Method explanation

The procedure can be explained as it follows:

1) My raw data are words' popularity indeces (tokens) for each US county. These indeces are float numbers and they are divided in multiple files. I imported everything and normalized them in a way that if you sum all the new values for each word, you get 1. I could have done that on the other axis (counties), but in this way the method would have excluded the rarest words, which should carry more information than the most common ones. I also saved the county info (state, longitude, latitude) in a different csv table.

2) Now I have a matrix (no. of counties) x (no. of words) with indices from 0 to 1. This can be read as the biadjacency matrix of a complete bipartite weighted network, where the lowest weights are about 1e-5. In order to make the network unweighted, I defined a 'link threshold' t with values [0.1, 0.01, 0.001, 0.0001, 0]. If the weight of the link is less than t, the link is cut off. Otherwise its entry becomes 1.

3) I project this new bipartite network, obtaining a network of the counties, which are connected if they have at least a word in common. If t <= 0.001 the projected network is complete.

4) After the projection, I validate the new network. The procedure is here explained very shortly and a full and much clearer explanation can be found in the attached paper 'journal.pone.0017994.pdf'. The validation steps are the following:
i) I assume the null-hypothesis that the bipartite network is fully random, but the degrees of the nodes are the same as in the real network. Thus for each pair of nodes A-B in the projected one, I calculate its p-value. This value is calculated using the hypergeometric complementary cdf with parameters N_AB (number of common words of A and B in the bipartite network), N (number of projected nodes), N_A (degree of the node A in the bipartite network) and N_B (degree of the node B in the bipartite network). The p-values answer therefore the following question: "what is the probability that the counties A and B have at least N_AB common words?"
ii) I define a threshold, here t = 0.01. Because of the multiple hypothesis testing, I have to apply a correction. Here I used both FDR and Bonferroni correction, generating two versions of each network. The FDR is a less restrictive correction, thus all the Bonferroni networks are fully included in the FDR ones.
iii)If the p-value of a link is over the corrected threshold, the link can be considered random and it doesn't contain any information about the counties, thus it is cut off. Also the nodes that become isolates after the validation are removed from the final network.

5) The validated networks are referred here as SVN (Statistically Validated Networks) and their sizes can be found in the 'size_table.csv' file. It contains info about the four networks (bipartite, projected, svn_bonferroni, svn_fdr) at each threshold. The info consist in the number of nodes, the number of edges and the ratio between the number of edges in the validated network and in the projected one. This last column measures the effectiveness of the method, that is high if the number of surviving nodes is very low, compared the original one. Therefore the conclusion is that it doesn't work very well with the 0.001 threshold, but it's very effective at higher thresholds. Seeing this result, I didn't validate the networks with t=0.0001 nor the one with t=0, because the validation would have returned a very dense network with very little information about its nodes.

6) After the validation, I run two different community detection algorithms (infomap and louvain) on each one of the six networks. They returned different results and an overall view of these result can be found in the 'table_svn_s1_over-expr.csv' file. This file contains for each network the number of isolate nodes, the number of components, the number of infomap/louvain communities and the modularity returned by the louvain algorithm. From the modularity it is immediate to see that the quality, of the louvain partitioning, gets lower for lower values of t. The 0.1 networks have more than 2800 isolates and the survived nodes are grouped in very small components, so the number of communities is almost the same of the number of components (it's exactly the same for louvain). The 0.001 ones are very dense and both algorithms struggled to partition the networks, returning 2 or 3 communities. The 0.01 ones have a number of components/communities comparable to the number of US states, expecially the bonferroni n_components (48) and n_infomap_communties (52) and the fdr n_infomap_communities (also 52).

7) Using the LONG and LAT data about each county, I plotted all of them colouring them according to the community. These plots are not very accurate because the colouring is random and different communities may have the same colour. All the isolate nodes are coloured in grey. The different size of the markers doesn't carry any information, it's just an aestethic adjustment. These plot are in the folder usa_plots.

8) To have more information about which states are similar to which communities, I applied a community characterization method, also based on the hypergeometric distribution. The details of this method can be found in the attached paper 'jstat.pdf'. Here follows a short explanation:
i) I assume the null-hypothesis that counties are thrown randomly into communities, but their size is the same as in the real network. Thus I calculate a p-value for each pair Community-State using the hypergeometric complementary cdf with parameters N_CS (number of common nodes of C and S), N (number of nodes in the network), N_C (size of the community in the network) and N_S (size of the state in the network). The p-values answer therefore the following question: "what is the probability that the community C and the state S have at least N_CS common nodes?"
ii) I define a threshold, here t = 0.01. Because of the multiple hypothesis testing, I have to apply a correction. Here I used both FDR and Bonferroni correction, generating two versions of each community characterization.
iii)If the p-value of a pair is over the corrected threshold, the pair can be considered random and it doesn't contain any information about the counties. All the pairs, whose p-values are lower than the threshold are marked in red in the heatmaps and they are considered 'over-expressions of the state in the community'. The code also calculated under-expressions, but as long as there are none in these networks I will not explain it here. The heatmaps can be found in the 'heatmaps' folder.