RESEARCH ARTICLE

Cluster Assignment and Covariate-Constrained Randomization for Spatially-Correlated Data

Patrick Iben MS[1] | **Kathryn Colborn PhD**[2]

[1]Biostatistics and Bioinformatics, University of Colorado Anschutz Medical Campus, Aurora, CO, 80045, USA

[2]Biostatistics and Bioinformatics, University of Colorado Anschutz Medical Campus, Aurora, CO, 80045, USA

Correspondence
Patrick Iben, Biostatistics and Bioinformatics, University of Colorado Anschutz Medical Campus, Aurora, CO, 80045, USA
Email: patrick.iben@ucdenver.edu

Funding information
No funding was required for this study.

The purpose of this study was to compare the efficacy of four popular clustering algorithms - *K*-means, Clustering Large Applications (CLARA), Ward-Like Hierarchical, and Partitioning Around Medoids (PAM) - in cluster randomized trial (CRT) design based on their: resultant degree of covariate imbalance after covariate-constrained randomization (CCR) assignment of unit clusters to treatment arms (both weighted and unweighted); overall spatial heterogeneity of treatment arms after CCR; and CPU run times for each major stage of the process - cluster optimization, weighted CCR and unweighted CCR.

KEYWORDS
K-means, CLARA, Hierarchical, PAM, CRT, CCR, Malaria

1 | INTRODUCTION

Cluster randomized trials (CRTs) are commonly implemented when treatments cannot be randomized to individuals. However, randomization of clusters often results in imbalance with respect to important confounders. To circumvent this imbalance, researchers have applied constraints to the covariate make-up of the clusters so that imbalance is minimized, so-called covariate-based constrained randomization (CCR) [1] [2]. Briefly, in CCR, constraints are applied to all possible randomization schemes, and among those that satisfy these constraints one is chosen at random. In some disciplines, such as vector-borne disease research, spatial contamination of the intervention is often of concern, because individuals and pathogens can easily cross into regions receiving the opposite intervention arm. This introduces the need for an additional randomization constraint.

In this study, we compared spatial cluster algorithms for assignment of units to clusters - and clusters to treatment arms via CCR. We explored methods for first spatially clustering the units for randomization to mitigate the potential for geographic contamination before performing CCR. We also explored the use of regression weights in the CCR and

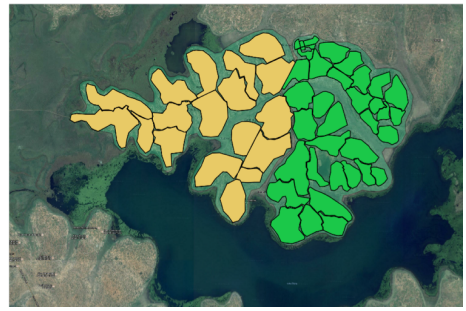

FIGURE 1 Study 1 villages. Yellow (left) indicates Kapujan County; green (right) indicates Toroma County

estimated the extent to which the defined clusters result in a randomization that suggests geographic contamination has been mitigated.

We compared these methodologies using two spatial datasets. The first is an aerial dataset of villages in a rural part of Uganda, where the objective was to randomize clusters of villages to intervention or control arms and simultaneously mitigate spatial contamination of the interventions due to mosquito and population mixing. The second is a spatial points dataset of neighborhoods in a section of New York City, where the objective was to randomize clusters of neighborhoods to intervention or control arms and mitigate spatial contamination of the interventions due to pollution. We chose these examples because they have mixed covariate data types (e.g. numerical, categorical), and the two problems are of very different size. Four popular clustering algorithms currently implemented in R were used and compared: *K*-means, Clustering Large Applications (CLARA), Ward-Like Hierarchical Clustering, and Partitioning Around Medoids (PAM). The algorithms were compared with respect to their: resultant degree of cluster covariate imbalance after CCR assignment (both weighted and unweighted); overall spatial heterogeneity of treatment arms after CCR; and CPU run times for each major stage of the process - cluster optimization, weighted CCR and unweighted CCR.

Data Scenario 1: Malaria in a Localized Uganda Community

In malaria research, interventions are often applied to communities, not individuals, such as spraying of insecticides, distribution of medications to health centers, or bed net campaigns. CCR is a potentially beneficial method for assigning treatments to communities. However, spatial heterogeneity needs to be considered, as mosquitoes and people move frequently, and the potential for treatment contamination is high. Thus, it is desirable to optimize community assignment to clusters that are as contiguous as possible, yet still achieve random assignment of clusters to intervention arms.

We present a study of two malaria interventions in Uganda where we were tasked with assigning villages (see figure 1) to one of the two interventions. It is clear from figure 1 that assignment of villages at random could introduce serious spatial contamination. Furthermore, imbalance of important confounders, such as malaria prevalence, whether or not the village borders the lake, and population density, could contaminate the intervention. Thus, our goal was to first assign villages to clusters prior to assigning these clusters to intervention arms via CCR.

The treatment group for this study was proactive screening and treatment (ProACT) of the community by trained community members (VHT). Once per week, the trained VHTs will visit each household in their village and evaluate and treat residents with malaria. Residents are also allowed to visit the VHT's home at any time in between (intermittent community case management, iCCM). In control communities, only passive case detection and treatment is exercised

(i.e., residents are allowed to visit the VHT's home at any time - iCCM).

Data Scenario 2: New York leukemia data taken from the data sets supporting Waller and Gotway 2004 [3]

To demonstrate the efficacy of the clustering and CCR methods on a larger-scale CRT scenario for spatial points, example spatial data from the "spData" package was used to develop a hypothetical CRT trial in which CCR could help study design - and where spatial contamination could be inherently present [4]. The data frame consisted of 281 census tract-level units of a region of New York. There is evidence that Trichloroethylene (TCE) exposure results from a leak into the ground soil near contamination sites, which is then released into ground water sources or evaporates into the air [5]. TCE is a volatile organic compound (VOC) that has shown some evidence of inducing testicular cancer and leukemia in rats and lymphomas and lung tumors in mice [5]. Simple carbon-based filters have proven effective at removing volatile compounds like TCE from water [6]. In 1989, NASA released their report "Interior Landscape Plants For Indoor Air Pollution Abatement" [7]. The report details the total micrograms of TCE removed by a wide variety of plants, in a sealed indoor setting during a 24-hour time period. One plant that showed particular promise for removing TCE from the air was the Gerbera daisy. Suppose that funding has become available to provide such services to the study area of interest. The hypothetical study question was: does providing the community homes with Gerbera daisy plants lead to a change in Leukemia rates? The control group was: carbon-based water filters provided to homes in the study area. The experimental group was: carbon-based water filters and Gerbera daisy plants provided to homes in the study area.

2 | METHODOLOGY

R v.3.5.3 (64-Bit) was used for all data visualization, cluster assignment, and CCR. The "sp" and "ggplot2" packages are used for spatial visualization of data [8] [9] [10]. All calculations and their respective CPU run times were derived from a PC with an Intel(R) Xeon(R) i7-6500U CPU E3-1280 v6 @ 3.90 GHz processor and 64.0GB of RAM, running Windows 10 Enterprise on an x64-based processor.

2.1 | Cluster optimization and assignment

As CRTs tend to have only two study arms for which to randomize clusters - treatment and control - only even number of clusters were considered. This ensured that an equal number of clusters were assigned to study arms. To measure covariate and geographical dissimilarity between units, the daisy() function in the "cluster" package was used [11].

To avoid the ambiguous and highly-subjective process of graphically determining the optimal clustering for a given dataset, objective, numerical cluster quality statistics were optimized. For clustering on geographical location, optimal clustering was determined by average silhouette width, via the wcClusterQuality() function from the "WeightedCluster" package [12] [13]. Average silhouette width is based on the coherence of the assignment of an observation to a given group, comparing the average weighted distance of an observation from the other members of its group and its average weighted distance from the closest group. A value is calculated for each observation, as well as an average value for each group. If the average value is low, then the groups are not clearly separated. The authors of the "WeightedCluster" package have defined average silhouette width as the following:

Let k be the group of observation i, W_k the sum of the weightings of the observations belonging to group k, w_i the weight of observation i and l one of the other groups, a_i the average weighted distance of observation i with the other members of its group and the average weighted distance from the closest group, labeled b_i. Then, the silhouette of

an observation, s_i, is computed as follows:

$$a_i = \frac{1}{W_k - 1} \sum_{j \in k} w_j d_{ij} \tag{1}$$

$$b_i = \min_l \frac{1}{W_l} \sum_{j \in l} w_j d_{ij} \tag{2}$$

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{3}$$

The mean of all s_i observations is called the *average silhouette width*.

Ward-like Hierarchical and PAM were used to cluster on both geographic location and mixed data type covariates. For Hierarchical, optimal clustering was determined by proportions of explained pseudo-inertia, via the "ClustGeo" package [14]. For PAM, optimal clustering was determined by average silhouette width, via the wcClusterQuality() function from the "WeightedCluster" package.

If only two clusters were created for randomization to treatment arms, then the number of clusters would equal the number of treatments to be assigned, and there would be no reason to perform CCR. If only four clusters were created, then there would have been a total of $\binom{4}{2} = 6$ configurations to select randomly from for CCR. If a cutoff of 10 percent was used for CCR, then this leaves only one treatment allocation scheme to select from. This creates a deterministic selection for cluster assignment - and no longer qualifies as a CRT. To remove this possibility, only even cluster sizes greater than or equal to six were considered.

To mitigate spatial contamination by creating a "buffer" between treatment units, for all but the Ward-Like Hierarchical clustering, the maximum number of clusters was determined by the closest even value such that there was an average of at least 3 units per cluster. For example: if there was 55 clusters, then the maximum cluster size tested would have been 18. The "ClustGeo" package limits the number of clusters to 55; thus, the maximum number of clusters that were tested was 54. Logically, units were not permitted to belong to multiple clusters, and for these examples, all units needed to be assigned to a cluster.

2.1.1 | *K*-means with Euclidean Distances

The K-means method is one of the most widely used to cluster data. Unfortunately, this algorithm is restricted to only numerical data, and thus cannot be used with mixed data types. Overall quality of the K-means clustering allocation scheme was determined solely on Euclidean distance between village centroids. Briefly, K-means works by first randomly assigning N data points into K disjoint subsets S_j containing N_j data points [15]. The mean distance point is computed for each pair of points in the set; points are then assigned to the cluster whose centroid is closest to that point. These two steps are alternated until within-cluster variability between data points is minimized.

Euclidean distance is defined as the following [16]:

Let $(x_1, y_1), (x_2, y_2)$ be points on a 2-D plane. Then, the Euclidean distance between them is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{4}$$

Unit-level geographic data (i.e., longitude and latitude) were extracted from spatial dataframe objects using the coordinates() function from the "sp" package. The Euclidean dissimilarity matrix was created using the daisy() function from the "cluster" package, with metric set to "euclidean".

For both studies, arrays were initialized for every cluster size value tested. For the first study, between 6 and 18 clusters were tested; for the second study, between 6 and 92.

Using the kmeans() function from the "stats" package and the wcClusterQuality() function, along with the Euclidean dissimilarity matrix, average silhouette width was iteratively collected for each cluster size with a for-loop [17]. The Hartigan and Wong (1979) algorithm, the default for the kmeans() function, was used [18]. This algorithm defines within-cluster variation as the sum-of-squared Euclidean distances between items and the corresponding centroid:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2, \tag{5}$$

where x_i is a data point belonging to the cluster C_k; μ_k the mean value of the points assigned to the cluster C_k [19].

Often, the process of optimizing cluster size for K-means entails graphical interpretation of within sum-of-squares variation [19]. For automation purposes, and to avoid this subjective approach, K-means was calculated and optimized with average silhouette width for all applicable cluster sizes. For reproducibility, a seed was set within the loop - before the clustering. After performing this sensitivity analysis to determine the optimal cluster size with the wcClusterQuality() function from the "WeightedCluster" package, and saving the value that optimized average silhouette width, the kmeans() clustering algorithm was then re-run with the optimal value. The clustering scheme was then extracted and merged back into the main study data. If several values were equal to the optimal statistic, then the value in the minimum index position in the array was selected.

2.1.2 | Clustering Large Applications (CLARA) with Manhattan Distances

To perform CLARA (Clustering LARge Applications) clustering, the clara() function from the "cluster" package was used. CLARA is performed in two major steps [12]. First, a sample is drawn from the set of objects and clustered into K subsets using the K-medoid method, giving K representative objects [20]. Then, each object not belonging to the sample is assigned to the nearest of the K representative objects. The K-medoid clustering method is a variant of K-means that uses an actual point in the cluster to represent the cluster, rather than a mean point as the center. This allows K-medoids to be less sensitive to noise and outliers than K-means. By considering sub-datasets of a fixed size, CLARA may be able to handle much larger datasets than other partitioning methods like K-means and PAM [11].

As with K-means, average silhouette width was iteratively collected for each cluster size, and cluster size was selected on maximized average silhouette width. Geographic dissimilarity was calculated using Manhattan distances, with metric set to "manhattan". Manhattan distance, also referred to as the taxicab metric, is defined as the following [21]:

Let $(x_1, y_1), (x_2, y_2)$ be points on a 2-d plane. Then, the Manhattan distance between them is:

$$d = |x_2 - x_1| + |y_2 - y_1|. \tag{6}$$

If several values were equal to the optimal statistic, then the minimum cluster size was selected.

2.1.3 | Ward-like Hierarchical Clustering with Optimized Weighting of Geographic-Covariate Dissimilarities

The "Clustgeo" package allows users to achieve Ward-like hierarchical clustering with non-Euclidean dissimilarity measures and non-uniform weights for units. This allows accommodation of mixed data types (e.g., numeric, ordinal, binary), which may often be used for spatial feature data. Non-uniform weighting may be of use when the ultimate outcome (e.g., unit prevalence of disease) may be directly proportional to some other factor (e.g., unit population). To perform the clustering, the hclustgeo() function from the "Clustgeo" package was used. Unlike the standard hclust() function, hclustgeo() can automatically accommodate non-Euclidean dissimilarities for mixed data types.

For both studies, clusters were weighted by unit population.

First, primary spatial feature data for both studies were extracted. Next, using the daisy() function from the "cluster" package, dissimilarity matrices were created for the covariates and geographical location variables separately. In addition to accommodating non-Euclidean dissimilarities, the "Clustgeo" package allows for weighting of spatial feature and geographical dissimilarity importance during cluster assignment. To test the varying weightings of dissimilarity matrices for our mixed data types, Gower dissimilarity was utilized using the daisy() function with metric set to "gower". Gower dissimilarity uses the Dice coefficient for binary data and range-normalized Manhattan distance for numeric data [22]. Let *i* and *j* be any two subjects in a dataset; $\delta_{ijk} = 1$ when character *k* can be compared for *i* and *j*, and 0 otherwise. When $\delta_{ijk} = 0$, score s_{ijk} is unknown - but is conventionally set to 0. The similarity between subjects *i* and *j* is defined as:

$$S_{ij} = \frac{\sum_{k=1}^{v} s_{ijk}}{\sum_{k=1}^{v} \delta_{ijk}}. \tag{7}$$

When $\delta_{ijk} = 0$ for all characters *k*, S_{ij} is undefined. When all comparisons are possible, $\sum_{k=1}^{v} \delta_{ijk} = v$. Scores s_{ijk} are assigned as the following:

| Dichotomous characters

Let + and - be the possible values of character *k*, and assume no unknown values of *k*. If *k* = + for *i* and *k* = + for *j*, then $s_{ijk} = 1$ and $\delta_{ijk} = 1$. If *k* = + for *i* and *k* = - for *j*, then $s_{ijk} = 0$ and $\delta_{ijk} = 1$. If *k* = - for *i* and *k* = + for *j*, then $s_{ijk} = 0$ and $\delta_{ijk} = 1$. If *k* = - for *i* and *k* = - for *j*, then $s_{ijk} = 0$ and $\delta_{ijk} = 0$.

| Qualitative characters

If *i* and *j* agree on character *k*, then $s_{ijk} = 1$; $s_{ijk} = 0$ if they differ.

| Quantitative characters

Let $x_1, x_2, ..., x_n$ be values for character k for the total sample of n cases. Then,

$$s_{ijk} = 1 - \frac{|x_i - x_j|}{R_k}$$

where R_k is the range of character k. When $x_i = x_j$, s_{ijk} = 1; when x_i and x_j are on opposite ends of range R_k, s_{ijk} is minimized (0 when determined from sample).

The weighting for dissimilarity matrix importance ranges from 0 to 1, in increments of 0.1 - referred to as alpha by developers of the package. 0 represents full weighting on spatial feature data, while 1 represents full weighting on geographical location. To determine the optimal cluster size and alpha level, the choicealpha() function from the "Clustgeo" package was used to calculate the proportion of explained pseudo-inertia for each cluster size of interest. Pseudo-inertia has been defined by the authors of the package as the following [23]:

Consider partition $P_K = (C_1, ..., C_K)$ in K clusters; w_i the weight of the i-th observation; w_j the weight of the j-th observation; d_{ij} the dissimilarity measure between observations i and j. The pseudo-inertia of a cluster C_k generalizes the inertia to the case of dissimilarity data (Euclidean or not) in the following way:

$$I(C_k) = \sum_{i \in C_k} \sum_{j \in C_k} \frac{w_i w_j}{2\mu_k} d_{ij}^2 \tag{8}$$

where $\mu_k = \sum_{i \in C_k} w_i$ is the weight of C_k. The smaller the pseudo-inertia $I(C_k)$ is, the more homogeneous are the observations belonging to the cluster C_k. The pseudo within-cluster inertia of the partition P_K is therefore:

$$W(P_K) = \sum_{k=1}^{K} I(C_k).$$

The smaller this pseudo within-inertia $W(P_K)$ is, the more homogeneous is the partition in K clusters. For Ward-like hierarchical clustering with non-Euclidean dissimilarities, the optimization problem is defined by "ClustGeo" authors in the following manner:

To obtain a new partition P_K in K clusters from a given partition P_{K+1} in $K + 1$ clusters, two clusters A and B of P_{K+1} are aggregated such that the new partition has minimum within-cluster pseudo-inertia, that is:

$$arg \min_{A,B \in P_{K+1}} W(P_K), \tag{9}$$

where $P_K = P_{K+1}\ (A, B) \bigcup (A \bigcup B)$ and

$$W(P_K) = W(P_{K+1}) - I(A) - I(B) + I(A \bigcup B).$$

Since $W(P_{K+1})$ is fixed for a given partition P_{K+1}, the optimization problem is equivalent to:

$$\min_{A,B\in P_{K+1}} I(A\bigcup B) - I(A) - I(B). \tag{10}$$

The optimization problem is therefore achieved by defining

$$\delta(A, B) := I(A\bigcup B) - I(A) - I(B)$$

as the aggregation measure between two clusters which is minimized at each step of the hierarchical clustering algorithm. $\delta(A, B) = W(P_K) - W(P_{K+1})$ can also be seen as the increase of within-cluster pseudo-inertia.

The Ward-like hierarchical clustering process for non-Euclidean dissimilarities is further defined by authors as the following:

1. Step $K = n$: the initial partition P_n in n clusters is unique.
2. Step $K = n - 1, ..., 2$: obtaining the partition in K clusters from the partition in $K + 1$ clusters. At each step K, the algorithm aggregates the two clusters A and B of P_{K+1} according to the optimization problem above such that the increase of the pseudo within-cluster inertia is minimum for the selected partition over the other ones in K clusters.
3. Step $K = 1$: stop. The partition P_1 in one cluster (containing the n observations) is obtained.

The hierarchically-nested set of all partitions is defined as $P_n, ..., P_K, ..., P_1$, where the height of a cluster $C = A \bigcup B$ is $h(C) := \delta(A, B)$.

The optimal cluster size and alpha level were determined by the value at which the difference between the explained pseudo-inertia for covariate and geographical dissimilarity was minimized. If several values were equal to the minimized difference, then the minimum cluster and alpha values were selected.

2.1.4 | Partitioning Around Medoids (PAM) with Optimized Weighting of Geographic-Covariate Dissimilarities

Like CLARA, partitioning around medoids (PAM) creates k clusters about some actual point in the center. Like the hclustgeo() function in the "ClustGeo" package, the wcKMedoids() function from the "WeightedCluster" package allows users to perform PAM clustering with non-uniform weights. For both studies, unit population was again used to weight during clustering. For use of the weights, the "PAMonce" method is utilized by the package, as defined by Reynolds et al. (2006) [24]:

First, medoid object i and non-medoid object j are selected that produce the best clustering when their roles are switched. The objective function used is the sum of the distances from each object to the closest medoid.

Starting with an empty set of medoids, objects are then added one at a time until k medoids have been selected. At each step, the new medoid is selected so as to minimize the objective function, and the the cost of each move to a neighboring solution. Rather than calculating the cost of a neighboring solution from scratch each time, only the change in cost is determined.

Consider the effect of removing object i from the set of medoids and replacing it with object h. Let the change in the cost of the solution be T_{ih}, and let the contribution of object j to this change be C_{jih}; so:

$$T_{ih} = \sum_j C_{jih}.$$

Let D_j be the distance of object j from the closest medoid, before the swap is performed. Let E_j be the distance of object j from the second closest medoid. If j is further from both i and h than from another medoid, then C_{jih} is zero, since item j will remain in the same cluster. If j is closer to i than any other medoid before the swap and $d(j,h) < E_j$, then it will be assigned to the new cluster. Hence, the contribution of object j to the swap cost is $C_{jih} = d(j,h) - d(j,i)$. If j is closer to i than any other medoid before the swap and $d(j,h) \leq E_j$, then $C_{jih} = E_j - D_j$. If j is further from i than from some other medoid, but closer to h than any, then it will be assigned to the new cluster; so: $C_{jih} = d(j,h) - D_j$. If T_{ij} is negative, then the move gives an improvement in the clustering. Although the closest medoid to each object must still be found, calculating T_{ij} in this way does reduce the amount of addition required. Further improvements can be made by taking advantage of the fact that the whole neighborhood is evaluated in each iteration of the algorithm. Neighboring solutions can be evaluated in two steps by first removing a medoid and then adding the new medoid. C_{jih} can be given as the sum of CR_{ji} – the change in cost for object j when medoid i is removed – and CA_{jh} – the change in cost for object j when medoid h is added. If j is closer to i than any other medoid, $CR_{ji} = E_j - D_j$, otherwise $CR_{ji} = 0$. After removing medoid i, let F_j be the distance of j from the closest remaining medoid. If j is closer to h than any remaining medoid, $CA_{jh} = d(j,h) - F_j$. Since the entire neighborhood is evaluated, a medoid can be removed just once before the addition of all potential alternative medoids (rather than n times for n data points), producing a further improvement in the efficiency of the algorithm.

To weight covariate and geographic dissimilarity matrices as with the "Clustgeo" package, a list of fused dissimilarity matrices was created - one for every value of alpha (0 to 1, in increments of 0.1) - from the Gower covariate and geographic dissimilarity matrices. When alpha equals 0, 100% of the covariate dissimilarity matrix is used for clustering; when alpha equals 1, 100% of the geographic dissimilarity matrix is used for clustering.

To determine optimal cluster size and mix of geographical-covariate dissimilarity, the same method used for K-means and CLARA was applied, but with the added dimension of selecting optimal alpha based on maximized average silhouette width. Next, the index position for the optimal alpha level chosen was saved. If several values matched on optimal clustering statistic, then the minimum alpha level was selected. Then, the weighted PAMonce algorithm with the wcKMedoids() function was run with the corresponding fused dissimilarity matrix, optimal cluster value, and user-defined weights to assign clusters.

2.2 | Covariate-Constrained Randomization

The cvrall() function from the "cvcrand" package was used to conduct CCR, and to compare the overall balancing of subsequent cluster assignment to treatments [25].

The pseudo-algorithm for CCR is as follows [1]:

1. Form a list of all the possible allocations. For a pair-matched design, this will have 2^m entries, where m is the number of pairs; for a completely randomized (at the group level) design, there will be $\binom{2m}{m}$ entries, where $2m$ is the total number of groups.
2. Making a pass through all entries, select those allocations that meet the specified criteria. These criteria could mean achieving some level of balance on a given set of covariates.
3. Make a matrix whose elements are the number of times, from among those allocations identified in step 2, each

pair is together.

4. Accept the constrained list of possibilities and go to step 5; or relax or tighten criteria and go to step 2; or change the stratification and go to step 1.
5. Randomly select one allocation from among the ones that have been selected as being acceptable in step 2.

The $I2$ balance metric was utilized with cvrall(), developed from Raab and Butcher (2001) [26]:

Suppose n, n_T, and n_C are the total number of clusters, the number of clusters in the treatment arm and the control arm respectively. Suppose also that there are K cluster-level variables including the continuous covariates as well as the dummy variables created from the categorical covariates. x_{ik} is the k-th covariate. $k = 1, \ldots, K$ of cluster i. $\bar{x}_{Tk} = \sum_{i=1}^{n_T} x_{ik}/n_T$ and $\bar{x}_{Ck} = \sum_{i=n_T+1}^{n} x_{ik}/n_C$ are the means of the k-th cluster-level variable in the treatment arm and the control arm, respectively; $\omega_k = 1/s_k^2 = \frac{n-1}{\sum_{i=1}^{n}(x_{ik}-\bar{x}_k)^2}$ with $\bar{x}_k = \sum_{i=1}^{n} x_{ik}/n$. Then, the $I2$ balance metric is defined as:

$$B_{(I2)} = \sum_{k=1}^{K} d_k \omega_k (\bar{x}_{Tk} - \bar{x}_{Ck})^2, \tag{11}$$

where d_k is the user-defined weight for the k-th variable. By default, $d_k = 1$ for all variables. It should be noted that this is technically a measurement of covariate *imbalance*.

For both studies, standardized regression coefficients from a Poisson cell-means model with an offset via the glm() function were used [17]. Standardized unit-level data were used for modeling; to standardize the data, the scale() function from base R was used. The regression coefficients were then standardized such that their absolute values sum to 1 - the relative weight of a given covariate is defined as the absolute value of its model coefficient divided by the sum of the absolute values of all model coefficients. The effects of these custom weights on model were compared with CCR results from un-weighted data (where $d_k = 1$ for all variables).

Size was set to 25,000,000 - if the total number of possible schemes exceeds 25,000,000, then cvrall() simulates from the complete randomization space and selects 25,000,000 unique schemes for the randomization sample space. Only the top 10 percent (in terms of covariate balance) of all possible assignment schemes were considered for random selection - the default cutoff for cvrall(). Thus, the maximum number of assignment schemes to be considered while randomizing for CCR was 2,500,000.

"cvcrand" only allows for data sets with row number equal to the number of clusters. Thus, unit-level covariate data were aggregated into cluster-level summary statistics for CCR. Both weighted and unweighted CCR were performed for the four clustering methods.

2.3 | Spatial Heterogeneity of Treatment Arms

To measure the spatial heterogeneity of treatment arms after CCR, mean nearest neighbor distances (k=1) by treatment arm were found with the nndist() function from the "spatstat" package [27]. Spatial coordinates (e.g., latitude and longitude) were used to calculate mean nearest neighbor distances based on Euclidean distance. Treatment arm mean nearest neighbor distances were then compared with Welch Two-Sample Independent t-tests with unequal variance using the t.test() function from the "stats" package [17] [28].

Let $\bar{x_A}$,$\bar{x_B}$ be the sample means for groups A and B, respectively; μ_A,μ_B the population means for groups A and B; s_A^2,s_B^2 the sample standard deviations for groups A and B; n_A,n_B the sample sizes for groups A and B; n the total

sample size. Then, the test statistic of interest is:

$$t = \frac{(\bar{x_A} - \bar{x_B}) - (\mu_A - \mu_B)}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}} \tag{12}$$

with $n - 1$ degrees of freedom. The significance level was set at $\alpha = 0.05$.

If there was no difference in mean nearest neighbor by treatment arms, then there was spatial homogeneity within the study space. This may increase spatial proximity of treatment arm units to one another. As a primary aim of this study was to mitigate spatial contamination of treatment arms, this was considered an unfavorable outcome. Thus, an ideal study design would have spatial heterogeneity of treatment arms (e.g., significantly differing mean nearest neighbor distances).

3 | RESULTS

3.1 | Case Study 1

All results can be found in Table 1. In terms of selected balance score, the top four performers were: Ward-Like Hierarchical clustering with weighted CCR (0.154); PAM with weighted CCR (0.237); CLARA with weighted CCR (0.245); K-Means with weighted CCR (0.248). In terms of the maximum (10%) cutoff score, the top four performers were: Ward-Like Hierarchical clustering with weighted CCR (0.178); CLARA with weighted CCR (0.247); K-Means with weighted CCR (0.294); PAM with weighted CCR (0.301). PAM with weighted CCR had the lowest minimum score in their randomization space (0.001), while PAM with unweighted CCR had the highest maximum score in their randomization space (70.615); it should be noted that both of these CCR tests had the highest number of clusters in each treatment arm (9), and thus the highest number of possible treatment allocations in their randomization space (4862 possible assignment schemes).

At a significance level of $\alpha = 0.05$, the only CCR allocations that resulted in significantly different mean nearest neighbor distances between treatment arms were K-Means with weighted CCR and Ward-Like Hierarchical with weighted CCR.

In terms of CPU run times, the quickest performers were K-means and CLARA (there were no discernible differences between them). Ward-Like Hierarchical clustering took the longest to run, due to its cluster optimization phase, though by only a few extra seconds. PAM clustering took more time than K-Means and CLARA and less time than Hierarchical, but negligibly so for this example.

3.2 | Study 2

All results for study 2 can be found in Table 2. In terms of selected balance score, the top four performers were: K-Means with weighted CCR (0.098); CLARA with weighted CCR (0.203); Ward-Like Hierarchical clustering with weighted CCR (0.303); K-Means with unweighted CCR (0.695). In terms of the maximum (10%) cutoff score, the top four performers were: K-Means with weighted CCR (0.098); CLARA with weighted CCR (0.203); K-Means with unweighted CCR (0.695); Ward-Like Hierarchical clustering with weighted CCR (0.753). It should be noted that the K-Means and CLARA clustering methods were optimized at K=6 clusters, and thus only had 2 allocations in the 10% cutoff space to randomly select from.

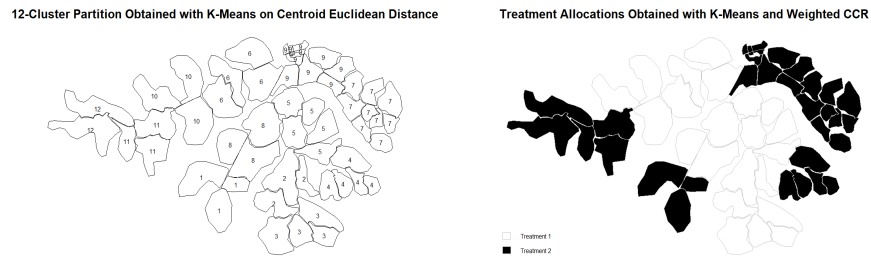

FIGURE 2 Study 1 *K*-means cluster assignments using Euclidean distance (left, cluster assignment numbers provided within units); Study 1 treatment assignments for *K*-means clusters, obtained via covariate-constrained randomization (CCR) with regression model weighting of covariates (right).

Again, PAM with weighted CCR had the lowest minimum score in their randomization space (0.000), while PAM with unweighted CCR had the highest maximum score in their randomization space (874.316); again, both of these CCR tests had the highest number of clusters in each treatment arm (92), and thus the highest number of possible treatment allocations in their randomization space (2,500,000 - the maximum allowed in the randomization space size).

At a significance level of $\alpha = 0.05$, the *K*-Means and CLARA weighted and unweighted CCR allocations resulted in significantly different mean nearest neighbor distances (k=1) between treatment arms, as well as the PAM weighted CCR allocation.

In terms of CPU run times, the quickest performer was *K*-means, followed by CLARA. Again, Ward-Like Hierarchical clustering took the longest to run, largely due to its cluster optimization phase, followed by PAM. However, scaling from 55 to 281 clusters drastically increased run times.

4 | CONCLUSIONS

Strategic weighting of covariates during CCR, regardless of clustering method, served to drastically improve covariate balance during randomization into CRT treatment arms.

For small studies, Ward-Like Hierarchical Clustering with mixed covariate-and-geographical dissimilarities and weighted CCR may be an optimal study design strategy. However, for the larger second study, *K*-Means and CLARA performed better than Hierarchical or PAM in terms of selected and cutoff scores, albeit with only two possible allocation schemes with which to randomize. The selected schemes for weighted *K*-means also maintained spatial heterogeneity of treatment assignments for both studies. Thus, *K*-means with Euclidean geographic dissimilarities and covariate weighting during CCR may be the optimal method tested for assigning spatially-correlated units to treatment arms. The selected clustering scheme and CCR treatment assignments for study 1 are displayed in Figure 2.

However, while simply using geographic dissimilarity may suffice for balancing on spatial feature data, it does not guarantee that the scheme selected at the end of CCR avoids spatial contamination of treatment arms. To do so requires additional methodology, in which only allocation schemes that result in spatial heterogeneity of treatment arms are considered during CCR.

| Limitations

For simplicity, the only cluster quality statistic used to optimize K-means, CLARA, and PAM was average silhouette width. It may be of interest to examine additional, potential cluster quality statistics that are applicable for clustering on a variety of data types (e.g., numerical, nominal, ordinal).

"ClustGeo" limits the maximum number of clusters to 55. This may be impractical for some medium-to-large CRT studies. It should also be noted that Ward-Like Hierarchical Clustering took significantly longer to optimize and perform while scaling up in study size within the constraint of 55 clusters, and thus may not be practically scale-able for some studies.

This study did not examine the effects of altering the balance score cutoff point - we simply set it to 10%. It is unclear as to whether there may or may not be an enhanced method of setting this cutoff point, and thus it deserves further investigation. As previously mentioned, with a minimum number of testable clusters set to six and a cutoff of 10%, there can be a minimum of only two randomization schemes to select from during CCR. While not technically deterministic, some CRT designers make take issue with an almost entirely deterministic randomization space.

Due to computational power limitations, this study only allowed for a maximum consideration of 25,000,000 randomization schemes with "cvcrand". It may unwise to compare clustering methods when it is not possible to have the entire sample space for consideration during CCR, as the simulated subset is not fully representative of the entire space. Ideally, future studies would compare the efficacy of clustering algorithms in CCR design without restriction on the randomization space. It should be noted that the drastically-increased CCR run times for study 2's Hierarchical and PAM clustering methods, optimized with 50 and 92 clusters, respectively, indicate that doing so with "cvcrand" may also be prohibitively time-consuming with larger studies.

Additionally, "cvcrand" utilizes only one main method of calculating imbalance for a given covariate at the treatment level - difference in treatment means, normalized with overall standard deviation. It may be of interest to allow users to select their own summary statistics to compare and normalize treatment imbalances (e.g., median comparison and range normalization).

In both studies, PAM yielded the lowest minimum and the highest maximum CCR scores in the randomization space. However, for both studies, PAM was optimized for the greatest number of clusters as well. Thus, it appears as if minimum and maximum CCR scores may be influenced by number of clusters to be randomized into treatment arms. It may be unwise to interpret these scores.

Clustering and treatment assignments, as well as conclusions on spatial heterogeneity of treatment assignments, may be influenced by geographical barriers between units (e.g., mountains, canyons, swamps). These barriers may prevent spatial contamination of subjects across units, but they may not be relevant for a spatially-dependent outcome in which these barriers may not have practical meaning (e.g., a malaria-infected mosquito flying across swamps).

Acknowledgments

Acknowledgments to Drs. Katerina Kechris and Farnoush Banaei-Kashani for their generous support and guidance during the research process.

Conflict of Interest

There were no conflicts of interest with this study.

references

[1] Moulton LH. Covariate-based constrained randomization of group-randomized trials. Clinical Trials 2004;1(3):297–305. https://doi.org/10.1191/1740774504cn024oa, pMID: 16279255.

[2] Dickinson LM, Beaty B, Fox C, Pace W, Dickinson WP, Emsermann C, et al. Pragmatic Cluster Randomized Trials Using Covariate Constrained Randomization: A Method for Practice-based Research Networks (PBRNs). The Journal of the American Board of Family Medicine 2015;28(5):663–672. https://www.jabfm.org/content/28/5/663.

[3] Waller L, A Gotway C. Applied Spatial Statistics for Public Health Data. Applied spatial statistics for public health data 2004 01;.

[4] Bivand R, Nowosad J, Lovelace R, spData: Datasets for Spatial Analysis; 2019. Dataset retrieved from "spData" package version 0.3.0, https://CRAN.R-project.org/package=spData.

[5] Toxic Substances Portal - Trichloroethylene (TCE). Centers for Disease Control and Prevention; 2015. https://www.atsdr.cdc.gov/phs/phs.asp?id=171&tid=30.

[6] 'A Civil Action' Carcinogen Pollutes Tap Water Supplies for 14 Million Americans; 2018. https://www.ewg.org/childrenshealth/carcinogen-pollutes-tap-water-supplies-14-million-americans/.

[7] Wolverton JA B C, Bounds K. Interior Landscape Plants for Indoor Air Pollution Abatement, Final Report, September N.A.S.A. 1989 Stennis Space Centre MS. Water, Air, and Soil Pollution 1989 September;https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930073077.pdf.

[8] Pebesma EJ, Bivand RS. Classes and methods for spatial data in R. R News 2005 November;5(2):9–13. https://CRAN.R-project.org/doc/Rnews/.

[9] Bivand RS, Pebesma E, Gomez-Rubio V. Applied spatial data analysis with R, Second edition. Springer, NY; 2013. http://www.asdar-book.org/.

[10] Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2016. http://ggplot2.org.

[11] Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K. cluster: Cluster Analysis Basics and Extensions; 2018, r package version 2.0.7-1 — For new features, see the 'Changelog' file (in the package source).

[12] Kaufman L, Rousseeuw P. Finding Groups in Data: An Introduction To Cluster Analysis; 1990.

[13] Studer M. WeightedCluster Library Manual: A practical guide to creating typologies of trajectories in the social sciences with R. LIVES Working Papers 24; 2013.

[14] Chavent M, Kuentz V, Labenne A, Saracco J. ClustGeo: Hierarchical Clustering with Spatial Constraints; 2017, https://CRAN.R-project.org/package=ClustGeo, r package version 2.0.

[15] Bishop CM. Neural Networks for Pattern Recognition. New York, NY, USA: Oxford University Press, Inc.; 1995.

[16] Distance;. http://mathworld.wolfram.com/Distance.html.

[17] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria; 2018, https://www.R-project.org/.

[18] Hartigan JA, Wong MA. Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics) 1979;28(1):100–108. http://www.jstor.org/stable/2346830.

[19] K-means Cluster Analysis;. https://uc-r.github.io/kmeans_clustering#kmeans.

[20] Jin X, Han J. In: Sammut C, Webb GI, editors. K-Medoids Clustering Boston, MA: Springer US; 2010. p. 564–565. https://doi.org/10.1007/978-0-387-30164-8_426.

[21] Krause EF. Taxicab geometry an adventure in non-Euclidean geometry. Dover Publ.; 1986.

[22] Gower JC. A General Coefficient of Similarity and Some of Its Properties. Biometrics 1971;27(4):857–871. http://www.jstor.org/stable/2528823.

[23] Chavent M, Kuentz-Simonet V, Labenne A, Saracco J. ClustGeo: an R package for hierarchical clustering with spatial constraints. Computational Statistics 2018;33(4):1799–1822. https://link.springer.com/content/pdf/10.1007/s00180-018-0791-1.pdf.

[24] Reynolds AP, Richards G, de la Iglesia B, Rayward-Smith VJ. Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms. Journal of Mathematical Modelling and Algorithms 2006 Dec;5(4):475–504. https://doi.org/10.1007/s10852-005-9022-1.

[25] Yu H, Li F, Gallis JA, Turner EL. cvcrand: Efficient Design and Analysis of Cluster Randomized Trials; 2019, https://CRAN.R-project.org/package=cvcrand, r package version 0.0.3.

[26] Raab GM, Butcher I. Balance in cluster randomized trials. Statistics in Medicine 2001;20(3):351–365.

[27] Baddeley A, Turner R. spatstat: An R Package for Analyzing Spatial Point Patterns. Journal of Statistical Software 2005;12(6):1–42. http://www.jstatsoft.org/v12/i06/.

[28] Elizabeth A Albright P, Two Independent Samples Unequal Variance (Welch's Test). Duke; 2018. https://sites.nicholas.duke.edu/statsreview/means/welch/.

5 | APPENDIX

Source code has been made available to the public at https://github.com/patrick-iben/Cluster-Optimization-and-CCR-for-Spatially-Correlated-Data.

TABLE 1 Study 1 CCR Results and CPU Run Times (55 Units).

Method	Minimum Score	Selected Balance Score	10% Cutoff Score	Maximum Score
K-Means Weighted	0.105	0.248	0.294	4.854
K-Means Unweighted	1.389	3.403	3.824	37.177
CLARA Weighted	0.162	0.245	0.247	2.470
CLARA Unweighted	2.095	2.982	3.229	13.661
H-Clust Weighted	0.089	0.154	0.178	5.727
H-Clust Unweighted	0.956	1.618	1.853	43.003
PAM Weighted	0.001	0.237	0.301	12.920
PAM Unweighted	0.012	2.367	2.950	70.615

Method	N Clusters in Each Treatment Arm	N Allocations in Randomization Space
K-Means Weighted	6	92
K-Means Unweighted	6	92
CLARA Weighted	4	7
CLARA Unweighted	4	7
H-Clust Weighted	6	92
H-Clust Unweighted	6	92
PAM Weighted	9	4862
PAM Unweighted	9	4862

Method	Clustering of treatment arms present in selected scheme (α = 0.05)?
K-Means Weighted	Yes
K-Means Unweighted	No
CLARA Weighted	No
CLARA Unweighted	No
H-Clust Weighted	Yes
H-Clust Unweighted	No
PAM Weighted	No
PAM Unweighted	No

Method	Cluster Optimization Time (Seconds)	Weighted CCR Time (Seconds)	Unweighted CCR Time (Seconds)
K-Means	0.073	0.063	0.032
CLARA	0.073	0.016	0.016
H-Clust	5.206	0.016	0.016
PAM	0.232	0.197	0.200

CCR, Covariate-Constrained Randomization; CLARA, Clustering Large Applications; H-Clust, Ward-Like Hierarchical Clustering; PAM, Partitioning Around Medoids.

TABLE 2 Study 2 CCR Results and CPU Run Times (281 Units).

Method	Minimum Score	Selected Balance Score	10% Cutoff Score	Maximum Score
K-Means Weighted	0.098	0.098	0.098	1.192
K-Means Unweighted	0.695	0.695	0.695	8.222
CLARA Weighted	0.203	0.203	0.203	0.899
CLARA Unweighted	2.096	2.096	2.096	6.191
H-Clust Weighted	0.145	0.303	0.753	68.817
H-Clust Unweighted	0.980	2.795	7.872	461.911
PAM Weighted	0.000	0.708	1.397	108.048
PAM Unweighted	0.001	6.779	13.247	874.316

Method	N Clusters in Each Treatment Arm	N Allocations in Randomization Space
K-Means Weighted	3	2
K-Means Unweighted	3	2
CLARA Weighted	3	2
CLARA Unweighted	3	2
H-Clust Weighted	25	2500000
H-Clust Unweighted	25	2500000
PAM Weighted	46	2500000
PAM Unweighted	46	2500000

Method	Clustering of treatment arms present in selected scheme (α = 0.05)?
K-Means Weighted	Yes
K-Means Unweighted	Yes
CLARA Weighted	Yes
CLARA Unweighted	Yes
H-Clust Weighted	No
H-Clust Unweighted	No
PAM Weighted	Yes
PAM Unweighted	No

Method	Cluster Optimization Time (Seconds)	Weighted CCR Time (Seconds)	Unweighted CCR Time (Seconds)
K-Means	1.395	0.016	0.040
CLARA	11.937	0.019	0.071
H-Clust	620.302	1205.195	1230.907
PAM	68.336	1370.509	1239.719

CCR, Covariate-Constrained Randomization; CLARA, Clustering Large Applications; H-Clust, Ward-Like Hierarchical Clustering; PAM, Partitioning Around Medoids.