**Source Tracking Algorithm - User Manual**

**Authors: Sai S. Nudurupati & Erkan Istanbulluoglu**

**What is Source Tracking Algorithm?**

STA is a Landlab utility that operates on a grid, which is a Landlab’s RasterModelGrid instance.

**What does the Source Tracking Algorithm do?**

STA traverses the grid, and records all upstream nodes for each node, given the flow directions. Using the array that associates each grid’s node to a coarser grid ID, it converts the

This algorithm is a customized variant of ‘flow accumulation’ algorithm. This algorithm uses brute force method, which visits each node within a Model Domain (MD), sometimes multiple times, to calculate all unique upstream contributing Hydrologic Source Domain (HSD) nodes and their fractions of contribution.

Expand on ”to calculate all unique upstream contributing Hydrologic Source Domain (HSD) nodes” meaning.

Here you need to directly tell STA returns what results.

We may just give the dictionaries that the reader will see in the last line up here and with a line of explanation, so if they want to read it they will if they don’t then they know the minimum they need to know from this.

HSD is a generic name given to a field of vertical flux output such as groundwater recharge or runoff, R [L/T], or evapotranspiration, ET [L/T], obtained from a distributed hydrologic model. Typical examples that may require running STA include averaging a distributed field of R or ET upstream of a node. HSD can be at a coarser resolution than the MD. In the application illustrated here, the STA is used to return upslope contributing HSD cell IDs and their fractions at each node of the MD in order to calculate steady-state relative wetness for mapping landslide probability in Strauch et al. (GMD, submitted).

MD - Model Domain in 30-m grid resolution. HSD: R [mm/d] output from the Variable Infiltration Capacity (VIC) macroscale hydrologic model (Liang et al., 1994) in 1/16th degree resolution. This coarser grid is first reshaped to the MD resolution, thus, each MD grid retains the HSD ID of the coarser HSD grid.

Note: This utility only works on Landlab’s RasterModelGrid (structured grid) object.

**Inputs**:

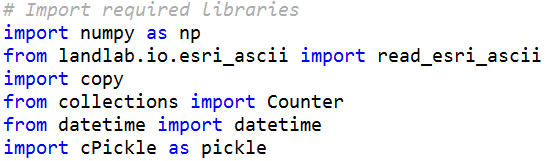
1. **elevation.txt**: Elevation data - ESRI ascii DEM
2. **flow\_direction.txt**: Flow Directions (D4/D8) from ESRI ArcGIS - ESRI ascii format
3. **vic\_idsnoca.txt**: HSD grid resampled to MD - ESRI ascii format
4. **slp\_g16msk.txt**: Mask for Areas with slope greater than 16 deg - ESRI ascii format

**Outputs**:

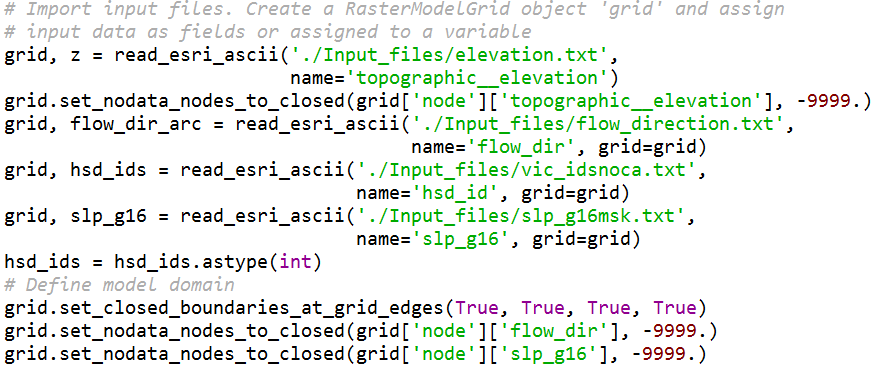
1. **dict\_uniq\_ids.p**: (*dictionary*) Mapping between MD node and unique (i.e., no repeats) upstream contributing HSD nodes. Returns a list/dictionary of upstream contributing HSD nodes for each MD node ID.
2. **dict\_coeff.p**: (*dictionary*) Mapping between MD node and fractions of contribution of each unique upstream HSD nodes. Returns a list/dictionary of fractions of each HSD contributing to a MD node

**Basic Steps in this Algorithm**:

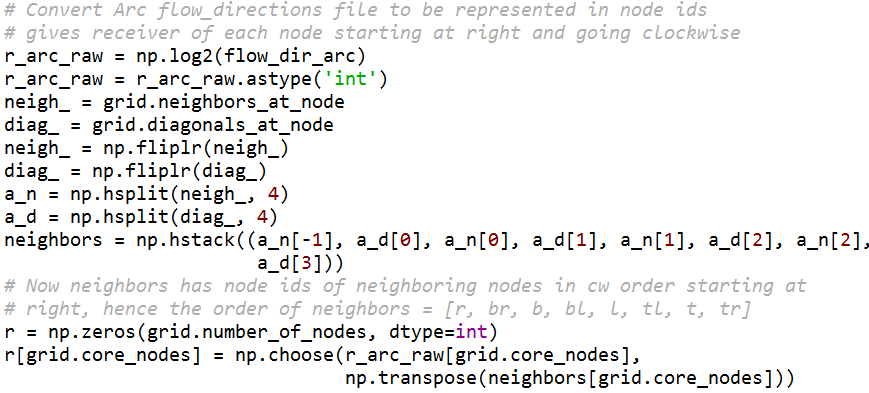
1. **Import necessary libraries**: numpy, *read\_esri\_ascii()* (Landlab utility), copy, collections.Counter, cPickle.pickle are used. Datetime.datetime is optional.



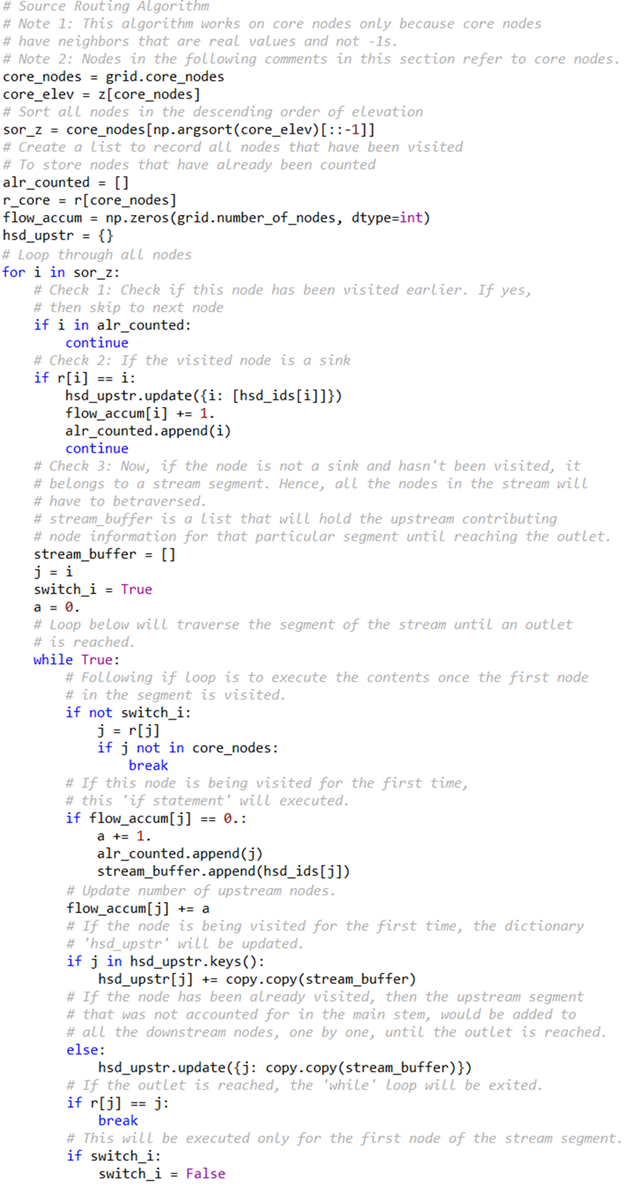
1. **Import inputs and define model domain**: Import all input files (mentioned above) using *read\_esri\_ascii()*. *read\_esri\_ascii()* imports an ESRI ascii file and instantiates a RasterModelGrid. If a ‘name’ is provided through optional argument ‘name’, a nodal ‘field’ will be created with the given ‘name’. If a ‘grid’ is provided through an optional argument ‘grid’, the data will be added to an existing ‘grid’.



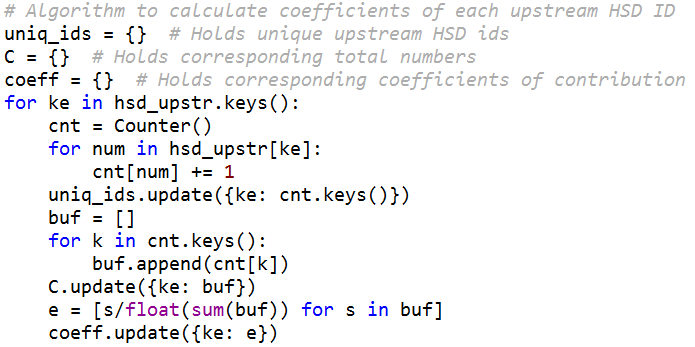
1. **Determine receiver nodes**: Input Flow Directions in ESRI ArcGIS format is converted to Landlab’s receiver nodes, which are nodes XXXXX. If a For a node ‘x’ in MD, the array element ‘r[x]’ would point to the node that will receive water from node ‘x’.



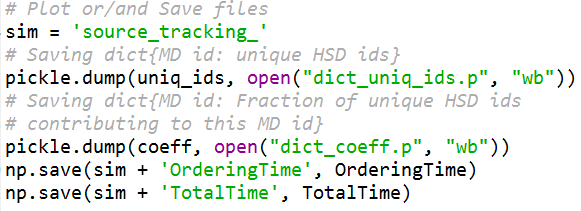
1. **Traverse all MD nodes and record upstream contributing nodes**: All MD nodes are traversed and for each MD node, all upstream contributing HSD node IDs are recorded in the *dictionary* called ‘hsd\_upstr’. ‘hsd\_upstr’ ({MD node id: [upstream contributing HSD ids]}) maps each MD node to a list of all HSD ids that are upstream to this MD node.



1. **Process ‘hsd\_upstr’ dictionary into two dictionaries**: Once all MD nodes have been traversed, the dictionary ‘hsd\_upstr’ will be processed to create two dictionaries, ‘uniq\_ids’ and ‘coeff’. ‘uniq\_ids’ ({MD node id: [unique upstream contributing HSD ids]}) maps each MD node id to a list of all unique HSD ids that are upstream to this MD node. ‘coeff’ ({MD node id: [fractions of contribution of unique upstream contributing HSD ids]}) maps each MD node id to a list of fractions of contributions of all unique upstream HSD ids (in the same order as uniq\_ids[MD node]).



1. **Optionally save the dictionaries**: The library *cPickle* is used to store the dictionaries as outputs (called *pickles*) that can be easily loaded into other Python scripts if the STA utility is executed separately from where it is used using the syntax:  
   *HSD\_id\_dict = pickle.load(open('dict\_uniq\_ids.p', 'rb'))*.



Note: We assumed that the user of this utility/reader of this manual has working knowledge of Python (programming language).

Add references for D-8 flow accumulation algorithm and any others.