On community structure validation in real networks

Luisa Cutillo[1]
Joint work with Mirko Signorelli[2]

[1]School of Mathematics, Department of Statistics, University of Leeds (UK)

[2]Mathematical Institute, Leiden University (NL)

UNIVERSITY OF LEEDS

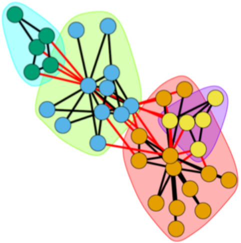
Aim of this talk

We propose a set of indices for community structure validation of network partitions.

The **R code** is available at github.com/mirkosignorelli/csv
Paper: Signorelli, M., Cutillo, L. On community structure validation in real networks. Comput Stat (2021).
https://doi.org/10.1007/s00180-021-01156-6

What is a community structure?

The study of the structure of a graph is often achieved by decomposing it into its constituent modules or communities [Girvan (2002), Girvan and Newman (2004)].



Community Structure Summary

Nodes within a network are connected together in tightly joined groups, while between those groups connections are looser.

How to measure community structure?

Modularity is the fraction of edges falling within the given groups minus the expected fraction, if edges were distributed at random.

Modularity Q

- Q measures the strength of division of a network into modules
- Q only relies on the distributions of nodes
- $Q \in [-1/2, 1]$

How can communities be identified?

Challenges

- The communities that constitute a network are usually unknown
- A network may not have any property of community structure
- Once the communities have been estimated, the analyst is then left with questions on the **adequacy** of the retrieved clusters

When is a Network partition *meaningful*?

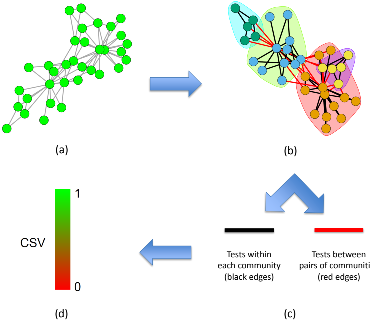
An association between some features and the clusters may be taken as a confirmation of the goodness of the clusters.

Problems with this practice

- A community structure could not be related to any observed feature of the nodes
- It does not take into account network topology
- Features information might not be available

Our Approach

- Focus on the distribution of **links between nodes** in the different clusters.
- Based on a significance testing procedure for the number of links that are observed between and within the communities
- Tests results are combined into a community structure validation (CSV) index


(a)
(b)
Tests within each community (black edges)
Tests between pairs of communities (red edges)
(c)
CSV
1
0
(d)

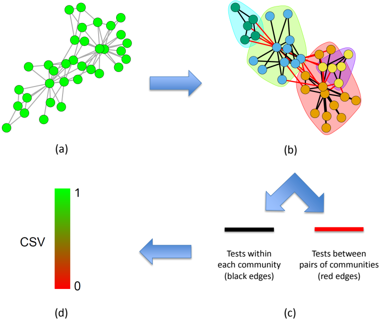
Our Approach

a Graph of interest

b Define a partition of the nodes into q clusters

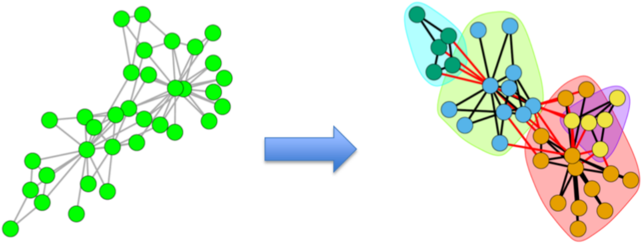
c Perform q tests of enrichment within each community and $q(q-1)/2$ tests of enrichment between each pair of different communities.

d Combine the results in the Community Structure Validation Index (CSV)


(a)
(b)
Tests within each community (black edges)
Tests between pairs of communities (red edges)
(c)
CSV
1
0
(d)

Inferential Procedure: settings

- We denote by $\mathcal{P}_V = \{C_1, ..., C_q\}$ a partition of V into q disjoint sets, such that $C_r \cap C_s = \emptyset$ if $r \neq s$ and $\cup_{r=1}^{q} C_r = V$.



Does $\mathcal{P}_V$ induce a community structure in $\mathcal{G}$?

- We implement a one-tailed adaptation of NEAT, the Network Enrichment Analysis Test proposed by Signorelli et al. (2016).

NEAT hypergeometric null model

Let n_{AB} be the observed number of edges between [arrows from] nodes $\in A$ to nodes $\in B$ (directed [undirected] $\mathcal{G}$);

undirected networks

$$N_{AB} \sim \text{hypergeom}\left(n = d_A, K = d_B, N = d_V\right), \tag{1}$$

where d_A, d_B and d_V denote the total degrees of sets A, B and V.

directed networks

$$N_{AB} \sim \text{hypergeom}\left(n = o_A, K = i_B, N = i_V\right), \tag{2}$$

where o_A denotes the outdegree of A and i_B and i_V are the indegrees of B and V.

NEAT enrichment testing

Original implementation: two-tailed alternative

- expected number of edges (arrows) between A and B, $\mu_{AB} = E(N_{AB})$
- expected number of links $\mu^0_{AB} = E(N_{AB}|H_0) = nK/N$
- H_0: $\mu_{AB} = \mu^0_{AB}$ (no enrichment)
- H_1: $\mu_{AB} \neq \mu^0_{AB}$ (enrichment)

Our implementation: one-tailed alternative

- H_1: $\mu_{AB} > \mu_0$ **Over**enrichment
- H_1: $\mu_{AB} < \mu_0$ **Under**enrichment

Our Testing Strategy

We assess the extent to which $\mathcal{P}_V$ generates a community structure by testing

1. overenrichment within each community C_r, $r \in \{1, ..., q\}$:

$$H_0: \ \mu_{rr} = \mu^0_{rr}, \ \ H_1: \mu_{rr} > \mu^0_{rr}, \tag{3}$$

2. underenrichment between each pair of communities (C_r, C_s), with $r < s \in \{1, ..., q\}$ if $\mathcal{G}$ is undirected or $r \neq s$ if it is directed:

$$H_0: \ \mu_{rs} = \mu^0_{rs}, \ \ H_1: \mu_{rs} < \mu^0_{rs}, \tag{4}$$

3. Apply multiple testing correction procedure proposed by Heyse, 2011 for discrete test statistics and derive the adjusted pvalues $\tilde{p}_{rr}$ and $\tilde{p}_{rs}$.

Community Structure Validation Index (CSV)

CSV

- Undirected graphs:

$$CSV_U = \frac{\sum_{r=1}^{q} I(\tilde{p}_{rr} \leq \alpha) + \sum_{r>s} I(\tilde{p}_{rs} \leq \alpha)}{q(q+1)/2}$$

- Directed graphs:

$$CSV_D = \frac{\sum_{r=1}^{q} I(\tilde{p}_{rr} \leq \alpha) + \sum_{r\neq s} I(\tilde{p}_{rs} \leq \alpha)}{q^2}$$

- CSV_U and $CSV_D \in [0,1]$
- the higher values of CSV the stronger evidence that a graph partition induces a community structure.

Weighted CSV

Consider a weighted version of CSV, where we weight each rejection $I(\tilde{p}_{rs} \leq \alpha)$ by $\frac{\alpha - \tilde{p}_{rs}}{\alpha} \in [0,1]$.

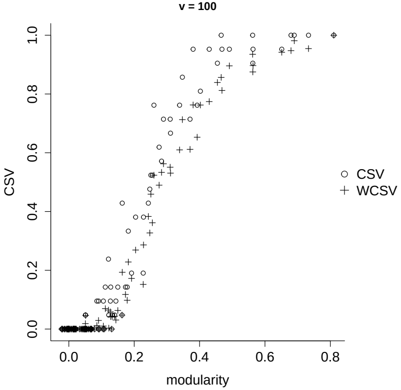
Undirected graph

$$WCSV_U = \frac{\sum_{r=1}^{q} I(\tilde{p}_{rr} \leq \alpha)\frac{\alpha - \tilde{p}_{rr}}{\alpha} + \sum_{r>s} I(\tilde{p}_{rs} \leq \alpha)\frac{\alpha - \tilde{p}_{rs}}{\alpha}}{q(q+1)/2}.$$

- $WCSV \in [0,1]$
- $WCSV \leq CSV$
- CSV and WCSV differ for small graphs, and tend to achieve the same value for larger graphs

Robustness with respect to modularity

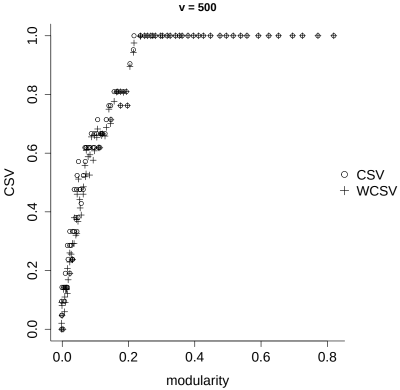
1. Generate graphs from stochastic blockm. with increasing Q.
2. Test network enrichment between **true** communities.
3. Compute UCSV and WCSV (ideally $= 1$).



1. CSV lower in small networks ($v = 100$)
2. CSV higher in larger networks ($v \in \{500, 1000, 5000\}$) if $\mathbf{Q \geq 0.2}$
3. $Q < 0.2$ weak community structure

Robustness with respect to modularity

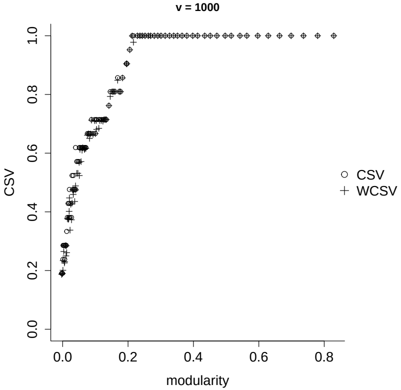
1. Generate graphs from stochastic blockm. with increasing Q.
2. Test network enrichment between **true** communities.
3. Compute UCSV and WCSV (ideally $= 1$).



1. CSV lower in small networks ($v = 100$)
2. CSV higher in larger networks ($v \in \{500, 1000, 5000\}$) if $\mathbf{Q \geq 0.2}$
3. $Q < 0.2$ weak community structure

Robustness with respect to modularity

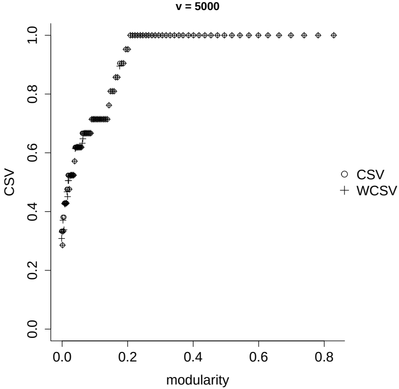
1. Generate graphs from stochastic blockm. with increasing Q.
2. Test network enrichment between **true** communities.
3. Compute UCSV and WCSV (ideally $= 1$).



1. CSV lower in small networks ($v = 100$)
2. CSV higher in larger networks ($v \in \{500, 1000, 5000\}$) if $\mathbf{Q \geq 0.2}$
3. $Q < 0.2$ weak community structure

Robustness with respect to modularity

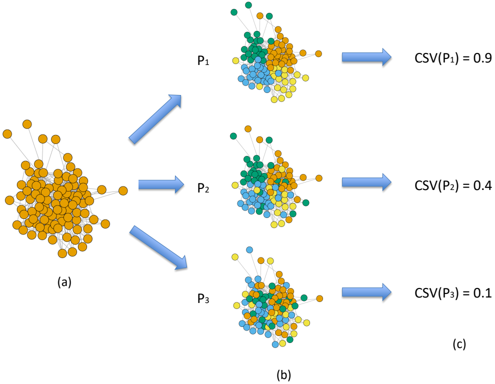
1. Generate graphs from stochastic blockm. with increasing Q.
2. Test network enrichment between **true** communities.
3. Compute UCSV and WCSV (ideally $= 1$).



1. CSV lower in small networks ($v = 100$)
2. CSV higher in larger networks ($v \in \{500, 1000, 5000\}$) if $\mathbf{Q \geq 0.2}$
3. $Q < 0.2$ weak community structure

Application: comparing different partitions

a. Network with 100 nodes and 4 communities from a degree-corrected stochastic blockmodel.
b. Partition induced by true communities (top) and two partitions where 20% (centre) and 40% (bottom) of the nodes are assigned to the wrong cluster.
c. Computed *CSV*



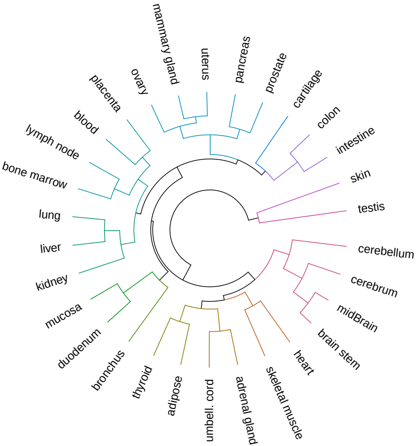
Application: comparing different networks

- choose a community detection method and apply it to $\mathcal{G}_1$ so as to derive its partition in q communities $\mathcal{P}_1 = \{C_{11}, ..., C_{1q}\}$. Similarly, obtain $\mathcal{P}_2$ from $\mathcal{G}_2$;
- compute the community structure validation indices of $\mathcal{P}_1$ in $\mathcal{G}_1$ and in $\mathcal{G}_2$, and of $\mathcal{P}_2$ in $\mathcal{G}_1$ and in $\mathcal{G}_2$;
- compute the relative indices

$$R_{CSV}\left(\mathcal{P}_i|\mathcal{G}_j\right) = \frac{CSV(\mathcal{P}_i|\mathcal{G}_j)}{CSV(\mathcal{P}_i|\mathcal{G}_i)},\ i \neq j \in \{1, 2\},$$

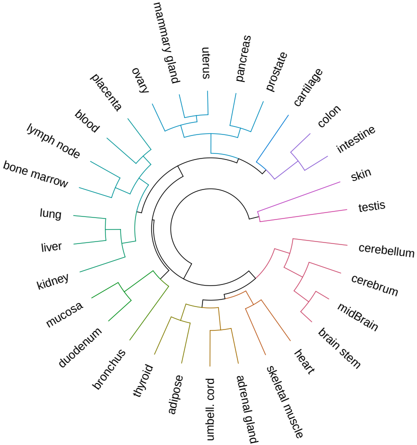
Case Study: Tissue comparison

- 30 tissue-specific human gene co-regulation networks were reverse engineered. We refer to the original paper, Gambardella et al. (2013), for details.
- We find 13 clusters, highlighted in different colours.
- For each $\mathcal{G}_i$ $\forall i, \in \{1, \dots, 30\}$, we use Louvain community extraction method to obtain a partition $\mathcal{P}_i$.



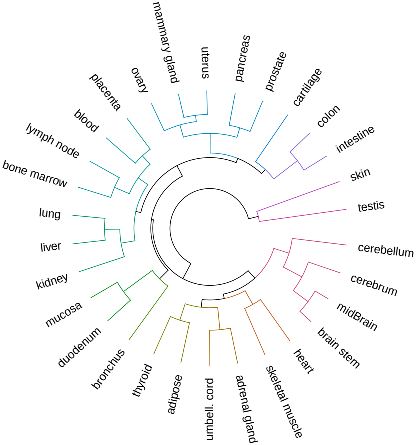
Case Study: Tissue comparison (2)

- We compute the relative indices $R_{i,j} = R_{UCSV}(\mathcal{P}_i|\mathcal{G}_j)$ a $\forall i, j \in \{1, \ldots, 30\}, i \neq j$.
- We build a similarity matrix $S = \left(R + R^T\right)/2$, and distance matrix $D = 1 - S$.



Case Study: Tissue comparison (3)

- All cerebral tissues co-clustered (cerebrum, cerebellum, mid brain and brain stem)
- The only two striated muscles (heart and skeletal muscle) co-clustered.
- Female reproductive organs (mammary gland, uterus and ovary) are linked together.
- Colon and intestine form together a unique cluster.



Comparison of network clustering algorithms

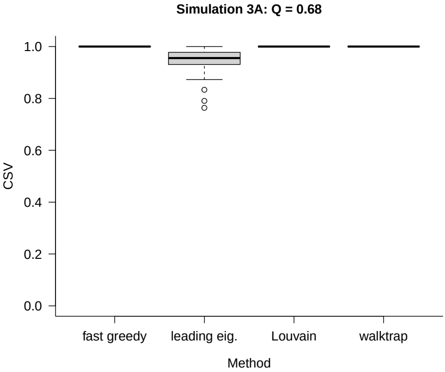
Simulation Settings

Generated graphs with $v = 1000$, $q = 8$, from stochastic blockmodels with increasing Q. 100 random graphs for each modularity level.

We apply to each of the graphs the following clustering algorithms:

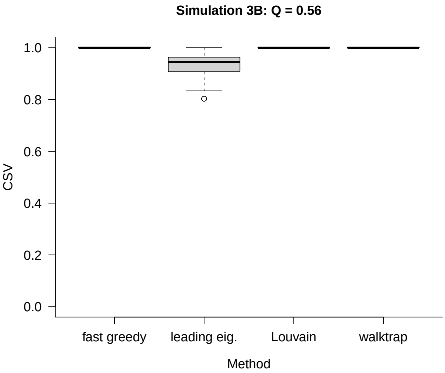
1. fast greedy, proposed by Clauset (2004)
2. leading eigenvalue, proposed by Newman (2006)
3. Louvain, proposed by Blondel (2008)
4. Walktrap, proposed by Pons (2005)

Comparison of network clustering algorithms



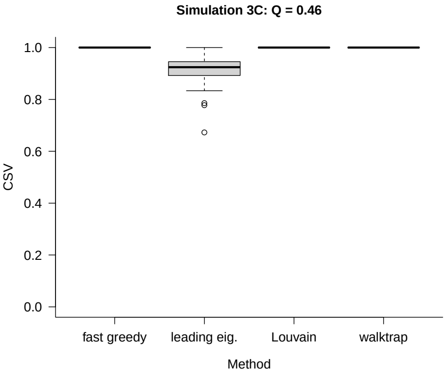
- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Comparison of network clustering algorithms



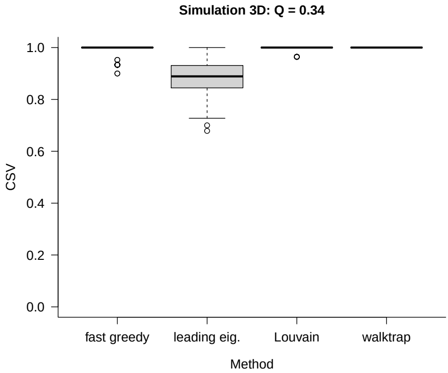
- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Comparison of network clustering algorithms



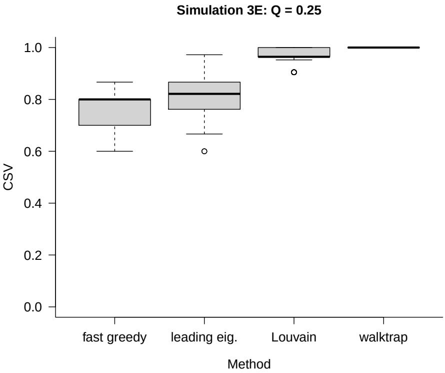
- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Comparison of network clustering algorithms



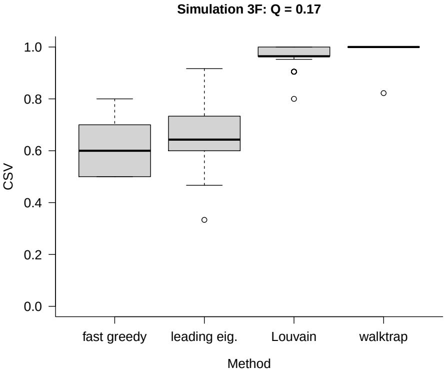
- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Comparison of network clustering algorithms



- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Comparison of network clustering algorithms



- Leading eigenvalue performs always poorly.
- Fast greedy works only with high Q.
- Walktrap detects modular structure also with low modularity.

Summary

We develop an inferential procedure to check if a partition of nodes is a valid community structure for a network.

This approach can be used to

- compare different partitions of the same graph;
- compare two networks by
    1. evaluating if $\mathcal{P}_1 \equiv \mathcal{P}_2$;
    2. detecting the most preserved modules.
- compare the performance of different clustering methods.

Acknowledgement

This collaboration is supported by the COST Action CA15109 - **European Cooperation for Statistics and Network Data Science (COSTNET)** and **Marie S. Curie IF under HORIZON 2020**

cost
EUROPEAN COOPERATION
IN SCIENCE AND TECHNOLOGY

Information and Communication
Technologies (ICT)

MARIE CURIE ACTIONS

References

Signorelli, M. and Vinciotti, V. and Wit, E.(2016) NEAT: an efficient Network Enrichment Analysis Test *BMC Bioinformatics* 352(17), 1:17.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*. Series B. 289:300.

Heyse, J. F. (2011). In Recent advances in biostatistics: False discovery rates, survival analysis, and related topics. *World Scientific*, 43–58.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1).

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3).

Pons, P. and Latapy, M. (2005). Computing communities in large networks using random walks. *International Symposium on Computer and Information Sciences*, 284:293. Springer.

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397). 8:19.

Wright, S. P. (1992). Adjusted p-values for simultaneous inference. *Biometrics* 1005:1013.

midp-values

1. Apply clustering algorithm to $\mathcal{G}_1$ to obtain $\mathcal{P}_1^* = \{C_1^*, C_2^*, ..., C_q^*\}$
2. Test network enrichment within and between C_r^* in $\mathcal{G}_2$
   - **overenrichment within each community** C_r:

     $$H_0 : \ \mu_{rr} = \mu_{rr}^0, \ \ H_1 : \mu_{rr} > \mu_{rr}^0,$$

     mid-p-value: $p_{rr} = \frac{1}{2} P\left(N_{rr} = n_{rr}\right) + P\left(N_{rr} > n_{rr}\right).$
   - **underenrichment between each pair of communities** (C_r, C_s):

     $$H_0 : \ \mu_{rs} = \mu_{rs}^0, \ \ H_1 : \mu_{rs} < \mu_{rs}^0,$$

     mid-p-value: $p_{rs} = \frac{1}{2} P\left(N_{rs} = n_{rs}\right) + P\left(N_{rs} < n_{rs}\right)$

Network generating process

We employ a **degree-corrected stochastic blockmodel** for undirected graphs [Karren and Newman, 2011]:

- flexible model to generate networks with a given community structure and degree distribution;
- model parameters (p_{IN} and p_{OUT}) can be modified to change modularity Q.

$$y_{ij} | i \in C_i,\, j \in C_j \sim Bern(\pi_{ij}), \text{ where}$$

$$\pi_{ij} = min(w_i w_j \theta_{C_i\, C_j}, 1),$$

$$\theta_{C_i\, C_j} \in [0,1] \text{ and } \sum_i w_i I(C_i = C_r) = n_r \;\; \forall C_i, C_j.$$

P-value

For a discrete test statistic T and $H_1 : \theta \neq \theta_1$:

- $p_1 = 2min\left[P_0(T \geq t), P_0(T \leq t)\right]$ can exceed 1;
- naive adjustment: $p_2 = min(p_1, 1) \in [0, 1]$;

We compute the p-value using

$$p = 2\min\left[P_0(T > t), P_0(T < t)\right] + P_0(T = t) =$$

$$2\min\left[P(N_{AB} > n_{AB}|H_0), P(N_{AB} < n_{AB}|H_0)\right] + P(N_{AB} = n_{AB}|H_0).$$

Hypergeometric: notation

$$Pr\left(X = x\right) = \frac{\binom{K}{x}\binom{N-K}{n-x}}{\binom{N}{n}},\ max(0, n + K - N) \leq x \leq min(n, K)$$

Interpretation of the parameters:

- x is the number of observed successes;
- K the maximum number of possible successes;
- n is the number of draws;
- N is the population size, from which **we sample without replacement**.