Community quality metrics for disjoint community structure

Please refer to the main function in **CommunityQuality.java** as an example about how to run the code to calculate community quality metrics for disjoint community structure.

There are totally four functions to calculate metrics for disjoint communities:
(1) public static double[] computeQualityWithoutGroundTruth(String networkFile, boolean isUnweighted, boolean isUndirected, String communityFile);
----To calculate the community quality metrics without ground truth. To know more about those metrics, please refer to the three papers below.
Mingming Chen, Tommy Nguyen, and Boleslaw K. Szymanski, "**A New Metric for Quality of Network Community Structure**", ASE Human Journal, vol. 2, no. 4, Sep. 2013, pp. 226-240.

Mingming Chen, Tommy Nguyen, and Boleslaw K. Szymanski, "**On Measuring the Quality of a Network Community Structure**", The ASE/IEEE International Conference on Social Computing (SocialCom), Washington D.C., Sep. 2013, pp. 122-127.

Lancichinetti, A., Fortunato, S., Kert´esz, J.: **Detecting the overlapping and hierarchical community structure in complex networks**. New Journal of Physics 11(3), 033,015 (2009)

Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen, "**Quantitative function for community detection**," Physical Review E, vol. 77, no. 3, p. 036109, 2008.

@Parameters:
networkFile: the file path of the network dataset;
isUnweighted: whether the network is unweighted or not;
isUndirected: whether the network is undirected or not;
communityFile: the file path of the communities detected by community detection algorithms.

The following three functions calculate the community quality metrics based on ground truth communities. To know more about those metrics, please refer to Section 4.1 in the paper below:
Mingming Chen, Konstantin Kuzmin, and Boleslaw Szymanski, "**Community Detection via Maximization of Modularity and Its Variants**", IEEE transactions on Computational Social Systems 1(1), Mar. 2014, pp. 46-65.

(2) public static double[] computeInformationEntropyMetrics(String disCommunityFile, String realCommunityFile);
----To calculate information theory based metrics;
@Parameters:
disCommunityFile: the file path of the communities detected by community detection algorithms;
realCommunityFile: the file path of the ground truth communities.

(3) public static double[] computeClusterMatchingMetrics(String disCommunityFile, String realCommunityFile);
----To calculate clustering matching based metrics;
@Parameters: the same with the second function.

(4) public static double[] computeIndexMetrics(String disCommunityFile, String realCommunityFile)
----To calculate pair counting based metrics.
@Parameters: the same with the second function.

Format of network:
src_node_id dst_node_id weight (separated with space or tab, weight is not necessary for unweighted network)

Format of communities:
Each line corresponds to a community, nodes in each line is separated with space.

Community quality metrics for overlapping community structure

Please refer to the main function in **OverlappingCommunityQuality.java** as an example about how to run the code to calculate community quality metrics for overlapping community structure.

There are totally six functions to calculate metrics for overlapping communities:
(1) public static double[] computeOvQualityWithoutGroundTruth(String networkFile, boolean isUnweighted, boolean isUndirected, String communityFile, int belongingVersion, int belongingFunVersion)
(2) public static double computeQovLink(String networkFile, boolean isUnweighted, boolean isUndirected, String communityFile, int belongingVersion, int belongingFunVersion) {
----To calculate the community quality metrics without ground truth for fuzzy overlapping community structure. To know more about those metrics, please refer to the two papers below.
Mingming Chen and Boleslaw K. Szymanski, "**Fuzzy Overlapping Community Quality Metrics**", Social Network Analysis and Mining 5:40, July 2015.

Mingming Chen, Konstantin Kuzmin, and Boleslaw K. Szymanski, "**Extension of Modularity Density for Overlapping Community Structure**", IEEE/ACM ASONAMWorkshop on Social Network Analysis in Applications (SNAA), Beijing, China, Aug. 2014, pp. 856-863.

@Parameters:
networkFile: the file path of the network dataset;
isUnweighted: whether the network is unweighted or not;
isUndirected: whether the network is undirected or not;
communityFile: the file path of the communities detected by community detection algorithms;
belongingVersion: the version of belonging coefficient adopted;
belongingFunVersion: the version of belonging function adopted.

The following four functions calculate the community quality metrics for overlapping communities based on the ground truth communities. To know more about those metrics, please refer to the two papers below:
J. Xie, S. Kelley and B. K. Szymanski, "**Overlapping Community Detection in Networks: the State of the Art and Comparative Study**", ACM Computing Surveys, 2013.

Wu H, Gao L, Dong J, Yang X (2014) **Detecting Overlapping Protein Complexes by Rough-Fuzzy Clustering in Protein-Protein Interaction Networks**. PLoS One 9: e91856.

(3) public static double computeNMI(String disCommunityFile, String realCommunityFile)
----To calculate Normalized Mutual Information (NMI) for overlapping communities;
@Parameters:
disCommunityFile: the file path of the communities detected by community detection algorithms;

realCommunityFile: the file path of the ground truth communities.

(4) public static double computeOmegaIndex(String disCommunityFile, String realCommunityFile)
----To calculate Omega Index for overlapping communities;
@Parameters: the same with the third function.

(5) public static double[] computeFscore(String disCommunityFile, String realCommunityFile)
----To calculate F-score for overlapping communities;
@Parameters: the same with the third function.

(6) public static double[] computeF1(String disCommunityFile, String realCommunityFile)
----To calculate F1 for overlapping communities;
@Parameters: the same with the third function.

Format of network:
src_node_id   dst_node_id   weight (separated with space or tab, weight is not necessary for unweighted network)

Format of communities:
Each line corresponds to a community, nodes in each line is separated with space.

Note: please be noted that OverlappingCommunityQuality.java uses some functions from CommunityQuality.java. Thus, please include CommunityQuality.java in your directory too when trying to use OverlappingCommunityQuality.java.