LocalControl: An **R** Package for Comparative Safety and Effectiveness Research

Nicolas R. Lauve
University of New Mexico

Stuart J. Nelson
University of New Mexico

S. Stanley Young
CGStat, LLC

Robert L. Obenchain
Risk Benefit Statistics, LLC

Christophe G. Lambert
University of New Mexico

Abstract

The **LocalControl** R package implements novel approaches to address biases and confounding when comparing treatments or exposures in observational studies of outcomes. While designed and appropriate for use in comparative safety and effectiveness research involving medicine and the life sciences, the package can be used in other situations involving outcomes with multiple confounders. **LocalControl** is an open-source tool for researchers whose aim is to generate high quality evidence using observational data. The package implements a family of methods for non-parametric bias correction when comparing treatments in observational studies, including survival analysis settings, where competing risks and/or censoring may be present. The approach extends to bias-corrected personalized predictions of treatment outcome differences, and analysis of heterogeneity of treatment effect-sizes across patient subgroups.

Keywords: bias, R, survival, Kaplan-Meier, competing risks.

1. Introduction

Envision a day when high-quality comparative safety and effectiveness research is performed, scrutinized, and updated within a culture of reproducibility, then deployed at point-of-care to improve patient outcomes. While the gold standard of evidence is considered to be randomized controlled trials, such trials have limitations. Randomized controlled trials can approach many questions, using randomization and subject selection criteria to reduce the likelihood of confounders affecting study results, but such studies are expensive, limited in generalizability by their exclusions, and provide little information about long-term outcomes due to short

duration. The advent of large observational data sets has created new opportunities to generate comparative safety and effectiveness evidence that would not be feasible with randomized trials. While biases and confounders can create major challenges in making robust treatment comparisons with observational data, we suggest ways to mitigate these issues.

The traditional approach to addressing biases in observational studies is to model confounder effects as covariates in linear models. While widely accepted and useful, regression methods have difficulty modeling nonlinearity, have convergence problems when analyzing correlated covariates, and are problematic when multiple mechanisms drive the outcome. Propensity scoring approaches have gained wide use in correcting treatment biases (Rosenbaum and Rubin 1983, 1985). On average, they outperform alternative methods, including regression in large scale patient records analyses (Stang *et al.* 2010; Ryan, Madigan, Stang, Overhage, Racoosin, and Hartzema 2012). A weakness of propensity scoring is that there is no assurance of patient similarity on confounders. Patients being compared simply have similar probability of treatment (Iacus, King, and Porro 2012). Thus, if an elderly female has the same propensity for treatment as a young male, they might be grouped for comparison, even though this makes very little biological sense.

Often it is more appropriate in observational studies to employ survival analysis to model time-to-events of interest (Kaplan and Meier 1958). While visually intuitive, Kaplan-Meier curves do not address biases. Methods that do, include linear survival models like Cox regression (Cox 1972), and competing risks regression (Gray 1988; Fine and Gray 1999). In recent years, propensity scoring has also been extended to a survival framework and evaluated (Gayat, Resche-Rigon, Mary, and Porcher 2012; Austin 2014; Austin and Schuster 2016). However, parametric methods such as Cox and competing risk regression suffer from the limitations of linear models, as well as making the assumption of proportional hazards; propensity scoring has the same weakness in survival settings as those described earlier.

The local control method (Obenchain 2012, 2010; Obenchain and Young 2013; Lopiano, Obenchain, and Young 2014) provides a powerful and conceptually intuitive approach to adjustment for biases and confounders in large-scale observational data. Local control can overcome the limitations above. It enables the non-parametric estimation of overall treatment effects and provides a framework for investigating heterogeneity of treatment effects across subpopulations. Local control has been successfully used to compare treatments for major depressive disorder (Obenchain and Young 2013; Faries, Chen, Lipkovich, Zagar, Liu, and Obenchain 2013), and to evaluate the effect of air quality on mortality (Young, Obenchain, and Lambert 2015, 2016). However, until this publication, local control methodology did not support survival analysis.

This article introduces the R (R Core Team 2020) package **LocalControl** (Lauve, Nelson, Young, Obenchain, and Lambert 2020) which implements novel approaches to address biases and confounding when comparing treatments or exposures in observational studies. The key idea behind local control is to form many homogeneous clusters of observations within which one can compare alternate treatments, statistically correcting for measured biases and confounders, analogous to a randomized block design within randomized controlled trials (Student 1911; Fisher 1992; Addelman 1969). The **LocalControl** package implements:

- `LocalControlClassic()`: The local control approach was originally introduced by Robert Obenchain. `LocalControlClassic()` compares treatment outcomes in observational data, employing hierarchical clustering on confounders to reveal and correct

for treatment selection bias. Local treatment effect-size estimates from individual clusters are bias-corrected estimates of the difference in expected outcomes between two treatments. This function provides no support for survival analysis.

- `LocalControl()`: New forms of local control adjustment for observational studies, including those modeled through survival analyses, are introduced here. Rather than using hierarchical clustering without replacement, these new adjustments match observations with all neighboring points that fall within a radius of similarity in covariate space. Selecting neighbors with replacement means that some observations may reside within multiple clusters. With `LocalControl()`, each observation becomes the centroid of its own neighborhood of similar observations, maximizing informative samples. The `outcomeType` parameter allows us to extend this methodology to enable bias-corrected comparison of treatments in survival/time-to-event settings, including support for competing risks analysis.

Local control enables the comparison of outcomes for two different treatments. The variants of local control included with this package can analyze both real-valued outcomes, as well as time-to-event data. The survival-based local control can create bias-corrected Kaplan-Meier curves, as well as competing risk estimates of cumulative incidence, along with corresponding estimates of the confidence intervals. In the remainder of this paper, the methodology behind the functions listed above is described, along with one or more examples for each. The classic local control, developed by Obenchain is described in Section 2. In Section 3 extensions are introduced which are necessary to make the transition from the classic implementation, to the nearest-neighbor variant. Section 4 describes how local control is adapted to support survival-based treatment comparisons. Section 5 contains a detailed case study using local control to examine the effects of smoking on the competing risks of death and hypertension in patients from the Framingham Heart Study. Section 6 discusses bias-corrected subgroup analysis to address questions of heterogeneity of treatment effect. The data required to perform all of the following examples, and case study, are included with the **LocalControl** package. The package, along with further documentation and instruction, can be found on the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=LocalControl.

2. Classic local control

2.1. Methodology

Local control analysis concepts were originally introduced to the R community in 2005, as a suite of functions in the package, "Unsupervised and Supervised Propensity Scoring in R", or **USPS** (Obenchain 2012). Local control is an unsupervised non-parametric approach to adjust for bias in confounder space when comparing a pair of alternative treatments (Obenchain 2010; Obenchain and Young 2013). Local control focuses on making "fair" comparisons (Lopiano *et al.* 2014) using experimental units with confounding characteristics that are as well-matched as possible. Furthermore, local control does not restrict the attention to only treatment "main-effect" comparisons; distributions of local effect-sizes are estimated and displayed. Rather than estimating treatment main-effects, local control estimates local effect-sizes within subgroups (clusters) of relatively well-matched observations.

Outcome	Treatment	X_1	X_2	...	X_k
9.6	A	red	0	...	98.6
3.4	B	green	1	...	99.2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
2.8	A	blue	0	...	86.4

Table 1: Non-survival data format. A data frame where one column contains a numerical observed outcome. The treatment column contains a discrete variable with two unique values. At least one of the remaining columns contains pre-treatment covariates used for grouping similar observations.

Local control uses clustering of observations in much the same way that design-of-experiments uses blocking (Box, Hunter, and Hunter 2005). Clustering hierarchies are built using confounders believed to be sources of treatment selection bias. `LocalControlClassic()` allows users to select between different algorithms for clustering observations. Choice of clustering algorithm can affect both runtime and the properties of the resulting clusters. After forming a similarity hierarchy, the clustering tree is cut, dividing observations into small or large subgroups, depending on the location of the cut. After cutting the tree, each of the resulting clusters is tested for informativeness. Informative clusters contain at least one member of both treatment groups. The local treatment difference (LTD) is the difference in average outcome between the two treatment groups within a cluster.

One of the major exploratory features of local control is its interest in controlling the level of similarity within clusters. This is done within `LocalControlClassic()` by adjusting the height at which the hierarchical clustering trees are cut. Cutting towards the top of a tree results in a small number of large clusters, each containing a relatively broad distribution of confounder values. Cutting towards the bottom of the tree results in more clusters that are smaller in the sense of containing only patients with more similar confounder values. As the number of clusters increases and the size of clusters is reduced, the probability that a cluster will not contain observations from both treatment groups, and hence be uninformative, also increases. This represents a variance-bias trade-off where variability definitely increases as bias is possibly reduced. In the remainder of Section 2, `LocalControlClassic()` is applied to the cardiology data set analyzed in Kereiakes *et al.* (2000).

2.2. Non-survival data format

Each of the local control functions described in this paper requires that users provide valid R data frames in order to execute. The data frames must exist in the user's global R environment, prior to calling any of the local control functions. While the input requirements vary slightly between survival and non-survival analyses, both forms require data frames where the rows contain individual records, while the columns correspond to various patient attributes. The two non-survival analysis functions, `LocalControlClassic()`, and `LocalControl(outcomeType = "default")`, require that the input data frame has column vectors corresponding to the following variables:

- Treatment: Factor column with two unique values indicating the treatment for each observation.

- Outcome: Discrete or continuous outcome variable which will be compared between treatment groups.

- Covariates: The baseline (pre-treatment) X-confounder variables used to determine patient similarity.

The local control functions require the outcome variable column name, `outcomeColName`, the treatment variable column name, `treatmentColName`, and a vector of one or more covariate column names, `clusterVars`. The values of the covariates may be logical, categorical, or continuous. Each of the covariate columns must have a standard deviation greater than zero and cannot contain missing values. When missing values exist in a data frame, the base R function, `complete.cases`, can be used to remove incomplete records entirely. If removal is not an option, imputation of missing values would be required. An example data set is displayed in Table 1.

The **LocalControl** package includes two data sets which adhere to the format described in Table 1, `cardSim` and `lindner`. These data sets are used to demonstrate the capabilites of **LocalControl**, starting with an analysis of `lindner` using `LocalControlClassic()`.

2.3. Example: `LocalControlClassic()`

The following example uses data from a study conducted at the Ohio Heart Health Center in 1997, known as the Lindner study (Kereiakes *et al.* 2000). The study examines post-procedure effects of the treatment, Abciximab, a glycoprotein IIb/IIIa receptor antagonist, plus usual care compared with outcomes from patients who received usual care alone. The data contain two possible outcome measures: a binary estimate of life years preserved, and the total cardiac-related cost incurred in the twelve months following treatment.

Data: `lindner`

Variables in the `lindner` data:

- `lifepres`: Life years preserved post treatment: 0 (died within 1 year) vs. 11.6 (survived at least 1 year).

- `cardbill`: Cardiac related billing within 12 months.

- `abcix`: Did the patient receive Abciximab augmentation of usual care? 1 = yes, 0 = no.

- `stent`: Was a stent deployed? 1 = yes, 0 = no.

- `height`: Patient height in centimeters.

- `female`: Patient sex: 1 = female, 0 = male.

- `diabetic`: Was the patient diabetic? 1 = yes, 0 = no.

- `acutemi`: Had the patient suffered an acute myocardial infarction within the last seven days? 1 = yes, 0 = no.

- `ejecfrac`: Left ventricular ejection fraction.

- `ves1proc`: Number of vessels involved in the first percutaneous coronary intervention procedure. 1 = yes, 0 = no.

Walkthrough

From within R, **LocalControl** can be installed and loaded with the following commands:

```
R> install.packages("LocalControl")
R> library("LocalControl")
R> data("lindner", package = "LocalControl")
```

When calling **LocalControl** functions, users must specify relevant columns in the given data frame. The `treatmentColName`, and `outcomeColName` parameters each take a single string which is the name of their respective column in the data frame. The `clusterVars` parameter takes a vector of strings where each element corresponds to the name of a column containing a clustering variable. Note that clustering high-dimensional data is problematic because in high dimensions, every point is likely different from every other point, due to the curse of high dimensions. Selection of a relatively low dimensional space is important. Researchers should explore the possibility of using only a subset of the available X-covariates in clustering. This can have a substantial effect when a large variety of partially redundant covariates are available. In Section 6, a method is described for identifying critical clustering variables. The `clusterCounts` parameter takes a list of integers representing the desired numbers of clusters to form. For each unique element in this list, a different set of clusters is generated, analyzed, and returned.

In the example considered here, a vector of variable names is passed via the `clusterVars` parameter which comprises seven of the lindner covariates that may drive treatment bias. Eleven different cluster sizes are specified in a second vector, ranging from 1 to 50.

```
R> all7Vars <- c("stent", "height", "female", "diabetic",
+    "acutemi", "ejecfrac", "ves1proc")
R> numClusters <- c(1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50)
```

For some parameters, `LocalControlClassic()` only accepts columns with a specific data type. For example, the column containing the outcome variable must be a numeric type. The treatment column can be of any type, however, it must have exactly two levels ("treated" and "control") when converted to a factor.

```
R> linResults <- LocalControlClassic(data = lindner, clusterVars = all7Vars,
+    treatmentColName = "abcix", outcomeColName = "cardbill",
+    clusterCounts = numClusters)
```

Calling `LocalControlClassic()` returns an R environment containing one object for each value in the list passed as an argument to `clusterCounts` with prefix `UPSnnltd` and the cluster count value, as well as a summary of the entire analysis. In this example, the `linResults` object is an environment containing eleven `UPSnnltd*` objects. Each `UPSnnltd*` object is a list containing 34 unique elements, each of which are described on the `LocalControlClassic()` help page. After calling `LocalControlClassic()`, a useful first impression of the output can
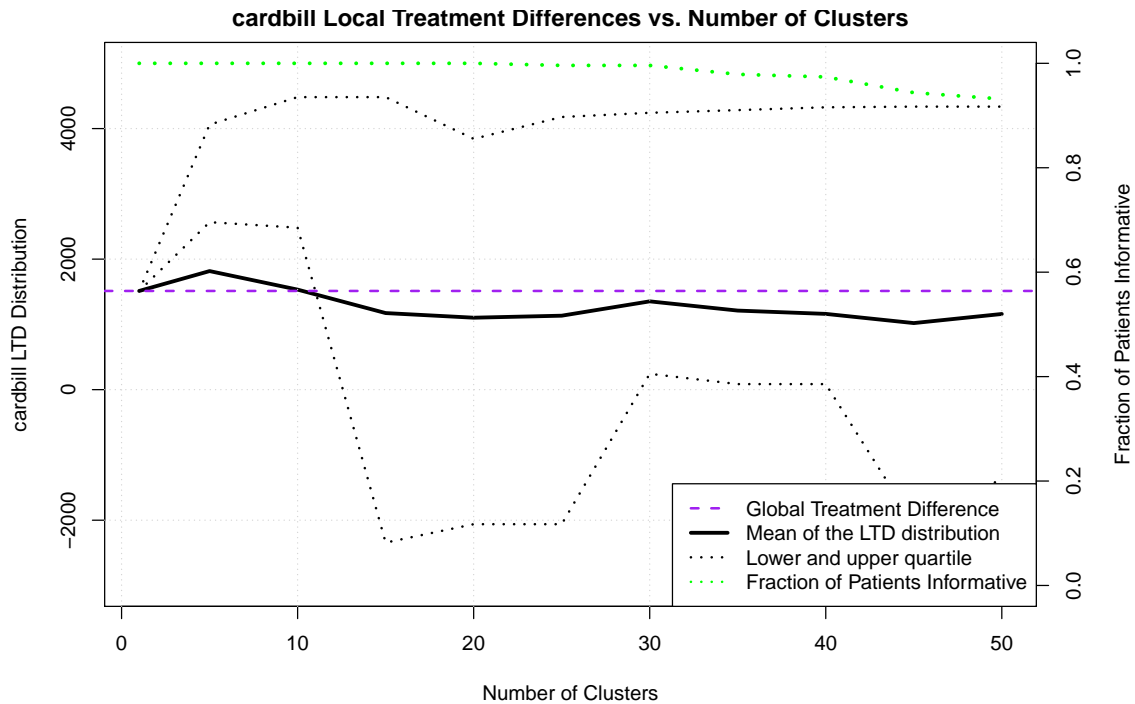
Figure 1: `LocalControlClassic()` analysis describing the distribution of LTD estimates for the Lindner data set. As the number of clusters is increased, within-cluster patient similarity increases, and the estimated treatment outcomes trend towards the results found in the Kereiakes *et al.* study. The green line shows the percentage of the patients that fall within informative clusters, which decreases as much smaller clusters are created. Along the spectrum of cluster counts from 15 to 50, the average treatment difference across all clusters is lower than the $1512 uncorrected estimate.

be created by plotting statistics describing the distribution of local outcome differences as a function of the number of clusters and fraction of informative patients (Figure 1). This plot is created by passing the returned environment to the `LocalControlClassic()` plotting function:

```
R> UPSLTDdist(linResults, ylim = c(-2500, 5000))
```

In the original analysis of Kereiakes *et al.*, the uncorrected $1,512 treatment difference between the patients with and without Abciximab is reduced to $950 after accounting for biases. In Figure 1, the estimated local treatment difference main-effect drops beneath the global average as the number of clusters exceeds 10. Classic local control provides a host of other features which are not mentioned in this article. Further information about the **USPS** package and Obenchain's classic method can be found in the R help pages, or the package documentation (Obenchain 2012).

Figure 2 provides an additional diagnostic for confirming that adjustment for treatment selection bias has occurred. The observed distribution of LTDs is compared with an artificial null distribution based upon the assumption that the available X-confounders are ignorable. In this case, the observed clusters were formed randomly. Thus, when there is strong evidence
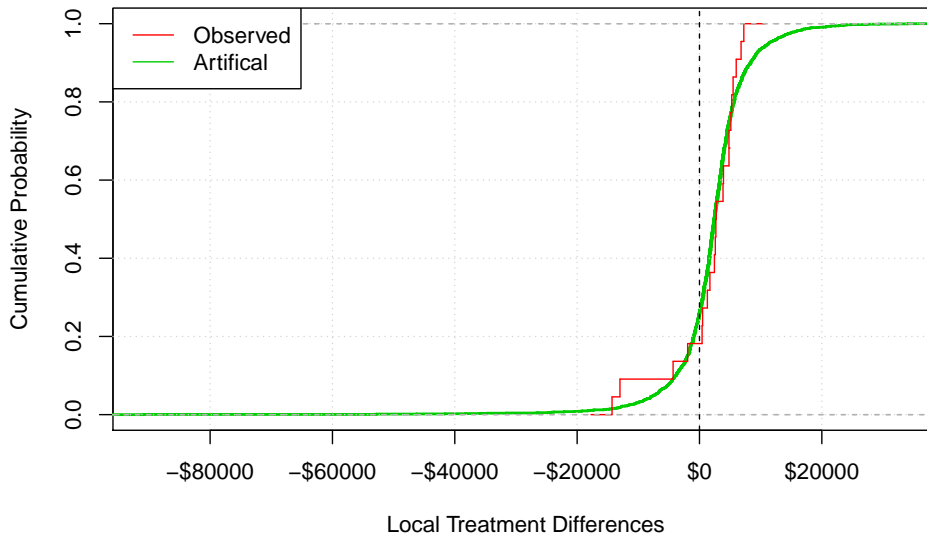
Figure 2: Observed (red) vs. artificial (green) LTD empirical cumulative distribution functions generated using 30 clusters.

that the observed and null distributions are different, adjustment for treatment selection bias has been confirmed. The `ecdf()` function from **stats** (R Core Team 2020) is used to generate the curves for both distributions. A Kolmogorov-Smirnoff test comparing the two distributions results in a D statistic $= 0.42208$, with an approximate $p < 2.2 \times 10^{-16}$. Because the test expects a continuous distribution, but the artificial and observed distributions contain many exact ties in LTD estimates, resampling without replacement is again employed to generate an empirical p value. To accomplish this, null D statistics are calculated which compare the artificial distribution to another 10,000 random permutations of cluster assignments. Of the 10,000 null D statistics computed, only 21 exceed 0.42208 (p value $= 0.0021$). The significant difference between the observed and artificial distributions indicates that X-confounders are not ignorable and that LTDs with reduced bias are adjusting for local imbalances. In Section 6 this data set will be revisited in the context of subgroup analysis, where a patient subgroup accounts for a large portion of bias in the global estimate of treatment difference.

3. Nearest-neighbors local control

3.1. Methodology

Nearest-neighbors clustering

Independently two methods have been developed that share some similarities in addressing how to match patients with covariates in observational studies to correct for biases and confounders: (1) coarsened exact matching, developed by Iacus, King, and Porro (2011); Iacus *et al.* (2012), and (2) the approach developed by members of our team, namely, local control (Obenchain 2012, 2010; Obenchain and Young 2013; Lopiano *et al.* 2014; Faries *et al.* 2013; Young *et al.* 2015). Iacus *et al.* made a key observation: if one has perfectly comparable

patients with respect to the variables that matter for a given question, then one can make a model-free treatment comparison. But as the patients compared become more dissimilar, the (often unarticulated) assumptions behind the implied model that assigns a relative importance to different variables become ever more influential on the estimation process. For instance, is the difference between being male or female as important as a difference of 50 years in age, or a difference in genotype, when grouping patients for comparison? A pharmacogenomic genotype might have a huge bearing on a drug comparison question, but little impact on a surgical comparative effectiveness question. Selecting the correct variables for measurement and decisions about the relative importance of different dimensions creates a need for subject matter experts and leads to uncertainty when trying to defend assumptions that may not be knowable. An innovation of Iacus *et al.* was to explicitly divide the analysis between perfect or near-perfect matches where no assumptions are required, and imperfect matches where one makes assumptions that the patients are "close enough" for the question at hand.

Rather than assessing patient similarity as perfect vs. imperfect matches, local control matches along a continuum. Patients are clustered for similarity on variables that are thought to be sources of bias and confounding. An easily interpretable graph can be created to illustrate how the estimated difference in outcome between two treatments change, on average, across all clusters, as a function of using smaller and more homogenous clusters (Figures 5, 6, and 10). This is analogous to combining a host of smaller studies that are each homogeneous within themselves, but represent the spectrum of variability of people across diverse subpopulations. As the clusters get smaller, some of them can become noninformative, whereby all cluster members contain only one treatment, and there is no basis for comparison. This is actually a feature of the method: for example, if treatment A is given to people of all ages, and treatment B is only given to adults, there is no basis for comparing the drugs for pediatric use. The power of these methods becomes apparent as the sample size increases. For example, treatment A might be commonly used, whereas treatment B is rarely performed on people with the same characteristics. However, when larger sample sizes become available for analysis, it is possible to find close matches for the two treatment groups, with dependence on model assumptions diminishing.

An open issue with classic local control is how the choice of clustering methodology affects treatment comparisons. Because optimal clustering is non-deterministic polynomial-time hard (Dasgupta 2008), numerous "greedy" algorithms exist to create clusters according to different criteria. Even with optimal clustering, a patient that may be quite close to another one, and useful for treatment comparison, could end up in a different cluster. Outlier patients that may still have a few near neighbors with both treatments are frequently separated when one clusters without replacement, preventing their inclusion in comparing treatment outcomes.

To address this limitation of hierarchical clustering, the nearest-neighbors to a given patient are used to estimate treatment differences, instead of clustering without replacement, where patients reside in only a single cluster. Each patient has a unique set of near-neighbors, and the approach becomes more akin to a non-parametric density estimate using similar patients within a covariate hypersphere of a given radius. The local treatment difference is taken as the average of the treatment differences from the neighborhood around each point.

While `LocalControlClassic()` uses the number of clusters as a varying parameter to visualize treatment differences as a function of patient similarity, this function uses a varying radius. The maximum radius enclosing all patients corresponds to the biased estimate which

compares the outcome of all patients with treatment A versus all patients with treatment B. It is useful to plot both the treatment difference and the fraction of the patients who have an informative neighborhood as a function of decreasing radius, delineating a zone bounded by the smallest radius that includes 100% of the data, along with the radius that retains 80% of the data. While these boundaries fit the behavior in our example, it is not always the case that these are critical points in a data set.

One of the largest differences to consider with the new local control functions is that the observations are now sampled with replacement. As a result, the outcome of an individual observation can potentially contribute to the local treatment difference in multiple clusters. With the new method of clustering, each observation becomes the centroid of a cluster, meaning that the number of clusters created is always N. The number of neighbors in clusters, along with the "level" of patient similarity, becomes a function of the clustering radius, r. This is to say, for all N patients, a cluster C_i centered around patient i, has k_i nearest-neighbors where k_i is the number of patients that are within r units of X-space distance to patient i.

By default, local control generates a set of radii whose lengths range inclusively from 0, to the largest distance between any two points in the provided data. The maximum distance is calculated using an open-source implementation (https://github.com/hbf/miniball) of the fast smallest-enclosing-ball algorithm (Fischer, Gärtner, and Kutz 2003). It is important to consider the significance of the minimal and maximal enclosing radius. At the maximal radius which encloses all samples, every cluster is identical. This means that the within-cluster treatment difference, as well as the average across all clusters, will always be equal to the uncorrected global treatment difference. Conversely, when the radius has length 0, the clusters are formed using only patients whose covariates match perfectly. This opens several avenues, such as the coarsening of variables (Iacus *et al.* 2011, 2012), which can be used in conjunction with local control to embed model assumptions about ranges of variables within which treatment outcomes are not expected to vary.

Nearest-neighbors confidence estimates

Nearest-neighbors local control uses bootstrapping to generate confidence estimates for treatment comparisons. The `LocalControlNearestNeighborsConfidence()` function repeatedly resamples rows of the provided data frame with replacement to generate an empirical distribution of the treatment difference. The 95% quantile is drawn from the distribution of results to produce confidence intervals for the LTD at each radius. The number of bootstrapping iterations can be set using the `nBootstrap` parameter.

3.2. Simulated example

Data: Case-control simulation

This data demonstrates the effects of local control on correcting a treatment dosage bias. In this simulation, a cohort of N patients is generated with weights drawn from a normal distribution. The population is divided into two treatment groups, treatment 1 (T1) and treatment 0 (T0), and a bias is introduced where treatment 1 is dosed with a higher variance, σ^2, than treatment 0. The outcome variable, adverse drug reaction (ADR), for both treatments is assigned using the same function: ADR = |target dose − actual dose|mg. In
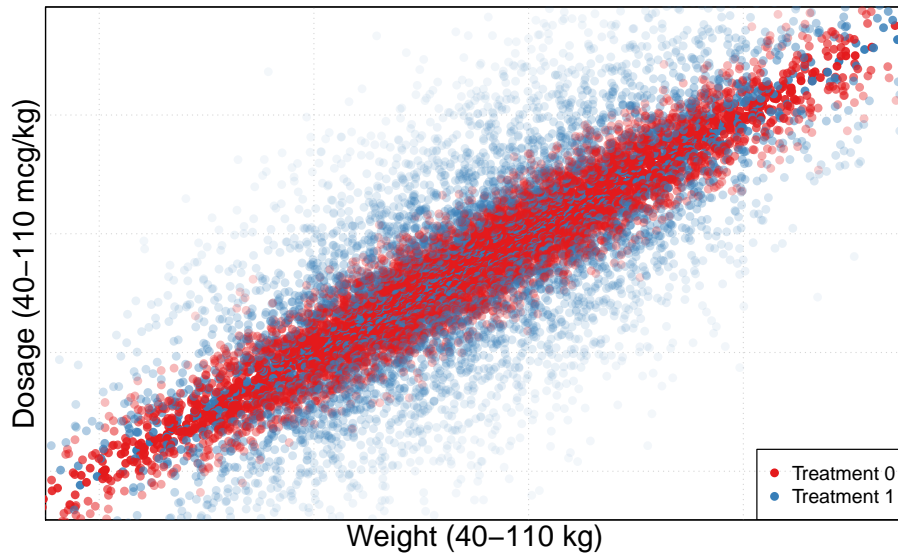
Figure 3: Adverse drug reaction as a function of weight and dosage. The ideal treatment for the simulated data should lie on the diagonal where weight (kg) = dosage (mg). The blue treatment has a higher variance than the red treatment. The pale points indicate patients with a greater adverse reaction to the treatment, while the dark points represent those with smaller reactions.

		T1+T0	T0	T1	p value
N (patients)		10000	5000	5000	
`weight` (kg)	μ	74.76	74.72	74.8	0.804
	σ	14.97	14.99	14.94	0.800
`dosage` (mg)	μ	74.77	74.7	74.84	0.701
	σ	18.69	15.82	21.18	<2.20E-16
`ADR` (mg)	μ	8.03	4.01	12.06	<2.20E-16
	σ	7.86	2.99	9.07	<2.20E-16

Table 2: Case-control simulation cohort summary. A *t*-test shows that there is no statistically significant difference in weight or dosage between the two treatment groups. However, with an *F*-test, there is a highly significant difference in dosage variance between treatments.

this simulation, the optimal dosage is equal to one mg per kg of the patient's weight. Using an absolute value function to generate the outcome makes the data difficult to fit linearly. Glancing at this data without correction makes it appear as though the adverse drug reaction is greater among those who received the first treatment. Table 2 shows the distribution of observations in this data set, Figure 3 graphs the ADR, weight, and dosage, and Figure 4 displays a histogram of the ADR colored by treatment group before and after correction. The simulated data can be created using the R code below:
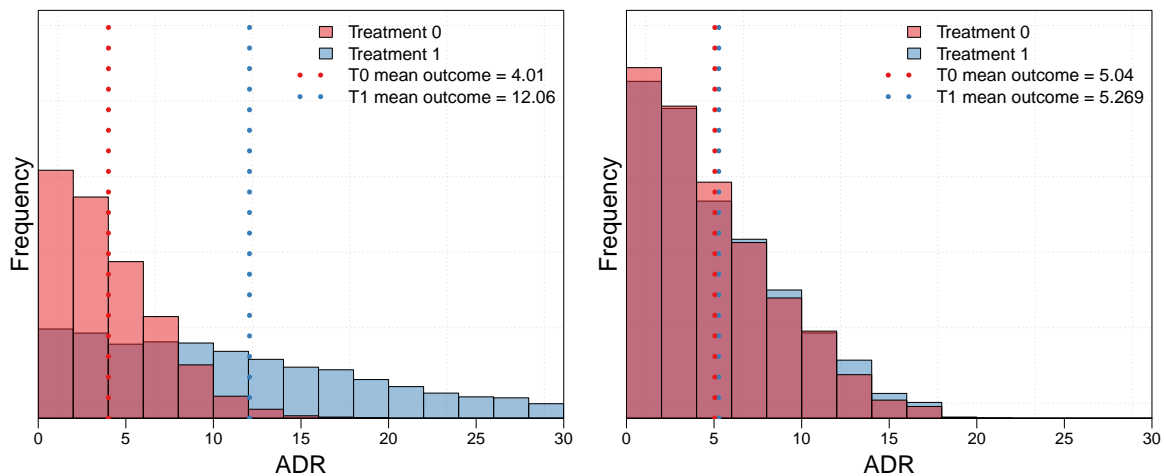
```
R> set.seed(253748)
R> N <- 10000
```

Figure 4: (Left) Histogram of adverse drug reaction outcome in the simulated data. The simulated data has the two drugs affect patients equally, however, it appears that the patients in the 'Treatment 0' group have a much better average outcome due to the lower variance in dosing. (Right) Corrected histogram of adverse drug reaction outcome in the simulated data. In this histogram, the estimated outcomes of T0 and T1 are not appreciably different after accounting for the bias of T1 having a higher variance in dosages. That is, when patients are clustered to have similar weight and dosage, the treatment difference approaches the true value of zero on average across all clusters.

```
R> weight <- c(rnorm(N / 2, 75, 15), rnorm(N / 2, 75, 15))
R> dosage <- weight + c(rnorm(N / 2, 0, 15), rnorm(N / 2, 0,  5))
R> trmt <- c(rep(1, N / 2), rep(0, N / 2))
R> ADR <- abs(weight - dosage)
R> noise1 <- rnorm(n = N, 0, 1)
R> noise2 <- rnorm(n = N, 0, 1)
R> xSim <- data.frame(weight, trmt, dosage, ADR, noise1, noise2)
```

Due to the differences in the two clustering schemes, the parameters for calling the nearest-neighbors function differ slightly from the classic method. This function does not require users to supply the `clusterCounts` parameter. Instead, it automatically generates a set of radii to fit the covariates if one is not provided. Additionally, there are three optional parameters which control the generation of cluster radii. `radStepType` determines if the rate of decay between radii will be uniform or exponential. `radDecayRate` determines the stepsize between radii. If `radStepType` is uniform, `radDecayRate` is subtracted from the the prior radius each iteration, starting from the maximum radius. If `radStepType` is exponential, then `radDecayRate` is multiplied by the prior radius at each iteration, starting from the maximum radius. Last, users can specify the size of the second smallest radius (before zero) as a fraction of the maximum radius with `radMinFract`.

By default, the radii generated by `LocalControl()` decay exponentially by 80% each iteration, with a minimum of 1% the length of the maximum. As with the classic method, the column containing the outcome variable must be of numeric type. The treatment column can be of any type, however, if the treatment variable contains more than two values, users must

weight	dosage	noise1	noise2	difs
−1	−1	−1	−1	0.00
1	−1	−1	−1	−0.01
−1	1	−1	−1	0.78
1	1	−1	−1	3.83
−1	−1	1	−1	−0.01
1	−1	1	−1	−0.01
−1	1	1	−1	0.81
1	1	1	−1	3.72
−1	−1	−1	1	0.01
1	−1	−1	1	−0.01
−1	1	−1	1	0.82
1	1	−1	1	3.74
−1	−1	1	1	0.02
1	−1	1	1	0.01
−1	1	1	1	0.95
1	1	1	1	3.72

Table 3: Regression input for full factorial analysis. The difs column shows the average difference in the corrected LTD from the global treatment difference for each of the 16 combinations. A value of −1 for a clustering variable means that it is excluded, while a value of 1 represents including it in the model.

provide the `treatmentCode` parameter to specify the "primary" treatment group, T1. All remaining values are considered the alternate group, T0. The following code chunk performs the `LocalControl()` analysis on the simulated data, saving the resulting object to a variable in the global environment:

```
R> xSults <- LocalControl(data = xSim, treatmentColName = "trmt",
+    treatmentCode = 1, outcomeColName = "ADR",
+    clusterVars = c("weight", "dosage"), radMinFract = .01,
+    radDecayRate = 0.95, numThreads = 4)
```

When working with a large set of data, or a large number of covariates, it may be beneficial to increase the number of threads used in the local control calculations. This can be done by assigning the `numThreads` parameter a value greater than one. A performance increase is only possible if the running computer is capable of multicore processing.

After calling the function, a histogram is produced using the corrected outcome data produced from local control (right histogram in Figure 4). In the corrected histogram, the two treatment outcomes are nearly identical.

3.3. Choice of clustering variables (feature selection)

One of the open areas for research in local control is how to choose the relevant covariates for bias correction. One approach that is viable for a modest number of covariates is a full factorial regression analysis of how significant each covariate is in modeling the treatment difference. The full factorial approach is illustrated, but note that for more variables, a

fractional factorial approach could be employed for greater efficiency (Box *et al.* 2005). A full factorial design of experiments approach first runs all 2^k combinations of including or excluding each of the k covariates in the local control model. One can then model with linear regression the outcomes as a function of the binary variables (main effects and interactions) that designate which cluster variables were employed in the local control runs. To account for the change in dimensionality during the factorial analysis, the radius length is scaled according to the number of variables in use. Two dummy "noise" variables are included to show the effects of using uncorrelated variables with `LocalControl()`. These outcomes are compared by calculating the average difference from the global estimate for each of the curves.

Table 3 shows the 16 combinations of cluster variables, along with the treatment difference between the uncorrected and corrected estimates with various variables excluded (-1) or included (1) in the correction. The first row has all variables excluded, and thus has no correction. Positive values in the difs column indicate that the combination of variables used in bias correction leads to a decrease over the global biased estimate, and negative values show the opposite. In this case, inclusion of weight and dosage leads to a decrease in the estimate, indicating that the global estimate without covariate correction is high relative to the ground truth of zero embedded in the simulation.

This analysis begins with a call to `LocalControl()`:

```
R> noisyVars <- c("weight", "dosage", "noise1", "noise2")
R> noisySults <- LocalControl(xSim, treatmentColName = "trmt",
+    outcomeColName = "ADR", clusterVars = noisyVars,
+    radMinFract = .01, radDecayRate = 0.95)
R> fixedRads <- summary(noisySults)$radius
```

The radius lengths are saved to be scaled and reused in the coming step. A matrix is created to store all of the different combinations of clustering variables, followed by a call to local control with each combination.

```
R> varCombinations <- expand.grid(0:1, 0:1, 0:1, 0:1)
R> ltext <- apply(X = varCombinations, MARGIN = 1,
+    FUN = function(x) paste0(x, collapse = ""))
R> ltdVecs <- list()
R> ltdVecs[[1]] <- rep(summary(noisySults)$ltd[1],
+    nrow(summary(noisySults)))
R> for (i in 2:16) {
+    varSS <- noisyVars[which(varCombinations[i, ] == 1)]
+    scaleFactor <- sqrt(length(varSS)) / sqrt(length(noisyVars))
+    scaleRads <- fixedRads * scaleFactor
+    sults <- LocalControl(xSim, treatmentColName = "trmt",
+      outcomeColName = "ADR", clusterVars = varSS,
+      radiusLevels = scaleRads)
+    ltdVecs[[i]] <- summary(sults)$ltd
+ }
R> ltdFrame <- data.frame(ltdVecs)
R> names(ltdFrame) <- ltext
```
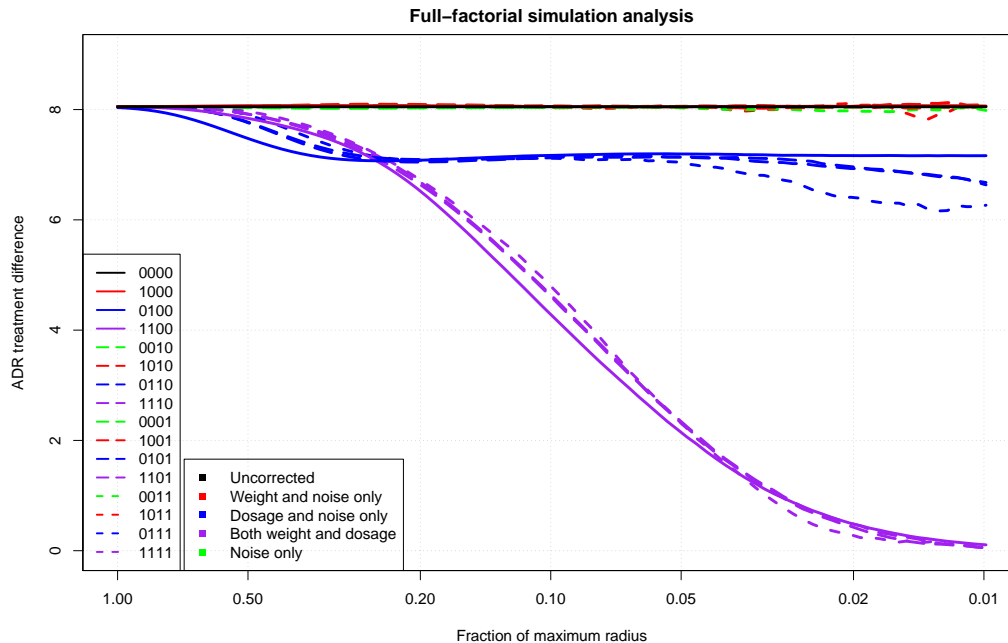
Full–factorial simulation analysis

Figure 5: Full factorial local control on the simulated data. This presents a graphical representation of the different covariate configurations. Each of the curves on the plot corresponds to one of the rows in Table 3. When both weight and dosage are included in the model (purple), the corrected treatment difference converges to the correct answer of zero. When only one of weight or dosage is used in the model (red or blue), or neither (green), then the biases remain, and the treatment difference estimate is non-zero. Because this simulated data contains no perfect matches, the corresponding section is excluded from this plot.

The `avgDif` function compares the LTD vectors to the global average. Using the results from the previous steps, the average difference is calculated for each combination to produce Table 3.

```
R> avgDif <- function(uncorrected, corrected) {
+     return(sum(uncorrected - corrected, na.rm = TRUE) /
+       length(which(!is.na(corrected))))
+ }
R> difs <- numeric()
R> difs[1] <- 0
R> for (i in 2:ncol(ltdFrame)) {
+     difs[i] <- avgDif(ltdFrame[1:92, 1], ltdFrame[1:92, i])
+ }
R> outmat <- data.frame(expand.grid(c(-1, 1), c(-1, 1), c(-1, 1), c(-1, 1)))
R> names(outmat) <- noisyVars
R> outmat$difs <- difs
```

Table 3 shows that regardless of inclusion of the noise terms, if both weight and dosage are included in local control bias correction, that the treatment difference estimate converges to the true difference, namely zero. Conversely, if either weight or dosage are not included in the model, a biased incorrect estimate remains. Figure 5 presents this data graphically.

	Estimate	Std. Error	t value	$\Pr(>|t|)$
(Intercept)	1.15	0.01	97.36	9.22E-19
dosage	1.15	0.01	97.24	9.36E-19
weight	0.72	0.01	61.37	2.32E-16
dosage:weight	0.73	0.01	61.82	2.13E-16

Table 4: Regression output from the full factorial analysis. A regression is performed to explore the effects of having each variable combination in the model. While weight and dosage are significant, the noise variables are not. This indicates that they should be dropped from the model.

Using the values from Table 3, a stepwise full factorial linear model is built to evaluate the significance of each variable with respect to the treatment difference. Table 4 shows that the noise terms are not significant using stepwise regression, either alone or in combination in affecting the treatment estimate, and thus should be removed as covariates from the local control model. Table 4 can be created as follows:

```
R> model <- difs ~ (weight + dosage + noise1 + noise2)^4
R> fit <- glm(difs ~ 1, data = outmat, family = gaussian)
R> fit.AIC <- step(fit, model, direction = "both", k = 2, trace = 0)
R> regTable <- summary(fit.AIC)$coef
```

While the full factorial can be performed for quick results in this small example, the number of runs doubles with the inclusion of each additional covariate. One approach to reducing the dimensionality of local control analysis, while accounting for many sources of bias, is to employ a propensity score as one of the clustering variables to collapse information from many covariates related to treatment bias.

3.4. Lindner analysis with `LocalControl()`

In Section 2, `LocalControlClassic()` was used to analyze the data from the Lindner Abciximab study. Here an analogous analysis is performed using `LocalControl()` to provide a comparison of the methods and results. The `LocalControl()` function is called using the Lindner data frame from Section 2. Average within-cluster treatment difference is plotted as a function of observation similarity within clusters (radius length). Confidence intervals are generated using the `LocalControlNearestNeighborsConfidence()` function. In Figure 6, observe that the `LocalControl()` results show a reduction in treatment cost as the level of correction increases, similar to the original study.

This analysis can be reproduced in R with the following commands:

```
R> linRes <- LocalControl(data = lindner, clusterVars = all7Vars,
+    treatmentColName= "abcix", outcomeColName = "cardbill",
+    treatmentCode = 1)
R> linCI <- LocalControlNearestNeighborsConfidence(data = lindner,
+    clusterVars = all7Vars, treatmentColName = "abcix",
+    outcomeColName = "cardbill", treatmentCode = 1, nBootstrap = 100)
R> plot(linRes, nnConfidence = linCI,
+    main = "LocalControl confidence intervals")
```
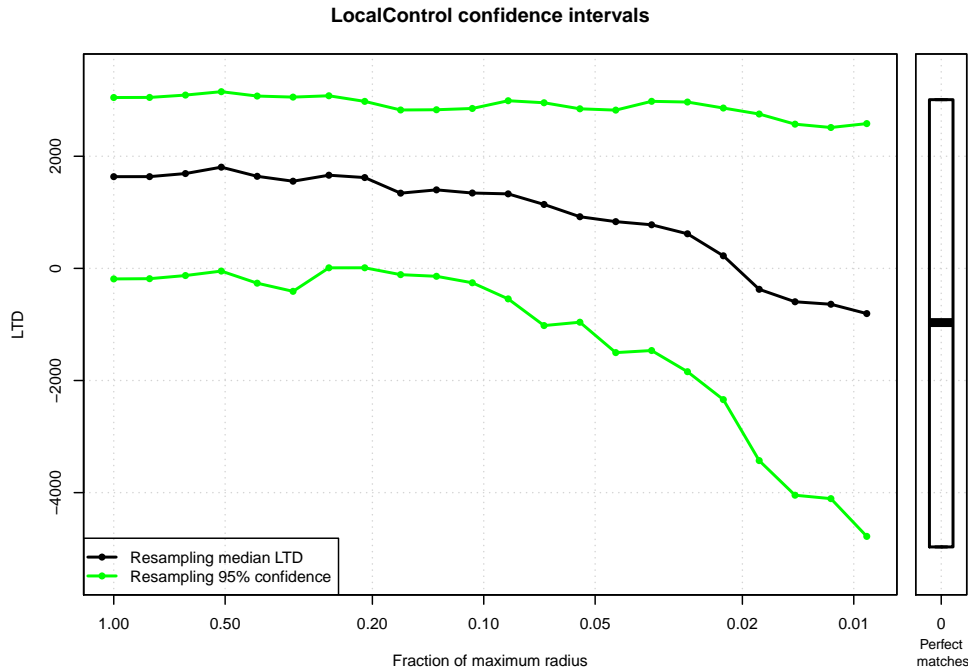
Figure 6: `LocalControl()` confidence estimates from 100 resamples. Confidence intervals are generated by repeatedly resampling N patients with replacement from the original population. `LocalControl()` is run once for each of the resampled populations, storing the results from each run as elements of a list. After 100 automated calls to `LocalControl()`, the 95% confidence intervals are drawn from the resampled results.

4. Survival local control

4.1. Methodology

The `LocalControl()` function is an extension of the nearest-neighbor local control introduced in Section 3. The major variation here is that this adaptation supports survival analysis. In the previous versions of local control, the outcome differences within clusters were examined as a function of the cluster radius. With temporal data, a non-parametric counting method is adopted to compare bias-corrected survival curves. Note that using Kaplan-Meier estimates with a single outcome with potentially censored observations is a special case of the more general competing risks problem where there are one or more competing risks (outcomes). Thus a single function is provided for both Kaplan-Meier estimates and the more general case of multiple competing risks.

Kaplan-Meier survival curves provide an intuitive visualization of time-to-event data. Unfortunately, due to the nature of the counting process, the curves generated with Kaplan-Meier do nothing to correct for covariates in a model, and are thus normally suitable only for randomized studies. With nearest-neighbors clustering, the Kaplan-Meier counting process is adapted to compute survival curves within clusters that are aggregated to produce globally corrected survival curves. These covariate-adjusted survival curves can be easily interpreted, tested, and compared with one another. As a non-parametric method, local control does not

rely on the proportional hazards assumption or the assumption of linear effects of covariates as is the case for Cox regression. Recall that the Kaplan-Meier estimate for survival at time t, $S(t)$, is equal to the product of the number of observations remaining after the events of time t, divided by the number surviving before those events for all times leading up to t (Kaplan and Meier 1958), or:

$$S(t) = \prod_{t_j \leq t} \frac{\text{atrisk}_j - \text{failures}_j}{\text{atrisk}_j}.$$

Bias-corrected survival curves are generated by aggregating survival outcomes from within each cluster. The contribution from each cluster is scaled to ensure that the total number of observations considered at risk never exceeds the original number of observations in the study. Each informative cluster, regardless of the number of nearest-neighbors, contributes equally to the curves generated at a given radius. For example, if cluster j (observation j and its nearest-neighbors) has five observations, while cluster k has twenty, both would increment the total at risk for each treatment by one. Similarly, in the case where the radius reaches all $N-1$ observations, the total at risk would be N.

The concept of "fractional observations" is introduced, whereby within clusters, the contributions to the number at risk, and the event (failure and censor) bins are scaled with respect to the number of neighbors on a given treatment. As an example, consider a cluster j, which contains three T1 and two T0 observations. Cluster j is informative, so it increases the number at risk by one for both treatment groups. Because the total number of events must be equal to the number at risk, the contributions to the event bins within a cluster must sum to one for both treatment groups. In cluster j, the outcomes must be scaled such that each observation contributes only $\frac{1}{\text{numT1}_j} = \frac{1}{3}$ or $\frac{1}{\text{numT0}_j} = \frac{1}{2}$ to the event bin at their respective time. After considering each cluster, for both treatment groups, the total at risk and the sum of all fractional outcomes is equal to the number of informative clusters. After aggregating the surviving fractions across all informative clusters, the Kaplan-Meier counting process is applied to generate survival curves for each radius of correction. With the global radius, this process generates the same Kaplan-Meier curves that can be created from the data naively. This process iterates over decreasing radius lengths to produce curves across many different levels of similarity.

For a given competing risk, the estimator is the cumulative sum over each time interval of the probability neither event occurs before time t (the Kaplan-Meier estimate where both competing risks are combined as an event, and the censored observations are treated as censored) multiplied by the fraction experiencing a given event type out of those still at risk at that time.

Because competing risks is also a counting process, the extension of local control Kaplan-Meier to support competing risks is straightforward. Using the same method of creating fractional observations, a cumulative incidence function (CIF) is created for each type of risk using the following formula:

$$\text{CIF}_{risk}(t) = \sum_{t_j \leq t} \frac{\text{events}_{\text{risk},j}}{\text{atrisk}_j} S(t_{j-1}).$$

Combining the bias correction of local control with a competing risks framework enables computation of bias-corrected cumulative incidence curves while accounting for all possible outcomes. Section 4.2 provides an example using survival based local control to correct bias

Treatment	Outcome	Time	x_1	x_2	...	x_k
A	death	3	red	0	...	98.6
B	censored	9	green	1	...	99.2
⋮	⋮	⋮	⋮	⋮	⋱	⋮
A	cancer	4	blue	0	...	101.1

Table 5: Data frame for survival-based local control. Contains all of the columns which are necessary to run `LocalControl()`. The first three from the left, treatment, outcome, and time must be included for all survival analyses. The remaining x columns correspond to the covariates used for clustering observations.

in a simulated set of data. Section 5 presents an in-depth competing risks case study using the publicly available Framingham Heart Study data.

Competing risks confidence intervals

The `LocalControlCompetingRisksConfidence()` function produces pointwise standard error estimates for the `LocalControl()` cumulative incidence functions. This is done using an implementation of Choudhury's approach (Choudhury 2002) that supports local control's fractional observations. Users can pass the object returned from the competing risks function to `LocalControlCompetingRisksConfidence()`, which produces confidence intervals corresponding to each of the calculated CIFs. The function currently supports the creation of 90%, 95%, 98%, and 99% confidence intervals. Additionally, this function allows users to choose between the linear, log(-log), and arcsine confidence interval transformations which are detailed in Choudhury's work.

Competing risks hypothesis testing

In addition to the confidence intervals described above, `LocalControl()` also supports hypothesis testing using the Pepe and Mori method (Pepe and Mori 1993). This test compares two CIFs using the area between the curves, weighting the differences to account for time passed and the number of observations remaining. The function gives higher weights to differences which occur earlier in time, where more patients remain at risk. The code used to perform the hypothesis testing is derived from the `compCIF()` function provided in Pintilie (2006). As with Choudhury's, this modified function also works with fractional observations. At each radius, the test is performed on the CIFs from the first risk for the two treatment groups. Each test returns a χ^2 and p value which can be retrieved by calling `summary()` on the object returned from `LocalControl()`.

Survival data format

For survival analysis performed using `LocalControl(outcomeType = "survival")`, the outcome variable must be categorical, where the values correspond to types of risk, or right-censoring (specified with `cenCode`). Additionally, the data frame must contain a variable representing the time that the outcome occurred. Table 5 displays an example of a valid survival data frame.

There are two major parameter differences for `LocalControl()` when working with survival outcomes.

	A + B	A	B	p value
N (patients)	10000	4708	5292	
age (years)	41.48	38.36	44.26	2.11E-108
BMI ($\frac{\text{kg}}{\text{m}^2}$)	25.98	25.58	26.34	3.84E-21

Table 6: Survival simulation cohort summary. A hypothetical hypertension Treatment A (blue) is prescribed more frequently to younger, healthier patients with a low body mass index (BMI), Treatment B (red) is prescribed to older patients with a higher body mass index. Significant treatment biases exist for age and BMI.

- Time to outcome: Rather than severity or magnitude of an outcome, this function takes as input the time to an event. This means that an additional column must be specified, containing the amount of time it took to reach the observed outcome. This function supports time in both integer and floating point formats.

- Categorical outcomes: This function is used with survival or competing risks data. The column of outcomes provided should correspond to the category of outcome, rather than a measure of effect. With right-censored survival data not involving competing risks, the outcome column is generally binary or logical with a value of 1 for patients who experienced the outcome, and 0 for those who were right-censored. For competing risks data, multiple factors can be included, with one of them representing right-censoring.

4.2. Example

Data: Survival simulation

This simulated data demonstrates the effects of local control on correcting bias within survival data. In this simulation, a treatment bias is introduced which skews the global treatment difference. Treatments A and B are two pharmaceutically equivalent treatments. The true effects of these drugs are masked by assigning treatment A to younger, lower BMI patients, and treatment B to those who are older and have a higher BMI. That is, the two treatments affect all patients equally, but one drug is given to the healthier patients, making the alternative superficially appear to have inferior outcomes due to the detrimental effects of age and obesity. Table 6 describes the two treatment groups.

The following code can be used to generate this data in an R session:

```
R> weibullSim <- function(N, lambda, rho, betaage, betabmi) {
+    bmi <- rnorm(N, mean = 26, sd = 4)
+    age <- runif(N) * 47 + 18
+    pbmi <- (bmi - min(bmi)) / (max(bmi) - min(bmi)) * 0.8 + 0.1
+    page <- (age - min(age)) / (max(age) - min(age)) * 0.8 + 0.1
+    drug <- 1 - rbinom(N, 1, (pbmi + page) / 2)
+    et <- exp(bmi * betabmi + age * betaage)
+    Tlat <- (-log(runif(n = N)) / (lambda * et))^(1 / rho)
+    C <- runif(N) * 30
+    time <- pmin(Tlat, C)
+    status <- as.numeric(Tlat <= C)
+    data.frame(id = 1:N, drug, age, bmi, time, status)
```
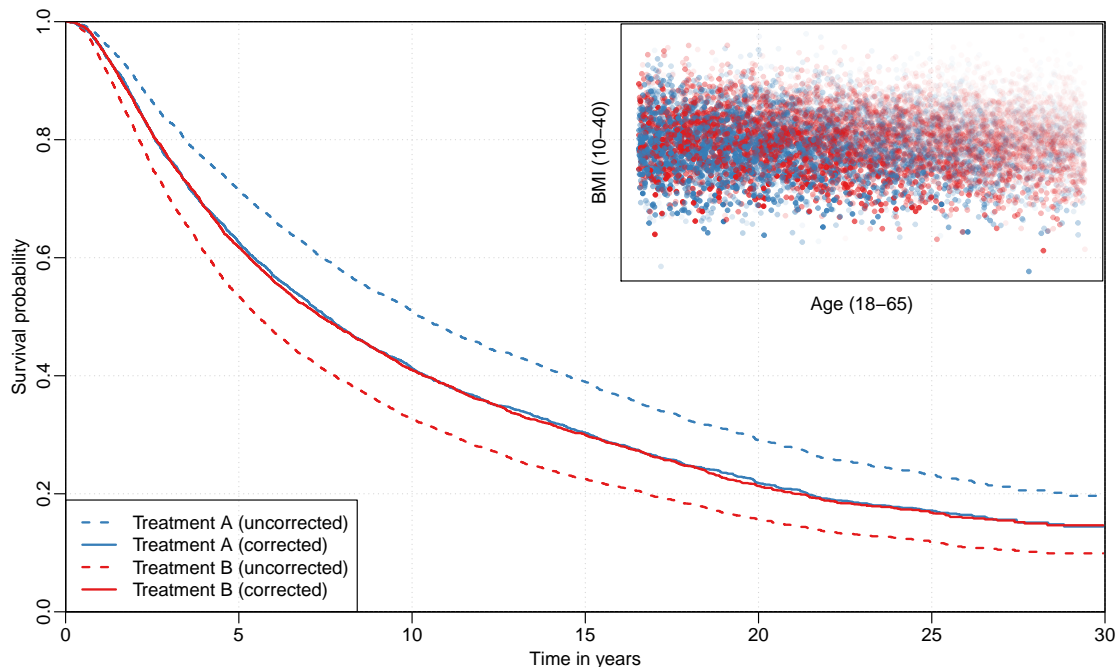
Figure 7: Treatment bias correction using local control on the survival simulation. Because of the treatment assignment bias, patients on A appear to have better outcomes than those on B (dotted lines on Kaplan-Meier plot). However, the local control corrected curves (solid lines) show the true treatment effect, that the two treatments are identical, when patients are clustered for similarity of age and BMI. The upper right subfigure shows a scatterplot of age and BMI in the survival simulation. The shading of points indicates the time to failure, with light shading corresponding to a short survival time, while darker points represent a longer survival time. The color of the points represents the treatment group of an observation. Blue and red points indicate whether a patient received treatment A or B, respectively.

```
+   }
R> survSimData <- weibullSim(10000, 1e-10, 2.6, log(1.2), log(1.45))
```

The **LocalControl** package also includes a saved copy of this simulation, `cardSim`, which can be loaded using `data("cardSim", package = "LocalControl")`. After generating or loading the data, the covariates are specified and `LocalControl()` is invoked.

```
R> results <- LocalControl(data = cardSim, outcomeType = "survival",
+     treatmentColName = "drug", timeColName = "time",
+     outcomeColName = "status", clusterVars = c("age", "bmi"))
```

The object returned from `LocalControl()` is an R list containing vectors, data frames, and nested lists. The `results$KM` element contains the Kaplan-Meier survival curves for both treatment groups at each radius. The `results$CIF` entry contains a list of lists for each different risk in the model. These sublists each contain a pair of data frames (T1 and T0) with CIFs for each radius. If there is only one possible type of failure (not including censoring) in the data provided, then both treatment groups will have one cumulative incidence curve generated per radius which are equivalent to 1 minus the Kaplan-Meier estimate. Figure 7 illustrates

the correction that occurs when calling `LocalControl(outcomeType = "survival")` on the biased simulated data discussed previously in Section 4.2. The dotted lines show the survival curves generated from the raw data. Without correction, it appears that the blue (treatment A) and red (treatment B) patients have nearly identical outcomes. The solid lines on the plot represent the curves generated across local control clusters at a much smaller radius (7.61 vs. 0.82 radius units). In Section 5, local control survival analysis is applied to real data from the Framingham Heart Study.

5. Case study: Framingham heart patients

The effects of smoking on the time to the competing risk of either reaching death, or being diagnosed with hypertension are analyzed using local control. Those who leave the study prior to reaching either of these outcomes, or reach the study conclusion without either outcome occurring, are considered to be right-censored observations. The available covariates are tested for a significant impact on the outcome and examine the results produced along with their interpretation.

Data: `framingham`

The Framingham study data tracks the cardiac health of more than 4000 patients over the course of twenty-four years (Dawber, Meadors, and Moore Jr 1951). A subset of the data is provided that has been approved for training and testing purposes. More information about the Framingham Heart Study can be found at `https://www.framinghamheartstudy.org/`. While the original data includes several additional variables, only the following are used in this analysis:

- `female`: Sex of the patient. 1 = female, 0 = male.

- `totchol`: Total cholesterol of patient at study entry in milligrams per deciliter ($\frac{mg}{dL}$).

- `age`: Age in years of the patient at study entry.

- `bmi`: Patient body mass index in kilograms per square meter ($\frac{kg}{m^2}$).

- `BPVar`: Average units of systolic and diastolic blood pressure above normal in millimeters of mercury (mm Hg): $((SystolicBP - 120)/2) + (DiasystolicBP - 80)$.

- `heartrte`: Patient heartrate in beats per minute (bpm) taken at study entry.

- `glucose`: Patient blood glucose level in milligrams per deciliter ($\frac{mg}{dL}$).

- `cursmoke`: Whether or not the patient was a smoker at the time of study entry.

- `outcome`: Did the patient die, experience hypertension, or leave the study without experiencing either event.

- `time_outcome`: The time at which the patient experienced outcome.

- `cigpday`: Number of cigarettes smoked per day at time of study entry.

	All patients	Smokers	Non-smokers	*p* value
N (patients)	2316	1238	1078	
`female`	0.56	0.48	0.65	3.34E-16
`totchol` ($\frac{\text{mg}}{\text{dL}}$)	230.34	229.18	231.67	1.55E-01
`age` (years)	47.43	46.12	48.94	8.15E-17
`BMI` ($\frac{\text{kg}}{\text{m}^2}$)	24.78	24.28	25.35	2.61E-14
`BPVar` (mm Hg)	-3.45	-4.49	-2.26	2.28E-06
`heartrte` (bpm)	74.17	74.94	73.28	3.93E-04
`glucose` ($\frac{\text{mg}}{\text{dL}}$)	78.54	78.11	79.03	6.92E-02

Table 7: Framingham Heart Study cohort biases. Patients with preexisting cardiovascular conditions are dropped from the study. Fisher's exact test is used for the comparison of the female binary covariate. For the remaining continuous covariates, a *t*-test is used to compare the two groups. Smoking "treatment" bias significantly affects sex, age, BMI, blood pressure, and heart rate.

Due to the high correlation between diastolic and systolic blood pressure, the two variables are combined by centering them at the threshold of ideal/pre-high blood pressure, then scaling comparably and summing them to create `BPvar`. Patients with preexisting conditions are also removed to form a more comparable population (Table 7). The competing risks of hypertension and death are analyzed.

```
R> data("framingham", package = "LocalControl")
R> framVars <- c("female", "totchol", "age", "bmi", "BPVar", "heartrte",
+    "glucose")
R> FHSResults <- LocalControl(data = framingham, outcomeType = "survival",
+    treatmentColName = "cursmoke", treatmentCode = 1,
+    timeColName = "time_outcome", outcomeColName = "outcome",
+    clusterVars = framVars)
R> summary(FHSResults)
```

The summary frame contains the percentage of informative information across all levels of radius correction (Table 8). Figure 8 shows the cumulative incidence curves for both risks and treatment groups first without any correction, then with the correction observed at the 11th radius, corresponding with 78.3% of the data being informative.

The uncorrected plot of Figure 8 shows that after a long exposure, the cumulative incidence of death in the smoking treatment group is higher than that of the non-smokers. What is surprising is that it appears as though smoking protects individuals from hypertension. After correcting with local control, the hypertension curve for non-smokers shifts down towards the smoking group, and is no longer significantly different. Note that the death CIFs remain almost identical in both of these plots. Does smoking protect from hypertension? An early article claimed that cigarette smoking inhibits blood pressure (Seltzer 1974), but a more recent review suggests the evidence is inconclusive (Virdis, Giannarelli, Neves, Taddei, and Ghiadoni 2010). Even if smoking reduced hypertension, the competing risk of death is still higher for smokers.
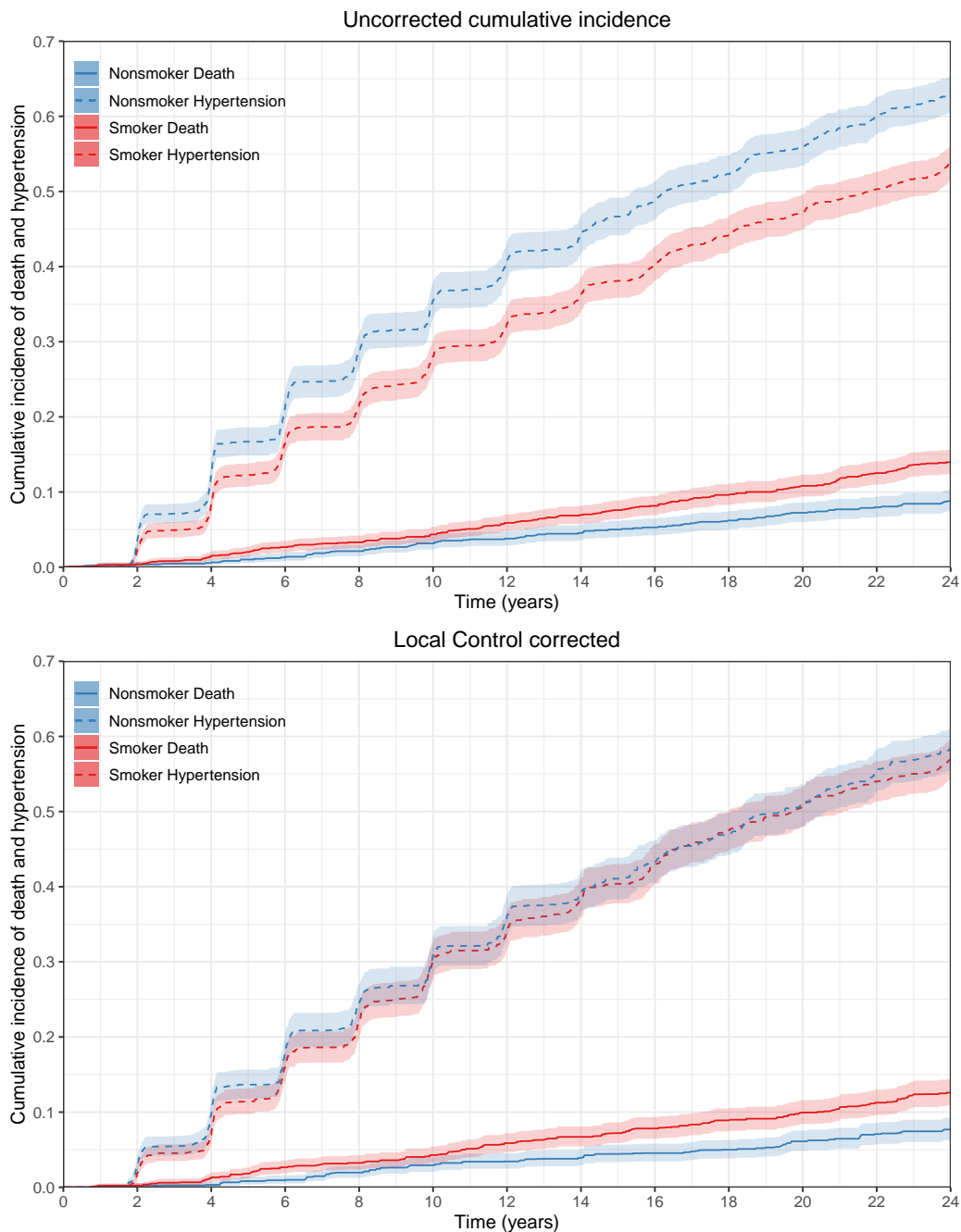
Figure 8: Competing risks of hypertension and death among smokers and non-smokers in the Framingham Heart Study. The top plot shows the cumulative incidence without any correction for covariates. This biased estimate suggests that non-smokers have a higher risk for hypertension and lower risk of death. The bottom plot displays the results from local control after correcting for sex, cholesterol, age, BMI, heart rate, blood pressure, and blood glucose level. The competing risks local control bias-corrected curves show us that, among comparable patients, there is almost no difference in the rate of hypertension over time, but that the greater risk of death remains for smokers. The shaded areas represent the 95% confidence interval estimates.

	Radius	Pct informative	Pct radius	χ^2	p value
1	11.41	1.00	1.00	20.45	6.13E-06
2	9.13	1.00	0.80	20.45	6.13E-06
3	7.30	1.00	0.64	20.38	6.36E-06
4	5.84	1.00	0.51	19.73	8.90E-06
5	4.67	1.00	0.41	19.38	1.07E-05
6	3.74	1.00	0.33	15.78	7.12E-05
7	2.99	1.00	0.26	5.99	1.44E-02
8	2.39	0.99	0.21	3.08	7.93E-02
9	1.91	0.97	0.17	1.19	2.75E-01
10	1.53	0.92	0.13	1.44	2.31E-01
11	1.23	0.78	0.11	2.25	1.33E-01

Table 8: Framingham local control summary. Each row corresponds to one radius of correction. The values in the first column are the radius lengths in normalized units. The second column contains the fraction of observations who are informative at the given radius. The pct radius column is the size of the radius as a fraction of the maximum distance between any two observations. The last two columns contain the results from the hypothesis tests comparing the hypertension CIFs for the two treatment groups (as described in Section 4.1).

6. Patient level prediction/heterogeneity of treatment effect

Heretofore, covariates have been used to group patients for comparison to estimate a bias-corrected global treatment difference between a pair of treatments within a population. Such evidence is useful in making generalizations that one treatment may be safer or more effective than another on average. However, this does not answer the question of what is the expected outcome from a given treatment for a particular patient. Patient level prediction recognizes that there may be heterogeneity of treatment effect, namely that patients can have very different outcomes depending on patient characteristics. Traditional approaches will use regression models or machine learning on patient covariates to predict patient outcomes. While these approaches can provide patient level predictions, the interpretation of such models could be distorted by the biasing variables. Instead, after bias correction, regression or machine learning can be applied to model bias-corrected treatment differences, giving insight into what variables modify the difference in outcome from one treatment to another, unpolluted by variables that govern choice of treatment.

In Section 2, an analysis of the Lindner data was presented using `LocalControlClassic()`. In Section 3, `LocalControl()` was used to provide a comparison of the results from the two methods. The Lindner data is now analyzed for the third time, for the investigation of patient subgroups. This analysis continues from Section 3.4, having just called `LocalControl()` on the Lindner data, and plotting the results (Figure 6). Recursive partitioning is used to explore patient subgroups with statistically significant differences in bias-corrected treatment difference as a function of patient covariates, including the clustering variables (Obenchain and Young 2013; Faries *et al.* 2013; Young *et al.* 2015, 2016). Statistical significance was adjusted to account for multiple comparisons. A clustering radius must be selected to begin the analysis. The problem of radius selection is similar to that of selecting bin sizes when using propensity scoring. It is difficult to say which radius is "correct", and the results may vary
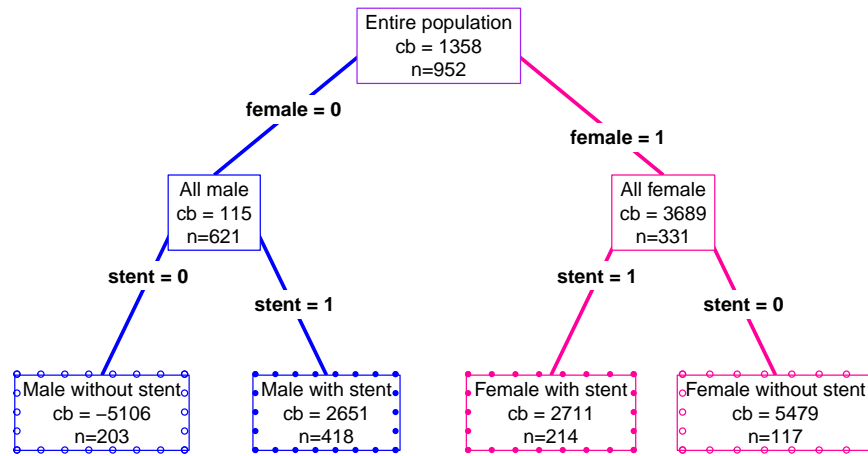
Figure 9: Recursive partitioning tree. Using the results from the analysis in Section 3.4 as input to recursive partitioning, variables are identified which produce significant treatment differences. The color of the nodes is used to differentiate between the entire population (purple), subgroups containing only women (pink), and those with only male patients (blue). The dots bordering the leaves represent a second partitioning of men and women. Solid dots represent patients with a stent, while hollow dots represent those without. The `LocalControl()` outcomes for each of these subgroups are displayed in Figure 10.
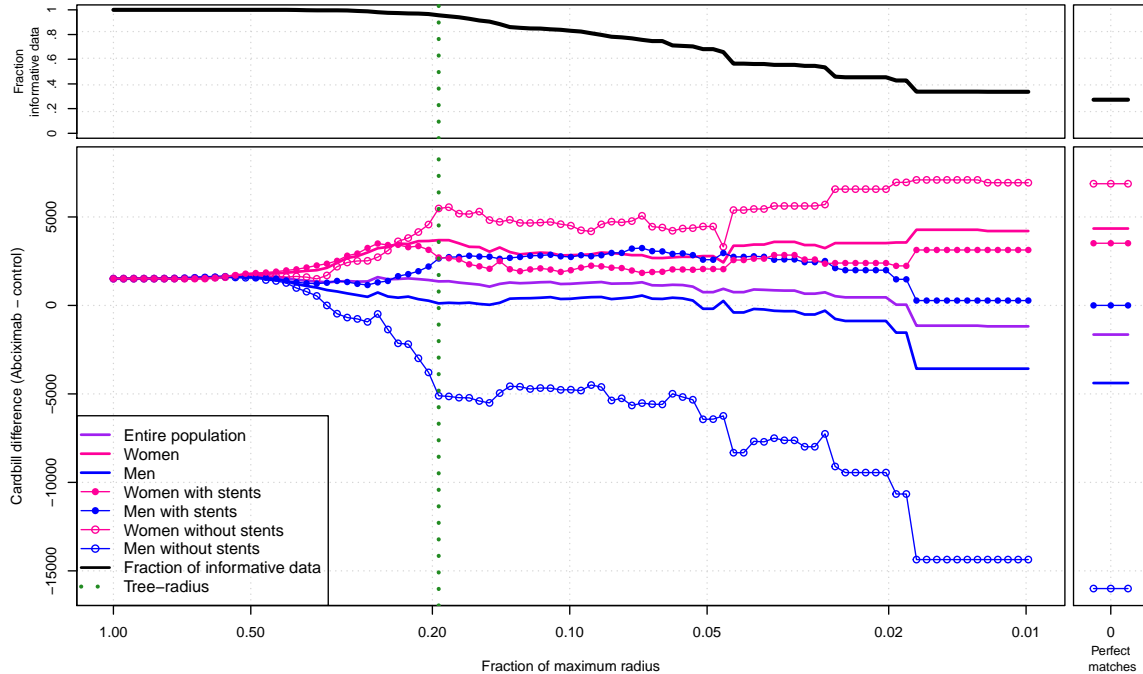

Figure 10: Local control subgroup analysis. After identifying significant subgroups with recursive partitioning, the subgroup treatment differences are graphed as a function of radius. Observe that the men without stents have a much lower billing cost on Abciximab vs. control than each of the other subgroups.

significantly from one to the next. It is thus important to examine the behavior of the estimates across a range of radii, as well as compare those results to perfectly matched patients, if they exist. When a radius must be selected, it is useful to plot the fraction of patients who are considered informative at a given radius. In this example, a radius is chosen where 95% of the data is informative, and where the estimates are also plateauing (Figure 6). Each patient is assigned the average treatment difference produced within their cluster at the selected radius. With local treatment difference as the dependent variable, and patient covariates as the independent variables, recursive partitioning is used to classify patient subgroups. In Figure 9, recursive partitioning identifies four mutually exclusive subgroups: men and women with and without stents. Patients are then divided into these identified subgroups to examine the average local treatment differences per subgroup (Figure 10). The data suggests over a wide range of radii of bias correction, that men without stents result in lower cost of care on Abciximab, but that all other subgroups have a lower or neutral cost of treatment on usual care alone.

In large data sets it can be true that an "average/overall" effect is meaningless. The answer is that "it depends". For example, a drug might work for women, but not for men. When there is treatment response heterogeneity, a recommendation of one-size-fits-all is problematic and even a bias-corrected overall effect is misleading. Local control enables the analysis of both the bias-corrected average effect, as well as creates insight into subgroup outcome heterogeneity.

7. Conclusion

The R **LocalControl** package has been presented with examples of bias-corrected estimation of treatment outcome differences for observational studies, including time-to-event data with competing risks. Patient level prediction and heterogeneity of treatment effect analysis is currently not implemented for survival analysis. It remains as future work to adapt this approach to survival-based outcomes, for example by extensions to survival-based recursive partitioning trees (Bou-Hamad, Larocque, and Ben-Ameur 2011).

Computational details

The results in this paper were obtained using R 3.6.0 with the **LocalControl** 1.1.2.1 package. R itself and the following packages which have been used throughout the paper are available from the Comprehensive R Archive Network (CRAN, https://CRAN.R-project.org/): **xtable** (Dahl 2019) **TeachingDemos** (Snow 2020) **gplots** (Warnes *et al.* 2020) **dendextend** (Galili 2015) **data.table** (Dowle and Srinivasan 2020) **colorspace** (Zeileis *et al.* 2020) **RColorBrewer** (Neuwirth 2014) **gridExtra** (Auguie 2017) **ggplot2** (Wickham 2009) **rpart** (Therneau and Atkinson 2019) **rpart.plot** (Milborrow 2020).

Acknowledgments

References

Addelman S (1969). "The Generalized Randomized Block Design." *The American Statistician*, **23**(4), 35–36. doi:10.2307/2681737.

Auguie B (2017). **gridExtra**: *Miscellaneous Functions for* **grid** *Graphics*. R package version 2.3, URL https://CRAN.R-project.org/package=gridExtra.

Austin PC (2014). "The Use of Propensity Score Methods with Survival or Time-to-Event Outcomes: Reporting Measures of Effect Similar to Those Used in Randomized Experiments." *Statistics in Medicine*, **33**(7), 1242–1258. doi:10.1002/sim.5984.

Austin PC, Schuster T (2016). "The Performance of Different Propensity Score Methods for Estimating Absolute Effects of Treatments on Survival Outcomes: A Simulation Study." *Statistical Methods in Medical Research*, **25**(5), 2214–2237. doi:10.1177/0962280213519716.

Bou-Hamad I, Larocque D, Ben-Ameur H (2011). "A Review of Survival Trees." *Statistics Surveys*, **5**(2011), 44–71. doi:10.1214/09-ss047.

Box GEP, Hunter JS, Hunter WG (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York.

Choudhury JB (2002). "Non-Parametric Confidence Interval Estimation for Competing Risks Analysis: Application to Contraceptive Data." *Statistics in Medicine*, **21**(8), 1129–1144. doi:10.1002/sim.1070.

Cox DR (1972). "Regression Models and Life-Tables." *Journal of the Royal Statistical Society B*, **34**(2), 187–220. doi:10.1111/j.2517-6161.1972.tb00899.x.

Dahl DB (2019). **xtable**: *Export Tables to LATEX or HTML*. R package version 1.8-4, URL https://CRAN.R-project.org/package=xtable.

Dasgupta S (2008). "The Hardness of *K*-Means Clustering." *Technical Report CS2008-0916*, Department of Computer Science and Engineering, University of California, San Diego.

Dawber TR, Meadors GF, Moore Jr FE (1951). "Epidemiological Approaches to Heart Disease: The Framingham Study." *American Journal of Public Health and the Nation's Health*, **41**(3), 279–281. doi:10.2105/ajph.41.3.279.

Dowle M, Srinivasan A (2020). **data.table**: *Extension of* `data.frame`. R package version 1.13.0, URL https://CRAN.R-project.org/package=data.table.

Faries DE, Chen Y, Lipkovich I, Zagar A, Liu X, Obenchain RL (2013). "Local Control for Identifying Subgroups of Interest in Observational Research: Persistence of Treatment for Major Depressive Disorder." *The International Journal of Methods in Psychiatric Research*, **22**(3), 185–194. doi:10.1002/mpr.1390.

Fine JP, Gray RJ (1999). "A Proportional Hazards Model for the Subdistribution of a Competing Risk." *The Journal of the American Statistical Association*, **94**(446), 496–509. doi:10.1080/01621459.1999.10474144.

Fischer K, Gärtner B, Kutz M (2003). "Fast Smallest-Enclosing-Ball Computation in High Dimensions." In *Algorithms – ESA 2003*, pp. 630–641. Springer-Verlag.

Fisher RA (1992). "The Arrangement of Field Experiments." In *Breakthroughs in Statistics*, pp. 82–91. Springer.

Galili T (2015). "**dendextend**: An R package for Visualizing, Adjusting, and Comparing Trees of Hierarchical Clustering." *Bioinformatics*, **31**(22), 3718–3720. doi:10.1093/bioinformatics/btv428.

Gayat E, Resche-Rigon M, Mary JY, Porcher R (2012). "Propensity Score Applied to Survival Data Analysis Through Proportional Hazards Models: A Monte Carlo Study." *Pharmaceutical Statistics*, **11**(3), 222–229. doi:10.1002/pst.537.

Gray RJ (1988). "A Class of K-Sample Tests for Comparing the Cumulative Incidence of a Competing Risk." *The Annals of Statistics*, **16**(3), 1141–1154. doi:10.1214/aos/1176350951.

Iacus SM, King G, Porro G (2011). "Multivariate Matching Methods That Are Monotonic Imbalance Bounding." *Journal of the American Statistical Association*, **106**(493), 345–361. doi:10.1198/jasa.2011.tm09599.

Iacus SM, King G, Porro G (2012). "Causal Inference without Balance Checking: Coarsened Exact Matching." *Political Analysis*, **20**(1), 1–24. doi:10.1093/pan/mpr013.

Kaplan EL, Meier P (1958). "Nonparametric Estimation from Incomplete Observations." *Journal of the American Statistical Association*, **53**(282), 457–481. doi:10.1080/01621459.1958.10501452.

Kereiakes DJ, Obenchain RL, Barber BL, Smith A, McDonald M, Broderick TM, Runyon JP, Shimshak TM, Schneider JF, Hattemer CR, Roth EM, Whang DD, Cocks D, Abbottsmith CW (2000). "Abciximab Provides Cost-Effective Survival Advantage in High-Volume Interventional Practice." *American Heart Journal*, **140**(4), 603–610. doi:10.1067/mhj.2000.109647.

Lauve NR, Nelson SJ, Young SS, Obenchain RL, Lambert CG (2020). ***LocalControl**: Nonparametric Methods for Generating High Quality Comparative Effectiveness Evidence*. R package version 1.1.2.2, URL https://CRAN.R-project.org/package=LocalControl.

Lopiano KK, Obenchain RL, Young SS (2014). "Fair Treatment Comparisons in Observational Research." *Statistical Analysis and Data Mining*, **7**(5), 376–384. doi:10.1002/sam.11235.

Milborrow S (2020). ***rpart.plot**: Plot **rpart** Models: An Enhanced Version of* plot.rpart. R package version 3.0.9, URL https://CRAN.R-project.org/package=rpart.plot.

Neuwirth E (2014). ***RColorBrewer: ColorBrewer** Palettes*. R package version 1.1-2, URL https://CRAN.R-project.org/package=RColorBrewer.

Obenchain RL (2010). "The Local Control Approach Using **JMP**." In D Faries, AC Leon, JM Haro, RL Obenchain (eds.), *Analysis of Observational Health Care Data Using* SAS, pp. 151–194. SAS Institute, Cary, NC.

Obenchain RL (2012). **USPS**: *Unsupervised and Supervised Propensity Scoring in R*. R package version 1.2-2, URL `https://CRAN.R-project.org/src/contrib/Archive/USPS`.

Obenchain RL, Young SS (2013). "Advancing Statistical Thinking in Observational Health Care Research." *Journal of Statistical Theory and Practice*, **7**(2), 456–506. `doi:10.1080/15598608.2013.772821`.

Pepe MS, Mori M (1993). "Kaplan-Meier, Marginal or Conditional Probability Curves in Summarizing Competing Risks Failure Time Data?" *Statistics in Medicine*, **12**(8), 737–751. `doi:10.1002/sim.4780120803`.

Pintilie M (2006). *Competing Risks: A Practical Perspective*, volume 58. John Wiley & Sons.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Rosenbaum PR, Rubin DB (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika*, **70**(1), 41–55. `doi:10.1093/biomet/70.1.41`.

Rosenbaum PR, Rubin DB (1985). "Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity Score." *The American Statistician*, **39**(1), 33–38. `doi:10.1080/00031305.1985.10479383`.

Ryan PB, Madigan D, Stang PE, Overhage JM, Racoosin JA, Hartzema AG (2012). "Empirical Assessment of Methods for Risk Identification in Healthcare Data: Results from the Experiments of the Observational Medical Outcomes Partnership." *Statistics in Medicine*, **31**(30), 4401–4415. `doi:10.1002/sim.5620`.

Seltzer CC (1974). "Effect of Smoking on Blood Pressure." *American Heart Journal*, **87**(5), 558–564. `doi:10.1016/0002-8703(74)90492-x`.

Snow G (2020). **TeachingDemos**: *Demonstrations for Teaching and Learning*. R package version 2.12, URL `https://CRAN.R-project.org/package=TeachingDemos`.

Stang PE, Ryan PB, Racoosin JA, Overhage JM, Hartzema AG, Reich C, Welebob E, Scarnecchia T, Woodcock J (2010). "Advancing the Science for Active Surveillance: Rationale and Design for the Observational Medical Outcomes Partnership." *The Annals of Internal Medicine*, **153**(9), 600–606. `doi:10.7326/0003-4819-153-9-201011020-00010`.

Student (1911). "Appendix to Mercer and Hall's Paper on 'The Experimental Error of Field Trials'. Journal of Agricultural Science 4, 128–131." In E Pearson, J Wishart (eds.), *"Student's" Collected Papers*, pp. 49–52.

Therneau T, Atkinson B (2019). **rpart**: *Recursive Partitioning and Regression Trees*. R package version 4.1-15, URL `https://CRAN.R-project.org/package=rpart`.

Virdis A, Giannarelli C, Neves MF, Taddei S, Ghiadoni L (2010). "Cigarette Smoking and Hypertension." *Current Pharmaceutical Design*, **16**(23), 2518–2525. `doi:10.2174/138161210792062920`.

Warnes GR, Bolker B, Bonebakker L, Gentleman R, Huber W, Liaw A, Lumley T, Maechler M, Magnusson A, Moeller S, Schwartz M, Venables B (2020). **gplots**: *Various R Programming Tools for Plotting Data*. R package version 3.0.4, URL https://CRAN.R-project.org/package=gplots.

Wickham H (2009). **ggplot2**: *Elegant Graphics for Data Analysis*. Springer-Verlag. doi:10.1007/978-0-387-98141-3.

Young SS, Obenchain RL, Lambert C (2015). "Bias and Response Heterogeneity in an Air Quality Data Set." arXiv:1504.00975 [stat.AP], URL https://arxiv.org/abs/1504.00975.

Young SS, Obenchain RL, Lambert CG (2016). "A Problem of Bias and Response Heterogeneity." In A Alan Moghissi, G Ross (eds.), *Standing With Giants: A Collection of Public Health Essays in Memoriam to Dr. Elizabeth M. Whelan*, pp. 153–169. American Council on Science and Health, New York.

Zeileis A, Fisher JC, Hornik K, Ihaka R, McWhite CD, Murrell P, Stauffer R, Wilke CO (2020). "**colorspace**: A Toolbox for Manipulating and Assessing Colors and Palettes." *Journal of Statistical Software*, **96**(1), 1–49. doi:10.18637/jss.v096.i01.

Affiliation:

Nicolas R. Lauve
University of New Mexico
Department of Computer Science
MSC01-1130
1 University of New Mexico
Albuquerque, NM 87131, United States of America
E-mail: nlauve@unm.edu

Stuart J. Nelson
University of New Mexico
Health Sciences Library and Informatics Center
Division of Translational Informatics
Department of Internal Medicine
University of New Mexico Health Sciences Center
MSC09-5100
1 University of New Mexico
Albuquerque, NM 87131, United States of America
E-mail: stuart.james.nelson@gmail.com

S. Stanley Young
CGStat, LLC
Raleigh, NC, United States of America
E-mail: genetree@bellsouth.net

Robert L. Obenchain
Risk Benefit Statistics, LLC
Indianapolis, IN, United States of America
E-mail: wizbob@att.net

Christophe G. Lambert *(corresponding author)*
Center for Global Health
Division of Translational Informatics
Department of Internal Medicine
University of New Mexico Health Sciences Center
MSC10-5550
1 University of New Mexico
Albuquerque, NM 87131, United States of America
E-mail: cglambert@unm.edu
URL: http://christophelambert.org/