The Journal of Open Source Software

RivGraph: Automatic extraction and analysis of river and delta channel network topology

Jon Schwenk[*1] **and Jayaram Hariharan**[2]

1 Los Alamos National Laboratory, Division of Earth and Environmental Sciences **2** Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin

DOI: 10.21105/joss.02952

Software

- Review
- Repository
- Archive

Editor: Katy Barnhart

Reviewers:

- @eriknes
- @leotrs

Submitted: 01 January 2021
Published: 09 March 2021

Summary

River networks sustain life and landscapes by carrying and distributing water, sediment, and nutrients throughout ecosystems and communities. At the largest scale, river networks drain continents through tree-like *tributary* networks. At typically smaller scales, river deltas and braided rivers form loopy, complex *distributary* river networks via avulsions and bifurcations. In order to model flows through these networks or analyze network structure, the topology, or connectivity, of the network must be resolved. Additionally, morphologic properties of each river channel as well as the direction of flow through the channel inform how fluxes travel through the network's channels.

`RivGraph` is a Python package that automates the extraction and characterization of river channel networks from a user-provided binary image, or mask, of a channel network (Fig. 1). Masks may be derived from (typically remotely-sensed) imagery, simulations, or even hand-drawn. `RivGraph` will create explicit representations of the channel network by resolving river centerlines as links, and junctions as nodes. Flow directions are solved for each link of the network without using auxiliary data, e.g., a digital elevation model (DEM). Morphologic properties are computed as well, including link lengths, widths, sinuosities, branching angles, and braiding indices. If provided, `RivGraph` will preserve georeferencing information of the mask and will export results as ESRI shapefiles, GeoJSONs, and GeoTIFFs for easy import into GIS software. `RivGraph` can also return extracted networks as `networkx` objects for convenient interfacing with the full-featured `networkx` package (Hagberg et al., 2008). Finally, `RivGraph` offers a suite of topologic metrics that were specifically designed for river channel network analysis (Tejedor et al., 2015b).

*Corresponding author

The Journal of Open Source Software

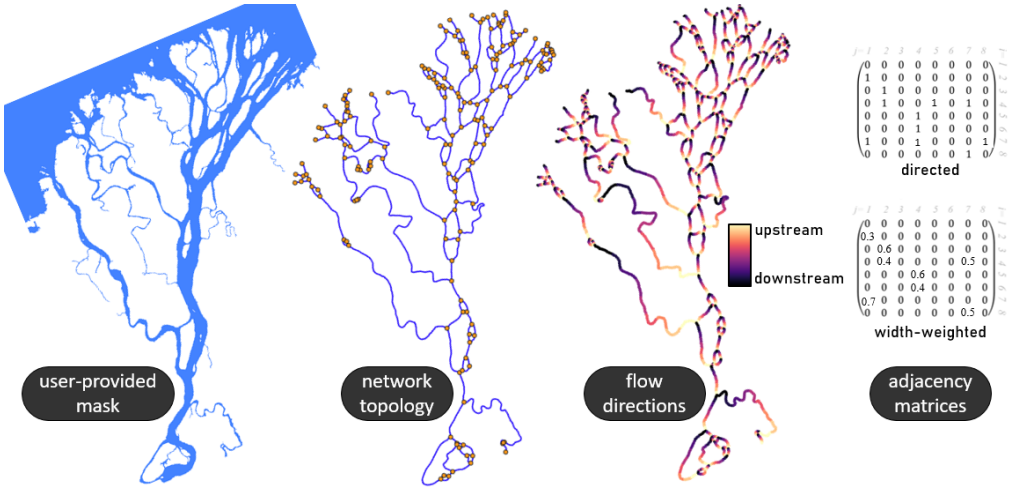

Figure 1: The core functionality of RivGraph for a delta channel network.

Statement of need

Satellite and aerial photography have provided unprecedented opportunities to study the structure and dynamics of rivers and their networks. As both the quantity and quality of these remotely-sensed observations grow, the need for tools that automatically map and measure river channel network properties has grown in turn. The genesis of `RivGraph` is rooted in the work of Tejedor et al. (2017) in a revitalized effort to see river channel networks through the lenses of their network structure. The authors were relegated to time-consuming hand-delineations of the delta channel networks they analyzed. `RivGraph` was thus born from a need to transform binary masks of river channel networks into their graphical representations accurately, objectively, and efficiently.

`RivGraph` has already been instrumental in a number of investigations. The development of the flow directions algorithms itself provided insights into the nature of river channel network structure in braided rivers and deltas (Schwenk et al., 2020). For deltas specifically, `RivGraph`-extracted networks have been used to study how water and sediment are partitioned at bifurcations (Dong et al., 2020), to determine how distance to the channel network plays a controlling role on Arctic delta lake dynamics (Vulis et al., 2020), and to construct a network-based model of nitrate removal across the Wax Lake Delta (Knights et al., 2020). For braided rivers, `RivGraph` was used to extract channel networks from hydrodynamic simulations in order to develop the novel "entropic braiding index" (eBI, Tejedor et al., 2019), and a function for computing the eBI (as well as the classic braiding index) for braided rivers is provided in `RivGraph`. The work of Marra et al. (2014) represented an effort to understand braided rivers through their topologies, although their networks were apparently extracted manually. Ongoing, yet-unpublished work is using `RivGraph` to study river dynamics, delta loopiness, and nutrient transport through Arctic deltas.

We are aware of one other package that extracts network topology from channel network masks. The `Orinoco` Python package (Marshak et al., 2020) uses a fast marching method to resolve the channel network in contrast to `RivGraph`'s skeletonization approach. `Orinoco` uses only a shortest-path approach for setting flow directions rather than `RivGraph`'s exploitation of many morphologic features (including shortest path) to set flow directions. If a DEM of the channel network is available, the Lowpath (Hiatt et al., 2020) add-on to the Topological Tools for Geomorphological Analysis package may be of interest. `RivGraph`'s along-river mesh generation for braided rivers (Fig. 2) was inspired by RivMAP (Schwenk et al., 2017).

Functionality

RivGraph requires the user to provide a binary mask of a channel network. If the provided mask is georeferenced (e.g., a GeoTIFF), RivGraph will export results in the same coordinate reference system (CRS) for easy analysis with a Geographical Information System (GIS) software. Otherwise, a "dummy" CRS is applied, and calculated physical quantities (e.g., length and width) will be in units of pixels. The channel mask is the basis for all RivGraph processing, so the user should consider carefully the features to include and the desired level of smoothing. RivGraph respects the connectivity of the channel mask such that all groups of pixels connected in the mask will be connected in the vectorized representation as well. The user may therefore wish to preprocess their mask to fill small or unwanted islands or smooth channel boundaries. RivGraph's `im_utils()` module contains a number of functions for achieving these tasks, including island-filling and morphological operators. Detailed information about how to create and prepare masks is provided in the documentation.

Basic Functionality

RivGraph was designed with an emphasis on user-friendliness and accessibility, guided by the idea that even novice Python users should be able to make use of its functionality. Anticipated common workflows are gathered into two classes that manage georeferencing conversions, path management, and I/O with simple, clearly-named methods. Beginning users will want to instantiate either a `delta` or a (braided) `river` class and apply the relevant methods, which are as follows:

- `skeletonize()` : skeletonizes the mask; minor conditioning of the skeleton is performed to simplify the topology. For example, if a "+" pattern with the center pixel "off" appears in the skeleton, the center pixel will be added to the skeleton to reduce the number of branchpoints from four to one.
- `compute_network()` : walks along the skeleton to resolve the links and nodes. All pixels in the unpruned skeleton will be visited and therefore represented in the vectorized output.
- `prune_network()` : removes portions of the network that do not contribute meaningfully to its topology. The skeletonization process often results in many "dangling links," or links connected to the network at only one end. During pruning, all dangling links are removed except those connected to inlet or outlet nodes. For the `delta` class, user-provided shoreline and inlet nodes files are required so that RivGraph can prune the network to the shoreline and identify the inlet and outlet nodes. Additionally, bridging links, or links whose removal results in two subnetworks, are removed if one of the resulting subnetworks contains no inlet or outlet nodes. The corresponding subnetwork without inlet or outlet nodes is also removed.
- `compute_link_width_and_length()` : adds width and length attributes to each link. Width is computed via sampling a distance transform image along the link (centerline) coordinates and multiplying by two. Length is the sum of the Euclidean distance between each pair of pixels along a link.
- `assign_flow_directions()` : uses a suite of algorithms to set the flow direction of each link in the network. Separate "recipes" are provided for the `delta` and `river` classes, but users may also create their own. Rationale of the various algorithms and details of recipe construction are given in Schwenk et al. (2020).

Additional methods are available for plotting, exporting GeoTIFFs and geovectors, saving/loading the network, converting to adjacency matrices, computing junction angles, and finding islands.

Braided rivers should be analyzed with the `river` class, which instead of a user-provided shoreline requires a two-character string denoting the *exit sides* of the river with respect to the mask, e.g., 'NS' for a river whose upstream terminus is at the top of the image and downstream at the bottom. `RivGraph` exploits the general direction of the braided river's channel belt to set flow directions and generate an along-river mesh (Fig. 2) that can be used for characterizing downstream changes. In addition to the methods above, the `river` class also features:

- `compute_centerline()` : computes the centerline of the holes-filled river mask (not individual channels)
- `compute_mesh()` : creates a mesh of evenly-spaced transects that are approximately perpendicular to the centerline. The user can specify the mesh spacing, transect width, and degree of smoothing.

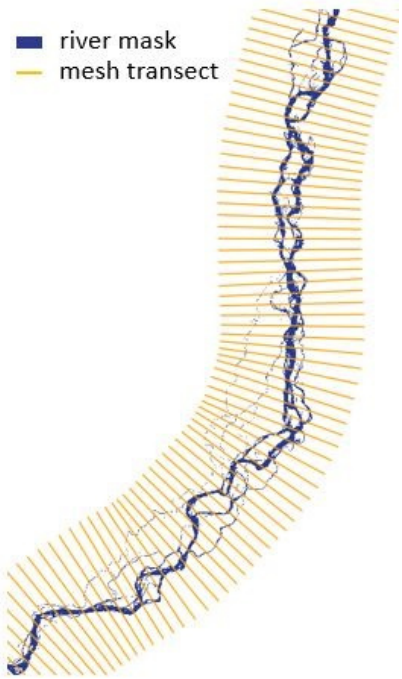

Figure 2: Figure 2. A RivGraph-generated mesh for a mask of the Indus River.

Advanced Functionality

`RivGraph` is organized into a set of modules such that users can find particular functions based on their general class. Customized workflows can be created by calling appropriate functions from these modules, which include

- `classes` : contains the `river` and `delta` classes and associated methods
- `directionality` : algorithms for setting flow directions that are not specific to deltas or braided rivers
- `geo_utils` : functions for handling geospatial data
- `im_utils` : image processing utilities, including morphologic operators
- `io_utils` : functions for reading and writing data and results
- `ln_utils` : functions for building and manipulating the links and nodes of the network
- `mask_to_graph` : the algorithm for converting the mask to a set of links and nodes
- `walk` : functions for walking along the skeleton and identifying branchpoints

- `deltas/delta_directionality` : delta-specific algorithms for setting flow directions
- `deltas/delta_metrics` : functions for computing topologic metrics
- `deltas/delta_utils` : algorithm for pruning deltas and clipping the delta network by the shoreline
- `rivers/river_directionality` : river-specific algorithms for setting flow directions
- `rivers/river_utils` : algorithms for pruning rivers and generating along-river meshes

Dependencies

`RivGraph` relies on functionality from the following Python packages: GDAL (GDAL/OGR contributors, 2020), NumPy (Harris et al., 2020), Matplotlib (Hunter, 2007), GeoPandas (Jordahl et al., 2020), Shapely (Gillies & others, 2007), Fiona (Gillies & others, 2011), pyproj (Snow et al., 2020), scikit-image (van der Walt et al., 2014), OpenCV (Bradski, 2000), networkx (Hagberg et al., 2008), and fastdtw (Slaypni, 2020).

Acknowledgements

We thank Efi Foufoula-Georgiou, Alejandro Tejedor, Anthony Longjas, Lawrence Vulius, Kensuke Naito, and Deon Knights for providing test cases and feedback for `RivGraph`'s development. We are also grateful to Anastasia Piliouras and Joel Rowland for providing valuable insights and subsequent testing of `RivGraph`'s flow directionality algorithms.

`RivGraph` has received financial support from NSF under EAR-1719670, the United States Department of Energy, and Los Alamos National Laboratory's Lab Directed Research and Development (LDRD) program. Special thanks are due to Dr. Efi Foufoula-Georgiou for providing support during the nascent phase of `RivGraph`'s development.

References

Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*.

Dong, T. Y., Nittrouer, J. A., McElroy, B., Il'icheva, E., Pavlov, M., Ma, H., Moodie, A. J., & Moreido, V. M. (2020). Predicting Water and Sediment Partitioning in a Delta Channel Network Under Varying Discharge Conditions. *Water Resources Research*, *56*(11), e2020WR027199. https://doi.org/10.1029/2020WR027199

GDAL/OGR contributors. (2020). *GDAL/OGR geospatial data abstraction software library* [Manual]. Open Source Geospatial Foundation.

Gillies, S., & others. (2007). *Shapely: Manipulation and analysis of geometric objects*.

Gillies, S., & others. (2011). *Fiona is OGR's neat, nimble, no-nonsense API*.

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the 7th Python in Science Conference*, 5.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hiatt, M., Sonke, W., Addink, E. A., Dijk, W. M. van, Kreveld, M. van, Ophelders, T., Verbeek, K., Vlaming, J., Speckmann, B., & Kleinhans, M. G. (2020). Geometry and Topology of Estuary and Braided River Channel Networks Automatically Extracted From Topographic Data. *Journal of Geophysical Research: Earth Surface*, *125*(1), e2019JF005206. https://doi.org/10.1029/2019JF005206

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jordahl, K., Bossche, J. V. den, Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., … Leblanc, F. (2020). *Geopandas/geopandas: V0.8.1*. Zenodo. https://doi.org/10.5281/zenodo.3946761

Knights, D., Sawyer, A. H., Barnes, R. T., Piliouras, A., Schwenk, J., Edmonds, D. A., & Brown, A. M. (2020). Nitrate Removal Across Ecogeomorphic Zones in Wax Lake Delta, Louisiana (USA). *Water Resources Research*, *56*(8), e2019WR026867. https://doi.org/10.1029/2019WR026867

Marra, W. A., Kleinhans, M. G., & Addink, E. A. (2014). Network concepts to describe channel importance and change in multichannel systems: Test results for the Jamuna River, Bangladesh. *Earth Surface Processes and Landforms*, *39*(6), 766–778. https://doi.org/10.1002/esp.3482

Marshak, C., Simard, M., Denbina, M., Nilsson, J., & Van der Stocken, T. (2020). Orinoco: Retrieving a River Delta Network with the Fast Marching Method and Python. *ISPRS International Journal of Geo-Information*, *9*(11), 658. https://doi.org/10.3390/ijgi9110658

Schwenk, J., Khandelwal, A., Fratkin, M., Kumar, V., & Foufoula-Georgiou, E. (2017). High spatiotemporal resolution of river planform dynamics from Landsat: The RivMAP toolbox and results from the Ucayali River. *Earth and Space Science*, *4*(2), 46–75. https://doi.org/10.1002/2016EA000196

Schwenk, J., Piliouras, A., & Rowland, J. C. (2020). Determining flow directions in river channel networks using planform morphology and topology. *Earth Surface Dynamics*, *8*(1), 87–102. https://doi.org/10.5194/esurf-8-87-2020

Slaypni. (2020). Fastdtw: A python implementation of FastDTW. *GitHub Repository*.

Snow, A. D., Whitaker, J., Cochran, M., Bossche, J. V. den, Mayo, C., de Kloe, J., Karney, C., Ouzounoudis, G., Dearing, J., Lostis, G., Heitor, Filipe, May, R., Itkin, M., Couwenberg, B., Berardinelli, G., Badger, T. G., Eubank, N., Dunphy, M., … Wiedemann, B. M. (2020). *Pyproj4/pyproj: 2.6.1 Release*. Zenodo. https://doi.org/10.5281/zenodo.3783866

Tejedor, A., Longjas, A., Edmonds, D. A., Zaliapin, I., Georgiou, T. T., Rinaldo, A., & Foufoula-Georgiou, E. (2017). Entropy and optimality in river deltas. *Proceedings of the National Academy of Sciences*, *114*(44), 11651–11656. https://doi.org/10.1073/pnas.1708404114

Tejedor, A., Longjas, A., Zaliapin, I., & Foufoula-Georgiou, E. (2015a). Delta channel networks: 1. A graph-theoretic approach for studying connectivity and steady state transport on deltaic surfaces: Graph-theoretic approach for delta channel networks. *Water Resources Research*, *51*(6), 3998–4018. https://doi.org/10.1002/2014WR016577

Tejedor, A., Longjas, A., Zaliapin, I., & Foufoula-Georgiou, E. (2015b). Delta channel networks: 2. Metrics of topologic and dynamic complexity for delta comparison, physical inference, and vulnerability assessment. *Water Resources Research*, *51*(6), 4019–4045. https://doi.org/10.1002/2014WR016604

Tejedor, A., Schwenk, J., Kleinhans, M., Carling, P., & Foufoula-Georgiou, E. (2019). The Braiding Index 2.0: eBI. *AGU Fall Meeting*. https://doi.org/10.1002/essoar.10502024.1

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & scikit-image contributors. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

Vulis, L., Tejedor, A., Schwenk, J., Piliouras, A., Rowland, J., & Foufoula-Georgiou, E. (2020). Channel Network Control on Seasonal Lake Area Dynamics in Arctic Deltas. *Geophysical Research Letters*, *47*(7), e2019GL086710. https://doi.org/10.1029/2019GL086710