Function Documentation

1 Function: proposal_filter

Usage: `proposal_filter(proposal_csv, proposal_times)`

Description: This function filters out inactive addresses.

- `proposal_csv`: Voting record file for a specific house, containing results of either `FOR` or `AGAINST`.
- `proposal_times`: Filtering threshold, indicating the minimum number of voting participations required. Addresses below this threshold will be filtered out.

2 Function: final_voting_result

Usage: `final_voting_result(voting_res_filter)`

Description: This function organizes the final voting results of proposals (approved or not approved).

- `voting_res_filter`: File containing voting results for all proposals. Each `proposal_id` has two lines of data for votes `For` and `Against`. Ensure the column name for voting results is `'proposal_choice'`.

3 Function: proposal_result_merge

Usage: `proposal_result_merge(voting_res_filter_final, prop_filter)`

Description: This function merges the final results of proposals into the voting record file for a single house.

- `voting_res_filter_final`: File containing processed proposal voting results from `final_voting_result`.
- `prop_filter`: Filtered voting record file for a single house.

4 Function: Similar_matrix

Usage: `Similar_matrix(prop_filter_add, num_same_choice)`

Description: This function computes the similarity matrix of addresses, calculating the number of times addresses voted the same way.

- `prop_filter_add`: File containing proposal result information (i.e., result after `proposal_result_merge`).
- `num_same_choice`: Filtering threshold for the minimum number of same votes; values less than or equal to this threshold will be set to 0.

5 Function: Community_detection

Usage: `Community_detection(M_tranc, min_mod, max_resolution)`

Description: This function performs network clustering and returns category information for each address and clustering results.

- `M_tranc`: Similarity matrix obtained from `Similar_matrix`.
- `min_mod`: Minimum modularity parameter for the clustering algorithm (set to 0.1; range: $0 < min\_mod < 1$).
- `max_resolution`: Parameter for clustering algorithm (set to 3); a higher value indicates more community divisions.

6 Function: Logistic_reg_single

Usage: `Logistic_reg_single(prop_filter_add, Louvain_member, average=TRUE)`

Description: This function calculates the effect of addresses in a single house using logistic regression.

- `prop_filter_add`: Voting record file for a single house (i.e., result after `proposal_result_merge`).
- `Louvain_member`: Community partition information for each address from `Community_detection`.
- `average`: If `TRUE`, assigns the same effect to elements in the same group; if `FALSE`, retains individual logistic regression results.

Note: Remove NA and NaN before calculations to avoid errors.

7 Function: Logistic_reg_multiple

Usage: `Logistic_reg_multiple(data1, data2, ...)`

Description: This function performs logistic regression on common addresses across multiple houses, returning a vector of each address's effect.

- Each parameter `datai` must be a voting record file for a single house (i.e., result after `proposal_result_merge`).

8 Function: CT_Logistic_reg_single

Usage: `CT_Logistic_reg_single(prop_filter, Louvain_member, average=TRUE)`

Description: This function calculates the effect of addresses in a single house using counterfactual logistic regression.

- `prop_filter`: Voting record file for a single house (i.e., result after `proposal_result_merge`).
- `Louvain_member`: Community partition information for each address from `Community_detection`.
- `average`: If `TRUE`, assigns the same effect to elements in the same group; if `FALSE`, retains individual logistic regression results.

Note: Remove NA and NaN before calculations to avoid errors.

9 Function: CT_Logistic_reg_multiple

Usage: `CT_Logistic_reg_multiple(data1, data2, ...)`

Description: This function performs counterfactual logistic regression on common addresses across multiple houses, returning a vector of each address's effect.

- Each parameter `datai` must be a voting record file for a single house (i.e., result after `proposal_result_merge`).

10 Function: Centrality_statistics

Usage: `Centrality_statistics(data)`

Description: This function calculates various centrality measures and returns a list of results.

- `data`: A vector from which NA and NaN values have been removed.