

betweenness_w

Last Updated Monday, 03 August 2009

This function calculates betweenness scores for nodes in a weighted network based on the on the distance_w-function. It also outputs the edge betweenness.

Note: This algorithm relies on the RBGL package's implementation of the boost C++-library. This implementation includes Brandes' (2001) algorithm, and finds multiple paths if they have exactly the same distance. For example, if one path is found over the direct tie with a weight of 1 (distance = $1/1 = 1$) and a second path is through an intermediary node with two ties with weights of 2 (distance = $1/2 + 1/2 = 1$), the two paths have exactly the same distance. However, if there is a third path through two intermediaries with three ties with weights of 3 (distance = $1/3 + 1/3 + 1/3$), computers read these values as 0.3333333 and the sum of these values is 0.9999999. Thus, it is not exactly equal to the distance of the other two paths. In fact, this path is considered shorter than the others ($0.9999999 < 1$).

```

Example## load tnet
library(tnet)## load sample data
sampledata <- rbind(
  c(1,2,4),
  c(1,3,2),
  c(2,1,4),
  c(2,3,4),
  c(2,4,1),
  c(2,5,2),
  c(3,1,2),
  c(3,2,4),
  c(4,2,1),
  c(5,2,2),
  c(5,6,1),
  c(6,5,1))## Run function
betweenness_w(sampledata)
[[1]]
  node betweenness
[1,] 1      0.0
[2,] 2      8.5
[3,] 3      0.0
[4,] 4      0.0
[5,] 5      4.0
[6,] 6      0.0[[2]]
  i j betweenness
[1,] 1 2      4.5
[2,] 1 3      0.5
[3,] 2 3      4.5
[4,] 2 4      5.0
[5,] 2 5      8.0
[6,] 5 6      5.0

```