Measures of interpersonal network (I x I) ---- authority structure
- ***Size: the number of people in the organization. (INTEGER) [getnsizeval() in netaccess.c]
- Levels: the number of levels. [levels() in netdescriptive.c]
- Span of control: average # of lower links per manager. [use degree() in netdescriptive.c]
- Isolates: the number of people with no links. [use degree() in netdescriptive.c]
- ***Density: the number of actual links in a network divided by the number of all possible links in the network (DOUBLE)\(^1\) [density() in netdescriptive.c]

\[
\text{Density} = \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij}, \quad \alpha_{ij} \text{ is a directed link from } i \text{ to } j
\]

- ***Conductivity: the product by multiplying the number of people linked directly into the focal person and the number of people linked directly out from it. [use degree() in netdescriptive.c]

\[
\text{Conductivity} = \sum_{i=1}^{N} \left( \left( \sum_{j=1 \text{and } j \neq i}^{N} \alpha_{ji} \right) \times \left( \sum_{j=1 \text{and } j \neq i}^{N} \alpha_{ij} \right) \right)
\]

- Group degree centralization: Freeman’s formula for graph degree centralization. \(C_D(n_i)\) is the actor degree index for node \(n_i\). \(C_D(n^*)\) is the largest actor degree index. \(\alpha_{ij}\) is the direct or adjacent link between actor \(i\) and actor \(j\). [degreecentralization() in netdescriptive.c]

\[
C_D = \frac{\sum_{i=1}^{N} (C_D(n^*) - C_D(n_i))}{(N - 1)(N - 2)}, \quad \text{where } C_D(n_i) = \sum_j \alpha_{ij}
\]

- Group closeness centralization: Freeman’s formula for graph closeness centralization. \(C_C(n_i)\) is the standardized actor closeness index for node \(n_i\). \(C_C(n^*)\) is the largest

\(^1\) Mainly based on PCANNS model, the relationships between personnel, knowledge, resources, and tasks are represented by the following symbols: \(\alpha_{ij}(PxP), K_{ik}(PxK), C_{ir}(PxR), A_{it}(PxD), R_{kt}(KxT), N_{rt}(RxT), S_{rt}(RxR), \) and \(P_{tt}(TxT)\).
standardized actor closeness index. \(d(n_i, n_j)\) is the number of lines in the geodesic linking actor \(i\) and actor \(j\). \([\text{closenesscentralization()}\] in netdescriptive.c)

\[
C_c = \sum_{i=1}^{N} \frac{(C'_c(n^*) - C'_c(n_i))}{(N-1)(N-2)/(2(N-3))}, \quad \text{where } C'_c(n_i) = (N-1)(\sum_{j=1}^{N} d(n_i, n_j))^{-1}
\]

- Group betweenness centralization: Freeman’s formula for graph betweenness centralization. \(C_B(n_i)\) is the actor betweenness index for node \(n_i\). \(C_B(n^*)\) is the largest actor betweenness index. \(\beta_{jk}(n_i)\) is the number of geodesics linking two actors that contain actor \(i\). \([\text{betweennesscentralization()}\] in netdescriptive.c)

\[
C_B = \sum_{i=1}^{N} \frac{(C_B(n^*) - C_B(n_i))}{(N-1)^2(N-2)/2}, \quad \text{where } C_B(n_i) = \sum_{j<k} (\beta_{jk}(n_i)/\beta_{jk})
\]

- K’s graph connectivity: \(G_c = 1 - \frac{V}{\text{Max}V}\), where \(V\) is the number of pairs of points that are not mutually reachable, and MaxV is the maximum number of violations of the total number of pairs of points. The maximum number of violations is the total number of pairs of points, say MaxV = \(N (N-1) / 2\). \([\text{connectedness()}\] in netdescriptive.c)

- K’s graph hierarchy: \(G_h = 1 - \frac{V}{\text{Max}V}\), where \(V\) is the number of unordered pairs of points that are symmetrically linked, and MaxV is the number of unordered pairs of points where \(P_i\) is linked to \(P_j\) or \(P_j\) is linked to \(P_i\). \([\text{hierarchy()}\] in netdescriptive.c)

- K’s graph efficiency: \(G_e = 1 - \frac{V}{\text{Max}V}\), where \(V\) is the number of links in excess of \(N_n - 1\), summed over all components, and MaxV is the maximum number of links in excess of \(N_n - 1\) possible, summed over all components. \([\text{efficiency()}\] in netdescriptive.c)

- K’s graph least upper boundedness: \(LUB = 1 - \frac{V}{\text{Max}V}\), where \(V\) is the number of pairs of points that have no LUB in each component summed across all components, and MaxV is the maximum number of pairs of points that could possibly have no LUB. \([\text{lubness()}\] in netdescriptive.c)

Measures of Commitment network (IxR)

- Cognitive load: defined for each person, which is equal to \((1+2+3+4+5+6)/6\)
  1. \# of people person \(i\) interacts with / total \# of people in the group;
  2. \# of resources person \(i\) manages / total \# of resources
  3. \# of tasks person \(i\) is assigned to / total \# of tasks
  4. sum of \# resources required by the tasks person \(i\) does / (total \# of tasks * total \# of resources)
  5. sum of \# people who do the same tasks person \(i\) does / (total \# of tasks * total \# of people)
6. sum of negotiation needs person i needs to do for each task / total possible negotiations

\[ CL_i = \left( \frac{\sum_{j=1}^{N} \alpha_j + \sum_{r=1}^{R} C_{ir} + \sum_{t=1}^{T} A_{it} \sum_{r=1}^{R} N_{ir} \wedge N_{ir} \wedge N_{ir} \pmod{N} + \sum_{i=1}^{N} \sum_{r=1}^{R} A_{ir} \wedge A_{ir} + \sum_{r=1}^{R} \sum_{i=1}^{N} (C_{ir} - A_{ir} \wedge N_{ir})}{R \times T \times N \times T} \right) / 6 \]

- **Resource load:** average number of resources each individual has access to. [use edgecount() in netdescriptive.c]

\[ R_{load} = \frac{\sum_{i=1}^{N} \sum_{r=1}^{R} C_{ir}}{N} \]

- **Resource specialization:** the number of resources known by only one person divided by the number of resources in the group. [use degree() in netdescriptive.c]

\[ R_{Specialization} = \frac{\sum_{r=1}^{R} (\sum_{i=1}^{N} C_{ir} = 1)}{R} \]

- **Skill complexity:** the product of the # of people and the # of resources. [use getnsizeval() in netaccess.c]

- **Consensus:** for each resource, count the number of people who are in the majority, sum the counts across all the resources, and divide the sum by the number of resources and the number of people (this measure should arrange between 0.5 and 1, the higher it is, the higher the consensus is).

\[ Consensus = \frac{\sum_{r=1}^{R} ((majority == 1) \times (\sum_{i=1}^{N} C_{ir} \wedge N = \sum_{i=1}^{N} C_{ir}))}{R \times N} \]

- **Access redundancy:** average number of people who have access to each resource, in excess of 1. (if greater than 1, there is redundancy) [redundancy() in netorg.c]

\[ Access \ redundancy = \frac{\sum_{r=1}^{R} \left( \sum_{i=1}^{N} C_{ir} - 1 \right)}{N} \]

Measures of Assignment network (IxT)

- **Assigned workload:** the average # of tasks each individual is assigned. [use edgecount() in netdescriptive.c]
\[ A_{\text{Workload}} = \frac{\sum_{i=1}^{N} \sum_{r=1}^{T} A_{ir}}{N} \]

- **Task specialization**: the number of tasks assigned to only one person divided by the number of tasks in the group. [use degree() in netdescriptive.c]

\[ T_{\text{Specialization}} = \frac{\sum_{i=1}^{T} (\sum_{r=1}^{N} A_{ir} = 1)}{T} \]

- **Assignment complexity**: the product of the # of people and the # of tasks. [use getnsizeval() in netaccess.c]

- **Consensus**

\[ \text{Consensus} = \frac{\sum_{i=1}^{T} ((\text{majority} = 1)? \sum_{r=1}^{N} A_{ir} : (N - \sum_{r=1}^{N} A_{ir}))}{T \times N} \]

- **Assignment redundancy**: the average # of people who are assigned to a task in excess of 1 (If > 1, there is redundancy) [redundancy() in netorg.c]

\[ \text{Assignment redundancy} = \frac{\sum_{i=1}^{T} \left( \sum_{r=1}^{N} A_{ir} - 1 \right)}{N} \]

**Measures of Substitute network (RxR)**

- **Size / resource complexity**: the number of resources in the organization. [getnsizeval() in netaccess.c]

- **Density**: the existing links between resources in the network divided by the number of all possible links (DOUBLE) [density() in netdescriptive.c]

- **Conductivity**: the product by multiplying the number of resources linked directly into the focal resource and the number of resources linked directly out from it. [use degree() in netdescriptive.c]

**Measures of Needs network (RxT)**

- **Need for negotiation**: the number of tasks which need resources / the number of tasks (DOUBLE)

\[ \text{Need for negotiation} = \sum_{i=1}^{T} \left( \sum_{r=1}^{R} ((N_{r,}'-C_{r}) = 1?1:0) \right) \]

- **Under supply**: the average number of needed resources per task.
\[ Under \_ s = \sum_{r=1}^{R} \left( \sum_{t=1}^{T} \left( (N_{rt} - \sum_{r=1}^{R} C_{tr} \land C_{tr}) \right) \right) \]

- **Resource redundancy**: the average number of resources per task in excess of 1. [redundancy() in netorg.c]

\[ R_{\_ redundancy} = \frac{\sum_{r=1}^{R} \left( \sum_{t=1}^{T} N_{rt} - 1 \right)}{T} \]

Measures of Precedence network (TxT)
- **Size / complexity**: the # tasks in the organization. [getnsizeval() in netaccess.c]
- **Density**: the existing links between tasks in the network divided by the number of all possible links (DOUBLE) [density() in netdescriptive.c]
- **Conductivity**: the product by multiplying the number of tasks linked directly into the focal task and the number of tasks linked directly out from it. [use degree() in netdescriptive.c]

Multiple Network Measures
- "Actual" (Predicted) Workload: for actor i, the total number of relevant skills i could contribute to the performance of tasks to which he or she is assigned. [actualworkload() in netorg.c]

\[ Act_\_ Workload(i) = \left| KRA^T \right|_{ii} \]

- Communication Congruence: one minus the normalized distance between the precedence network and the network formed by the union of paths in the task->people->people->task network. [communicationcongruence() in netorg.c]

\[ Com_\_ Cong = 1 - \frac{\sum_{i=1}^{T} \sum_{j=1}^{T} \left| P_{ij} - \left[ A^T \otimes \alpha \otimes A \right]_{ij} \right|}{T^2} \]

- Potential Workload: for actor i, the total number of relevant skills i could contribute to all organizational tasks. [potentialworkload() in netorg.c]

\[ Pot_\_ Workload(i) = \sum_{t=1}^{T} \left| K_{t} \right|_{i} \]

- Resource Congruence: one minus the normalized distance between the needs network and the network formed by the union of paths in the resource->people->task network. [resourcecongruence() in netorg.c]

\[ Re_\_ Cong = 1 - \frac{\sum_{r=1}^{R} \sum_{t=1}^{T} \left| N_{rt} - \left[ C^T \otimes A \right]_{rt} \right|}{RT} \]

- Resource Usage: the raw difference between the needs network and the product of the transposed commitment and assignment networks. [resourceusage() in netorg.c]

\[ Re_\_ Usage = N - C^T A \]
- Skills Usage: the raw difference between the requirements network and the product of the transposed knowledge and assignment networks. \(\text{[skillsusage()}\) in netorg.c\]
- \(\text{Ski}_\text{Usage} = R - K^T A\)