## Package ‘SBN’

January 17, 2022

**Title** Generate Stochastic Branching Networks  
**Version** 1.0.0  
**Description** Generate Stochastic Branching Networks ('SBNs'). Used to model the branching structure of rivers.  
**License** MIT + file LICENSE  
**Encoding** UTF-8  
**RoxygenNote** 7.1.2  
**Imports** igraph, stats

**URL** [https://flee598.github.io/SBN/](https://flee598.github.io/SBN/)

**NeedsCompilation** no

**Author** Finnbar Lee [aut, cre]

**Maintainer** Finnbar Lee <lee.finnbar@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-17 19:42:42 UTC

### R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbn_change_dir</td>
<td>2</td>
</tr>
<tr>
<td>sbn_create</td>
<td>2</td>
</tr>
<tr>
<td>sbn_down_dir</td>
<td>3</td>
</tr>
<tr>
<td>sbn_get_downstream</td>
<td>4</td>
</tr>
<tr>
<td>sbn_get_hw</td>
<td>4</td>
</tr>
<tr>
<td>sbn_get_outlet</td>
<td>5</td>
</tr>
<tr>
<td>sbn_get_upstream</td>
<td>5</td>
</tr>
<tr>
<td>sbn_strahler</td>
<td>6</td>
</tr>
<tr>
<td>sbn_to_mtx</td>
<td>7</td>
</tr>
</tbody>
</table>

**Index** 8
sbn_change_dir  
*Change the upstream/downstream direction of an SBN*

**Description**

Change the upstream/downstream direction of an SBN to either, reversed or undirected.

**Usage**

```r
sbn_change_dir(g, method = c("rev", "undir"))
```

**Arguments**

- `g`: a river network as an igraph object. Must be a downstream directed graph.
- `method`: one of "rev" or "undir", determining what to convert the network to.

**Value**

A river network as an igraph object.

**Examples**

```r
g <- sbn_create(10, 0.7)
sbn_change_dir(g, method = "rev")
```

---

sbn_create  
*Create SBNs*

**Description**

An SBN river network as a downstream directed igraph object.

**Usage**

```r
sbn_create(n, p)
```

**Arguments**

- `n`: desired number of nodes.
- `p`: branching probability, from 0 - 1. Passed to `stats::rbinom()`, the probability of success in two attempts at adding upstream branches.
Details

SBNs are generated using a stochastic branching process. The network generation process starts from an initial downstream node (the river mouth). At each iteration a random node in the network, with no upstream connections is selected, and zero, one or two nodes are added upstream of it, depending on a branching probability \( p \). This process is repeated until a pre-determined number of nodes across the entire network is attained \( n \).

Value

A river network as an igraph object.

Examples

sbn_create(10, 0.7)

___________________________

sbn_down_dir  Convert to a downstream directed network
___________________________

Description

Convert an upstream directed or non-directed network to a downstream directed network.

Usage

sbn_down_dir(g, mouth)

Arguments

g  a river network as an igraph object.
mouth  river mouth vertex id.

Value

A downstream directed network.

Examples

g <- sbn_create(10, 0.7)

# to undirected
g <- sbn_change_dir(g, method = "undir")

# undirected to downstream directed
sbn_down_dir(g, mouth = 1)
sbn_get_downstream  \hspace{1cm} \textit{Find all downstream nodes}

\textbf{Description}

Find all nodes downstream of a given node.

\textbf{Usage}

\begin{verbatim}sbn_get_downstream(g, node)\end{verbatim}

\textbf{Arguments}

- \texttt{g} \hspace{1cm} a river network as an igraph object. Must be a downstream directed graph.
- \texttt{node} \hspace{1cm} target node to get all downstream nodes of.

\textbf{Value}

a vector of downstream node id’s.

\textbf{Examples}

\begin{verbatim}g <- sbn_create(10, 0.7)
sbn_get_downstream(g, 10)\end{verbatim}

sbn_get_hw  \hspace{1cm} \textit{Find all headwater nodes}

\textbf{Description}

Find all headwater nodes in a network.

\textbf{Usage}

\begin{verbatim}sbn_get_hw(g)\end{verbatim}

\textbf{Arguments}

- \texttt{g} \hspace{1cm} a river network as an igraph object. Must be a downstream directed graph.

\textbf{Value}

A vector of headwater node id’s.


**sbn_get_outlet**

*Find river mouth node*

**Description**

Find river mouth node from a directed graph.

**Usage**

```
sbn_get_outlet(g)
```

**Arguments**

- `g`: a river network as an igraph object. Must be a downstream directed graph.

**Value**

An integer identifying the id of river mouth node.

**Examples**

```
g <- sbn_create(10, 0.7)
sbn_get_hu(g)
sbn_get_outlet(g)
```

---

**sbn_get_upstream**

*Find all nodes upstream of a given node*

**Description**

Find all nodes upstream of a given node.

**Usage**

```
sbn_get_upstream(g, node)
```

**Arguments**

- `g`: a river network as an igraph object. Must be a downstream directed graph.
- `node`: target node to get all upstream nodes of.
**Value**

A vector of upstream node id’s.

**Examples**

```r
g <- sbn_create(10, 0.7)
sbn_get_upstream(g, 2)
```

---

**sbn_strahler**

*Get node strahler order*

**Description**

Calculate the reach (node) Strahler for all nodes in a river network. The function will not work if any of the nodes in the network have more than two adjacent upstream reaches (e.g. some networks generated by the OCNet package).

**Usage**

```r
sbn_strahler(g)
```

**Arguments**

- **g**: a river network as an igraph object. Must be a downstream directed graph.

**Value**

a vector of stream Strahler orders.

**Examples**

```r
g <- sbn_create(10, 0.7)
sbn_strahler(g)
```
sbn_to_mtx

Convert network to various adjacency/distance matrix formats

Description
Convert a downstream directed SBN to various adjacency or distance matrix formats.

Usage
sbn_to_mtx(
  g,
  method = c("dwn_mtx", "undir_mtx", "up_mtx", "n2n_dist_up", "n2n_dist_dwn",
              "n2n_dist_undir"),
  unconnected = Inf,
  weights = NULL
)

Arguments

  g                  a river network as an igraph object. Must be a downstream directed graph.
  method             one of "dwn_mtx", an adjacency matrix for a downstream directed SBN, "up_mtx", an adjacency matrix for a upstream directed SBN, "undir_mtx", an adjacency matrix for a undirected SBN,"n2n_dist_up", "n2n_dist_dwn" or "n2n_dist_undir", an adjacency matrix of upstream, downstream or undirected node to node distances.
  unconnected        when generating node-to-node distance matrices, what value should be used for unconnected elements. For example, in a downstream directed network, all upstream links are considered unconnected. Default value is Inf but other options are possible, such as NA or 0.
  weights            passed to igraph::shortest.paths(). Possibly a numeric vector giving edge weights. If this is NULL and the graph has a weight edge attribute, then the attribute is used. If this is NA then no weights are used (even if the graph has a weight attribute).

Value
An adjacency or distance matrix.

Examples

  g <- sbn_create(10, 0.7)
  sbn_to_mtx(g, method = "dwn_mtx")
Index

igraph::shortest.paths(), 7

sbn_change_dir, 2
sbn_create, 2
sbn_down_dir, 3
sbn_get_downstream, 4
sbn_get_hw, 4
sbn_get_outlet, 5
sbn_get_upstream, 5
sbn_strahler, 6
sbn_to_mtx, 7
stats::rbinom(), 2