RANG Network Generator Manual

RANG generator background information

- Uses python3.
- There are two versions of the RANG generator available.
    - auto: Can be used when the user has a social network present that can be provided to our generator.
    - manual: Can be used when the user does NOT have access to a social network, and will instead manually provide statistical data about their social network to our RANG generator.
- There are two different generation models available.
    - BWRN model.
    - WRG model.
    - Both models are discussed in the following paper:
        - "A Synthetic Network Generator for Covert Network Analytics"
        - https://arxiv.org/abs/2008.04445

Available versions of the RANG generator

- Auto version:
    - Sample command to run generator:
        - **python3 networkGeneratorSBM.py auto modelName p randomization networkFile originalNetworkGroups randomNodeIdsFile outputFileNames numberOfNetworks**

    - Command line arguments:
        - **auto**: Indicates the usage of the auto version of RANG generator.
            - Must be passed in as "auto".
        - **modelName**: User has the choice between "BWRN" and "WRG" models.

- **p**: Probability of creating an edge using the specified model.
- **randomization**: If a user wants to randomize node IDs for increased anonymity, the user must pass in “Randomize”, otherwise the user passes in “noRandomize”.
- **networkFile**: Network file in the following format.
    - Source node, target node, weight
- **originalNetworkGroups**: Original groups of the provided network. These groups can be found using either ground truth data, or a community detection method.
- **randomNodeIdsFile**: File with a mapping of the original node IDs to the new randomized node IDs.
    - This file should not be provided if the user chooses “noRandomize” for the randomization command line argument.
    - Format:
        - originalNodeId RandomNodeId
        - originalNodeId column will have the original node IDs, and randomNodeId column will have the new random node IDs assigned to the nodes of the network.
- **outputFileNames**: the name of the file that the generated network will be printed to
    - If more than one network will be generated, the file names will automatically be numbered
- **numberOfNetwork**: the number of networks that the user wants to generate

- Example run using the Karate club network and auto version:
    - Auto/Random version, and BWRN model:
        - **Python3 networkGeneratorSBM.py auto BWRN 0.875 Randomize karate_club.txt networkGroups.txt randomNodeIds.txt karate 10**

        - Auto/noRandom version, and WRG model:
            - **Python3 networkGeneratorSBM.py auto WRG 0.875 noRandomize karate_club.txt networkGroups.txt karate 10**
    - All files are attached to further clarify their expected format.

- Manual version:
    - Sample command to run generator:
        - **python3 networkGeneratorSBM.py manual modelName p randomization originalNetworkGroups totalDegreeOfNetwork hierarchyOfNetwork edgesBetweenGroups nodePreference randomNodeIdsFile outputFileNames numberOfNetworks**

    - Command line arguments:
        - **manual**: Indicates the usage of the manual version of RANG generator.
            - Must be passed in as “manual”.
        - **modelName**: User has the choice between “BWRN” and “WRG”.
        - **p**: Probability of creating an edge using the specified model.
        - **randomization**: If a user wants to randomize node IDs for increased anonymity, the user must pass in “Randomize”, otherwise the user passes in “noRandomize”.
        - **originalNetworkGroups**: Original groups of the provided network. These groups can be found using either ground

truth data, or a community detection method of the users choice.

- **totalDegreeOfNetwork**: Integer value representing the total degree of the network.
- **hierarchyOfNetwork**: File will indicate the leader, and the managers of the network.
    - Sample file for the karate club is provided for clarification.
- **edgesBetweenGroups**: This file will list out the edges between the hierarchical groups in our network.
    - Sample file for the karate club network is provided for clarification.
    - File is in a matrix format:
        - Rows represent the source group, columns represent the target groups.
        - Edges between a source and target group are printed out as space separated values.
        - Edges between different groups are semicolon separated.
- **nodePreference**: This file will be composed of each group's node preference. The preference of each node is relative to other nodes within its group, therefore leader and manager nodes will be left out, since they are each in a group of their own, and thus automatically have a preference value of 1.
    - Sample file for the karate club network is included for clarification.
    - Nodes belonging to the same group are listed out on the same line (different group nodes are printed on different lines).
    - Only low level nodes are included in the preference file.

        - **randomNodeIdsFile**: File with a mapping of the original node IDs to the new randomized node IDs.
            - This file should not be provided if the user chooses “noRandomize” for the randomization command line argument.
            - Format:
                - originalNodeId RandomNodeId
                - originalNodeId column will have the original node IDs, and randomNodeId column will have the new random node IDs assigned to the nodes of the network.
        - **outputFileNames**: the name of the file that the generated network will be printed to
            - If more than one network will be generated, the file names will automatically be numbered
        - **numberOfNetwork**: the number of networks that the user wants to generate

- Example run using the Karate club network and manual version:
- Manual/Random version, and BWRN model:
    - **python3 networkGeneratorSBM.py manual BWRN 0.875 Randomize networkGroups.txt 156 manualHierarchy.txt manualEdgesBetweenGroups.txt manualPreferences.txt randomNodeIds.txt karate 10**
- Manual/noRandom version, and WRG model:
    - **python3 networkGeneratorSBM.py manual WRG 0.875 noRandomize networkGroups.txt 156 manualHierarchy.txt manualEdgesBetweenGroups.txt manualPreferences.txt karate 10**
- All files are attached to further clarify their expected format.