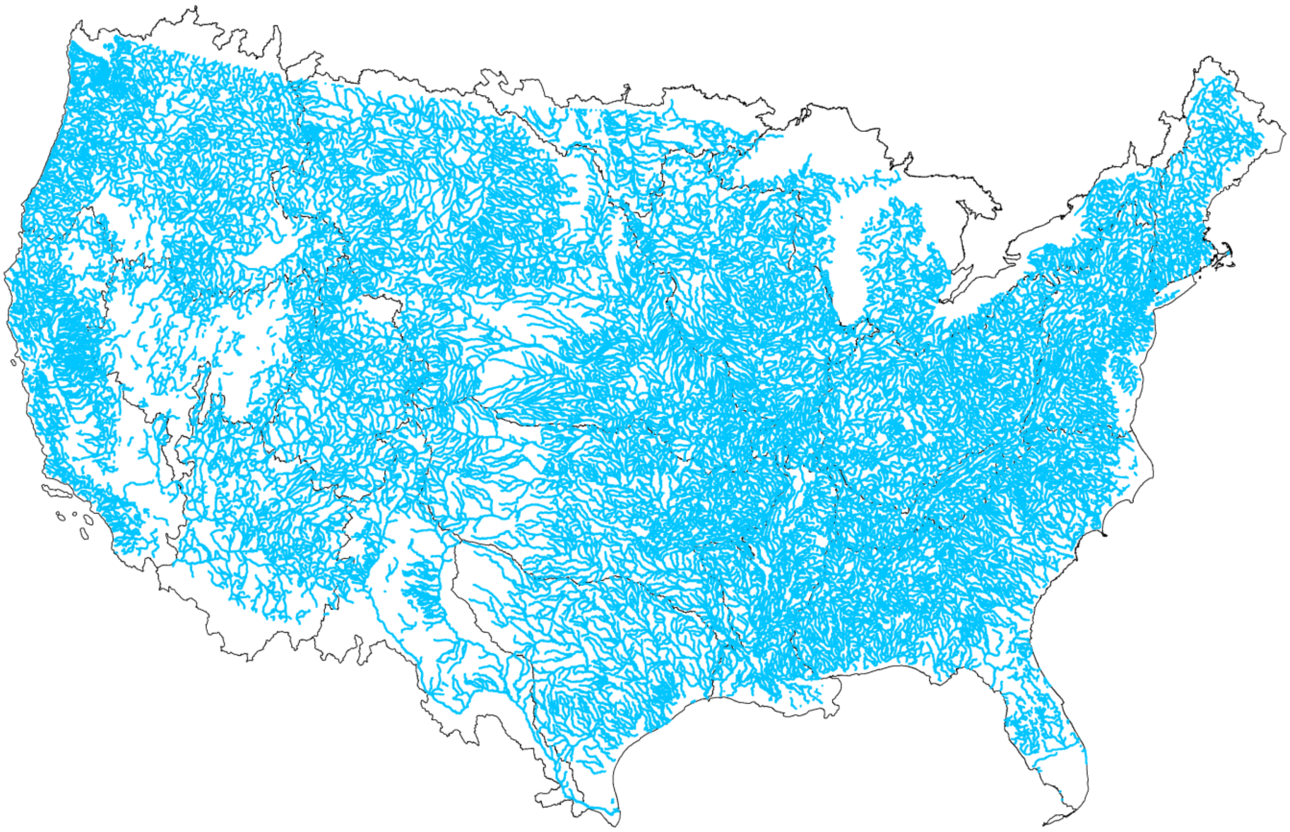
mizuRoute: A River Network Routing Tool (version 1)

User Manual

Naoki Mizukami, Martyn P.Clark, and Kevin Sampson

[1]National Center for Atmospheric Research, Boulder CO

Updated on 9/2/2015



National Center for Atmospheric Research
P.O. Box 3000
Boulder, CO 80307

Table of Contents

1 Overview of mizuRoute river network touting tool

The river network routing tool, mizuRoute version 1, post-processes total runoff (i.e., the sum of surface and subsurface runoff) produced by any hydrologic models or land surface models to estimate spatially distributed streamflow along the river network. Figure 1 shows overall procedures for mizuRoute. The routing routine in the mizuRoute tool uses a two-step process to route basin runoff. First, basin runoff is routed from each hillslope to the river segment using a gamma-distribution-based unit-hydrograph. Second, the delayed flow that enters the river segment is routed to downstream river segments along the river network. The mizuRoute routing routine includes two channel routing schemes; 1) Kinematic Wave Tracking (KWT) and 2) Impulse Response Function-Unit Hydrograph (IRF-UH). The routing time step is the same as runoff output from the hydrologic model, typically an hourly or daily time step. The details on the routing algorithms are described in the main paper.

There are three subsets of the tool included in mizuRoute; 1) topology preprocessor (**process_river_topology.exe**) which augment river network topology necessary for the routing computation, 2) routing program (**runoff_route.exe**), which performs the runoff routing computations, and 3) python and bash scripts to process hydrologic model output (mapping modeled runoff to HRU in Figure 1). Prior to performing river routing, all upstream river segments for each segment are identified with **process_river_topology.exe**. This augmentation of the river network topology is described more in Section 2. The routing program starts with computation of the hillslope routing to compute the delayed runoff from a hydrologic response unit (HRUs) for the corresponding river segment. The delayed runoff from the HRU is poured at the downstream end of the corresponding stream segment. This is denoted by the circled number in the river basin shown in Figure 2. For the river routing, KWT routing and/or IRF-UH routing are performed to route the delayed runoff along the river network and estimate discharge at the outlet of all the segments at one time step. Since headwater catchments do not have an upstream stream segment, the segments attached to headwater HRUs (e.g., basin 1 and 3 in

Figure 2) require no river routing. The outputs (i.e., routed flow at each segment as well as intermediate flow information) are output in NetCDF version 4 (http://www.unidata.ucar.edu/downloads/netcdf/index.jsp). Further information on output NetCDF formatting is given in section 2.3.

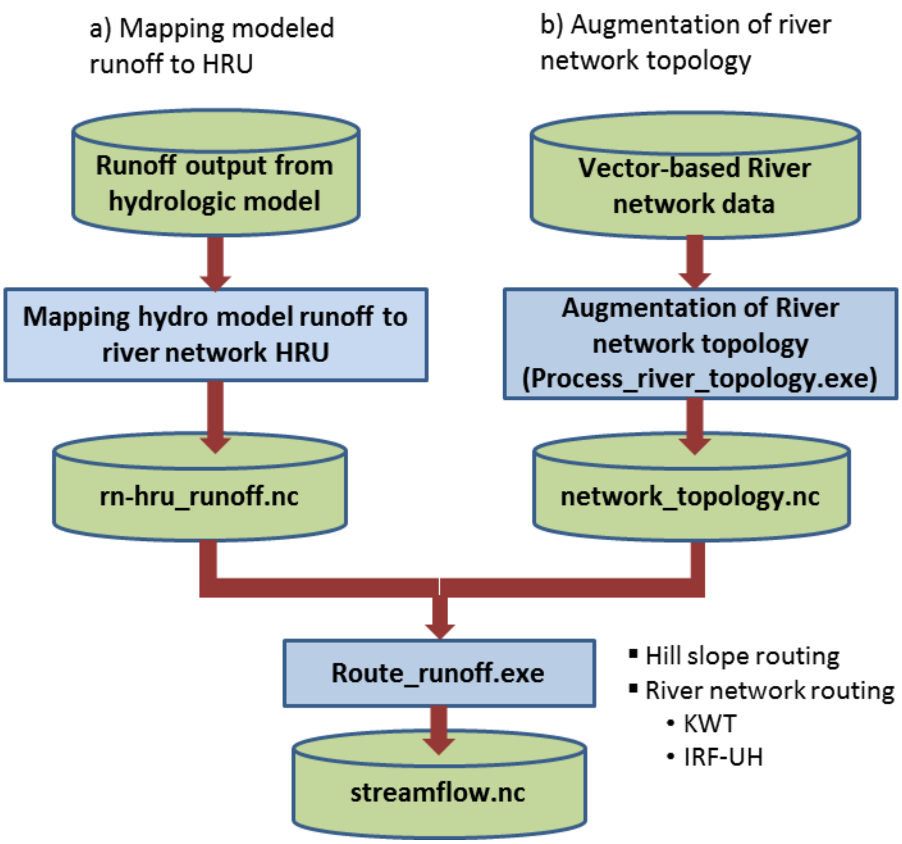


Figure 1. Overall procedure of mizuRoute.

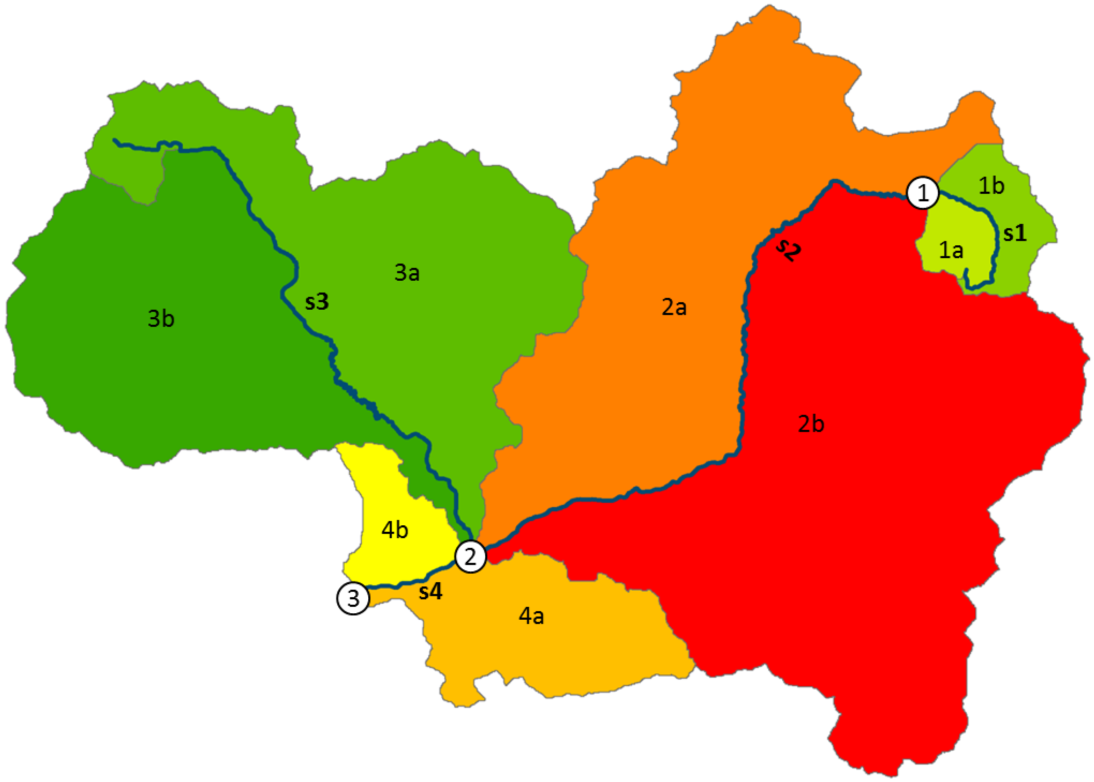


Figure 2. Illustration of HRUs with associated river segments in the river network data. The map is drawn using United States Geological Survey (USGS) Geospatial Fabric (GF) data.

2 Datasets

2.1 Geospatial Fabric (GF) and its augmentation

The manual uses the United States Geological Survey's Geosptial Fabric (GF) datasets (Viger 2014; http://wwwbrr.cr.usgs.gov/projects/SW_MoWS/GeospatialFabric.html) as an example river network dataset. The GF dataset were developed by USGS based on aggregation of finer catchments defined by the 1st version of NHDPlus at point of interests (POI) such as streamflow gage location, river confluence points, etc. The mizuRoute tool utilizes river segments (line geometry – 'nsegment') and catchments (polygon geometry – 'nhru') defining HRUs with some basic physical attributes for both river segment (i.e., length and slope) and HRUs (areas). There is also point data indicating POI location, but this is not required for the routing tool. Table 1 shows all the necessary vector data attributes for mizuRoute. The GF datasets already have two pieces of topological information - 1) ID of downstream segment and 2) association between river segment and HRU. Since the HRU in the GF is left or right bank of each segment (shown in Figure 2), relation between river segment and HRU is one-to-many relation. Although this manual uses GF dataset for the illustration, users can develop their own river network dataset similar to GF by processing Digital Elevation Model (DEM) data with Geographic Information System (GIS). The left and right banks of the segment need not to be defined as the HRU in the river network data. How to generate river network data via digital terrain analysis is out of scope in this manual.

To enable applications on the scale of the continental United States (CONUS), we augment the topological information in GF and produce additional data as a NetCDF input file (See network_topology.nc in Table 2). The augmentation of GF topology is done with topology preprocessor (**process_river_topology.exe**). Specifically, the program first combines two attributes from river segment and HRU polygons to link each segment to HRU polygon into the same attribute table. Second, using the information on downstream segment for each segment, immediate "upstream" river segment(s) from each segment are identified and then all the upstream segments for each segment are listed by tracing upstream segments. Third, we include additional geometric parameters needed for the routing (transferred from GF); the area of HRUs draining to each river segment (basinArea) and sum of all the HRU areas contributing to that river segment (UpstreamArea), and total length of upstream segments (upReachTotalLength). Using the example basin in Figure 1, the basinArea for segment s4 is sum of HRUs 4a and 4b. There are two river paths for outlet 3 (s3-s4, and s1-s2-s4), therefore the topology program computes total length of upstream segments for both paths for outlet 3. In many cases, two river segments are confluent into one river segment. Therefore, many outlet points can have multiple upstream river networks and corresponding total upstream lengths.

Table 1. Attributes in GF dataset required for routing tools.

Vector data	Shape Type	Attribute name	Data type	Descriptions
nhru	polygon	hru_id	int	ID of HRU
		hru_segment	int	ID of segment to which the HRU discharge
		hru_area	float	Area of HRU [m^2]
nsegment	line	seg_id	int	ID of segment
		tosegment	int	ID of immediate downstream segment
		length	float	length of segment [m]
		slope	float	Slope of the channel [-]

2.2 Input datasets for mizuRoute

Table 2 provides all the information contained in two input NetCDF data files for the routing tools: 1) HRU average runoff time series and 2) augmented river network topology. The names of all the variables and attribute as well as NetCDF filename can be freely changed by users, but specified in the control file (see Section 3.2) accordingly.

The runoff NetCDF including the HRU averaged runoff is saved as 2-D variables (runoff (nTime, nHRU**)** or 3-D variables (runoff (nTime, nHRU, nModel**)** for ensemble hydrologic model applications.

Typically, time dimension is the record dimension, but this is not a requirement. Note that most of the existing hydrologic models use a spatial discretization different than the HRUs defined in the river network data (e.g., many hydrologic model use grid). In such case, the HRU-averaged runoff needs to be computed to create runoff NetCDF. This process is done by taking the area-weighted runoff of the intersecting hydrologic model HRUs for each HRU. We developed the python scripts to identify the intersected hydrologic model HRUs for each river network HRU and their fractional areas to the river network HRU area to assist with this process.

The augmented river network topology NetCDF is a more complex dataset with 4 dimensions, including various information about river segment connectivity and the HRU associated with each river segment. This dataset also includes some geometric information such as river segment length and HRU area as described in the previous section. All the required information is given in Table 2.

Table 2. Input NetCDF Dataset required and their attribute information for mizuRoute.

Input data	**Dimension name**	**Variable (attribute) name**
hru_avg_runoff.nc	1. Hru_id(nHru) 2. Time (nTime)	1. hru_id (nHru) – list of hru_id 2. runoff (nTime, nHru) – total runoff time series 3. time (nTime) – list of time stamps
network_topology.nc	1. sHRU(nHru) – nHru=count of HRUs 2. sSeg (nSeg) – nSeg=count of segment 3. sUps(nUps) – nUps= sum of counts of immediate upstream segment over all the segments. 4. sAll(nAll) – nAll= sum of counts of all the upstream segments over all the segments. All the dimensions are index, e.g., [1,…nHru] for sHru dimension.	<u>Segment information</u> 1.reachIndex(nSeg) – segment index [1,..,nSeg] 2.reachID(nSeg) – segment ID 3.reachLength(nSeg) – length of segment (m) <u>HRU information</u> 1.hruIndex(nHru) 2.hru_id(nHru) <u>Downstream information</u> 1. downReachIndex(nSeg) – immediate downstream segment index [1,…nSeg] 2. downReachID(nSeg) – immediate downstream segment ID <u>Upstream information</u> 1. upReachIndex(nUps) – immediate upstream segment index [1,…nUps] 2. upReachID(nUps) – immediate upstream segment ID 4.basinArea (nSeg) –sum of HRU areas draining to the segment 5.upstreamArea (nSeg) – immediate upstream HRU area 6.totalArea(nSeg) – sum of HRU area and all the upstream HRU area 7.reachList(nAll) – list of indices of all the upstream segments 8.upReachTotalLength(nAll) – total length from each upstream segment <u>Miscellaneous information</u> 1.reachStart (nSeg) –start index in upstream reach listed in reachList 2.reachCount (nSeg) –number of all the upstream segments for each segment 3.upReachStart (nSeg) –start index in immediate upstream reach listed in upReachID 4.upReachCount (nSeg) –number of immediate upstream segments for each segment. 5.upHruStart (nSeg) –start index in upstream Hru listed in hru_id 6.upHruCount (nSeg) –number of upstream Hrus for each segment

2.3 Output

Simulated routed runoff at each segment is saved in NetCDF format. Various intermediate runoff values are also included for diagnostic purposes. The output NetCDF includes information on some of river network topology and HRU. Table 3 lists two dimension variables as well as variables related to runoff estimation.

Table 3. Attributes in GF dataset required for routing tools.

Variables	Dimension	Unit	Descriptions
Time	Unlimited(time)	days	time (number of days from reference time). This is record dimension
reachID	(sSeg)	-	Segment ID
instBasinRunoff	(time, sSeg)	m^3/sec	Instantaneous total runoff from contributing HRUs
dlayBasinRunoff	(time, sSeg)	m^3/sec	Delayed total runoff via hillslope routing from contributing HRUs
sumUpstreamRunoff	(time, sSeg)	m^3/sec	Sum of Instantaneous total runoff from all the upstream HRUs at each segment
UpBasRoutedRunoff	(time, sSeg)	m^3/sec	Sum of delayed total runoff from all the upstream HRUs at each segment
KWroutedRunoff	(time, sSeg)	m^3/sec	Routed runoff at each segment with KWT method
IRFroutedRunoff	(time, sSeg)	m^3/sec	Routed runoff at each segment with IRF-UH method

3 Organization of mizuRoute tool

3.1 Directory structure

The mizuRoute tool consists of two sets of the Fortran programs: 1) topology preprocessor (executable- **process_river_topology.exe**) and 2) routing program (executable- **runoff_route.exe**). Each program, source codes, and ancillary data are stored in the separate directories (*ntopo.v1* for topology preprocessor and *route.v1* for routing program). In addition, there are various python and bash scripts to process the ESRI shapefile, and NetCDF in directory called *scripts*. One of the useful scripts is map_poly2poly.py, which identifies the intersected hydrologic model HRUs for each river network HRU and output their fractional areas to the river network HRU area in the NetCDF file. This "mapping NetCDF" is used to map the runoff output from the hydrologic models to the river network HRUs with Upscale.nc.sh. Users should look at procedure.txt for instruction of these scripts.

Tables 4 and 5 show the subdirectory structures for topology preprocessor and routing program, respectively. This section explains what data should be stored in each directory. For topology preprocessor, all the Fortran 90 source codes are located in the *build* directory. The *build* directory also includes the Makefile and the compiled executable is saved in the *bin* directory (See more detail on compiling in section 3.4). For a default, the executable is named **process_river_topology.exe**. The network topology data in *ancillary_data* directory are NetCDF file converted from the river network data in ESRI shapefile (only the attribute information, see Table 1). Output of **process_river_topology.exe**, which is augmented river network topology NetCDF is also stored in *ancillary_data* directory. The *setting* directory includes a control file for **process_river_topology.exe**. The subdirectory structure for the routing program (Table 5) is similar to the topology preprocessor subdirectory. Exceptions are there are runoff input datasets required to run **route_runoff.exe** (See section 2.2) – HRU runoff NetCDF, which should be stored in *input* directory and output of **route_runoff.exe** stored in *output* directory. Another input is augmented river network topology

NetCDF from **process_river_topology.exe**, which should be located in the *ancillary_data* directory. The *setting* directory includes a control file and Fortran namelist containing 6 routing parameters (See Table 6).

Table 4. Directory structure for ntopo_v1 directory.

Subdirectory name	Contents
./ancillary_data	River network attribute NetCDF converted from GF vector data
./bin	Complied executable
./build	Source codes and Makefile
./settings	Control file

Table 5. Directory structure for the route_v1 directory.

Subdirectory name	Contents
./ancillary_data	River network topology NetCDF
./bin	Complied executable
./build	Source codes and Makefile
./input	HRU average runoff NetCDF
./output	Routed flow NetCDF
./settings	Control file and routing parameter namelist file
./verification	Visualization scripts to analyze the streamflow outputs

3.2 Control file

Control file is an argument input for the routing executable (i.e., **process_river_topology.exe** and **runoff_route.exe**). Here the control file for **runoff_route.exe** is described since the control file for **process_river_topology.exe** is similar. The program reads a control file that defines input (runoff NetCDF), output, and ancillary directories (River network topology NetCDF) as well as parameter namelist file defining routing parameters (See section 3.3). The contents in the control file and descriptions of each specification are shown in Table 6 and also see Figure A1 in the Appendix. The control file specifies ID of an outlet segment where river routing is performed for all the upstream segments above the specified segment (3.Outlet segment in table 6). If a negative ID (e.g., -999) is specified, the routing program routes all the segments included in the river network topology NetCDF file. The option of all segment routing is useful when routing over regions where multiple river basins have their own outlets (e.g., New England, Mid-Atlantic, South-east regions etc.).

Table 6. Control file for **process_river_topology.exe**. See example in Figure A1

	Tag of information	**descriptions**
1. Directory	<ancil_dir>	Directory for ancillary data
	<input_dir>	Directory for input runoff data
	<output_dir>	Directory for output data
2. River network topology	<fname_ntop>	NetCDF name of River network topology file
3.Outlet segment	<seg_outlet>	ID of outlet segment
4.Runoff	<fname_qsim>	NetCDF name of HRU runoff time series
	<vname_qsim>	Name of runoff variable in NetCDF
	<vname_time>	Name of time variable in NetCDF
	<vname_hruid>	Name of HRU id variable in NetCDF
	<units_qsim>	Unit of runoff input
	<dt_qsim>	Time interval of the runoff
5. Output	<fname_output	NetCDF name of routed flow time series file
6.Routing parameter	<param_nml>	Namelist name of routing parameters

3.3 Parameter specification

There are two hillslope routing parameters and two routing parameters for each of scheme (KWT and IRF-UH) as shown in Table 6. Currently, all parameters are uniform across all the river segments and HRUs. One future enhancement of the routing tool will be implementing spatially-varying parameters based on the physical characteristics of the HRU (area, slope etc.) for hillslope routing parameters and river channel properties, particularly, manning coefficient in KWT and velocity and diffusivity in IRF-UH.

Table 7. Routing model parameters.

Parameters	Routing methods	descriptions	Default values
A	hillslope	Shape factor [-]	0.01[-]
Θ	hillslope	Time scale factor [T]	86400 [s]
N	KWT	Manning coefficient [-]	0.01 [-]
W	KWT	River Width scale factor [-]	0.001[-]
C	IRF-UH	Wave velocity [LT^{-1}]	1.5 [m/s]
D	IRF-UH	Diffusivity [L^2T^{-1}]	800 [m^2/s]

4 Compilation and required programs

Users can select either GNU (gfortran), Intel (ifort), or PGI (pgf90) by editing the compiler name in the Makefile. Since the routing tool uses NetCDF for input and output, a prerequisite for successful compilation of the code is installation and building of the NetCDF library. The user can download current NetCDF version 4.4.2 (as of 3/4/2015) from Unidata website (http://www.unidata.ucar.edu/downloads/netcdf/index.jsp) with instructions on how to build them. The Makefile specifies the paths to the NetCDF library; therefore users need to modify it in order to locate the NetCDF library. To test the compilation with NetCDF library properly installed and specified, there are small F90 test codes with Makefile in *netcdf_test*.

The Python and Bash processing scripts are also available in GitHub. The python scripts process ESRI Shapefiles and NetCDF data and require GDAL, SHAPELY, NetCDF4 package. In addition, netCDF Operator (NCO; http://nco.sourceforge.net/) is also needed.

5 Example of mizuRoute run

This section provides sample streamflow simulations using mizuRoute for one basin within the upper Colorado River basin (Colorado River near Cameo; see Figure 3). The routing parameters used for this example simulation are listed in Table 7. This example simulation uses the daily historical runoff simulated by Variable Infiltration Capacity (VIC) model from January 1st, 1950 through December 31st, 1950 at 1/8° resolution forced by meteorological data from Maurer et al. (2002). For the routing simulation, the outlet segment was set to the most downstream segment (red-highlighted segment in Figure 3). mizuRoute produces streamflow for each of upstream river segments and modeled streamflow for any upstream segment can be extracted from the NetCDF output file with segment IDs corresponding to the point of interest. Figure 4 shows daily time series of simulated for the period from January 1st, 1950 to December 31st, 1992. All the data used for this example simulation is provided together with the source codes (a matlab code used to create Figure 4 is given in *verification* directory). Also, snapshots of the control file and contents of input data (river network topology NetCDF and runoff NetCDF) used for the simulation are shown in Appendix.

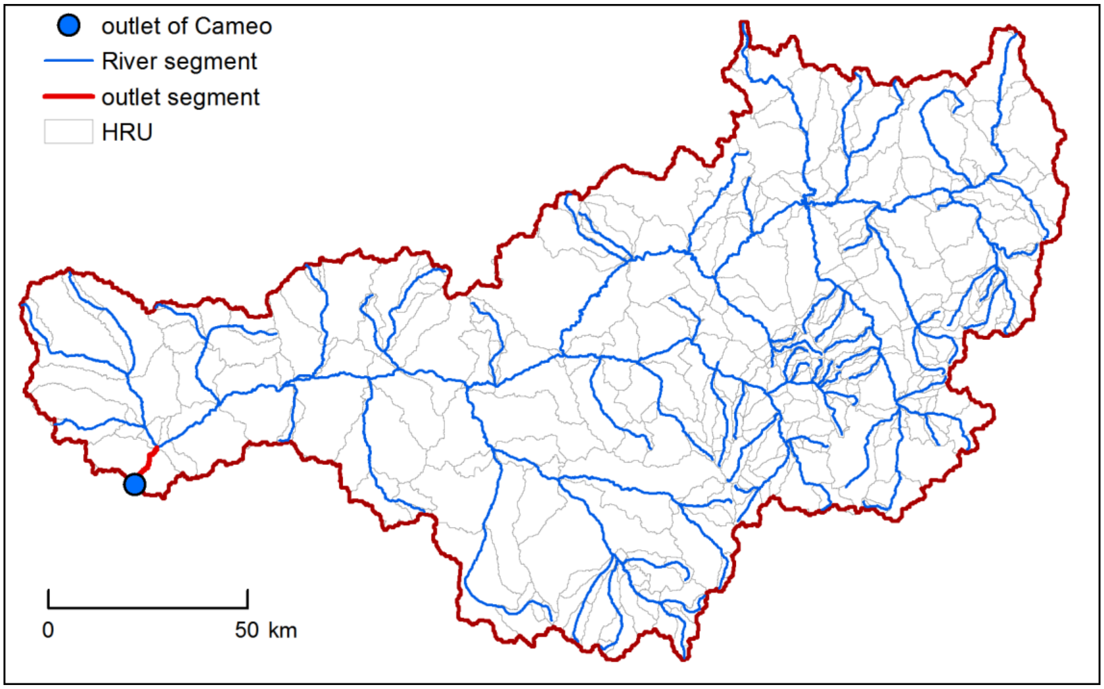


Figure 3. GF River segments and corresponding HRUs for the upstream basins at Cameo, Colorado.

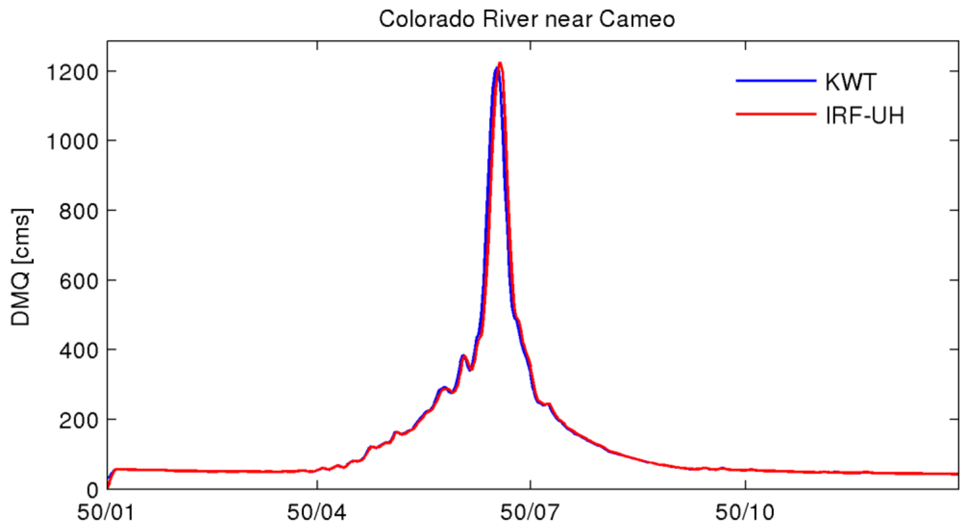


Figure 4. Plots of simulated streamflow at Colorado River near Cameo.

6 Code availability and Test data

The source codes for river network topology and routing programs along with test data are freely available on GitHub (https://github.com/NCAR/mizuRoute).

Appendix

This appendix provides examples of the control file, input and output NetCDF files (input: HRU runoff time series and river network topology, and output: simulated routed runoff).

```
! ****************************************************************************************************************************
! ****************************************************************************************************************************
! ***** DEFINITION OF MODEL CONTROL INFORMATION *******************************************************************************
! ****************************************************************************************************************************
! ****************************************************************************************************************************
! Note: lines starting with "!" are treated as comment lines -- there is no limit on the number of comment lines.
!       lines starting with <xxx> are read till "!"
!
! ****************************************************************************************************************************
! PART 1: DEFINE DIRECTORIES
! --------------------------
<ancil_dir>    /home/mizukami/mizuRoute/route.v1/ancillary_data/  ! directory containing ancillary data
<input_dir>    /home/mizukami/mizuRoute/route.v1/input/           ! directory containing input data
<output_dir>   /home/mizukami/mizuRoute/route.v1/output/          ! directory containing output data
! ****************************************************************************************************************************
! PART 2: DEFINE FINE NAME AND DIMENSIONS
! ---------------------------------------
<fname_ntop>  NTOPO_sample_mod.nc                                 ! dimension name of the stream segments
! ****************************************************************************************************************************
! PART 3: DEFINE DESIRED VARIABLES FOR THE NETWORK TOPOLOGY
! ---------------------------------------------------------
<seg_outlet>   14000645                                           ! seg_id of outlet streamflow segment -9999 for all segments
! ****************************************************************************************************************************
! PART 4: DEFINE RUNOFF FILE
! ----------------------------------
<fname_qsim>   runoff_in.nc                                       ! name of file containing the HRU runoff
<vname_qsim>   RUNOFF                                             ! name of HRU runoff variable
<vname_time>   time                                               ! name of time variable in the HRU runoff file
<vname_hruid>  hru_id2                                            ! name of the HRU id
<units_qsim>   mm/s                                               ! units of runoff
<dt_qsim>      86400                                              ! time interval of the runoff
! ****************************************************************************************************************************
! PART 5: DEFINE OUTPUT FILE
! ---------------------------
<fname_output> q_out.nc                                           ! filename for the model output 1730
! ****************************************************************************************************************************
! PART 6: Namelist file name
! ---------------------------
<param_nml>    /home/mizukami/mizuRoute/route.v1/settings/param.nml.default ! directory containing ancillary data1.0
! ****************************************************************************************************************************
! ****************************************************************************************************************************
```

Figure A1. Control files used for the routing at Colorado River Near Cameo in Section 4.

```
hydro-c1:~/mizuRoute/route.v1/input> ncdump -h runoff_in.nc
netcdf runoff_in {
dimensions:
        time = UNLIMITED ; // (365 currently)
        hru_id2 = 520 ;
variables:
        float RUNOFF(time, hru_id2) ;
                RUNOFF:units = "mm/day" ;
                RUNOFF:_FillValue = -9999.f ;
        float time(time) ;
                time:units = "days since 1949-01-01 00:00:00" ;
                time:long_name = "time" ;
                time:calendar = "noleap" ;
        int hru_id2(hru_id2) ;
                hru_id2:longname = "HRU ID" ;

// global attributes:
                :source = "/home/mizukami/hydro_nm/nHRU_routing/script/Calc_hru_wgtavg_nc.py" ;
                :projection = "lat/lon grid" ;
                :x-resolution = "12 km" ;
                :y-resolution = "12 km" ;
                :matlab\ file = "/d3/mizukami/route/nHRU_routing_v0/preprocess/script/M0_Region2Conus.m" ;
                :NCO = "4.3.4" ;
                :history = "Thu Jun 25 20:03:28 2015: ncks -A RUNOFF_hru_temp.nc runoff_in.nc\nThu Jun 25 2
0:03:19 2015: ncks -O -v hru_id2 test_data/nhru_cameo.nc RUNOFF_hru_temp.nc\nCreated Thu Jun 25 20:00:14 20
15" ;
}
```

Figure A2. NetCDF input containing HRU runoff time series used for the routing at Colorado River Near Cameo in Section 4. This is a screenshot of the NetCDF header output from ncdump utility.

```
hydro-c1:~/mizuRoute/route.v1/ancillary_data> ncdump -h NTOPO_sample_mod.nc
netcdf NTOPO_sample_mod {
dimensions:
	sSeg = 267 ;
	sUps = 266 ;
	sAll = 5337 ;
	sHRU = 520 ;
variables:
	int reachIndex(sSeg) ;
		reachIndex:long_name = "Reach Index (0,1,..nrch-1)" ;
		reachIndex:units = "-" ;
	int reachID(sSeg) ;
		reachID:long_name = "Reach ID" ;
		reachID:units = "-" ;
	double reachSlope(sSeg) ;
		reachSlope:long_name = "Slope of reach" ;
		reachSlope:units = "-" ;
	double reachLength(sSeg) ;
		reachLength:long_name = "Length of reach" ;
		reachLength:units = "m" ;
	double basinArea(sSeg) ;
		basinArea:long_name = "Local basin area" ;
		basinArea:units = "m2" ;
	double upstreamArea(sSeg) ;
		upstreamArea:long_name = "Area upstream of each reach" ;
		upstreamArea:units = "m2" ;
	double totalArea(sSeg) ;
		totalArea:long_name = "Basin area + Upstream area" ;
		totalArea:units = "m2" ;
	int hruIndex(sHRU) ;
		hruIndex:long_name = "Index of hru dimension" ;
		hruIndex:units = "-" ;
	int hru_id(sHRU) ;
		hru_id:long_name = "Hru id" ;
		hru_id:units = "-" ;
	double hru_lon(sHRU) ;
		hru_lon:long_name = "Longitude of hru centroid" ;
		hru_lon:units = "degree" ;
	double hru_lat(sHRU) ;
		hru_lat:long_name = "Latitude of hru centroid" ;
		hru_lat:units = "degree" ;
	double hru_elev(sHRU) ;
		hru_elev:long_name = "Average hru elevation" ;
		hru_elev:units = "m" ;
	double hru_area(sHRU) ;
		hru_area:long_name = "hru area" ;
		hru_area:units = "m2" ;
	double hru_weight(sHRU) ;
		hru_weight:long_name = "Areal weight to total basin area" ;
		hru_weight:units = "-" ;
```
Figure A3. NetCDF input containing augmented river network topology used for the routing at Colorado River Near Cameo in Section 4. This screenshot includes first half of the content.

```
	double reachLat1(sSeg) ;
		reachLat1:long_name = "Start latitude" ;
		reachLat1:units = "-" ;
	double reachLat2(sSeg) ;
		reachLat2:long_name = "End latitude" ;
		reachLat2:units = "-" ;
	double reachLon1(sSeg) ;
		reachLon1:long_name = "Start longitude" ;
		reachLon1:units = "degree" ;
	double reachLon2(sSeg) ;
		reachLon2:long_name = "End longitude" ;
		reachLon2:units = "degree" ;
	double upReachTotalLength(sAll) ;
		upReachTotalLength:long_name = "Total upstream length" ;
		upReachTotalLength:units = "m" ;
	int downReachIndex(sSeg) ;
		downReachIndex:long_name = "Immidiate Downstream reach index" ;
		downReachIndex:units = "-" ;
	int downReachID(sSeg) ;
		downReachID:long_name = "Immidiate Downstream reach ID" ;
		downReachID:units = "-" ;
	int upReachIndex(sUps) ;
		upReachIndex:long_name = "Immidiate Upstream reach index" ;
		upReachIndex:units = "-" ;
	int upReachID(sUps) ;
		upReachID:long_name = "Immidiate Upstream reach ID" ;
		upReachID:units = "-" ;
	int reachList(sAll) ;
		reachList:long_name = "List of all upstream reach indices" ;
		reachList:units = "-" ;
	int reachStart(sSeg) ;
		reachStart:long_name = "start index for list of upstream reaches" ;
		reachStart:units = "-" ;
	int reachCount(sSeg) ;
		reachCount:long_name = "number of upstream reaches in each reach" ;
		reachCount:units = "-" ;
	int upReachStart(sSeg) ;
		upReachStart:long_name = "start index for list of immediate upstream reaches" ;
		upReachStart:units = "-" ;
	int upReachCount(sSeg) ;
		upReachCount:long_name = "number of immediate upstream reaches in each reach" ;
		upReachCount:units = "-" ;
	int upHruStart(sSeg) ;
		upHruStart:long_name = "start index for list of upstream Hru" ;
		upHruStart:units = "-" ;
	int upHruCount(sSeg) ;
		upHruCount:long_name = "number of upstream Hru in each reach" ;
		upHruCount:units = "-" ;
}
```
Figure A4. The same file as Figure A3 except for second half of the content.

```
netcdf q_out {
dimensions:
        time = UNLIMITED ; // (365 currently)
        sSeg = 267 ;
        sUps = 5337 ;
variables:
        double time(time) ;
                time:long_name = "time" ;
                time:units = "days since 1949-01-01 00:00:00" ;
        int reachID(sSeg) ;
                reachID:long_name = "reach ID" ;
                reachID:units = "-" ;
        int reachOrder(sSeg) ;
                reachOrder:long_name = "processing order" ;
                reachOrder:units = "-" ;
        int reachList(sUps) ;
                reachList:long_name = "list of upstream reaches" ;
                reachList:units = "-" ;
        int listStart(sSeg) ;
                listStart:long_name = "start index for list of upstream reaches" ;
                listStart:units = "-" ;
        int listCount(sSeg) ;
                listCount:long_name = "number of upstream reaches in each reach" ;
                listCount:units = "-" ;
        double basinArea(sSeg) ;
                basinArea:long_name = "local basin area" ;
                basinArea:units = "m2" ;
        double upstreamArea(sSeg) ;
                upstreamArea:long_name = "area upstream of each reach" ;
                upstreamArea:units = "m2" ;
        double instBasinRunoff(time, sSeg) ;
                instBasinRunoff:long_name = "instantaneous basin runoff in each reach" ;
                instBasinRunoff:units = "m3/s" ;
        double dlayBasinRunoff(time, sSeg) ;
                dlayBasinRunoff:long_name = "delayed basin runoff in each reach" ;
                dlayBasinRunoff:units = "m3/s" ;
        double sumUpstreamRunoff(time, sSeg) ;
                sumUpstreamRunoff:long_name = "sum of upstream runoff in each reach" ;
                sumUpstreamRunoff:units = "m3/s" ;
        double KWTroutedRunoff(time, sSeg) ;
                KWTroutedRunoff:long_name = "KWT routed runoff in each reach" ;
                KWTroutedRunoff:units = "m3/s" ;
        double UpBasRoutedRunoff(time, sSeg) ;
                UpBasRoutedRunoff:long_name = "sum of upstream basin routed runoff in each reach" ;
                UpBasRoutedRunoff:units = "m3/s" ;
        double IRFroutedRunoff(time, sSeg) ;
                IRFroutedRunoff:long_name = "IRF routed runoff in each reach" ;
                IRFroutedRunoff:units = "m3/s" ;
}
```

Figure A5. NetCDF output file containing routed runoff simulations from the routing at Colorado River Near Cameo in Section 4.

References

Maurer, E. P., A. W. Wood, J. C. Adam, D. P. Lettenmaier, and B. Nijssen, 2002: A Long-Term Hydrologically Based Dataset of Land Surface Fluxes and States for the Conterminous United States, *Journal of Climate*, 15, 3237-3251,

Viger, R. J., 2014, Preliminary spatial parameters for PRMS based on the Geospatial Fabric, NLCD2001 and SSURGO, US Geological Survey,, http://dx.doi.org/doi:10.5066/F7WM1BF7