PROJECT MUSE®

Visual Pruner: Visually Guided Cohort Selection for Observational Studies

Lauren R. Samuels, Robert A. Greevy Jr.

Observational Studies, Volume 4, Issue 1, 2018, pp. 151-170 (Article)

Published by University of Pennsylvania Press

DOI: <https://doi.org/10.1353/obs.2018.0015>

➔ *For additional information about this article*

<https://muse.jhu.edu/article/793380/summary>

Visual Pruner: Visually Guided Cohort Selection for Observational Studies

Lauren R. Samuels

*Department of Biostatistics
Vanderbilt University School of Medicine
Nashville, TN 37203, USA*

lauren.samuels@vanderbilt.edu

Robert A. Greevy, Jr.

*Department of Biostatistics
Vanderbilt University School of Medicine
Nashville, TN 37203, USA*

robert.greevy@vanderbilt.edu

Abstract

Observational studies are a widely used and challenging class of studies. A key challenge is selecting a study cohort from the available data, or “pruning” the data, in a way that produces both sufficient balance in pre-treatment covariates and an easily described cohort from which results can be generalized. **Visual Pruner** is a free, easy-to-use R **shiny** web application that facilitates both of these goals by letting analysts use updatable linked visual displays to refine a study’s inclusion criteria. By helping analysts see how pre-treatment covariate distributions relate to the estimated probabilities of treatment assignment (propensity scores), the app lets analysts make pruning decisions based on patterns that are otherwise hard to discover. The app yields a set of transparent covariate-based inclusion criteria that can be easily applied in any statistical software package. Thus, while the app is interactive and facilitates iterative decision-making, it can also easily be incorporated into a reproducible-research workflow.

Keywords: study design, propensity scores, interactive graphics, visualization

1. Introduction

1.1 Why prune?

Observational studies, commonly conducted in fields including medicine, public health, epidemiology, economics, and political science, enable researchers to assess the effect of a treatment or exposure in situations where random assignment to study groups would be unethical or infeasible. An important concern in observational studies is whether the groups being studied are comparable in terms of baseline covariates (variables measured before the treatment or exposure) that might affect the outcome of interest. This comparability, or *balance*, has two components: in a perfectly balanced study, the covariate distributions in the two groups overlap completely (have common support) and have the same shape (Ho et al., 2007). Poor balance in pretreatment covariates can lead to bias and model dependence in treatment effect estimates (Ho et al., 2007). Popular approaches to mitigating imbalance in observed baseline covariates include matching on exact covariate values or on a multivariate distance measure; matching, weighting, stratification, or regression adjustment using the

estimated propensity score, or probability of being in the treated group given the observed covariates (Rosenbaum and Rubin, 1983); and statistical modeling using the covariates.

When applied before any of these techniques, *pruning*, or removing some units from the available data to select a smaller study cohort, can further reduce covariate imbalance, leading to less model-dependent (Ho et al., 2007) and thus more convincing results. Pruning is especially helpful in ensuring common support but can also improve the similarity of the distributions in the newly restricted range. Furthermore, pruning can be essential to the development and evaluation of the propensity score model in analyses that use propensity scores (Ho et al., 2007; King and Zeng, 2006), as discussed further below.

1.2 Current pruning methods

Although we present pruning here as a technique to be applied before the use of other methods, some methods for observational studies incorporate pruning implicitly. For example, if the original sample has fewer treated units than control units, one-to-one matching will discard some units from the original sample as part of the matching process itself. While this “implicit pruning” can certainly be successful in producing a sample with better covariate balance, the resulting sample will have been selected mainly by the matching algorithm, rather than by the researcher’s own explicitly-stated inclusion criteria. For studies where the covariate distributions have limited overlap, we join with Ho et al. (2007) and other authors in recommending at least the consideration of some explicit pruning before the use of one-to-one matching and other techniques that algorithmically eliminate study units.

In particular, with the goal of having clearly defined study inclusion criteria, we join with other authors in recommending not only that explicit pruning be considered before proceeding with other techniques, but also that pruning, if necessary, be based on the distribution of the baseline covariates. In contrast, some analysts prune based on the distribution of the estimated propensity scores, eliminating observations outside the region of overlap of the estimated scores before proceeding with matching or weighting. This approach is less than ideal for two reasons. First of all, as King and Zeng (2006) point out, it is hampered by “a fundamental problem of infinite regress: we cannot use the propensity score to identify regions of extrapolation until we can verify that the estimated propensity score is valid, but we cannot verify the validity of the estimated propensity score until we have first removed the regions requiring extrapolation.” Even if this were not a problem, Rosenbaum argues that it may be preferable to select a study cohort based on covariate values rather than on propensity scores: “A population defined in terms of [the propensity score] is likely to have little meaning to other investigators, whereas a population defined in terms of one or two familiar covariates ... will have a clear meaning” (Traskin and Small, 2011; Rosenbaum, 2010, p. 86).

Several purely covariate-based pruning approaches are currently available, yet none is ideal in all cases. The hyper-rectangle approach (Porro and Iacus, 2009) yields a simple, easily understandable set of inclusion criteria, but because of its simplicity and lack of flexibility, in some settings this approach can remove too many subjects on the edges of the restricted covariate space while leaving large areas of imbalance in the interior. The convex-hull approach (King and Zeng, 2006), while leaving fewer large areas of imbalance

in the interior of the covariate space, can also remove too many subjects on the edges and does not result in easily understandable inclusion criteria.

The challenge, then, lies in balancing the simplicity of covariate-based pruning techniques with the more nuanced but less transparent approaches possible with other methods. Two techniques exist to translate more complex inclusion guidelines into simpler terms. Traskin and Small (2011) introduce an approach to generate transparent covariate-based inclusion criteria derived from the optimal cohort selection methods developed by Rosenbaum (2012) (a matching algorithm that prunes implicitly while optimizing the sample size and the sum of the distances between subjects) and Crump et al. (2009) (a propensity-score-based pruning technique that optimizes the precision of the treatment-effect estimator). Traskin and Small’s approach uses classification trees to approximate the cohort selected by one of the optimal methods with a similar cohort that can be described solely in terms of the covariates and their interactions. In recent work, Fogarty et al. (2016) develop a maximal-box approach that can approximate a cohort selected by any prespecified binary inclusion rule with a cohort described by even simpler univariate inclusion criteria than those resulting from the Traskin and Small method, using a small number of covariates selected by the analyst.

1.3 Visual Pruner’s contribution

While the techniques proposed by Traskin and Small (2011) and Fogarty et al. (2016) succeed in producing covariate-based inclusion criteria that can be more nuanced than the hyper-rectangle or convex-hull approaches, to use them the analyst must begin by choosing a “black box” binary inclusion rule or definition of overlap. In some cases, however, analysts would like to have simple inclusion criteria while taking a more flexible approach to the definition of overlap. Rather than beginning with a predetermined binary inclusion rule, the web application **Visual Pruner** allows analysts to visually discover the interplay between the distributions of the covariates and the estimated propensity scores in a dynamic, iterative process, and thus to develop flexible, data-driven, and transparent inclusion criteria.

In addition to its pruning capabilities, **Visual Pruner** has additional features that can be especially valuable in studies where the analyst will go on to use estimated propensity scores in matching or inverse probability weighting (IPW). Austin and Stuart (2015) recommend that analysts using IPW do extensive checking of covariate balance in the weighted sample, and they endorse the advice given by Rosenbaum and Rubin (1984) that development of the propensity score model may need to be an iterative process. Unfortunately, Austin and Stuart (2015) found that many published studies using IPW do not contain any evidence of balance checking or iterative propensity score development. **Visual Pruner** makes it easy for analysts to follow Austin and Stuart’s excellent recommendations for best practice.

Visual Pruner can be readily incorporated into a reproducible-research workflow. After pruning a dataset with **Visual Pruner**, analysts can download files containing the final propensity score model and inclusion criteria and then proceed with the remainder of the statistical analysis as planned. Use of the app can improve both the credibility and the transparency of observational studies.

2. Using the app

The app, freely available at <https://statcomp2.vanderbilt.edu/VisualPruner/>, is built using the R **shiny** framework (R Core Team, 2017; Chang et al., 2017). Analysts running the app from the Web do not need R or its packages. Analysts wishing to run the app on their local machine can either copy the app’s source files, displayed on the app’s R tab and also available at <https://github.com/LaurenSamuels/VisualPruner>, into a local directory and run them with **shiny**’s `runApp()` function, or use **shiny**’s `runGitHub()` function, with arguments `repo= "VisualPruner"`, `username= "LaurenSamuels"`.

Using the app consists of moving (in most cases, iteratively) through five steps, each conducted on a separate tabbed page within the app. The steps are described in detail in the subsections below and summarized in Table 1. Additional information is contained in Notes sections at the bottom of some of the tabbed pages, as well as on the app’s About tab.

2.1 Tab 1 (“Upload”): Upload data

Visual Pruner accepts datasets in .csv or .rds format. After the analyst uploads a file, the app displays the number of columns and rows as a means of verification. The app then presents a list of all dichotomous variables in the dataset (numeric or character), and the analyst chooses one to use as the treatment indicator. The app displays the levels of the chosen treatment indicator.

Rather than uploading a dataset, new users may prefer to select the “Use example dataset” radio button. An example dataset consisting of 1000 observations of seven variables (a treatment indicator and six pre-treatment covariates) will then be generated. This dataset will be used for illustration in Section 4.

2.2 Tab 2 (“Specify”): Specify a propensity-score estimation method and model

As a first step on the Specify page, the analyst specifies how the app should handle missing values in fitting the propensity score model. The default is to do a single imputation, replacing any missing values with the univariate mean or mode for purposes of model fitting. This approach, together with the use of missingness indicators (see below), can result in unbiased treatment effect estimates under certain conditions; see Cham and West (2016) and Rosenbaum (2010, pp. 193–194, 240–242) for more details. Alternatively, the analyst can choose to estimate propensity scores for only those subjects who have no missing values for the variables used in the model. Regardless of which option is chosen, the choice applies to only those variables used in the propensity score model, and it applies only during the process of model fitting. No pruning of the dataset is done at this stage.

The app currently offers two methods of propensity-score estimation: logistic regression and probit regression, both conducted using the appropriate functions from the **rms** package (Harrell Jr, 2018). Analysts with R programming experience may wish to add other estimation methods; instructions for doing so are in Appendix A.

After choosing an estimation method, the analyst specifies the right-hand side of a propensity score model. The app displays a list of the variables in the dataset for conve-

Table 1: Summary of **Visual Pruner**’s seven tabbed pages.

Tab	Key Actions	Notes
Upload (§ 2.1)	Upload data Select treatment indicator	Accepted formats: .csv, .rds Click to confirm/update selection
Specify (§ 2.2)	Choose method for handling missingness in model-fitting Choose propensity-score estimation method Specify right-hand side of propensity-score model	
Prune (§ 2.3)	Choose variables to view and restrict Choose cutoff for discreteness Modify graphical parameters (optional) Brush propensity-score histogram (optional) View displays for selected covariates Make pruning decisions	Plots will not appear until variables have been selected and “(Re-)make graphs ...” button has been clicked. Click one or both pruning buttons (above first covariate display) to apply changes.
Compare (§ 2.4)	View SMD plot	
Download (§ 2.5)	Download or copy final inclusion criteria and propensity score model	
About	Read more about the app	
R	See R session information and source code	

nience. The model formula can use the full range of base R and **rms** features, including interactions and restricted cubic splines. (Users unfamiliar with R can find a guide to R formulas at <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html>. A link to the guide is also available in the app. For more information on the types of splines available in the **rms** package, see <https://cran.r-project.org/web/packages/rms/rms.pdf>.) If the analyst has chosen to impute values for variables with missing values, the app automatically generates missingness-indicator variables that can be included in the propensity score formula (here “missingness” refers to the state in the original dataset, before imputation). Dynamically generated directions for using the new variables will appear in the app.

After the analyst clicks the “I have finished typing” button, the app checks the basic syntax of the formula and the validity of the variable names and then attempts to fit the model. If the model can be fit, the app displays histograms of the estimated propensity scores on both the probability and the log-odds scale; while the probability scale is often intuitively easier to grasp, the log-odds scale is more often used in matching applications and allows enhanced viewing of the tails of the distribution. Below the plots, the app displays summary information from the propensity-score estimation procedure.

2.3 Tab 3 (“Prune”): Refine inclusion criteria

The Prune page consists of an upper section and a lower section, separated by two buttons that perform the actual pruning. The upper section contains general controls and displays, while the lower section contains individual controls and displays for each of the selected covariates. In the first step in the upper section, the analyst must choose variables to view and restrict; in most cases, analysts will want to select at least those variables used in the propensity score model, but this is not required. The analyst’s next choice involves the display of discrete numeric variables. By default, when the covariate graphs are produced, numeric variables will be displayed as discrete variables if they have fewer than ten unique values in the original dataset. After selecting variables and possibly modifying the cutoff for discreteness, the analyst clicks the “(Re-)make graphs using updated variable list and/or discreteness preferences” button. Two other graphical preferences can be modified here: the opacity (alpha) and the plotting symbol size. These (as well as the other two) can be updated at any time; the only difference is that the first two require the click of the “(Re-)make graphs...” button in order to take effect.

In addition to these controls, the upper portion of the Prune page has two key displays: a sample-size table, and a brushable histogram of the estimated propensity scores. Analysts can brush a portion of this plot, and the units that have propensity scores within that range will be highlighted in the covariate plots below. This plot also contains the legend that is used for all plots on the page. The logit scale is used for this plot as well as for the propensity scores in the plots displayed in the lower section of the page.

The bottom part of the Prune page contains a section for each of the variables selected for viewing and restriction. Each section consists of a multi-panel plot, a pruning tool, and a percent-missing table. For continuous variables, the main panel in the multi-panel plot is a scatterplot, with the covariate values on the x-axis, the estimated propensity scores on the y-axis, and a loess curve. Above the main panel is a histogram of the covariate.

If, on the Specify page, the analyst chose not to estimate propensity scores for units with missing values for the variables in the propensity score model, the histogram may contain observations that are not included in the scatterplot below; if this is the case, a message will appear. To the right of the main panel is a stripchart showing the propensity score values for units with missing values for this particular covariate, if applicable. The barchart in the upper right panel displays the counts of missing values by group. For discrete variables, the layout of the multipanel plot is similar, except that the main scatterplot is replaced by a stripchart, and the histogram above the main panel is replaced by a barchart. Rather than showing a loess smooth, the stripcharts for discrete variables contain horizontal lines marking the mean propensity score value within each level.

In addition to the multi-panel plot, the section for each variable contains a two-part pruning tool and a percent-missing table. The first part of the pruning tool consists of either text boxes for the maximum and minimum (for continuous variables) or checkboxes for each level (for discrete variables). The second part is a choice of whether to keep (the default) or exclude units with missing values for the variable. The analyst can make pruning decisions for one variable at a time, or for several variables at once, restricting each one to an appropriate range or set of values. In order for the pruning decisions to take effect, the analyst must click one of the two buttons that separate the top part of the Prune page from the bottom part. The “Update covariate graphs to reflect pruning choices” button applies the pruning decisions, then updates the covariate graphs to reflect the pruned sample; in addition, the sample-size table at the top of the page will also be updated. The “Recalculate PS for pruned sample (will also update all graphs)” button will first apply the pruning criteria, then re-estimate the propensity scores on the pruned sample, then update the graphs and the sample-size table. Clicking this second button is the approach we recommend, but the first one may be useful as part of an incremental approach with larger datasets. Pruning decisions can be modified multiple times, and the analyst can also go back to the Specify tab and change the form of the propensity score model multiple times based on insights gained from the plots and tables.

2.4 Tab 4 (“Compare”): View the standardized mean difference plot

While the Prune tab provides a detailed graphical display for each selected covariate, the Compare tab shows a condensed summary of one aspect of the covariate balance between the two groups, in the form of an absolute standardized mean difference (SMD) plot for the selected variables. The absolute standardized mean differences, calculated here using the **tableone** package (Yoshida and Bohn, 2018), are the absolute differences in the group means, scaled by a sample-size-neutral pooled standard deviation (Flury and Riedwyl, 1986; Austin, 2009b; Yang and Dalton, 2012). The dotted vertical line at 0.1 marks a degree of imbalance that some researchers consider to be unacceptable (Austin, 2009a).

Unlike the plots on the Prune tab, the initial SMD plot on the Compare tab can be viewed before initial specification of the propensity score model. Analysts may wish to consult the Compare tab before specifying an initial propensity score model and before making pruning decisions, and then after making changes to either the model or the pruning criteria. As on the Compare tab, analysts must select variables to be displayed on this page as a first step. Before specification of a propensity score model and initial pruning,

the SMD plot contains a set of points (optionally connected by a line for clarity) for the original sample, as well as an identical set for the pruned sample; after pruning (once the “Update...” or “Recalculate...” button is clicked on the Prune page), the values for the pruned sample will be updated.

Analysts who are considering conducting a weighted analysis can also choose to add points for three types of weighted samples to the SMD plot, once propensity scores have been estimated: ATE weighting, used to estimate the average treatment effect on a sample or population that resembles the pruned sample; ATT weighting, used to estimate the average treatment effect on a sample or population that resembles the treated units in the pruned sample, and ATM weighting, used to estimate the average treatment effect on a sample or population that resembles the evenly matchable units from both groups in the pruned sample. ATM weighting uses weights introduced as “matching weights” by Li and Greene (2013); the weights are discussed further in a forthcoming manuscript (Samuels and Greevy, Jr., 2018). All weighting is conducted using the **survey** package (Lumley, 2017, 2004).

In general it is important to consider standardized mean differences for higher-order terms and interactions (Austin and Stuart, 2015), as well as for missingness indicators. We hope to add automatic generation and inclusion of these variables in the future, but in the meantime we recommend that analysts add them to their datasets manually before importing so that they can be viewed within the app.

2.5 Tab 5 (“Download”): Download the inclusion criteria and propensity score model

The analyst can iterate through the steps of specifying a propensity score model, pruning the dataset, and viewing the standardized mean difference plot many times and in any order after the initial model specification and selection of variables to view. Ideally, at some point the dataset will have a degree of covariate balance that the analyst finds acceptable.

Rather than providing a final version of the dataset, **Visual Pruner**’s Download page provides downloadable (or copiable) R code for the study’s inclusion criteria and the final propensity score model and function call, in order to facilitate reproducible research. The R code used here is simple enough that users of any statistical software should be able to understand it and translate it if necessary.

3. After using the app

In some cases, pruning according to the criteria downloaded from **Visual Pruner** will be sufficient to create treatment groups that have good balance on all observed covariates, in terms of both the overlap and the shape of the distributions. In most cases, however, after the analyst applies the inclusion criteria to the data, the groups will not be perfectly balanced. The pruning can then be followed by matching or weighting on the propensity score (estimated on the pruned sample, using the final formula downloaded from the app) or by other methods, alone or in combination with statistical modeling (Ho et al., 2007). Note that for analyses using propensity score estimation, if the estimation process used within the app relied on the imputation of missing values, the values will need to be re-imputed before or as part of the propensity-score estimation process. Similarly, if the final propensity

score model contains missing-value indicators, these variables will need to be added into the pruned dataset before propensity-score estimation if they were not already present in the data.

4. Illustrations

In this section we show several plots from the app, using the example dataset included with the app for illustration. The treatment indicator variable in this dataset is **exposed**, with possible values “Yes” (300 subjects) and “No” (700 subjects). Six baseline covariates are also included: **age** (age in years), **height_ft** (height in feet), **systolicBP** (systolic blood pressure in mmHg), **gender**, (gender: female/male/other), **smoker** (smoking status: current/former/never), and **ABO** (ABO blood type: A/B/AB/O). Some of the variables have missing values.

4.1 Illustrations from the Prune tab

Our first illustrations come from the Prune tab. Prior to visiting the Prune tab, on the Specify tab we left unchanged the default methods for handling missing values and propensity-score estimation, and we specified the following right-hand side for our initial propensity score model: **age + height_ft + systolicBP + gender + smoker + ABO**. The first plot shown on the Prune tab is the plot of estimated propensity scores, Figure 1. Although this plot is brushable in the app (and linked to the other plots on the Prune tab), we do not illustrate this feature here.

Figure 1: Estimated logit propensity scores from the original model, as shown in the Prune tab. The legend used for this plot, showing “No” and “Yes” for the levels of the treatment indicator variable, **exposed**, is the legend used for the plots that follow as well.

A section for each selected variable, consisting of a graphical display, a pruning tool, and a percent-missing table, appears below the propensity score plot on the Prune tab. Figure 2 shows a screenshot of the full section for age. In Figures 3–5, we zoom in on the graphical displays from the sections for the continuous variables in the dataset, before any pruning or any modification of the propensity score model, and give possible interpretations and suggested actions. Figure 6 does the same for one of the categorical variables. Section 2.3 has more information about these displays in general.

Figure 2: Screenshot of the full section for **age** (age in years) from the Prune tab, before any pruning or any modification of the propensity score model. The left half of the section contains the graphical display, and the right half contains the two-part pruning tool (top) and the percent-missing table (bottom). In Figure 3 we zoom in on the graphical display from this screenshot and give possible interpretations and suggested actions.

Figure 3: First graphical display for **age** (age in years), from the Prune tab. The screenshot in Figure 2 gives the visual context for this display. Both the scatterplot and the histogram show a lack of overlap, as well as a difference in the shapes of distributions within the region of overlap: the subjects in the exposed group (orange) are in general older than the subjects in the unexposed group (blue). The scatterplot and the loess curve also show the strong relationship between age and the estimated propensity scores. Analysts unwilling to extrapolate into regions containing subjects from only one group can use the app’s pruning tool to restrict the study’s inclusion criteria to a narrower range of ages, for example 30–60. Note, however, that in contrast to **height.ft** (Figure 4), where simple pruning can completely balance the groups, **age** will still be imbalanced even after use of the app’s pruning tool, due to the difference in the shapes of the distributions in the two groups. The empty panels above the label “Missing” show that there are no subjects with missing values for **age**.

Figure 4: First graphical display for `height_ft` (height in feet), from the Prune tab. Both the scatterplot and the histogram show a lack of overlap, with the exposed group (orange) having a much narrower range than the unexposed group (blue), but we see similarly shaped distributions within the region of overlap. Note that the initial propensity score model, in which height is included using a single linear term, does a very poor job of estimating the probability of treatment for subjects in the outer portions of the range of heights. Analysts unwilling to extrapolate into regions with no exposed subjects can use the app’s pruning tool to restrict the study’s inclusion criteria to a narrower range of heights, for example 5.25 ft–6 ft. The loess curve suggests that within the region of overlap, height has only a weak association with the estimated propensity scores. The panels on the right, together with the percent-missing table for this variable (not shown), suggest that for this variable, missingness does not differ by treatment status.

Figure 5: First graphical display for `systolicBP` (systolic blood pressure, in mmHg), from the Prune tab. For this variable, both the scatterplot and the histogram show good overlap and similarly shaped distributions. The loess curve suggests that there is little to no relationship between this variable and the estimated propensity scores. Looking at the missing-value plots on the right, together with the percent-missing table (not shown), we see that the proportion of subjects missing this variable in the exposed group (orange) is much higher than that in the unexposed group (blue). As a next step, we recommend returning to the Specify tab and adding a missingness-indicator variable for `systolicBP` into the model.

Figure 6: First graphical display for **smoker** (smoking status), from the Prune tab. We see in both the main stripchart and the main barchart that smoking status differs heavily by exposure group. Most noticeably, we see that there are only three current smokers in the exposed group (orange). In deciding whether to exclude current smokers from the study, an analyst would need to weigh the loss of power from eliminating these subjects against the risks inherent in making inferences about the effect of exposure in current smokers when there are so few current smokers in the exposed group.

4.2 Illustrations from the Compare tab

As discussed in Section 2.4, analysts may wish to alternate consulting the detailed covariate displays on the Prune tab and the condensed summary display on the Compare tab. Figure 7 shows the first absolute standardized mean difference (SMD) plot for the original sample from the Compare tab. Figure 8 shows one possible SMD plot after updating the propensity score model and trying some initial pruning.

Figure 7: Absolute standardized mean difference (SMD) plot for the original sample, from the Compare tab. We have chosen to connect the points with a line (checkbox not shown); some analysts may prefer to view the plot without this option. As noted in Figures 3 and 6, there are large imbalances in **age** and **smoker**; **gender** is quite imbalanced as well. The absolute SMDs for these variables are greater than 0.1, the value marked on the plot by the dashed grey line and considered to be the highest acceptable level of imbalance by some researchers. Although the absolute SMDs for **height_ft**, **systolicBP**, and **ABO** are less than 0.1, it is important to note that the SMD plot gives no indication of the extreme lack of overlap in **height_ft** or of the informative missingness in **systolicBP**, both of which can be seen in the more detailed plots on the Prune tab (Figures 4 and 5).

Figure 8: Absolute standardized mean difference (SMD) plot for the sample after updating the propensity score model and trying some initial pruning, from the Compare tab. After taking note of the informative missingness in `systolicBP` (see Figure 5), we returned to the Specify tab and updated the right-hand side of the propensity score model to include a missingness indicator for that variable: `age + height_ft + systolicBP + gender + smoker + ABO + is.na.systolicBP`. In addition, we pruned `age` to restrict the sample to subjects between the ages of 30 and 60, and we pruned `height_ft` to restrict the sample to subjects with heights in the range 5.25 ft–6 ft. This initial pruning reduced the sample to 224 exposed and 536 unexposed, for a total of 760 subjects. In addition to the points for the original sample (red circles, solid lines), the SMD plot now contains points for the pruned sample (blue squares, dashed line) and points for the ATM-weighted pruned sample (orange triangles, long-dashed line). Although we have chosen to connect the points from each sample with a line, some analysts may prefer to view the plot without this option (checkbox not shown). We see that the pruning improved the balance in terms of SMDs only slightly, suggesting that to get unbiased treatment effect estimates, we will want to follow the pruning with matching or weighting. We see that the ATM-weighted sample would have very good balance in terms of SMDs. Depending on the priorities for analysis, a next step might be loosening the restrictions on `age` or `height` to see whether we can still achieve good SMDs while keeping the sample a little bigger.

5. Discussion

With **Visual Pruner**, pruning decisions are informed by the propensity score distribution but not determined by it. Because the app allows iterative development of the propensity score model as well as easy re-estimation and reinspection of the estimated propensity scores after each pruning decision, it facilitates covariate-based cohort definition enhanced by extra information from the estimated propensity scores. As long as the analyst does not include the outcome variable in either the propensity score model or the set of variables to view and restrict, this iterative process is impervious to design bias from cherry-picking a cohort that yields a desired outcome. Observational studies in which the analyst has used **Visual Pruner** to assist with cohort selection can be both more convincing, because of the improved covariate balance and reduction in model dependence, and more generalizable, because of the simple univariate inclusion criteria.

As noted in Section 1.2, we recommend against making pruning decisions solely based on estimated propensity scores. We recognize, however, that there are differences of opinion on this issue. While **Visual Pruner** does not offer direct pruning based on an iteratively-developed propensity score, some analysts may wish to fit an initial propensity score model prior to uploading data to the app, and then save the predicted values as an additional column in the dataset. The analyst can then prune directly on this preliminary propensity score as if it were any other covariate.

In future versions of the app, the authors hope to add the ability to handle multiple treatment groups, as well as a default pruning feature implementing a modified version of one or more algorithmic pruning methods (such as those discussed in Section 1.2), for use as a starting place or comparison. In addition, while the app has benefitted greatly from the enhanced speed provided by the **data.table** (Dowle and Srinivasan, 2017) and **dplyr** (Wickham et al., 2017) packages, we would like to continue improving its speed with large datasets. We welcome feedback from researchers and analysts about other ways **Visual Pruner** could be improved.

Acknowledgments

The authors wish to thank Dale Plummer and the Department of Biostatistics, Vanderbilt University School of Medicine, for hosting **Visual Pruner**.

Portions of this paper first appeared in Poster Abstracts of IEEE VIS 2015. The authors are grateful to Meira Epplein, Bryan Shepherd, Matt Shotwell, and two anonymous reviewers for their comments and suggestions on the current manuscript. They and several other people, listed on the app’s About tab, have also contributed valuable suggestions on the app itself.

Appendix A

In this appendix we describe the procedure for adding new estimation methods as options on the Specify page.

1. In the file `ui.R`, look for the following code, which sets up the radio buttons for choosing an estimation method on the Specify page:

```
radioButtons("psMethod", NULL,
  c(
    "Logistic regression" = 0,
    "Probit regression" = 1
  ),
  0
)
```

and modify it as follows:

```
radioButtons("psMethod", NULL,
  c(
    "Logistic regression" = 0,
    "Probit regression" = 1,
    "<your method name here>" = 2
  ),
  0
)
```

2. In the file `server.R`, look for the following code, which translates the numeric value from the radio buttons into a short phrase used in later steps:

```
psFitMethod <- reactive({
  if (input$psMethod == 0) {
    "logistic"
  } else if (input$psMethod == 1) {
    "probit"
  } else {
    NA
  }
})
```

and modify it as follows:

```
psFitMethod <- reactive({
  if (input$psMethod == 0) {
    "logistic"
  } else if (input$psMethod == 1) {
    "probit"
  }
})
```

```

    } else if (input$psMethod == 2) {
      "<short version of your method name here>"
    } else {
      NA
    }
  })
})

```

3. In the file `psFunctions.R`, modify the function `getPSCall()`, which sets up the call to the selected estimation function, as follows:

```

getPSCall <- function(Method, form) {
  if (Method == "logistic") {
    mycall <- call("lrm", form, data= quote(dat))
  } else if (Method == "probit") {
    mycall <- call("Glm", form, data= quote(dat),
      family= quote(binomial(link= "probit")))
  } else if (Method == "<short name from psFitMethod in previous step>") {
    mycall <- call("<function name here>",
      <arguments to your function here,
      following the pattern for "logistic" and "probit" above>)
  } else {
    mycall <- NULL
  }
  return(mycall)
}

```

The call will be executed by the `psFit()` reactive function in `server.R`.

4. In the file `psFunctions.R`, similarly modify the function `getPS()`, which extracts the fitted values from the estimation procedure.
5. In the file `psFunctions.R`, similarly modify the function `showPSSummary()`, which produces the summary information shown at the bottom of the Specify page.

References

- Austin, P. C. (2009a). Balance Diagnostics for Comparing the Distribution of Baseline Covariates between Treatment Groups in Propensity-Score Matched Samples. *Statistics in Medicine*, 28(25):3083–3107.
- Austin, P. C. (2009b). Using the standardized difference to compare the prevalence of a binary variable between two groups in observational research. *Communications in Statistics-Simulation and Computation*, 38:1228–1234.

- Austin, P. C. and Stuart, E. A. (2015). Moving Towards Best Practice When Using Inverse Probability of Treatment Weighting (IPTW) Using the Propensity Score to Estimate Causal Treatment Effects in Observational Studies. *Statistics in Medicine*, 34(28):3661–3679.
- Cham, H. and West, S. G. (2016). Propensity Score Analysis with Missing Data. *Psychological methods*, 21(3):427–445.
- Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2017). *shiny: Web Application Framework for R*. R package version 1.0.5. Available from: <https://CRAN.R-project.org/package=shiny>.
- Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. (2009). Dealing with Limited Overlap in Estimation of Average Treatment Effects. *Biometrika*, 96(1):187–199.
- Dowle, M. and Srinivasan, A. (2017). *data.table: Extension of ‘data.frame’*. R package version 1.10.4-3. Available from: <https://CRAN.R-project.org/package=data.table>.
- Flury, B. K. and Riedwyl, H. (1986). Standard distance in univariate and multivariate analysis. *The American Statistician*, 40(3):249–251.
- Fogarty, C. B., Mikkelsen, M. E., Gaieski, D. F., and Small, D. S. (2016). Discrete Optimization for Interpretable Study Populations and Randomization Inference in an Observational Study of Severe Sepsis Mortality. *Journal of the American Statistical Association*, 111(514):447–458.
- Harrell Jr, F. E. (2018). *rms: Regression Modeling Strategies*. R package version 5.1-2. Available from: <https://CRAN.R-project.org/package=rms>.
- Ho, D. E., Imai, K., King, G., and Stuart, E. A. (2007). Matching As Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference. *Political analysis*, 15(3):199–236.
- King, G. and Zeng, L. (2006). The Dangers of Extreme Counterfactuals. *Political analysis*, 14(2):131–159.
- Li, L. and Greene, T. (2013). A Weighting Analogue to Pair Matching in Propensity Score Analysis. *The International Journal of Biostatistics*, 9(2):215–234.
- Lumley, T. (2004). Analysis of complex survey samples. *Journal of Statistical Software*, 9(1):1–19. R package version 2.2.
- Lumley, T. (2017). **survey**: Analysis of complex survey samples. R package version 3.32.
- Porro, G. and Iacus, S. M. (2009). Random recursive partitioning: A matching method for the estimation of the average treatment effect. *Journal of Applied Econometrics*, 24(1):163–185.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Available from: <https://www.R-project.org/>.

- Rosenbaum, P. R. (2010). *Design of Observational Studies (Springer Series in Statistics)*. Springer.
- Rosenbaum, P. R. (2012). Optimal Matching of an Optimally Chosen Subset in Observational Studies. *Journal of Computational and Graphical Statistics*, 21(1):57–71.
- Rosenbaum, P. R. and Rubin, D. B. (1983). The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika*, 70(1):41–55.
- Rosenbaum, P. R. and Rubin, D. B. (1984). Reducing Bias in Observational Studies Using Subclassification on the Propensity Score. *Journal of the American Statistical Association*, 79(387):516–524.
- Samuels, L. R. and Greevy, Jr., R. A. (2018). The Average Treatment Effect on the Evenly Matchable Units (ATM): A Valuable Estimand in Causal Inference. Under review.
- Traskin, M. and Small, D. S. (2011). Defining the Study Population for an Observational Study to Ensure Sufficient Overlap: A Tree Approach. *Statistics in Biosciences*, 3(1):94–118.
- Wickham, H., Francois, R., Henry, L., and Miller, K. (2017). *dplyr: A Grammar of Data Manipulation*. R package version 0.7.4. Available from: <https://CRAN.R-project.org/package=dplyr>.
- Yang, D. and Dalton, J. E. (2012). A unified approach to measuring the effect size between two groups using SAS®. *SAS Global Forum*.
- Yoshida, K. and Bohn, J. (2018). *tableone: Create ‘Table 1’ to Describe Baseline Characteristics*. R package version 0.9.2. Available from: <https://CRAN.R-project.org/package=tableone>.