

Kepler - Bug #1332

GARP improvements and further implementation

02/09/2004 08:30 AM - Chad Berkley

Status:	Resolved	Start date:	02/09/2004
Priority:	Immediate	Due date:	
Assignee:	Dan Higgins	% Done:	0%
Category:	pipelines	Estimated time:	0.00 hour
Target version:	1.0.0alpha6	Spent time:	0.00 hour
Bugzilla-Id:	1332		
Description The GARP pipeline needs to be extended to include the entire workflow from start to finish. This has been prototyped but not implemented. also need to modify the garp actors so that the inputs and outputs are more intuitive and use streams instead of filenames. expose garp params (look at desktop garp) ----- runs convergence limit max iterations --develop xml schema for the internal garp data rep. (cellset and ruleset xml files) --get dave's and ricardo's garp code integrated with the kepler garp code (use a single repository) --standardize IO file format			
Related issues:			
Has duplicate SEEK - Bug #1720: Create an example Kepler workflow for ENM/GAR...		Resolved	10/21/2004
Blocks Kepler - Bug #1338: find GIS system for inclusion in kepler		Resolved	02/09/2004
Blocks Kepler - Bug #1665: Review ENM/GARP pipeline so it works with openMode...		Resolved	08/31/2004

History

#1 - 02/09/2004 08:32 AM - Chad Berkley

the prototype garp model

#2 - 02/09/2004 08:33 AM - Chad Berkley

this file needs to be unzipped into the kepler/dynsrc directory for the full-garp.xml model to work.

#3 - 02/10/2004 10:24 AM - Chad Berkley

This is a list of the actors that need to be implemented to complete the preprocessing of the GARP pipeline. These are actors that would be needed to take a raw DiGIR query and get it into the form needed to submit to the current GARP actors.

--DiGIRQuery: actor to query DiGIR for a set of species. This should probably be done via the ecogrid.

--EnvLayerMetadatExtractor: extracts the metadata from the env layers needed to get user input. the user's input includes: scaleValue, minSampleSizeThreshold, SubSampleMethod (choose from medians, quartiles, infinite or manual), numberOfReplicates (how many times to run the garp algorithm).

--RescaleValues: This actor rescales the presence data values from their current form to a value between 1 and 254. 0 and 255 are reserved and not used. this could probably be implemented in java, though i'm not sure of the heterogeneity of the input files....it might be better to use a gis system if there are a variety of input formats.

--Clipping: For each species, this actor creates a minimum convex polygon of the species plots (lats/longs). It then buffers the polygon to n times the area of the polygon. the environmental layers are then clipped to the size of the new polygon. This should probably be implemented in a GIS system.

--Sampling: take in a list of species and occurrence data. randomly sample the data into two sets: intrinsic data and extrinsic data. the intrinsic data is used in the actual garp pipeline, the extrinsic data is used to test the garp output for omission and commission error.

--Rescale (general binning utility): Take in a spatial file and rescale to a specific scale. this probably needs to be implemented in a gis...possibly could write java to do it.

--SpeciesOccuranceCount: take in a list of species with occurrence data that is adjusted to the same scale. output a vector of species and counts.

--SpeciesSelection: rescale a list of species coverages to a single scale. select the species that have an occurrence count greater than a user defined threshold. this should output a list of rescaled species (with occurrence data) that have an occurrence rate greater than the threshold. This can probably be implemented in a composite actor using existing (or soon to exist) actors.

#4 - 02/25/2004 10:26 AM - Chad Berkley

I have decided to implement the GRASS functionality needed by garp in web services because of the difficulty in getting grass to work on multiple platforms. I have created web services for the following grass functions: s.in.ascii, s.info, r.in.arc, r.info, s.hull, v.info, v.support, v.to.rast, r.mapcal, r.out.arc. I have produced a shell script which makes use of these services to correctly clip a convex hull as described by Town at the SEV meeting. I now need to implement this as a workflow in kepler, but I am waiting for the web service actor to be extended to allow for the returning of objects.

The web services are currently running on trestles and the wsdl is available at <http://trestles.nceas.ucsb.edu:8004/seek/grass.wsdl>. I am running the services on the GLUE server, but could switch over to AXIS if needed.

#5 - 02/27/2004 12:01 PM - Chad Berkley

Here is a list of the web service method names and the grass executable that they use:

WS method	GRASS exe
importArcToGrass	r.in.arc
exportGrassToArcFile	r.out.arc
importAsciiData	s.in.ascii
asciiDataInfo	s.info
rasterDataInfo	r.info
vectorDataInfo	v.info
createHull	s.hull
createVectorSupportObjects	v.support
convertVectorToRaster	v.to.rast
calculateMap	r.mapcalc

#6 - 06/14/2004 02:31 PM - Chad Berkley

dan is now working on GARP

#7 - 01/20/2005 09:19 AM - Chad Berkley

moving to beta1

#8 - 01/21/2005 12:09 PM - Chad Berkley

the bugs for this are all in the seek bugzilla.

- This bug has been marked as a duplicate of 1720 ***

#9 - 03/27/2013 02:16 PM - Redmine Admin

Original Bugzilla ID was 1332

Files

full-garp.xml	63.2 KB	02/09/2004	Chad Berkley
dynsrc.tar.gz	4.89 KB	02/09/2004	Chad Berkley