Contributions to Missing Data Methods in Single-Cell Genomics and Survival Analysis

Citation

Permanent link
http://nrs.harvard.edu/urn-3:HUL.InstRepos:42029653

Terms of Use
This article was downloaded from Harvard University’s DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Share Your Story
The Harvard community has made this article openly available. Please share how this access benefits you. Submit a story.

Accessibility
Contributions to Missing Data Methods in Single-Cell Genomics and Survival Analysis

a dissertation presented
by
Samuel Tracy
to
The Department of Biostatistics
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Biostatistics

Harvard University
Cambridge, Massachusetts
May 2019
Contributions to Missing Data Methods in Single-Cell Genomics and Survival Analysis

Abstract

Missing data occurs when individual data values are not recorded for an observation of interest within a sample. Such events may significantly bias subsequent analyses if ignored. This dissertation discusses solutions to inference and prediction in the presence of data missing at random relevant to single-cell genomics and survival analysis. Chapter 1 presents a computational method to mitigate technical bias due to capture efficiency in single-cell RNA-sequencing data. Framing incorrectly observed zero gene expression values as a missing at random problem, we impute the zero values using information from samples with similar expression patterns. By comparative analysis of simulated and real single-cell RNA-seq datasets, we outperform existing methods in terms of imputation accuracy and increase the precision of cell-type identification. In Chapter 2, we develop a three-step algorithm to infer upon unmeasured spatial patterns of gene expression through the integration of single-cell RNA-seq and sequential fluorescence in situ hybridization data. Through analysis of mouse visual cortex data, we show that this is a useful tool for predicting the spatial pattern of cell-type and domain-specific genes. Chapter 3 presents minimal-area confidence bands for time-to-event functions using a related optimization problem with local time processes. Some event times are unobserved, or censored, resulting in partial missingness. We assume that the censoring mechanism is independent of event time given the observed information, which is analogous to missing at random. The finite-sample performance of the proposed method is assessed by simulation studies and then applied to clinical trial data to evaluate survival times for primary biliary cirrhosis patients treated with D-penicillamine.
# Contents

1 RESCUE: imputing dropout events in single-cell RNA-sequencing data .................................................. 1
   1.1 Introduction ........................................................................................................................................... 2
   1.2 Results ................................................................................................................................................ 2
      1.2.1 Overview of the RESCUE method .................................................................................................. 2
      1.2.2 RESCUE recovers under-detected expression in simulated data ................................................. 4
      1.2.3 RESCUE recovers differential expression across mouse cell types .............................................. 7
      1.2.4 RESCUE improves cell-type classification of mouse cells ............................................................ 9
   1.3 Discussion ............................................................................................................................................. 11
   1.4 Methods ............................................................................................................................................... 12
      1.4.1 Simulating single-cell RNA-sequencing data .................................................................................... 12
      1.4.2 Mouse Cell Atlas data and processing ............................................................................................ 12
      1.4.3 Generating dropout events ............................................................................................................. 12
      1.4.4 Mathematical details of RESCUE .................................................................................................. 13
      1.4.5 Analysis with scImpute and DrImpute ............................................................................................ 14
      1.4.6 Evaluation of clustering outcomes and marker genes ...................................................................... 14

2 SERA: inferring spatial patterns of gene expression through integrating single-cell RNA-seq and sequential fluorescence in situ hybridization data ............................................................... 16
   2.1 Introduction ............................................................................................................................................ 17
   2.2 Results .................................................................................................................................................. 18
      2.2.1 Overview of the method .................................................................................................................. 18
      2.2.2 SERA captures expression patterns for hundreds of genes in seqFISH data .................................... 20
      2.2.3 SERA captures the expression pattern of cell-type specific genes ................................................ 22
   2.3 Discussion ............................................................................................................................................. 24
   2.4 Methods ............................................................................................................................................... 24
      2.4.1 Imaging data pre-processing ........................................................................................................... 24
      2.4.2 scRNA-seq data pre-processing ...................................................................................................... 25
      2.4.3 Comparative analysis ..................................................................................................................... 25
      2.4.4 Mathematical details of the algorithm ............................................................................................. 25
      2.4.5 Evaluating estimates of expression ................................................................................................. 27

3 OptBand: optimal confidence bands for functions to characterize time-to-event distributions ............... 28
   3.1 Introduction ............................................................................................................................................ 29
   3.2 Methods ............................................................................................................................................... 31
      3.2.1 Functional approximation of the critical value parameter ............................................................. 31
      3.2.2 OptBand for the Cumulative Hazard Function .............................................................................. 32
      3.2.3 OptBand for the Survival Function .................................................................................................. 33
   3.3 Results .................................................................................................................................................. 34
      3.3.1 Simulation ........................................................................................................................................ 34
      3.3.2 Primary biliary cirrhosis data analysis ............................................................................................. 36
   3.4 Discussion ............................................................................................................................................. 39

Appendix A Supplementary material to accompany Chapter 1 .................................................................... 40

Appendix B Supplementary material to accompany Chapter 2 .................................................................... 50
### Appendix C  Supplementary material to accompany Chapter 3

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Minimal-area bands from Kendall et al. (2007)</td>
<td>55</td>
</tr>
<tr>
<td>C.2 Derivation of OptBand for the Survival Function</td>
<td>57</td>
</tr>
</tbody>
</table>

### References

63
List of figures

1.1 A motivation of the RESCUE imputation pipeline illustrated with a hypothetical example of simulated data. (a) Heatmap of a log-transformed normalized expression matrix with cell type clustering affected by dropout. (b) t-SNE visualizations of cell clusters determined with the principle components of many subsamples of informative genes and a histogram showing the bootstrap distribution of the within-cluster non-zero gene expression means for one missing expression value in the data set. (c) Heatmap of the expression data after imputing zero values with a summary statistic of the bootstrap distributions. 3

1.2 Estimation bias after imputing simulated data (Table A; Primary). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout. 5

1.3 Data visualization and cell-type clustering before and after imputing simulated data (Table A; Primary). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results. 6

1.4 Estimation bias and recovery of differential expression after imputing the MCA uterus tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout. 8

1.5 Data visualization and cell-type clustering before and after imputing the MCA data. (a) t-SNE visualization of the original uterus tissue data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results for all four tissue types. 10

2.1 Motivation and visualization of the method concept. (a) A scatter plot of the seqFISH data by spatial-coordinates lacks cell-type information. (b) A principal component plot of the scRNA-seq data is labeled by cell-type but lacks spatial information. (c) A cartoon depicting the disjoint expression matrices of the two data sets illustrates how the data sets might be integrated. 18

2.2 Final predictive results for the seqFISH data. (a) Heatmaps of the observed and estimated expression patterns for the poorly estimated Rnf165 gene and the accurately estimated Ptb gene. (b) Relative sum-of-squares and correlation score distributions. (c) A diagnostic plot of the decision rule threshold. 21

2.3 Final predictive results for 138 marker genes determined in the original seqFISH study. (a) Heatmaps of the observed and estimated expression pattern annotated by the cell types specific to each gene. (b) Relative sum-of-squares and correlation score distributions. 23

3.1 OptBand shows improvement in area over the HW and EP bands with regards to the estimated survival functions when applied to the PBC trial data. 37
3.2 OptBand was applied to the placebo (a) and then the D-penicillamine (b) group of the PBC trial data to compute 95% CB’s for the cumulative hazard functions. In addition, we are able to compute 95% CB’s for the difference in cumulative hazard functions (c) between the two groups. These bands suggest no difference in the cumulative hazard functions between the two groups.

A.1 Estimation bias after imputing simulated data (Supplemental Table 1; Scenario 2). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout.

A.2 Data visualization before and after imputing simulated data (Supplemental Table 1; Scenario 2). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results.

A.3 Estimation bias after imputing simulated data (Supplemental Table 1; Scenario 3). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout.

A.4 Data visualization before and after imputing simulated data (Supplemental Table 1; Scenario 3). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results.

A.5 Estimation bias after imputing the MCA bladder tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.

A.6 Estimation bias after imputing the MCA lung tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.

A.7 Estimation bias after imputing the MCA pancreas tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.

A.8 Data visualization before and after imputing the MCA bladder tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.
A.9 Data visualization before and after imputing the MCA lung tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.

A.10 Data visualization before and after imputing the MCA pancreas tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.

A.11 Data visualization and clustering results before and after dropout in the MCA bladder tissue. (a) t-SNE visualization of the original uterus tissue data labeled by estimated clusters. (b) t-SNE after dropout. (c) Similarity measures in clustering outcomes show little difference between the two data sets due to over-clustering.

B.1 SERA results for the seqFISH validation data. (a) Relative sum-of-squares and correlation score distributions for a multi-response LASSO prediction model applied to the seqFISH data without first quantile normalizing the scRNA-seq data indicates a platform effect. (b) Results after initial quantile-normalization of the scRNA-seq data mitigate the platform effect. (c) Visualizing the decision rule for gene estimability. The local regression mean is indicated by the solid line, the upper quantile of estimate standard deviations by the dashed line.

B.2 Quantile-quantile plots of the z-score standardized gene expression vectors for the highest performing genes from SERA results for the seqFISH validation data, ranked by their final estimated-observed gene expression vector correlation from high to low. Differences in the initial z-score expression distributions do not appear to affect the quality of the final estimates derived from the quantile-normalized data.

B.3 Exploratory analysis of the estimates and model fit when trained upon both the seqFISH validation data and scRNA-seq data indicate association of the estimate-observed expression correlation with the standard deviation of predicted expression vectors, as well as with the $L_1$ norm and $L_2$ norm of the corresponding model coefficients.

B.4 Final predictive results for a leave-one-out cross-validation analysis of the 112 genes shared between the seqFISH and scRNA-seq data. (a) Heatmaps of the observed and estimated expression pattern annotated by the cell types and domain types specific to each gene. (b) Relative sum-of-squares and correlation score distributions. (c) A diagnostic plot of the decision rule threshold. The local regression mean is indicated by the solid line, the upper quantile of estimate standard deviations by the dashed line.

B.5 Gene-gene correlation patterns in (a) the full 112 gene subset of the seqFISH data, (b) the estimable gene subset determined by SERA of the seqFISH data and (c) the estimable gene subset determined by SERA of the scRNA-seq data.

C.1 Simulated coverage in terms of $\kappa$ (vertical axis) and $L$ (horizontal axis) suggests a smooth relationship (a) that is closely matched by a functional approximation (b).
List of tables

3.1 Simulation results (actual coverage, relative area) for $n = 1000$. 35
A.1 Splatter simulation parameters. 40
A.2 Significant differentially expressed genes undetected after dropout. 40
Acknowledgments

In addition to what follows in this dissertation, I have learned much along the way through the mentorship of incredible scholars. I am deeply grateful to my advisor Dr. Guo-Cheng Yuan for his support, teaching, and guidance while pursuing my thesis. Sincere thanks also goes to Committee Members Dr. Giovanni Parmigiani and Dr. Franziska Michor, whose depth and breadth of knowledge I aspire to and whose thoughtful questions trained me to critically analyze the scientific questions driving each development written here.

My path to this point would not have been possible without the faith and support of my undergraduate advisor Dr. Ronghui Xu, or my community college professors Ken Johnson and Daniel Balaguy. Another special note of thanks to Dr. Nan Laird, who has been very supportive of my development as a statistician here at Harvard, and Jelena Follweiler, who has been a beacon of encouragement for us all.

Finally, my family, friends, roommates and colleagues have supported me throughout my time as a student, and I am very thankful.
1

RESCUE: imputing dropout events in single-cell RNA-sequencing data

Samuel Tracy\textsuperscript{1,2}, Guo-Cheng Yuan\textsuperscript{1,2}, Ruben Dries\textsuperscript{2}

\textsuperscript{1} Department of Biostatistics, Harvard T. H. Chan School of Public Health, Boston, MA, USA
\textsuperscript{2} Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA, USA

Abstract

Single-cell RNA-sequencing technologies provide a powerful tool for systematic dissection of cellular heterogeneity. However, the prevalence of dropout events imposes complications during data analysis and, despite numerous efforts from the community, this challenge has yet to be solved. Here we present a computational method, called RESCUE, to mitigate the dropout problem by imputing gene expression levels using information from other cells with similar patterns. Unlike existing methods, we use an ensemble-based approach to minimize the feature selection bias on imputation. By comparative analysis of simulated and real single-cell RNA-seq datasets, we show that RESCUE outperforms existing methods in terms of imputation accuracy which leads to more precise cell-type identification. Taken together, these results suggest that RESCUE is a useful tool for mitigating dropouts in single-cell RNA-seq data.

This work was supported by grants NIH R01HL119099 and NIH T32CA009337.
1.1 Introduction

Single-cell RNA-seq (scRNAseq) analysis has been widely used to systematically characterize cellular heterogeneity within a tissue sample and offered new insights into development and diseases [23]. However, the quality of scRNAseq data is typically much lower than traditional bulk RNAseq. One of the most important drawbacks is dropout events, meaning that a gene which is expressed even at a relatively high level may be undetected due to technical limitations such as the inefficiency of reverse transcription [22]. Such errors are distinct from random sampling and can often lead to significant error in cell-type identification and downstream analyses [17].

Several computational methods have been recently developed to account for dropout events in scRNAseq data, either directly imputing under-detected expression values [25, 12], adjusting all values according to some model of the observed expression [34, 20] or implicitly accounting for missingness through the extraction of some underlying substructure [47]. Here we focus on directly imputing the missing information. In this context, imputation assumes that cells of a particular classification or type share identifiable gene expression patterns. Additionally, that missingness varies across cells within each type so that it is useful to borrow information from across cells with similar expression patterns, or cell neighbors. However, a challenge is that cell neighbor identification also relies on dropout-‘infected’ data, thus creating a chicken-and-the-egg problem.

To overcome this challenge, we develop an algorithm called the REcovery of Single-Cell Under-detected Expression (RESCUE). The most important contribution of RESCUE is that the uncertainty of cell clustering is accounted for through a bootstrap procedure, thereby enhancing robustness. We apply RESCUE to simulated and biological data sets with simulated dropout and show that it accurately recovers gene expression values, improves cell-type identification and outperforms existing methods.

1.2 Results

1.2.1 Overview of the RESCUE method

To motivate RESCUE, we note that cell-type clustering is typically restricted to a subset of informative genes, such as the most highly variable genes (HVGs) across all cells [27]. If there is bias in the expression patterns
of these HVGs, then clustering will be affected. To illustrate this, we consider a hypothetical example of 500 cells containing five distinct cell types of near equal size. The introduction of dropout events distorts the pattern of gene expression and confounds clustering results by cell type (Figure 1.1a). Our solution to this problem is to use a bootstrap procedure to generate many subsets of HVGs. Based on each subset of genes, we cluster cells based on the corresponding gene expression signatures (Figure 1.1b), and created an imputation estimate by within-cluster averaging (Figure 1.1c). The final imputed data set provides an accurate representation of the cell types and their gene expression patterns (Figure 1.1d). Of note, this approach circumvents a number of limitations inherent to current imputation methods reviewed by [51], as
we’ve made no assumptions of the dropout generating mechanism or number of cell types and observed expression values are preserved.

More explicitly, given a normalized and log-transformed expression matrix, the RESCUE algorithm proceeds as follows. First, we consider the most informative features for determining cell neighbors. In this case, the most variable genes across all cells. We take a greedy approach and retain the top 1000 HVGs. The influence of any one group of genes is mitigated by repeatedly subsampling a proportion of HVGs with replacement, using the standard bootstrapping procedure [6] but with an additional clustering step for each estimator. Within each subsample, the gene expression data are standardized and reduced to their principal components to inform clustering and classify similar cells. In principle, any single-cell clustering method [10] can be applied. Here we focus on the shared nearest neighbors (SNN), which has been shown to be effective in numerous studies [48, 24]. As similar cells are assumed to share expression patterns, we calculate the average within-cluster expression for every gene in the data set as sample-specific imputations. In the end, the sample-specific imputation values are averaged for a final imputation. The mathematical details of the algorithm are described in the Methods section, where the sample-specific imputations are defined by Equation 1.1. The R package rescue is freely available at https://github.com/seasamgo/rescue and released under the GPL 3.0 license.

1.2.2 RESCUE recovers under-detected expression in simulated data

As a ground truth is not generally known with experimental data, we first considered simulations for validation of RESCUE. Count data and dropout were simulated for a benchmark data set reflective of our hypothetical motivating example using generalized linear mixed models implemented by Splatter [50]. These data consisted of 500 cells having 10,000 genes and were composed of 5 distinct groups with equal probabilities of membership. Approximately 40% of observations had a true simulated count of 0 and approximately 30% of the overall transcripts counts experiencing additional dropout. To quantify the effect of dropout and imputation, absolute count estimation error was evaluated relative to the simulated true counts. This measure is presented as the percent difference from the true counts over the data containing dropout so that 0% is best and greater than 100% indicates additional error. We used t-distributed stochastic neighbor embedding (t-SNE) [46] to visualize the data and determine the quality and separation of clusters by cell types. Additionally, we evaluated predicted cell type labels by computing their Shannon entropy, normalized mutual
Figure 1.2 Estimation bias after imputing simulated data (Table A; Primary). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout.

information (NMI), adjusted Rand Index (ARI), and Jaccard Index against their known cell type labels. The outcomes for these measures are presented as the percent improvement over the data containing dropout so that 100% is best and 0% is no improvement.

Missing counts showed marked improvement (Figure 1.2a) and RESCUE achieved a median reduction in total relative absolute error of 50% (Figure 1.2b), indicating that it can accurately recover the under-detected expression at a broad level. To ensure that missing expression values important to the classification of cell types were recovered, we considered the relative error for the top 2 most significantly differentially expressed marker genes for each cell type determined using the true counts (MAST [8] likelihood ratio test $p < 1e-5$; log-fold change $>0.5$). RESCUE consistently achieved at least a 50% median reduction of relative error.
Figure 1.3 Data visualization and cell-type clustering before and after imputing simulated data (Table A; Primary). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results.

For each of these 10 markers (Figure 1.2c). Additionally, RESCUE showed clear visual (Figure 1.3a-c) and quantitative (Figure 1.3f) improvement of cell-type classification. All 5 cell types were completely separated and clustering outcomes equivalent to the full data with a 0% difference from the true labels.

For comparison, we also imputed the dropout data with DrImpute [12] and scImpute [25], two recently developed methods designed to estimate under-detected expression values. Both methods reduced the total absolute relative error (Figure 1.2b) and DrImpute consistently reduced the relative absolute error across
all 10 marker genes (Figure 1.2c), but to a lesser degree than RESCUE. scImpute did not achieve the same reduction in error, instead having a noticeable increase in error for 6 of the 10 genes, possibly due to an overestimation of some counts (Figure 1.2a). Both methods showed notable visual (Figure 1.2d,e) and quantitative (Figure 1.3f) improvement of clustering outcomes over the data set containing dropout, greater than 30% for DrImpute and greater than 90% for scImpute, but not to the same extent as RESCUE.

These outcomes were replicated in additional simulations (Figure A.1-4) that considered variations in cell group size, the number of cell types, degrees of differential expression, and the prevalence of dropout events outlined in Table A. Collectively, the simulations suggest that RESCUE is effective at recovering under-detected expression and outperforms existing methods in terms of estimation bias and clustering outcomes with regard to cell-type classification.

1.2.3 RESCUE recovers differential expression across mouse cell types

To extend the application of RESCUE to a real data set where the underlying truth and mechanism are not fully known, we made use of the Mouse Cell Atlas (MCA) Microwell-seq data set [15]. Previous studies have identified 98 major cell types across 43 tissues [16]. We randomly selected four tissues — uterus, lung, pancreas and bladder — each of 1,500 samples to test the performance of RESCUE. For each tissue, we only retained the cells that can be classified in a major cell-type for evaluation purposes. Since it is impossible to distinguish dropout events from biologically relevant low expression in this real dataset, we artificially introduced additional dropout events by using Splatter [50]. More than 10% of additional dropouts were introduced for each tissue. Genes were removed if the quality of the resulting data was too low. As a result, the data matrix for each tissue contained approximately 98% zero counts.

Missing counts showed a global median improvement of only 3% after imputing the uterus tissue data (Figure 1.4a). However, RESCUE achieved a notable reduction of relative error across several of the most differentially expressed significant cell-type specific marker genes determined through a differential expression analysis (MAST [8] likelihood ratio test p <1e-5; log-fold change >2) of the original counts (Figure 1.4b). In particular, the Ccl11 and Mmp11 genes had a median reduction in error of 42% and 68%, respectively. This recovery of expression at a broad level and across marker genes was further replicated across the other three tissue types (Figure A.5-7). We also evaluated the recovery of log-fold changes (LFCs) in gene expression for cell-type specific genes that went undetected in the data containing simulated dropout. RES-
Figure 1.4 Estimation bias and recovery of differential expression after imputing the MCA uterus tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.

CUE recovered 53 of the 77 significant genes in the uterus tissue (Table A), with 6 of these being among the top 2 most significant differentially expressed marker genes for each cell type (Figure 1.4c). Similar results were achieved for the bladder, lung and uterus tissue data where LFC patterns were recaptured for a majority of each of the top 2 marker genes across cell types (Figure A.5-7). In contrast, other imputation methods achieved improvements in parts but not all of these elements. scImpute did not noticeably reduce count bias due to dropout events but recovered 100 marker genes across the cell types of each tissue (Table A). DrImpute had more similar results to RESCUE, reducing the overall relative error and error across marker genes, though not to the same degree. For example, the Ccl11 and Mmp11 genes had a median reduction in error of 64% and 80%, respectively (Figure 1.4b). DrImpute also recovered an additional 5 marker genes in the lung tissue data (Table A) and the second most significant differentially expressed marker, Wfde2, for urothelium cells in the bladder tissue, where RESCUE did not
However, RESCUE managed to recover several other markers in each tissue that were not detected after imputing with the other methods, including top markers Mdk (Figure 1.4c), H2–Ab1 and Myl9 (Figure A.5c), Ms4a6c (Figure A.6c) and Gsn (Figure A.7c). Together with the reduction in count bias, these results indicate that RESCUE can recover patterns of differential expression with regard to cell-type specific marker genes in the presence of heavy dropout.

### 1.2.4 RESCUE improves cell-type classification of mouse cells

To test whether RESCUE is useful for improving the accuracy of cell type identification, we overlaid the known cell-type annotation on t-SNE maps reconstructed from original, dropout, and imputed data (Figure 1.5). RESCUE greatly enhanced the visual quality of the data clusters in the uterus tissue (Figure 1.5a-c), clearly separating all 6 cell types. In particular, the endothelial cells and osteoblasts were indistinguishable from the other cells after dropout but visually distinct after imputation. A small number of cells were inseparable across cell types. However, this is seen in the original data and may be due to other sources of bias. RESCUE also improved clustering outcomes with regards to all considered measures (Figure 1.5f). We compared estimated cell clusters with the cell-type labels identified using the full 60,000 sample data set in the original MCA study [16]. The relative entropy between these labels improved by 27%, NMI by 53%, ARI by 68%, and the Jaccard Index by 49%. To test if the improvement is robust, we repeated the analysis for three additional tissues: bladder (Figure A.5), lung (Figure A.6) and pancreas tissues (Figure A.7). In all cases, we observed varying degree of improvement of RESCUE compared to existing methods.

Some of the more similar cell types were inseparable after additional dropout. For example, the dendritic cells and monocytes in the lung tissue are partly distinct in the original data but cluster together and remain indistinguishable after imputation (Figure A.9c). This could be due to a complete loss of some information distinguishing these cells, as differential expression for top dendritic cell markers was not recovered (Figure A.6c). However, we see this again with the dendritic cells and macrophages in the bladder tissue (Figure A.8c). These three immune cell types are known to greatly overlap in both functional characteristics and patterns of gene expression [13], confounding their separate classification. Thus, this event may simply be confined to similarly expressing immune cells in the presence of other dissimilar cell types. We do observe that the immune cells of both tissues become visibly distinct from other cell types with imputation, indicating a meaningful improvement in overall cell-type classification.
Other methods followed RESCUE in these outcomes. scImpute increased the similarity indexes for the uterus and bladder tissue data but did not reduce entropy or increase the NMI between the known cell labels or improve clustering outcomes across the other tissue types (Figure 1.5f). Visualization of the data with t-SNE did not improve either (1.5d, A.8-10). In contrast, DrImpute showed visible improvement across all measures predicted clustering quality for the uterus and bladder tissue data but to a lesser degree than RESCUE; this
was not seen with the pancreas and lung tissue data (Figure 1.5f) and was not fully apparent in visualization of the data with t-SNE (Figure 1.5e, A.8-10). We conclude that RESCUE improves clustering outcomes and the accuracy of cell-type classification, while outperforming other existing methods in the presence of heavy dropout.

1.3 Discussion

The identification of cell types is at the core of scRNAseq data analysis but confounded by high rates of under-detected expression that bias informative patterns of gene expression. RESCUE effectively recovered the information lost to these dropout events in both simulations and publicly available data with additional simulated dropout. Count error and feature selection bias were significantly reduced and differential expression patterns important to cell-type classification were recovered, significantly improving downstream cell-type clustering. This was achieved through two important additions to the literature. First, a solution to the circular dependency of cell-type classification and estimation of gene expression by subsampling informative genes. Second, retaining the single-cell nature of the data without strict model assumptions by applying the bootstrap across all possible clustering outcomes.

Single-cell experiments and analyses have greatly improved over the last decade and are now considered an essential component in many research areas. However, their focus has primarily been at the transcriptome level, which is only one of many regulatory layers that explains single-cell heterogeneity. Recently, additional high-throughput single-cell sequencing protocols have been developed for analyzing patterns in DNA methylation and chromatin accessibility, such as the single-cell assay for transposase-accessible chromatin (ATAC-seq) [4]. These data are unique to scRNA-seq data but present similar challenges due to high amounts of background noise and low read-coverage [26]. The RESCUE method may not be directly applicable to these other data but, given its simplicity and straightforward approach, we place interest in future extensions. Taken together with the above, we believe that RESCUE can be a useful addition to the current and developing toolsets used in the analysis of single-cell data.
1.4 Methods

1.4.1 Simulating single-cell RNA-sequencing data

Simulated data were generated using Splatter, which is a gamma-Poisson hierarchical model that samples mean gene expression from a gamma distribution and subsequent cell counts from a Poisson distribution [50]. We considered three scenarios outlined in Table A, with remaining parameters kept at their default values. If any genes were to have zero counts across all cells, we removed them from that data set before imputation.

1.4.2 Mouse Cell Atlas data and processing

We obtained the Mouse Cell Atlas (MCA) data set of 60,000 single cells from the Gene Expression Omnibus under accession code GSE108097 [15]. Our selected 4-tissue subset was filtered by cell types to those having at least 50 cells present in each data set, with this threshold being lowered to 25 cells for the bladder tissue in order to capture more cell types. In this way, we reduced bias in the final clustering analysis due simply to rare cell types. We also filtered genes with a very low detection threshold across the remaining cells (<10% nonzero counts within every remaining cell type). Both the simulated and sequenced data were processed with the Seurat pipeline implemented in R [36] using default parameters for quality control, normalization (log-transformed counts-per-million), UMI regression of the MCA data, and scaling (z-score).

1.4.3 Generating dropout events

The Splatter model generates dropout in a manner in consonance with the findings of [17]. Specifically, dropout probabilities are defined by use of the logistic function \( f(x) = \{1 + \exp[-a(x - x_0)]\}^{-1} \) fit between the log means of the normalized counts and the proportion of under-detected counts. Dropout is then generated with these probabilities and counts replaced by zero as such events occur. These methods are implemented in the R package Splatter [50]. We fixed the dropout.midpoint location parameter \( x_0 = 0 \) for all data sets. Dropout for the simulated data was generated with the parameters given in Table A. Data specific parameters for the MCA data were estimated using the splatEstimate function. The dropout.shape scale parameter was fixed at \( a = -1 \) and the model parameter dropout.type to ‘experiment’. We then generated
an index of dropout events using the splatSimulate function with cell type probability parameter group.prob set to the proportion of known cell types. Counts sampled in this way were changed to zero. This resulted in more than 10% additional dropout across each of the MCA tissues we evaluated.

1.4.4 Mathematical details of RESCUE

RESCUE takes as input a normalized and log-transformed gene expression matrix. The algorithm then proceeds as follows:

1. HVGs are determined by binning genes according to their average expression and dispersion, then calculating z-scores for dispersion within each bin with the FindVariableGenes function in the R package Seurat [36]. We took the top 1000 HVGs.

2. Simulations across multiple proportions $p$ suggested a window in which the variation of informative clustering outcomes was optimal. We fixed $p$ at a conservative 0.6 within this window to capture a simple majority of HVGs and ensure that the expression pattern of each subsample was representative of cell type but flexible across all HVGs.

3. Cell clusters are also determined via the Seurat package with the FindClusters function. This implementation of SNN borrows heavily from [24] and first draws a KNN graph over the Euclidean distance of informative principal components. We determined the number of principal components by examining elbow plots and these may be increased as desired. The graph edge weights are refined by the Jaccard distance between local neighborhoods and groups of highly connected cells are partitioned by the Louvain modularity optimization method proposed by [1]. This requires a resolution parameter as input to adjust the granularity of the community partitions: greater than 1 induces more clusters, while less than 1 induces fewer clusters. We kept this parameter at a moderate value of 0.9, the original authors suggested best results for 0.6-1.2, but it could be increased for large data sets where a greater number of unique cell types are expected.

4. Expression averages are calculated for each cluster.

5. Steps 2-4 are performed $N$ times to extrapolate the distribution of expression averages across all possible cell neighbors. We fixed $N$ at 100 to ensure consistency of the bootstrap after evaluating these distributions under simulation.

6. Take $c_i$ to be a series of these estimated similar cell cluster identities assigned to some cell $c$ with cluster size $n_{c_i}$ and for $i = 1, \ldots, N$.

Take some gene $g$ having cluster-specific expression vectors $x_{gc_i}$ for $i = 1, \ldots, N$, and denote its cluster-specific expression mean by $\theta_{gc}$. We define the estimated expression averages $\bar{x}_{gc_i} = n_{c_i}^{-1}$. 

13
\[ \sum_j x_{gc,i,j} \text{ for } j = 1, \ldots, n_{c_i}. \text{ Then, statistics computed with the estimator defined by} \]

\[
\hat{\theta}_{gc} = \left\{ \sum_{i=1}^{N} n_{c_i} \cdot \bar{x}_{gc,i} \right\} \cdot \left\{ \sum_{i=1}^{N} n_{c_i} \right\}^{-1}
\]

(1.1)

are the bootstrapped mean expression estimates of \( \theta_{gc} \) for gene \( g \) in cell \( c \). Zero counts are imputed with their respective estimates and the algorithm ends.

1.4.5 Analysis with scImpute and DrImpute

scImpute initially clusters similar cells with KMeans applied to a spectral decomposition of the data [31] to reduce the computational effort of fitting a separate generalized linear mixed model to every sample, which takes as input the expected number of cell states [25]. scImpute performed better without informing the clustering algorithm and so we fixed the initial clustering parameter \( k_s \) at 1. The authors state that this is fine as the method chooses similar cells with a model-based approach at a later step. Each data set was imputed before processing, as the method takes counts as input. DrImpute implements multiple applications of KMeans clustering and correlation distances, suggesting a range of numbers of clusters for the applications of KMeans that are at least as large as the number of expected clusters [12] (the default is 10:15). Let \( k \) be the number of known cell types. We fixed the range of clusters for DrImpute to be \( \{k, \ldots, k + 5\} \). All other parameters were fixed at their default values.

1.4.6 Evaluation of clustering outcomes and marker genes

Principal component analysis, SNN clustering and t-SNE visualization were implemented using the R package Seurat [36]. The entire filtered set of genes present in the data containing dropout was used for all evaluations. We measured count bias by retaining cell library sizes before imputation and applying an inverse function of the log-transform normalization \( g^{-1}(x) = \{\exp(x) - 1\}10^{-4}\text{library\_size} \). Log-fold changes and marker genes were determined through a differential expression analysis of the original filtered data with known cell-type labels using the FindMarkers function in the R package Seurat and MAST [8], a GLM method developed specifically for scRNAseq data that models the cell detection rate as a covariate. Genes were filtered by the magnitude of their LFC (>2.0 for the MCA data, >0.5 for the simulated data) and sorted
by significance (likelihood ratio test \( p < 1\text{e}-5 \)). Top marker genes were selected from each cell type in the original data set if they also went undetected in a subsequent analysis applied to the data set containing dropout. Similarity measures for predicted cell types were computed with the `external_validation` function in the R package ClusterR [28].

SNN does not predict a fixed number of clusters, instead producing a final number of clusters as a product of the optimal community partitions. Yet most measures of clustering quality are sensitive to variations in the number of unique clusters. Thus, it was necessary to reduce larger numbers of predicted clusters to the number of unique cell types for a quantitative evaluation of similarity to cell type labels. This was achieved by merging predicted clusters with average-linkage of the Euclidean distance across the same number of principal components used to inform the SNN clustering. The need for this is seen in the MCA bladder tissue data set, where the initial predicted clusters from the original data seem to poorly match cell types according to the plotted similarity measures (A.11c). However, the original data is quite clearly accurate according to the t-SNE plots (A.11a) when contrasted against the known cell labels (A.8a).
2

SERA: inferring spatial patterns of gene expression through integrating single-cell RNA-seq and sequential fluorescence in situ hybridization data

Samuel Tracy\textsuperscript{1,2}, Guo-Cheng Yuan\textsuperscript{1,2}

\textsuperscript{1}Department of Biostatistics, Harvard T. H. Chan School of Public Health, Boston, MA, USA
\textsuperscript{2}Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Boston, MA, USA

Abstract

The single-cell RNA sequencing technology has been widely used for the systematic dissection of biological tissues and understanding cell heterogeneity. However, a major limitation is that spatial information is lost due to cell isolation prior to sequencing. Recently, sequential fluorescent in situ hybridization has emerged as an alternative strategy for single-cell gene expression profiling while preserving spatial information, but the number of genes is often limited to a few hundreds. Here we present a computational method, called SERA, to combine the strengths of these two technologies. SERA systematically characterizes the intrinsic relationship between expression patterns of different genes based on scRNA-seq data, then applies the relationship to extrapolate seqFISH data in order to infer transcriptome-wide spatial distribution. Through analysis of published data in mouse visual cortex, we show that SERA is a useful tool for predicting the spatial pattern of cell-type and domain-specific genes.

This work was supported by the grant NIH R01HL119099.
2.1 Introduction

Single-cell RNA-sequencing (scRNA-seq) has been widely used to systematically characterize cellular heterogeneity across the transcriptome [16, 35]. Several protocols have been developed across a variety of available features, such as high precision and sequencing depth or low expense with very high throughput [30]. A primary drawback to scRNA-seq is the complete loss of spatial information as cells are isolated for sequencing [49]. Imaging based technologies, such as sequential fluorescence in situ hybridization (seqFISH) [37] provide one solution to this problem. Through signal amplification and error correction, seqFISH profiles gene expression in situ, thus preserving spatial information. However, the throughput of seqFISH is limited to a few hundred genes. Although a recent improvement of the technology has made it possible to profile around 10,000 genes in single cells [7], it will take several years for this version of the technology to mature and become accessible to the broad community. As such, we are motivated to develop a computational method to combine the strengths of seqFISH and scRNA-seq technologies to predict the spatial transcriptomics.

In previous work, [52] integrated scRNA-seq and seqFISH data and successfully mapped the spatial distribution of eight major cell types in the mouse visual cortex. While the mapping was based on the expression levels of about 40 genes, the gene signature similarity between matched cell-types extends to over 100 genes. Additionally, it was shown that platform differences in the standardized distributions of gene expression were minor for most genes and overcome through a simple normalization. Taken together, these results suggest it is possible to accurately predict spatial patterns of gene expression beyond those directly measured by seqFISH. In the present study, we take one step forward by developing a computational tool, called the Spatial Expression Regression Algorithm (SERA), to systematically predict the spatial transcriptome at the single-cell resolution. SERA combines multi-response least absolute shrinkage and selection operator (LASSO) regression with a decision rule for estimable gene expression vectors based upon a local polynomial regression of the fitted model components. We tested and validated this approach with published scRNA-seq and seqFISH data.
2.2 Results

2.2.1 Overview of the method

Figure 2.1 Motivation and visualization of the method concept. (a) A scatter plot of the seqFISH data by spatial-coordinates lacks cell-type information. (b) A principal component plot of the scRNA-seq data is labeled by cell-type but lacks spatial information. (c) A cartoon depicting the disjoint expression matrices of the two data sets illustrates how the data sets might be integrated.

To motivate the SERA algorithm, first consider the primary features of the seqFISH and scRNA-seq platforms. The seqFISH data presented by [37] contains high quality expression measurements and spatial coordinates for only 125 genes (Figure 2.1a). The scRNA-seq data presented by [39] contains expression measurements for many thousands of genes, allowing for the robust classification of 8 major cell types (Figure 2.1b), but at the loss of any spatial information. A complete integration of these data types would
allow a mapping of the transcriptome to a known spatial context. As cells are classified across a diverse spectrum of cell types and states through the identification of similar gene expression patterns, it is intuitive to categorize the seqFISH cells by their shared gene expression patterns with the scRNA-seq cells as done by [52]. We propose to take this integration one step further and estimate unknown gene expression values in seqFISH cells using cell-cell associations between single-cell platforms and gene-gene associations in the scRNA-seq data (Figure 2.1c).

For robust evaluation, we use the larger seqFISH data published by [7] for validation and testing. This data consists of 523 filtered cells and 10,000 genes, while the scRNA-seq data consists of 1,727 filtered cells (see Methods Sections 2.4.1 and 2.4.2) and 8 primary cell types determined using an iterative clustering algorithm defined by [39]. Both data sets were subset to a shared list of 9,465 genes. 86 genes of these genes were also shared with the 125-gene data set presented by [37], and these were used as predictor genes to reflect a more typical data set. 200 cells were sampled from the seqFISH data for validation, the remaining 323 cells were used for testing. The scRNA-seq data was used for training SERA in its entirety.

Normalizing data between technologies. Although technology specific, [52] showed that seqFISH and scRNA-seq data may be made comparable through quantile-normalization [3]. Thus, the first step in SERA was to quantile-normalize the entire seqFISH data set to the scRNA-seq data set to ensure identical statistical distributions. A residual platform difference was apparent in the initial results (Figure B.1a), and appeared to be due to extreme outliers in the scRNA-seq data. This was solved by first quantile-normalizing the scRNA-seq data (Figure B.1b). Quantile-quantile plots of the highest performing genes indicated that this ensured initial platform differences in standardized distributions had little-to-no effect on the final estimates (Figure B.2).

Vectors of gene expression are estimated using LASSO. Initially, we implemented a k-nearest neighbors (KNN) algorithm to estimate gene expression values. Briefly, each seqFISH cell was matched to $k$ neighboring seqFISH cells in the remaining 323 using the Pearson correlation distance across all 86 predictor genes for $k = 1, 3$ and 5. The remaining 9,379 gene expression values were then estimated for each cell with the expression means of its $k$ neighbors. This method performed very poorly and suggested the need for a more complex framework, as the few predictor genes cannot be assumed to either co-express or even be associated with most of the other thousands of genes. Thus, only associated predictor genes should inform
the estimate for each outcome gene.

The least absolute shrinkage and selection operator (LASSO) regression [45] provides one solution to this challenge, and has been extended to multiple responses by means of the elastic net [38]. Indeed, the distinct advantage of the LASSO regression over previous methods [18] is in its sparsity. Although the more general elastic net was shown to be optimal when many predictors are highly correlated or in the \( p >> N \) case, few genes are highly correlated in the 125-gene seqFISH data and our interest is when \( p \) is low. Additionally, this generalization requires an additional tuning parameter. Therefore, we decided to focus on the LASSO. The LASSO model was fit to the scRNA-seq data and the estimated coefficients used to predict expression for the outcome genes in the seqFISH data. The mathematical details of this step are described in Methods Section 2.4.4.

Estimable genes are identified through local polynomial regression. Despite an overall moderate prediction accuracy, there are a subset of genes that can be accurately predicted. We were interested to identify the determining properties that are associated with this subset of genes. Through an explorative data analysis (Figure B.3), we determined a relationship between the \( L_1 \) norm of the model coefficients, and the variance of the model estimates. This is intuitive, as the LASSO model penalizes upon the same norm. Further, a larger norm for coefficient \( j \) implies stronger associations for gene \( j \) among predictor genes. Lastly, highly variable genes present a stronger signal across cell expression patterns and are expected to be better estimable. With this in mind we developed the following rule (Figure B.1c): fit a local polynomial regression [5] of the estimate standard deviation against the \( L_1 \) norm of the model coefficients, then select the most variable (e.g. upper quartile) genes with outcomes above the mean. The mathematical details of this step are described in Methods Section 2.4.4. The R package sera is freely available at https://github.com/seasamgo/sera and released under the GPL 3.0 license.

2.2.2 SERA captures expression patterns for hundreds of genes in seqFISH data

SERA was applied to the 323-cell seqFISH sample in the same way as the 200-cell validation sample for testing. Model estimates were filtered according to the local polynomial regression rule with the additional variation threshold fixed at the upper quantile, resulting in estimates for 2,112 genes. The need for this is seen in Figure 2.2, where the filtered estimate for the Rnf165 gene is clearly inferior in quality to the
Figure 2.2 Final predictive results for the seqFISH data. (a) Heatmaps of the observed and estimated expression patterns for the poorly estimated Rnf165 gene and the accurately estimated Ptlb gene. (b) Relative sum-of-squares and correlation score distributions. (c) A diagnostic plot of the decision rule threshold.

selected estimate for the Ptlb gene. Estimates were evaluated using the proportion sum of squared error and sum of squares explained by regression, as well as their Pearson correlation $\rho$ to the observed values. The correlation score distribution for these estimates was centered at 0.18 with more than 98% having a correlation score above 0 and 1.5% having a correlation score above 0.5 (Figure 2.2b). More than 57% of estimates had $\text{SSE}/\text{SST} < 1$, suggesting a better than random guess. An evaluation of the decision rule shows that estimates with high correlation scores are all notably above the decision rule threshold, suggesting
that a stricter threshold on estimate variance would remove most of the estimates with low correlation scores (Figure 2.2c).

Score distributions were notably in favor of the scRNA-seq derived estimates over the seqFISH derived estimates — a platform difference which was not apparent in the validation stage. This is likely due to the smaller size of the training data, as only 200 cells were used for training the model in the testing stage, decreasing cell diversity and the amount of available information in the training data. Indeed, only 2 of the 8 major cell types comprise approximately 90% of the scRNA-seq data and a larger, more diverse data set may also have advantages in predicting the expression of smaller cell types and cell subtypes. It is worth noting that a majority of the predicted genes had a coefficient $L_1$ norm near or below the decision rule threshold, suggesting a lack of co-expression between many of the predicted genes and the 86 predictor genes. Taken together with the above, these results indicate that SERA can determine estimable genes with a low criterion. Additionally, estimation may be improved by either increasing the size and diversity of training data or increasing the decision rule threshold.

### 2.2.3 SERA captures the expression pattern of cell-type specific genes

To evaluate whether SERA is useful for inferring cell-type and domain-specific expression patterns, we considered the cell-type marker genes and domain genes identified in the original seqFISH study [7]. Of 138 marker genes, only 4 had been included as predictors in the 86 gene subset — Tbr1 (cluster 9; L6), Laptm5 (cluster 3; Microglial cells), Cldn5 (cluster 4; Endothelial cells) and Gja1 (cluster 5; Astrocytes). The cell-types associated with the latter 3 marker genes are the most distinct in the original scRNA-seq principal component analysis depicted in Figure 1b. 62 other marker genes were selected as estimable by SERA. Notably, the clusters associated with these genes were the most identifiable (Figure 2.3a). The distribution of sum of squares ratios and correlation scores echoed that of the entire selected gene set (Figure 2.3b), with only 2 having a correlation score below 0. Out of 121 domain-specific genes, only 2 had been included as predictors in the 86 gene subset and the remaining 119 performed poorly. Further analyses with the marker and domain-specific genes as predictors showed great improvement.

Additionally, we applied the algorithm to a leave-one-out cross-validation of a smaller seqFISH data set composed of 125 genes. To increase the number of genes available for analysis, we used a slightly modified version of the scRNA-seq data (see Methods Section 2.4.2) which resulted in 112 shared genes. This analysis
Figure 2.3 Final predictive results for 138 marker genes determined in the original seqFISH study. (a) Heatmaps of the observed and estimated expression pattern annotated by the cell types specific to each gene. (b) Relative sum-of-squares and correlation score distributions.

reflected the previous results, with several variable and strongly co-expressing marker genes performing well and other genes performing poorly (Figure 2.3a). Mfge8, Gja1, Mertk and Slco1c1 presented estimated expression patterns similar to the observed and all had correlation scores above 0.3 (Figure 2.3b). These genes co-express almost exclusively in the Astrocyte and Endothelial cells, which also happen to be the most distinct primary scRNA-seq cell types in the principle component analysis shown in Figure 1b. In fact, these are the only genes that co-express strongly out of the 112 shared genes in the seqFISH data set (Figure B.5). Collectively, these results confirm the implications of the previous analysis and imply that variability combined with co-expression of the predictor genes is essential for estimability.
2.3 Discussion

Retaining spatial information at transcriptome-level coverage is a core challenge for single-cell sequencing and imaging technologies. SERA was able to infer the overall expression patterns of hundreds of seqFISH genes with moderate accuracy using only tens of genes that were integrated with scRNA-seq data. This was the consequence of a three-step algorithm that normalized against sequencing-technology specific gene expression distribution differences, implemented a multi-response LASSO regression for the selection of associated predictor genes, and determined estimable genes using a local polynomial regression against components of the fitted model. The performance of SERA was validated and tested against published mouse brain scRNA-seq and seqFISH data. As small increases in the number and diversity of training data cells and cell types improved estimate expression patterns and evaluation scores, we propose the selection of specifically highly co-expressing and variable genes for seqFISH data collection and eventual integration with scRNA-seq data.

Single-cell technologies are developing at a rapid pace. The ability to simultaneously infer the spatial context of gene expression and completely profile expression genome-wide may eventually develop. However, the field is currently at an impasse. We developed SERA to help address this issue and infer upon the spatial expression patterns of unmeasured genes by integrating high quality scRNA-seq and seqFISH data. Taken together with the above, we believe that SERA can be a useful addition to the current and developing toolsets used in the analysis of single-cell data.

2.4 Methods

2.4.1 Imaging data pre-processing

The processed seqFISH expression data from mouse visual cortex were obtained from the authors of [52]. Briefly, the sequenced count data was log-transformed by \( \log_2(x + 1) \) for each expression value \( x \) and then standardized by gene-rows and cell-columns to remove gene and cell-specific bias. Field imaging bias was estimated and removed by using a multi-image regression algorithm, as detailed by [52].
2.4.2 scRNA-seq data pre-processing

The processed scRNA-seq Tasic data set [40] from mouse visual cortex was similarly obtained, having been converted to counts-per-million and then log-transformed and standardized in the same manner as the seqFISH data. This data set is also available from the Gene Expression Omnibus under accession code GSE71585. Each gene was then quantile-normalized to mitigate the effect of expression outliers. The seqFISH data was then quantile-normalized to the scRNA-seq data gene-by-gene to ensure they shared nearly identical distributions (Figure B.2a,b). Quantile-normalization was performed using the normalize.quantiles functions implemented in the R package preprocessCore [2]. A second version of this data was generated for the seqFISH leave-one-out cross-validation analysis, using the original counts filtered to only those cells passing quality control measures defined by [39] and genes having nonzero counts.

2.4.3 Comparative analysis

The entire scRNA-seq data set was used as training data. 200 cells were subsampled from the seqFISH data for validation. The remaining 323 cells were retained as training data for a platform comparison and as the final testing set, since they were not involved with the construction of the model or decision rule. 86 genes were shared between the scRNA-seq and both seqFISH data sets. These were used as predictors for the remaining 9,379 genes. The secondary seqFISH data set was also subset to a shared list of 112 genes with the secondary version of the scRNA-seq data set and used for leave-one-out cross-validation.

2.4.4 Mathematical details of the algorithm

*Estimating seqFISH data with scRNA-seq data.* Let $Y$ be a $N \times K$ matrix of gene expression outcome values and $X$ be a $N \times p$ matrix of predictor gene expression vectors. Let $\beta$ be a $p \times K$ matrix of coefficients. Denote the Frobenius norm for some $m \times n$ matrix $A$ by

$$\|A\|_F = \left( \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$
Then a multi-response elastic net with a Gaussian objective solves

\[
\min_{\beta \in \mathbb{R}^{p \times K}} \frac{1}{2} \|Y - X\beta\|_F^2 + \lambda \left\{ (1 - \alpha)\frac{1}{2} \|\beta\|_F^2 + \alpha \sum_{j=1}^p \|\beta_j\|_2 \right\}, \tag{2.1}
\]

over a grid of values for \(\lambda\) via block-wise coordinate descent, where the coefficient vector component estimates are determined sequentially across both rows and columns [38]. We selected \(\lambda\) to minimize the mean-squared-error of a \(k\)-fold cross-validation with \(k = 10\). This is one of three methods suggested by [45] for selecting \(\lambda\), which remains relevant in the case of multiple responses and is optimal with regards to prediction. Note that \(\alpha = 1\) reduces the penalty term on the right-hand side to the case of a modified multi-response LASSO regression, while \(\alpha = 0\) results in a ridge regression. We fixed \(\alpha = 1\) for sparsity. However, this can be relaxed toward 0 to encourage the inclusion of more predictors when many genes co-express or when \(p >> N\). This model was trained on the scRNA-seq data using the R package glmnet [11] with the default \(\lambda\) sequence. The seqFISH data was then estimated by

\[
\hat{Y} = X_{\text{seqFISH}} \times \hat{\beta}_{\text{scRNA-seq}}. \tag{2.2}
\]

**Implementing the decision rule.** Define the estimated coefficient \(L_1\) norm \(\|\hat{\beta}_k\|_1 = \sum_j |\hat{\beta}_{jk}|\) and estimate variance \(\sigma^2(\hat{Y}_k) = (N - 1)^{-1} \sum_i (\hat{Y}_{ik} - N^{-1} \sum_i \hat{Y}_{ik})^2\) for predictor genes \(j = 1, \ldots, p\) and outcome genes \(k = 1, \ldots, K\) across samples \(i = 1, \ldots, N\). We fit a local 2nd-degree polynomial regression [5] of \(\sigma(\hat{Y}_k)\) onto \(\|\hat{\beta}_k\|_1\) using local weighted least-squares with a neighborhood span of 0.75. Take \(\delta < 1\) to be some proportion of the points \((\|\hat{\beta}_k\|_1, \sigma(\hat{Y}_k))\). Weights were defined by the traditional tri-cubic weight function \(\left\{1 - \left[d / \max(d)\right]^3\right\}^3\) for the Euclidean distance \(d\) between each point and their neighbors within the proportion \(\delta\) of points about them. This approach was implemented through the loess function in the stats R package [41]. Genes were filtered if \(\sigma(\hat{Y}_k)\) was less than the local regression mean at \(\|\hat{\beta}_k\|_1\) or less than the upper quartile across all estimate standard deviations, as those with low values were not expected to perform well (Figure B.4).
2.4.5 Evaluating estimates of expression

Similarity between the estimated and observed expression patterns was evaluated with Pearson correlation $\rho$. We also considered the sum of squared error $SSE_k = \sum_i (Y_{ik} - \hat{Y}_{ik})^2$ and the sum of squares explained by the regression $SSR_k = \sum_i (\hat{Y}_{ik} - \bar{Y}_k)^2$, relative to the sum of squares total $SST_k = \sum_i (Y_{ik} - \bar{Y}_k)^2$, to give the scoring ratios $SSE_k/SST_k$ and $SSR_k/SST_k$ for $k = 1, \ldots, K$. Note that, as each gene is individually evaluated in this way and the regression is multi-response, $SST$ will not in general be equivalent to the sum of $SSE$ and $SSR$. However, these ratios will indicate the predictive value of each estimated gene expression vector over a random guess, with $SSE/SST$ approaching 0 and $SSR/SST$ approaching 1 being ideal.
3

OptBand: optimal confidence bands for functions to characterize time-to-event distributions

Tom Chen\textsuperscript{1†}, Samuel Tracy\textsuperscript{1†}, Hajime Uno\textsuperscript{2}

1 Department of Biostatistics, Harvard T. H. Chan School of Public Health, Boston, MA, USA
2 Department of Biostatistics and Computational Biology, Dana-Faber Cancer Institute, Boston, MA, USA
†These authors contributed equally to the work

Abstract

Classical simultaneous confidence bands for survival functions (i.e., Hall-Wellner and Equal Precision) are derived from transformations of the asymptotic Brownian nature of the Nelson-Aalen or Kaplan-Meier estimators. Due to the properties of Brownian motion, a theoretical derivation of the highest confidence density region cannot be obtained in closed form. Instead, we provide confidence bands derived from a related optimization problem with local time processes. These bands can be applied for the one-sample problem regarding both cumulative hazard and survival functions. In addition, we present a solution for the two-sample problem regarding cumulative hazards. The finite-sample performance of the proposed method is assessed by Monte Carlo simulation studies. The proposed bands are applied to clinical trial data to assess survival times for primary biliary cirrhosis patients treated with D-penicillamine.

This work was supported by grants NIH T32ES007142 and NIH T32CA009337.
3.1 Introduction

For time-to-event outcomes in clinical studies, inference of the cumulative distribution function or survival function itself becomes of interest. The Kaplan-Meier (KM) estimator and its corresponding asymptotic variance are often used to construct point-wise confidence intervals. When interest lies in inference for the entire survival function, a “simultaneous” confidence band (CB) needs to be constructed instead of a set of point-wise confidence intervals.

For some continuous function \( \varphi(t) \) which characterizes the event time distribution, we say two stochastic processes \( \mathcal{L}(t) \) and \( \mathcal{U}(t) \) are \( 1 - \alpha \) coverage level simultaneous CB’s for \( \varphi(t) \) on the interval \([t_L, t_U]\) if

\[
P \left( \mathcal{L}(t) \leq \varphi(t) \leq \mathcal{U}(t) \quad \forall t \in [t_L, t_U] \right) \geq 1 - \alpha
\]

Common choices for \( \alpha \) are 0.10 and 0.05. We distinguish this definition from the standard point-wise CB’s, which constructs the bands such that at each time point \( t \), \( P(\mathcal{L}(t) \leq \varphi(t) \leq \mathcal{U}(t)) = 1 - \alpha \). Clearly, point-wise CB’s will be narrower than simultaneous CB’s, but they will also not attain the nominal coverage level \( 1 - \alpha \) for the entire curve. Two classical simultaneous CB’s based on asymptotics of the KM estimator were originally developed by [14] and [29], which are now popularly referred to as the Hall-Wellner (HW) and Equal Precision (EP) bands. Both of these aforementioned papers derive a pivotal quantity for the estimated survival function and show that it weakly converges to a Brownian motion or Brownian bridge, from which one transforms to obtain simultaneous CB’s for the survival function. [19] introduced simultaneous CB’s based on the likelihood ratio approach for confidence intervals first introduced by [43]. [32] introduced a perturbation resampling method as a generalized approach in calculating critical values for transformed distributions, and notably applied this in the two-sample problem [33]. Finally, [44] derived bands which target the highest confidence density region (HCDR), but their approach requires the standard specifications and tuning procedures that accompany a Markov chain Monte Carlo and can be computationally expensive.

In this chapter, we propose a novel procedure to construct analytical CB’s \( (\mathcal{L}(t), \mathcal{U}(t)) \) for \( \varphi(t) \) which are approximately HCDR. That is, such bands aim to minimize

\[
\mathbb{E} \left( \int_{t_L}^{t_U} \{ \mathcal{U}(t) - \mathcal{L}(t) \} \, dt \right) \quad \text{s.t.} \quad P \left( \mathcal{L}(t) \leq \varphi(t) \leq \mathcal{U}(t) \quad \forall t \in [t_L, t_U] \right) = 1 - \alpha
\]  

(3.1)
To accomplish the objective in Eq (3.1), we utilize an analytic result from [21], which approximates the solution to a related optimization problem:

$$\min_u \left( \int_{t_L}^{t_U} u(t) dt \right) \quad \text{s.t.} \quad \mathbb{P}\left( -u(t) \leq W(\sigma^2(t)) \leq u(t) \quad \forall t \in [t_L, t_U] \right) = 1 - \alpha$$  \hspace{1cm} (3.2)

with solution $u^*_\kappa(t) = \psi(\kappa \sigma^2(t)) \sigma(t)$, where $W(\sigma^2(t))$ is a mean-zero Brownian motion with strictly-increasing variance function $\sigma^2(t)$, $\kappa$ is critical value related to the desired coverage level $1 - \alpha$, $\psi(x) = (-W_{-1}(-x^2))^{1/2}\mathbb{1}(x \leq e^{-1/2})$, $W_{-1}$ is the Lambert $W$ function on the $-1$ branch, and $\mathbb{1}$ is the indicator function (see Section C.1 in Appendix C for details of the derivation). The solution $u^*_\kappa(t)$ is approximate in the sense that it replaces the probability constraint in Eq (3.2) with its local time portion from a Doob-Meyer decomposition. Nevertheless, [21] have demonstrated its success in practical contexts, and henceforth we coin our simultaneous CB’s derived from their results as simply “OptBand”. To wit, our strategy is to form pivotal quantities, which are asymptotically Brownian, apply the result by [21], and transform into confidence intervals for $\varphi(t)$, which could be either the cumulative hazard function or survival function.

There are two challenges in adopting the work of [21] in practice. The first is the burden in calculating the critical value parameter $\kappa$. As shown in Section C.1 of Appendix C, $\kappa$ is dependent upon $t_L, t_U, \sigma^2(t)$, and $\alpha$, which are typically study-specific and depend on clinical research interest. Secondly, how to apply this theory to construct the OptBand for survival functions is not entirely clear. Non-linear pivotal quantity transformations do not necessarily preserve the HCDR property from the optimization in Eq (3.2) to the optimization in Eq (3.1). The pivotal quantity transformation for the cumulative hazard function happens to be exactly linear and hence it is straightforward to derive the OptBand for these cases. This is not the case for survival function, for which clever means must be utilized. In Section 3.2.1, we derive a highly-accurate functional approximation in calculating $\kappa$ so that users may automatically determine $\kappa$ for their specific problems. We construct the OptBand for the one-sample cumulative hazard function in Section 3.2.2 and include a solution for the two-sample cumulative hazard difference, as this follows immediately from the one-sample result. The OptBand for the one-sample survival function is derived in Section optbandsurv. We evaluate OptBand’s performance against that of Hall-Wellner and Equal Precision in Section 3.3.1 and illustrate with an application to clinical trial data concerning primary biliary cirrhosis in Section 3.3.2. The paper concludes with a discussion in Section 3.4.
3.2 Methods

3.2.1 Functional approximation of the critical value parameter

Akin to the critical values required for the HW or EP bands, OptBand’s critical value, $\kappa$, is required to attain a desired coverage level $1 - \alpha$. Such value is computed to satisfy

$$P\left(\left|W(\sigma^2(t))\right| \leq \psi(\kappa\sigma^2(t))\sqrt{\sigma^2(t)}\quad \forall t \in [t_L, t_U]\right) = 1 - \alpha$$

[21] only performed a Monte Carlo simulation for the case $t_L = 0$, $t_U = 1$, $\alpha = 0.05$, $\sigma^2(t) = t$ (the standard Brownian motion on the unit interval) in approximating $\kappa \approx 0.105$. We provide an approximation procedure that generalizes the latter.

Without loss of generality, we may consider minimizing the interval around $W(\sigma^2(t)/\sigma^2(t_U))$ for $t \in [t_L, t_U]$, since by properties of Brownian motion, $\sigma^{-1}(t_U)W(\sigma^2(t))$ and $W(\sigma^2(t)/\sigma^2(t_U))$ are equal in distribution. Hence, we need only to compute $\kappa$ such that the event

$$P(|W(s)| \leq \psi(\kappa s)s^{1/2} \quad \forall s \in [L, 1]) = 1 - \alpha \tag{3.3}$$

which is now only a function of $\alpha$ and $L = \sigma^2(t_L)/\sigma^2(t_U) \in [0, 1]$. Through Monte Carlo simulations calculated using Eq (3.3), we form an array of $(1 - \alpha)$ coverage values corresponding to $(\kappa, L)$. While this array is sufficient as a lookup table for the critical values, the functional relationship is very accurately approximated with

$$1 - \alpha \approx 1 + (a + bL)\kappa + a\kappa^2,$$

or

$$\kappa \approx -\frac{(a + bL) + \{(a + bL)^2 - 4a[1 - (1 - \alpha)]\}^{1/2}}{2a} \tag{3.4}$$

where $a = -0.4272$, $b = 0.2848$ (see Appendix Section C.1). As a measure of goodness of fit, the adjusted $R^2$ for Eq (3.4) based on our Monte Carlo data is 0.9987. Furthermore, the resulting $(1 - \alpha)$ from the simulation ranges from 0.871 to 0.999, encompassing accurate interpolation for the clinically-relevant coverage level of 0.95. Given the standard Brownian motion on the unit interval, we calculate $\kappa = 0.106$ with our
3.2.2 OptBand for the Cumulative Hazard Function

The one-sample problem Let $T$, having some cumulative distribution function $F$, and $C$, having some cumulative distribution function $G$, denote failure times and censoring times, respectively. Let $\{T_i, C_i\}_{i=1}^n$ be i.i.d copies from $(T, C)$. Throughout the paper, we assume $T$ and $C$ are independent. Our observed random variables are $(eT_i, \Delta_i)_{i=1}^n$, where $eT_i = \min(T_i, C_i)$ and $\Delta_i = \mathbb{1}\{T_i \leq C_i\}$. Let $\hat{H}(t)$ be the Nelson-Aalen estimator of the true cumulative hazard $H(t)$. Restricting our interval of interest to $[t_L, t_U]$, standard asymptotic results dictate that $n^{1/2}(\hat{H}(t) - H(t))/\sigma(t_U)$ weakly converges to $W(\sigma_2(t)/\sigma_2(t_U))$, where

$$\sigma_2(t) = \int_0^t \frac{dH(s)}{(1 - F(s))(1 - G(s))}$$

Our inference is based on the pivotal quantity

$$\left| n^{1/2} \frac{\hat{H}(t) - H(t)}{\sigma(t_U)} \right| \leq \psi \left( \frac{\kappa \sigma_2(t)}{\sigma_2(t_U)} \right) \frac{\sigma(t)}{\sigma(t_U)}$$

Since we do not observe the true variance $\sigma_2(t)$, we replace it with its empirical counterpart, most commonly that computed from Greenwood’s formula:

$$\hat{\sigma}_2(t) = n \sum_{t_j \leq t} \frac{d_j}{n_j(n_j - d_j)}$$

where $d_i = \sum_{j:T_j=\tilde{T}_i} T_j \Delta_j$ is the number of failures at $\tilde{T}_i$, and $n_i = |\{j : T_j \geq \tilde{T}_i\}|$ is the number of individuals at risk at time $\tilde{T}_i$. Pivoting, the asymptotic $(1 - \alpha)$ level OptBand for $H(t)$ is $\{\Omega(t), \Upsilon(t)\} = \hat{H}(t) \pm c_{CH}(t)$, where

$$c_{CH}(t) = \psi \left( \frac{\kappa \hat{\sigma}_2(t)}{\sigma_2(t_U)} \right) \frac{\hat{\sigma}(t)}{n^{1/2}}$$

and $\kappa$ is computed using Eq (3.4).
The two-sample problem  The two-sample problem considers CB’s for \( H_1(t) - H_2(t) \), where \( H_1(t) \), \( H_2(t) \) are the cumulative hazards which give rise to data \((\tilde{T}_i,\Delta_i)_{i=1}^{n_1}\) and \((\tilde{T}_{2i},\Delta_{2i})_{i=1}^{n_2}\), respectively. The quantity

\[
\left[\tilde{H}_1(t) - \tilde{H}_2(t)\right] - [H_1(t) - H_2(t)] = [\tilde{H}_1(t) - H_1(t)] - [\tilde{H}_2(t) - H_2(t)]
\]

weakly converges to \( W\left( n_1^{-1}\sigma_1^2(t) + n_2^{-1}\sigma_2^2(t) \right) \). Applying the same strategy from before, our pivotal quantity must satisfy, with probability \((1 - \alpha)\),

\[
\left| (n_1 + n_2)^{-1/2} \left[ \tilde{H}_1(t) - H_1(t) \right] - \left[ \tilde{H}_2(t) - H_2(t) \right] \right| \leq \psi \left( \frac{\sigma_p^2(t)}{\sigma_p(t)} \right) \sigma_p(t) \left( \frac{\sigma_p^2(t)}{\sigma_p(t)} \right)
\]

where \( \sigma_p^2(t) = (n_1 + n_2) \left( \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2} \right) \). Replacing \( \sigma_p^2(t) \) with estimator \( \widehat{\sigma}_p^2(t) \) and pivoting, we derive the \((1 - \alpha)\) level OptBand for \( H_1(t) - H_2(t) \) as \( \{ \mathcal{L}(t), \mathcal{U}(t) \} = [\tilde{H}_1(t) - \tilde{H}_2(t)] \pm c_{2CH}(t) \), where

\[
c_{2CH}(t) = \psi \left( \frac{\kappa \sigma_p^2(t)}{\sigma_p(t) \left( n_1 + n_2 \right)^{1/2}} \right)
\]

As before, \( \kappa \) is computed using Eq (3.4).

3.2.3 OptBand for the Survival Function

The one-sample problem  Let \( \widehat{S}(t) \) be either the Fleming-Harrington [9] or Kaplan-Meier estimator for the true survival function \( S(t) \). Standard asymptotic results show that

\[
n^{1/2} \frac{\widehat{S}(t) - S(t)}{S(t)} \text{ weakly converges to } W \left( \sigma^2(t) \right) \tag{3.5}
\]

As noted in the Introduction, we cannot apply the same strategy with this pivotal quantity as with the cumulative hazards, for it would ultimately target the minimization of the within CB area surrounding \((\widehat{S}(t) - S(t))/S(t)\), not \( \widehat{S}(t) - S(t) \). We fix this problem by weighting \( S(t) \) to Eq (3.5) and re-optimize; that is, we target \( \tilde{u}(t) = S(t)u(t) \) so that \( n^{1/2}|\widehat{S}(t) - S(t)| \leq \tilde{u}(t) \) with the same optimization strategy from [21]. As shown in Appendix Section C.2, the \( 1 - \alpha \) level OptBand for \( S(t) \) is \( \{ \mathcal{L}(t), \mathcal{U}(t) \} = \widehat{S}(t) \left( 1 \pm c_S(t) \right) \),
where

$$c_S(t) = \psi \left( \kappa \tilde{S}(t) \frac{\tilde{\sigma}^2(t)}{\sigma^2(t_U)} \right) \frac{\tilde{\sigma}(t)}{n^{1/2}}$$

and $\kappa$ is calculated according to Eq (3.6) to obtain a coverage level of $1 - \alpha$:

$$\kappa = -\frac{\tilde{b} + \{\tilde{b}^2 - 4\tilde{a}\tilde{c}\}^{1/2}}{2\tilde{a}} \quad (3.6)$$

where

$$\tilde{a} = a S_{K-1}^2$$

$$\tilde{b} = \frac{b}{\sigma^2(t_U)} \left\{ \sum_{i=1}^{K-2} S_i(\tilde{\sigma}^2(\xi_i) - \tilde{\sigma}^2(\xi_{i+1})) \right\} + \left( a + b \frac{\tilde{\sigma}^2(\xi_{K-1})}{\sigma^2(t_U)} \right) S_{K-1}$$

$$\tilde{c} = \alpha$$

and $S_i = (\tilde{S}(\xi_i) + \tilde{S}(\xi_{i+1}))/2$, where $\xi_1 \leq \xi_2 \leq \cdots \leq \xi_K$ are the failure time points we observe in the sample and $K$ is the total number of observed points.

### 3.3 Results

#### 3.3.1 Simulation

To compare our proposed OptBand against the HW and EP bands, we generate $T_i$ from a unit exponential distribution and $C_i$ from an exponential distribution with rate parameters 0, 0.25, 1 and 9 to encompass 0%, 20%, 50% and 90% censored observations, respectively. This was done at sample sizes $n = 500$ and 1000 for a total of eight sets of experiments, across 10,000 iterations. We restrict our time intervals to the four possible truncated quantile combinations: (0.01, 0.99), (0.2, 0.8), (0.01, 0.8), (0.2, 0.99). That is, we trimmed 1% and/or 20% of the starting and ending times. Coverage and area between bands are reported for the sample size of 1000 and censored proportions of 0% and 20% in Table 3.3.1. The areas are normalized.
Table 3.1 Simulation results (actual coverage, relative area) for $n = 1000$.

<table>
<thead>
<tr>
<th>Censored</th>
<th>Coverage</th>
<th>Method</th>
<th>Restriction quantiles</th>
<th>Restriction quantiles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.01,0.99</td>
<td>0.20,0.80</td>
</tr>
<tr>
<td>none</td>
<td>90%</td>
<td>HW</td>
<td>0.91, 1.00</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.88, 0.73</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.88, 0.71</td>
<td>0.90, 0.97</td>
</tr>
<tr>
<td>95%</td>
<td>HW</td>
<td>0.96, 1.00</td>
<td>0.96, 1.00</td>
<td>0.96, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.93, 0.72</td>
<td>0.96, 0.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.93, 0.69</td>
<td>0.95, 0.97</td>
</tr>
<tr>
<td>99%</td>
<td>HW</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.98, 0.69</td>
<td>0.99, 0.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.98, 0.67</td>
<td>0.99, 0.97</td>
</tr>
<tr>
<td>20%</td>
<td>90%</td>
<td>HW</td>
<td>0.91, 1.00</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.89, 0.71</td>
<td>0.91, 1.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.89, 0.69</td>
<td>0.90, 0.99</td>
</tr>
<tr>
<td>95%</td>
<td>HW</td>
<td>0.96, 1.00</td>
<td>0.96, 1.00</td>
<td>0.96, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.93, 0.70</td>
<td>0.95, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.94, 0.68</td>
<td>0.95, 0.98</td>
</tr>
<tr>
<td>99%</td>
<td>HW</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.98, 0.67</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.98, 0.66</td>
<td>0.99, 0.98</td>
</tr>
<tr>
<td>50%</td>
<td>90%</td>
<td>HW</td>
<td>0.91, 1.00</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.88, 0.72</td>
<td>0.91, 1.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.90, 0.70</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td>95%</td>
<td>HW</td>
<td>0.95, 1.00</td>
<td>0.95, 1.00</td>
<td>0.95, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.93, 0.70</td>
<td>0.95, 1.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.95, 0.68</td>
<td>0.95, 0.99</td>
</tr>
<tr>
<td>99%</td>
<td>HW</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.98, 0.68</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.99, 0.66</td>
<td>0.99, 0.98</td>
</tr>
<tr>
<td>90%</td>
<td>90%</td>
<td>HW</td>
<td>0.86, 1.00</td>
<td>0.91, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.77, 0.88</td>
<td>0.90, 0.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.89, 0.89</td>
<td>0.90, 0.90</td>
</tr>
<tr>
<td>95%</td>
<td>HW</td>
<td>0.91, 1.00</td>
<td>0.95, 1.00</td>
<td>0.95, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.83, 0.87</td>
<td>0.94, 0.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.92, 0.87</td>
<td>0.94, 0.88</td>
</tr>
<tr>
<td>99%</td>
<td>HW</td>
<td>0.96, 1.00</td>
<td>0.99, 1.00</td>
<td>0.99, 1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP</td>
<td>0.91, 0.84</td>
<td>0.98, 0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OptBand</td>
<td>0.96, 0.84</td>
<td>0.98, 0.85</td>
</tr>
</tbody>
</table>
by the HW areas under the same conditions (e.g., area of EP region / area of HW area). As exhibited in the tables, and also originally noted by [14], the HW band is overly conservative with a significantly larger regional area within the confidence band than the other two methods. All three bands closely approximate the targeted coverage levels for the \((n = 500, n = 1000) \times (0\% \text{ censor}, 20\% \text{ censor})\) combinations. Among the restriction cases, EP outperforms HW in terms of area between bands for all except the \((0.01, 0.8)\) truncation case. OptBand outperforms the other bands for all restriction cases. Notably, OptBand and EP outperform HW significantly when there is little (1%) restriction at the upper time point of the interval. However, HW becomes more competitive to EP when we consider significant (20%) restriction on the upper time point, although OptBand still outperforms both. Further simulations with heavier censoring proportions of 50% and 90% show that OptBand retains the nominal level of coverage, while EP suffers more than 0.05 below the nominal coverage levels.

### 3.3.2 Primary biliary cirrhosis data analysis

The Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver, described in Appendix D of [9] and accessible in the survival package in R [42], was a double-blind, randomized, placebo-controlled study conducted between 1974 and 1984 to evaluate efficacy of the treatment drug D-penicillamine. The primary endpoint was death, but some patients received a liver transplant. A total of 312 patients were enrolled in the randomized trial, while an additional 112 patients participating in a validation arm. The mean follow-up time for those randomized was 5.5 years with approximately 46% censoring across groups before the final time point. For illustrative purposes, we compare D-penicillamine treatment to placebo, and consider a liver transplant to be a censoring event.

We construct the HW, EP, and OptBand for time to endpoint on both the D-penicillamine and placebo groups, as displayed in Figure 3.1. Figures 3.1(a) and 3.1(b) compare OptBand with HW and EP against the KM estimate of the survival function in the placebo group, respectively, while Figures 3.1(c) and 3.1(d) make this comparison against the KM estimate of the survival function for the treatment group, respectively. Furthermore, we constructed OptBand for the difference in cumulative hazards, as seen in Figure 3.2, where 3.2(a) and 3.2(b) depict the 95% CB’s for the placebo and treatment groups, respectively, while 3.2(c) depicts the difference between cumulative hazard functions of the two groups. Regarding the within-band area for the estimated survival functions, OptBand just barely outperformed the EP band, relative to the HW band, by
approximately 0.3% and 0.2% in the placebo and treatment group, respectively. This difference is negligible. However, the clinical trial data was also heavily censored and previous simulations showed that the EP band fails to achieve the targeted coverage under heavy censoring.

As the estimated cumulative hazard functions appear linear, we fit an exponential accelerated failure time model to the data and performed band coverage simulations with a target coverage of 95%. In order to ensure that censoring reflects the original data, we calculate the censoring proportion with the final time point left out and then force right truncation at the final time point of the clinical trial. Specifically, we use the fitted estimates and the observed censoring proportions within both treatment arms to generate 10,000 replicate

Figure 3.1 OptBand shows improvement in area over the HW and EP bands with regards to the estimated survival functions when applied to the PBC trial data.
Figure 3.2 OptBand was applied to the placebo (a) and then the D-penicillamine (b) group of the PBC trial data to compute 95% CB’s for the cumulative hazard functions. In addition, we are able to compute 95% CB’s for the difference in cumulative hazard functions (c) between the two groups. These bands suggest no difference in the cumulative hazard functions between the two groups.

samples of size $n = 1000$, then fit the data with KM survival function estimates and calculate the appropriate CB’s with a 1% restriction from the initial and final time points. OptBand and HW bands still nearly achieve the targeted coverage at approximately 94% for the OptBand in both trial groups and approximately 96% and 95% for the HW bands in the placebo and D-penicillamine groups, respectively. The EP band was notably lower in coverage than the other two methods at 93% in the placebo group and 92% in the D-penicillamine group. These results are similar to those in the simulations discussed previously, regarding trends for coverage and area of the OptBand, EP and HW bands.
3.4 Discussion

In this work, we presented analytical CB’s for the one-sample cumulative hazard, two-sample cumulative hazard, and one-sample survival functions that approximately minimize the area between bands, and hence approximates the HCDR. Classic bands such as HW and EP simply require a table of critical values and hence are fast to compute. The method proposed by [44], while exactly targeting the HCDR, requires conducting a MCMC and is more time consuming. OptBand strikes a delicate balance between simplicity with an analytical solution while also roughly targeting the HCDR. Intuitively, one would expect a CB to take the form of an estimator \( \hat{\theta}(t) \) plus or minus some variation \( \alpha(t) \cdot \text{se}(\hat{\theta}(t)) \). The OptBand takes exactly that form, with \( \alpha(\cdot) \) encompassing the \( \psi(\cdot) \) function within. Hence, the \( \psi(\cdot) \) function can be viewed as the appropriate weighting function that shapes the bands to have minimal area.

Unfortunately, designing OptBand for the risk difference \( S_2(t) - S_1(t) \), at least with the current framework derived from [21], remains an intractable problem. Indeed, there also are no equivalent analytical versions of HW or EP for the difference of two survival curves. The difficulty lies in the fact that \([S_2(t) - S_1(t)] - [S_2(t) - S_1(t)] \) converges weakly to \( S_1(t)W(\sigma_1^2(t)) + S_2(t)W(\sigma_2^2(t)) \), which cannot be further manipulated into a form befitting to derive OptBand, HW, or EP.

These methods have been published for use in the R package optband on CRAN, and are freely available at https://github.com/seasamgo/optband and released under the GPL 3.0 license. We recommend the use of OptBand with some minor restrictions of the event times, (0.01, 0.99) quantiles for example, as in our simulations. Our experience with OptBand show no negative effects from censoring, even at 90%.
Supplementary material to accompany Chapter 1

Table A.1 Splatter simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data Scenario</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary</td>
<td>Scenario 2</td>
<td>Scenario 3</td>
<td></td>
</tr>
<tr>
<td>nGenes</td>
<td>10000</td>
<td>1000</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>batchCells</td>
<td>500</td>
<td>501</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>seed</td>
<td>940</td>
<td>940</td>
<td>940</td>
<td></td>
</tr>
<tr>
<td>group.prob</td>
<td>(0.2, 0.2, 0.2, 0.2, 0.2)</td>
<td>(0.1,0.3,0.6)</td>
<td>(0.2,0.4,0.4)</td>
<td></td>
</tr>
<tr>
<td>de.prob</td>
<td>0.05</td>
<td>0.1</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>dropout.mid</td>
<td>(0, 0, 0, 0)</td>
<td>(0, 0, 0)</td>
<td>(0, 0, 0)</td>
<td></td>
</tr>
<tr>
<td>dropout.shape</td>
<td>(-0.5, -0.5, -0.5)</td>
<td>(-0.1, -0.1, -0.1)</td>
<td>(-0.25, -0.25, -0.25)</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2 Significant differentially expressed genes detected after dropout.

<table>
<thead>
<tr>
<th>Method</th>
<th>Bladder</th>
<th>Lung</th>
<th>Pancreas</th>
<th>Uterus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original data</td>
<td>123</td>
<td>87</td>
<td>106</td>
<td>77</td>
</tr>
<tr>
<td>RESCUE</td>
<td>86</td>
<td>60</td>
<td>85</td>
<td>53</td>
</tr>
<tr>
<td>scImpute</td>
<td>27</td>
<td>20</td>
<td>32</td>
<td>19</td>
</tr>
<tr>
<td>DrImpute</td>
<td>77</td>
<td>65</td>
<td>61</td>
<td>50</td>
</tr>
</tbody>
</table>
**Figure A.1** Estimation bias after imputing simulated data (Supplemental Table 1; Scenario 2). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout.
Figure A.2 Data visualization before and after imputing simulated data (Supplemental Table 1; Scenario 2). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout. (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results.
Figure A.3 Estimation bias after imputing simulated data (Supplemental Table 1; Scenario 3). (a) Scatter plots compare the true transcript counts (x-axis) to estimated counts (y-axis) for those lost to dropout. The red diagonal indicates unbiased estimation. (b) The percent absolute error for all missing counts. (c) The percent error for counts specific to the top ten marker genes across cell types. The dashed lines indicate 100% error, or no improvement over dropout.
Figure A.4 Data visualization before and after imputing simulated data (Supplemental Table 1; Scenario 3). (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute. (f) The percent improvement after imputation over the data containing dropout in similarity measures between known cell types and clustering results.
Figure A.5 Estimation bias after imputing the MCA bladder tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.
Figure A.6 Estimation bias after imputing the MCA lung tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.
Figure A.7 Estimation bias after imputing the MCA pancreas tissue data. (a) The percent absolute error for all missing counts. (b) The percent error for counts specific to top marker genes across cell types. Above 100% indicates no improvement over the data containing simulated dropout. (c) Log-fold changes in the two most differentially expressed marker genes for each cell type that went undetected after dropout.
Figure A.8 Data visualization before and after imputing the MCA bladder tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.

Figure A.9 Data visualization before and after imputing the MCA lung tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.
**Figure A.10** Data visualization before and after imputing the MCA pancreas tissue data. (a) t-SNE visualization of the original data labeled by cell type. (b) t-SNE after dropout (c) t-SNE after application of RESCUE. (d) t-SNE after application of scImpute. (e) t-SNE after application of DrImpute.

**Figure A.11** Data visualization and clustering results before and after dropout in the MCA bladder tissue. (a) t-SNE visualization of the original uterus tissue data labeled by estimated clusters. (b) t-SNE after dropout. (c) Similarity measures in clustering outcomes show little difference between the two data sets due to over-clustering.
Figure B.1 SERA results for the seqFISH validation data. (a) Relative sum-of-squares and correlation score distributions for a multi-response LASSO prediction model applied to the seqFISH data without first quantile normalizing the scRNA-seq data indicates a platform effect. (b) Results after initial quantile-normalization of the scRNA-seq data mitigate the platform effect. (c) Visualizing the decision rule for gene estimability. The local regression mean is indicated by the solid line, the upper quantile of estimate standard deviations by the dashed line.
Figure B.2 Quantile-quantile plots of the z-score standardized gene expression vectors for the highest performing genes from SERA results for the seqFISH validation data, ranked by their final estimated-observed gene expression vector correlation from high to low. Differences in the initial z-score expression distributions do not appear to affect the quality of the final estimates derived from the quantile-normalized data.
Figure B.3 Exploratory analysis of the estimates and model fit when trained upon both the seqFISH validation data and scRNA-seq data indicate association of the estimate-observed expression correlation with the standard deviation of predicted expression vectors, as well as with the $L_1$ norm and $L_2$ norm of the corresponding model coefficients.
Figure S4

(a) Heatmaps of the observed and estimated expression pattern annotated by the cell types and domain types specific to each gene. (b) Relative sum-of-squares and correlation score distributions. (c) A diagnostic plot of the decision rule threshold. The local regression mean is indicated by the solid line, the upper quantile of estimate standard deviations by the dashed line.

Figure B.4 Final predictive results for a leave-one-out cross-validation analysis of the 112 genes shared between the seqFISH and scRNA-seq data. (a) Heatmaps of the observed and estimated expression pattern annotated by the cell types and domain types specific to each gene. (b) Relative sum-of-squares and correlation score distributions. (c) A diagnostic plot of the decision rule threshold. The local regression mean is indicated by the solid line, the upper quantile of estimate standard deviations by the dashed line.
Figure B.5 Gene-gene correlation patterns in (a) the full 112 gene subset of the seqFISH data, (b) the estimable gene subset determined by SERA of the seqFISH data and (c) the estimable gene subset determined by SERA of the scRNA-seq data.
C

Supplementary material to accompany Chapter 3

C.1 Minimal-area bands from Kendall et al. (2007)

Below are the results specified by [21] with a bit more generality. Suppose we intend to find a continuous function \( u \) on \( [t_L, t_U] \) that is a solution to the following optimization problem:

\[
\min_u \left( \int_{t_L}^{t_U} u(t) dt \right) \quad \text{s.t.} \quad \mathbb{P}(-u(t) \leq W(\sigma^2(t)) \leq u(t) \quad \forall t \in [t_L, t_U]) = \gamma
\]

where \( W(\sigma^2(t)) \) is a zero-mean Weiner process with strictly-increasing variance function \( \sigma^2(t) \). We call \( \gamma \) the coverage level, and is more commonly thought as \( \gamma = 1 - \alpha \), where \( \alpha \) is the significance level. The objective to be minimized is clearly one-half the area between the bands, which is exactly the objective of finding the HCDR. We may consider the dual problem

\[
\min_u 1 - \mathbb{P}(-u(t) \leq W(\sigma^2(t)) \leq u(t) \quad \forall t \in [t_L, t_U]) \quad \text{s.t.} \quad \left( \int_{t_L}^{t_U} u(t) dt \right) = \beta
\]

where \( \beta \) is a constant related to \( \gamma \). Due to the intractability of the objective function, the key observation Kendall et al. noted was to consider a related problem based on local times:

\[
\min_u \mathbb{E}[L^u[t_L, t_U] + L^{-u}[t_L, t_U]] \quad \text{s.t.} \quad \left( \int_{t_L}^{t_U} u(t) dt \right) = \beta
\]

where \( L^u[t_L, t_U] \) is the local time accumulated by \( W(\sigma^2(t)) \) along curve \( u \) on the interval \( [t_L, t_U] \). We can calculate

\[
\mathbb{E}L^u[t_L, t_U] = \mathbb{E} \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_{t_L}^{t_U} \mathbb{I}\{u(t) \leq W(\sigma^2(t)) < u(t) + \epsilon\} dt
\]

\[
= \frac{1}{\sqrt{2\pi}} \int_{t_L}^{t_U} \exp \left( -\frac{u(t)^2}{2\sigma^2(t)} \right) \frac{1}{\sigma(t)} dt
\]

55
and similarly for $\mathbb{E} L^{-u}[t_L, t_U]$. So we ultimately have

$$\min_u \frac{2}{\sqrt{2\pi}} \int_{t_L}^{t_U} \exp \left( -\frac{u(t)^2}{2\sigma^2(t)} \right) \frac{1}{\sqrt{\sigma^2(t)}} dt \quad \text{s.t.} \quad \left( \int_{t_L}^{t_U} u(t)dt \right) = \beta$$

which can be solved through the Euler-Lagrange equation to obtain

$$u^*_\kappa(t) = \psi(\kappa\sigma^2(t))\sigma(t)$$

---

**Figure C.1** Simulated coverage in terms of $\kappa$ (vertical axis) and $L$ (horizontal axis) suggests a smooth relationship (a) that is closely matched by a functional approximation (b).

For all combinations of $(\kappa, L)$ from $\kappa = 0.01, 0.02, \cdots, 0.25$ and $L = 0.005, 0.010, \cdots, 0.995$, we estimated $\gamma = 1 - \alpha$ with

$$\widehat{\gamma}_{\kappa,L} = N^{-1} \sum_{n=1}^{N} \mathbb{I} \left( |\widehat{W}_{\tau,n}(s)| \leq \psi(\kappa s)\sqrt{s} \right) \forall s = \left\{ L, L + \frac{1}{\tau}, L + \frac{2}{\tau}, \cdots, 1 - \frac{2}{\tau}, 1 - \frac{1}{\tau}, 1 \right\}$$

where $\widehat{W}_{\tau,n}(t) = \frac{1}{\sqrt{\tau}} \sum_{k=1}^{\tau} t Z_{k,n}$, with $Z_{k,n}$ being iid standard normal, and $t = \frac{1}{\tau}, \cdots, 1$. This provides an approximation to the standard Brownian motion on the unit interval, with the step-size parameter $\tau$ controlling how granular the approximated process is and $N$ being the number of Monte Carlo simulations. $N = 4 \times 10^5$ and $\tau = 10^5$ in our simulations. Note that this choice of $\tau$ partitions the interval $[L, 1]$ into intervals of equal size, given the sequence for $L$. We further justify our functional approximation with a
heatmap in Figure C.1. The borders between the colors on the heatmap are smooth and closely match up with the $\gamma$ contours defined from our functional approximation

$$\tilde{\gamma}_{\kappa,L} = 1 + (a + bL)\kappa + a\kappa^2$$

proposed in the main text.

## C.2 Derivation of OptBand for the Survival Function

In order to exactly target the bands around $\tilde{S}(t) - S(t)$, we may consider the optimization problem

$$\min_u \frac{2}{\sqrt{2\pi}} \int_{t_L}^{t_U} \exp \left( -\frac{u(t)^2}{2\sigma^2(t)} \right) \frac{1}{\sigma(t)} \, dt \quad \text{s.t.} \quad \left( \int_{t_L}^{t_U} S(t)u(t) \, dt \right) = \beta$$

which, through similar steps taken in Section C.1 of the Appendix, gives us the solution

$$u^*(t) = \psi(\kappa S(t)\sigma^2(t))\sigma(t)$$

Again, making the transformation $\sigma^2(t) \mapsto \sigma^2(t)/\sigma^2(t_U)$, find $\kappa$ so that the event

$$|W(s)| \leq \psi(\kappa S(t)s) \sqrt{s} \quad \forall s \in \left[ \frac{\sigma^2(t_L)}{\sigma^2(t_U)} \right]$$

occurs with probability $\gamma$, where $t = [\sigma^{-2}]^{-1} \{ \sigma^2(t_U)s \}$ and $[\sigma^{-2}]^{-1}$ denotes the inverse variance function.

We now face a problem where our $\kappa$ is also a function of $S(t)$. We propose the following computation process. Denote

$$\mathcal{E}(\eta(\cdot), \tau, v) = \left\{ \left| W \left( \frac{\sigma^2(t)}{\sigma^2(t_U)} \right) \right| \leq \psi \left( \eta(t) \frac{\sigma^2(t)}{\sigma^2(t_U)} \right) \frac{\sigma(t)}{\sigma(t_U)} \quad \forall t \in [\tau, v] \right\}$$
Now if \( \eta(\cdot) = \eta \) is a constant, we can calculate using the functional approximation:

\[
\mathbb{P}(\mathcal{E}(\eta, \tau, \upsilon)) = \mathbb{P}\left( W \left( \frac{\sigma^2(t)}{\sigma^2(\upsilon)} \right) \leq \psi \left( \eta \frac{\sigma^2(\upsilon)}{\sigma^2(t \cup U)} \right) \sigma(t) \frac{\sigma(\upsilon)}{\sigma(t)} \quad \forall t \in [\tau, \upsilon] \right) 
\approx 1 + \left( a + \frac{b^2}{\sigma^2(\upsilon)} \right) \left( \sigma^2(\upsilon) \right) + a \left( \eta \frac{\sigma^2(\upsilon)}{\sigma^2(t \cup U)} \right)^2
\]

In what follows, we will need the fact

\[
\eta_1 \leq \eta_2 \implies \mathcal{E}(\eta_1, \tau, \upsilon) \supseteq \mathcal{E}(\eta_2, \tau, \upsilon) \tag{C.1}
\]

which follows, since a larger \( \eta \) corresponds to tighter bands.

Let \( t_L = \xi_1 \leq \xi_2 \leq \cdots \leq \xi_K = t_U \) be some partition of the interval \([t_L, t_U]\), with the most natural choice being the observed event time points \( \{t_i : t_L \leq t_i \leq t_U\} \). Let \( \overline{S}_i = (\overline{S}(\xi_i) + \overline{S}(\xi_{i+1}))/2 \). The idea is to approximate \( S(t) \) for \( t \in [\xi_i, \xi_{i+1}] \) with a midpoint Riemann sum \( \overline{S}_i \). Then apply functional approximation for each locally constant portion of \( \overline{S}_i \). More specifically,

\[
\gamma = \mathbb{P}\left( \bigcap_{i=1}^{K-1} \mathcal{E}(\kappa \overline{S}_i, \xi_i, \xi_{i+1}) \right)
\approx \mathbb{P}\left( \bigcap_{i=1}^{K-1} \mathcal{E}(\kappa \overline{S}_i, \xi_i, \xi_{i+1}) \right) \tag{C.2}
\]

The interpretation of \( \bigcap_{i=1}^{K-1} \mathcal{E}(\kappa \overline{S}_i, \xi_i, \xi_{i+1}) \) is the event that our Brownian motion falls within our bands on interval \([\xi_i, \xi_{i+1}]\) for each \( i \). Equivalently, since \( \kappa \overline{S}_i \) is non-increasing, hence \( \mathcal{E}(\kappa \overline{S}_i, \xi_{i+1}, \xi_K) \) are tighter than \( \mathcal{E}(\kappa \overline{S}_{i+1}, \xi_{i+1}, \xi_K) \) by Eq C.1, we can imagine first having to fall within \( \mathcal{E}(\kappa \overline{S}_1, \xi_1, \xi_K) \), then next having to fall within \( \mathcal{E}(\kappa \overline{S}_2, \xi_2, \xi_K) \), and so on, leading to the equivalence

\[
\bigcap_{i=1}^{K-1} \mathcal{E}(\kappa \overline{S}_i, \xi_i, \xi_{i+1}) = \bigcup_{i=1}^{K-1} \mathcal{E}(\kappa \overline{S}_i, \xi_i, \xi_K)
\]
Continuing Eq C.2,

\[ \gamma = \mathbb{P}\left( \bigcup_{i=1}^{K-1} \mathcal{E}(\kappa \mathcal{S}_i, \xi_i, \xi_K) \right) \]

\[ = \mathbb{P}\left( \bigcup_{i=1}^{K-1} \left[ \mathcal{E}(\kappa \mathcal{S}_{i-1}, \xi_i, \xi_K) \ \cup \ \mathcal{E}(\kappa \mathcal{S}_i, \xi_i, \xi_K) \right] \right) \]

Define \( \mathcal{E}(\kappa \mathcal{S}_0, \xi_0, \xi_K) = \emptyset \)

\[ = \sum_{i=1}^{K-1} \left\{ \mathbb{P}(\mathcal{E}(\kappa \mathcal{S}_i, \xi_i, \xi_K)) - \mathbb{P}(\mathcal{E}(\kappa \mathcal{S}_{i-1}, \xi_i, \xi_K)) \right\} \]

\[ \approx \frac{\kappa b}{\sigma^2(t_U)} \left( \sum_{i=1}^{K-2} \mathcal{S}_i (\sigma^2(\xi_i) - \sigma^2(\xi_{i+1})) \right) + 1 + \left( a + b \frac{\sigma^2(\xi_{K-1})}{\sigma^2(t_U)} \right) (\kappa \mathcal{S}_{K-1}) + a (\kappa \mathcal{S}_{K-1})^2 \]

which is a quadratic function in \( \kappa \), and hence we can solve for \( \kappa \) analytically. We also estimate \( \sigma^2(t) \) with \( \hat{\sigma}^2(t) \), as before. Finally, we pivot to get a 100\( \gamma \)% CI:

\[ \gamma \approx \mathbb{P}\left( \left| n^{1/2} \frac{\hat{S}(t) - S(t)}{\hat{\sigma}(t) / \sigma(t)} \right| \leq \psi \left( \kappa \hat{S}(t) \frac{\hat{\sigma}^2(t)}{\sigma^2(t)} \right) \frac{\hat{\sigma}(t)}{\sigma(t)} \ \forall t \in [t_L, t_U] \right) \]

\[ = \mathbb{P}\left( \hat{S}(t) \left( 1 - c_s(t) \right) \leq S(t) \leq \hat{S}(t) \left( 1 + c_s(t) \right) \ \forall t \in [t_L, t_U] \right) \]

where \( c_s(t) = \psi \left( \kappa \hat{S}(t) \frac{\hat{\sigma}^2(t)}{\sigma^2(t)} \right) \frac{\hat{\sigma}(t)}{n^{1/2}} \).
References


