

Connectivity Analysis Toolkit

User guide

Table of Contents

Connectivity Analysis Toolkit	3
License	4
Version history	5
Getting Started	7
Overview of Methods	9
Three Methods for Connectivity Analysis	10
Subset Centrality: Mapping Linkages	12
All-pairs Centrality: Mapping Landscape Connectivity	14
Network Flow Analysis of Time Series	16
Input files	18
Output files	20
Running the Toolkit from the GUI	22
Hexmap Tab	23
Graphs Tab	25
Connectivity Tab	27
Output Tab	30
Running the Toolkit from the Command Prompt	31
Flowchart of analysis process	33
Examples	34
Example 1: Shortest-path betweenness centrality (BC)	35
Example 2: Shortest-path betweenness subset centrality	36
Example 3: Comparison of subset and all-pairs current flow BC	37
Example 4: Comparison of shortest-path and current flow BC	38
Tutorial	39
Exercise 1 - Mapping linkages with min-cost-max-flow subset centrality	40
Exercise 2 - Mapping linkages with shortest-path subset centrality	42
Exercise 3 - Mapping landscape connectivity with all-pairs centrality	43
Exercise 4 - Mapping connectivity under climate change: analysis of time series	44
Troubleshooting	46
Related Resources	47
System requirements	49
Contacts & Credits	50

Connectivity Analysis Toolkit

The Connectivity Analysis Toolkit is a software interface that provides conservation planners with advanced tools for linkage mapping and centrality analysis.

Why is connectivity important?

Conservation biologists have long recognized that natural areas that are linked into networks are often more effective than isolated areas at preserving certain components of biodiversity (Harris 1984, Noss and Harris 1986). Although in the past connectivity usually was considered in a static and purely structural sense as habitat corridors, we now recognize that planning for connectivity requires incorporating process as well as pattern into conservation planning, and that connectivity can be achieved by means other than corridors. There is increased awareness of the importance of ecological and demographic dynamics, including the dynamics resulting from human-associated landscape change, in determining the persistence of biodiversity.

Considering connectivity in landscape planning can facilitate:

- 1) Maintenance of demographic flows - The traditional focus of connectivity, e.g., rescue effect (Brown and Kodric-Brown 1977) of dispersal on populations inhabiting small patches.
- 2) Maintenance of genetic flows - Avoidance of inbreeding depression, long-term maintenance of genetic adaptability, diversification of evolutionary lineages, etc. through dispersal (Young and Clarke 2000)
- 3) Resilience of populations to landscape conversion by agriculture, forestry, and development, by evaluating the adequacy of reserve networks in the light of landscape-level threat processes.
- 4) Resilience of populations to climate change, by designing reserve networks that allow species to track suitable habitat (e.g., altitudinal migration) as it shifts in response to climate change
- 5) Maintenance of nutrient and hydraulic flows in freshwater systems
- 6) Maintenance of characteristic ecosystem processes necessary for population persistence - e.g., patch dynamics of disturbance or resources.

The focus of connectivity planning has increasingly shifted from preserving static elements of landscape structure such as corridors to facilitating these ecological processes. Structural connectivity is no longer seen as a distinct conservation goal, but rather as a means to maximize retention of functional connectivity, and thus achieve broader conservation goals such as species' persistence (Pressey et al. 2007).

How can connectivity be assessed during the landscape planning process?

A group of landscape metrics based on graph theory, termed centrality metrics, provides a means to quantitatively incorporate connectivity within the planning process by ranking the importance of sites as gatekeepers for flow across a landscape network. Computational advances now allow such metrics to be applied to landscapes of continuous habitat gradients represented as lattice-based graphs with large numbers of nodes. This approach avoids both the binary classification of landscapes into patch and matrix required by patch-based graph analyses, and the focus on single paths between individual pairs of source and target patches characteristic of most corridor mapping methods. The Connectivity Analysis Toolkit allows users to develop and compare three contrasting centrality metrics based on input data representing habitat suitability or permeability, in order to determine areas where conservation measures might facilitate connectivity and dispersal for a particular species.

The Connectivity Analysis Toolkit also allows application of these approaches to the more commonly considered problem of mapping habitat linkages between pairs of source and target patches. This represents a special case of centrality analysis, termed subset centrality, that also may benefit from comparison of results from the different metrics contained in the Toolkit.

License

Connectivity Analysis Toolkit (CAT) Manual, Version 1.3

Copyright (c) 2013

Carlos Carroll, Klamath Center for Conservation Research

This documentation is provided both as html help and as a pdf manual found in the directory to which the program is installed.

The software and a copy of this documentation can be downloaded from:

<http://www.connectivitytools.org>

An appropriate citation for this software is:

Carroll, C., B. McRae, and A. Brookes. 2012. Use of Linkage Mapping and Centrality Analysis Across Habitat Gradients to Conserve Connectivity of Gray Wolf Populations in Western North America. *Conservation Biology* 26:78-87.

Connectivity Analysis Toolkit is based in part on portions of the following software modules:

Hexsim, developed by Nathan Schumaker et al.: <http://www.epa.gov/hexsim>

LEMON, developed by Egervary Combinatorial Optimization Research Group, <http://lemon.cs.elte.hu/>

NetworkX, developed by Aric Hagberg et al.: <http://networkx.lanl.gov/>

Those portions of the Connectivity Analysis Toolkit developed from the LEMON and NetworkX modules are covered by the following copyright/license.

Copyright (c) 2013 Carlos Carroll, Klamath Center for Conservation Research

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This software is provided by the developers "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall developers or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Version history

Version 1.0 August 10, 2010

- Initial release

Version 1.1 October 14, 2010

- Major enhancement: Implementation of time-series min-cost-max-flow.
- Several minor bugfixes: e.g., GUI now passes correct alpha value to PageRank.

Version 1.12 October 18, 2010

- Fixed configuration issue between the python libraries py2exe and elementtree that caused python-based functions (betweenness centrality, etc.) to fail with ImportError message.

Version 1.2 October 24, 2011

- Approximate shortest-path betweenness centrality allows faster computation of this metric
- Approximate current flow betweenness centrality allows faster computation of this metric; function also uses sparse matrices for lower RAM requirements
- Network flow functions updated to LEMON version 1.2.2
- Updates to manual and tutorial dataset

Version 1.2.1 April 2, 2012

- Updated python and python packages, including NetworkX to version 1.6.
- Current flow betweenness centrality subset also now uses sparse matrices for lower RAM requirements. (Note that there is no approximate function for current flow betweenness centrality subset, so it is still slow for large datasets).

Version 1.3 July 12, 2013

- Updated python and python packages, including NetworkX to version 1.8dev.
- Updated Hexsim-based functions to Hexsim version 2.5.1, which improves conversion of .asc input file to hxn file.
- Added support for locales (e.g., areas in Europe) that use a comma instead of a period as decimal separator.

Version 1.3.1 August 19, 2013

- Updated LEMON to version 1.3.

- Turned off arc mixing in network flow functions to resolve potential division by zero error.
- Updated NetworkX to version 1.8.1.
- Updated Hexsim-based functions to Hexsim version 2.5.3.

Planned future enhancements: Better support for multi-threaded implementation of network flow due to release of LEMON 1.3.

Getting Started

Download the 32-bit or 64-bit version of the Connectivity Analysis Toolkit from www.connectivitytools.org. If you have a 64-bit Windows operating system, you can install either version. You may want to install the 64-bit version if you have more than 3GB of RAM.

Although most analyses can be performed at acceptable resolutions with the 32-bit version, the 64-bit version may be preferable if you are attempting analyses at high spatial resolution over larger extents (resulting in a graph with large numbers of nodes)(see System Requirements).

Installation

The Toolkit is supplied as a self-extracting installer. Run the installer and either select the default options or choose a custom installation directory. You will want to copy the 'Tutorial' directory to another location on your system in order to work through the tutorial analyses. Installation of the Toolkit may require administrator privileges on some systems. After installation, start the Toolkit from the Start Menu or Desktop shortcut.

Workflow

Typical workflow begins with import of an ASCII (.asc) file created from habitat data in a GIS.

The Toolkit processes this raster data into a vector format (a shapefile) and associated hexagon file (.hxn, a file format used by the Toolkit) and node coordinate file.

The hexagon file is then used to produce a textfile representing a graph which has a node for each of the hexagons in the shapefile.

The graph data is then written in one of two 'graph file formats', the edgelist format accepted by the NetworkX module or the LEMON graph format (lgf) required by the LEMON module ([see below](#)).

The user then performs one of several types of centrality analysis on the graph and produces output giving node centrality values.

This output file can then be joined to the shapefile using a GIS software such as ESRI's ArcGIS.

Note that graph format files created in another application (e.g., as representations of patch networks rather than landscape lattices) may also be analyzed using the Toolkit's 'Connectivity' tab.

Why can't this all be integrated into a GIS software?

While modern GIS software such as ESRI's ArcGIS are powerful tools, and ArcGIS in particular is increasingly integrated with scripting languages such as Python, important limitations remain. For example, the Toolkit's graph analysis algorithms require highly optimized versions of Python's numerical libraries that may conflict with the versions used by ArcGIS.

Why can't this all be done without GIS software?

We have focused on creating a Toolkit with advanced connectivity analysis methods that are not available in GIS software, rather than trying to duplicate GIS functions available in other software. We assume users will have access to software such as ArcGIS to create .asc files, identify source and target areas for subset centrality analysis, and perform post-processing of output, such as joining centrality output files to the shapefile.

But the Toolkit's methods may take hours, whereas I can map corridors in a GIS in minutes. Why is it worth the trouble?

We make the case in the remainder of this manual that the Toolkit's connectivity analysis methods can complement methods commonly available in GIS, and allow planners to better evaluate alternate assumptions on how to represent wildlife movement and ecological processes. Land-use planning for biodiversity conservation necessarily involves decisions that have large economic and social impacts, and thus merits

use of the most rigorous and informative tools available. The analyses described in this manual, although they do sometimes require substantial computation time (see System Requirements), would not have even been possible on desktop computer systems as recently as 2 years ago. Versions of the software starting with version 1.2 include approximation-based algorithms that greatly increase the speed of analyses when compared to earlier versions of the software. The scope and resolution of analyses feasible with the Toolkit's methods will continue to expand over the coming years, as the algorithms underlying these methods are under active development.

Overview of Methods

The connectivity analysis methods in this Toolkit are based on the concept of 'centrality'. Centrality analysis encompasses a diverse group of metrics that provide information on the relative role of nodes in facilitating movement across a graph, i.e., as 'gatekeepers' for functional connectivity (Bunn et al. 2000; Borgatti 2005). Metrics that consider the role of a node in mediating flow between all other nodes must consider all possible pairwise combinations of nodes, and thus increase in computational complexity at a polynomial rate (typically quadratic to cubic) as the number of nodes increases (Ahuja et al. 1993). Thus their previous use in connectivity analysis has been limited to contexts where a landscape can be represented a system of patches (numbering at most in the hundreds) embedded in a matrix of non-habitat.

Computationally-efficient algorithms for analysis of large networks, which have recently been developed for purposes such as ranking web pages on the internet, now allows analysis of landscape connectivity at a resolution that makes simplifying assumptions less necessary (Hagberg et al. 2008). This facilitates application of centrality metrics to landscape and species contexts where a continuous habitat gradient is more ecologically relevant than a binary patch-matrix framework (Cushman et al. 2010). Within the limits of computational feasibility, resolution of the landscape lattice can be scaled to match the scale of habitat selection of the species of interest. Below we first describe three alternative methods of delineating habitat linkages between a single source and target patch (least-cost path, current flow, and network flow). We then describe analogous centrality metrics that analyze connectivity across a landscape without reference to specific source and target patches.

For more information on the methods used in the this software, see Carroll, C., B. McRae, and A. Brookes. 2012. Use of Linkage Mapping and Centrality Analysis Across Habitat Gradients to Conserve Connectivity of Gray Wolf Populations in Western North America. *Conservation Biology* 26:78-87. Available at [the journal website](#).

Three Methods for Connectivity Analysis

This software allows users to apply three types of linkage mapping methods: shortest or least-cost path, current flow, and network flow (specifically min-cost-max-flow). These methods consider respectively 1) the single shortest path, 2) probabilistic flow across all possible paths, and 3) optimal flow which considers, but may not use, all possible paths.

3 pairwise linkage mapping methods

We can illustrate the contrast between these methods with a simple graph with 5 nodes and 6 edges (below).

a) Edge values shown here may be derived from habitat suitability models, and may be represented as proportional to conductance (circuit flow) and flow capacity (network flow), and inversely proportional to cost (not shown); b) A shortest- or least-cost path solution assigns all priority to the path A-C-E; c) current flow analysis with a 3 ampere source at A identifies edges A-C and D-E as showing highest current flow, with all other edges showing lower but non-zero current levels; d) Network flow maximum-flow analysis identifies both a high-capacity path A-C-E and a lower capacity path A-B-D-E, but none of the maximum flow will transit edge C-D. If all edges had the same cost, the min-cost-max-flow solution from A to E would be identical to the max-flow solution on this simple graph.

3 analogous centrality metrics

The 3 centrality metrics considered here are variants of betweenness centrality (BC), in that they measure to what extent a node contributes to paths or flows between all other nodes (Borgatti & Everett 2006; Newman 2010). Shortest-path BC identifies the one or several shortest (geodesic) paths that connect each pair of nodes on a graph and counts the number of such shortest paths in which a node is included (Borgatti & Everett 2006). Current-flow BC assesses the centrality of a node on the basis of how often, summed over all node pairs, the node is traversed by a random walk between 2 other nodes (Newman 2005). Minimum-cost-maximum-flow BC evaluates a node's contribution to connectivity on the basis of portion of the minimum-cost-maximum-flow that must pass through that node, summed over all node pairs (Freeman et al. 1991). In the figure below, shortest-path BC (e) resembles shortest-path results between a node pair (b) because it assigns high centrality to node C, which lies on the shortest path between many node pairs, and zero centrality to nodes (B, E), which do not lie on the shortest paths between any pair of nodes. Current-flow BC (f) ranks the importance (for facilitating flow) of nodes similarly as does shortest-path BC, but centrality values are more evenly distributed among nodes and there are no nodes of zero centrality due to the model's random-walk behavior. Maximum-flow BC ranks nodes similarly to current flow BC, but values are distributed more evenly (g). If all edges have equal cost, results of minimum-cost maximum-flow BC (not shown) resemble maximum-flow BC.

Graphs are classified as either directed or undirected dependent on whether the edge weight (data representing attributes such as cost, conductance, or capacity) of edge $i-j$ (from node i to j) equals the weight of the edge $j-i$. All the metrics described here, with the exception of those based on time series analysis, can use undirected graphs with edge weights derived from the mean weight of the two end nodes connecting that pair. Untransformed habitat values from a habitat model may be used as weights for current flow and network flow algorithms. The reciprocal of habitat value is typically used to represent cost in calculating the least-cost or betweenness metrics, although other transformations are possible as well.

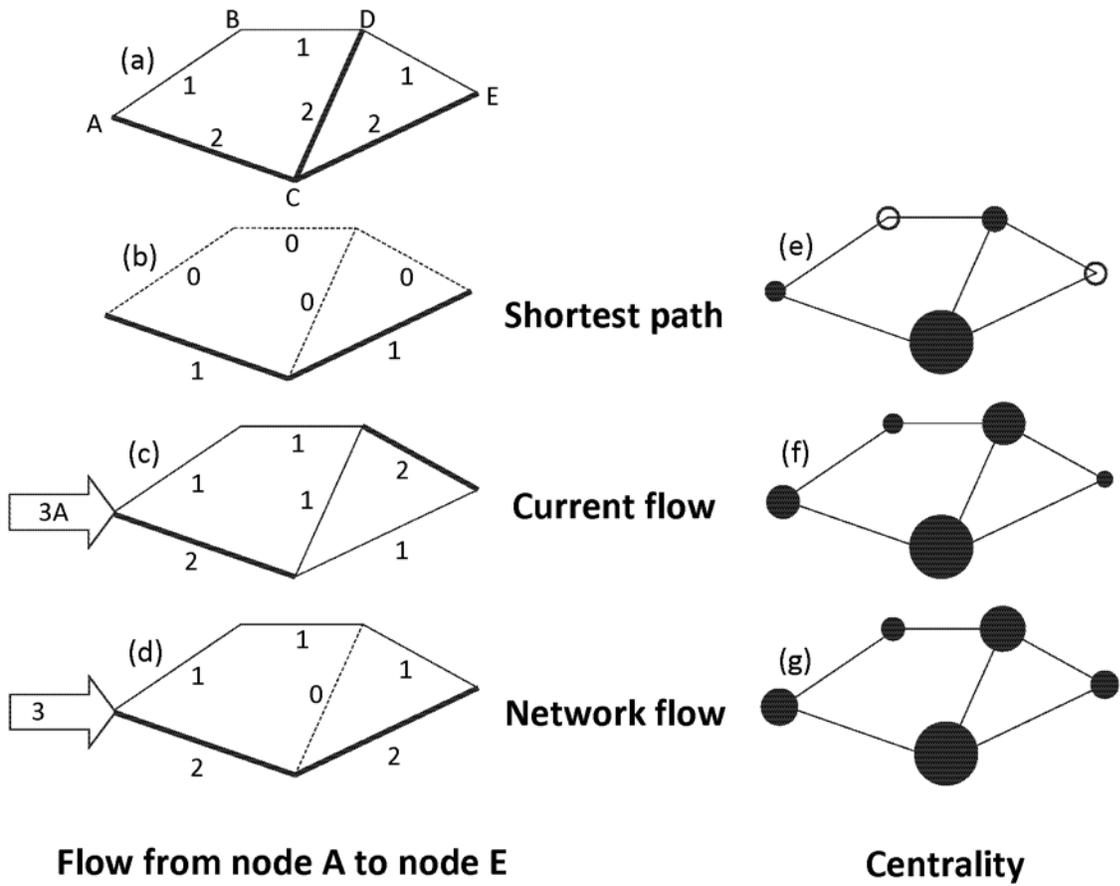


Figure adapted from Carroll et al. 2011

Subset Centrality: Mapping Linkages

Methods used for mapping linkages between a source and target patch represent a special case of centrality analysis, termed subset centrality, which considers paths between the subset of nodes that fall within the source and target patches (Hagberg et al. 2008).

Pairwise method	Betweenness Centrality (BC)	Edge attributes
Shortest or least-cost path	Shortest-path BC	Cost; distance
Current flow	Current flow BC	Conductance; resistance
Maximum flow	Flow BC	Capacity
Min-cost-max-flow	Min-cost-max-flow BC	Cost and capacity

The figure below shows an example analysis of habitat connectivity for wolves between central Idaho and Yellowstone National Park (Carroll et al. in review).

Shortest- or least-cost path betweenness centrality (BC) (shown in a) identifies a minimal network of linkages analogous to those identified by least-cost-path corridor mapping methods.

Current flow BC (b) identifies a more diffuse landscape network that may allow prioritization of redundant linkages.

Min-cost-max-flow BC (c) integrates consideration of both land cost and habitat capacity. In this example, a simple cost value was used that assigned private lands four times the cost of public lands, reflecting common policies that seek to develop options for species conservation on public lands before regulating management of private lands. The results this identify a linkage allowing maximum-flow that lies preferentially on public lands.

We recommend that planning efforts focused on connecting a single pair of core areas (subset centrality) compare results from the 3 methods to identify primary and alternative linkage options. In the example shown below, the comparison suggests it would be informative to evaluate 2 alternative or complementary linkage zones. In minimum-cost-maximum-flow sensitivity analyses, the southern linkage zone, which is longer than the northern linkage zone but contains less private land, received increasing priority as the difference in cost between public and private land increased (not shown). Unlike shortest-path analyses, which may combine land cost and habitat quality into a single aggregate index, minimum-cost-maximum-flow incorporates the 2 as distinct criteria, facilitating such sensitivity analyses.

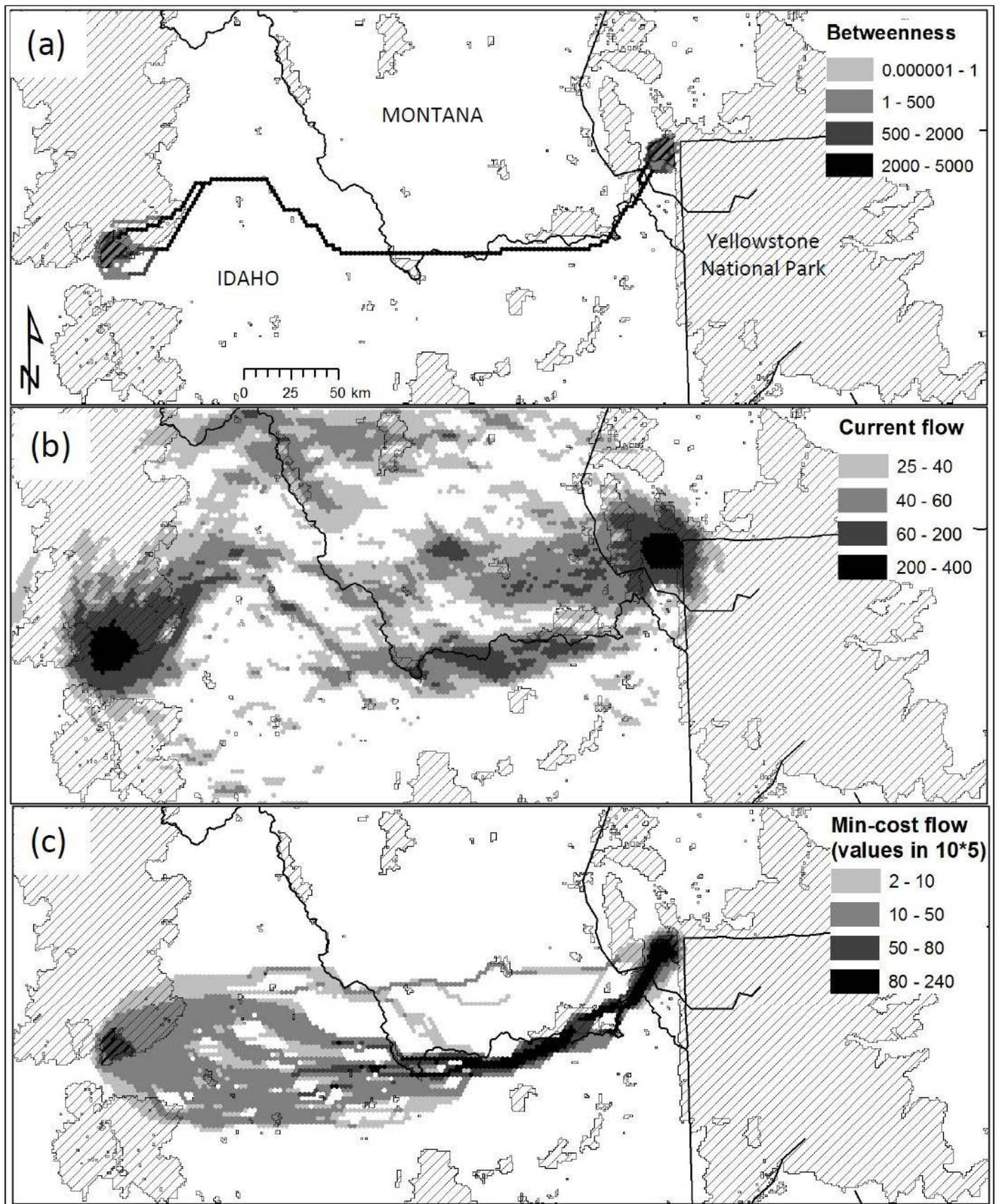


Figure adapted from Carroll et al. 2011.

All-pairs Centrality: Mapping Landscape Connectivity

Three centrality metrics, which consider connections between all pairs of sites (nodes) on a landscape, are analogous to the linkage mapping methods described above: shortest-path betweenness, current flow betweenness, and min cost flow betweenness (Ahuja et al. 1993; Newman 2005; Freeman et al. 1991).

As described above, shortest-path betweenness centrality identifies the one or several shortest (geodesic) paths that connects each pair of nodes on a graph, and counts the number of such shortest paths in which a node participates (Borgatti & Everett 2006). The loss of a node that lies on a large proportion of the shortest paths in the network would disproportionately lengthen distances or transit times between nodes (Brandes 2001). Like least-cost path, betweenness centrality inherently assume optimal paths and thus global knowledge of landscape by the dispersing individual (Freeman 1977).

Current flow or random-walk betweenness centrality assesses the centrality of a node by how often, summed over all node pairs, the node is traversed by a random walk between two other nodes (Newman 2005). Thus this metric counts essentially all paths between nodes, not just the shortest, and is analogous to the behavior of electrical flow in circuits (McRae et al. 2008).

Approximation of shortest-path and current-flow betweenness centrality

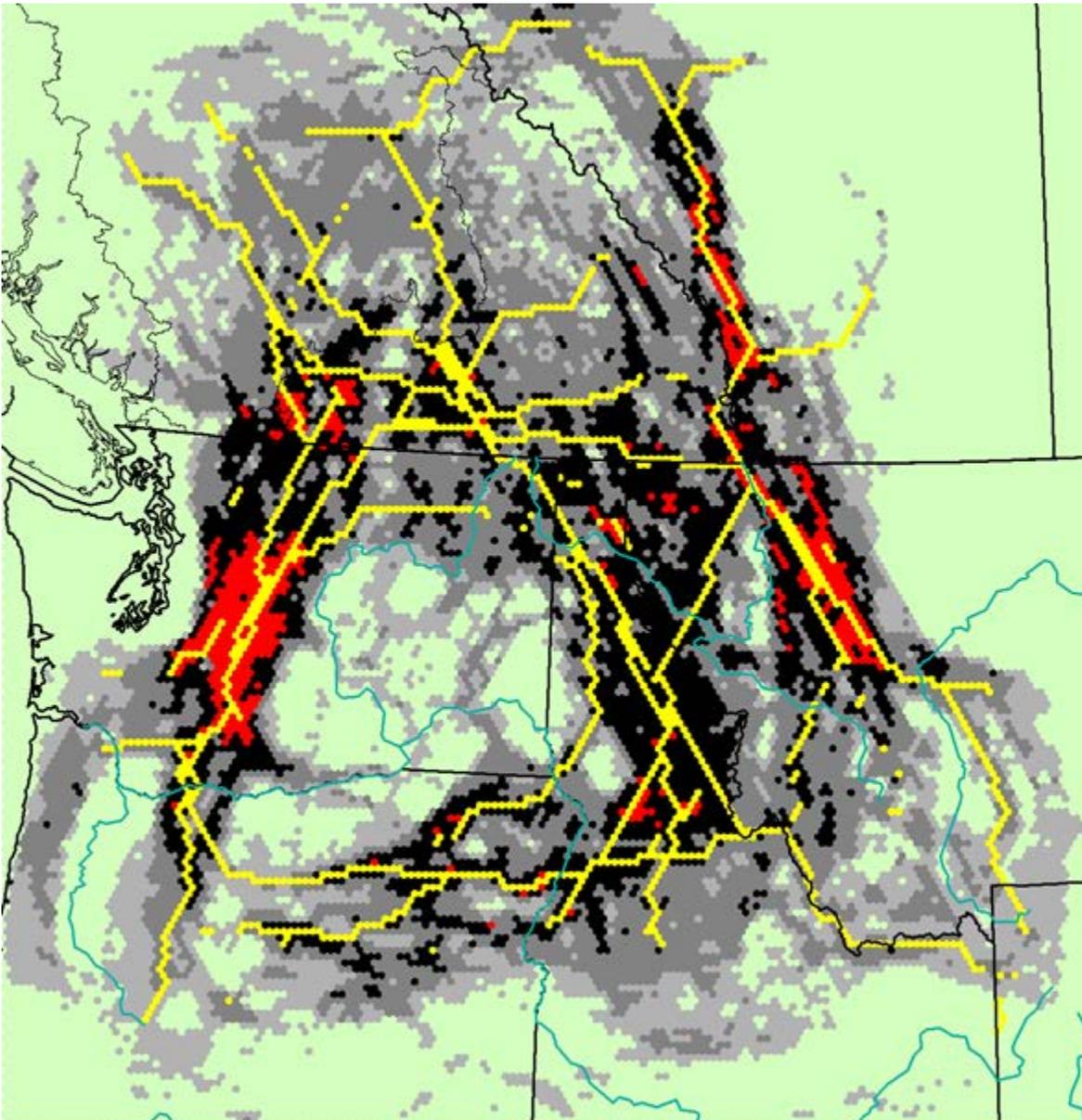
Beginning in version 1.2, the CAT's python-based all-pairs centrality functions (shortest-path BC and current-flow BC) by default use algorithms that approximate the exact solutions (Brandes and Fleischer 2005, Brandes and Pich 2007). The exact algorithms also remain available as described below. When used with the default error tolerance values, these approximations provide solutions that are correlated by >99% with the exact solutions, while requiring between 0.1 and 10% of the computational time needed by the exact solutions. This allows BC metrics to be derived for large graphs that were not previously computationally feasible to analyze.

Network flow centrality

Several centrality metrics based on network flow have been proposed in the field of social network theory (Freeman et al. 1991; Borgatti and Everett 2005). In a max-flow algorithm, each edge is assigned a flow which maximizes total flow between a pair of nodes. Flow betweenness centrality bases the importance of a node on the portion of the maximum flow which must pass through that node, summed over all node pairs (Freeman et al. 1991; Borgatti and Everett 2006). However, in graphs such as the landscape lattices considered here, where all nodes have similar numbers of edges, a large number of equivalent maximum flow paths exist. We have found that the flow betweenness centrality of Freeman et al. (1991) was less informative than a centrality metric based on minimum-cost-maximum-flow (Ahuja et al. 1993). Whereas least-cost path and max-flow algorithms only consider cost and capacity, respectively, min-cost-max-flow algorithms simultaneously consider cost and capacity to identify the set of edges or nodes of minimum cost that permit a set level of flow (Ahuja et al. 1993). The min-cost-max-flow betweenness centrality metric we developed first calculates the max-flow between a pair of nodes, and then maps the min-cost path for that flow. The importance of a node is based on the portion of the min-cost-max-flow which must pass through that node, summed over all node pairs. Simple cost value might assign private lands greater cost of public lands, reflecting common policies that seek to develop options for species conservation on public lands before regulating management of private lands. Or a uniform (positive non-zero) cost value may be used, which results in identification of the maximum flow path of minimum total area.

While pairwise linkage analysis using min-cost-max-flow is relatively rapid ([Exercise 1](#)), the computational time required for min-cost-max-flow betweenness centrality analysis of large graphs is currently on the order of thousands of hours. In some contexts, it may be preferable and more feasible to analyze a 'bounded-distance' min-cost-max-flow centrality metric that considers only pairwise relationships of nodes separated by less than a threshold distance. Borgatti and Everett (2006) suggested use of 'bounded-distance' betweenness in contexts where very long connections may be expected to be less relevant to system processes.

Given that it is computationally challenging to derive minimum-cost-maximum-flow BC over regional extents, we suggest regional planning efforts compare results from shortest-path and current-flow BC analyses. Shortest-path BC results identify the minimal set of linkages whose loss would greatly reduce regional connectivity. In contrast, the zones identified by current-flow BC assist in incorporating redundancy within a linkage network, which may be important for designing networks that are resilient to changing climate and land-use patterns or environmental catastrophes. Areas of high current flow BC may indicate pinchpoints where much potential dispersal flow encounters relatively constrained linkages. Higher-resolution analysis of individual linkages ([Exercise 2](#)) can be placed in context using the priority assigned to the linkage area in regional analyses.



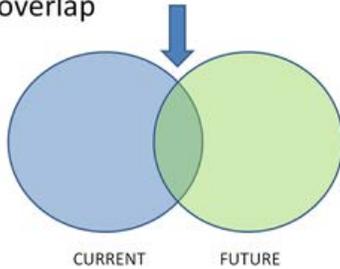
Shortest-path BC results overlaid on current flow BC results for wolf habitat in the northwestern US and southwestern Canada.

Network Flow Analysis of Time Series

In addition to allowing generation of graphs from single data layers, the Connectivity Analysis Toolkit also allows creation of graphs from a series of related layers of identical dimensions. These are typically 'time series' depicting e.g., how a landscape's suitability for a species is projected to change over time (with climate change, forest succession, or restoration). Phillips et al. (2008) ([pdf](#)) applied this method to identify 'climate corridors' for endemic plants in South Africa's Cape Province. Under their model, a species could persist if a currently suitable site either remained suitable under future climates or was adjacent to future suitable sites, allowing dispersal to new areas even if the current site was no longer suitable.

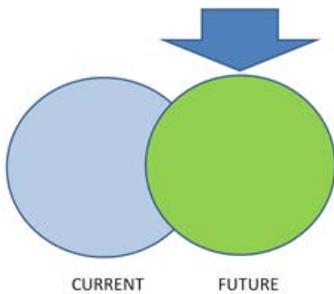
Assumption of no dispersal:

Species persists only where current and future habitat overlap



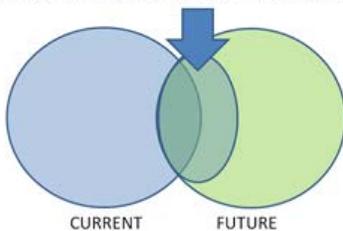
Assumption of unconstrained dispersal:

Species persists wherever future habitat is present

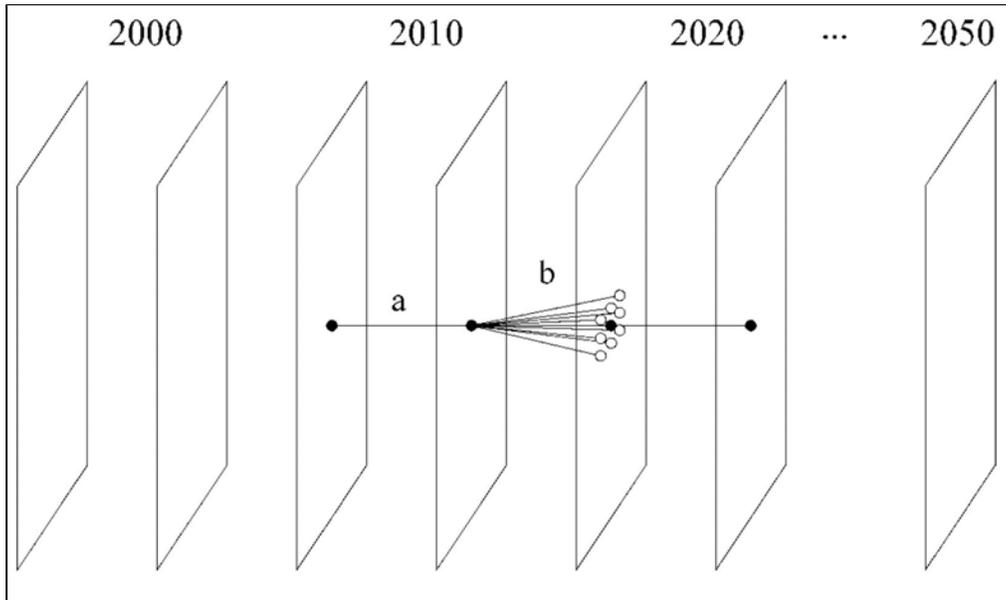


Network flow analysis allows assumption of constrained dispersal:

Species persists wherever future habitat is within dispersal distance of current habitat



This assumption can be represented as a graph with edges between sites at different steps within the time series (Phillips et al. 2008). Each timestep is used to create two timestep layers in the graph. The 'within-year' arcs (**a** in figure below) that connect the two layers in each timestep represent the probability of survival at that site during that timestep. 'Between-year' arcs connect the second layer of a timestep with the first layer of the next timestep. These represent dispersal between sites over time. Between-year arcs connect a site with the same site and adjacent sites at the next time period (**b** below). The radius within which a site can be considered adjacent is based on the species dispersal ability (Phillips et al. 2008).



From Phillips et al. (2008).

In this type of model structure, the second layer in the final time step will have no outward flow, so all flow values for that layer are zero. The values of interest are those for nodes in the FIRST layer of the first through final time steps.

See Phillips et al. 2008 ([link to pdf](#)) for more discussion on interpretation of output of time-series min-cost-max-flow.

The CAT implementation of this function does not include a second aspect of the approach used in Phillips et al. (2008), which simultaneously identifies linkages optimal for conserving multiple species. This method may be incorporated in a future version of the CAT.

See Tutorial Exercise 3 for more information on time-series network flow analysis. The interpretation of output from this function will depend on the biological context, e.g., dispersal ability of species of concern, etc. For species for which the assumption of constrained dispersal is relevant, inclusion of areas of high min-cost-max-flow in a reserve network might be expected to ensure persistence more efficiently than would a focus on the overlap between current and future habitat.

Input files

ASCII file

Habitat data is input to the Toolkit in the form of an .asc or ASCII file produced in a GIS software such as ArcGIS.

(Note: Input data cannot be in a 'geographic' projection, because this system uses units such as decimal degrees rather than meters).

This format consists of an array of data values preceded by a header containing a set of keywords:

```
<NCOLS xxx>
<NROWS xxx>
<XLLCENTER xxx | XLLCORNER xxx>
<YLLCENTER xxx | YLLCORNER xxx>
<CELLSIZE xxx>
{NODATA_VALUE xxx}
row 1
row 2
.
.
.
row n
```

For example,

```
ncols      725
nrows      1229
xllcorner  -2364022.7551035
yllcorner  1232329.6699838
cellsize   1000
NODATA_value -9999
-9999 -9999 -9999 -9999 -9999
.....
```

This format can be produced by a variety of software. Here we briefly describe methods used within ESRI's ArcGIS. From the ArcGIS Desktop interface (ArcMap or Arc Toolbox), you can use the From Raster toolset's 'Raster to ASCII' function to convert the information in a raster dataset such as a grid to an .asc file. Alternately, from the arc command prompt of ArcGIS Workstation, you can use the gridascii command:

```
gridascii <input grid> <output .asc>
```

The resolution of the grid and thus of the .asc file should be appropriate to the resolution at which you plan to run the graph analysis (see below). You should typically retain enough resolution (if present in the original data) to have many (e.g., 100+) pixels within each hexagon, but excessively high resolution may slow hexmap generation somewhat.

Treatment of background or NODATA values

ASCII files such as those produced by ArcGIS typically represent areas without data (background, ocean, etc.) with a value such as -9999. These areas can be ignored (excluded) during creation of the graph file as described below.

Treatment of zero values

A more difficult question occurs when some areas for which data is available are considered 'non-habitat' and assigned a value of zero in the ASCII file. In the context of connectivity analysis, an area with a conductance or capacity value of zero represents a complete barrier to dispersal. Thus, in the context of the methods described here, no analysis of connectivity across such barriers can occur. If it is known that such habitat really is a complete barrier to dispersal, and other options (potential linkage zones) for linking the areas of interest are available, then assigning non-habitat a zero value may be appropriate. However, often it is more informative to assign such areas a low but non-zero value so that it does not represent a complete barrier to dispersal. If habitat value are derived from a probabilistic model such as Maxent, then it may be useful to round up values so that the lowest probability value is distinguishable from zero in the input ASCII file. Unlike areas with value -9999, areas with zero value are NOT excluded from graph creation by default.

If too much of the landscape is composed of non-habitat which has been assigned a value if zero, the centrality analysis may fail due to the fact that no feasible connections exist between the areas of interest.

In min-cost-max-flow analyses, cost values of zero may also present problems. Therefore, even when using a uniform cost layer, cost should be set to a positive non-zero value.

Source and target input files for subset centrality

As described below, subset centrality requires input of text files identifying the subset of nodes (hexes) forming the source and target areas. These files have the following format:

For python-NetworkX based functions (shortest-path and current flow betweenness centrality subset), the required format is a single column of node ids per file:

```
1
2
3
```

...

Network flow methods (e.g., min-cost-max-flow centrality) can also accept such input as a two line file with a space-separated row of source ids in the first line and target ids in the second line,

```
1 2 3 4
5 6 7 8
```

While python-based functions can analyze connections between a group of source nodes (A) and target nodes (B), they cannot limit this analysis to specific node pairs, but will instead look at all possible pairs that combine a node in A and a node in B.

Network flow functions, however, can analyze a list of specific node pairs, entered as a file of more than two lines, in which each line specifies the space-separated ids of a source/target pair.

```
1 5
2 6
3 7
4 8
```

...

Output files

Output textfiles

Results of the centrality analyses are output as tab-delimited text files (.tab) of the format:

hex_id value

Output shapefiles

Shapefiles are produced by the Toolkit when the .asc file is initially converted to the hexmap format used by the Toolkit. These shapefiles allow the text file of results to be represented spatially.

In order to display the results spatially, the text file (or files if several different analyses were run) will need to be joined to the 'hex_id' field of the shapefile created during the analysis process (see below). In ArcGIS, you can access data in delimited text files and work with them as tables using either ArcCatalog or the Add Data dialog box in ArcMap. Files with a .tab extension should be automatically interpreted as tab delimited by ArcGIS.

You can join the table to a shapefile using either the Add Join function (under Data Management) or the Join Data dialog box, accessed by right-clicking a layer in ArcMap.

Alternately, using ArcGIS Workstation, you can create an INFO file from the textfile and use JOINITEM to join it to a coverage created from the shapefile.

We are considering development of a ArcGIS Toolbox to facilitate postprocessing of Toolkit output. If you have suggestions for this, please post comments to the Google Groups forum.

Hexmaps files represent a format used internally by the Toolkit (and Hexsim, a program for spatial population viability analysis [see Related Resources])

Graph format files

Various types of graph format files are produced in the course of analysis. Each one is appropriate to a different set of connectivity analysis methods. Thus it is possible to get meaningless results by producing the wrong input graph format for the analysis desired.

The types of graph format files available are:

LEMON graph format files (LGF)

Nodes are listed first. Then edges (arcs) are listed.

There are five fields per line in the edge section of the lgf: source_node, target_node, edge_label, edge_capacity, edge_cost.

- 1) LGF Directed - Used for time-series analysis using min-cost-max-flow
- 2) LGF Undirected - Used for min-cost-max-flow analyses (non-time-series, i.e., of a single map)

Example of lgf file

@nodes

1

2

...

@arcs

```
label capacity cost
1 2 0 1000 1000
1 63 1 1000 1000
2 3 2 1000 1000
2 64 3 1065 1000
2 63 4 1000 1000
2 1 5 1000 1000
3 4 6 1000 1000
```

Edgelist textfiles

Each line represents an edge.

There are three fields per line: from_node, to_node, attribute

- 3) Edge list with Distance attribute - Used for shortest-path betweenness analyses
- 4) Edge list with Conductance attribute - Used for current flow betweenness analyses
- 5) Edge list - Directed - Used for PageRank analyses (experimental)

Example of edgelist file

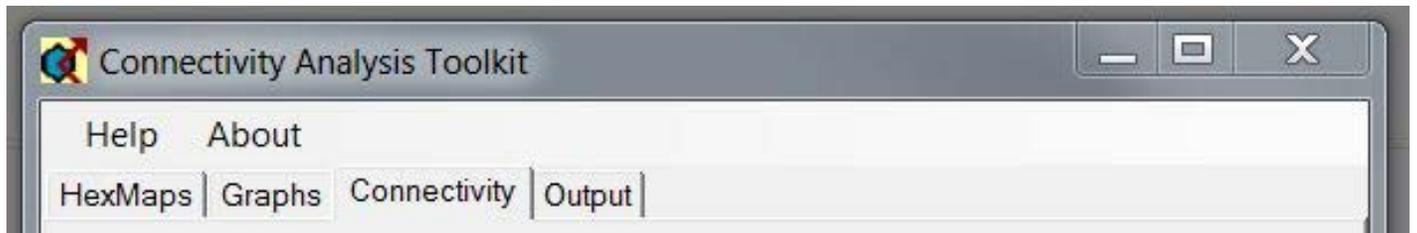
```
1 2 642.6068
1 157 644.6428
2 3 726.2068
2 158 704.6395
2 157 695.9301
2 1 642.6068
3 4 669.7904
3 159 679.8489
3 158 736.9523
3 2 726.2068
4 5 615.8793
4 160 612.6219
...
```

Running the Toolkit from the GUI

The Toolkit will typically be run from the GUI (graphical-user interface) by launching the program from a Desktop shortcut or the Start Menu.

The following sections describe how the Toolkit GUI can be used to process data from the input .asc file to the output textfile containing the centrality results.

Note: Some processes can take many hours, and users on multi-core systems may want to initiate a second analysis while the first analysis is completing. Currently this cannot be done by simply switching to another tab of the GUI. In order to start another analysis, the user must launch a second 'instance' of the program by relaunching the program from the Start Menu or Desktop shortcut.



The Toolkit GUI is composed of four primary tabs, as well as pulldown menus to access help information. The standard workflow will progress from the left to right tab, from hexmap and shapefile generation to graph generation, connectivity analysis, and output to a csv format textfile.

Hexmaps

This tab allows generation of hexmaps and shapefiles from input raster data (.asc files).

Graphs

This tab allows generation of graph format files from hexmaps.

Connectivity

This tab allows application of the connectivity analysis methods to the graph format files.

Output

This tab displays information on the progress (completion, error messages, etc.) of the functions occurring in the other tabs.

Help and About Tabs

Access the Help menu and About tab here. The help information accessed here is identical to that present in the pdf manual.

Hexmap Tab

The screenshot shows the Hexmap Tab interface with the following fields and buttons:

- Graph Source:** C:\WorkSpace\habitat.asc (Callout 2: Input filename)
- HexMap Output File Location:** C:\WorkSpace (Callout 4: Output folder)
- Name:** habitat
- Hexagon Area:** 500.0000
- Run** button (Callout 7: Run to generate hexmap)
- Browse** buttons for Graph Source and HexMap Output File Location (Callout 1: Browse for .asc file, Callout 3: Browse for hexmap)

Additional callouts on the left side:

- 5:** Name of hexmap (points to the Name field)
- 6:** Hexagon area (points to the Hexagon Area field)

This tab allows generation of hexmaps and associated shapefiles and node coordinate files from input raster data (.asc files).

1 Browse for .asc file

Locate the .asc file you have produced in a GIS program to represent habitat data.

2 Input filename

Name of the .asc file ...

3 Browse for hexmap

Specify the location where the hexmap file will be placed

4

Output folder

The .hxn suffix will be added to the filename you specify.

5

Name of hexmap

Specify the name of the output hexmap. This prefix will also be used to name the output shapefile (which consists of associated .shp, .shx, .dbf files) and a textfile with node coordinates (.crd extension).

6

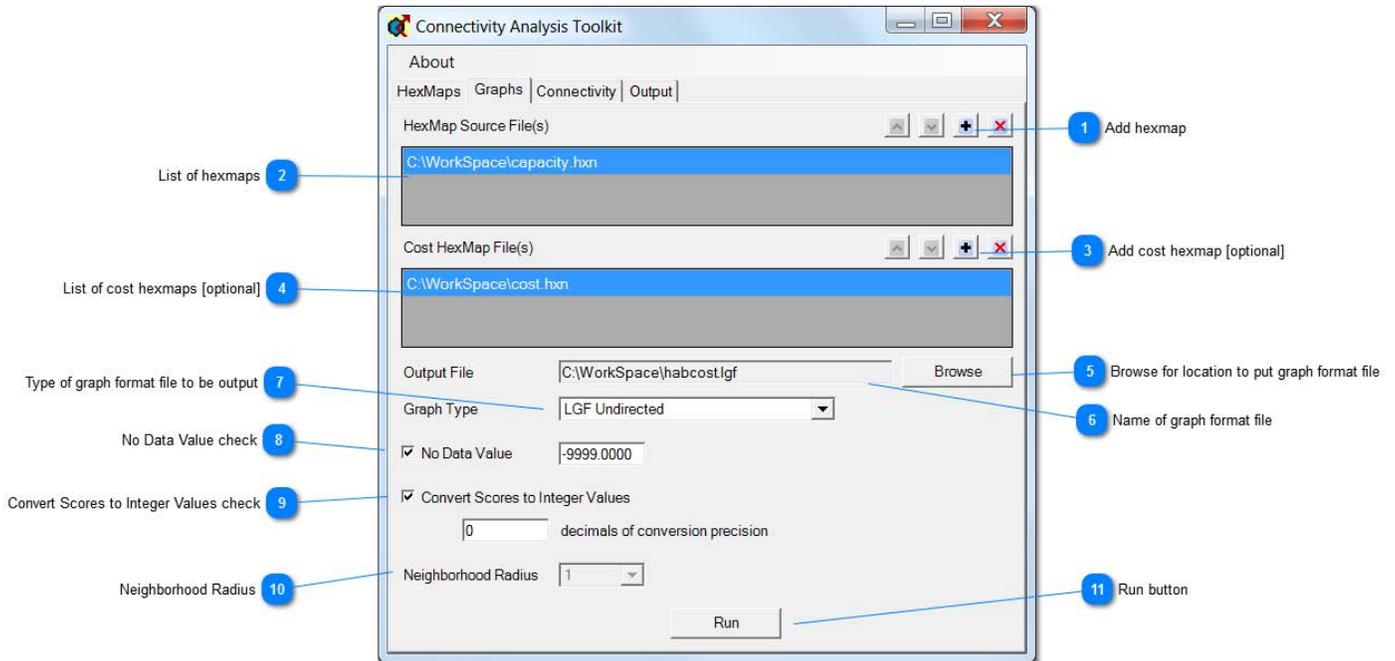
Hexagon area

Specify the hexagon size in hectares. You can use the extent of your .asc and the maximum size of the graph you can process to derive an approximate desired hexagon size.

7

Run to generate hexmap

Graphs Tab



This tab allows generation of graph format files from hexmaps.

1 Add hexmap

These buttons allow you to browse and add (and remove) hexmaps representing habitat values that will be used to derive distance, conductance, or capacity values in the output graph. In the case of time series analysis, several hexmaps may be listed in order of their time series step.

2 List of hexmaps

Except in time series analysis, only one hexmap will be listed.

3 Add cost hexmap [optional]

A hexmap representing cost, which is only supported when using the min-cost-max-flow method. Note that cost here is used in a different context than the attribute considered in 'least-cost-path' analysis (which is here termed 'distance').

4 List of cost hexmaps [optional]

Cost hexmap(s) are listed here if the min-cost-max-flow method is being used. In time series analysis, each step for which a habitat hexmap is specified (above) must also have a cost hexmap listed.

5 Browse for location to put graph format file

6 Name of graph format file

7 Type of graph format file to be output

The types of graph format files available are:

- 1) LGF Directed
- 2) LGF Undirected
- 3) Edge list - Distance
- 4) Edge list - Conductance
- 5) Edge list - Directed

See 'Output files' documentation page above for description.

8 No Data Value check

Typically, if the input .asc file was generated in ArcGIS, it will have represented areas without data as the value -9999. However, if a different NODATA value was used, it can be specified here. In almost all cases, users will want to check this option so as to exclude areas without data from being considered as nodes in graph file generation.

9 Convert Scores to Integer Values check

The network flow methods (e.g., min-cost-max-flow) use algorithms that perform better with integer than with floating-point values. Therefore if a network flow method is selected, this box will be checked by default. Conversely, this step is not allowed for the betweenness and current flow methods. Floating-point values may be converted to integers and yet retain adequate precision by multiplying by a power of 10 to retain digits that occur after the decimal point in the floating-point input. For example, if 2 digits of precision are specified, then the value 12.546 would be converted to 1255 (values are rounded to the next highest integer). Do not specify more digits than necessary as this will slow subsequent calculation of connectivity metrics, and may in extreme cases prevent the function from completing.

10 Neighborhood Radius

Used in time-series LGF creation only. Determines radius (in hexagons) up to which hexagons will be considered neighbors connected by between-year arcs.

11 Run button

Create the graph format file.

Connectivity Tab

The screenshot shows the 'Connectivity Tab' interface. It features a central panel with various input fields and buttons. On the left, there are 12 numbered callouts (1-12) pointing to specific elements. On the right, there are 3 numbered callouts (2, 4, 7) pointing to 'Browse' buttons. The interface includes a dropdown for 'Function' (set to 'Betweenness Centrality'), a 'Graph File' field with a 'Browse' button, an 'Input File(s) Type' dropdown (set to 'Generic Data'), and three 'Input File' fields with 'Browse' buttons. Below these are 'Maximum Distance' (0.0000), 'Output File' (C:\Workspace\betweenness.txt) with a 'Browse' button, and several checkboxes: 'Use Scaling', 'No Data Value' (with a value of -9999), and 'Number of Threads' (8). There are also input fields for 'Supply Fraction' (1.0000) and 'Probability' (0.0000). A 'Run' button is located at the bottom center.

This tab allows application of the connectivity analysis methods to the graph format files.

1

Connectivity Analysis Method

Options are:

- 1) Shortest-path betweenness centrality
- 2) Shortest-path betweenness centrality subset
- 3) Current flow betweenness centrality
- 4) Current flow betweenness centrality subset
- 5) Minimum Cost Flow [min-cost-max-flow, both all-pairs and subset]
- 6) Time series minimum cost flow

In addition, the following network flow methods are available but not currently documented:

PageRank

Minimum cut

Maximum flow

Maximum flow - flow value

Maximum flow - normalized

2

Browse for graph format file

Browse to locate the input graph format file produced in the previous step.

3

Optional files

Several files are optionally used as input during connectivity analysis.

Options in this menu are:

- 1) None specified - All pairs of nodes will be analyzed (typical option for centrality analysis)
- 2) Generic data - Subset centrality using network flow methods can accept data on the node (hex) ids of the specific source/target node pairs to be analyzed. The input file can either be a two line file with a space-separated row of source ids in the first line and target ids in the second line (see file provided for Tutorial exercise 1 (sourcetarget.txt) for an example), or a file of more than two lines, in which each line specifies the space-separated ids of a source/target pair. Non-network flow methods (shortest-path, current flow, and PageRank) cannot accept this input format.
- 3) Coordinate data - This is a textfile with .crd extension produced during hexmap generation (see 'Hexmap Tab'), with three space-separated fields: node (hex) id, x-coordinate, y-coordinate. This coordinate file is used in analyses where only pairs of nodes less than a threshold distance apart are analyzed (termed 'bounded-distance' analysis).
- 4) Source and target data - Input for subset centrality using either network flow methods or python-NetworkX based methods (shortest-path and current flow). This input consists of two files that respectively identify the node (hex) ids of those nodes which are the source and target nodes in subset centrality analyses. Format is a single column of node ids per file (see file provided for Tutorial exercise 2 (sourcetarget2.txt) for an example).

Node id files are typically produced in a GIS by overlaying points or patches of interest on the shapefile produced by the Toolkit.

4

Browse for optional input files

Browse to locate the optional input file(s) if needed. These options will be disabled if not relevant to the method chosen.

5

Distance threshold

Distance threshold used for bounded-distance analyses can be entered here (min cost flow only). Distances should be expressed in the units of the original .asc file (typically meters).

6

Use Scaling

If checked, min-cost-max-flow output will be scaled by the $(n - 1)(n - 2)$, where n is the number of nodes. (Shortest-path and current flow BC results are scaled by default). Do not use this option for subset analyses.

7

Browse for output

Browse for the location of the output file.

8

No Data Value

Typically, nodes with NODATA (usually -9999) values will not be included in the analysis. However, as they were typically filtered out in the generation of the graph format file ('Graphs' tab), this option is redundant except in cases where files that were created outside the application are being analyzed.

9

Number of threads

Users of systems with multi-core CPUs can improve performance by using multiple threads. This value will be set by default to the maximum number of virtual CPUs available. The value would be modified by the user who wanted the analysis to leave some CPU capacity available for other programs. This parameter is only relevant to when the min cost flow method is used.

10

Supply Fraction

Min-cost-max-flow analyses first identify the maximum flow between a node pair, and then the min-cost flow path for that flow. If desired, the user can specify that a min-cost flow path will be mapped that accommodates a set fraction of the max-flow. This may be useful in cases where planners want to maintain e.g., 80% of existing connectivity at minimum cost.

11

Probability

This value specifies that alpha parameter used in shortest-path BC, current-flow BC, and PageRank analyses. This value represents different parameters in the 3 functions:

- 1) Shortest-path BC: Alpha represents the proportion of nodes sampled (k/n) in the approximation process. The default value of 0.05 results in 5% of the nodes being subsampled, which typically results in an approximation that is >99% correlated with the exact value. Entering a value of 1 in this field will cause the exact algorithm to be used (warning: this may cause the function to take a long time to complete).
- 2) Current-flow BC: Alpha represents the absolute error tolerance (epsilon) in the approximation function. The default value of 0.1 typically results in an approximation that is >99% correlated with the exact value. Entering a value of 0 in this field will cause the exact algorithm to be used (warning: this may cause the function to use large amounts of RAM or fail with a memory error if sufficient RAM is not available).
- 3) Pagerank: Alpha represents $(1 - \text{the probability of a 'jump' to a random node in the graph})$. A default alpha value of 0.85 is used in typical webpage ranking applications, but a default alpha value of 0.95 is more appropriate in this context.

12

Run button

Output Tab

```
Rows: 590 Columns: 301 Hexagons: 177590  
Generating from source: C:\WorkSpace\habitat.asc  
Output = 0  
Resolution: 100  
Writing hexagon file: C:\WorkSpace\capacity.hxn
```

The output tab displays information on the progress of the analysis (initial command issued by the program, completion of process, error messages) of the functions occurring in the other tabs. The command string visible in the output tab when a function is started in the CAT GUI can help users check the proper syntax for calling the CAT functions from the command line (next section),

Running the Toolkit from the Command Prompt

It is possible to run the Toolkit's connectivity analysis functions from the command line. This might be useful in order to run many analyses in succession using a batch file (.bat).

It also may be used to facilitate integration of the Toolkit's graph analysis functions into other software programs.

The executable file 'Network Flow.exe' is found in the top-level directory where the toolkit is installed (e.g., C:\Program Files\ConnectivityToolkit\).

The executable file 'pythonflow.exe' is found in the second-level directory 'pythonflow' where the toolkit is installed (e.g., C:\Program Files\ConnectivityToolkit\pythonflow).

These files depend on other files within those directories, so must be called using their existing pathnames rather than copied to another directory.

The command-line syntax for pythonflow.exe, the Toolkit's NetworkX based function is:

```
pythonflow.exe <option> <input.txt> <output.tab>
```

The input file format (typically produced via the GUI 'Graphs' tab) is a space delimited textfile with three columns: from_node to_node edge_weight

The output file format is a tab-delimited textfile with two columns: node<tab>value

The following flags determine the method used:

- b // Betweenness centrality
- d // Subset betweenness centrality
- c // Current flow betweenness centrality
- e // Subset current flow betweenness centrality
- f // PageRank

The following flags identify input files:

- x // [required for options -d and -e] input file for sources, format is textfile with single column of node ids
- y // [required for options -d and -e] input file for targets, format is textfile with single column of node ids
- a // [required for options -b, -c, and -f] alpha value for shortest-path BC, current-flow BC, and PageRank (see description in previous sections)

The command-line syntax for 'Network Flow.exe', the Toolkit's LEMON based function is:

```
"Network Flow.exe" <option> <input.txt> <output.tab>
```

The input file format (typically produced via the GUI 'Graphs' tab) is a lgf (LEMON graph format) file

The output file format is a tab-delimited textfile with two columns: node<tab>value

The following flags determine the method used:

- mcf // min-cost-max-flow non-threaded
- mcfT // min-cost-max-flow threaded using Cost Scaling
- mcf2 // min-cost-max-flow threaded using Network Simplex (the default, as it is typically faster than cost scaling)
- mcf2 // time series min-cost-max-flow (threaded)

The following additional methods are available, but are experimental and not documented here:

- mc // min-cut non-threaded

-mf // max-flow non-threaded
-mft // max-flow threaded
-mfvt // max-flow value threaded
-mfnt // max-flow normalized threaded

The following flags determine options for the above methods:

-t // number of threads
-d // maximum distance for 'bounded-distance' metrics
-f // supply fraction for min-cost-max-flow method
-s // scale (divide) the output by $(n-1)(n-2)$ where n = number of nodes

The following flags identify input files:

-i // input file
-c // node coordinate file

Example syntax:

Linkage analysis:

"Network Flow.exe" -mcf2 -t 8 -i sourcetarget.txt input.lgf output.tab

All-pairs centrality with bounded distance:

"Network Flow.exe" -mcf2 -t 8 -c xy.txt -d 300000 input.lgf output.tab

Flowchart of analysis process

The following table shows the different types of input and output files associated with the different methods.

METHOD	MAIN INPUT	AUXILIARY INPUT	GRAPH FORMAT FILE
Shortest-path betweenness centrality (BC)	Habitat (1 hxn file)	None	Edgelist textfile (distance)
Shortest-path BC subset	Habitat (1 hxn file)	Source/target file(s)	Edgelist textfile (distance)
Current-flow BC	Habitat (1 hxn file)	None	Edgelist textfile (conductance)
Current-flow BC subset	Habitat (1 hxn file)	Source/target file(s)	Edgelist textfile (conductance)
Minimum cost flow	Capacity and cost (2 hxn files)	Optional: Source/target file(s)	LGF file
Time series minimum cost flow	Capacity and cost for n time steps (n * 2 hxn files)	None	LGF file

Examples

The following section gives four examples of how the CAT functions can be applied in conservation planning. Contact us if you have examples from your application of the CAT that you would like to be included in future versions of the manual.

Example 1: Shortest-path betweenness centrality (BC)

Case study: Analysis of landscape permeability in California with all-pairs shortest-path betweenness centrality

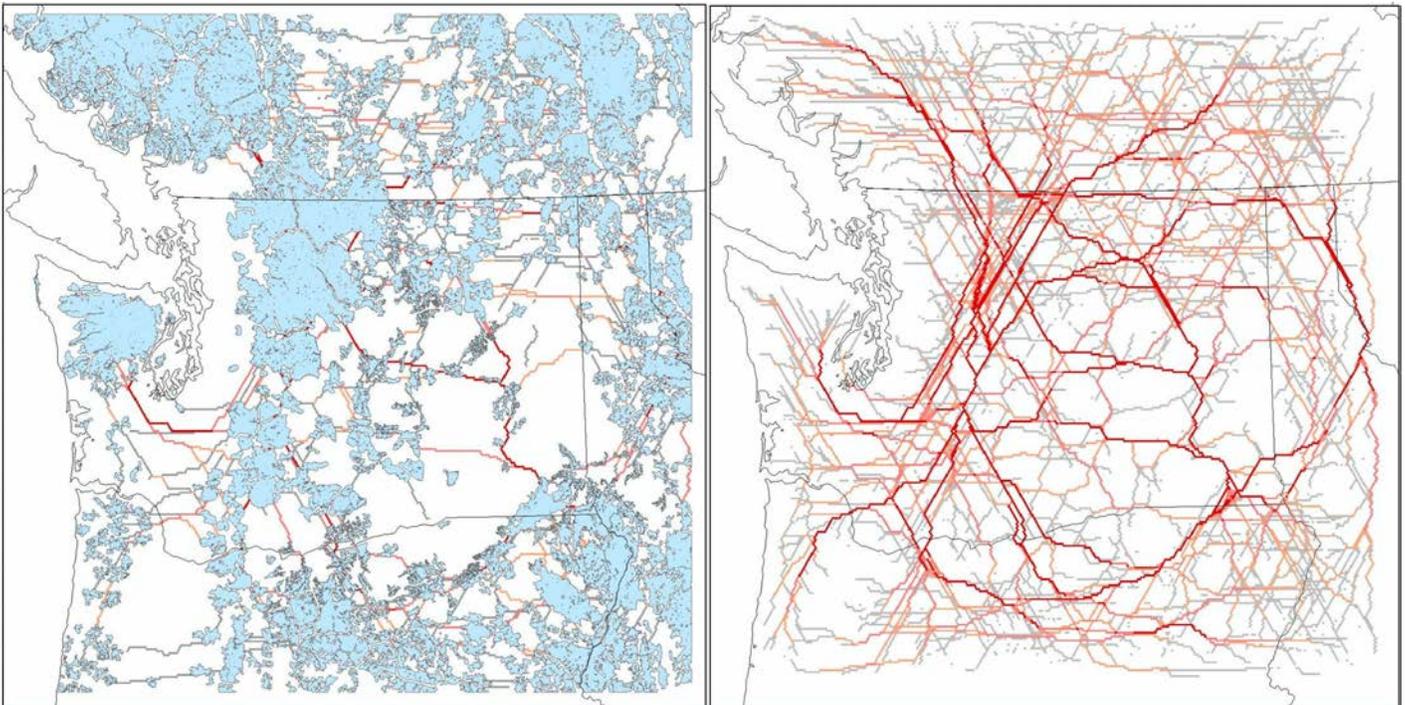
As part of the development of the report "California Essential Habitat Connectivity Project: A Strategy for Conserving a Connected California", Spencer et al. (2010) developed a data layer representing landscape resistance for the state of California. This data is used here to develop an example of how centrality analysis may be applied to such generalized representations of landscape pattern and composition to produce connectivity results that are not species-specific. The resistance data, based on landscape features such as vegetation, roads, and human land use, was on an ordinal scale of 1 to 25. Because the Toolkit expects habitat data to be scaled as conductance (i.e., here worst=1, best=25) rather than resistance, we first inverted the scaling. (In retrospect, it would be technically more correct to have taken the reciprocal of conductance to derive resistance. However, an ordinal scale cannot be assumed to directly represent conductance, so the best approach may be to try several different transformations.) We then squared the values to accentuate the contrast between poor and good values, a step that is often necessary when input data are on an ordinal scale rather than e.g., representing probabilities of occurrence. 82,000 hexagons of 5 square kilometers each were used to represent the region in a graph format. The exact algorithm for shortest-path betweenness centrality analysis required 0.2 GB and 20 hours to complete. Approximate solutions required less than 1 hour. The results are shown here overlaid on a background showing public lands within California. This is an exploratory analysis, so may be most appropriately used as a tool for comparison with results from other methods described in Spencer et al. (2010).



Example 2: Shortest-path betweenness subset centrality

Case study: Analysis of landscape permeability in Washington with subset shortest-path betweenness centrality

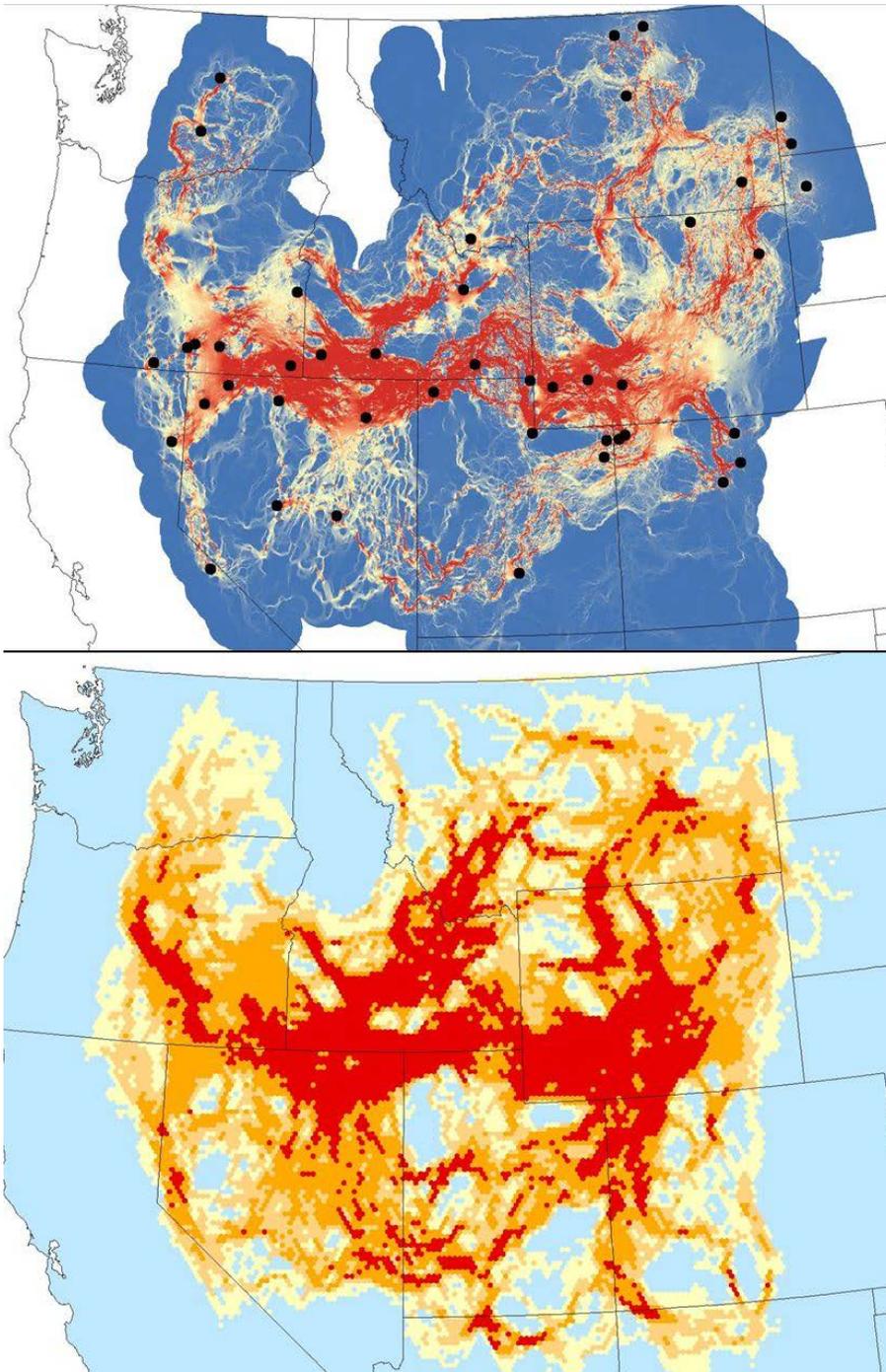
A similar landscape permeability layer was developed by the Washington State Wildlife Habitat Connectivity Working Group ([website](#)). The project also developed a map of core areas in the state with high levels of 'landscape intactness'. To provide an example of the contrast between subset and all-pairs methods, these core areas (blue areas in left panel below) were used as source and target patches in an analysis using shortest-path betweenness subset centrality. Subset centrality results (left panel below) show only linkages between pre-defined core areas rather than between all sites on the landscape. Linkages of greater importance are shown in darker orange shades. These results may be easier for planners to interpret than all-pairs centrality results (right panel below), especially if the input core areas were based on an ecologically-relevant classification.



Example 3: Comparison of subset and all-pairs current flow BC

Case study: Habitat connectivity for sage grouse

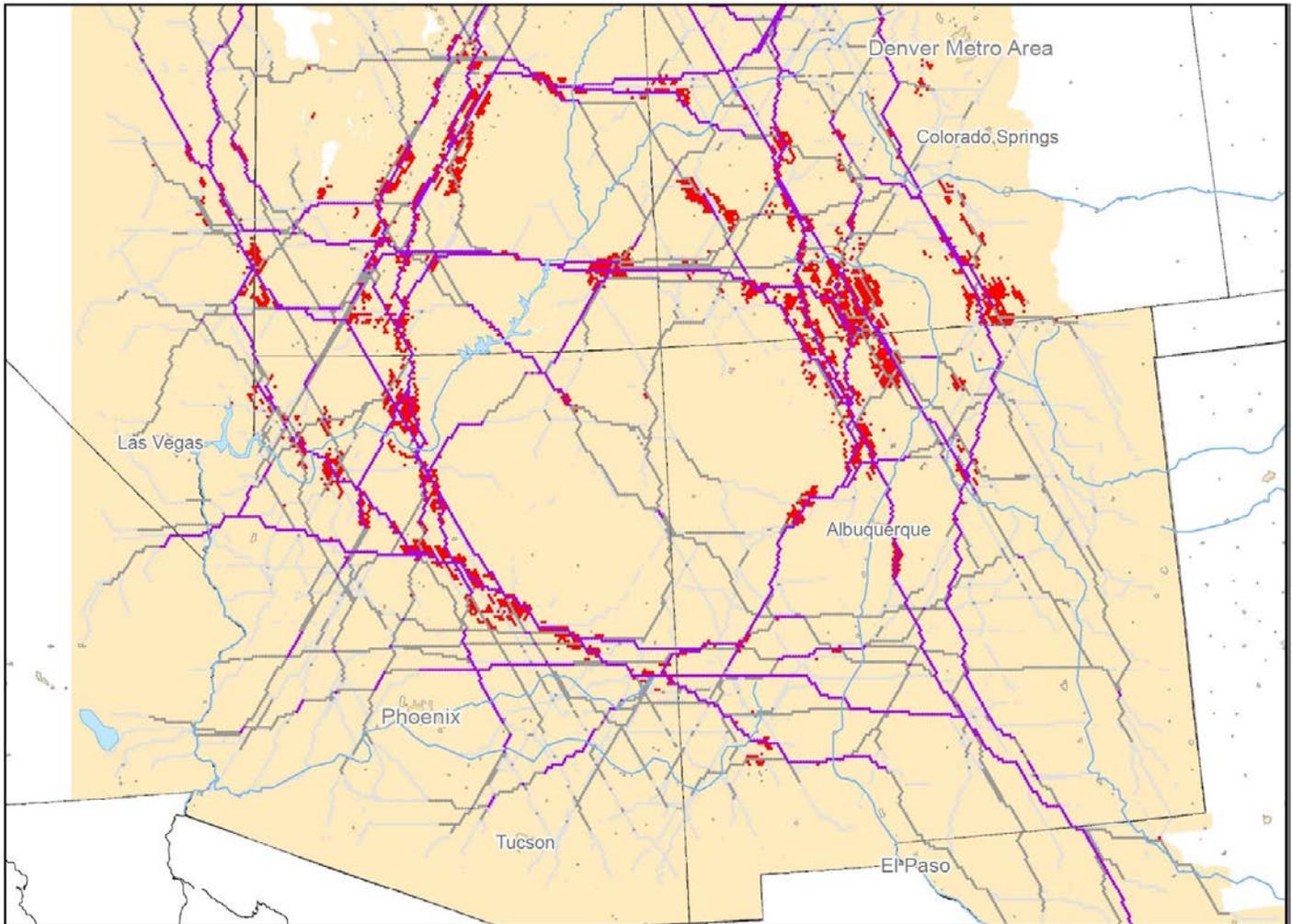
The Greater Sage Grouse (*Centrocercus urophasianus*) is a species of conservation concern in the sagebrush biome of the western United States. We used a habitat model for sage grouse as input to 2 related functions: subset current flow centrality, as calculated by the program [Circuitscape](#) (top panel), and all-pairs current flow centrality calculated by the CAT (bottom panel). Subset nodes in Circuitscape (shown in black) were based on the geographic centroids of sage grouse subpopulations. Linkages of greater importance (more current) are shown in darker orange shades. The results shown in the two panels are similar, and as the number of subset nodes is further increased, Circuitscape results will more closely approximate the results of all-pairs current flow centrality.



Example 4: Comparison of shortest-path and current flow BC

Case study: Habitat connectivity for the Mexican wolf

The endangered Mexican wolf (*Canis lupus baileyi*) is being restored to the southwestern United States. Conserving habitat linkages may be important for facilitating dispersal between populations. We used a habitat model for the species as input to an analysis of shortest-path betweenness centrality and current-flow betweenness centrality (BC). The figure below shows the most important linkages from shortest-path BC in purple, with the red areas in the background showing areas of high current flow, which are connectivity 'pinchpoints' where much potential dispersal flow is being routed through relatively limited habitat. Comparing results from the 2 methods provides more information for linkage conservation planning than does either method alone. Conservation guidelines might focus on the most important shortest-path linkages but give additional attention to areas with high current flow as well.



Tutorial

This tutorial provides a brief introduction to several functions of the CAT. Please refer to the sample files included in the Tutorial directory which is found in the directory where the program was installed on your system. You will want to copy the 'Tutorial' directory to another location on your system in order to work through the tutorial analyses.

Exercise 1 - Mapping linkages with min-cost-max-flow subset centrality

In this exercise, we will be using network flow analysis to analyze habitat connectivity for wolves between central Idaho and Yellowstone National Park. The example is taken from Carroll et al. (2011).

Please refer to the manual for a description of the analysis method, which is an algorithm termed min-cost-max-flow. This algorithm requires two types of data, representing respectively habitat suitability and land cost. The method identifies the set of paths that allows maximum flow between two areas at minimum total cost. In this example, wolf habitat suitability is derived from data on land cover, productivity, slope, roads, and human population, while cost is based on a simple model that assigns greater cost to private vs. public lands.

Habitat data is input to the Connectivity Analysis Toolkit (CAT) in the form of an .asc or ASCII file produced in a GIS software such as ArcGIS. From the ArcGIS Desktop interface (ArcMap or Arc Toolbox), you can use the From Raster toolset's 'Raster to ASCII' function to convert the information in a raster dataset such as a grid to an .asc file. Alternately, from the arc command prompt of ArcGIS Workstation, you can use the gridascii command:
gridascii <input grid> <output .asc>

1) Hexmaps Tab –

Refer to p. 16 of the Toolkit manual for a description of the Hexmaps Tab.

In the directory 'Exercise_1', there are 2 files named habitat.asc and cost.asc.

There is also a subdirectory named OtherFiles that contains copies of all the files we will generate during the analysis process. Please refer to these if your attempt to create these files is unsuccessful.

We will create two hexagon (.hxn) files from these .asc files.

In the 'Hexmaps' tab, select the directory Exercise_1 and the file habitat.asc

Set 'hexagon area' to 500 hectares. Set the name of the output to habitat5

Press Run to generate the hxn file.

Repeat the process to create cost5 from cost.asc

The hxn files are a binary format only usable within the CAT (and Hexsim).

However, the program created several associated files for each hxn file.

Shapefiles are also produced by the Toolkit when the .asc file is initially converted to the hxn format. These shapefiles allow the text file of results to be represented spatially.

Add the shapefile habitat5.shp to an ArcMap composition. We will use this composition later.

Open the file habitat5.crd in a text editor (e.g., Notepad). This file contains the x,y coordinates of the centroids of each hexagon. This data can be used to constrain analyses to not consider flow between those nodes that are greater than a maximum distance apart. For example, if 800 km was the maximum recorded dispersal distance for wolves, we might want to perform at least one set of analyses excluding node pairs greater than 800 km apart.

2) Graphs Tab –

Refer to p. 18 of the Toolkit manual for a description of the Graphs Tab.

We will create the LGF (LEMON graph format) input file we will use in the subsequent connectivity analysis.

Press the + tab above the upper box and add the file habitat5.hxn to the upper box (Hexmap Source File).

Press the + tab above the lower box and add the file cost5.hxn to the lower box (Cost Hexmap).

Set graph type as LGF Undirected.

Set the Output file name as habcost5.lgf

Leave checked 'No Data Value -9999'.

Set the decimals of 'convert scores to integer values' as 1 (this is appropriate based on the range of values [1-1000] in the input .asc files).

Press Run.

Examine the file habcost5.lgf in a text editor. Note how the node and arc sections of this file are created and attributed.

3) Connectivity Tab - Refer to p. 21 of the Toolkit manual for a description of the Connectivity Tab.

A subset file is needed that lists the hexagons (nodes) between which flow is mapped. This is typically created using an ArcMap composition which displays the shapefile, by identifying the hex_id values of hexagons at the endpoints of the linkage path. We have created a subset file beforehand, a textfile named sourcetarget.txt.

Open sourcetarget.txt in a text editor to see the format required for the subset file.

In the Connectivity Tab, set the Function as 'Minimum Cost Flow'.

Browse and locate the graph file, habcost5.lgf.

Leave input file type as 'generic data'.

Browse and select sourcetarget.txt as the input file for this field.

Browse and name the output file: mcfhabcost5.tab.

Press Run to run the analysis.

The progress of the analysis will be displayed in the 'Output' tab.

4) Working with the CAT output files

In order to display the results spatially, the text file (or files if several different analyses were run) will need to be joined to the 'hex_id' field of the shapefile created during the hxn file generation process.

Open the file mcfhabcost5.tab in a text editor. Notice the line at the beginning naming the fields:

"HEX_ID, VALUE". This information allows the output to be joined to the HEX_ID field in the shapefile.

Open your ArcMap composition.

Right-click on the shapefile habitat5 (in the list of datasets).

Select 'Joins and Relates' and then 'Join'.

In the menu box that appears,

In menu item #1 (join field in shapefile), choose the field HEX_ID.

In #2 (choose table), select the textfile mcfhabcost5.tab.

In #3 (join field in table), choose the field HEX_ID.

Choose the default option 'Keep All Records'.

Run the Join.

Now you will be able to go to the Symbology tab of the layer 'habcost5' and display the results of the CAT analysis using a graduated color scheme. You should choose to make hexagons with zero flow be transparent in the map composition.

Warning: the display of a joined field is slow in ArcMap. If you prefer, the data in the textfile can be permanently joined to the shapefile to make future display of the results more rapid. Alternately, you can produce a raster grid from the shapefile using ArcGIS's "polygon to raster" tool.

Exercise 2 - Mapping linkages with shortest-path subset centrality

This exercise is similar to Exercise 1, except that we will use shortest-path subset centrality and review the types of file formats necessary for that function.

Please refer to the manual for a description of the analysis method, which is an algorithm termed shortest- or least-cost path. (Note that 'cost' in this metric's name is energetic cost in terms of habitat permeability, not monetary cost as in the previous exercise). This algorithm requires a single input layer, representing habitat suitability. The method identifies the set of least-cost paths between two areas. In this example as in the previous one, wolf habitat suitability is derived from data on land cover, productivity, slope, roads, and human population.

1) Hexmaps Tab –

Instead of creating a new hxn file, we will use a copy (located in the directory 'Exercise_2') of the habitat5.hxn file created in Exercise 1.

There is also a subdirectory named OtherFiles that contains copies of all the files we will generate during the analysis process. Please refer to these if your attempt to create these files is unsuccessful.

2) Graphs Tab –

Refer to p. 18 of the Toolkit manual for a description of the Graphs Tab.

We will create the edgelist input file we will use in the subsequent connectivity analysis.

Press the + tab above the upper box and add the file habitat5.hxn to the upper box (Hexmap Source File).

Set graph type as Edgelist - Distance (Betweenness).

Set the Output file name as hab5dist.txt

Leave checked 'No Data Value -9999'.

Press Run.

Examine the file hab5dist.txt in a text editor. Note how the edges are created and attributed.

3) Connectivity Tab - Refer to p. 21 of the Toolkit manual for a description of the Connectivity Tab.

A subset file is needed that lists the hexagons (nodes) between which flow is mapped. This is typically created using an ArcMap composition which displays the shapefile, by identifying the hex_id values of hexagons at the endpoints of the linkage path. We have created a subset file beforehand, a textfile named sourcetarget.txt.

Open sourcetarget.txt in a text editor to see the format required for the subset file.

In the Connectivity Tab, set the Function as 'Shortest-Path Betweenness Centrality Subset'.

Browse and locate the graph file, hab5dist.txt.

Change input file type to 'Source and Target data'.

Browse and select sourcetarget2.txt as the input file for the first field.

Browse and select sourcetarget2.txt as the input file for the second field.

Open this file in a text editor and note the difference in format from the subset file 'sourcetarget.txt' used in the previous exercise.

Browse and name the output file: hab5betweensub.tab.

Press Run to run the analysis.

The progress of the analysis will be displayed in the 'Output' tab.

4) Working with the CAT output files

In order to display the results spatially, the text file 'hab5betweensub.tab' will need to be joined to the 'hex_id' field of the shapefile as described in Exercise 1.

Exercise 3 - Mapping landscape connectivity with all-pairs centrality

For this example, we will analyze 'all-pairs' rather than subset centrality.

Our example is based on the same data set of wolf habitat suitability in the Yellowstone region used in Exercises 1 and 2. This analysis may require more processing time to complete than previous exercises.

1) **Hexmaps Tab** – Once again, we can skip this step because we will use the same habitat5.hxn file created in Exercise 1.

2) **Graphs Tab** –

Refer to p. 18 of the Toolkit manual for a description of the Graphs Tab.

In Exercise 2, we created an edgelist input file named hab5dist.txt. In this exercise we will use this file again, as well as an additional file of conductance values.

Press the + tab above the upper box and add the file habitat5.hxn to the upper box (Hexmap Source File).

Set graph type as Edgelist - Conductance (Current Flow).

Set the Output file name as hab5cond.txt

Leave checked 'No Data Value -9999'.

Press Run.

Examine the file hab5cond.txt in a text editor. Note how the edge values are the reciprocal of those shown in the first edgelist file we created (hab5dist.txt).

3) **Connectivity Tab** - Refer to p. 21 of the Toolkit manual for a description of the Connectivity Tab.

First analysis:

In the Connectivity Tab, set the Function as 'Shortest-Path Betweenness Centrality'.

Browse and locate the graph file, hab5dist.txt.

Because we are not using a subset file, leave input file type as 'Generic data'.

Browse and name the output file: hab5betweenness.tab.

Set 'probability value' to 0.02 (=2%), which is lower than the default sampling proportion of 5% and will result in faster completion of the function.

Press Run to run the analysis.

The progress of the analysis will be displayed in the 'Output' tab.

Second analysis:

In the Connectivity Tab, set the Function as 'Current Flow Betweenness Centrality'.

Browse and locate the graph file, hab5cond.txt.

Because we are not using a subset file, leave input file type as 'Generic data'.

Browse and name the output file: hab5currentflow.tab.

Set 'probability value' to 0.2, which is higher than the default error tolerance of 0.1 and will result in faster completion of the function.

Press Run to run the analysis.

The progress of the analysis will be displayed in the 'Output' tab.

4) **Working with the CAT output files**

Join the 2 output files to the to the HEX_ID field in the shapefile as described in previous exercises.

Exercise 4 - Mapping connectivity under climate change: analysis of time series

In addition to allowing generation of graphs from single data layers, the Connectivity Analysis Toolkit also allows creation of graphs from a series of related layers of identical dimensions. These are typically 'time series' depicting e.g., how a landscape's suitability for a species is projected to change over time (with climate change, forest succession, or restoration).

1) Graphs Tab –

We will use 6 hxn files that have previously been created to create the LGF (LEMON graph format) input file we will use in the climate connectivity analysis. These files contain data depicting habitat suitability for Northern Spotted Owl in the years 1961-1990 ("current"), 2011-2040 ("2025"), and 2061-2090 ("2075").

As this capability is not present in your version of the CAT (version 1.0), please follow along as the example is demonstrated on the screen.

Press the + tab above the upper box and add the file owlcurrent.hxn to the upper box (Hexmap Source File).

Press the + tab above the upper box and add the file owl2025.hxn to the upper box. It will be listed below the spowcur.hxn file.

Press the + tab above the upper box and add the file owl2075.hxn to the upper box.

Three hxn files are now listed in the upper box.

Press the + tab above the lower box and add the file cost.hxn to the lower box (Cost Hexmap).

In this example, we have no data on how cost varies over time, so we use the same (area-based or uniform) cost hxn for all timesteps.

Do this a second and third time until 3 hxn files are listed in the lower box.

Set graph type as LGF Directed.

Set the Output file name as owl3n.lgf

Leave checked 'No Data Value -9999'.

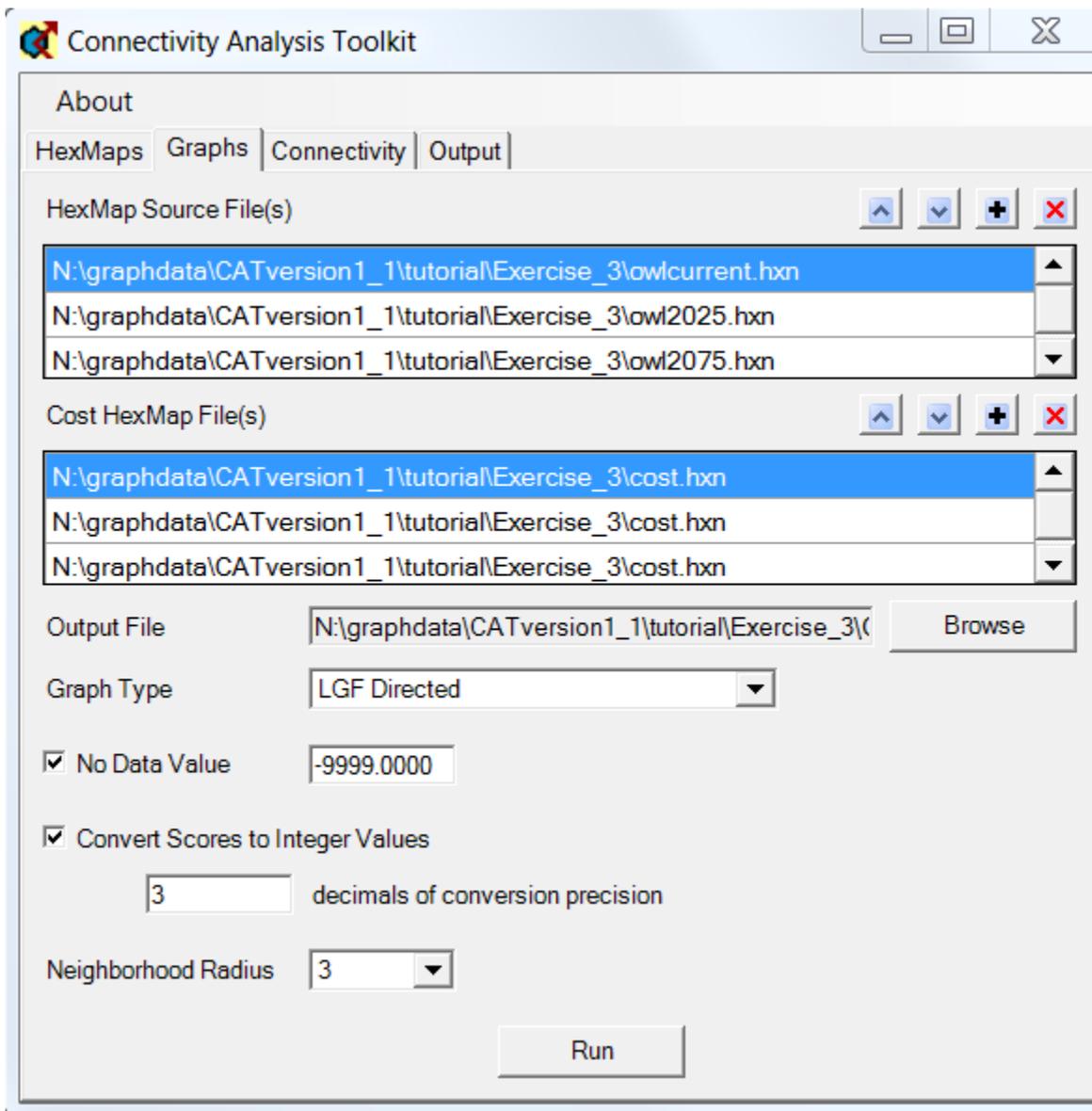
Set the decimals of 'convert scores to integer values' as 3.

Set Neighborhood Radius to 3.

Press Run.

Examine the file owl3n.lgf in a text editor. Note how the node and arc sections of this file are created and attributed.

Examine the file owl3n.tab in a text editor. This file specifies the correspondence between nodes in the time-series graph and hexagons in the input hxn files.



3) Connectivity Tab -

Browse and locate the graph file, owl3n.lgf.

In the Connectivity Tab, set the Function as 'Time-Series Minimum Cost Flow'.

Browse and name the output file: owl3nout. [NOTE: Do not add a file suffix (e.g., .tab), or networkflow.exe will crash).

Press Run to run the analysis.

The progress of the analysis will be displayed in the 'Output' tab.

4) Working with the CAT timeseries output files

The CAT will produce a separate output file for each 'layer' in the time-series graph. Thus a graph created from 3 hxn files will result in $3 * 2 = 6$ output files. Numbering of the output files is done starting with '0', so the output from the first step of the first time step will be named owl3nout_0_0.tab. These files contains the hex_id and outgoing flow values for each layer. These files can be joined to the shapefile output in ArcGIS as described in Exercise 1. The output files from the first layer in the second through final time steps will be most relevant to planning.

Troubleshooting

Slow processing times and memory errors

Many of the Toolkit's functions for graph analysis ('Connectivity' tab) are highly demanding of computational time and RAM (see System Requirements). If you are encountering memory errors or runtimes that exceed your patience level, try an initial analysis using a smaller graph (larger hexagon size) or a more approximate solution.

Use of incorrect input file format

See [Flowchart of analysis process](#) for correct graph format file for each function. Input of the wrong graph format file (e.g., a lgf file into shortest-path BC) will result in an error message.

Use of input layers with zero-valued habitat or cost

Although in some instances CAT functions can complete when input habitat layers contain small areas of zero habitat value (which will be treated as barriers), generally the minimum habitat value should be a low but non-zero value. No areas of zero value should be present in the cost data used in min-cost-max-flow analysis.

Use of inappropriate approximation parameters

If the probability value (described in the section [Connectivity Tab](#)) for shortest-path and current-flow BC approximations is changed from the default value to a value that requires too large a subsample of the graph, the program may use the exact algorithm or may in some cases fail with an error message.

"Network Flow.exe" crashes

This may be caused by the issue described above (input layers with zero-valued habitat or cost). Or the 'convert decimals to integers' value may be set too high. Or the output filename given by the user (e.g., owl3nout in Tutorial Exercise 4) may have been incorrectly given a file suffix (e.g., .txt). If changing these parameters does not resolve the problem, please contact the developer.

Use in regions/locales where the decimal separator is a comma rather than a period

Although version 1.3 revises most functions to work in these regions (e.g., most of the world outside of the US and UK), there remain a couple of exceptions where this may causes errors:

1) Use of networkflow functions (e.g., min-cost flow) with either of these 2 options:

-d // maximum distance for 'bounded-distance' metrics

-f // supply fraction for min-cost-max-flow method

To resolve this, use a distance value which is an integer, and use a supply fraction of 1 (the default).

Reporting bugs

There are likely to be issues that arise due to software bugs. The output tab may provide error messages that are helpful in identifying the problem. Please post a description of any problems to the CAT discussion forum or email the developer at klamathconservation@gmail.com.

<http://groups.google.com/group/connectivitytoolkit?hl=en>

Related Resources

Publications using the Connectivity Analysis Toolkit

Carroll, C., B. McRae, and A. Brookes. 2012. Use of Linkage Mapping and Centrality Analysis Across Habitat Gradients to Conserve Connectivity of Gray Wolf Populations in Western North America. *Conservation Biology* 26:78-87.

Related software resources

Circuitscape ([website](#)) is a software designed for analyzing current flow between two or more source locations on high-resolution raster surfaces (habitat maps). Both the CAT and Circuitscape calculate current flow as a measure of connectivity, but Circuitscape is designed to calculate current flow between a subset of nodes or core areas on raster datasets, whereas the CAT is better suited to current flow centrality calculations. While Circuitscape can be used to calculate current flow centrality by summing currents between all pairs, it is not optimized for this purpose. Because Circuitscape is designed to calculate current, voltage, and resistance, it does not calculate other connectivity measures employed by the CAT, such as least-cost paths. It does, however, accommodate very large raster datasets (with millions of nodes, depending on availability of RAM).

Hexsim ([website](#)) is a software designed to perform spatial population viability analysis. CAT functionality such as hexmap generation was derived from Hexsim. Users who are interested in exploring more complex and realistic simulations of functional connectivity (e.g., dispersal barriers) may be interested in using Hexsim.

UNICOR ([website](#)) is a software that allows calculation of a metric that is similar to approximate shortest-path betweenness centrality on high-resolution rasters.

Publications cited in this documentation

Adriaensen, F., J. P. Chardon, G. D. Blust, E. Swinnen, S. Villalba, H. Gulinck, and E. Matthysen. 2003. The application of “least-cost” modeling as a functional landscape model. *Landscape and Urban Planning* 64:233–247.

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin. 1993. *Network flows: theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs, New Jersey.

Beier, P., D. R. Majka, and W. D. Spencer. 2008. Forks in the road: choices in procedures for designing wildland linkages. *Conservation Biology* 22:836–851.

Bodin, O., M. Tengö, A. Norman, J. Lundberg, and T. Elmqvist. 2006. The value of small size: loss of forest patches and ecological thresholds in southern Madagascar. *Ecological Applications* 16:440–451.

Borgatti, S. P. 2005. Centrality and network flow. *Social Networks* 27:55-71.

Borgatti, S. P., and M. G. Everett. 2006. A graph-theoretic perspective on centrality. *Social Networks* 28:466–484.

Brandes, U. 2001. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology* 25:163-177.

Brandes, U. and D. Fleischer. 2005. Centrality Measures Based on Current Flow. *Proceedings of the 22nd Symposium Theoretical Aspects of Computer Science* 533-544.

Brandes, U., and C. Pich. 2007. Centrality estimation in large networks. *International Journal Of Bifurcation And Chaos (Special Issue On Complex Networks’ Structure And Dynamics)* 17:2303-2318.

Bunn, A. G., D. L. Urban, and T. H. Keitt. 2000. Landscape connectivity: a conservation application of graph theory. *Journal of Environmental Management* 59:265–278.

Chetkiewicz, C. L. B., C. C. St Clair, and M. S. Boyce. 2006. Corridors for conservation: integrating pattern and process. *Annual Review of Ecology, Evolution, and Systematics*, 37:317-342.

Cushman, S. A., K. Gutzweiler, J. S. Evans, and K. McGarigal. 2010. The gradient paradigm: a conceptual and analytical framework for landscape ecology. Pages 83-108 in S.A. Cushman and F. Huettmann, editors, *Spatial Complexity, Informatics, and Wildlife Conservation*. Springer, New York, New York.

Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269-271.

- EGRES. 2010. LEMON, Library for Efficient Modeling and Optimization in Networks. Egervary Combinatorial Optimization Research Group, EGRES, Budapest, Hungary. Available at <http://lemon.cs.elte.hu>.
- Estrada, E., and O. Bodin. 2008. Using network centrality measures to manage landscape connectivity. *Ecological Applications* 18: 1810–1825.
- Freeman, L. C. 1977. A set of measures of centrality based on betweenness. *Sociometry* 40:35-41.
- Freeman, L.C., S. P. Borgatti, and D. R. White. 1991. Centrality in valued graphs: a measure of betweenness based on network flow. *Social Networks* 13:141–154.
- Hagberg, A., D. Schult, and P. Swart. 2008. Exploring network structure, dynamics, and function using NetworkX. Pages 11–16 in G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, Pasadena, California.
- Lee-Yaw, J. A., A. Davidson, B. H. McRae, and D. M. Green. 2009. Do landscape processes predict phylogeographic patterns in the wood frog? *Molecular Ecology* 18:1863-1874.
- Long, R., P. MacKay, J. Ray, and W. Zielinski, editors. 2008. *Noninvasive survey methods for carnivores*. Island Press, Washington, D.C.
- McRae, B.H., and P. Beier. 2007. Circuit theory predicts gene flow in plant and animal populations. *Proceedings of the National Academy of Sciences of the USA* 104:19885-19890.
- McRae, B. H., B. G. Dickson, T. H. Keitt, and V. B. Shah. 2008. Using circuit theory to model connectivity in ecology and conservation. *Ecology* 10: 2712-2724.
- Moilanen, A., H. Kujala, and J. Leathwick. 2009. The Zonation framework and software for conservation prioritization. Pages 196-210 in A. Moilanen, K. A. Wilson, and H. P. Possingham, editors. *Spatial conservation prioritization: Quantitative methods and computational tools*. Oxford University Press, Oxford, UK.
- Newman, M. E. J. 2005. A measure of betweenness centrality based on random walks. *Social Networks* 27: 39-54.
- Phillips, S. J., P. Williams, G. Midgley, and A. Archer. 2008. Optimizing dispersal corridors for the Cape Proteaceae using network flow. *Ecological Applications* 18:1200–1211.
- Possingham, H. P., I. R. Ball, and S. Andelman. 2000. Mathematical methods for identifying representative reserve networks. Pages 291–306 in S. Ferson and M. Burgman, editors. *Quantitative methods for conservation biology*. Springer-Verlag, New York, New York, USA.
- Pressey, R. L., M. Cabeza, M. E. Watts, R. M. Cowling, and K. A. Wilson. 2007. Conservation planning in a changing world. *Trends in Ecology and Evolution* 22:583-592.
- Schumaker, N. H., T. Ernst, D. White, J. Baker, and P. Haggerty. 2004. Projecting wildlife responses to alternative future landscapes in Oregon’s Willamette basin. *Ecological Applications* 14:381–400.
- Schwartz, M. K., J. P. Copeland, N. J. Anderson, J. R. Squires, R. M. Inman, K. S. McKelvey, K. L. Pilgrim, L. P. Waits, and S. A. Cushman. 2009. Wolverine gene flow across a narrow climatic niche. *Ecology* 90:3222-3232.
- Spencer, W.D., P. Beier, K. Penrod, K. Winters, C. Paulman, H. Rustigian-Romsos, J. Strittholt, M. Parisi, and A. Pettler. 2010. *California Essential Habitat Connectivity Project: A Strategy for Conserving a Connected California*. Prepared for California Department of Transportation, California Department of Fish and Game, and Federal Highways Administration.
- Theobald, D. M. 2006. Exploring the functional connectivity of landscapes using landscape networks. Pages 416–443 in K. R. Crooks and M. A. Sanjayan, editors. *Connectivity conservation: maintaining connections for nature*. Cambridge University Press, Cambridge, UK.
- Urban, D. L., E. S. Minor, E. A. Treml, and R. S. Schick. 2009. Graph models of habitat mosaics. *Ecology Letters* 12:260–273.
- Urban, D., and T. Keitt. 2001. Landscape connectivity: a graph-theoretic perspective. *Ecology* 82:1205-1218.
- van Rossum, G. and Drake, F. L., editors. 2006. *Python reference manual*. Python Software Foundation. Available at <http://docs.python.org/ref/ref.html>.
- Young, A. G., and G. M. Clarke. 2000. *Genetics, demography and viability of fragmented populations*. Cambridge University Press, Cambridge, UK.

System requirements

Minimal System Requirements

For the 32-bit version: Microsoft Windows XP/Vista/7/8, 1.6 GHz processor, 2 GB RAM

For the 64-bit version: Microsoft Windows XP/Vista/7/8 64-bit, 1.6 GHz processor, 3 GB RAM

Additionally, the source code for the functions based on NetworkX and LEMON are available from www.connectivitytools.org. The Python script which is the basis for the betweenness, current flow, and PageRank functions will run within Python on other platforms such as Linux or Mac OSX. Source code for the LEMON-based functions is also available, and we hope to port this to Linux at a future date.

The various functions differ in requirements for computational time and memory (RAM). The exact current flow BC subset function is RAM-intensive. All other functions, including approximate current flow BC, can be run on most desktop systems (i.e., with <4GB RAM). Time requirements scale rapidly from several minutes for graphs of several thousand nodes to hours for graphs with tens of thousands of nodes. If analyses are taking too long to complete, several options are available to reduce computational time:

- 1) use a coarser resolution for hxn and graph generation (set more hectares per hexagon),
- 2) decrease the proportion of nodes sampled (shortest-path BC), or
- 3) increase the error tolerance (current-flow BC).

Contacts & Credits

Website:

<http://www.connectivitytools.org>

Support and discussion forum:

<http://groups.google.com/group/connectivitytoolkit?hl=en>

Or send support questions to the CAT developer at:

klamathconservation@gmail.com

Model design by Carlos Carroll and Brad McRae.

Software development by Allen Brookes, Kevin Djang, Nathan Schumaker, and Carlos Carroll.

The developers of the Connectivity Analysis Toolkit are grateful to the participants in the NCEAS Connectivity Working Group, especially P. Beier, S. Phillips, R. Pressey, N. Schumaker, D. Theobald, and to V. Shah, for comments and suggestions.

Connectivity Analysis Toolkit is based in part on portions of the following software modules:

Hexsim, developed by Nathan Schumaker et al.: <http://www.epa.gov/hexsim>

LEMON, developed by the Egervary Combinatorial Optimization Research Group, <http://lemon.cs.elte.hu/>

NetworkX, developed by Aric Hagberg et al.: <http://networkx.lanl.gov/>

Python, developed by Guido van Rossum et al.: www.python.org