The spatial layers used by GARP include a 'mask' layer. All points outside the mask are ignored in the calculation. In this case, the mask layer is replaced by the mask grid created from the species location convex hull. (Otherwise, the spatial layers are independent of the species locations.)

V – Calculate Rulesets

This block takes the occurrence list and the spatial layers and uses GARP to calculate 'rulesets' for predicted species locations. The rulesets predict species locations but may be different each time GARP is run (i.e. the results have statistical variations). The calculation is repeated a large number of times (~100-500). Each time, omission/commission values are calculated for the result using the input occurrence data (perhaps divided into training/testing sets). These values are saved, along with the ruleset, so that the 'best' rulesets can be determined.

VI – Calculate 'Best' Rulesets

The table of results generated in Block V is examined to determine the 'best' rulesets. Those 'best rulesets' are saved for use in further climate change calculations.

The general 'recipe' (provided by Ricardo Periera) used for determining these best rulesets is:

1) Get a set of presence data points (species occurrences - x, y coordinates) to test.
2) Project the GARP model onto geography (map generated by GarpProjection actor)
3) Overlay the presence points from item #1 onto the map generated on #2. The percentage of those points that fall in a pixed not predicted present is your OMISSION. Say, out of 100 points, only 45 fall on white pixels, the other 55 fall on black ones, your omission is 55% or 0.55.
4) Commission, when we don't have real absence points (our case) is the proportion of area predicted present with regard to the total area of interest, not counting masked pixels. So if 40% of the area is predicted present, your commission error is 40% or 0.40.
5) Then, select those GARP runs that show omission below a certain omission threshold, say 5 or 10%.
6) From those runs selected in #5, sort them by commission error, and then get the 50% of the models that are around the median value for commission. If you got, say, 20 models in item #5, now you have 10 models that make up your best subset of models.
7) Sum up the maps for the best subset of models in item #6, that is your final prediction map for your species.

Inside Block V – Calculate Rulesets

Block V of the overall workflow is the task where the GARP algorithm is iterated numerous times to get a statistical sample of predicted species distributions. The tasks that must be carried out as part of this block are shown in Figure 2. (This is the workflow

that is seen when one 'Looks Inside' the composite actor of Block V. Most of the actors here are composites; they are labeled with capital letters (A, B, …).

Every task in this workflow is repeated many times – i.e. inside a repeating loop. The 'Ramp' actor drives that loop. A 'Ramp' is basically just a counter where the user sets the minimum, maximum, and step size.

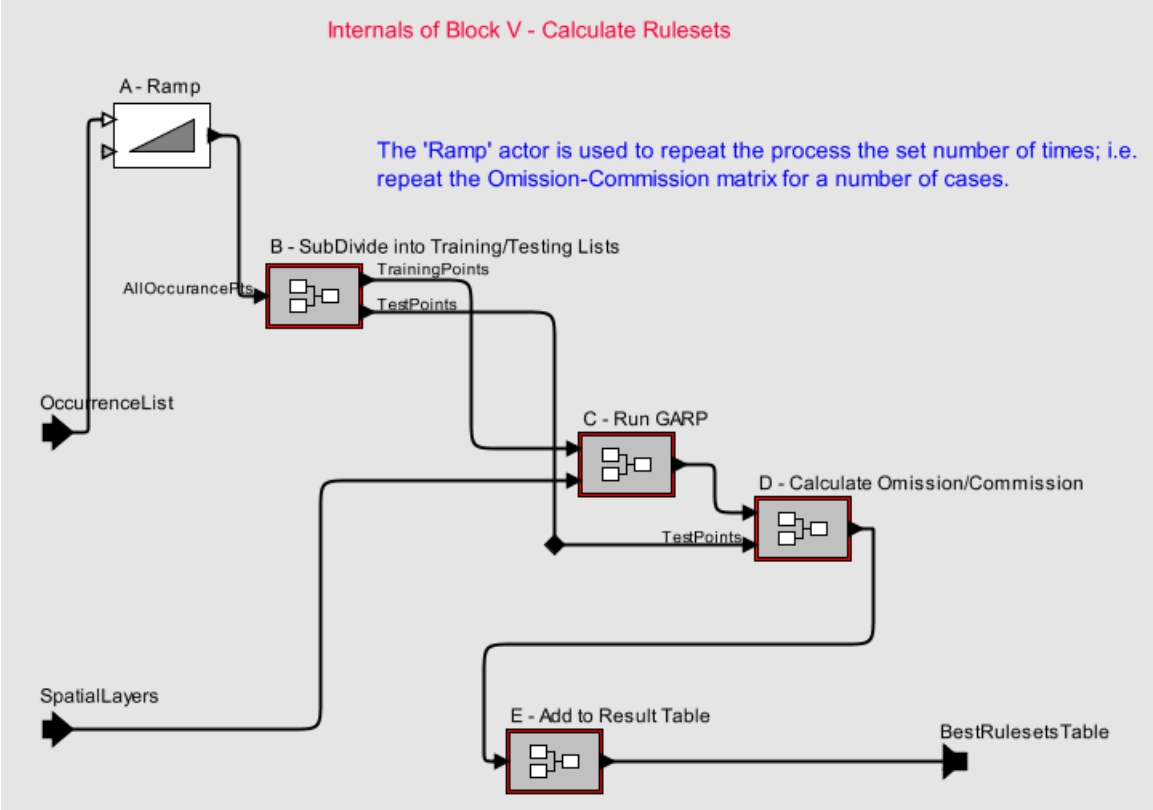


Figure 2 – Internals of Block V – Calculate Rulesets

On each iteration, the input occurrence data must be divided into two sets (Step B in Figure2), one used as training points for the GARP calculation and the second used as testing points to evaluate GARP results. Currently, this subdivision is simply random. GARP is then run using the training points. Omission and commission values are then calculated (Step D) and saved in a results table along with the ruleset generated by the GARP calculation (Step E). The ruleset is saved so that once the 'best' results are determined, the ruleset can be applied to generate new predictions for new climate layers.

Implementation

Previous discussions have been purely conceptual – (meaning that the workflows didn't actually run!). An actual Kepler implementation of the workflow has been created and the top-level workflow is shown in Figure 3.

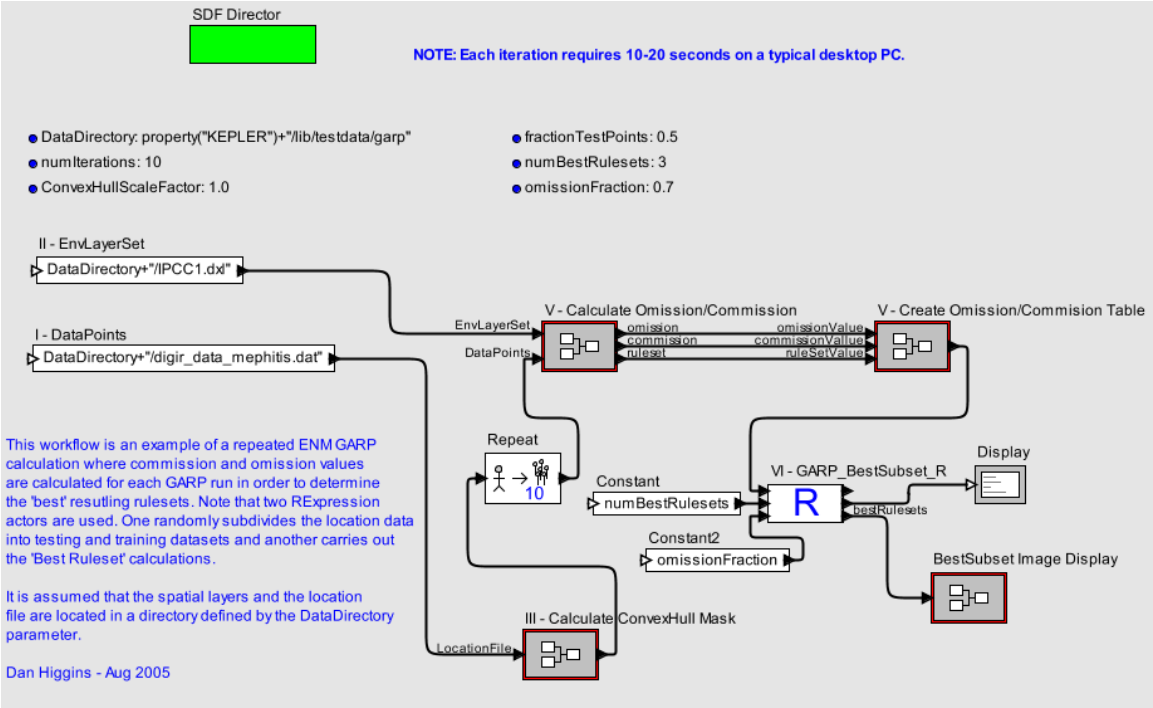


Figure 3 – Implementation of GARP Single Species Calculation. (This workflow is in the 'workflows/eco/' subdirectory under the name: "GARP_SingleSpecies_BestRuleSet_Aug2005.xml")

This implementation is organized slightly differently than the conceptual workflow but the same roman numerals are used to identify corresponding blocks.

Note that in this case, blocks I and II (creating known location lists and spatial layer information) have been reduced to simple file names. It is being assumed that various pre-processing has been done off-line and the resulting files are in the correct 'raw' format for GARP processing and have been stored in the indicated locations under the indicated names. Example files have been created from IPCC and Hydro1K data and saved in the locations shown. Details of how this was done is described along with the associated workflows elsewhere, but the point here is that this is a separate step that can be done prior to the multiple GARP runs.

Block III calculates the convex hull and creates a mask file limiting calculations to the area enclosed but this convex hull. If you "Look Inside" the compositeActor III in Figure 3, you will find the workflow shown in Figure 4.

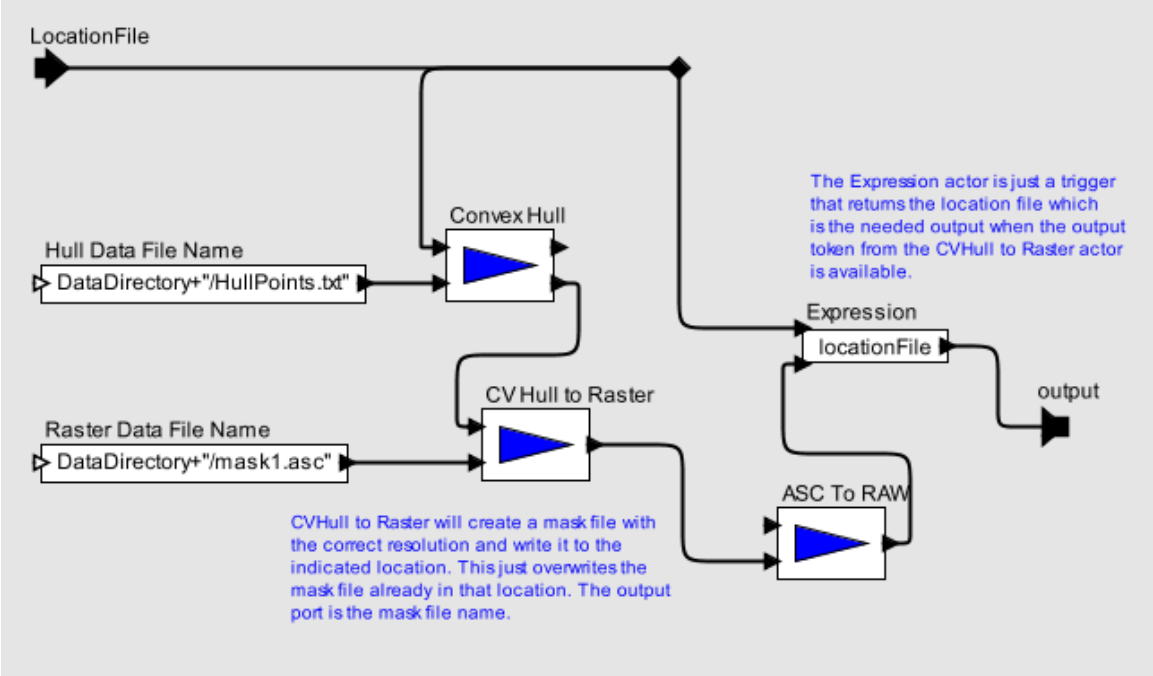


Figure 4 – Inside of Block III (Calculate Convex Hull Mask

This sub-workflow calculates the Convex Hull as a set of points, saves that to a file, then uses that file to create a spatial grid 'mask' file. Note that the *.asc grid must be converted to the *.raw format used by GARP. In this case, the mask1.raw file is just overwritten so that the XML-based IPCC1.dxl file, which summarizes all the spatial layers, need not be changed.

Looking inside Block V – Calculate Omission/Commission shows the sub-workflow of Figure 5. Although this part of the workflow basically just executes a single GARP calculation, there is some rather messy plumbing required. In particular, notice the "Ramp" actor. This actor is used to create an index that is added to the name of RuleSet file (e.g. "RuleSet5.xml" when the Ramp count is '5'), so that each is distinct.

There is also an RExpression actor which is used to randomly subdivide the location datapoints for every iteration. The R script used is:

```
# read the original data
df <- read.table(fileName)
# get number of rows (i.e. number of lines)
lll <- length(df$V1)
fraction <- 0.5
# create a list of subset indices
sss <- sample(1:lll, size=(fraction*lll))
# create 2 subsets
df1 <- df[sss,]
# write output file
write.table(df1, file="FirstSubset.txt", sep="\t", row.names=FALSE,
col.names=FALSE)
trainingLocations <- paste(getwd(),"/FirstSubset.txt",sep="")
trainingLocations
df2 <- df[-(sss),]
```

```
# write output file
write.table(df2, file="SecondSubset.txt", sep="\t", row.names=FALSE,
col.names=FALSE)
testingLocations <- paste(getwd(),"/SecondSubset.txt",sep="")
df1
df2
```

The primary command that creates the random samples is '`sample`'. The size of each sample is determined by the '`fraction`' variable that has a default value of `0.5`. This gives training subsets and testing subsets of the same size.

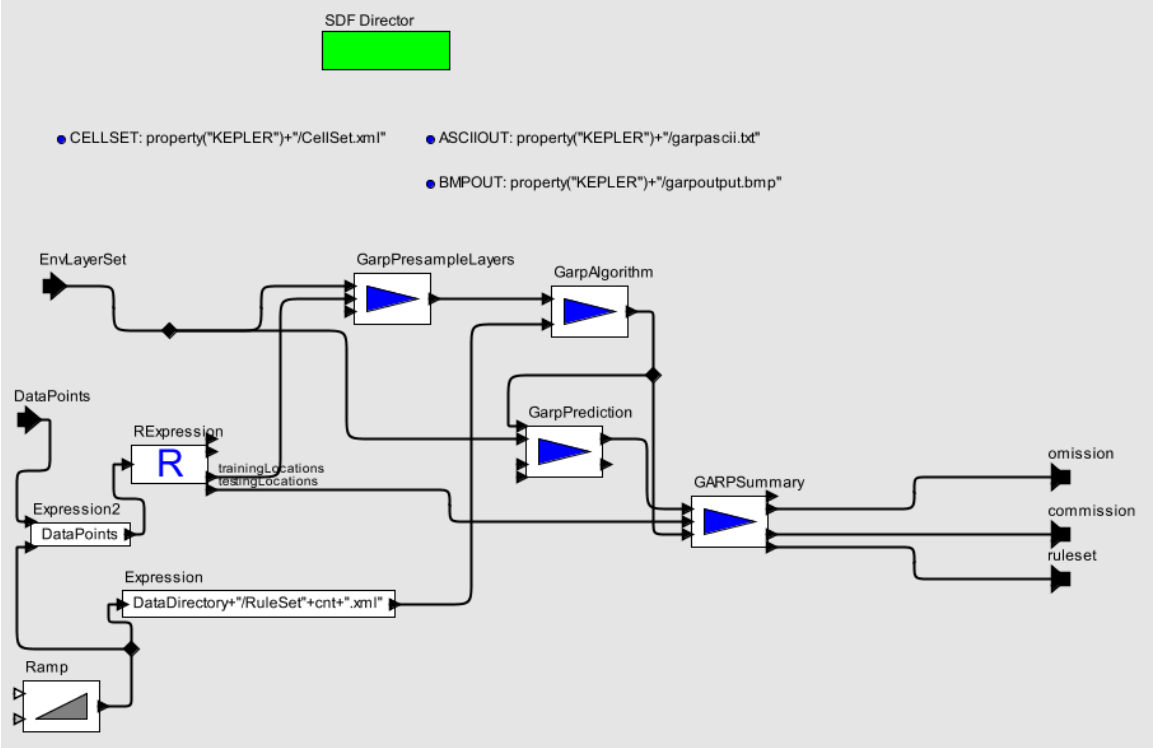


Figure 5 - Inside Block V – Calculate Omission/Commission

The GARPSummary actor actually calculates the omission and commission values and outputs them, along with the associated ruleset. Each set of these three parameters is then added as a 'row' in a table by the next composite in the top-level workflow (Figure 3). Looking inside this composite gives the subworkflow shown in Figure 6.

Figure 6 shows how a sequence of commission, omission, and ruleset values are first converted to arrays and the put into a Kepler/Ptolemy record. This record can be thought of as a three-column table. It can then passed to an REpression actor where it automatically becomes an R Dataframe.

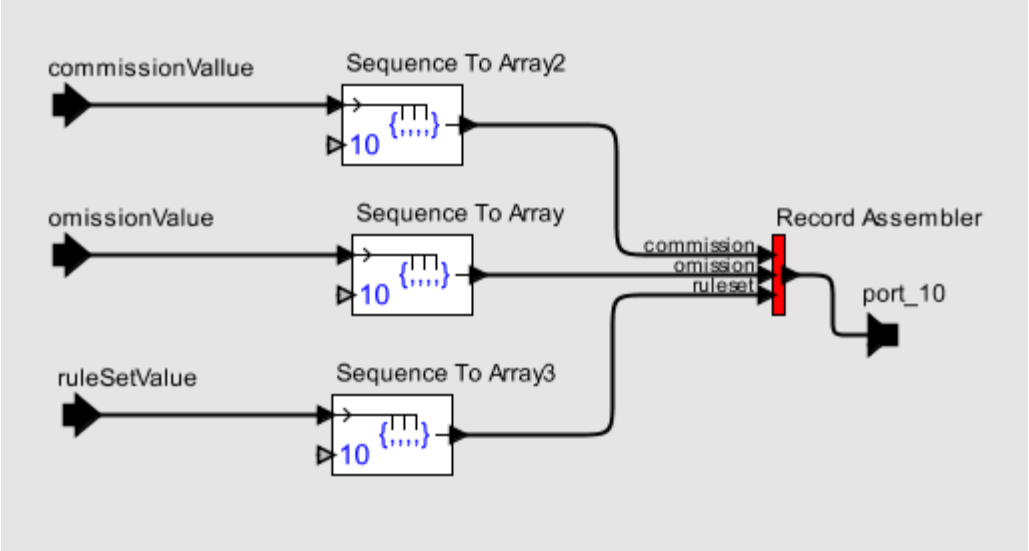


Figure 6 – Inside the Create Omission/Commission Table Composite Actor

Now consider the RExpression actor in Figure 3 (VI – GARP_BestSubset_R). The R script for this actor is:

```
df
df1 <- df[order(df$omission),]
df1
df2 <- df1[1:(fracOm*length(df1$omission)),]
df3 <- df2[order(df2$commission),]
df3
l <- length(df3$commission)
l
df4 <- df3[(0.25*l):(0.75*l),]
df4
cnt
vec <- as.vector(df4$ruleset)
bestRulesets <- vec[1:cnt]
```

This script sorts the table (dataframe df) by the `df$omission` column using the `order` function. It then creates a new table with the largest '`fracOm`' of the omission values. (The parameter '`fracOm`' is defined as a global parameter '`omissionFraction`'.) It then takes the middle 50% of the remaining values (i.e. the values from `0.25` to `0.75` of the length).

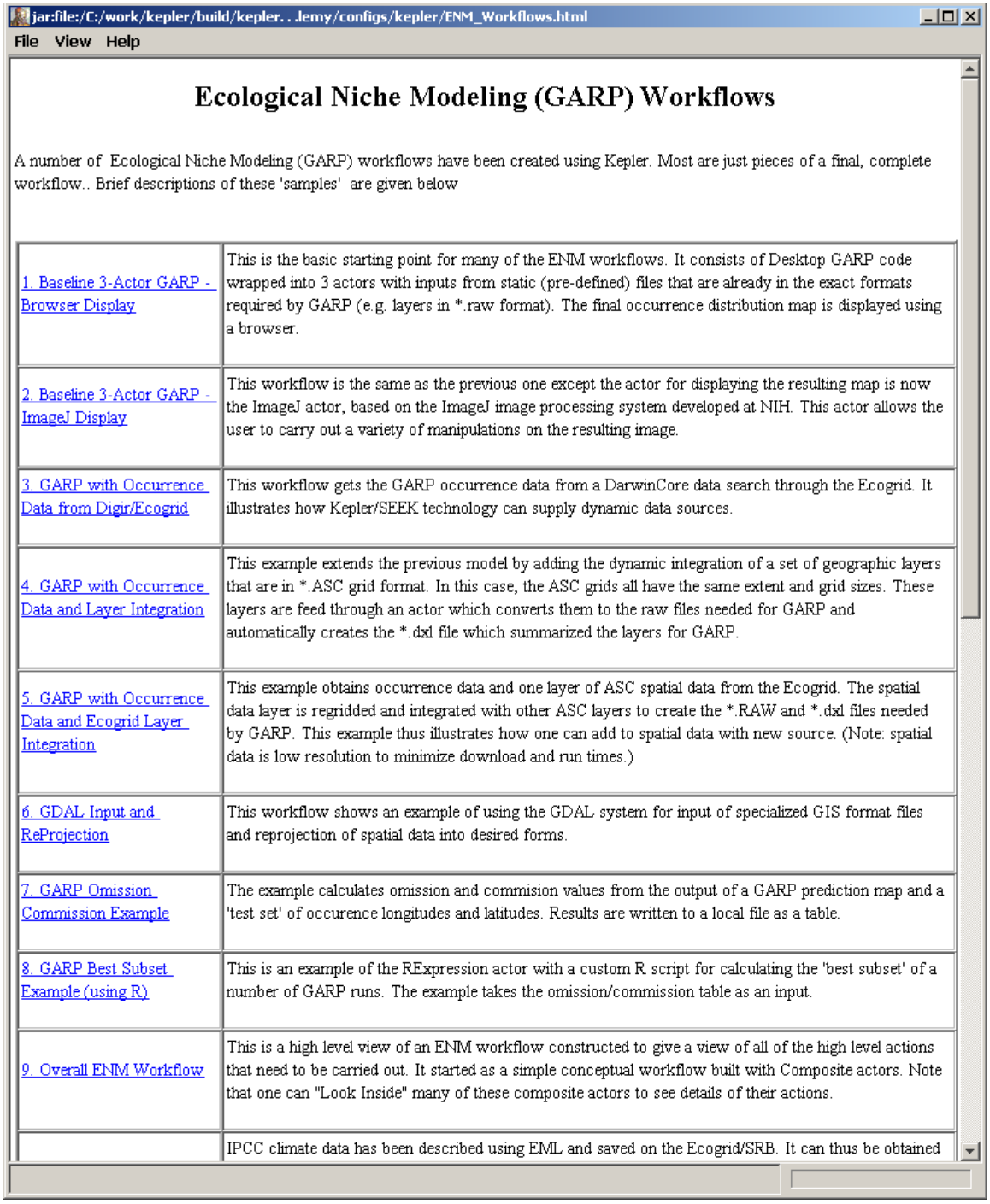
Finally, note the '`numIterations`' parameter near the top of Figure 3 with a displayed value of 100. The value of that parameter is the number of times the workflow is to be iterated. The SDF Director is set to repeat that number of times and the parameter is used in several other places in the workflow, including inside composite actors.

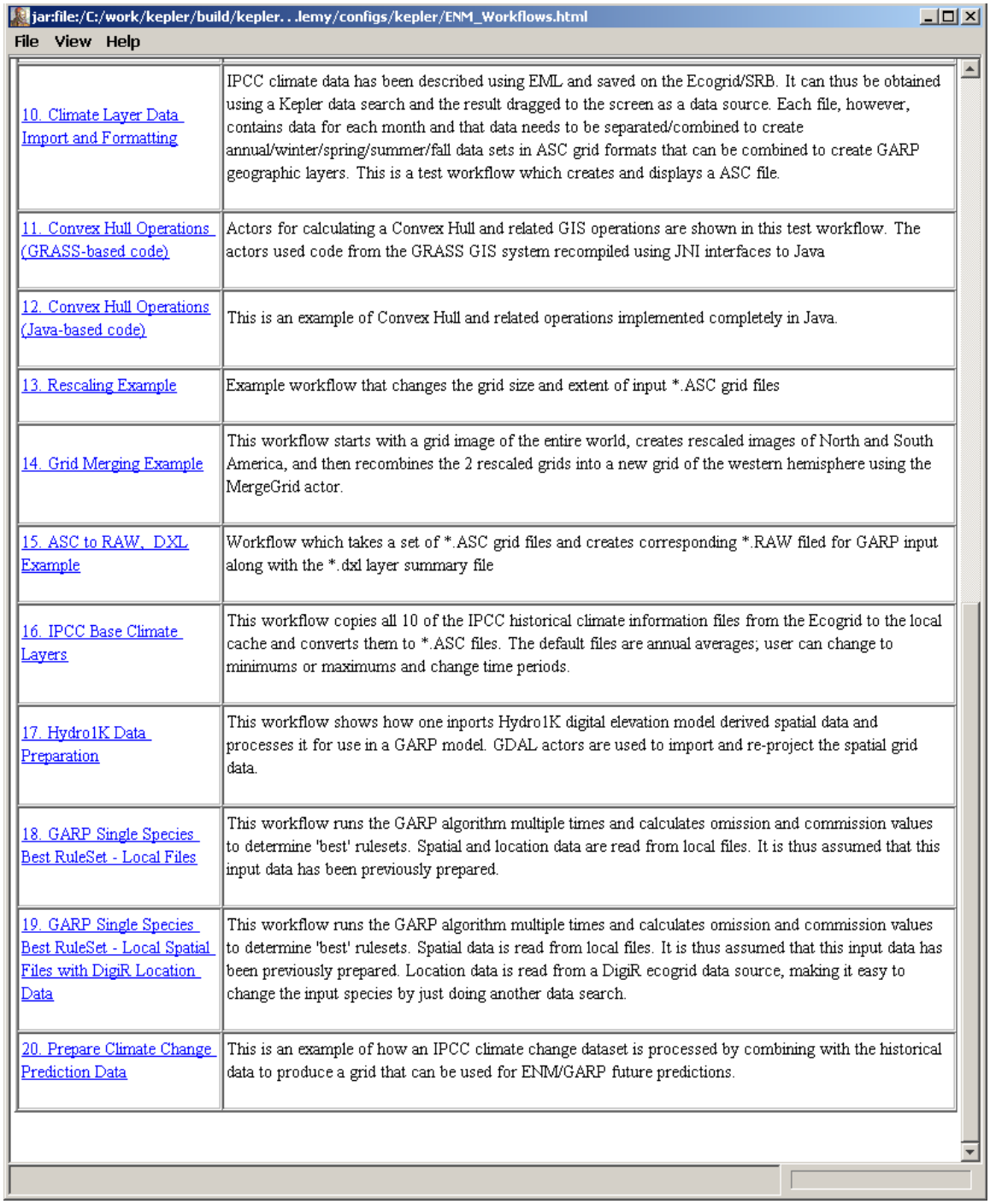
Note that supplying a simple constant file name for the environmental layer lists a total of '`numIterations`' times is not computational intensive, but we really don't want to calculate the same convex hull repeatedly. Thus, the 'Repeat' actor appears in Figure

3. This actor just repeats the token from the previous actor ‘`numIterations`’ times and thus avoids repeating the previous calculation.

Example ENM/GARP Workflows in Kepler

There is a link on the initial Kepler startup screen to another HTML display that has a table linked to various ENM/GARP workflows that illustrate how to carry out various parts of ENM/GARP predictions. These example workflow descriptions are shown below:





10. Climate Layer Data Import and Formatting	IPCC climate data has been described using EML and saved on the Ecogrid/SRB. It can thus be obtained using a Kepler data search and the result dragged to the screen as a data source. Each file, however, contains data for each month and that data needs to be separated/combined to create annual/winter/spring/summer/fall data sets in ASC grid formats that can be combined to create GARP geographic layers. This is a test workflow which creates and displays a ASC file.
11. Convex Hull Operations (GRASS-based code)	Actors for calculating a Convex Hull and related GIS operations are shown in this test workflow. The actors used code from the GRASS GIS system recompiled using JNI interfaces to Java
12. Convex Hull Operations (Java-based code)	This is an example of Convex Hull and related operations implemented completely in Java.
13. Rescaling Example	Example workflow that changes the grid size and extent of input *.ASC grid files
14. Grid Merging Example	This workflow starts with a grid image of the entire world, creates rescaled images of North and South America, and then recombines the 2 rescaled grids into a new grid of the western hemisphere using the MergeGrid actor.
15. ASC to RAW, DXL Example	Workflow which takes a set of *.ASC grid files and creates corresponding *.RAW filed for GARP input along with the *.dxl layer summary file
16. IPCC Base Climate Layers	This workflow copies all 10 of the IPCC historical climate information files from the Ecogrid to the local cache and converts them to *.ASC files. The default files are annual averages; user can change to minimums or maximums and change time periods.
17. Hydro1K Data Preparation	This workflow shows how one inports Hydro1K digital elevation model derived spatial data and processes it for use in a GARP model. GDAL actors are used to import and re-project the spatial grid data.
18. GARP Single Species Best RuleSet - Local Files	This workflow runs the GARP algorithm multiple times and calculates omission and commission values to determine 'best' rulesets. Spatial and location data are read from local files. It is thus assumed that this input data has been previously prepared.
19. GARP Single Species Best RuleSet - Local Spatial Files with DigiR Location Data	This workflow runs the GARP algorithm multiple times and calculates omission and commission values to determine 'best' rulesets. Spatial data is read from local files. It is thus assumed that this input data has been previously prepared. Location data is read from a DigiR ecogrid data source, making it easy to change the input species by just doing another data search.
20. Prepare Climate Change Prediction Data	This is an example of how an IPCC climate change dataset is processed by combining with the historical data to produce a grid that can be used for ENM/GARP future predictions.