Published in final edited form as:
Trans GIS. 2021 April ; 25(2): 1065–1081. doi:10.1111/tgis.12722.

Network Optimization Approach to Delineating Health Care Service Areas: Spatially Constrained Louvain and Leiden Algorithms

Changzhen Wang[1], **Fahui Wang**[1,*], **Tracy Onega**[2]

[1]Department of Geography & Anthropology, Louisiana State University, Baton Rouge, LA 70803;

[2]Dartmouth-Hitchcock Medical Center, One Medical Center Drive, HB 7927 Rubin 8, Lebanon, NH 03756

Abstract

Constructing service areas is an important task for evaluating geographic variation of health care markets. This study uses cancer care as an example to illustrate the methodology, with the nine-state Northeast Region of the U.S. as the study area. Two recent algorithms of network community detection are implemented to account for additional constraints such as spatial connectivity and threshold region size. The refined methods are termed "spatially-constrained Louvain (ScLouvain)" and "spatially-constrained Leiden (ScLeiden)" algorithms, corresponding to their predecessors Louvain and Leiden algorithms, respectively. Both are network optimization methods that maximize flows within delineated communities while minimizing inter-community flows. The service areas derived by the methods, termed "Cancer Service Areas (CSAs)", are more favorable than the commonly used comparable unit, Hospital Referral Regions (HRRs) for evaluating cancer-specific variation in care. Between the two, the ScLeiden performs better than ScLouvain in modularity, localization index and computational efficiency, and thus is recommended as an effective and efficient approach for defining functional regions.

Keywords

network optimization; community detection; localization index (LI); spatially-constrained Louvain (ScLouvain) algorithm; spatially-constrained Leiden (ScLeiden) algorithm; functional regions

1. Introduction

Delineating service areas is important for evaluating geographic variation of health services delivery, utilization and outcomes. The Dartmouth Atlas Project spearheaded the work by defining the Hospital Service Areas (HSAs) and Hospital Referral Regions (HRRs) in the U.S. (Wenberg and Cooper, 1998). These areas better capture local health care markets than administrative or census units, and thus have been used widely in evaluating health care systems (Kilaru et al., 2015; Klauss et al., 2015; Zhang et al., 2012; Zuckerman et al., 2010) and informing health policy (Newhouse et al. 2013). However, health care markets

*Corresponding author: Fahui Wang, fwang@lsu.edu.

evolve over time, and also vary by specific services. In order to derive service areas that are timely and cater to a particular service type, it is highly desirable to develop a method that is scientifically sound (i.e., capturing coherent market structure), automated (thus easy to implement and replicate), computationally efficient (therefore feasible for large-scale or national markets) and adaptable (e.g., scale-flexible both in terms of market size and number of sub-markets being delineated). This paper focuses on developing methods that meet these challenges.

Since prior methods involve uncertainties or some manual assignments of spatial units with limited automation (Hu et al. 2018), there have been several notable efforts to improve the method. Klauss et al. (2005) employed a slightly refined Dartmouth method to create population-based HSAs for health care planning in Switzerland. Jia et al. (2015) revised the Huff model by incorporating the best-fitting distance decay function to delineate the HSAs in Florida. However, the automated toolkit for the Huff model still requires some post-model manual processing in order to attain desirable localization index and spatial contiguity. The most promising work is based in community detection algorithms from network science. While the Dartmouth method aggregates small areas by emphasizing the greatest proportion of patient visits (i.e., the plurality rule), community detection considers almost all patient visits to service facilities. Dense connections between patients and facilities are grouped to form initial communities (or clusters), and dense connections between communities are further grouped to form higher-level communities until a single community for a whole study area is attained. The process is guided by maximizing a quality measure, “modularity”, for detected communities, which can form the foundation for service areas to be derived. The work by Hu et al. (2018) is among the first in applying the approach to delineate HSAs. They adopt a popular and high-performing community detection method, “Louvain algorithm” (Blondel et al. 2008, named after the University of Louvain, with which the authors are affiliated), to define HSAs and HRRs in Florida. Compared to prior methods, their results were more favorable in capturing the localization pattern of hospital visits and by other evaluation measures.

However, the modularity-based community detection algorithms are not free of concerns. The first one is the resolution limit problem (Fortunato and Barthélemy, 2007; Kumpula et al., 2007). Because of the intrinsic scale of modularity, it may fail to detect small communities by grouping them into larger communities, and thus conceal some significant structure of a network embedded in those small communities (Krings, 2011). One remedy involves tunable resolution parameters to allow community detection at different scales, and thus generate a series of units to facilitate a deep look at the community configuration (Reichardt and Bornholdt, 2006). By doing so, researchers may find a desirable resolution that yields the optimal delineation to represent the actual structure of a network. Others suggest resolution-limit-free quality functions such as the Constant Potts Model (CPM) to mitigate the limitations (Traag et al., 2011). The second concern refers to a recently-discovered deficiency of Louvain algorithm, which tends to yield arbitrarily poorly connected or even internally disconnected communities (Traag et al., 2019). Thirdly, when applying community detection algorithms to delineate geographic areas, as this study focuses on, we need to account for spatial constraints.

One major spatial constraint in deriving service areas is to ensure contiguity between geographic areas, a common challenge in the long tradition of regionalization. One approach is to enforce a spatial contiguity rule during or after the regionalization process (Guo, 2008). Another is to integrate spatial factors or models explicitly in the objective function of cluster (or region, network) partitions (Guo et al., 2018). To list a few, Expert et al. (2011) propose a modularity function by incorporating the gravity model to identify linguistic separation and use the absolute value of difference between the actual flows and estimated flows. Gao et al. (2013) use the ratio of actual flows over estimated flows (also based on the gravity model) as a modularity measure to detect small communities, which are more sensitive to the ratio change than large communities. Chen et al. (2015) develop a geo-distance-based method by introducing a connection weight as the inverse of distance to the power of n to the modularity. Jia et al. (2019) impose a spatial effect to constrain the network by controlling edges with topological distance less than value k and then apply the Louvain algorithm to detect regional delineations. However, it introduces extra assumptions (i.e. distance decay) and burden of defining more parameters (i.e. choice of different models and parameters) (Guo et al., 2018), or even induces fragmented regions (i.e. , orphan, enclaved or enclosed) that are inappropriate for further analyses and policy decision making.

The recent development of Leiden algorithm is clearly motivated by remedies to the aforementioned deficiencies in the Louvain algorithm (Traag et al., 2019). The Louvain algorithm is enormously popular, and the Leiden algorithm has just come out and is well received by the network science community. Therefore, this study builds upon these two community detection algorithms and develops their spatially constrained versions, termed “spatially-constrained Louvain (ScLouvain)”[1] and “spatially-constrained Leiden (ScLeiden)” algorithms. We apply the two refined methods to delineate service areas of health care, specifically cancer service areas (CSAs) in the nine-state Northeast Region of the U.S. We use the Dartmouth HRRs as a baseline to evaluate their possible advantages, and then compare the two in performance measured by modularity, localization and computational time. One method is recommended for future studies such as delineation of CSAs in the U.S. that are composed of tens of millions of nodes and hundreds of millions of service flows.

2. Spatially Constrained Community Detection Methods

2.1 Louvain and Leiden Algorithms

Modularity is one of the best-known quality measures in community detection studies (Newman, 2004b; Newman and Girvan, 2004). It compares the total number of edges (or total weights of edges for a weighted network) fallen within all communities in a given network to that in a null model (i.e., random network). The goal of community detection is to maximize the difference between the actual number of flows within each community and expected number of such flows by chance (random). As a result, the internal flows within each derived community are more tightly connected with each other while the inter-

[1] A new study (Wang et al. 2020) uses a refined Louvain method with spatial constraints to define health care service areas and reports promising results. However, as demonstrated in this paper, the spatially-constrained Louvain (ScLouvain) algorithm is outperformed by the spatially-constrained Leiden (ScLeiden) algorithm.

community flows are sparsely connected. Modularity is given by incorporating a tunable resolution parameter:

$$Q = \sum_{c \in C}\left(\frac{l_{in}^{c}}{m} - \gamma\left(\frac{k_{tot}^{c}}{2m}\right)^{2}\right) \tag{1}$$

where Q represents the modularity score that sums over each community $c \in C$, l_{in}^{c} is the total number of edge weights between all nodes within community c, k_{tot}^{C} is the sum of the edge weights between nodes in community c and nodes in other communities, and m is the total number of edge weights in the network. In Equation (1), the first term $\frac{l_{in}^{c}}{m}$ represents the fraction of the sum of the edge weights in community c, and the second term $\frac{k_{tot}^{c}}{2m}$ represents the expected fraction of edge weights between different communities. Therefore, their difference is between the fractions of edge weights in intra-communities and those in inter-communities. To maximize the modularity in Equation (1) is to simultaneously maximize the intra-community edge weights while minimizing the inter-community edge weights. The constant γ>0 is the resolution parameter. When γ =1, the modularity is equivalent to the original form (Newman, 2004a). A higher resolution γ leads to a larger number of communities. Generally, a higher modularity score represents higher-quality communities, and thus a more stable and robust network structure. However, increasing the number of communities does not necessarily increase the modularity as the contribution of each community varies. Some may have positive contributions while others may have negative ones. So Equation (1) provides a viable way to find the tradeoff between the value of each community and the number of communities, and assists us to search the global optimal modularity Q at a particular resolution (Fortunato and Barthélemy, 2007).

Correspondingly, the local gain of modularity of moving (grouping) a node i to a community c, denoted by $\Delta Q_{i \to c}$, is simplified as:

$$\Delta Q_{i \to c} = \frac{1}{m}\left(k_{i,in} - \gamma\frac{k_i * \Sigma_{total}}{2m}\right) \tag{2}$$

where γ >0 remains the resolution parameter, m is the total number of edge weights in the network, $k_{i,in}$ is the sum of edge weights from node i to nodes in community c, k_i is the sum of edge weights linked to node i, and Σ_{total} is the sum of edge weights linked to nodes in community c.

The Louvain algorithm is a popular and high-performing technique for modularity optimization (Blondel et al., 2008). It is simple and elegant to detect communities from network flows in two phases: (1) local nodes moving, and (2) network aggregation. The local nodes moving phase begins with treating each node of the network as a unique community (i.e., the number of communities is the number of nodes), and then removes each node from its original community to different communities temporarily and finds the community for which the local gain of modularity in Equation (2) is maximal. If the community is detected, the node is moved to this community permanently, otherwise, the node is placed back to its original community. Such a process iterates across all nodes until each node is optimally assigned to their communities and no further improvement of modularity is possible. The network aggregation phase starts from generating a new network by aggregating communities in the local nodes moving phase to be new nodes and computing the sum of the weights of the edges between nodes in the communities to be new weights of edges. It then repeats the first phase until obtaining a single community or no further improvement of modularity. Since the bottom-up process builds communities of communities iteratively, it forms a hierarchy in which each level has a local optimum of modularity. In fact, the levels of the hierarchical community structure correspond to the iteration of the first phase. Communities at different levels are nested with each other, hence allow researchers to investigate the internal configurations.

However, the Louvain algorithm is apt to yield poorly connected or even disconnected communities when iteratively running it (Traag et al., 2019). All nodes in the first phase are eventually moved to an expected community. If a node happens to be a network hub, when moving it from an old community to a new community that has maximal gain of modularity, the node may still act as a virtual bridge between other nodes in the old community. If other nodes are strongly connected to the old community in a way of sub-communities which are also internally strongly connected, then removing such a node will disconnect the old community if no further movement of other nodes. Perhaps doing so will increase the modularity but may not accurately uncover the network structure. Caution may be needed when using the single indicator, modularity, to measure the quality of network partition.

The Leiden algorithm is introduced to remedy the drawback of the Louvain algorithm (Traag et al., 2019). It guarantees communities well-connected in a higher modularity and higher computational efficiency. The Leiden algorithm is more complex and consists of three phases: (1) local nodes moving, (2) refinement of the partition, and (3) network aggregation based on the refinement, but using the non-refined partition to initialize the aggregated network. It integrates the advantage of fast local move (Bae et al., 2017; Ozaki et al., 2016) to speed up the first phase of the Louvain algorithm. It also incorporates smart local move (Waltman and Van Eck, 2013), which itself is an improved version of the Louvain algorithm and random neighbor move (Traag, 2015) to uncover better partitions with well-connected communities.

For the first phase, instead of keeping moving all nodes in the Louvain algorithm, the Leiden algorithm employs the fast local move procedure to only visit those whose neighboring communities have changed. In short, the fast local move procedure randomly loads all nodes of the network into a queue, it will then move the first node from the front of the queue to different communities and detect the one for which the local positive gain of modularity is maximal. The node's neighbors that are neither in the new community nor in the queue will be added to the rear of the queue. Such steps continue until the queue is empty. In this way, the Leiden algorithm is more efficient than the Louvain algorithm. For the Web UK 2005 network that is composed of 39 million nodes and 783 million edges, the Leiden algorithm is more than 20 times faster than the Louvain algorithm (Traag et al., 2019).

The Leiden algorithm uses smart local move in the refinement phase to detect well-connected subnetworks in each community and aims to provide more room to split them for high-quality partition. Specifically, it treats each community from the first phase as a subnetwork. For each subnetwork, it assigns each node to a community and only considers those well-connected within the subnetwork. It then uses random neighbor move to identify any community that increases the modularity rather than maxima, which allows broader exploration in the partition space. This process iterates until all communities from the first phase are visited. As a result, some communities may be split into sub-communities while others remain intact. Therefore, the Leiden algorithm resolves any poorly connected or disconnected communities.

The proposition of the Leiden algorithm was motivated by mitigating some of the problems associated with the Louvain algorithm that was developed more than a decade before. However, there is no guarantee that the Leiden algorithm outperforms the Louvain algorithm in all applications. Traag et al. (2019, pp.8, Figure 6) themselves provided such a counterexample when the benchmark network has 10^7 nodes and the mixing parameter is 0.2. Some uncertainty is also associated with the nodes moving step, where the order of nodes being moved is random in both algorithms. The Leiden algorithm is just new, and like any methods, a full validation of its advantages awaits as more applications are implemented in various fields.

2.2 Spatially-Constrained Leiden and Louvain Algorithms

This research refines the Louvain and Leiden algorithms to ensure that derived service areas are spatially contiguous, termed spatially constrained Louvain and Leiden methods (referred to as “ScLouvain” and “ScLeiden” hereafter). The ScLouvain method has three steps: (1) local nodes moving, (2) network aggregation, and (3) spatial contiguity and minimal region size guarantee. The ScLeiden method contains four phases: (1) local nodes moving, (2) refinement, (3) network aggregation based on the refined partition, and (4) spatial contiguity and minimal region size guarantee. The last step in both methods is added to the original algorithms by this research to make them spatially constrained.

Figures 1 and 2 illustrate the implementation of ScLouvain and ScLeiden methods in a simple network, respectively. Both methods derive a two-level hierarchical community structure (see ABC and CDE in Figure 1, and ABCD and DEFG in Figure 2) and four different continuous service areas (see Figure 1G and Figure 2I). To highlight the process, each network structure is displayed in different phases. As shown in Figure 1A and Figure 2A, there are 28 nodes in different colors to represent the locations of cancer patients or service facilities in 28 grey dash line polygons which can be ZIP code areas, census tracts, or any other geographic areas. The solid grey lines between the nodes are the network edges. Thick edges represent large service volumes whereas thin edges represent small service volumes between residence and facilities in both directions. The network has any two nodes connected with each other except the soft orange dot as there is no incoming or outgoing service volume. In our case, both patients and facilities are consolidated into ZIP code areas, represented by the polygons. All nodes of the simple network are derived from the population-weighted centroids of the ZIP code areas for better accuracy, especially in rural and suburban areas with less and uneven populations (Wang, 2015, pp.78). The nodes represent the location of patients or the location of facilities or both. The edge weights of the simple network are consolidated from the service volumes between the nodes in both directions. To explicitly elucidate the difference of the ScLouvain and ScLeiden algorithm, we demonstrate the partition process of two methods in detail.

In the nodes moving phase, both the ScLouvain and ScLeiden methods assign 28 nodes to 28 individual communities (see Figure 1A and Figure 2A). Then the ScLouvain method repeatedly moves each individual node to different communities and identifies the community for which the positive gain of local modularity is the greatest. This process continues until there is no further improvement of modularity at the current level. Eventually, it detects 7 communities in terms of the entire network and each community contains the same color nodes in Figure 1B. In contrast, the ScLeiden method uses the fast local move procedure to speed up the process as there may exist unnecessary movements of all nodes. Some nodes may still stay in their original community after a series of move in and out operations. As shown in Figure 2B, the ScLeiden method also derives 7 communities represented by 7 different colors of nodes when the partition is stable. Note that 2 internally separate nodes in blue color belong to the same community in terms of the entire network but they are 2 subcommunities in terms of the local subnetwork. This is also the significant difference between the ScLeiden and ScLouvain methods. We can see the lime green node in the middle has a strong connection with a nonadjacent node which is assigned to the same community for the maximal modularity gain. It also acts as a bridge to connect the blue nodes in the same community although they are internally disconnected.

Next, the ScLouvain method creates a new network that 7 communities become 7 nodes and the service volumes between any nodes in the community are aggregated to be service volumes between any new nodes (see Figure 1C). However, the ScLeiden method adopts smart local move and random neighbor move to refine the 7 communities by splitting the blue dots community into two subcommunities in which nodes are well-connected (see the shade area in Figure 2C). Since the non-refined partition is needed to initialize the aggregated network, we still obtain 7 communities but with one is locally separated into two communities (see two blue nodes community in Figure 2D). The reason to use the non-refined partition to initialize the aggregated network is to guarantee the modularity increases monotonically. Although the poorly connected communities detected in this phase have small edge weights between nodes in each community, having one big community in the aggregated network will have modularity higher than that derived from multiple subcommunities detected in the refinement phase.

Then both methods start the second iterations. Because the three nodes in the top-left corner of the network (see Figure 1D and Figure 2E) have strong connections with each other, two methods move them into one community for the greatest modularity gain in the local nodes moving phase. The ScLeiden method even moves the blue nodes in the lower-right corner to the green node community (see Figure 2E) as the movement maximizes the modularity gain. The ScLouvain method continues to aggregate the network to be a new one that contains 5 new nodes in Figure 1E. While the ScLeiden method refines the partition and aggregates the network to be a different one (see lower-right polygons in Figure 2G) but with the same number of nodes. Both methods obtain the best aspatial community structure with no further improvement of modularity. Thus, both detect communities within which the edge weights are maximal while between different communities, the edge weights are minimal.

The final phase is to enforce the spatial contiguity and minimal region size rules. We impose a predefined spatial adjacency matrix into the aspatial network to split the communities that are not spatially contiguous (e.g., orphan, enclaved or enclosed nodes). The lime green nodes' community is separated into two communities in Figure 1F and Figure 2H as the lower-left kaitoke green polygons in Figure 1E and Figure 2G is not adjacent to other lime green polygons respectively. Both methods derive 6 communities (or nodes) with a cost of decreasing modularity in Figure 1F and Figure 2H. If any node has a regional size that is below the threshold, there are three scenarios:

1. for an enclosed node with one neighboring node, merge it directly to the neighbor (not shown in Figures 1 and 2 due to limited space);
2. for an enclaved node that has one more neighboring nodes with connecting edges, merge it to the one that can achieve the maximal positive modularity gain (e.g., the lower-left kaitoke green node in the dash kaitoke green polygon in Figures 1F and 2H); and
3. for an orphan node without any edges to its neighboring nodes, group it to the one that has the smallest regional size (e.g., the soft orange node in Figures 1F and 2H).

This process continues until all nodes in the network have a minimal regional size and a global optimal modularity has been achieved. Eventually, both methods dissolve the corresponding ZIP code areas (or polygons) to be contiguous service areas within which the edge weights are maximal whereas they are minimal between different service areas, showing in Figure 1G and 2I. We can see the blue and lime green service areas are quite different as the ScLeiden method has the capability to identify the poorly or disconnected communities and refines them.

As stated previously, the Leiden algorithm is very likely to outperform the Louvain algorithm in many cases but not necessarily always. Adding the spatial constraints introduces further complexity to the process, especially in the network aggregation phase. Therefore, it is important to assess whether the expected advantages in one algorithm over the other are materialized in their spatialized counterparts.

3. Case Study of Delineating Cancer Service Areas (CSAs) in the Northeast Region

The case study applies the aforementioned ScLeiden and ScLouvain methods to delineating Cancer Service Areas (CSAs) in the Northeast region of the U.S. The Northeast region is composed of nine states: Maine, New Hampshire, Vermont, Massachusetts, Connecticut, Rode Island, New York, New Jersey, and Pennsylvania. The cancer care data represents cancer-directed surgery, chemotherapy, and radiation services from the Medicare beneficiary denominator file, the Medicare Provider Analysis and Review files (MedPAR), Outpatient

Files, and the Medicare Part B claim files. Based on the International Classification of Diseases, Ninth Revision, and Clinical Modification (ICD-9-CM) diagnosis, 26 cancer types are extracted from January 1, 2014 to September 30, 2015. Both patient residence and hospitals are geocoded to ZIP code areas, and thus the network of patient service volumes is constructed between ZIP code areas. For the study area, the network contains 5,969 nodes (represented by the ZIP code centroids) and 86,192 edges (i.e., service flows between ZIP code areas) with the total service volumes (sum of edge weights) of 2,443,538.

3.1 Spatial Constraints

As stated previously, two constraints are added to differentiate the ScLeiden and ScLouvain methods from their non-spatial predecessors. One is to ensure spatial contiguity in the derived CSAs, and another is the minimum (threshold) size for them. For the threshold size for CSAs, we choose a population of 120,000, similar to what is used for Dartmouth HRRs, since cancer care is also considered a highly specialized medical care. In other words, based on the 2010 census data, the total population of ZIP code areas in any derived CSA needs to be 120,000 or more.

Enforcing the spatial constraints has a substantial impact on computational time and modularity. The computational times for the original Leiden and Louvain algorithms for our case study are negligible (near zero) for the network of 5,969 nodes and 89,162 service flows. Section 3.3 discusses the computational times after imposing the constraints. Understandably, the modularity decreases at the cost of splitting up those enclaved or enclosed communities to ensure spatially continuity. For example, for the scenario of 17 CSAs, the ScLouvain and ScLeiden methods yield modularity values of 0.789 and 0.791, respectively, lower than 0.791 and 0.792 by the non-spatial Louvain and Leiden methods, respectively. Note the higher modularity value by the ScLeiden method than the ScLouvain method.

The spatial constraints also affect the configuration of CSAs. Figures 3A–3D show the configurations of 17 CSAs before and after enforcing spatial constraints. Maps from the original Louvain and Leiden algorithms, as shown in the insets of Figure 3A and 3C, contain many small isolated units in the form of enclaved, enclosed, or orphan areas. These sparse units are all eliminated in the refined Figures 3B and 3D by ScLouvain and ScLeiden methods, respectively. Section 4 discusses differences in the results between the two spatially-constrained methods.

3.2 Poorly Connected CSAs

As discussed previously, one possible problem of the Louvain algorithm is that it leads to some poorly connected communities. We selected the global modularity with a resolution parameter γ =1 for the experiment and implemented 5 times. The average percentage of poorly connected CSAs (including disconnected CSAs) and corresponded modularity found by two methods are shown in Figure 4. For the ScLouvain method, the first iteration generates 45.88% poorly connected CSAs and shows a large decrease in the second iteration, only 9.41% poorly connected CSAs is found. Then the percentage of poorly connected CSAs fluctuates around 12%−21% in the subsequent iterations. Correspondingly, the modularity keeps in pace with the percentage of poorly connected CSAs. It increases when lower percentage of poorly connected CSAs is detected in the second iteration, but then moves up and down around 0.789. For the ScLeiden method, it underperforms at the onset by yielding a highest percentage of poorly connected CSAs, and is capable of resolving the problem as the number of iterations increases. The poorly connected CSAs are completely eliminated at the fourth iteration by the ScLeiden method. The modularity increases and becomes stable after the third iteration. Therefore, the ScLeiden method is effective in detecting and eliminating poorly connected CSAs.

3.3 Computational Efficiency

As mentioned previously, the ScLeiden and ScLouvain methods allow users to generate a series of CSAs at different scales corresponding to predefined resolution values. This study simulates 1,000 scenarios with resolution ranging 0–10.0 with an increment of 0.01 in both methods. The ScLeiden method uses 0.01 to determine the degree of randomness in the refinement phase. We experimented with other values within the suggested range of [0.0005, 0.1] for the degree of randomness, and found only minor differences. The algorithms were coded in python and integrated with the Java packages, and all experiments were executed in a desktop of Inter(R) Core(TM) i7-4770 CPU@ 3.40GHz with 32GB of memory.

Figure 5A shows that the trend lines of modularity versus the number of CSAs from the two methods are largely consistent. Both yield the global optimum of 17 CSAs with the highest modularity. Note that for the same number of CSAs, there are multiple modularity values corresponding to different resolutions with small increments. As a higher modularity implies more stable and robust delineation, the scenario of 17 CSAs suggests the best configuration of the network. As shown in Figure 6A, for the 17 CSAs, the ScLeiden method yields the global maximal modularity of 0.791, slightly higher than 0.789 from the ScLouvain method. As discussed in the next section in more detail, this study also delineates 43 CSAs, comparable to the same number of Dartmouth HRRs in the Northeast region. The ScLouvain method yields 43 CSAs with a modularity of 0.597 when resolution is set 4.11, and the ScLeiden method generates 43 CSAs with a higher modularity of 0.601 when resolution is set 4.03.

Figure 5B shows that the two methods also perform consistently in terms of computational time corresponding to the same number of CSAs. On average, the ScLeiden method takes 8.415 seconds, lower than 8.468 seconds by the ScLouvain method. The computational time increases as the number of CSAs declines during the network aggregation process, and both methods follow a power function. Most of the computational time is consumed in the phase of enforcing the spatial constraints. This is consistent with the finding by Traag et al. (2019) that the Leiden algorithm is slightly faster than the Louvain algorithm. We speculate the saving time can be more significant when we move to delineating the nationwide CSAs.

4. Comparing CSAs by ScLouvain and ScLeiden Methods and to Dartmouth HRRs

4.1 Quality Measures

We select three commonly-used indices from health care studies to evaluate the quality of derived CSAs: localization index (LI), geographic compactness, and balanced region size. LI refers to the proportion of patients that receive treatment in the same HSA as where they live. This study defines LI as the ratio of service flows within a CSA divided by the total service flows originated from the same CSA. It represents the tendency of cancer patients to seek local hospitalization, and is considered the most important property in service area delineations. A higher LI is more favorable as it demonstrates that generated CSAs better reflect local cancer care markets. The other two measures (geographic compactness and balanced region size) are widely used to assess the quality of regionalization from a geographic perspective (Guo, 2008). Geographic compactness describes the regularity of a CSA's shape based on the perimeter (P)-area (A) corrected ratio or $PAC = P/\left(3.54\sqrt{A}\right)$. A lower PAC value implies a more compact CSA that is more acceptable in system planning (Mu and Wang, 2008). Balanced region size is often desirable so that CSAs are comparable in population size.

Two scales are selected here for evaluating the performance of our new methods: 17 CSAs with the global optimal modularity, and 43 CSAs as comparable to 43 Dartmouth HRRs in the study area. The results on the 17 CSAs by the two methods are first compared. As shown in Table 1, the mean LI of 17 CSAs from the ScLeiden method is slightly higher than that from the ScLouvain method (0.883 vs. 0.880, see Figure 6B). The range of LI from the ScLeiden method is significantly smaller (0.671–0.978 vs. 0.613–0.978) than that of the ScLouvain method, so is the standard deviation (0.076 vs. 0.087). Note that the lower bound of LI by the ScLeiden, 0.671, is significantly higher than that by the ScLouvain, 0.613. This is likely to be attributable to the advantage of ScLeiden in reducing poorly-connected networks. For the compactness index, the average value of PAC from the ScLeiden method is slightly lower than that from the ScLouvain method (3.632 vs. 3.648, see Figure 6C) with a slightly larger range and standard deviation. The lower mean PAC indicates more compact CSAs on average and thus an advantage in the ScLeiden method again, but their variability is a bit higher. On the population sizes of CSAs, the mean size is identical for the two methods (see Figure 6D) since the number of CSAs is the same (17). The ScLeiden method has a smaller range and a lower standard deviation, and thus more balanced CSAs than those from the ScLouvain method.

Comparison between the results on the 43 CSAs by the two methods leads to similar conclusions with one exception. The ScLeiden method yields the CSAs with a slightly higher average LI (0.746 vs. 0.745, see Figure 6B) and more balanced region size (standard deviation = 987,000 vs. 993,000) than those by the ScLouvain method, but slightly less compact in shape (average PAC = 3.056 vs. 3.054, see Figure 6C). Overall, the ScLeiden method outperforms the ScLouvain method in higher LIs and more balanced region size in generated CSAs.

Finally, we compare the same number of (43) CSAs derived by the two methods to the 43 HRRs defined by the Dartmouth methods. The advantages are far more obvious for the CSAs by the two methods: the mean LI values are 0.746 and 0.745 for the CSAs vs. 0.676 for the HRRs (see Figure 6B), the PAC = 3.056 and 3.054 for the CSAs vs. 3.138 for the HRRs (see Figure 6C), and the standard deviation of population sizes are 987 and 993 for the CSAs vs. 1213 for the HRRs. The derived CSAs enjoy higher LI, more compact shape and more balanced region size than their HRR counterparts. This once again demonstrates the values of deriving a distinctive service area unit for cancer care instead of simply adopting the generic HRRs to capture the cancer care market structure. Note that both ScLouvain and ScLeiden optimize the network segmentation with a clear objective of maximizing modularity and yield notable results in LI that are already high. So their advantages over the Dartmouth method are clear. In the meantime, the comparative advantage of ScLeiden over ScLouvain, which appearing minor in LI value, cannot be understated. Improvement over a widely popular algorithm, considered revolutionary at the time of its creation, is not an easy feat.

4.2 Spatial Configurations

This section compares the spatial configurations of CSAs, first on the 17 global optimal CSAs derived from the two methods, and then the 43 CSAs (between the methods and also to their comparable 43 HRRs).

As shown in Figures 7A and 7B, the boundaries of 17 CSAs derived from the ScLouvain and ScLeiden methods align well with the major service flows of cancer patients overall and demonstrate the scientific soundness of both the methods. In fact, 16 CSAs from both methods have LI values above 0.8. The CSAs by the two methods are also largely consistent with each other, with some minor discrepancies (see Figure 7C). Two insets on Figure 7C highlight the regions of discrepancies. In the lower-right inset, the lowest LI is located at the Poughkeepsie-Newburgh region by both methods, however, the CSA there by the ScLeiden has a higher LI value (0.671) and is slightly larger (population = 1,165,439) than the one there by the ScLouvain (LI = 0.613, population = 1,017,078). The significant improvement in LI for this CSA by the ScLeiden method comes with only a slight drop of LI for its southeastern neighboring CSA (from 0.964 to 0.957). In the upper-left inset, the Ogdensburg area in New York (with the Richard E Winter Cancer Center) switches its CSA affiliation between the two methods. Our close examination reveals that this swing area has relatively stronger connections with the pink CSA (to its east) than the purple one (to its south), and this is captured by the ScLeiden method better than the ScLouvain. Both showcase the advantages of the ScLeiden over the ScLouvain method.

The comparison is extended to the 43 Dartmouth-HRR-comparable CSAs. Again, the ScLeiden and ScLouvain methods derive CSAs with largely consistent boundaries. Due to the similarity between them, only the 43 ScLeiden-derived CSAs are overlaid with the service flows, shown in Figure 8A. Each CSA encloses major service flows of cancer patients while flows between other CSAs appear negligible. Figure 8B helps us identify some minor discrepancies between the 43 CSAs by the two methods. The upper-left inset shows that both the methods divide the region into 3 CSAs but with different spatial configurations. While the LIs differ little between the two (average LI = 0.877 for the 3 ScLeiden-derived CSAs < 0.880 for the 3 ScLouvain-derived CSAs), the ScLeiden method seems to yield more balanced region size among the three CSAs than the ScLouvain (by expanding the smallest CSA southward). The lower-right inset shows how the ScLouvain and ScLeiden methods segment the area into two CSAs differently, and the ScLeiden yields two CSAs with more similar LIs (0.843 and 0.808) than the ScLouvain (0.879 and 0.711), and the improvement of LI in one (0.808–0.711=0.097) clearly more than offsets the loss of LI in another (0.879–0.843=0.036). Once again, the ScLeiden method outperforms ScLouvain method in deriving better connected CSAs of more balanced size.

Figure 8C overlays the 43 Dartmouth HRRs with the service flows and reveals that the HRRs are less well suited to capture the cancer care market. As discussed in Section 4.1, the advantages of CSAs derived by the two methods are evident over the Dartmouth HRRs. Due to the similarity between the two systems of CSAs, only the 43 ScLeiden-derived CSAs are overlaid with the same number of HRRs, shown in Figure 8D. The discrepancies are apparent. Here two areas are highlighted as examples to illustrate the gaps. The upper-left inset in Figure 8D shows that the CSAs at the south corner (Pittsburgh region) with highly interwoven service flows and a high LI of 0.974 is split into three HRRs with much lower LIs, indicating an expanded coverage of cancer related services that attract more people from distant places. The lower-right inset in Figure 8D shows that both four CSAs with relatively high LIs in the Boston area are simply merged into one mega-HRR. Decomposition of such a large unit is favorable because it helps balance the CSA size and enables researchers to measure possible variations within it.

5. Concluding Comments

One recent advancement of the network science has been the development of efficient community detection algorithms. Community detection segments a network into densely-connected communities by optimizing a quality measure of network connectivity such as modularity. As a result, the connections are maximized within derived communities and minimized between the communities. Community detection is well suited for the task of delineating health care service areas, which are widely used for evaluation of geographic variation of health care market. This paper uses cancer care as an example to illustrate the methodology. The case study defines the Cancer Service Areas (CSAs) in the nine-state Northeast Region of the U.S., and the network is composed of service flows of cancer treatments between ZIP code areas.

Specifically, two algorithms of network community detection are selected: the widely popular Louvain algorithm, and the newly-developed Leiden algorithm aiming to address some identified concerns of the Louvain algorithm (e.g., the poorly-connected communities). For the purpose of defining CSAs, we refine the two algorithms by imposing additional constraints such as spatial connectivity and threshold region size. The refined methods are termed "spatially-constrained Louvain (ScLouvain)" and "spatially-constrained Leiden (ScLeiden)" algorithms. The service areas derived by the methods, termed "Cancer Service Areas (CSAs)", are far more favorable than the commonly-used comparable unit, Hospital Referral Regions (HRRs), from the Dartmouth Atlas Project. Between the two, the ScLeiden method outperforms the ScLouvain method in more favorable modularity and localization index, better balance in region size and more computational efficiency. In our case study, the ScLeiden method also eliminates poorly-connected CSAs after only three iterations. We plan to adopt the ScLeiden method for a future larger scale project of CSAs delineation in the U.S. While the methods are applied in the analysis of health care markets, they are proven as an effective and efficient approach for defining functional regions.

The two methods have some limitations. Like other heuristic computational methods, both the ScLouvain and ScLeiden algorithms introduce some uncertainty when the order of nodes being moved to a community is random. Neither algorithm can guarantee the convergence to a unique solution every time. One suggestion is to run the programs multiple times to yield a number of results, from which a best solution can be chosen. Our experience indicates that the solutions are largely consistent. In addition, while the three rules proposed to ensure spatial contiguity are all reasonable on some justifiable grounds, there is room to debate. More applications in various fields[2], specially those spatial networks, are likely to identify additional limitations, propose adequate solutions and further refine the methods. Finally, future work will consider incorporating the gravity model in the modularity measure as proposed by Expert et al. (2011) and refined by Gao et al. (2013) to examine whether such a modification improves the performance of our methods.

Acknowledgement:

Financial support from the National Cancer Institute (NCI), National Institutes of Health, under Grant R21CA212687, is gratefully acknowledged. Points of view or opinions in this article are those of the authors, and do not necessarily represent the official position or policies of NCI.

References

Bae S-H, Halperin D, West JD, Rosvall M, Howe B, 2017. Scalable and efficient flow-based community detection for large-scale graph analysis. ACM Transactions on Knowledge Discovery from Data11, 1–30.

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E, 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment2008, P10008.

Chen Y, Xu J, Xu M, 2015. Finding community structure in spatially constrained complex networks. International Journal of Geographical Information Science29, 889–911.

Expert P, Evans TS, Blondel VD, Lambiotte R, 2011. Uncovering space-independent communities in spatial networks. Proceedings of the National Academy of Sciences108, 7663–7668.

Fortunato S, Barthélemy M, 2007. Resolution limit in community detection. Proceedings of the National Academy of Sciences104, 36–41.

Gao S, Liu Y, Wang Y, & Ma X 2013. Discovering spatial interaction communities from mobile phone data. Transactions in GIS, 17(3), 463–481.

Guo D, 2008. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). International Journal of Geographical Information Science22, 801–823.

Guo D, Jin H, Gao P, Zhu X, 2018. Detecting spatial community structure in movements. International Journal of Geographical Information Science32, 1326–1347.

[2]Based on the citation data dated Oct 1, 2020, the Louvain method by Blondel et al. (2008) has been cited 12,090 times, and the Leiden method by Traag et al. (2019) has already been cited 235 times. Their enormous popularity ensures many more applications to come.

Hu Y, Wang F, Xierali IM, 2018. Automated delineation of hospital service areas and hospital referral regions by modularity optimization. Health Services Research53, 236–255. [PubMed: 27861822]

Jia P, Xierali IM, Wang F, 2015. Evaluating and re-demarcating the hospital service areas in Florida. Applied geography60, 248–253.

Jia T, Yu X, Shi W, Liu X, Li X, Xu Y, 2019. Detecting the regional delineation from a network of social media user interactions with spatial constraint: A case study of Shenzhen, China. Physica A: Statistical Mechanics and its Applications531, 121719.

Kilaru AS, Wiebe DJ, Karp DN, Love J, Kallan MJ, Carr BG, 2015. Do hospital service areas and hospital referral regions define discrete health care populations?Medical Care53, 510–516. [PubMed: 25961661]

Klauss G, Staub L, Widmer M, Busato A, 2005. Hospital service areas - A new tool for health care planning in Switzerland. BMC Health Services Research5, 15. [PubMed: 15720709]

Krings GB, Vincent D., 2011. An upper bound on community size in scalable community detection. arXiv preprint arXiv.

Kumpula JM, Saramäki J, Kaski K, Kertész J, 2007. Limited resolution in complex network community detection with Potts model approach. The European Physical Journal B56, 41–45.

Mu L, Wang F, 2008. A scale-space clustering method: mitigating the effect of scale in the analysis of zone-based data. Annals of the Association of American Geographers98, 85–101.

Newman MEJ, 2004a. Analysis of weighted networks. Physical Review E70, 056131.

Newman MEJ, 2004b. Fast algorithm for detecting community structure in networks. Physical Review E69, 066133.

Newman MEJ, Girvan M, 2004. Finding and evaluating community structure in networks. Physical Review E69, 026113.

Ozaki N, Tezuka H, Inaba M, 2016. A simple acceleration method for the Louvain algorithm. International Journal of Computer and Electrical Engineering8.

Reichardt J, Bornholdt S, 2006. Statistical mechanics of community detection. Physical Review E74, 016110.

Traag VA, 2015. Faster unfolding of communities: Speeding up the Louvain algorithm. Physical Review E92, 032801.

Traag VA, Van Dooren P, Nesterov Y, 2011. Narrow scope for resolution-limit-free community detection. Physical Review E84, 016114.

Traag VA, Waltman L, van Eck NJ, 2019. From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports9, 5233. [PubMed: 30914743]

Wang F, 2015. Quantitative methods and socio-economic applications in GIS, Second edition. ed. CRC Press.

Wang F, Wang C, Hu Y, Weiss J, Alford-Teaster J, Onega T. 2020. Automated delineation of cancer service areas in northeast region of the United States: a network optimization approach. Spatial and Spatio-temporal Epidemiology33:100338. [PubMed: 32370938]

Waltman L, Van Eck NJ, 2013. A smart local moving algorithm for large-scale modularity-based community detection. The European Physical Journal B86, 471.

Wenberg JE, Cooper MM, 1998. The Dartmouth Atlas of health care in the United States. American Hospital Publishing, Chicago, IL.

Zhang Y, Baik Seo H, Mark FA, Baicker K, 2012. Comparing local and regional variation in health care spending. The New England Journal of Medicine367, 1724–1731. [PubMed: 23113483]

Zuckerman SP, Waidmann TP, Berenson RMD, Hadley JP, 2010. Clarifying sources of geographic differences in Medicare spending. The New England Journal of Medicine363, 54–62. [PubMed: 20463333]

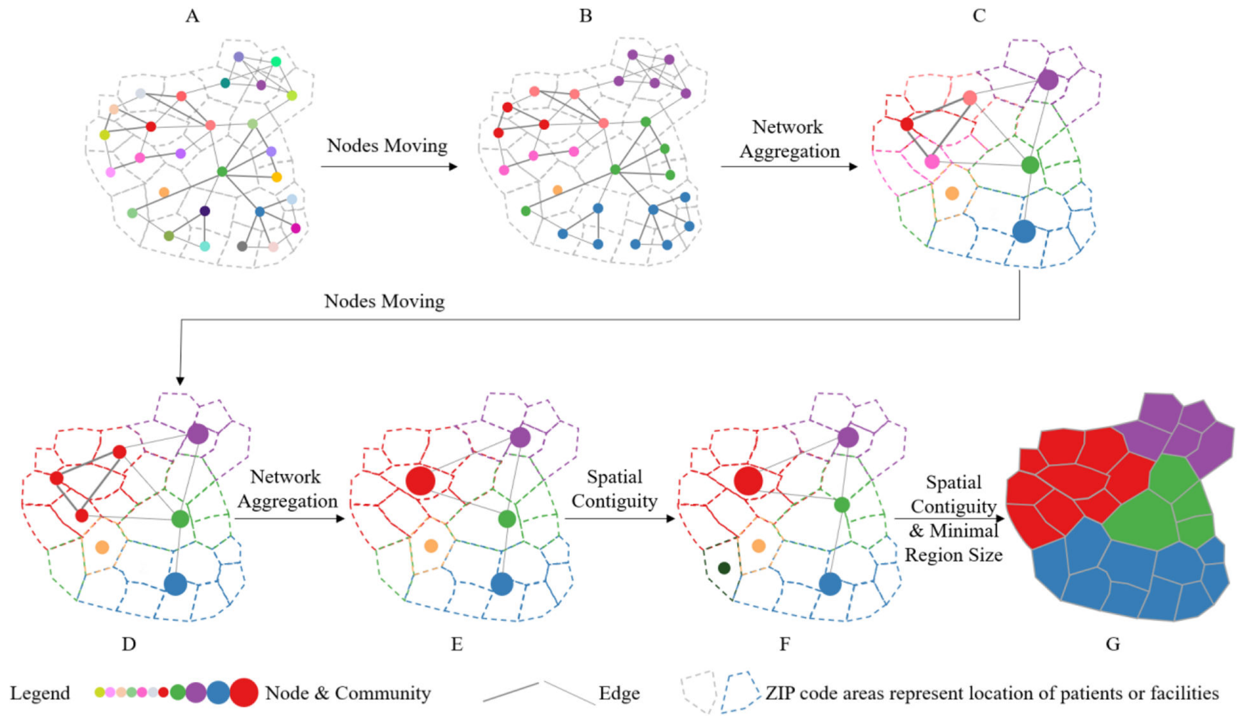

Figure 1.
Schematic illustration of ScLouvain method

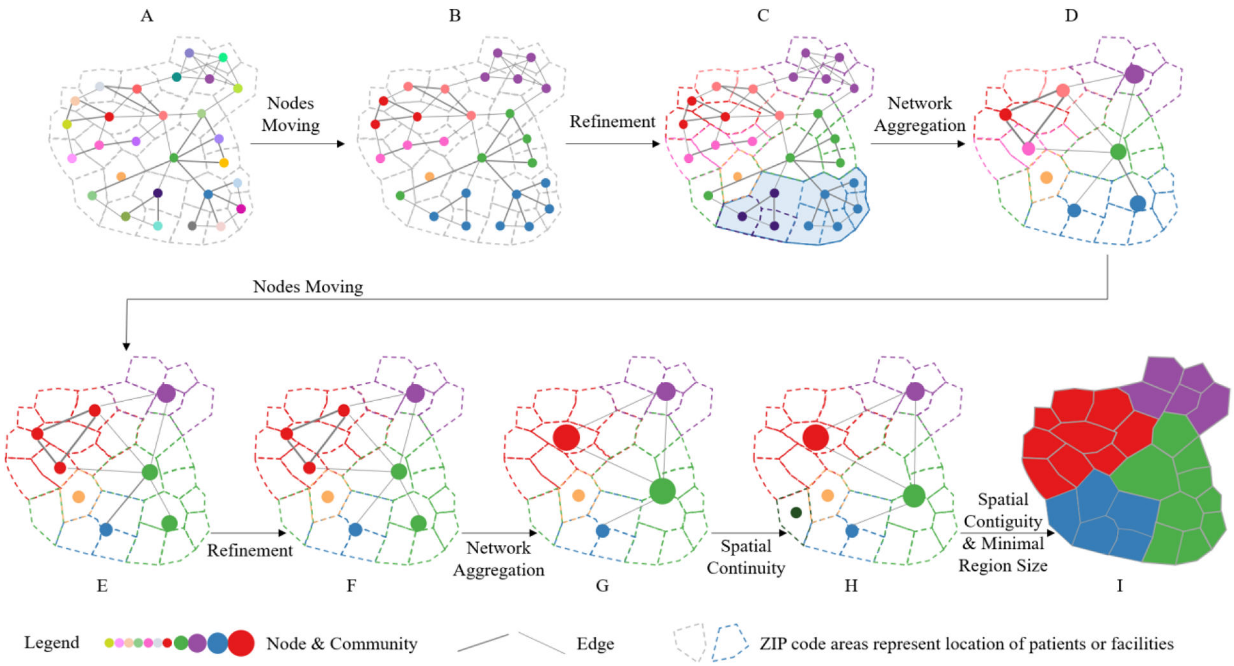

Figure 2.
Schematic illustration of ScLeiden method

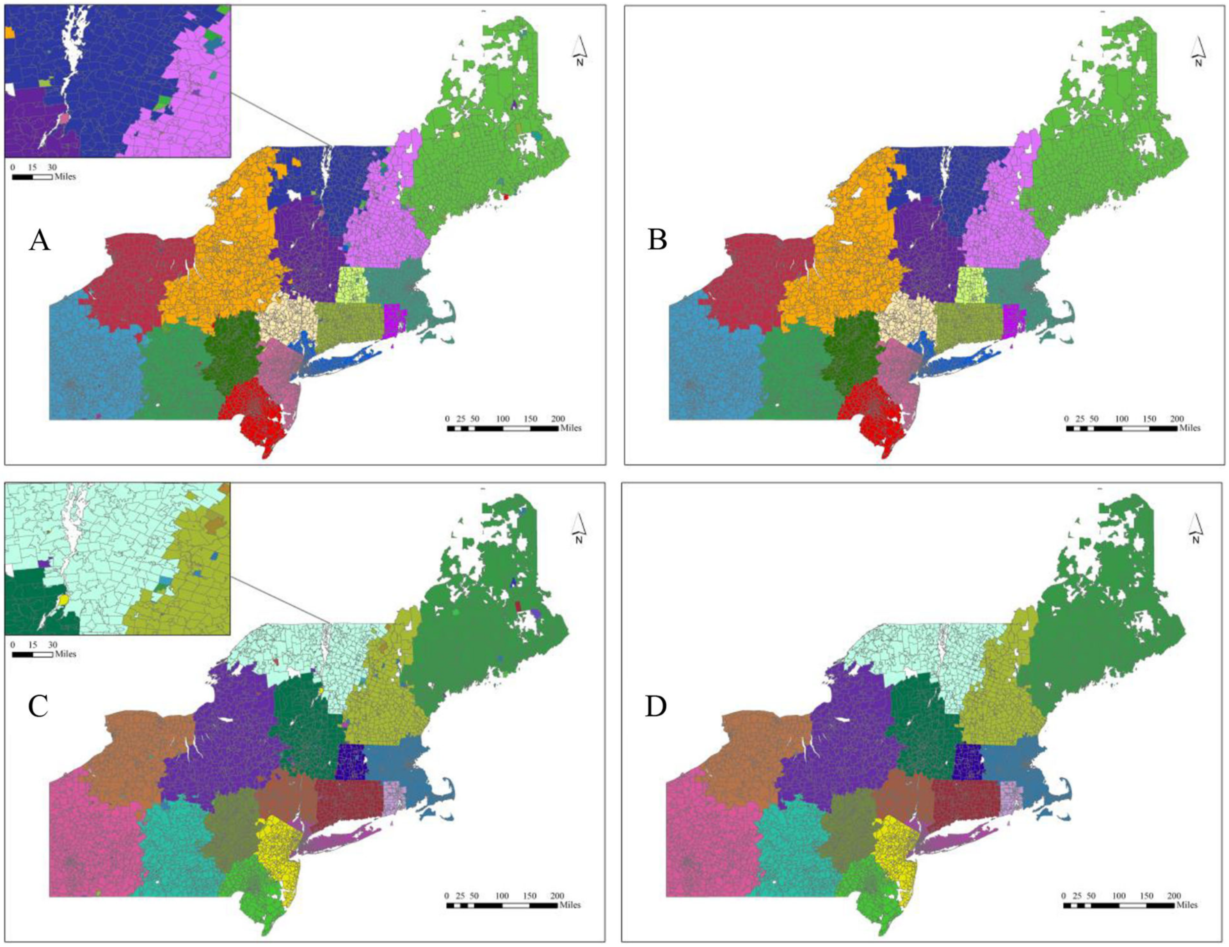

Figure 3.
17 CSAs derived by (A) Louvain, (B) ScLouvain, (C) Leiden, and (D) ScLeiden algorithms

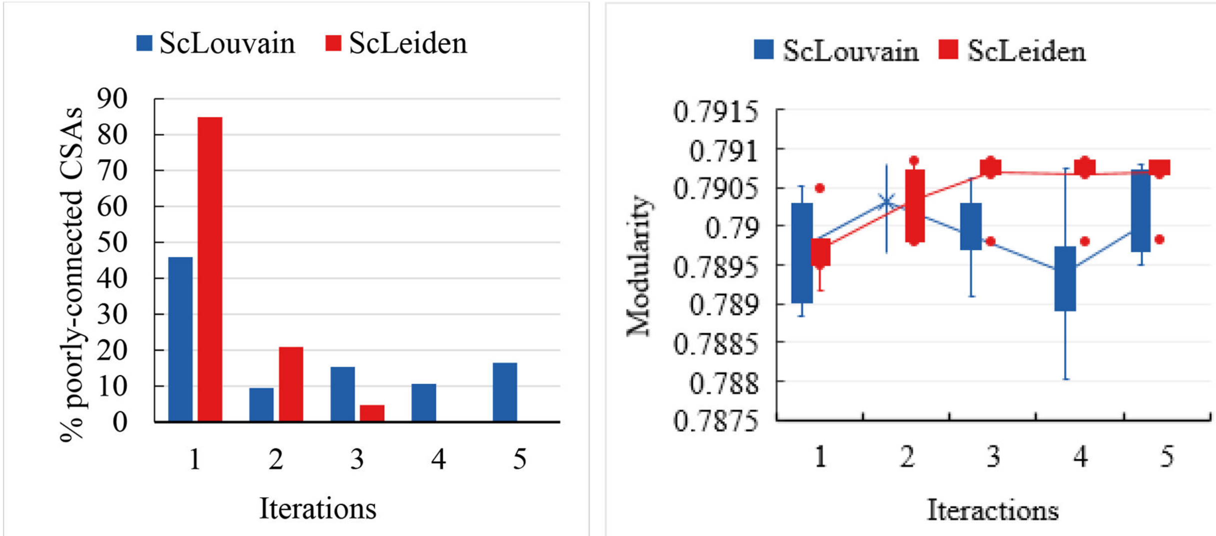

Figure 4.
Comparing the ScLouvain and ScLeiden methods: (A) percentage of poorly-connected CSAs vs. iterations, and (B) modularity vs. iterations

Author Manuscript
Author Manuscript
Author Manuscript

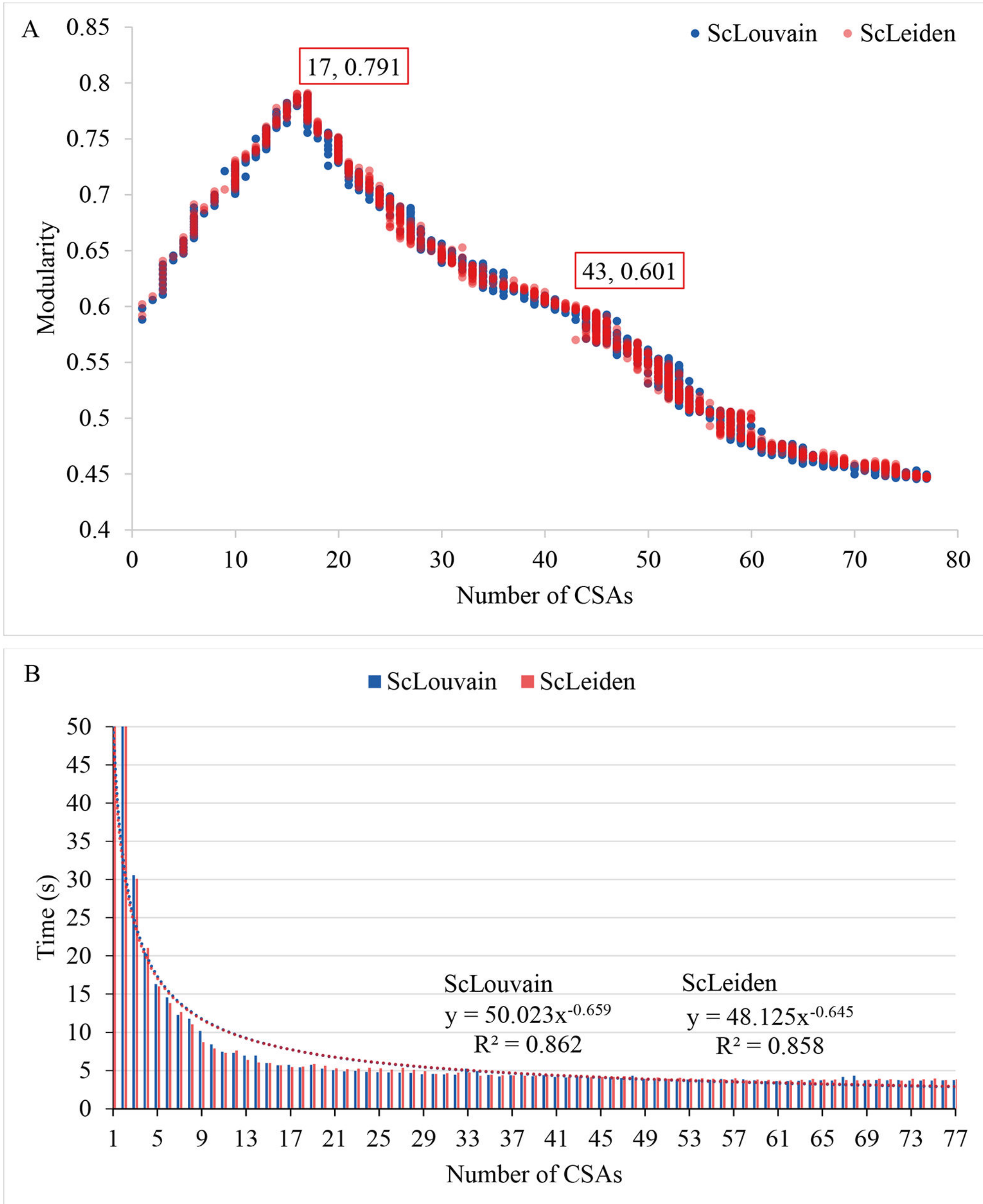

Figure 5.

Comparing the ScLouvain and ScLeiden methods: (A) modularity vs. No. CSAs, (B) computational time vs. No. CSAs

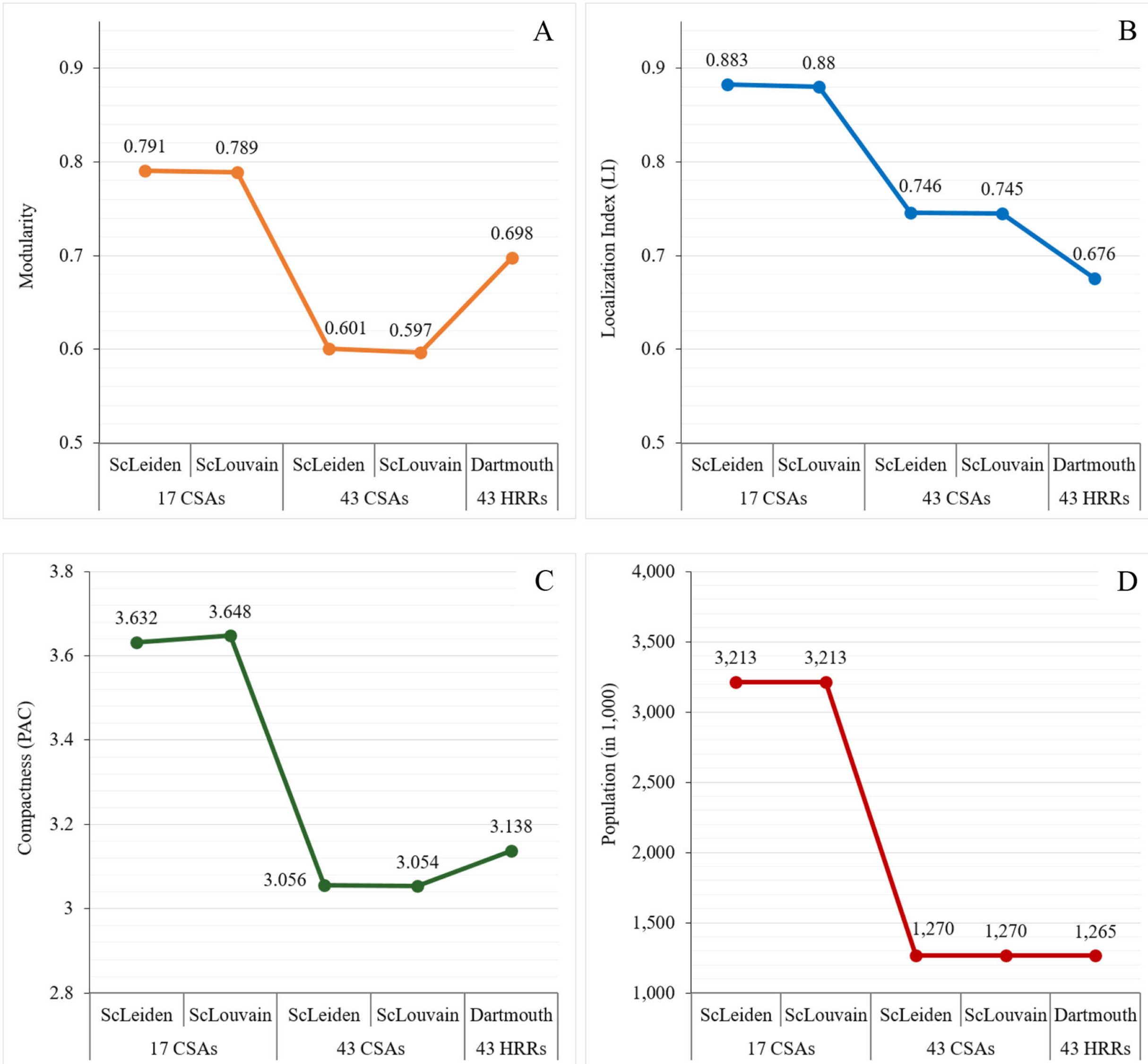

Figure 6.

Mean values of (A) Modularity, (B) LI, (C) Compactness, and (D) Population across CSAs and HRRs.

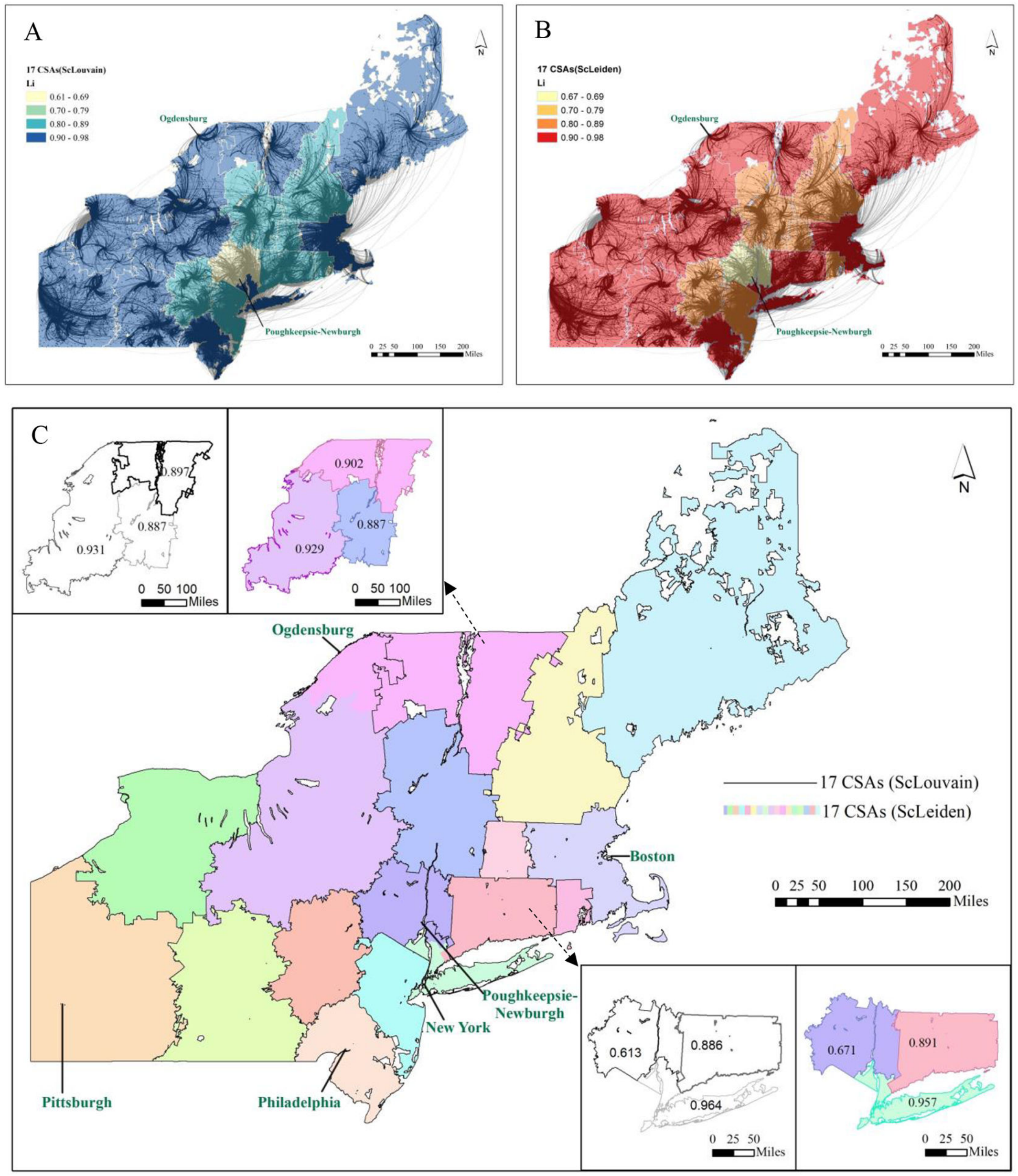

Figure 7.

(A) LIs of 17 ScLouvain-derived CSAs overlaid service flows, (B) LIs of 17 ScLeiden-derived CSAs overlaid service flows, and (C) 17 CSAs by ScLouvain vs. by ScLeiden

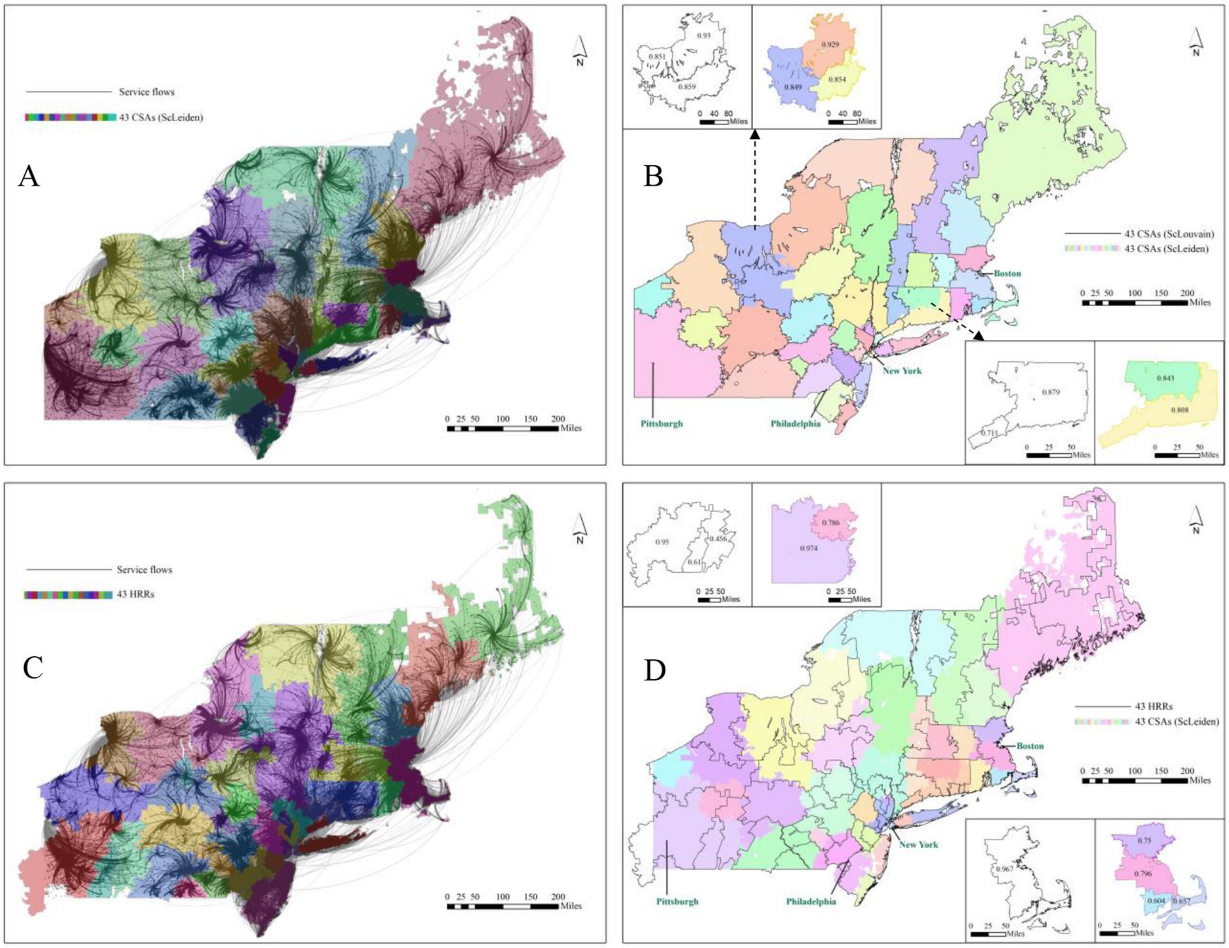

Figure 8.

(A) 43 ScLeiden-derived CSAs vs. service flows, (B) 43 ScLouvain-derived vs. ScLeiden-derived CSAs, (C) 43 HRRs vs. service flows, and (D) 43 ScLeiden-derived CSAs vs. 43 HRRs

Table 1.

Basic statistics of indices for various CSAs and HRRs in the Northeast region

Method		17 CSAs		43 CSAs		43 HRRs
		ScLeiden	ScLouvain	ScLeiden	ScLouvain	Dartmouth
Modularity		0.791	0.789	0.601	0.597	0.698
Localization Index (LI)	Min	0.671	0.613	0.407	0.411	0.185
	Max	0.978	0.978	0.974	0.974	0.967
	Mean	0.883	0.880	0.746	0.745	0.676
	S.D.	0.076	0.087	0.145	0.145	0.178
Compactness (PAC)	Min	1.887	1.868	1.56	1.56	1.536
	Max	7.700	7.596	7.379	7.379	7.834
	Mean	3.632	3.648	3.056	3.054	3.138
	S.D.	1.587	1.585	1.021	1.051	1.535
Population (in 1,000)	Min	661	544	159	159	200
	Max	11839	12049	4328	4300	4815
	Mean	3213	3213	1270	1270	1265
	S.D.	2843	2891	987	993	1213

Note: S.D. for standard deviation