# Extending the ConScape Library for Infinite Temperature Parameter

Pierre Leleux & Marco Saerens

May 2023

## 1 Introduction

The ConsCape Julia library [5] provides tools for connectivity modeling in ecological landscape based on the Randomized Shortest Path (RSP) framework [4, 6].

This framework models movement flows in a graph and interpolates between a random behavior and an optimal one, controlled by an inverse temperature parameter $\theta = 1/T$. It reflects the movement of a random walker starting from a source and moving towards a target, that follows a set of biased transition probabilities. The intensity of the bias is controlled by the parameter $\theta$. When $\theta$ gets larger, the agent is more likely to pick efficient transitions to reach the target. Thus, when $\theta \to \infty$ the agent starts behaving optimally, and follows the least-cost path(s) to its target. On the contrary, a value of $\theta \to 0^+$ results in a purely random behavior, where the agent makes moves according to the reference (unbiased) transition probabilities.

The ConSCape library allows to compute, based on this framework, distances between nodes (RSP dissimilarity) and betweenness metrics, among others. For the time being, the ConScape library handles the RSP framework for regular $\theta$ values (that is $\theta \in ]0, \infty[$), as well as the "purely optimal" extreme case ($\theta \to \infty$). This paper describes the ConScape library extension with new algorithms to tackle the other extreme case $\theta = 0$ (pure random).

## 2 The expected cost

The RSP dissimilarity represents the expected cumulative cost of the agent's journey to its target and therefore interpolates between the commute-cost distance (see e.g. [2]) and the least-cost distance for values of $\theta$ tending towards 0 and $\infty$ respectively.

Let us denote $\mathbf{P}$, the transition probability matrix and $\mathbf{C}$ the transition cost matrix, such that $p_{ij}$ represents the probability of making a transition to node $j$ if currently standing on node $i$ and $c_{ij}$ a cost that is incurred by doing so.

Furthermore, we also denote $\bar{c}_i(t)$ the expected cumulative cost to reach node $t$ from node $i$, following the transition probabilities $\mathbf{P}$.

Intuitively, the value $\bar{c}_i(t)$ can be computed based on the values of its successors as:

$$\begin{cases} \bar{c}_i(t) = \sum_{j=1}^{n} p_{ij}(c_{ij} + \bar{c}_j(t)) & \text{for } i \neq t \\ \bar{c}_t(t) = 0 & \text{(boundary condition)} \end{cases} \tag{1}$$

The first term from Equation (1) can be rewritten as follows:

$$\bar{c}_i(t) = \sum_{j=1}^{n} p_{ij}c_{ij} + \sum_{j=1}^{n} p_{ij}\bar{c}_j(t) \tag{2}$$

in matrix form:

$$\bar{\mathbf{c}}(t) = \bar{\mathbf{p}} + \mathbf{P}\bar{\mathbf{c}}(t) \tag{3}$$

where $\bar{\mathbf{p}} = (\mathbf{P} \circ \mathbf{C})\mathbf{e}$, where $\circ$ is the elementwise (Hadamard) product, and $\mathbf{e}$ is a vector full of 1's. Equation (3) thus provides the following system of linear equation to solve for expected cost:

$$(\mathbf{I} - \mathbf{P})\bar{\mathbf{c}}(t) = \bar{\mathbf{p}} \tag{4}$$

Finally, this system of linear equations should be modified to comply with the boundary condition. It is done by removing the row related to the computation of $\bar{c}_t(t)$ and replace it with the boundary condition from Equation (1), for this system of linear equation to be solvable.

The entire expected cost matrix $\bar{\mathbf{C}}$ can be computed by iterating over all target nodes $t$. Details on the computation of the matrix $\bar{\mathbf{C}}$ can be found in Algorithm 1.

**Algorithm 1** Computation of the commute cost matrix on a directed graph.

**Input:**
- A directed, strongly connected, graph $G$ containing $n = |\mathcal{V}|$ nodes.
- The $n \times n$ reference transition probabilities matrix $\mathbf{P}$ of $G$.
- The $n \times n$ non-negative cost matrix $\mathbf{C}$ assigned to $G$.

**Output:**
- The $n \times n$ directed commute cost matrix $\bar{\mathbf{C}}$.

1. $\bar{\mathbf{C}} \leftarrow \mathbf{e}\mathbf{e}^{\mathrm{T}}0$ {Initialization of the expected cost matrix}
2. **for** $t \in \mathcal{V}$ **do** {loop over all target nodes}
3. $\quad \bar{\mathbf{p}} \leftarrow (\mathbf{P} \circ \mathbf{C})\mathbf{e}$
4. $\quad \mathbf{B} \leftarrow \mathbf{I} - \mathbf{P}$
5. $\quad \mathbf{B}[t, :] \leftarrow \mathbf{0}\mathbf{e}^{\mathrm{T}}$ {Replace the row of $t$ from the system of equations}
6. $\quad \mathbf{B}[t, t] \leftarrow 1$ {Set up boundary condition in $\mathbf{B}$}
7. $\quad \bar{\mathbf{p}}[t] \leftarrow 0$ {Set up boundary condition in $\bar{\mathbf{p}}$}
8. $\quad \bar{\mathbf{c}}(t) \leftarrow \mathbf{B} \backslash \bar{\mathbf{p}}$ {Solve the system of linear equations $\mathbf{B}\bar{\mathbf{c}}(t) = \bar{\mathbf{p}}$ for $\bar{\mathbf{c}}(t)$}.
9. $\quad \bar{\mathbf{C}}[:, t] \leftarrow \bar{\mathbf{c}}(t)$
10. **end for**
11. **return** $\bar{\mathbf{C}}$

# 3 Node and edge betweenness

The ConScape library offers two weighted betweenness measures, to incorporate the quality of the pair source target (quality-weighted betweenness) or their proximity (proximity-weighted betweenness).

## 3.1 Quality-weighted betweenness

The quality-weighted ($q$-weighted) betweenness is based on the movement flow in each node (or edge, depending if we are interested in a node betweenness or edge betweenness). It is computed as the sum of the flow for every pair-source target, weighted by their respective quality.

### 3.1.1 Node betweenness

We thus have two vectors $\mathbf{q}_s$ and $\mathbf{q}_t$ (stored inside diagonal matrices $\mathbf{Q}_s$ and $\mathbf{Q}_t$ in the ConScape library) that indicate the quality of each node when considered as a source or as a target. The random-walk equivalent of the $q$-weighthed RSP betweenness for a given node is defined as follows:

$$\mathrm{bet}_i^{q-\mathrm{RW}} = \sum_{s,t=1}^{n} q_s^s \bar{n}_i^{st} q_t^t \tag{5}$$

where $\bar{n}_i^{st}$ represents the expected flow in node $i$ of a random walker starting in $s$ and ending in $t$.

In order to obtain the flows in node $i$, let us first start by defining the fundamental matrix[1] $\mathbf{Z}$ as defined in [1]:

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{e}\boldsymbol{\pi}^{\mathrm{T}})^{-1} \tag{6}$$

where $\boldsymbol{\pi}$ is the stationary distribution of the random walk following $\mathbf{P}$.

From this fundamental matrix, it can be shown [3] that the node flow can be computed as

$$\bar{n}_i^{st} = z_{si} - z_{ti} + \pi_i h_{st} \tag{7}$$

where $h_{st}$ represents the commute-times [1] that can be obtained using

$$h_{st} = (z_{tt} - z_{st})/\pi_t \tag{8}$$

It is also worth mentioning that the commute-cost $\bar{c}_s(t)$ from previous section can also be computed using the expected flows from Equation (7) as

$$\bar{c}_s(t) = \sum_{i=1}^{n} \bar{n}_i^{st} \sum_{j=1}^{n} p_{ij} c_{ij} \tag{9}$$

[1] Notice that this fundamental matrix is different from the fundamental matrix of the RSP used in ConScape, and they should not be confounded nor used interchangeably.

From Equation (7), we can see that a flow matrix $\mathbf{N}(t)$ can be obtained, such that $[\mathbf{N}(t)]_{si} = n_i^{st}$. In other words, the matrix $\mathbf{N}(t)$ contains the expected flow for every pair $s$ – $i$ (source node – transient node) for a given target node $t$.

$$\mathbf{N}(t) = \mathbf{Z} - \mathbf{e}\,\mathbf{z}_t^r + \mathbf{h}_t^c \boldsymbol{\pi}^{\mathrm{T}} \tag{10}$$

where $\mathbf{z}_t^r$ represents the row $t$ of the matrix $\mathbf{Z}$, and $\mathbf{h}_t^c$ is the column $t$ from the matrix $\mathbf{H}$ whose element $[\mathbf{H}]_{st} = h_{st}$ is defined from Equation (8). Weighting this matrix $\mathbf{N}(t)$ with the respective source and target quality and summing over all sources provides $\mathbf{bet}^{q-\mathrm{RW}}(t)$, the contribution of target node $t$ to the $q$-betweenness of every transient node $i$:

$$\mathbf{bet}^{q-\mathrm{RW}}(t) = \left(\mathbf{N}(t)^{\mathrm{T}}\mathbf{q}_s\right) q_t^t \tag{11}$$

Thus, the vector of $q$-betweennesses can be obtained by summing the contributions of every target node (see Algorithm 2).

**Algorithm 2** Computation of the $q$-weighted random walk betweenness.

**Input:**
- A directed, strongly connected, graph $G$ containing $n = |\mathcal{V}|$ nodes.
- The $n \times n$ reference transition probabilities matrix $\mathbf{P}$ of $G$.
- The $n \times n$ non-negative cost matrix $\mathbf{C}$ assigned to $G$.
- The $n \times n$ diagonal matrix of source quality $\mathbf{Q}_s$.
- The $n \times n$ diagonal matrix of target quality $\mathbf{Q}_t$.
- The $n \times 1$ stationary distribution $\boldsymbol{\pi}$ from $\mathbf{P}$.

**Output:**
- The $n \times 1$ $q$-weighted betweenness $\mathbf{bet}^{q-\mathrm{RW}}$.

1. $\mathbf{q}_s \leftarrow \mathbf{diag}(\mathbf{Q}_s)$ {Extract the source quality vector}
2. $\mathbf{q}_t \leftarrow \mathbf{diag}(\mathbf{Q}_t)$ {Extract the target quality vector}
3. $\mathbf{bet}^{q-\mathrm{RW}} \leftarrow \mathbf{e}\,0$ {Initialization of the betweenness}
4. $\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{P} + \mathbf{e}\boldsymbol{\pi})$ {Computation of the funamental matrix}
5. $\mathbf{H} \leftarrow (\mathbf{e}\,\mathbf{diag}(\mathbf{Z})^{\mathrm{T}} - \mathbf{Z}) \div \mathbf{e}\boldsymbol{\pi}^{\mathrm{T}}$ {Computation of the commute time matrix}
6. **for** $t \in \mathcal{V}$ **do** {Loop over all target nodes}
7. $\quad \mathbf{N}(t) \leftarrow \mathbf{Z} - \mathbf{e}\,\mathbf{z}_t^r + \mathbf{h}_t^c\boldsymbol{\pi}^{\mathrm{T}}$ {Expected flows for target node $t$}
8. $\quad \mathbf{bet}^{q-\mathrm{RW}}(t) \leftarrow \left(\mathbf{N}(t)^{\mathrm{T}}\mathbf{q}_s\right) q_t^t$ {Contribution of $t$ to the betweennesses}
9. $\quad \mathbf{bet}^{q-\mathrm{RW}} \leftarrow \mathbf{bet}^{q-\mathrm{RW}} + \mathbf{bet}^{q-\mathrm{RW}}(t)$ {Update of the betweennesses}
10. **end for**
11. **return** $\mathbf{bet}^{q-\mathrm{RW}}$

It is also noteworthy that, if one is only interested in the betweenness for one node (or a subset of nodes), the betweenness can also be computed node-wise by computing the matrix $\mathbf{N}(i)$ instead.

$$\mathbf{N}(i) = \left(\mathbf{z}_i^c\mathbf{e}^{\mathrm{T}} - \left(\mathbf{z}_i^c\mathbf{e}^{\mathrm{T}}\right)^{\mathrm{T}}\right) + \pi_i\mathbf{H} \tag{12}$$

We have $[\mathbf{N}(i)]_{st} = \bar{n}_i^{st}$, such that the matrix $\mathbf{N}(i)$ represents the expected flow on node $i$ for every pair source – target. From this matrix, the $q$-betweenness of node $i$ can be obtained as

$$\mathrm{bet}_i^{q-\mathrm{RW}} = \mathbf{q}_s^{\mathrm{T}}\mathbf{N}(i)\mathbf{q}_t \tag{13}$$

### 3.1.2 Edge Betweenness

The $q$-weighted edge betweenness is similar to its node counterpart (see Equation (5)) and is defined as

$$\mathrm{bet}_{ij}^{q-\mathrm{RW}} = \sum_{s,t=1}^{n} q_s^s \bar{n}_{ij}^{st} q_t^t \tag{14}$$

where $\bar{n}_{ij}^{st}$ is the expected flow on edge $(i \to j)$ during a random walk from $s$ to $t$. From this formulation, since we know that $\bar{n}_{ij}^{st} = \bar{n}_i^{st} p_{ij}$, we can easily see that

$$\begin{aligned}
\mathrm{bet}_{ij}^{q-\mathrm{RW}} &= \sum_{s,t=1}^{n} q_s^s \bar{n}_i^{st} p_{ij} q_t^t \\
&= p_{ij} \sum_{s,t=1}^{n} q_s^s \bar{n}_i^{st} q_t^t \\
&= p_{ij}\, \mathrm{bet}_i^{q-\mathrm{RW}}
\end{aligned} \tag{15}$$

The edge $q$-weighted betweenness can thus be easily computed by first computing the betweenness node-wise, then using the reference transition probabilities to obtain the values on edges.

## 3.2 Proximity weighted betweenness

The proximity-weighted (or $k$-weighted) betweenness makes use of a proximity matrix $\mathbf{K}$ whose element $[\mathbf{K}]_{st} = k_{st}$ represents the proximity between nodes $s$ and $t$. Note that, if necessary, the proximities themselves can be re-weighted to take the source/target qualities into account, using $\mathbf{K} \leftarrow \mathbf{Q}^s \mathbf{K} \mathbf{Q}^t$.

The $k$-weighted betweenness of a node is then defined as

$$\mathrm{bet}_i^{k-\mathrm{RW}} = \sum_{s,t=1}^{n} k_{st} \bar{n}_i^{st} \tag{16}$$

It can thus be computed similarly to the $q$-weighted betweenness, by computing the matrix $\mathbf{N}(t)$ from Equation (10) and use it to compute the contribution of target node $t$ to the betweenness vector:

$$\mathbf{bet}^{k-\mathrm{RW}}(t) = \mathbf{N}(t)^{\mathrm{T}} \mathbf{k}_t^c \tag{17}$$

where $\mathbf{k}_t^c$ is the column $t$ from $\mathbf{K}$. The vector of $k$-weighted betweennesses can thus be obtained using Algorithm 3.

It is noteworthy that, similarly to the $q$-weighted betweenness, an approach to compute the betweenness node-wise can also be performed using

$$\mathrm{bet}_i^{k-\mathrm{RW}} = \mathbf{e}^{\mathrm{T}} \left( \mathbf{N}(i) \circ \mathbf{K} \right) \mathbf{e} \tag{18}$$

**Algorithm 3** Computation of the $k$-weighted random walk betweenness.

**Input:**

- A directed, strongly connected, graph $G$ containing $n = |\mathcal{V}|$ nodes.
- The $n \times n$ reference transition probabilities matrix $\mathbf{P}$ of $G$.
- The $n \times n$ non-negative cost matrix $\mathbf{C}$ assigned to $G$.
- The $n \times n$ proximity matrix $\mathbf{K}$.
- The $n \times 1$ stationary distribution $\boldsymbol{\pi}$ from $\mathbf{P}$.

**Output:**

- The $n \times 1$ $q$-weighted betweenness $\mathbf{bet}^{q-\mathrm{RW}}$.

1. $\mathbf{bet}^{k-\mathrm{RW}} \leftarrow \mathbf{e}\,0$ {Initialization of the betweenness}
2. $\mathbf{Z} \leftarrow (\mathbf{I} - \mathbf{P} + \mathbf{e}\boldsymbol{\pi})$ {Computation of the funamental matrix}
3. $\mathbf{H} \leftarrow (\mathbf{e}\,\mathbf{diag}(\mathbf{Z})^{\mathrm{T}} - \mathbf{Z}) \div \mathbf{e}\boldsymbol{\pi}^{\mathrm{T}}$ {Computation of the commute time matrix}
4. **for** $t \in \mathcal{V}$ **do** {Loop over all target nodes}
5. $\quad \mathbf{N}(t) \leftarrow \mathbf{Z} - \mathbf{e}\,\mathbf{z}_t^r + \mathbf{h}_t^c \boldsymbol{\pi}^{\mathrm{T}}$ {Expected flows for target node $t$}
6. $\quad \mathbf{bet}^{k-\mathrm{RW}}(t) \leftarrow \mathbf{N}(t)^{\mathrm{T}} \mathbf{k}_t^c$ {Contribution of $t$ to the betweennesses}
7. $\quad \mathbf{bet}^{k-\mathrm{RW}} \leftarrow \mathbf{bet}^{k-\mathrm{RW}} + \mathbf{bet}^{k-\mathrm{RW}}(t)$ {Update of the betweennesses}
8. **end for**
9. **return** $\mathbf{bet}^{q-\mathrm{RW}}$

where $\mathbf{N}(i)$ is the expected flow in node $i$ for every pair source-target computed from Equation (12).

Note that, similarly like the $q$-weighted version, the $k$-weighted edge betweenness can be simply obtained from the node betweenness as

$$\mathrm{bet}_{ij}^{k-\mathrm{RW}} = p_{ij} \mathrm{bet}_i^{k-\mathrm{RW}} \tag{19}$$

# References


[1] D. Boley, G. Ranjan, and Z.-L. Zhang. Commute times for a directed graph using an asymmetric laplacian. *Linear Algebra and its Applications*, 435(2):224–242, 2011.

[2] F. Fouss, M. Saerens, and M. Shimbo. *Algorithms and models for network data and link analysis*. Cambridge University Press, 2016.

[3] I. Kivimäki. *Distances, centralities and model estimation methods based on randomized shortest paths for network data analysis*. PhD thesis, Université catholique de Louvain, 2018.

[4] I. Kivimäki, M. Shimbo, and M. Saerens. Developments in the theory of randomized shortest paths with a comparison of graph node distances. *Physica A: Statistical Mechanics and its Applications*, 393:600–616, 2014.

[5] B. Van Moorter, I. Kivimäki, A. Noack, R. Devooght, M. Panzacchi, K. R. Hall, P. Leleux, and M. Saerens. Accelerating advances in landscape connectivity modelling with the conscape library. *Methods in Ecology and Evolution*, 2022.

[6] L. Yen, A. Mantrach, M. Shimbo, and M. Saerens. A family of dissimilarity measures between nodes generalizing both the shortest-path and the

commute-time distances. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, pages 785–793, 2008.