Statistical Methods for Mobile Health and Genomics Data

HARVARD UNIVERSITY
Graduate School of Arts and Sciences

DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the

Department of Biostatistics

have examined a dissertation entitled

"Statistical Methods for Mobile Health and Genomics Data"

presented by Matthew Quinn

candidate for the degree of Doctor of Philosophy and hereby certify that it is worthy of acceptance.

Signature Rafael Irizarry
Rafael Irizarry (May 10, 2022 16:09 EDT)

Typed name: Prof. Rafael Irizarry

Signature Kimly Glass

Typed name: Prof. Kimberly Glass

Signature Junwei Lu (May 10, 2022 16:47 EDT)

Typed name: Prof. Junwei Lu

Signature ....

Typed name:

Date: May 9, 2022

Statistical Methods for Mobile Health and Genomics Data

a dissertation presented
by
Matthew Quinn
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 2022

Dissertation Advisors: Professors Kimberly Glass & Rafael Irizarry Matthew Quinn

Statistical Methods for Mobile Health and Genomics Data

Abstract

A common goal in statistical analyses is to differentiate signal from noise. This problem is ubiquitous to many fields, including mobile health (mHealth) and genomics, both of which have garnered tremendous interest in recent years as advancements in technology continue to make them even more prominent for studying human health. While this challenge of detecting signal is universal, the solutions to it are not. Different research applications introduce their own idiosyncrasies that can make existing approaches for signal detection insufficient for that specific context. In this dissertation, we present approaches for signal detection for three different problems in mHealth and genomics.

In Chapter 1, we study mHealth data, which are often collected through wearable devices, such as watches and other fitness trackers. The devices record and process data using algorithms that are subject to updates and glitches, which device manufacturers often do not publicize. As a result, devices can suddenly change how data are collected and reported over time. A researcher using mHealth data needs to be able to detect these changes in order to adjust for them. We propose Automated Selection of Changepoints using Empirical P-values and Trimming (ASCEPT) as an approach for objectively identifying where these changes occur. ASCEPT relies upon Monte Carlo simulations and regression models to accurately identify these algorithmic changes. We compare ASCEPT to an existing method on both simulated and real mHealth data.

In Chapter 2, we look at chromatin immunoprecipitation sequencing (ChIP-seq) data, which reflect where proteins bind to a genome. Researchers often compare individuals from different experimental groups or biological conditions to detect regions of the genome in which there is differential binding (DB). DB in particular regions may then be associated with different health outcomes between the two groups, in turn helping the researcher understand risk factors or mechanisms contributing to a particular disease. However, popular methods for detecting DB often do not fully account for autocorrelation within samples, biological variability across samples, or selection procedures used to find regions of interest. As a result, they often report inappropriate inference regarding the significance of DB regions. We present a permutation test pipeline for finding DB sites on a genome while accounting for autocorrelation, biological variability, and the selection procedure in order to provide accurate inference. We compare this pipeline to two popular methods on both real and simulated data.

In Chapter 3, we continue studying genomics data, but this time focus on ribonucleic acid sequencing (RNA-seq) data, which reflect gene expression. Researchers commonly use RNA-seq data to study gene co-expression, or how the expression of different genes are correlated with one another. One can use the co-expression between genes to construct networks to better understand gene regulation or biological mechanisms, often with the hope of learning more about the drivers of certain health outcomes. However, not all co-expressions in RNA-seq are genuine. Technical issues with sequencing and normalization procedures that researchers perform may introduce spurious signals. We present evidence that this problem arises for different genes in real RNA-seq data and that the characteristics of these false signals can vary depending both on the normalization procedure used and the tissue in which the expression occurs. We present different metrics for characterizing the presence of these spurious correlations and permutation tests for assessing their statistical significance.

While we present three different research problems, they are all manifestations of the same core challenge. Whether we detect algorithmic changes in mHealth data over time, regions on the genome that contain DB, or spurious correlations among genes, the same underlying challenge of differentiating true signal from noise comes up. Additionally, while the solution to each instance is unique, we find that computational techniques, like Monte Carlo simulations and permutation tests, are particularly helpful tools in each scenario. Thus, while both the specific type of signal detection and its solution will depend on the underlying research context, there are commonalities among signal detection problems that can be helpful for understanding and addressing them.

Contents

Listing of figures

List of Tables

Acknowledgments

The past five years have been an incredible experience and there are many people to thank for making the journey possible. I would like to thank my advisors, Rafa and Kimbie, both for their guidance on research and for their mentorship. Their encouragement and expertise have helped me immensely in developing the skills necessary to be a professional statistician.

I would like to thank Junwei for his insight and guidance on research as a member of my committee. I would like to thank Jelena for her helpfulness on every question I have ever had about the department and the program. I would like to thank the Department of Biostatistics for this great opportunity.

I thank my parents, Mary Anne and John, and my partner, Wes, for their endless encouragement and love. Reaching this capstone would not have been possible without them and I am forever grateful for all the ways in which they have supported me.

Finally, I would like to acknowledge our cats who have joined me through many hours of research, classes, and meetings. Our first cat, Pepper, is an adorable gray tuxedo who has sat on my lap while I have worked over the years. She is a great source of joy to us and is just about as cute as possible. She is truly the best pet someone could ask for. Our second cat, Tony, is her brother.

1

Automated Selection of Changepoints using Empirical P-values and Trimming (ASCEPT)

Abstract

One challenge that arises when analyzing mobile health (mHealth) data is that updates to the proprietary algorithms that process these data can change apparent patterns. Since the timings of these updates are not publicized, an analytic approach is necessary to determine whether changes in mHealth data are due to lifestyle behaviors or algorithmic updates. Existing methods for identifying changepoints do not consider multiple types of changepoints, may require pre-specifying the number of changepoints, and often involve non-intuitive parameters. We propose a novel approach, Automated Selection of Changepoints using Empirical P-values and Trimming (ASCEPT), to select an optimal set of changepoints in mHealth data. ASCEPT involves two stages: (1) identification of a statistically significant set of changepoints from sequential iterations of a changepoint detection algorithm; and (2) trimming changepoints within linear and seasonal trends. ASCEPT is available at https://github.com/matthewquinnl/changepointSelect. We demonstrate ASCEPT's utility using real-world mHealth data collected through the Precision VISSTA study. We also demonstrate that ASCEPT outperforms a comparable method, Circular Binary Segmentation (CBS), and illustrate the impact when adjusting for changepoints in downstream analysis. ASCEPT's only required parameters are a significance level and goodness-of-fit threshold, offering a more intuitive option compared to other methods. ASCEPT provides an approachable and useful way to identify which changepoints in mHealth data are likely the result of updates to the underlying algorithms that process the data.

1.1 Introduction

Recently, mobile health (mHealth) has taken on a growing importance in medicine and public health, among other fields [48, 74, 41]. mHealth devices, such as Fitbit smartwatches, often produce time series data by recording variables, like heart rate and number of steps, at regular intervals (e.g., hourly or daily). Studying these data can bring important insights into how health changes over time. For instance, an individual might greatly reduce their daily number of steps after an injury. This type of event is associated with a "changepoint," a time at which the distribution of data changes, and typically corresponds to a change in the mean of the data or a "mean-shift." However, in addition to changepoints due to lifestyle or behavioral changes, wearable devices also have both planned software or hardware updates and unexpected technical issues that can impact data collection and reporting. These can introduce "technological changepoints" to the data, which can be difficult to distinguish from behaviorally driven changes, obscuring patterns of interest. Therefore, it is necessary to identify and correct for these technological changepoints before proceeding with downstream analysis.

Unfortunately, mHealth device manufacturers often do not publicize the timing of planned updates and identified technological issues. Although an investigator could potentially monitor a manufacturer's release notes to determine when updates are pushed or manually inspect the mHealth data to find potential technological changepoints, these approaches are neither scalable nor practical, and are especially challenging when studies utilize multiple types of devices. Even a single manufacturer may not push updates to all devices simultaneously, or they may require users to first update an associated smartphone app. Thus, manufacturer updates sometimes do not even coincide with a single timepoint across users.

There are several existing approaches that detect changepoints in time series by solving an optimization problem [80], including Pruned Exact Linear Time (PELT) [45]. Using PELT generally entails specifying an optimization penalty when detecting multiple changepoints, which is difficult to do in practice. Changepoints for a Range of PenaltieS (CROPS) [34] allows one to efficiently run PELT under various penalties, but does not select a final or optimal set of changepoints. Thus, instead of proposing another method for changepoint "detection," we developed Automated Selection of Changepoints using Empirical P-values and Trimming (ASCEPT) to identify changepoints in mHealth data through changepoint "selection." ASCEPT performs multiple runs of PELT, considering iteratively larger sets of changepoints until the selected set would no longer offer a statistically significant improvement over the prior set. Next, ASCEPT removes changepoints that are likely to be associated with lifestyle or behavioral changes rather than technological issues, ultimately yielding a single optimal set of changepoints. It is worth noting that ASCEPT also shares similarities with Circular Binary Segmentation (CBS) [61], which performs "pruning" to identify a subset of statistically significant changepoints. However, CBS does not consider features common to mHealth data, such as seasonal patterns. For a more detailed review of changepoint detection, please refer to Appendix A.1.

In this study, we evaluate ASCEPT on both simulated data and Fitbit data collected by the Precision VISSTA study [18] to determine whether the method appropriately identifies changepoints. We compare the performance of ASCEPT to that of CBS [61] to examine whether ASCEPT provides better changepoint selection under comparable settings. Lastly, we perform a correction procedure to determine whether differences between the procedures have an impact when adjusting the mHealth data for the identified changepoints.

1.2 Materials and Methods

1.2.1 Data

1.2.1.1 Precision VISSTA Data

We evaluated the performance of ASCEPT on mHealth data from the Precision VISSTA study [18]. This data set included adults in the United States who had inflammatory bowel diseases and were part of the parent Internet cohort study. Users participated in a survey and could donate their personal wearable device data towards research, meaning that the study followed a bring-your-own-device model. Thus, the data set contained a number of different manufacturers and device types. Due to their prevalence in the cohort, we chose to focus on individuals who used a heart rate (HR) Fitbit device introduced in 2016-2019 (i.e., the Alta HR, Blaze, Charge 2, Charge 3, Inspire HR, Ionic, Versa, or Versa 2), multiple Fitbit devices over time, or an unknown Fitbit device (e.g., a Fitbit app). This subset of the data included 203,351 observations on 298 individuals recorded between May 15, 2015 and October 27, 2019.

These data included seven activity variables (steps, distance, floors, elevation, calories, and time active) and five sleep variables (total sleep, deep sleep, light sleep, REM sleep, time awake at night, and times woken). The median number of users contributing data on a given day ranged from 50 for REM, to 93-95 for the other sleep variables, to 131 for the activity variables. We excluded floors and elevation as their values largely stayed within narrow ranges near zero over the study period. We also excluded REM sleep due to a lack of any data between May 20, 2016 and March 26, 2017.

To help identify population-level changepoints, we focused on studying the daily median value of each variable across users. Figure 1.1a shows the daily median amount of deep sleep, which experienced an abrupt shift around July 2017. While it is possible that a single individual could have suddenly experienced large changes in deep sleep due to various life events, like an injury or the birth of a child, it is unlikely that the median deep sleep across many users truly decreased by 5-6 hours after July 19, 2017, only for it to later rebound multiple times. Instead, these shifts were more likely attributable to changes in how Fitbit's algorithms calculated deep sleep. Thus, it is critical to identify and control for these technological changepoints in order to correctly describe human behavioral changes relevant to health and disease.

1.2.1.2 Simulated Data

Shifts and patterns that appeared in the real data were often not defined well enough to serve as gold standards. For example, there appeared to be seasonality in the deep sleep data prior to July 19, 2017, but it was inconsistent (Figure 1.1a). Likewise, it was challenging to determine whether some points between May 15, 2015 and May 15, 2016 constituted behaviorally driven or technological changepoints because only 7 to 54 unique users contributed data during this time. Therefore, large behaviorally driven fluctuations were more likely during this period compared to later, when up to 160 unique users contributed sleep observations (Supplementary Figure A.1).

Due to these limitations, we first evaluated ASCEPT using a simulated time series containing 800 observations (Figure 1.1b). This data set had sudden mean-shift changepoints at indices 49, 60, 600, 699, and 700, an increasing linear trend between indices 201 and 400 inclusive, and a seasonal pattern between indices 401 and 600 inclusive.

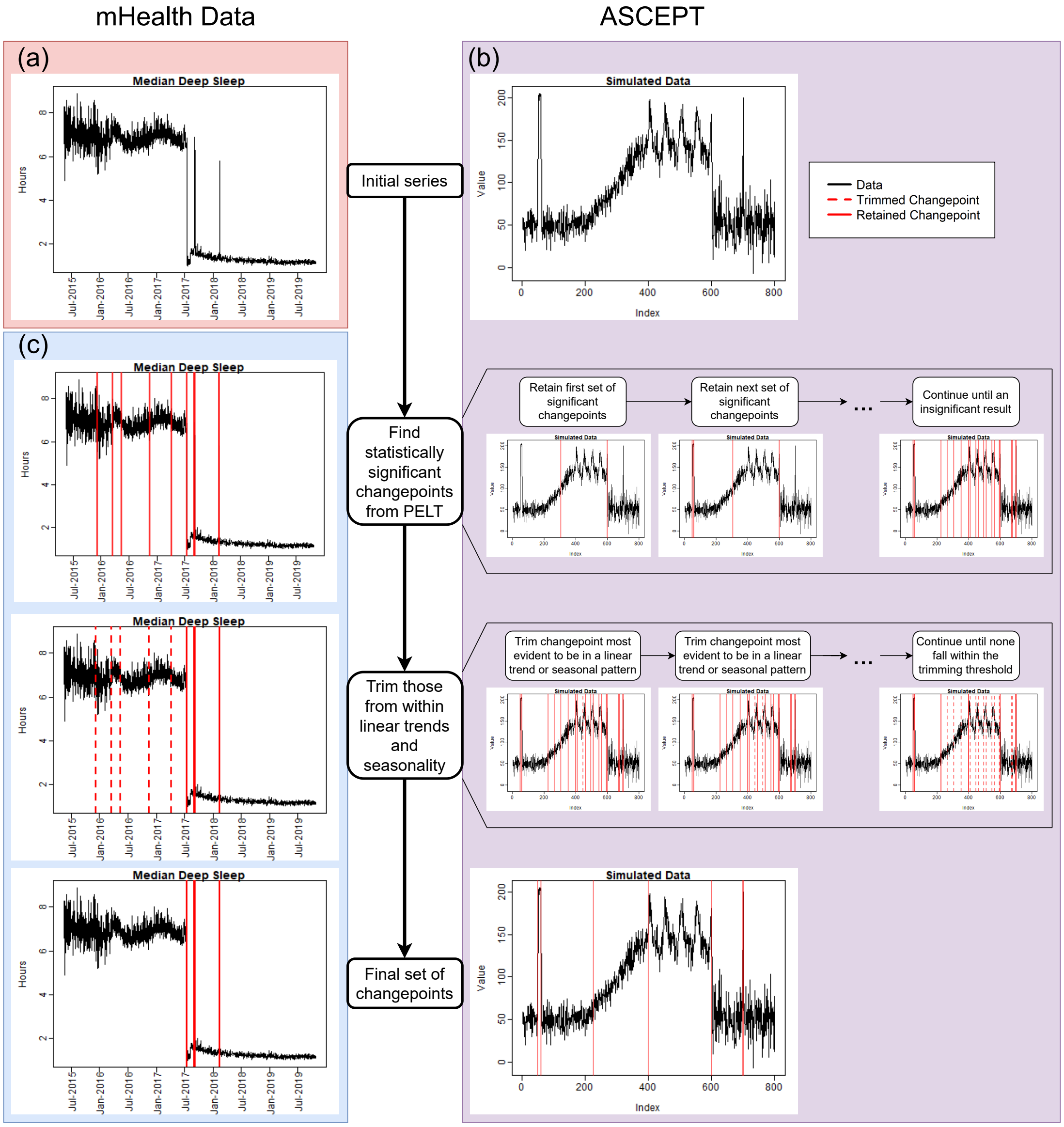

mHealth Data
ASCEPT
(a)
Median Deep Sleep
Hours
Jul-2015
Jan-2016
Jul-2016
Jan-2017
Jul-2017
Jan-2018
Jul-2018
Jan-2019
Jul-2019
(b)
Simulated Data
Value
Index
Data
Trimmed Changepoint
Retained Changepoint
(c)
Initial series
Find statistically significant changepoints from PELT
Trim those from within linear trends and seasonality
Final set of changepoints
Retain first set of significant changepoints
Retain next set of significant changepoints
...
Continue until an insignificant result
Trim changepoint most evident to be in a linear trend or seasonal pattern
Trim changepoint most evident to be in a linear trend or seasonal pattern
Continue until none fall within the trimming threshold

Figure 1.1: The ASCEPT workflow. (a) The daily median deep sleep from the Precision VISSTA study. (b) ASCEPT broken down by stage and applied to simulated data. The first row shows the original simulated time series. The second row shows significant changepoints being iteratively identified. The third row shows changepoints within linear and seasonal trends being iteratively trimmed. The fourth row shows the simulated time series with the final set of identified changepoints. (c) The same results as B but for the deep sleep data.

1.2.2 Stage 1: Changepoint Selection using Empirical P-values

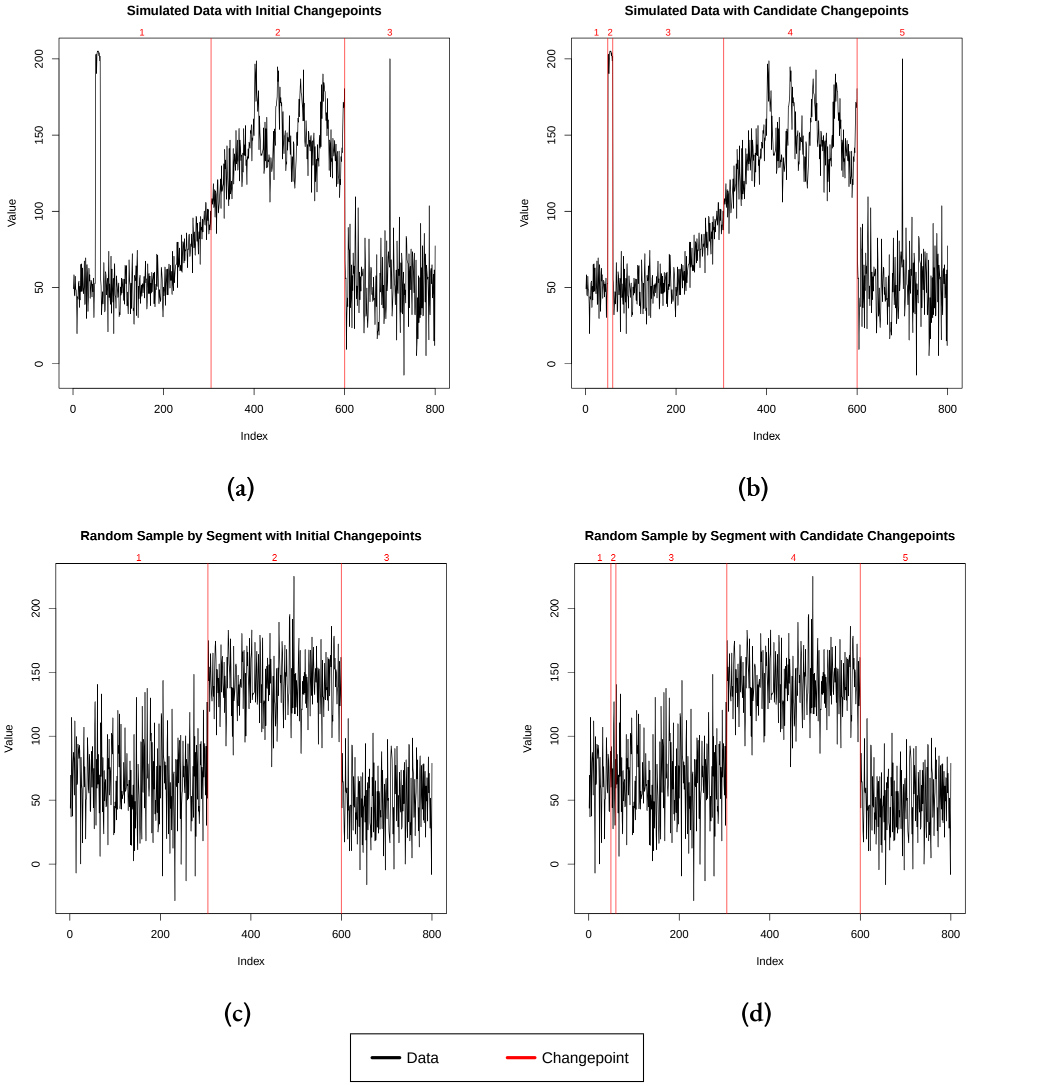
The first stage of ASCEPT incrementally includes more mean-shift changepoints detected by PELT [45] until the newly proposed changepoints do not offer a statistically significant improvement in goodness-of-fit.

First, we let a changepoint at position j indicate that the time series' distribution changes between j and $j+1$. $\mathcal{T}_k$ denotes the set of changepoints detected by step k, where $\mathcal{T}_0 = \emptyset$, such that the procedure starts with no identified changepoints. This corresponds with imposing a large optimization penalty with PELT. From step k, CROPS [34] decreases the optimization penalty associated with PELT to find the next set of changepoints, denoted as $\mathcal{T}^*_{k+1}$. $\mathcal{T}_k$ will normally, but not necessarily, be a subset of $\mathcal{T}^*_{k+1}$. Figures 1.2a and 1.2b depict a scenario in which we have detected changepoints $\mathcal{T}_k = \{305, 600\}$ and are evaluating $\mathcal{T}^*_{k+1} = \{49, 60, 305, 600\}$ as providing a significant improvement.

To assess whether or not $\mathcal{T}^*_{k+1}$ offers a significant improvement in goodness-of-fit, we must both choose a goodness-of-fit measure and assess its null distribution. For goodness-of-fit, we use the log-likelihood of normally distributed data. More specifically, between any two changepoints, or between a changepoint and the start or end of the series, the observations form a "segment." We assume that all observations are independent and normally distributed, but that those within the same segment are also identically distributed. This assumption largely follows the implementation of PELT in R's "changepoint" package [44].

We next assess the null hypothesis that $\mathcal{T}_k$ represents all of the true mean-shift changepoints in the time series. We do this is a manner that does not rely on asymptotic results, since mHealth time series can contain very small segments. We first generate a time series under the null by randomly drawing from normal distributions with the same means and standard deviations as the corresponding segments in the observed data. For example, Figure 1.2a shows the simulated data split into three segments by two changepoints at indices 305 and 600. Figure 1.2c illustrates a corresponding random sampling from the normal distributions that best fit each of these three segments. We record the log-likelihood for this random time series using the $\mathcal{T}_k$ changepoints. We then impose the changepoints in $\mathcal{T}^*_{k+1}$ onto this random time series and calculate the corresponding log-likelihood, as depicted in Figure 1.2d. Finally, we record the change in the log-likelihood under the null, comparing $\mathcal{T}^*_{k+1}$ with $\mathcal{T}_k$.

We repeat this process N times in order to calculate an empirical p-value for the observed change in the log-likelihood. If the observed change is statistically significant at the chosen level, α, then we reject the null that $\mathcal{T}_k$ represents all the true mean-shift changepoints for the time series, and instead select $\mathcal{T}^*_{k+1}$ as the current set of changepoints, $\mathcal{T}_{k+1}$. Figure 1.1b shows how the procedure continues, comparing $\mathcal{T}_{k+1}$ to $\mathcal{T}^*_{k+2}$ and so forth, until we obtain a statistically insignificant result. This hypothesis testing process is a "fixed-sequence" procedure and controls the family-wise error rate at the chosen significance level, α [55].



(a)

(b)

(c)

(d)

Figure 1.2: The process for assessing the significance of new changepoints in ASCEPT. (a) The simulated data with initial changepoints $\mathcal{T}_k = \{305, 600\}$. The log-likelihood, assuming independent and identically distributed observations within-segment, is -3727.3. (b) The simulated data set with the next set of changepoints $\mathcal{T}^*_{k+1} = \{49, 60, 305, 600\}$. The log-likelihood is -3512.3, thus the observed change in the log-likelihood is 215.0. (c) A Monte Carlo sample with the initial changepoints at $\{305, 600\}$ shown. The observations in each segment are randomly drawn from a normal with a mean and standard deviation equal to that for the corresponding segment in subfigure a. The log-likelihood is -3746.6. (d) The same Monte Carlo sample from subfigure c, but now with the next set of changepoints at $\{49, 60, 305, 600\}$ shown. The log-likelihood is -3745.0. The change in the log-likelihood for this Monte Carlo sample under the null is therefore 1.6. The process in subfigures c and d is repeated a large number of times to generate an empirical null distribution for the change in the log-likelihood. In all plots, the segments between the identified changepoints are numbered.

1.2.3 Stage 2: Trimming Changepoints within Linear or Seasonal Trends

Stage 1 identifies changepoints that include both technological changepoints, such as those associated with manufacturer updates, and changepoints from behaviorally driven patterns. In order to distinguish the former from the latter we note that, while software updates are likely to induce sudden mean-shifts in population-level mHealth data, behaviorally driven changes are more likely to be associated with linear or seasonal trends (see, for example, Figure 1.1a). For instance, individuals may walk more during the summer than the winter, or at the end of an exercise program compared to at the start. These trends technically contain a mean-shift at each point, but since these are all part of the same behaviorally driven pattern, ASCEPT aims to identify and remove them; for convenience we refer to these as "nuisance changepoints." In contrast, ASCEPT retains "relevant changepoints" that correspond with a sudden mean-shift, or that are at the start or end of a linear or seasonal trend (Figure 1.1c). We refer to ASCEPT's process of removing nuisance changepoints as "trimming." Although it is the same principle as "pruning" used by methods such as CBS [61], we avoid the term "prune" because PELT also uses "prune" to describe part of its optimization process [45].

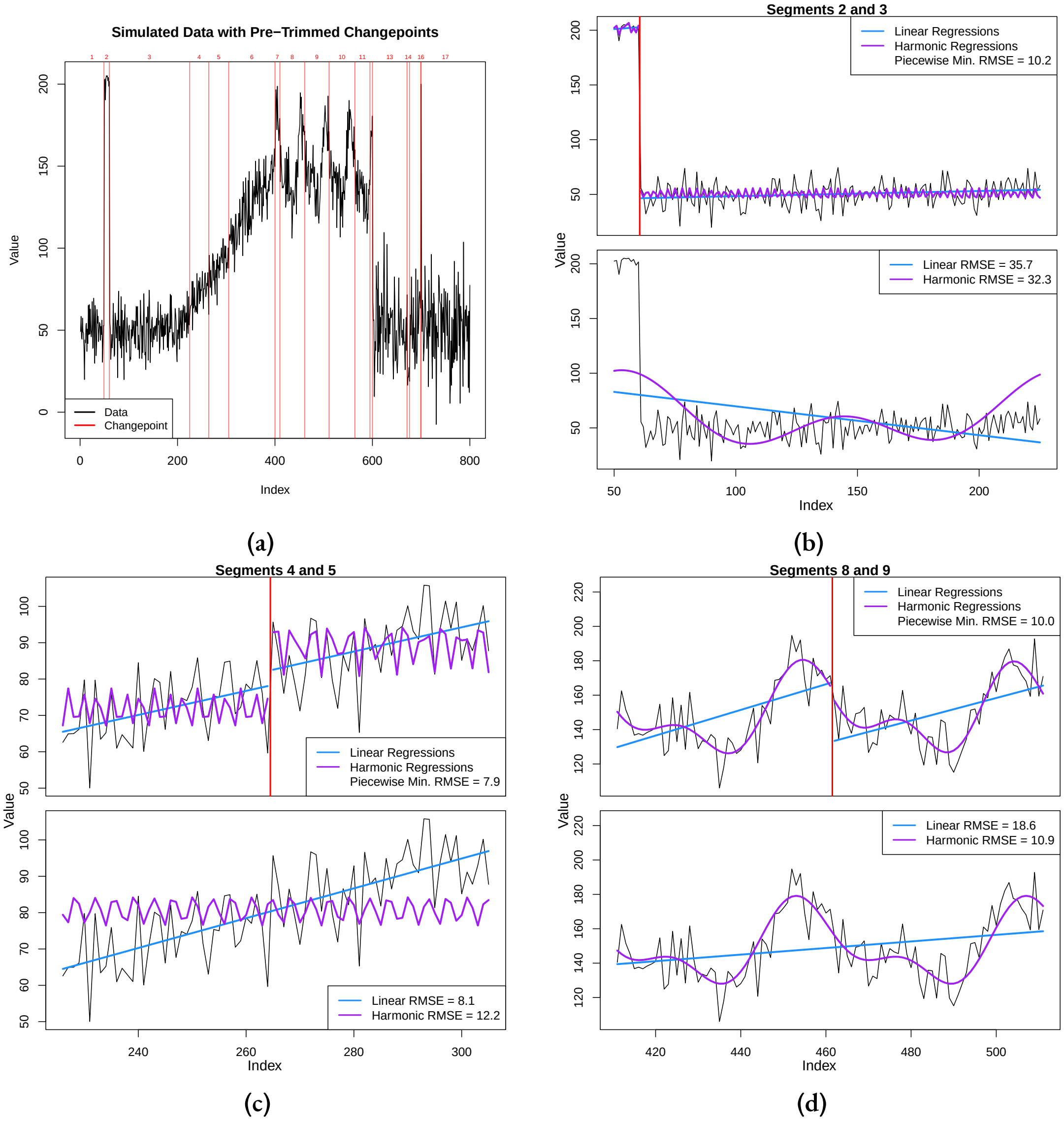
We illustrate ASCEPT's trimming process in Figure 1.3. Figure 1.3a shows a set of changepoints identified by Stage 1. For every changepoint, we perform two types of model fits. We first fit piecewise linear and harmonic regressions on each of the two segments located to either side of the changepoint. We then fit linear and harmonic regressions across the two segments, ignoring the changepoint. To fit the linear models, we regress the values in a segment against their indices. For harmonic regressions, we first estimate a segment's period using the frequency associated with the peak of the periodogram and then fit the harmonic regression with a linear model based on this estimate. For each type of model fit, we calculate the root mean square error (RMSE). For relevant changepoints, the piecewise fits should greatly outperform the cross-segment fits. However, for nuisance changepoints that are part of an ongoing linear or seasonal trend, the best cross-segment and piecewise fits should perform similarly.

To illustrate this, Figure 1.3b shows a sudden mean-shift at index 60. Here, the best piecewise fit outperforms the best cross-segment fit by nearly a factor of three, suggesting that this is a relevant changepoint. In contrast, Figure 1.3c shows a nuisance changepoint that is within a linear trend. In this case, a linear regression across both segments performs only marginally worse than the best piecewise fit to the segments. Similarly, Figure 1.3d shows a changepoint within a seasonal pattern. In that example, the cross-segment harmonic regression performs only marginally worse than the best piecewise fit.

ASCEPT preforms this process of fitting piecewise and cross-segment models for every changepoint identified in Stage 1. For each changepoint, we record the ratio of the RMSE for the best cross-segment fit to the RMSE for the best piecewise fit. The changepoint that corresponds to the smallest ratio is then removed if it falls below a chosen "trimming threshold." This process repeats for the remaining changepoints until no ratio falls below the threshold, as depicted in Figure 1.1b.

1.2.4 Segment Correction

We used changepoints identified by ASCEPT and CBS to fit constant, linear, and harmonic regressions to the corresponding segments. We declared a linear or harmonic regression to be the best fit to a segment if the ratio of the constant fit's RMSE to the best corresponding linear or harmonic regression's RMSE was greater than a given "fitting threshold." In these cases, we de-trended or de-seasonalized those segments. We then shifted and scaled all segments to match the location and scale of a chosen reference segment. The location was defined as the mean of the reference segment before any correction was performed and the scale was defined as the residual standard error for the best fitting model on that segment.



(a)

(b)

(c)

(d)

Figure 1.3: The trimming process in ASCEPT. (a) The simulated data with an initial set of changepoints. For illustrative purposes, only a subset of the changepoints found after running the first stage of ASCEPT is shown. (b) Assessing the changepoint between segments 2 and 3, a relevant changepoint. The cross-segment fits are more than 3 times worse than the best piecewise fit. (c) Assessing the changepoint between segments 4 and 5, a nuisance changepoint due to a linear trend. The cross-segment linear fit is only about 3% worse than the best piecewise fit. (d) Assessing the changepoint between segments 8 and 9, a nuisance changepoint due to seasonality. The cross-segment harmonic fit is only about 9% worse than the best piecewise fit.

1.2.5 Parameters

For all main text analyses, we ran Stage 1 of ASCEPT using a significance level of $\alpha = 0.01$ and $N = 10,000$ Monte Carlo simulations. We ran Stage 2 using a trimming threshold of 1.2, such that changepoints whose best cross-segment fit had an RMSE within 20% of the best piecewise fit were subject to removal. Supplementary Figure A.2 shows results from running ASCEPT on the simulated data using various trimming threshold values.

We ran CBS, as implemented in R's "DNAcopy" package [72], using a significance level of $\alpha = 0.01$ and 10,000 permutations. We set CBS's pruning threshold to 0.5; this yielded comparable results to ASCEPT's 1.2 trimming threshold in terms of the number of changepoints identified per time series.

For segment correction, we used a fitting threshold of 1.75. We shifted and scaled with respect to the seasonal segment, as captured by either ASCEPT or CBS, as the reference.

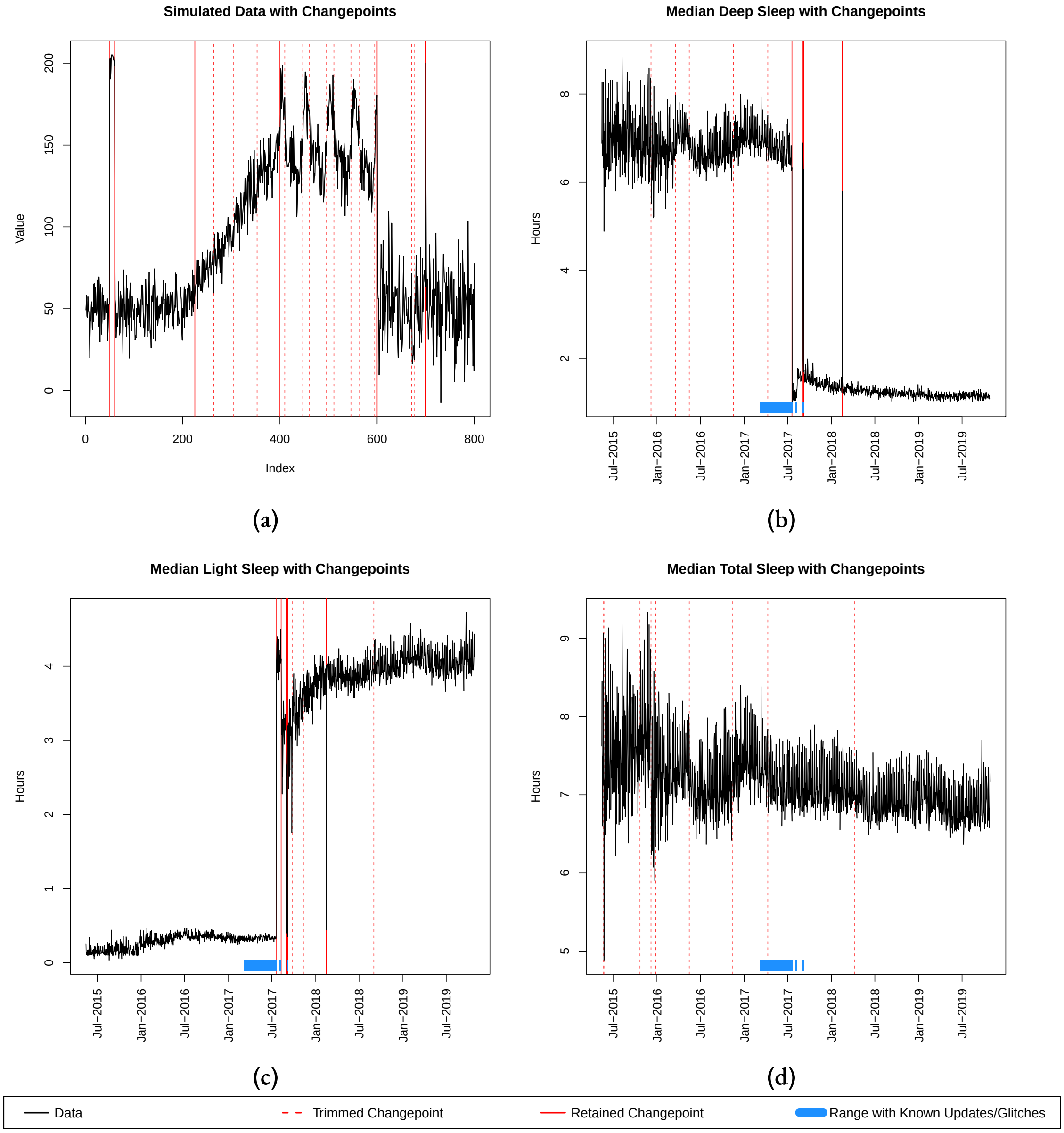
1.2.6 R Package

ASCEPT is implemented in "changepointSelect," an R package hosted on GitHub at `https://github.com/matthewquinnl/changepointSelect`.

1.3 Results

1.3.1 ASCEPT on the Simulated Data

When we first applied ASCEPT to simulated data, we found that Stage 1 detected relevant changepoints at indices 49, 60, 225, 400, 600, 699, and 700, as well as many nuisance changepoints that were subsequently trimmed in Stage 2 (Figure 1.4a). These results indicated that immediately after indices 49, 60, 225, 400, 600, 699, and 700, the simulated data experienced a statistically significant



(a)

(b)

(c)

(d)

Figure 1.4: Overall results from applying ASCEPT to (a) the simulated data, as well as mHealth data from the Precisions VISSTA study measuring (b) median deep sleep, (c) median light sleep, and (d) median total sleep.

mean-shift that was not attributable to an ongoing linear or seasonal trend. Among the relevant changepoints, five corresponded to sudden mean-shifts, while the other two segmented off the linear and seasonal trends.

1.3.2 ASCEPT on the Precision VISSTA mHealth Data

Next, we applied ASCEPT to mHealth data from the Precision VISSTA study. Figures 1.4b and 1.4c show the results for deep and light sleep. We observed similar results for these variables, which was expected because both contribute to total sleep. For deep sleep, ASCEPT identified changepoints on July 19, 2017, September 1, 2017, September 6, 2017, February 14, 2018, and February 15, 2018. For light sleep, it identified changepoints on July 19, 2017, August 9, 2017, August 31, 2017, September 6, 2017, February 14, 2018, and February 15, 2018. Based on this analysis, we hypothesize that Fitbit changed how it calculated sleep stage information immediately after these dates, impacting the relationship between deep and light sleep.

We further assessed these changepoints by cross-referencing with online information and found that some of the identified changepoints corresponded to known firmware updates and glitches. Alta HR received firmware update 26.62.6 between August 1, 2017 and August 10, 2017 [4], corresponding to the August 9, 2017 changepoint for light sleep. Likewise, Fitbit modified its calculation of sleep by introducing “Sleep Stages,” starting on March 6, 2017 [47]. Users reported glitches with Sleep Stages from within a week of the release through July 24, 2017 for Alta HR, Blaze, and Charge 2 devices [5], encompassing the changepoint on July 19, 2017. Users again reported glitches for Blaze devices between September 3, 2017 and September 7, 2017 [6], corresponding to the September 6, 2017 changepoint.

Next, we used the daily median total sleep as a negative control (Figure 1.4d). While there were some large fluctuations in median total sleep during 2015, this variance was likely because relatively few individuals (as few as seven; see Supplementary Figure A.1) contributed data on any given day.

Accordingly, ASCEPT did not identify any changepoints in this time series after trimming. We do not suspect that Fitbit changed the calculation of total sleep during the period of the study.

1.3.3 Comparison between ASCEPT and CBS

ASCEPT shares some principles with other methods, such as CBS [61] (see Appendix A.1). Therefore, we compared the changepoints identified by CBS to those identified by ASCEPT. For the simulated data (Figure 1.5a), we found that CBS failed to capture the single-point segment at index 700, while ASCEPT successfully did. ASCEPT also successfully segmented off the linear and seasonal trends, while CBS split the linear trend into four segments.

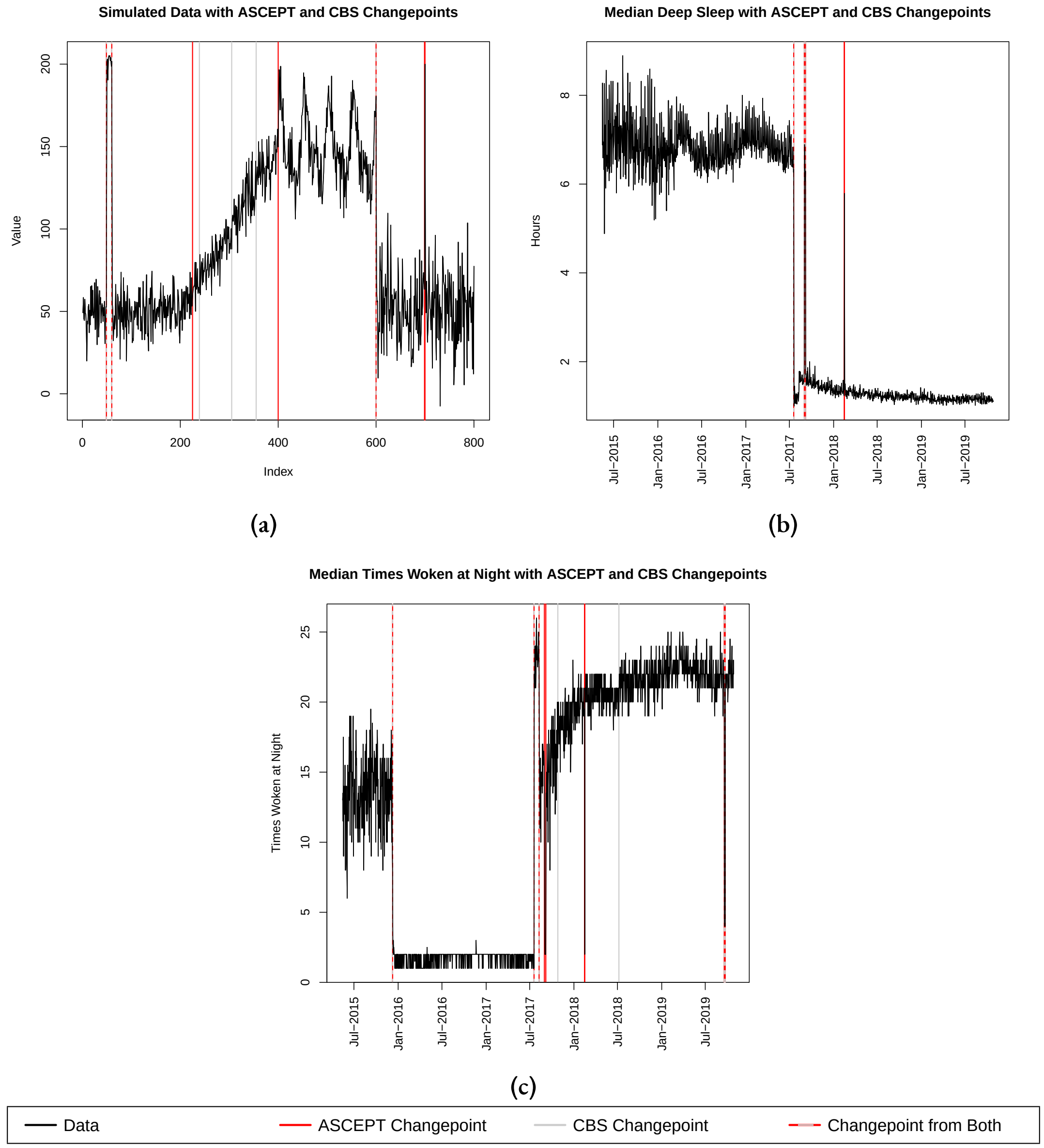
We also compared ASCEPT and CBS on mHealth data from the Precision VISSTA study. For most variables, the two procedures yielded similar changepoints, although there were some important differences. For example, CBS failed to detect changepoints for the single-day shift in deep sleep on February 15, 2018, while ASCEPT did (Figure 1.5b). The two procedures also greatly differed when applied to the times woken variable (Figure 1.5c). In particular, CBS failed to capture multiple changepoints from late 2017 to early 2018 and did not trim two nuisance changepoints that appeared to be within linear or seasonal trends. In contrast, ASCEPT successfully captured the major relevant changepoints and trimmed nuisance changepoints. We provide comparisons of ASCEPT and CBS for the remaining mHealth variables in Supplementary Figures A.3 and A.4, which demonstrate that ASCEPT generally outperformed CBS on real-world data.

While ASCEPT's primary purpose is to select changepoints, we also performed a simple correction to demonstrate the importance of accurately identifying changepoints. In particular, we found the best fit model for each segment (Figures 1.6a and 1.6b) and then adjusted the data to match the location and scale of the segment containing the seasonal pattern, which was accurately identified by ASCEPT as indices 401 to 600 inclusive and identified by CBS as indices 356 to 600 inclusive. If the changepoints were accurately identified, then we expected the transformed time series to look like normally distributed noise without any mean-shifts. We found this to be true for the ASCEPT segment-corrected time series (Figure 1.6c). In contrast, the CBS segment-corrected time series (Figure 1.6d) still contained trends, seasonality, and other mean-shifts due to the less accurate identification of changepoints. Supplementary Figures A.5 and A.6 show results when using fitting thresholds other than 1.75.

1.4 Discussion and Conclusion

We have presented an approach, ASCEPT, for identifying changepoints in mHealth data. ASCEPT builds upon the current state-of-the-art method, PELT, by incorporating the principles of statistical significance and trimming. ASCEPT adopts progressively larger sets of changepoints until the newly proposed set does not provide a statistically significant improvement in goodness-of-fit. ASCEPT then trims changepoints within linear or seasonal trends since, in mHealth data, these changepoints are often the result of behavioral or lifestyle changes rather than technological issues. This results in a set of changepoints that can be used to adjust mHealth data prior to additional downstream analysis.

ASCEPT offers many advantages over comparable methods. For example, using PELT to detect multiple changepoints requires specifying an optimization penalty while ASCEPT allows an investigator to specify a significance level, a more intuitive statistical parameter. Additionally, ASCEPT is specifically designed for mHealth data, which is not true of comparable methods like CBS. For instance, CBS uses a permutation test to obtain p-values for changepoints [61], but this approach has difficulty capturing segments containing only one observation, a feature we observed in the mHealth data from the Precision VISSTA study (Figure 1.1a). ASCEPT's Monte Carlo procedure does not run into this same problem and captures single-point segments (Figure 1.4b). In addition, CBS trims changepoints using a sum of squared within-segment deviations measure [61], while ASCEPT



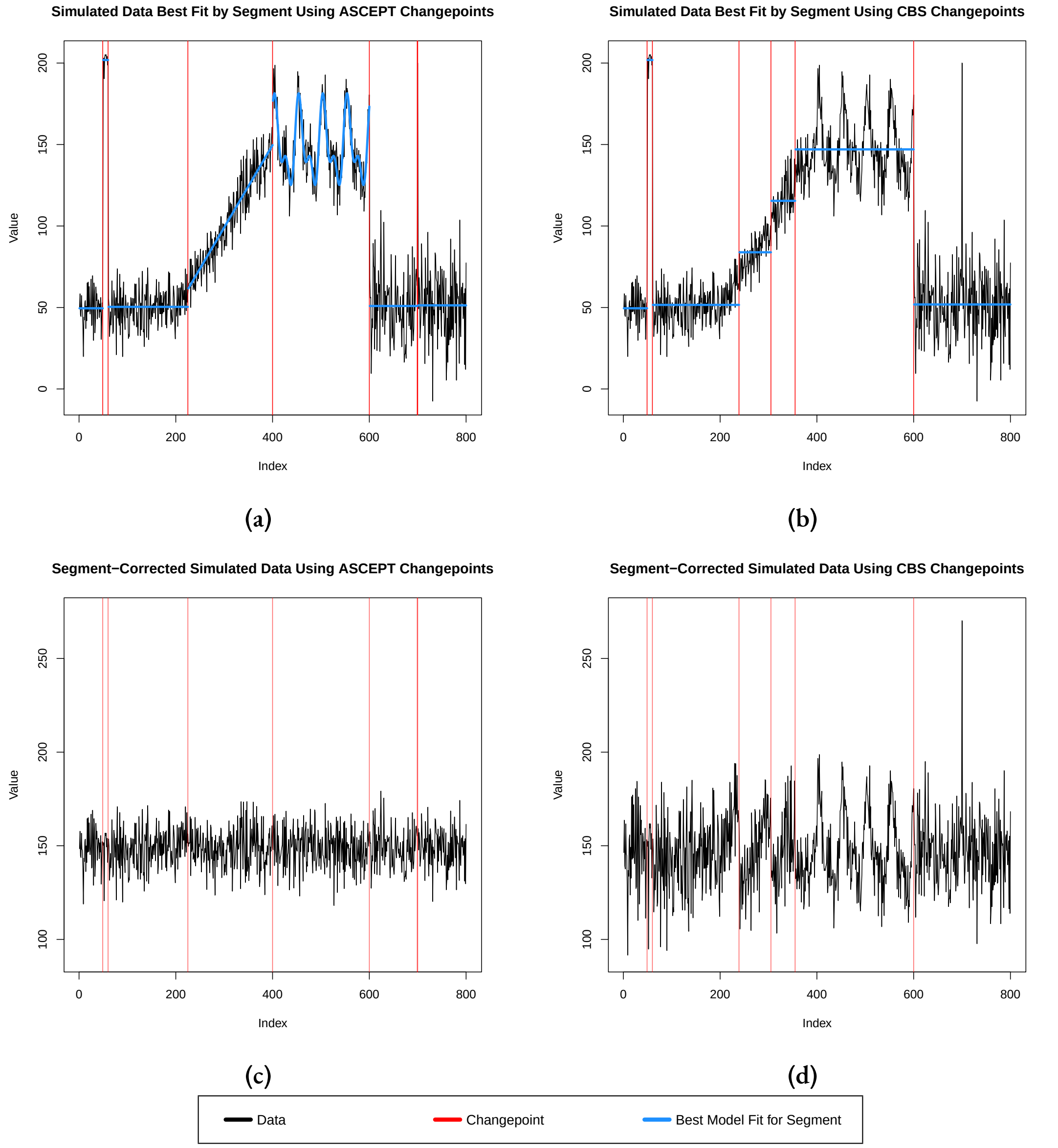
(a)

(b)

(c)

Figure 1.5: Comparison of ASCEPT with CBS when applied to (a) the simulated data, as well as mHealth data from the Precisions VISSTA study measuring (b) median deep sleep and (c) median times woken during the night.

directly models the linear and seasonal trends in mHealth data, thereby helping to differentiate between common behaviorally driven patterns and other patterns that may be a result of technological


Simulated Data Best Fit by Segment Using ASCEPT Changepoints

(a)

Simulated Data Best Fit by Segment Using CBS Changepoints

(b)

Segment−Corrected Simulated Data Using ASCEPT Changepoints

(c)

Segment−Corrected Simulated Data Using CBS Changepoints

(d)

Figure 1.6: Illustration of applying a simple correction process to simulated data after identifying changepoints using either ASCEPT or CBS. (a) The best model fits using ASCEPT changepoints. (b) The best model fits using CBS changepoints. (c) The corrected series using ASCEPT changepoints. (d) The corrected series using CBS changepoints.

changes, such as software or hardware updates to a wearable device.

Importantly, ASCEPT allows an investigator to identify potential technological changepoints automatically. For example, while Fitbit lists previous firmware versions online, it does not readily provide release dates or specific notes regarding each one [2]. Instead, a researcher needs to manually read through online community forums for details [1]. In our investigation of these online notes, we found that update rollouts and glitches often occurred over days or weeks, making it difficult to precisely determine when the data reflect these changes. Furthermore, some changes may not even be publicized, rendering a manual search useless. In contrast, ASCEPT provides an effective way to precisely identify when technologically driven changes occurred.

We note, however, that ASCEPT has some potential limitations. First, since it involves a Monte Carlo method, ASCEPT does not guarantee the same results over repeated runs; however, using a large number of simulations mitigates this issue. ASCEPT is also computationally intensive, but we parallelized its implementation for improved performance. In addition, ASCEPT assumes that the observations are normally distributed, which may not always be true. However, normality is appropriate to use in many scenarios, such as when using the sample mean or median of a variable [64], as we did in our application of ASCEPT to mHealth data from the Precision VISSTA study. Lastly, ASCEPT requires the selection of two thresholds: a significance level and a trimming threshold. While these parameters are intuitive, we recommend that investigators consider different trimming thresholds to select a value that is appropriate for their data. In our analyses, we found that the changepoints identified by ASCEPT were robust across a wide range of trimming threshold values.

While there are limitations, ASCEPT also has many strengths. For instance, while we developed ASCEPT for mHealth data and tested it on data from the Precision VISSTA study, the approach is generalizable. For example, a researcher could apply ASCEPT to select mean-shift changepoints in any univariate time series for which linear trends and seasonality induce nuisance changepoints. Additionally, instead of only applying ASCEPT to population-level data to identify technological changepoints, an investigator could apply ASCEPT to an individual's time series data to identify behavioral shifts that are not associated with broader seasonal or linear patterns. One could also modify ASCEPT to identify and remove nuisance changepoints within other trends, such as quadratic trends, which may be more common in other types of data [33]. One could similarly adjust ASCEPT's normality assumption to allow for other distributional assumptions. While the current presentation of ASCEPT uses PELT, a researcher could, in theory, also apply the same processes to other changepoint detection algorithms.

We designed ASCEPT as a formal process to select relevant changepoints among those proposed by PELT by modeling trends that are commonly associated with nuisance changepoints. Identifying these types of changepoints is a critical step for effectively analyzing mHealth data, which often contains changepoints both from sudden changes in the propriety algorithms used to record measurements and from changes in human behavior. ASCEPT automates this process and only requires selecting two intuitive parameters. This affords a distinct advantage over using other methods or performing a manual identification of technological changepoints, which supports ASCEPT's broad applicability to mHealth data analysis.

2

Detection and Statistical Inference for Differential Binding Sites

Abstract

ChIP-seq technology permits identifying DNA locations that interact with proteins, across the entire genome. In recent applications of this technology, biologists are interested in detecting regions showing differential binding (DB) across biological conditions or experimental groups. Although reliable algorithms, referred to as *peak callers*, are available for detecting binding sites, current computational approaches for DB detection do not provide accurate statistical inference. A major challenge in reporting uncertainty is that current methods depend on the outcome of peak calling algorithms to identify regions of interest, and that errors and uncertainty in this step are not all considered when performing statistical inference downstream. Furthermore, existing pipelines often do not appropriately account for within-sample autocorrelation or within-group variability. We overcome these issues by applying an integrated statistical approach to all samples, as opposed to current modular approaches that often first detect binding sites within individual samples and then detect DB within these sites. We apply a mixed-effects model to account for the different sources of variability and apply a permutation test to quantify uncertainty. We compare our method to two leading pipelines for the detection of DB sites on both simulated and experimental data. We find that our method improves on the performance of existing pipelines by providing reliable uncertainty summaries.

2.1 Introduction

Chromatin immunoprecipitation sequencing (ChIP-seq) is a process that provides a genome-wide profiling of protein binding sites on DNA [38]. Examples of ChIP-seq applications include the detection of transcription factor (TF) binding sites and histone modification (HM) enriched regions. Both TFs and histones are of interest because they are known to influence gene expression. When applied to medical contexts, for example when studying the molecular basis for disease [17, 42, 60], investigators are often interested in comparing different groups to evaluate the presence of differential binding (DB) sites [58, 50, 83, 24]. A number of data analysis pipelines have been developed [79] for the detection of DB sites. Some of these methods apply standard statistical tests to ChIP-seq data counts in predefined regions, such as transcription start sites associated with genes. Here, we focus on methods that can discover regions that are not necessarily predefined as this is one of the main advantages of whole genome technologies, such as ChIP-seq. Most approaches start by applying algorithms, referred to as *peak-callers*, such as MACS [88], HOMER [35], or SICER [84], to detect binding regions on each sample separately. Then in a next step, an ad-hoc approach is used to combine these regions across samples and then perform statistical tests. DiffBind [78, 69], for example, merges peaks identified across a chosen minimum number of different samples into regions referred to as *consensus peaks*. The part of a consensus peak of greatest enrichment is identified as a *summit* and a corresponding candidate region is constructed by extending upstream and downstream from this summit by a pre-specified value. In contrast, pipelines like csaw [54], PePr [87], and diffReps [73], slide a window of fixed size along the genome and merge neighboring windows to construct larger regions of interest. Furthermore, csaw applies a statistical test to each window and combines the p-values using Simes' procedure [75] to obtain a p-value for each region.

We find that current methods do not appropriately estimate false discovery rates. In practice, this can lead to incorrectly reporting DB sites. We note that, for example, DiffBind does not account for its selection procedure of candidate regions when performing inference, leading to an underestimate of false discovery rates. In the case of csaw, the use of Simes' procedure to combine p-values leads to overly conservative inferential statements due to the fact that neighboring windows are statistically correlated [75, 71, 70].

To overcome these challenges, we present a method that uses a sliding window approach and that appropriately models sources of variability while controlling region-level FDRs via a permutation test. Specifically, we first identify candidate regions using window-specific counts and then fit linear mixed effects (LME) models to estimate DB effects while accounting for autocorrelation and biological variability through random effects. We transform the data in such a way that the estimated DB effect is an approximately exchangeable statistic across regions. We then employ a permutation test to perform inference on the observed effects by pooling the exchangeable statistics into an empirical null distribution. This pooling enables the permutation test to perform well even for experiments with small sample sizes. We refer to our unified approach as *DBFinder*. We compare the performance of our approach to two of the most widely used approaches: DiffBind [78, 69] and csaw [54]. We do this using a simulation study and an application to real ChIP-seq data.

2.2 Data Description

The interest in performing experiments with multiple replicates to compare protein binding across groups is a fairly recent development. As a result, experiments with more than a couple biological replicates per group are not common. Here, we use a data set from a study that compared the transcriptional regulation of H3K4me3 in six samples from individuals with Huntington's Disease with six samples from neurologically normal individuals [24]. Additionally, to compare the different pipelines in settings where sensitivity, specificity, and observed FDRs can be analyzed, we used simulated data sets described below.

2.2.1 Simulated Data

The simulated data were generated from a modified version of the simulation study in [53]. This study included two different types of simulations. In the first we mimicked narrow or sharp peaks, often no more than about 200-300 base pairs wide, typically observed when studying TFs. The other mimicked broad peaks and *complex DB events* typically observed when studying HMs.

For every simulation, we generated 20,000 binding sites on a single chromosome. Of these sites, 1,000 were DB sites and 19,000 were non-DB. To mimic characteristics of TF binding, each binding event consisted of a single peak that was 200 base pairs wide, corresponding with a binding site in the center of the peak and an extension of 100 base pairs in either direction to reflect the average fragment length. Read counts for these sites were drawn from negative binomial distributions with group-specific means. The group-specific means were equal for non-DB sites and unequal for DB sites. To mimic biological variability that differs across binding sites, we assumed the dispersion factors of the negative binomial distributions followed an inverse scaled chi squared distribution with 20 degrees of freedom and a scaling factor of $\frac{1}{4}$.

To model characteristics of histone binding sites, complex binding events were 1,000 base pairs wide and consisted of three overlapping peaks. Correlated negative binomial counts were drawn for the peaks within a sample. The means of the negative binomial distributions were equal across groups. To induce DB, counts for one or two peaks in a given event were eliminated for one group's samples. To allow biological variability to differ by region, dispersion factors were sampled in the same manner as for the TF simulations.

To make these characteristics realistic, the correlation values for HM counts and the distribution of dispersion factors for all simulations were based on values estimated from the real study discussed in Section 2.2.2. To match smooth binding profiles seen in real experiments, counts for every peak were distributed according to a Beta(2,2) distribution. To include realistic noise, background enrichment was added to each simulation by drawing negative binomial counts and distributing counts uniformly over 2,000 base pair wide bins along the genome.

To assess how pipeline performance varied with sample size, we generated simulations for two experimental sizes. One included two samples in each group (*2 vs. 2*) and one included six samples in each group (*6 vs. 6*). We generated five simulations for each type of simulation for each experimental size.

For more details about the simulated data, please refer to Appendix B.1.1.

2.2.2 H3K4me3 in the Presence of Huntington's Disease

This study generated ChIP-seq data targeting H3K4me3 in postmortem prefrontal cortex samples from six individuals with Huntington's Disease and six individuals who were non-neurologic [24]. The authors performed DB analysis of H3K4me3 between the two groups and assessed the correlation between H3K4me3 enrichment and gene expression. We used this study for application and to adjust parameters of the simulation study in Section 2.2.1 because it was a relatively large ChIP-seq study (i.e., six samples in each group) and because its samples met most or all of the quality control metrics specified by Cistrome DB [57, 89]. Figure 2.1 displays an example of a region from

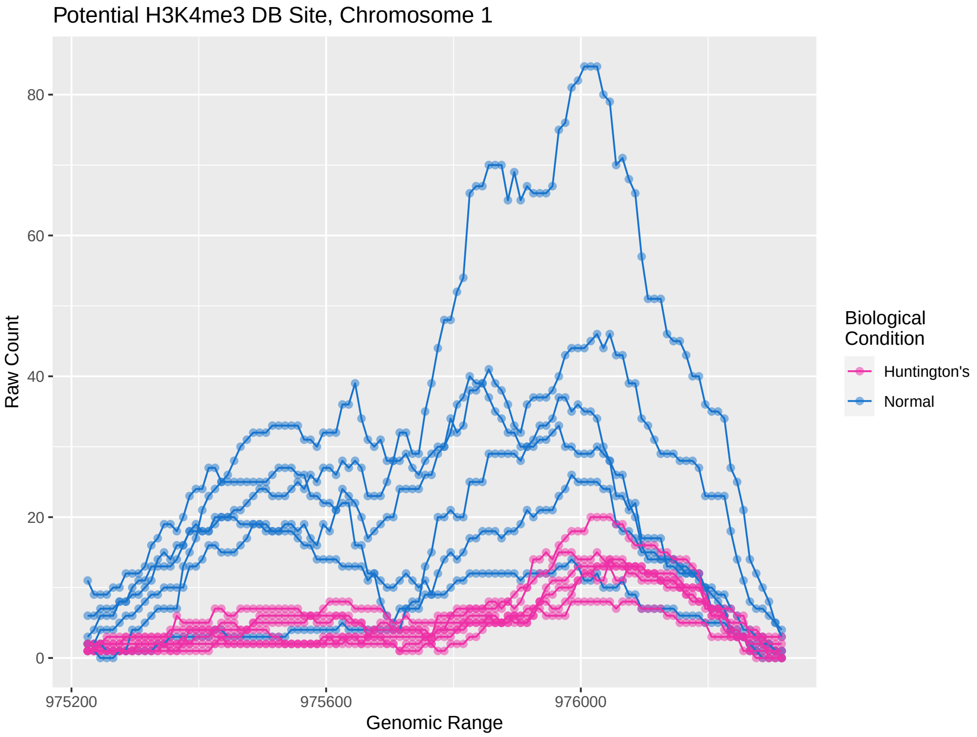


Figure 2.1: Example of a site with evidence of differential enrichment of H3K4me3 when comparing prefrontal cortex samples from six individuals with Huntington's Disease to six individuals who were neurologically normal. Each line represents an single sample.

these data that we would hope to detect because it exhibits some evidence of DB.

Additional details on the processing of these data can be found in Appendix B.1.2.

2.3 Analysis Framework

After processing the raw sequencing data into counts for small genomic windows and normalizing across samples (details described in the Appendix B.2.1), a two-step procedure is carried out. First, we define candidate regions by searching for contiguous windows with an average count difference above a predefined threshold and, second, we quantify the statistical significance of the observed DB in each region. This is done in a similar manner to the procedure developed by [46] for DNA methylation data. Below we describe each stage of the approach in detail.

2.3.1 Finding Candidate Regions

Note that the current DB detection methods first define binding regions by searching for local enrichment relative to background enrichment. However, this approach does not directly consider the primary outcome of interest: DB between groups. We instead define candidate regions using a difference in enrichment between groups as an initial DB effect estimate. Specifically, for each window j we compute the difference between the average counts in each group, and denote it as d_j. We then use a predefined threshold M to construct candidate regions. Each interval of contiguous windows for which $d_j > M$ or $d_j < -M$ for all windows in that interval is defined as a candidate region. Regions that contain gaps beyond a predefined size are partitioned into smaller candidate regions. Alternately, an investigator can specify a target number of candidate regions and M can be adjusted to obtain a number of candidate regions within a specified tolerance of that target.

2.3.2 Assessing Candidate Regions for Differential Binding

Given the identified candidate regions, we must quantify the evidence of differential binding within them and assess their significance. We do this in multiple substages, described in detail in the following sections.

2.3.2.1 Estimation of Differential Binding Effects Using LME Models

Data exploration indicate that ChIP-seq read count data, examined across samples from the same group, follow a negative binomial distribution. To accelerate the computational performance of our procedure by using LME models instead of generalized linear mixed effects models (GLMMs), we apply Anscombe's variance-stabilizing log transformation for negative binomial random variables

[12, 32]. Denote y_{ijr} to be the normalized and transformed count for sample i in window j in candidate region r, and assume that:

$$y_{ijr} = \beta_{0jr} + \beta_{1r} X_i + \eta_{1ir} Z_{1i} + \eta_{2ir} Z_{2i} + \varepsilon_{ijr} \tag{2.1}$$

We model the y_{ijr} for a given region r using an LME model that simultaneously accounts for autocorrelation within samples and biological variance across samples. We assume two groups, $g \in \{1, 2\}$, exist and that group g's corresponding set of samples is denoted as C_g. The β_{0jr} are window-specific fixed effects that account for how the binding profile changes over the course of a region. If the region contains a specified number of windows (i.e., if $W_r \geq 10$), then the fixed effects for windows are replaced with a natural cubic spline that accounts for smooth binding profiles common to many peaks. β_{1r} is the DB fixed effect of interest. $X_i = \mathbb{I}(i \in C_1)$ indicates which samples are in Group 1. We include random effects for each group specific to each sample to capture autocorrelation within that sample and biological variance within each group. η_{1ir} are the sample-specific random group 1 effects. $Z_{1i} = X_i = \mathbb{I}(i \in C_1)$ indicates which samples are in Group 1. η_{2ir} are sample-specific random group 2 effects. $Z_{2i} = \mathbb{I}(i \in C_2)$ indicates which samples are in Group 2. Lastly, ε_{ijr} is a within-sample error term. Optionally, an investigator can also include fixed effects to control for other variables of interest, such as blocks if using matched samples, or demographic information such as age and sex.

The LME model for candidate region r can be written in matrix notation as:

$$\mathbf{y}_r = \mathbb{X}_r \beta_r + \mathbb{Z}_r \eta_r + \boldsymbol{\varepsilon}_r \tag{2.2}$$

where $\mathbf{y}_r$ are the region's normalized and transformed counts, $\mathbb{X}_r$ is the region's fixed effects design matrix, β_r are the region's fixed effects, $\mathbb{Z}_r$ is the region's random effects design matrix, η_r are the region's random effects, and ε_r are the region's within-sample errors.

We estimate the LME models for candidate regions using `lmer()` in the `lme4` package in R [14]. We may measure the strength of the DB effect in region r using either $\hat{\beta}_{1r}$ or its corresponding moderated t-statistic, which is discussed in detail in Section 2.3.2.2. With either quantity, we assess its significance using a permutation test described in detail in Section 2.3.2.3.

2.3.2.2 Variances and Shrinkage

Instead of using $\hat{\beta}_{1r}$ from an estimate of Model 2.1 as the measure of the DB effect in a region, one could use the corresponding t-statistic that accounts for the uncertainty in this estimator. However, this uncertainty depends on both the biological variance and the within-sample error variance in a given region. Due to the small sample sizes common to ChIP-seq experiments, estimates of the biological variance will often be relatively poor. Therefore, the t-statistic for $\hat{\beta}_{1r}$ may not be a useful measure of the DB effect. However, we can impose structure on the biological variances across regions in order to improve the estimate of the biological variance within each region. With these improved estimates, we can calculate moderated t-statistics that more appropriately reflect the uncertainty in $\hat{\beta}_{1r}$. We do this following the empirical Bayes shrinkage procedure used by limma for residual variances in gene-specific linear models [65, 76]. We present the details of this process in the context of our LME models.

Given Model 2.2, we assume that the random effects are independent and that they follow a multivariate normal distribution:

$$\eta_r \sim \text{MVN}(0, \mathbf{T}_r)$$

with diagonal variance-covariance matrix $\mathbf{T}_r$. The variances on the diagonal of $\mathbf{T}_r$ reflect the biological variances across samples. There are two biological variances, one for each experimental group: $\text{Var}(\eta_{1ir}) = \tau^2_{1r}$ and $\text{Var}(\eta_{2ir}) = \tau^2_{2r}$. We also assume the errors to be independent conditional on random effects, such that:

$$\varepsilon_r \sim \text{MVN}(0, \sigma_r^2\mathbf{I})$$

where $\mathbf{I}$ is the identity matrix and $\text{Var}(\varepsilon_{ijr}) = \sigma_r^2$ is the region-specific error variance.

We obtain estimates of these biological variances and within-sample error variance using `lmer()` in the `lme4` package in R [14]. We note that in some cases, at least one biological variances may be estimated as 0, $\hat{\tau}_{1r}^2 = 0$ or $\hat{\tau}_{2r}^2 = 0$. However, in practice, neither group should have a biological variance that is exactly zero. Therefore, in candidate regions where this occurs, we instead fit a model with sample-specific random intercepts, η_{0ir}, that have only one biological variance associated with them. That is:

$$y_{ijr} = \beta_{0jr} + \beta_{1r}X_i + \eta_{0ir} + \varepsilon_{ijr} \tag{2.3}$$

We then use the estimated biological variance from this model, $\widehat{\text{Var}}(\eta_{0ir}) = \hat{\tau}_{0r}^2$ as the estimate of the group-specific biological variances for both groups. That is, we set $\hat{\tau}_{1r}^2$ and $\hat{\tau}_{2r}^2$ equal to $\hat{\tau}_{0r}^2$ and proceed.

Given the estimates of the biological and within-sample error variances, we can then write the estimated variance-covariance matrix of the log-normalized counts in region r as:

$$\widehat{\text{Var}}(\mathbf{y}_r) = \hat{\mathbf{V}}_r = \mathbb{Z}_r\hat{\mathbf{T}}_r\mathbb{Z}_r^T + \hat{\sigma}_r^2\mathbf{I}$$

The weighted least squares estimator of the fixed effects can be written as:

$$\hat{\beta}_r = (\mathbb{X}_r^T\hat{\mathbf{V}}_r^{-1}\mathbb{X}_r)^{-1}\mathbb{X}_r^T\hat{\mathbf{V}}_r^{-1}\mathbf{y}_r$$

We can then estimate the variance-covariance matrix of the estimated fixed effects as:

$$\widehat{\text{Var}}(\hat{\beta}_r) = (\mathbb{X}_r^T\hat{\mathbf{V}}_r^{-1}\mathbb{X}_r)^{-1}$$

which can be used to obtain the standard error and t-statistic for $\hat{\beta}_{1r}$.

However, in experiments with a small number of samples, it is difficult to obtain appropriate variance estimates, particularly for the group-specific biological variances. To address this, DBFinder applies the shrinkage procedure from limma [65, 76] separately to each set of biological variance estimates and the set of residual variances across candidate regions. The shrinkage procedure produces a estimate of each variance based on the mean of the posterior distribution of the true variances using an empirical Bayes approach. Using these posterior mean estimates, $\widetilde{\tau}^2_{1r}$, $\widetilde{\tau}^2_{2r}$, and $\widetilde{\sigma}^2_r$, we can recalculate a moderated estimate of the variance-covariance matrix of the fixed effects, $\widetilde{\text{Var}}(\hat{\beta}_r)$.

We then calculate a moderated t-statistic:

$$\tilde{t}_{1r} = \frac{\hat{\beta}_{1r}}{\widetilde{\text{s.e.}}(\hat{\beta}_{1r})}$$

where $\widetilde{\text{s.e.}}(\hat{\beta}_{1r}) = \sqrt{\widetilde{\text{Var}}(\hat{\beta}_{1r})}$. $\tilde{t}_{1r}$ can be recorded as a measure of the DB effect in region r instead of $\hat{\beta}_{1r}$.

2.3.2.3 Permutation Test

In order to provide inference for region-level DB effects that accounts for the selection procedure associated with finding candidate regions and that also implicitly accounts for biological variability, we use a permutation test. DBFinder permutes the group labels, $g \in \{1, 2\}$, of samples. If blocks are present, permutations are performed within them. For each permutation, the pipeline repeats the processes of finding candidate regions and modeling on the permuted data.

The pipeline uses the results obtained from permutations to generate an empirical null distribution of the desired region-level test statistic, either $\hat{\beta}_{1r}$ or $\tilde{t}_{1r}$. Since these statistics are approximately exchangeable between regions, results from regions across the genome are pooled together to form this empirical null distribution, allowing inference to be made in small sample settings. For instance, if only two permutations are performed, but 20,000 candidate regions are generated under each permutation, then the pipeline compares each observed candidate region's test statistic to the 40,000 test statistics from the permutation test. Using this empirical null distribution, DBFinder calculates region-level empirical p-values. It then reports region-level FDRs using the Benjamini-Hochberg procedure [15].

However, we find that using all possible permutations can cause multiple issues. In small sample settings, when reporting FDRs, it becomes difficult to identify any significant regions for small experimental setups because the observed results and the permutation in which the groups are entirely swapped comprise a nontrivial proportion of all possible permutations. In these cases, FDRs are reported beyond reasonable thresholds, even for sites that truly contain DB. Additionally, performing all permutations can become computationally prohibitive.

To overcome these issues, instead of performing every possible permutation of the experiment, DBFinder only performs *balanced* permutations. That is, the pipeline uses permutations that swap half, or as close to half as possible, of the available samples in a group with samples in the other group. This corresponds with the expected number of swaps made by a permutation in settings where the two groups have an equal number of samples. For instance, in an experiment with four samples in each group, we only consider permutations that swap two from each group. If the number of balanced permutations available is still computationally prohibitive, then we randomly select a subset of these permutations to use.

2.4 Results

We compare the performances of csaw, DiffBind, and DBFinder on the simulations. We evaluate both ROCs and inference to compare how accurately each pipeline ranks candidate regions and reports p-values and FDRs. Results are averaged over five simulations for each type and size of simulation. We also compare the candidate regions identified by each pipeline for the H3K4me3 ChIP-seq study [24].

Additional details regarding the settings used for each pipeline for both the simulated and H3K4me3 data sets can be found in Appendix B.2.2. Additional details regarding the calculation of ROCs and reported FDRs can be found in Appendix B.2.3.

2.4.1 Simulation Study

We present average partial ROC curves for each pipeline and experimental setup in Figure 2.2. The pipelines perform similarly in TF simulations. However, csaw slightly outperforms the others, particularly in the TF simulation with six samples in each group. In contrast, the performances vary more in the HM simulations that reflect complex binding events. DBFinder performs the best, whether using the estimated coefficient for the DB effect or the moderated t-statistic, followed by csaw, which is followed distantly by DiffBind. To visually demonstrate differences between the pipelines in their identification of these sites, we present a selection of sites and the corresponding candidate regions with reported FDRs from one of the HM 6 vs. 6 simulations in Figure 2.3. The approaches that csaw and DiffBind use to identify candidate regions ignore the primary metric of interest, the DB effect between groups. As a result, they often construct relatively large candidate regions that may contain a mix of DB and non-DB sites. For instance, DBFinder precisely identifies the DB portions of Sites 1-4 while csaw and DiffBind report relatively wide candidate regions that capture non-DB portions of Sites 1 and 3. Site 5 demonstrates a non-DB site identified by csaw and DiffBind but not DBFinder, though no pipeline identifies it as significant at any reasonable level. Likewise, Site 6 demonstrates a non-DB site in which DBFinder identifies a portion that appears to be DB by chance, though it again is not found to be significant at any reasonable level.

In Figure 2.4, we display histograms of p-values pooled across simulations for every experimental setup. These reflect the combined p-values reported by csaw and the region-level p-values reported

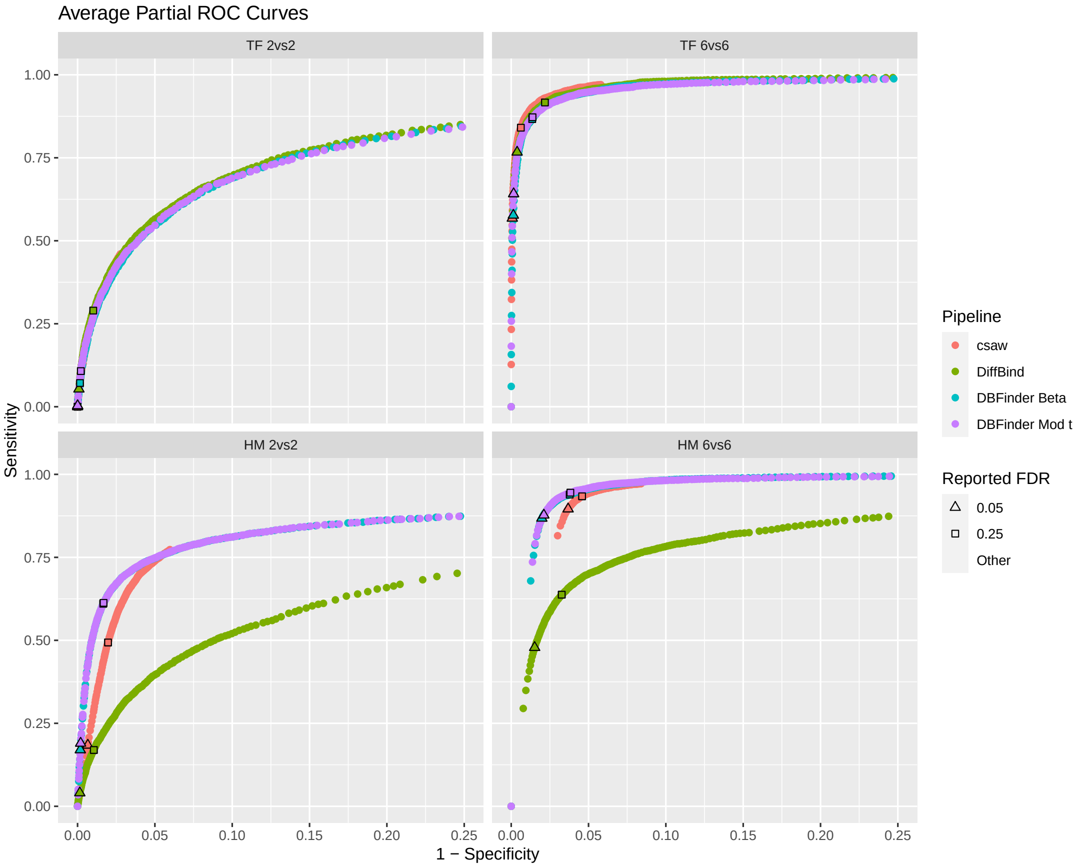


Figure 2.2: ROCs averaged over five simulations for each experimental setup (i.e., TF 2 vs. 2, TF 6 vs. 6, HM 2 vs. 2, HM 6 vs. 6). Points outlined in black from each pipeline correspond with a reported FDR of 0.05 or 0.25, depending on the shape of the outline.

by DiffBind and DBFinder. Ideally, all histograms should appear to follow zero-inflated uniform distributions. In practice, we find that this is approximately true for DiffBind and DBFinder across all experiments. However, we find that csaw's p-values do not exhibit this behavior for any experimental setup. Though they are zero-inflated, csaw's p-values are also inflated towards one and relatively few of them are middling.

We compare observed FDRs with those reported by each pipeline in Figure 2.5. All pipelines report relatively few regions with low FDRs for 2 vs. 2 experiments, compared with 6 vs. 6. Regardless,

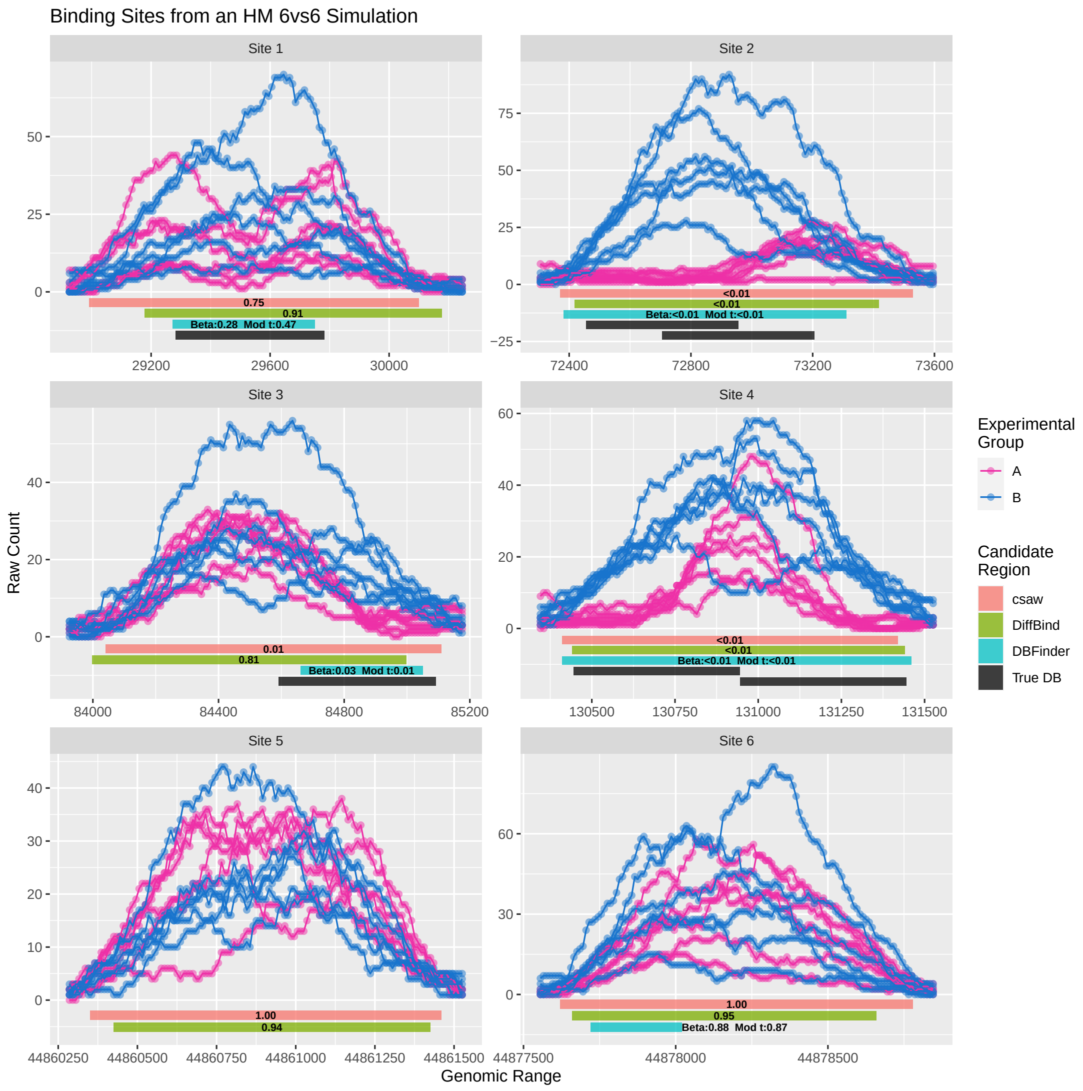


Figure 2.3: Binding sites from one of the HM 6 vs. 6 simulations. The first four sites are DB sites while the last two do not have any DB. The corresponding FDRs reported by each pipeline are within or immediately adjacent to their corresponding candidate regions at the bottom of each site.

we consistently find that DiffBind has relatively liberal behavior, reporting FDRs that are lower than what is actually observed, while csaw has relatively conservative behavior, reporting FDRs that are higher than what is actually observed. csaw's conservative FDRs align with its relatively high

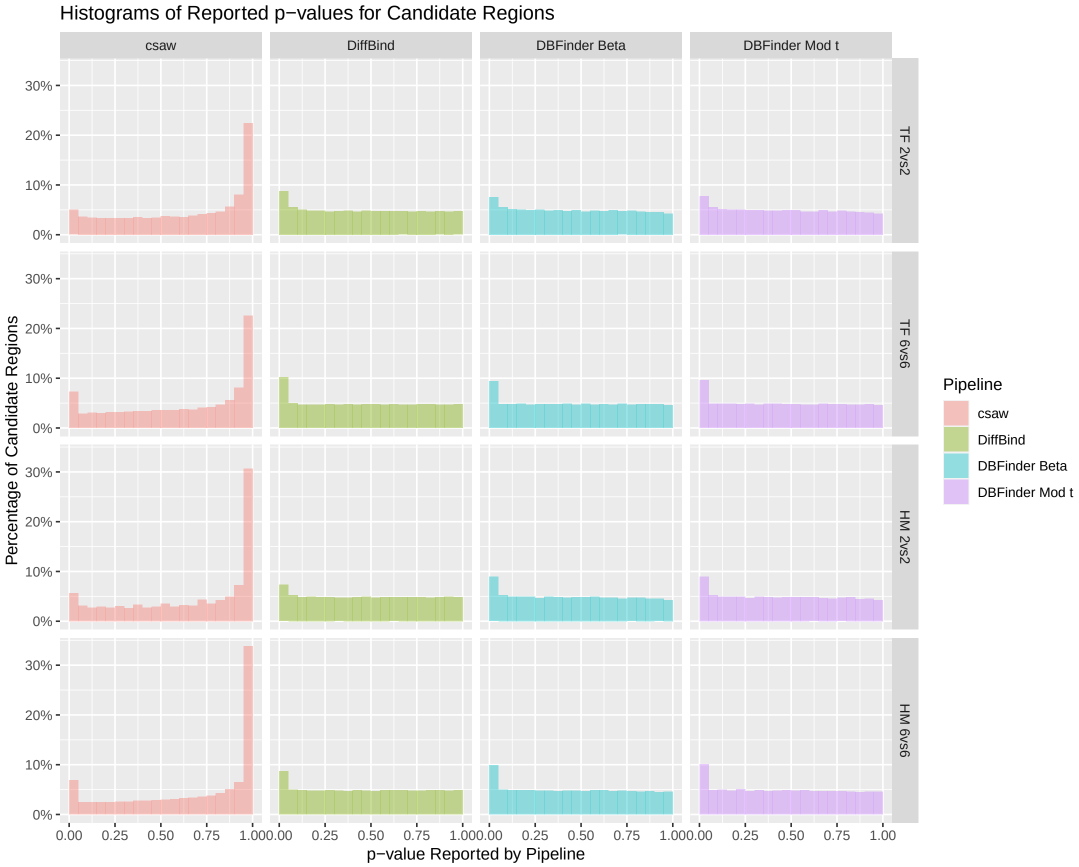


Figure 2.4: Histograms of reported p-values for candidate regions from each pipeline. csaw reports a combined p-value for each candidate region by applying Simes' method to individual window-level p-values. DiffBind and DBFinder outright report a p-value for each candidate region. p-values for the same experimental setup (i.e., TF 2 vs. 2, TF 6 vs. 6, HM 2 vs. 2, HM 6 vs. 6) are pooled together from across five simulations.

p-values in Figure 2.4. DBFinder tends to report FDRs that exactly or very closely align with what is observed. Each pipeline is implicitly forced to be conservative for relatively high FDRs because 5% of binding sites in each simulation truly contain DB.

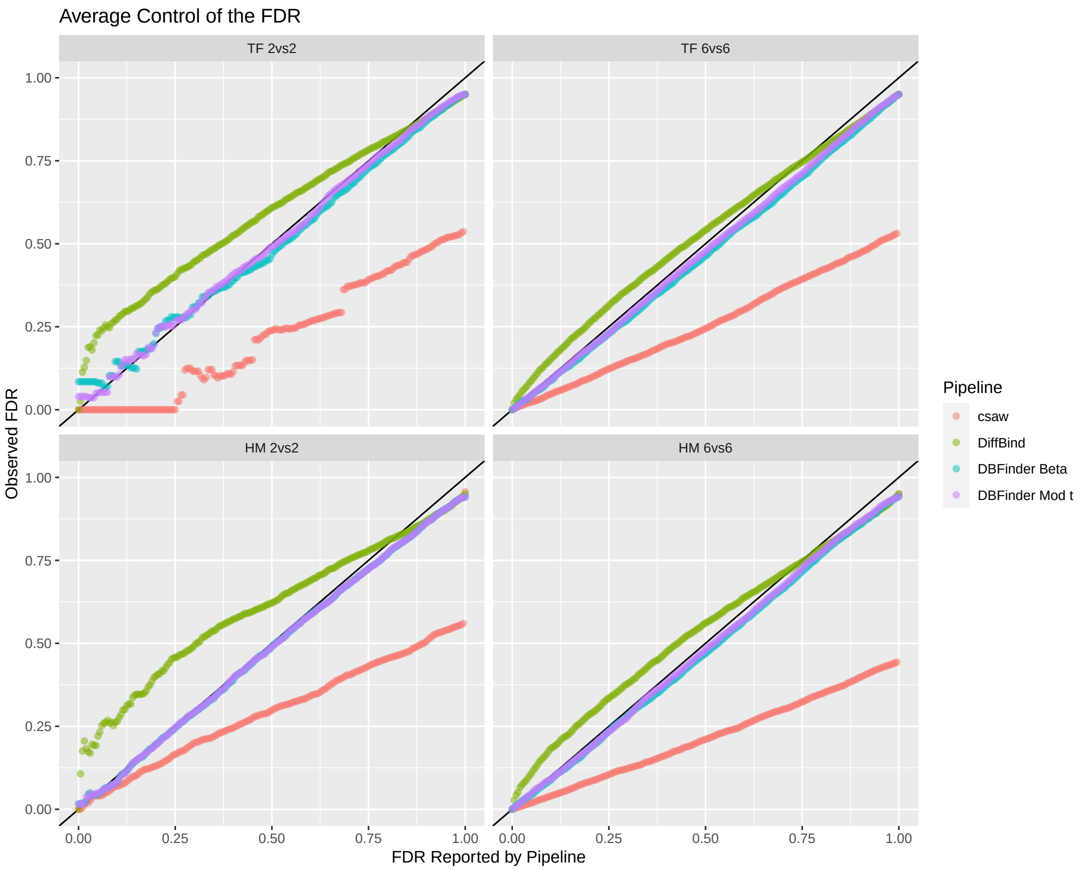


Figure 2.5: Observed FDRs averaged over five simulations against reported FDRs from each pipeline for each experimental setup (i.e., TF 2 vs. 2, TF 6 vs. 6, HM 2 vs. 2, HM 6 vs. 6). The identity line in black indicates perfect control of the FDR. However, no pipeline can achieved perfect control of the FDR for relatively high FDRs because 1,000 binding sites out of 20,000 are DB sites.

2.4.2 H3K4me3 in the Presence of Huntington's Disease

We compare the candidate regions identified by each pipeline for the H3K4me3 ChIP-seq study [24], described in Section 2.2.2. In Table 2.1, we present the number and percentage of candidate regions overlapping between every pairing of pipelines. We find that csaw and DiffBind agree on defining the vast majority, 83-92%, of their candidate regions. In contrast, DBFinder's candidate regions only overlap with 42-49% of those from csaw and DiffBind. However, all three pipelines only agree on around 47%-65% of candidate regions with reported FDRs below 0.05.

Table 2.1: The number and percentage of candidate regions overlapping between every pair of pipelines, requiring an overlap of at least 25 base pairs. The top half uses all candidate regions while the bottom half only uses candidate regions with reported FDRs < 0.05. An entry in row *i*, column *j* of either half of the table indicates the number and percentage of pipeline *j*'s candidate regions that are caught by pipeline *i*'s candidate regions.

Any Region	csaw	DiffBind	DBFinder (Beta)	DBFinder (Mod t)
csaw	21282 (100.00%)	20106 (82.68%)	19631 (99.11%)	19631 (99.11%)
DiffBind	19585 (92.03%)	24319 (100.00%)	17349 (87.59%)	17349 (87.59%)
DBFinder Beta	10380 (48.77%)	10143 (41.71%)	19808 (100.00%)	19808 (100.00%)
DBFinder Mod t	10380 (48.77%)	10143 (41.71%)	19808 (100.00%)	19808 (100.00%)
FDR < 0.05	csaw	DiffBind	DBFinder (Beta)	DBFinder (Mod t)
csaw	3583 (100.00%)	2209 (51.64%)	3238 (47.33%)	3244 (47.29%)
DiffBind	2176 (60.73%)	4278 (100.00%)	4161 (60.82%)	4171 (60.80%)
DBFinder Beta	2058 (57.44%)	2774 (64.84%)	6842 (100.00%)	6840 (99.71%)
DBFinder Mod t	2062 (57.55%)	2781 (65.01%)	6837 (99.93%)	6860 (100.00%)

To visually demonstrate differences at the region-level for real data, we show a selection of sites detected by the pipelines and their reported FDRs in Figure 2.6. For all sites, DBFinder defines relatively narrow and precise regions compared with csaw and DiffBind. For instance, in Site 3, DBFinder targets what appears to be a DB site within the larger region. In contrast, csaw and DiffBind capture the region as a whole, which appears to have comparatively less evidence of DB. Site 6 shows a case where DBFinder misses what appears to be a rather clear DB portion of the site to the left of its candidate region. Both csaw and DiffBind include portions of Site 6 that seem to lack evidence of DB. We also find that even when candidate regions are similar, the pipelines can report disparate FDRs. For instance, in Site 2, which contains a single outlier sample, DBFinder reports relatively high FDRs, around 0.21, compared to csaw and DiffBind, which report FDRs around 0.01 and 0.08, respectively.

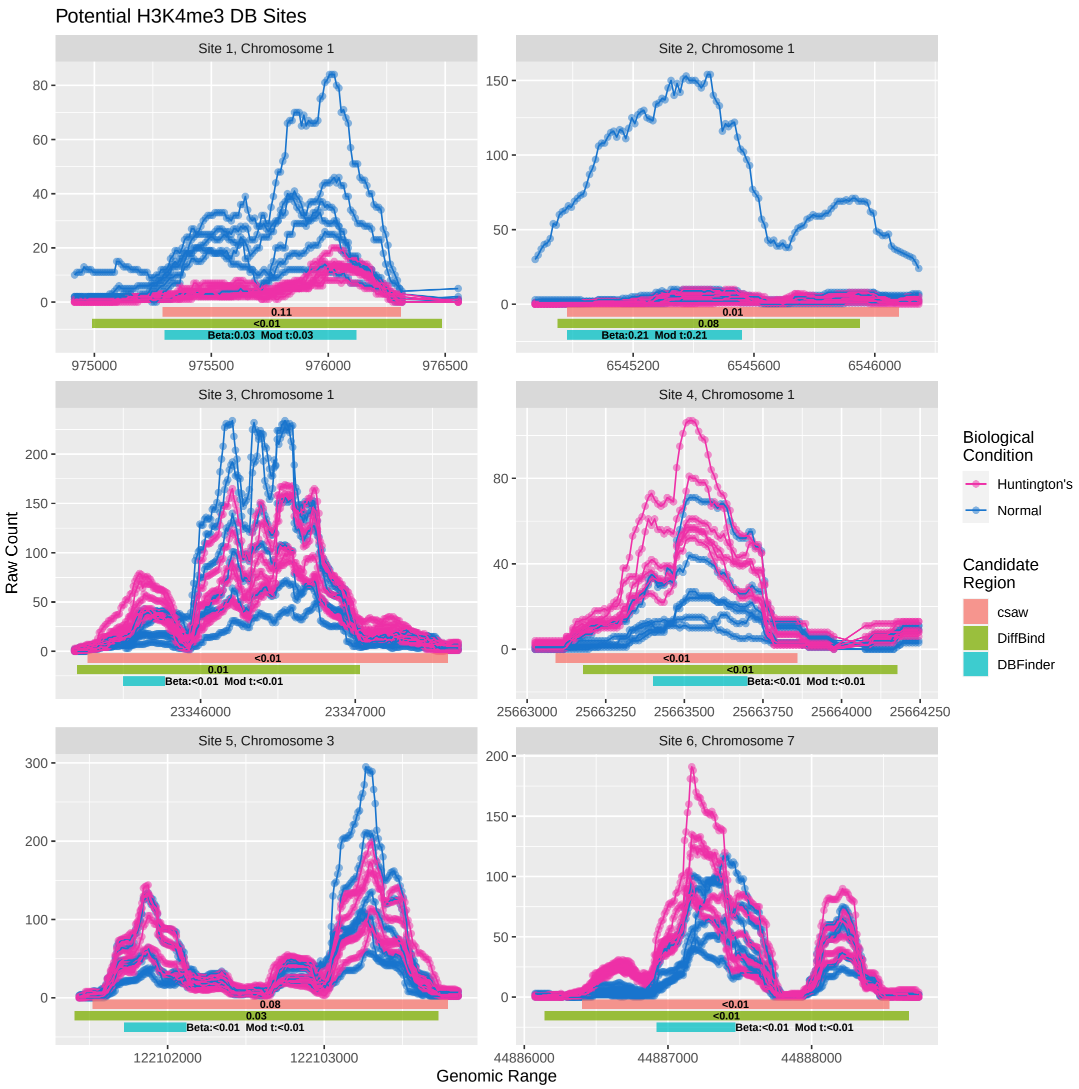


Figure 2.6: Sites with possible differential enrichment of H3K4me3 comparing prefrontal cortex samples from individuals with Huntington's Disease to those who were neurologically normal. The corresponding FDRs reported by each pipeline are within or immediately adjacent to their corresponding candidate regions at the bottom of each site.

2.5 Discussion and Conclusion

We have presented a new pipeline, DBFinder, for the de novo detection of DB sites using ChIP-seq data. Like other window-based approaches, DBFinder uses a sliding window along the genome to identify DB candidate regions. However, unlike other methods, DBFinder provides appropriate modeling and inference of the DB effect in candidate regions. We transform data using Anscombe's variance-stabilizing log transformation for negative binomial data [12]. This enables us to both model transformed values as approximately normal using LME models and to estimate a statistic that is approximately exchangeable across candidate regions along the genome. In these LME models, we explicitly account for autocorrelation and biological variance through random effects. We then use a permutation test with pooling to form an empirical null distribution against which we compare observed results. This permutation test accounts for the selection procedure and also implicitly incorporates biological variability into inferential statements while pooling makes the permutation test appropriate in small sample settings.

We compared DBFinder to two popular pipelines for detecting DB sites, csaw and DiffBind. We found that DBFinder performed better in terms of ROCs in HM simulations with complex binding events and also more accurately reported FDRs for all simulations than these competing methods.

While DBFinder is a useful approach for de novo detection of DB sites, there are some limitations worth noting. DBFinder is computationally intensive compared with methods such as csaw and DiffBind. We have taken care to optimize DBFinder where possible, but much of its runtime is attributable to fitting LME models and recalculating the standard errors of the estimated coefficients for the DB effects after shrinking variances. Currently, an analysis using DBFinder to identify 20,000 candidate regions using both estimated coefficients and moderated t-statistics in parallel with four cores and five permutations will take on the order of one to two hours. The runtime scales roughly in proportion to the number of candidate regions, such that reducing the desired number of candidate regions can reduce runtime by approximately same relative amount. However, the number of candidate regions should be kept at a number that is sufficiently high such that there are no concerns about missing true DB sites. In practice, this is not known, so it is safest to set it to approximately match the number of genes on the genome analyzed (e.g., around 20,000 for the human genome). While runtime is also roughly proportional to the number of permutations, DBFinder requires very few permutations to run properly because it pools results from across the genome. There appears to be little benefit to using more than about five permutations with DBFinder.

The second possible limitation of DBFinder is that it relies on a transformation to model counts as approximately normal and to produce an approximately exchangeable statistic across regions. In cases where window counts are relatively small, this may not be as appropriate as directly modeling counts as negative binomial. This is likely the reason why DBFinder slightly underperforms in TF simulations compared to csaw and DiffBind, which implicitly assume negative binomial counts when performing DB analysis through edgeR. In theory, one could adjust DBFinder to model regions using negative binomial GLMMs to overcome this issue, but this would also increase its runtime. Despite this concern, we found that DBFinder was competitive with csaw and DiffBind in all simulations, where counts were truly sampled from negative binomial distributions, giving csaw and DiffBind an advantage.

DBFinder shows promise as a helpful pipeline for investigators looking to identify DB sites using ChIP-seq data. We have found that this approach performed favorably to comparable methods, particularly in the presence of complex binding events and with regards to inference. Investigators who are concerned about obtaining exact p-values and FDRs, especially in the presence of autocorrelation or high biological variance, should consider using DBFinder as a tool in their analysis of DB sites.

3

Characterization of Spurious Correlations in RNA-seq Data

Abstract

Many studies in genomics analyze associations between different genes to better understand biological mechanisms and regulatory networks. These analyses often entail measuring the co-expression of genes using RNA-seq data. However, technical issues with sequencing and common normalization procedures may introduce or exacerbate spurious correlations in these data. In this study, we discuss different measures and statistical tests that enable a researcher to identify the presence of these spurious correlations. We demonstrate these procedures using RNA-seq data from the Genotype-Tissue Expression (GTEx) Consortium and provide evidence of spurious correlations in both raw and smooth quantile (qsmooth) normalized data for Y chromosome genes and tissue-specific genes.

3.1 Introduction

Ribonucleic acid sequencing (RNA-seq) is a popular technology that enables researchers to study the expression of genes for a wide variety of biological, clinical, and pharmaceutical applications [20, 23, 43, 31]. In addition to gene expression itself, researchers may analyze gene co-expression, or how different genes activate in a coordinated manner across samples. Many researchers analyze gene co-expression using RNA-seq data to study associations among genes, systems that regulate genes, and the roles that genes serve in different biological mechanisms [82, 85, 40, 11]. These studies commonly perform co-expression analysis for only one type of tissue at a time [86, 30, 29], using relatively homogeneous data sets where all samples arise from the same biological condition.

In settings where samples are homogeneous, researchers may use normalization procedures, such as quantile normalization or relative log expression (RLE) normalization, that assume differences across samples are due to technical, rather than biological, reasons [8, 16, 10]. However, many experiments involve heterogeneous data sets that incorporate samples from multiple types of biological conditions. For example, investigators may compare samples across different tissues, sexes, disease statuses, or may not even know the comparison of interest a priori. In such scenarios, these normalization procedures may improperly ignore biologically meaningful differences across samples. For instance, quantile normalization matches quantiles from different samples in order to force the samples to have identical distributions. However, samples from different biological conditions will often have inherent distributional differences that need to be preserved when performing downstream analyses. A popular normalization procedure that addresses this issue is smooth quantile normalization (qsmooth) [36, 62]. In contrast to quantile normalization, qsmooth accounts for biological condition to avoid erasing biologically-driven differences between samples when normalizing data. In particular, qsmooth determines the quantile for a sample based on a weighted average of the normalized quantile across all samples and the normalized quantile for that sample's biological condition.

While qsmooth appropriately handles biological differences across tissues, it also introduces new challenges in the context of co-expression analysis. In particular, qsmooth values may introduce or exacerbate spurious correlations in heterogeneous data sets. That is, genes can appear to have strongly correlated expression even if they should only be weakly correlated or uncorrelated by definition. For instance, prior research has found some evidence of spurious correlations among "tissue-specific" genes using RNA-seq data from the Genotype-Tissue Expression (GTEx) Consortium [37, 27, 28]. The authors of [37] identified genes that appear to be specifically expressed in one of four sample types: cerebellum, whole blood, lymphoblastoid cell lines (LCL), and liver. They presented examples in which these tissue-specific genes from different sample types have highly correlated qsmooth expression data, especially when comparing genes specific to whole blood, LCL, and liver. However, we expect that these tissue-specific genes from different tissues should be weakly co-expressed, if co-expressed at all, because their expression is specific to disjoint sets of samples by construction. While the authors presented an approach for adjusting expression values to remove spurious correlations, they did not further explore the existence of spurious correlations across other sets of genes and did not suggest metrics for detecting spurious correlations before applying a correction procedure.

In this paper, we present evidence of spurious correlations in GTEx data normalized using qsmooth by analyzing genes specific to the Y chromosome and genes specific to particular tissues. We explore how spurious correlations manifest in different tissues and how qsmooth expression data compare with raw expression data in their generation of false positive co-expressions. We present multiple approaches for identifying these spurious correlations and for testing their statistical significance.

3.2 Materials and Methods

3.2.1 Notation

Consider an experiment with G genes and N samples. The N samples are split among T tissues, or biological conditions more generally, where n_t indicates the sample size for tissue t such that $\sum_{t=1}^{T} n_t = N$. We let t_k indicate the k^{th} sample for tissue t, where $1 \leq k \leq n_t$.

Let $e_j^{(t_k)}$ indicate the expression value for gene j in sample t_k on the $\log_2$ scale. In cases where raw expression values are used, we add one before taking the log to avoid taking the log of zero. The co-expression for two genes, i and j, in tissue t is the Pearson correlation between their $\log_2$ expression values across all tissue t samples $t_k, 1 \leq k \leq n_t$. We denote this co-expression as $\rho_{ij}^{(t)}$. While we use Pearson correlation in our analyses, the methods discussed later are likewise appropriate for other correlations, such as Spearman's rank correlation.

3.2.2 Data

We downloaded RNA-seq data from GTEx release 6.0, a large-scale project that aims to provide data relevant to studying tissue-specific gene expression and regulation [27, 28]. The data were preprocessed using the Yet Another RNA Normalization (YARN) software pipeline in the same manner as described in [62]. In brief, samples were initially filtered based on annotation quality regarding sex identification. Different regions within body sites were then merged together if they were not distinguishable based on the first two principal coordinates from a principal coordinate analysis (PCoA) [26]. Genes were then filtered using a tissue-aware procedure that identified relatively more tissue-specific and differentially expressed genes compared to a tissue-agnostic approach and compared to the unfiltered data. The pre-processed data set contains RNA-seq count data on 30,333 genes across 9,435 samples taken from 549 individuals from 38 biological conditions (i.e., tissues and cell lines after merging subregions). 5,963 of the samples come from men while 3,472 come from women.

As part of our analysis, we choose to focus on genes for which correlations should be known to be either biologically reasonable or spurious. For this reason, we primarily analyze genes located on the Y chromosome, which would yield spurious correlations in female samples. In Figure 3.1, we display the qsmooth expression and co-expression data for these Y chromosome genes within skin samples. As expected, we find that female skin samples generally exhibit lower expression of these genes than male skin samples. The four most highly expressed genes in the female samples are pseudogenes (EIF4A1P2, PSMA6P1, CD24P4, MXRA5P1). However, we also find that the genes are highly co-expressed in female skin samples compared to male skin samples. Additionally, while nearly all of these Y chromosome gene are strongly co-expressed in female skin samples, the highest levels of co-expression occur among genes with similar median expression levels. We observe this same characteristic in male samples, where only lowly expressed genes and a few highly expressed genes exhibit high co-expression.

We also identify genes specific to each tissue, which could yield spurious correlations in tissues other than the one to which they are specific. For example, we expect that genes specific exclusively to liver should be weakly correlated or uncorrelated in spleen samples. We identify a gene as specific to a particular tissue using the same metric as in [77]. Specifically, let $m_j^{(t)}$ be the median expression level of gene j in samples from tissue t, $m_j^{(t)} = \text{med}\left(e_j^{(t_1)}, e_j^{(t_2)}, \ldots, e_j^{(t_k)}, \ldots, e_j^{(n_t)}\right)$. Likewise define $m_j^{(\text{all})}$ to be gene j's median expression across all samples and $\text{IQR}_j^{(\text{all})}$ to be the corresponding IQR. Then the specificity score of gene j to tissue t is defined as:

$$s_j^{(t)} = \frac{m_j^{(t)} - m_j^{(\text{all})}}{\text{IQR}_j^{(\text{all})}}$$

We consider gene j to be specific to tissue t if $s_j^{(t)} > 2$.

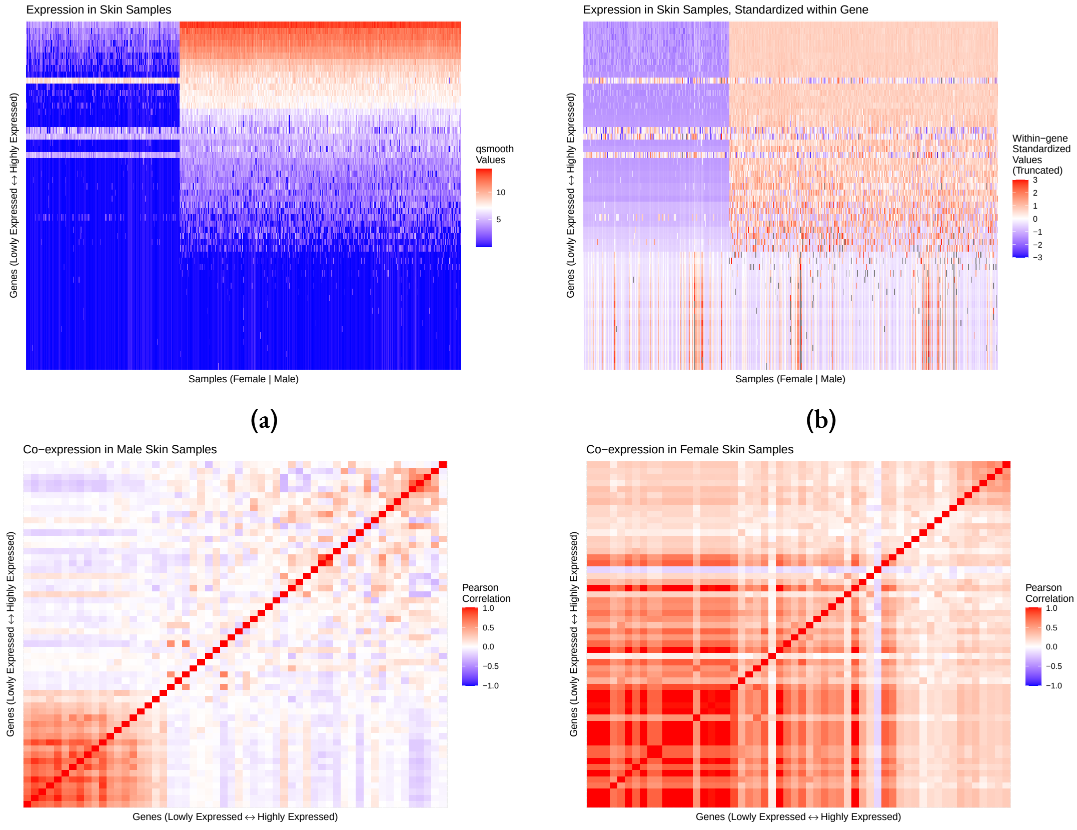
The multiplicity of a gene is the number of tissues to which it is considered specific. In our analysis, we select a subset of all tissues available in the GTEx data. We restrict our analysis to samples within that subset of tissues and to genes with a multiplicity of one among that subset of tissues. With this approach, we expect that the strongest co-expressions should be between genes specific to the same tissue, within samples of that tissue. Technically, genes associated with the same tissue could also co-express outside of that tissue. Likewise, genes associated with different tissues could co-express. However, we consider these associations to be false positives for our analysis. We refer to the tissue to which a given gene is specific as its "target" tissue. For instance, liver is the target tissue for liver-specific genes. The tissues to which a gene is not specific are its "non-target" tissues.

3.2.3 Co-expression Measures

In order to find spurious correlations, a researcher could consider looking at correlation matrix heatmaps for samples in which genes should not be co-expressed, as shown for Y chromosome genes in female skin samples in Figure 3.1d. However, this may be tedious in cases where an investigator wants to check for spurious correlations in many different biological conditions (e.g., tissue, sex, disease status) or wants to check different sets of genes for spurious correlations. A correlation matrix heatmap may also make it difficult to assess if spurious correlations are actually present since some pairs of genes may have relatively high correlations due to noise alone. Additionally, we find that spurious correlations may be associated with the expression level. In this case, it is helpful to have another measure that can make it easier to visualize the association between co-expression and expression.

An alternate approach to the correlation matrix heatmap is to use a summary measure. In particular, we may take:

1. The median co-expression of the upper triangle of the correlation matrix for all pairs of genes



(a)

(b)

(c)

(d)

Figure 3.1: Expression and co-expression of Y chromosome genes in skin samples, using qsmooth expression values on the $\log_2$ scale. The plots reflect data from 428 male samples and 233 female samples. Genes are sorted by median expression in male skin samples. Subfigures show (a) expression of genes in all skin samples, (b) same as (a) but with values standardized within-gene, (c) co-expression of genes within male skin samples, (d) co-expression of genes within female skin samples.

in tissue t,

$$\rho^{(t)}_{\text{med}} = \text{med}\left(\rho^{(t)}_{12}, \rho^{(t)}_{13}, \ldots, \rho^{(t)}_{1n_t}, \rho^{(t)}_{23} \ldots, \rho^{(t)}_{(n_t-1)n_t}\right)$$

Equivalently, this is the median co-expression among all discordant pairs of genes in tissue t.

2. The median co-expression of gene j with all other genes in tissue t,

$$\rho^{(t)}_{\text{med},j} = \text{med}\left(\rho^{(t)}_{1j}, \rho^{(t)}_{2j}, \dots, \rho^{(t)}_{(j-1)j}, \rho^{(t)}_{(j+1)j} \dots, \rho^{(t)}_{n_t j}\right)$$

   We calculate this for every gene j.

3. The median co-expression within blocks of genes. The genes are first sorted by their median expression in tissue t. In particular, for Y chromosome genes, we sort by their median expression only in male samples from tissue t. For genes specific to tissue t, we sort by their median expression in target tissue (i.e., tissue t) samples only. Once sorted, consecutive genes are then grouped into overlapping blocks of size M, shifted by one. For instance, if $M = 10$, then the first block, B_1, contains the ten genes with the lowest median expression in tissue t. The second block, B_2, contains genes 2 through 11 inclusive. The third block contains genes 3 through 12 inclusive and so on, until reaching the block with samples $n_t - 9$ through n_t inclusive. In an experiment with G genes, there are $G - M + 1$ blocks, where block b consists of genes b through $b + M - 1$.

   On a correlation matrix heatmap with genes sorted by expression level, these blocks of genes correspond with blocks of size $M \times M$ along the diagonal. We record the median co-expression of the upper triangle within these $M \times M$ blocks. Equivalently, this is the median co-expression among all discordant pairs of genes within each block.

$$\rho^{(t)}_{\text{med},B_b} = \text{med}\left(\rho^{(t)}_{b(b+1)}, \rho^{(t)}_{b(b+2)}, \dots, \rho^{(t)}_{b(b+M-1)}, \rho^{(t)}_{(b+1)(b+2)} \dots, \rho^{(t)}_{(b+M-2)(b+M-1)}\right)$$

The first measure, $\rho^{(t)}_{\text{med}}$, is relatively intuitive but lacks the granularity necessary to capture the association between spurious correlations and expression levels. If spurious correlations only arise among lowly or highly expressed genes, then the median correlation across all discordant pairs of genes may still be close to zero. For instance, in Figure 3.1c, $\rho_{\text{med}}^{(\text{skin})} = 0.0021$ among male samples. However, this median is much lower than the co-expressions observed among lowly and highly expressed genes, many of which exceed 0.5. Even among female samples in Figure 3.1d, $\rho_{\text{med}}^{(\text{skin})} = 0.2731$, which likewise does not adequately describe how co-expressions vary with expression levels.

The second and third measures, $\rho_{\text{med},j}^{(t)}$ and $\rho_{\text{med},B_b}^{(t)}$, provide relatively more granularity and allow for co-expression analysis based on expression level. As a result, these latter two measures can be visualized against gene expression to identify patterns that may be helpful for adjusting data for spurious correlations. Since blocking restricts both genes in each pairing to have similar expression levels, the third measure using blocks should more readily capture scenarios where spurious correlations arise near the diagonal of the sorted correlation matrix than the second measure.

In Figure 3.2, we present the co-expression data from Figure 3.1 for skin samples using the median co-expression associated with each gene, $\rho_{\text{med},j}^{(\text{skin})}$, and using the median co-expression within blocks, $\rho_{\text{med},B_b}^{(\text{skin})}$, for blocks of sizes $M = 5$ and $M = 10$. In each plot, we show the corresponding measure against median expression in male samples using qsmooth values on the $\log_2$ scale. In the case of blocks, we use the median expression for that block's central gene(s). This is equivalent to taking the median among the block's genes' median expressions.

As expected, blocking yields results that are more strongly associated with expression levels. In particular, blocking more aptly captures the high co-expression among genes that are lowly expressed and among genes that are highly expressed. Using the median co-expression for each gene does not capture this as well, especially among highly expressed genes. For this reason, we use the median co-expression within blocks as our measure of co-expression in our analysis. While blocks of size $M = 5$ and $M = 10$ yield similar visualizations, we use $M = 10$ due to its stronger smoothing effect.

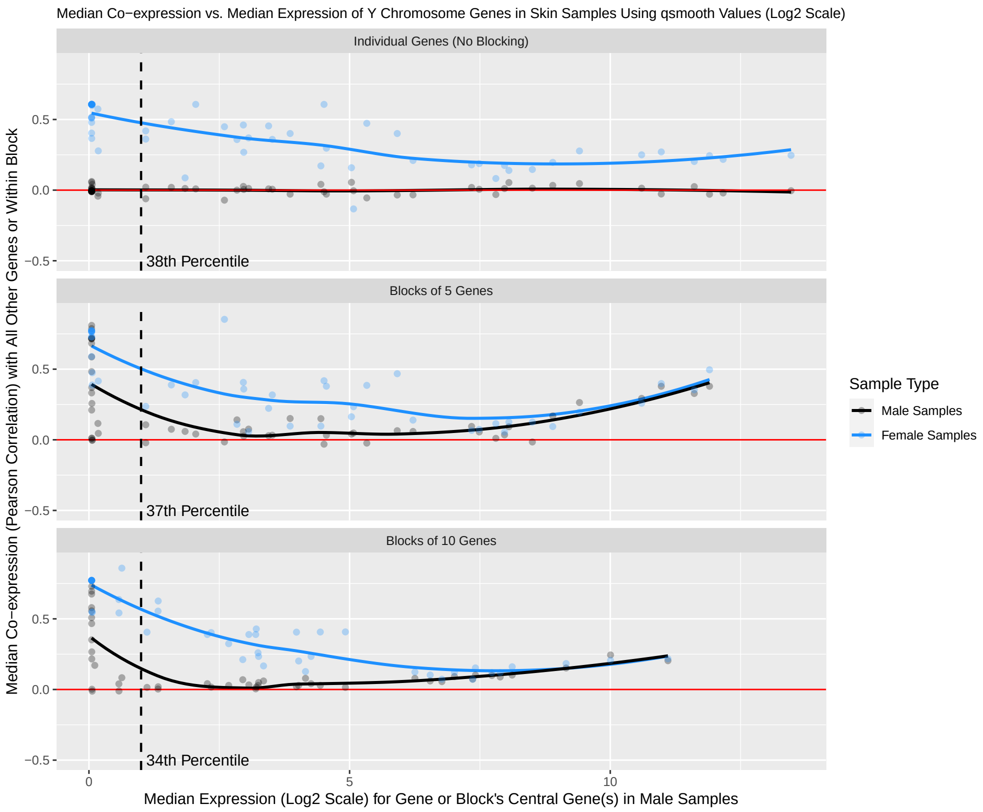


Figure 3.2: Different approaches for capturing co-expression across Y chromosome genes, using qsmooth values on the $\log_2$ scale. The top subfigure displays $\rho^{(\text{skin})}_{\text{med},j}$ for all genes $j = 1, \ldots, 56$. The middle subfigure displays $\rho^{(\text{skin})}_{\text{med},B_b}$ for blocks of size $M = 5$, yielding blocks $b = 1, \ldots, 52$. The bottom subfigure displays $\rho^{(\text{skin})}_{\text{med},B_b}$ for blocks of size $M = 10$, yielding blocks $b = 1, \ldots, 47$. Genes are sorted based on their median expression in male skin samples, but results and loess trends are otherwise stratified by sex. The dashed line and corresponding percentile indicate how many genes or blocks' central genes have a median expression less than or equal to 1 among male skin samples.

3.2.4 Statistical Tests

In addition to properly capturing co-expression patterns in RNA-seq data, it is often necessary to perform statistical tests to determine the significance of these co-expressions. In particular, spurious correlations that seem to arise could simply be due to random chance or noise, especially in cases where correlations are unlikely but biologically possible, as with tissue-specific genes in non-target tissues. In these cases, we need to use a statistical test to determine how likely the apparent co-expression could be generated due to noise alone when no co-expression is truly present.

Consider that the three co-expression measures discussed in Section 3.2.3 all use correlation. While one can use a t-test for Pearson correlation when data are bivariate normal, that normality assumption will often not be appropriate for RNA-seq expression values. In particular, raw expression values will often be zero-inflated, which will likewise impact normalized values, such as those from qsmooth. Additionally, we are generally interested in detecting significant correlations in groups for which the gene should be lowly or not expressed (e.g., females for Y chromosome genes, tissues for genes that are specific to a different tissue, etc.). In these samples, we reasonably expect expression values to be relatively small counts and to have relatively narrow ranges. Thus, instead of relying on an approximate test with inappropriate normality assumptions, we use permutation tests to exactly assess the significance of a given co-expression measure. While we use Pearson correlation, the same approaches discussed here are appropriate for performing exact tests of co-expression measures that use Spearman's rank correlation instead.

We perform a permutation test in the following steps:

1. Identify the genes of interest (e.g., Y chromosome genes, tissue-specific genes for a particular tissue) and the samples of interest (e.g., male, female, target tissue, non-target tissues).

2. Subset the original experiment to the genes and samples of interest.

3. Sort the genes by their median expression in samples for which expression and co-expression values are most biologically meaningful (e.g., male samples for Y chromosome genes, target tissue samples for tissue-specific genes). These samples may be distinct from those in the subset from Step 2.

4. Record the observed measure(s) of choice for co-expression in the subset: $\rho^{(t)}_{\text{med}}$ for the entire subset, $\rho^{(t)}_{\text{med},j}$ for each gene j, or $\rho^{(t)}_{\text{med},B_b}$ for each block b. Denote the set of observed test

statistics as $\mathcal{S}^* = \{S_1^*, S_2^*, \dots\}$. This set is of length 1 if using $\rho_{\text{med}}^{(t)}$, length G if using $\rho_{\text{med},j}^{(t)}$, or length $G - M + 1$ if using $\rho_{\text{med},B_b}^{(t)}$.

5. Perform L permutations. For each permutation, ℓ:

   (a) Permute the expression values within-gene for all genes in the subset. Note that the ordering of genes based on median expression does not change from the observed data.

   (b) Record the corresponding co-expression measure from Step 4 for the permuted data: $\rho_{\text{med}}^{(t)}$ for the entire permuted subset, $\rho_{\text{med},j}^{(t)}$ for each gene j, or $\rho_{\text{med},B_b}^{(t)}$ for each block b. Denote the set of the test statistics from permutation ℓ as $\mathcal{S}_\ell = \{S_{\ell 1}, S_{\ell 2}, \dots\}$.

6. Under the null hypothesis that there is no co-expression between genes, the expected value of S_i^* is zero. Therefore, for each observed test statistic, calculate an empirical p-value as the proportion of corresponding permuted test statistics that are at least as far from zero in absolute value as the observed statistic:

$$p_i = \frac{\sum_{\ell=1}^{L} \mathbb{I}(|S_{\ell i}| \geq |S_i^*|)}{L}$$

For a small number of genes, G, it may be feasible to perform every possible permutation. However, in most cases, this will not be computationally practical. Instead, we randomly select L permutations to perform. If an investigator is concerned about controlling the probability of a Type 1 Error across multiple test statistics, then they may control the family-wise error rate (FWER) with an approach like the Bonferroni correction, or control the false discovery rate (FDR) with an approach like the Benjamini-Hochberg method [15].

In some cases, many of the observed test statistics may be highly significant. That is, if the number of permutations, L, is not large enough or if the observed results are highly extreme, then many p-values may be zero. In these cases, if an investigator still wants to assess the relative extremity of results across different genes or blocks, then they may calculate a t-statistic for each observed test statistic in $\mathcal{S}^*$ based on the permuted results as:

$$\text{t-stat}_i = \frac{S_i^* - 0}{\hat{\sigma}_i} = \frac{S_i^*}{\hat{\sigma}_i}$$

where $\hat{\sigma}_i$ is the estimated standard deviation of S_i^* under the null, based on the permuted results:

$$\hat{\sigma}_i = \sqrt{\frac{1}{L}\sum_{\ell=1}^{L}(S_{\ell i} - 0)^2} = \sqrt{\frac{1}{L}\sum_{\ell=1}^{L}S_{\ell i}^2}$$

3.3 Results

We focused on analyzing gene co-expression in nine tissues: kidney cortex, minor salivary gland, spleen, liver, pancreas, stomach, thyroid, lung, and skin. These tissues represent a wide range of different sample sizes and all have tissue-specific genes associated with them. The sample sizes and number of tissue-specific genes for each tissue are displayed in Table 3.1. Additionally, we use 56 genes from the Y chromosome in the analysis.

3.3.1 Y Chromosome Genes

We show the median co-expression of Y chromosome genes in samples from each tissue using qsmooth expression values on the $\log_2$ scale in Figure 3.3. For this, we use blocks of 10 genes, which corresponds with the third measure, $\rho^{(t)}_{\text{med},B_b}$, in Section 3.2.3.

While the exact associations between co-expression and expression vary by tissue, there are several common traits. For each tissue, the highest median co-expression within blocks occurs for lowly expressed genes, often reaching values between 0.5 and 1 for females and around 0.25 for males. As expression levels increase, the median co-expression within blocks initially decreases, but then

Table 3.1: Sample sizes for each tissue regarding the number of male samples, female samples, samples from that tissue, samples from all remaining tissues, and the number of genes specific to that tissue with multiplicity equal to one for the subset of tissues analyzed. Additionally, there are 56 genes specific to the Y chromosome in our analysis.

Tissue	Male Sample Size	Female Sample Size	Target Tissue Sample Size	Non-target Tissue Sample Size	Number of Tissue-specific Genes
Kidney Cortex	28	8	36	2098	215
Minor Salivary Gland	46	24	70	2064	131
Spleen	69	49	118	2016	683
Liver	92	45	137	1997	600
Pancreas	115	78	193	1941	257
Stomach	117	87	204	1930	103
Thyroid	224	131	355	1779	183
Lung	238	122	360	1774	109
Skin	428	233	661	1473	428

increases back up as the blocks include the most highly expressed genes. For all tissues except for the minor salivary gland, female samples exhibit co-expressions levels as high or higher than those for male samples across all expression levels. In the minor salivary gland, the median co-expression within blocks for females dips slightly below that for males for highly expressed genes.

The most distinctive behavior occurs in kidney cortex samples. Here, the median co-expression within blocks for females starts very high, with values near 1, and steadily decreases until plateauing around 0.5 for highly expressed genes. This contrasts with the generally parabolic trend seen in other tissues. Additionally, kidney cortex samples exhibit the largest difference in co-expression levels between female and male samples. This is likely attributable the very small female sample size for the kidney cortex.

By only analyzing qsmooth expression values, it is difficult to ascertain whether these co-expression patterns are attributable to technical artifacts in the data, the normalization process, or both. Therefore, we display the same results as shown in Figure 3.3, but for the raw expression data on the $\log_2$ scale, in Figure 3.4. These raw data were subject to the same preprocessing steps in Section 3.2.2,

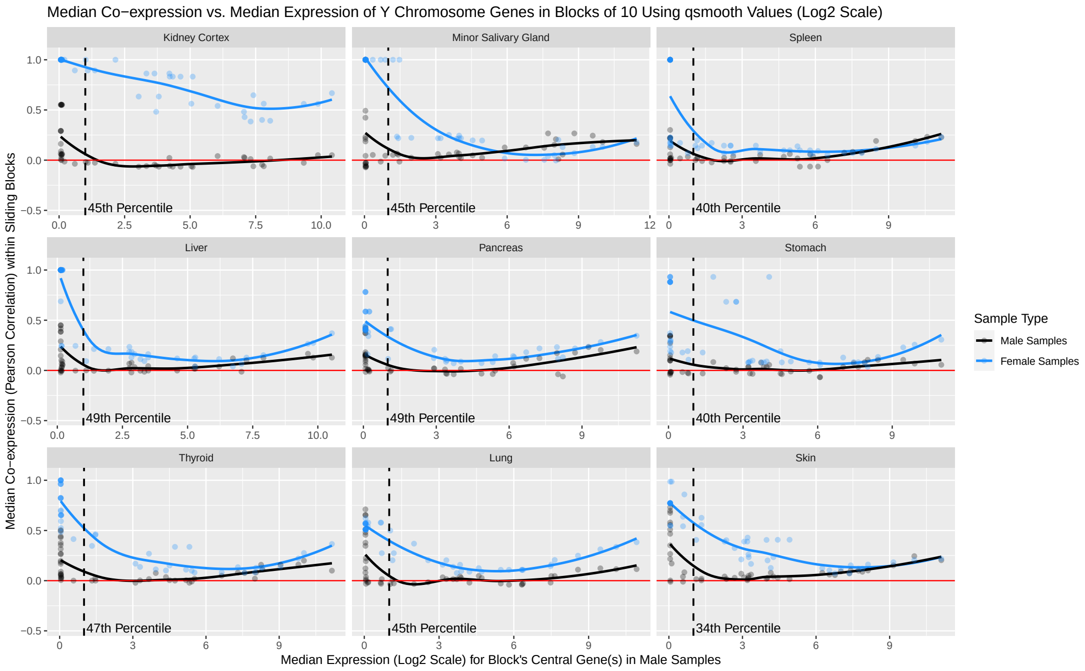


Figure 3.3: $\rho^{(t)}_{\text{med},B_b}$ across Y chromosome genes for blocks of size $M = 10$, using qsmooth expression values on the $\log_2$ scale. Genes are sorted based on their median expression in male samples for the given tissue, but results and loess trends are otherwise stratified by sex. The dashed line and corresponding percentile indicate how many blocks' central genes have a median expression less than or equal to 1 for that tissue among male samples.

except that they were not normalized using qsmooth. For these raw data, some genes had constant expression across all samples of interest. For example, 14 genes on the Y chromosome had zero expression for all female skin samples. These constantly expressed genes induce NA co-expression values because the standard deviations of their raw expression levels are zero. Therefore, we exclude such constantly expressed genes from the co-expression analysis and note the number of constantly expressed genes for each sex in each tissue in Figure 3.4.

We note several large discrepancies for these raw data results relative to the qsmooth results. Using raw expression, lowly expressed genes exhibit co-expression levels that are much closer to zero. This holds for both sexes in most tissues, with the main exceptions being female kidney cortex, minor salivary gland, and liver samples, which still exhibit moderate to high co-expression among lowly expressed genes. This is likely driven, in part, by their small sample sizes. For female samples in general, co-expression levels tend to be near zero except for among highly expressed genes. Additionally, among genes that are not lowly expressed, male samples now exhibit higher co-expression levels than female samples in every tissue. This suggests that qsmooth, for which females exhibit higher co-expression levels than males in Figure 3.3, effectively reverses the association between co-expression and sex for Y chromosome genes in the raw data. Lastly, for males, the association between co-expression and expression levels when using raw data is increasing and nearly linear in each tissue, in contrast to the more parabolic association that appears in the qsmooth data.

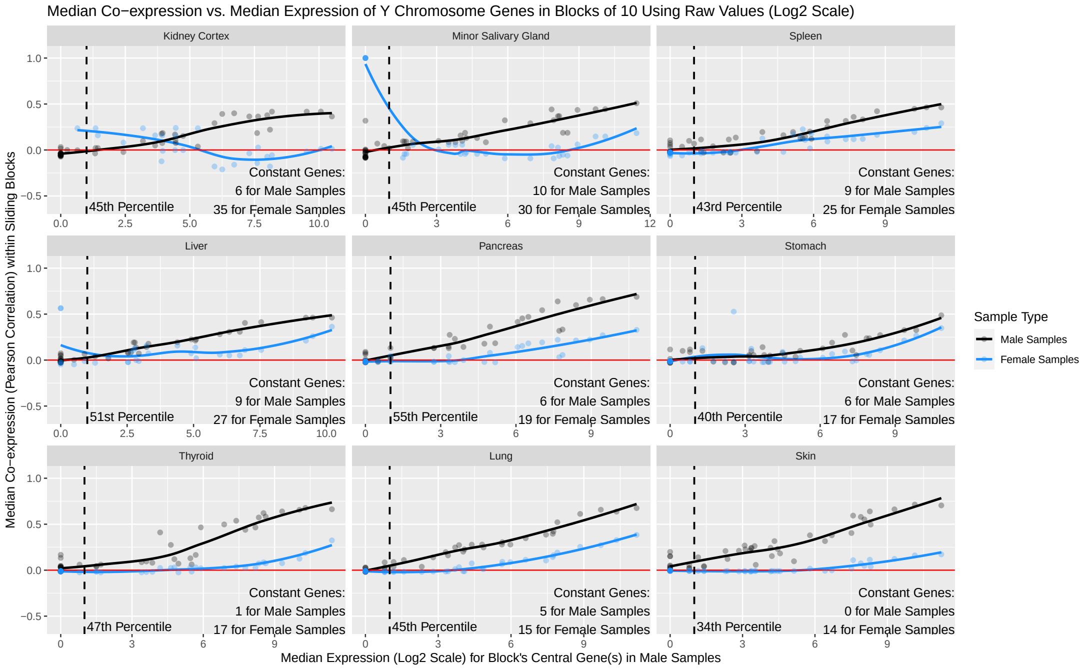


Figure 3.4: $\rho^{(t)}_{\text{med},B_b}$ across Y chromosome genes for blocks of size $M = 10$, using raw expression values on the $\log_2$ scale. Genes are sorted based on their median expression in male samples for the given tissue, but results and loess trends are otherwise stratified by sex. The dashed line and corresponding percentile indicate how many blocks' central genes have a median expression less than or equal to 1 for that tissue among male samples. The number of genes with raw expressions values of zero for all samples in the pertinent tissue/sex stratum are noted. The resulting NA correlations that they introduce are excluded from analysis.

In addition to visualizing patterns, we can also assess the significance of these co-expressions using formal statistical tests. For Y chromosome genes, it is not vital to assess the significance of co-expressions in female because any co-expression is known to be a false positive due to the underlying biology. However, it may still be helpful to use statistical tests to assess the relative extremity of co-expressions in the female samples compared to male samples. In Figure 3.5, we present the results from the permutation tests for median co-expressions within blocks from Section 3.2.4, using 250 permutations and the qsmooth data.

We find results that are largely consistent with those noted previously. Virtually all median co-expressions within blocks for female samples have empirical p-values equal to or near zero, with the primary exception occurring in the minor salivary gland. In contrast, the empirical p-values for males tend to be near zero only for highly expressed genes.

The corresponding t-statistics calculated from the permutation tests reinforce these findings. The t-statistics for median co-expressions in female samples are on par with, or more extreme than, the t-statistics for male samples across most tissues and most expression levels. The primary exceptions occur in the minor salivary gland, spleen, and skin where the t-statistics for male samples are slightly more extreme for highly expressed genes. Additionally, the t-statistics exhibit a similar parabolic trend as seen for median co-expressions within blocks in Figure 3.3. That is, relatively extreme t-statistics tend to occur for lowly and highly expressed genes, with a dip in the middle.

We do not perform a correction to control the Type 1 Error rate here because many, if not all, p-values are exactly zero in every tissue. In this case, a correction will generally preserve these p-values as zero and not meaningfully change the results. We display the percentage of empirical p-values equal to zero for each tissue for the Y chromosome genes in Table 3.2.

Table 3.2: The percentage of empirical p-values that are equal to zero for the permutation tests of median co-expressions within blocks of 10 Y chromosome genes, using $\log_2$ qsmooth expression values.

Tissue	Male	Female
Kidney Cortex	21.3	97.9
Minor Salivary Gland	40.4	80.9
Spleen	34.0	83.0
Liver	46.8	93.6
Pancreas	53.2	100
Stomach	44.7	93.6
Thyroid	59.6	100
Lung	55.3	100
Skin	72.3	100

3.3.2 Tissue-specific Genes

We now compare the associations between co-expression and expression for tissue-specific genes when using both qsmooth and raw data. We show the median co-expression of tissue-specific genes in blocks of 10 when using qsmooth expression values in Figure 3.6a and when using raw expression values in Figure 3.6b, both on the $\log_2$ scale. In the presence of no spurious correlations, we expect that the co-expression values among non-target tissues should be relatively close to zero across all expression levels. However, we do not find this in practice for all tissues.

Genes specific to the kidney cortex, liver, stomach, thyroid, and lung all appear to have relatively weak signals of spurious correlations when using qsmooth data. Their co-expression levels among non-target tissues stay near zero across all expression levels. However, minor salivary gland, spleen, pancreas, and skin all have tissue-specific genes whose co-expression levels suggest the presence of spurious correlations. In particular, spleen is the only tissue whose genes tend to have higher co-expression among non-target tissue samples than among target samples themselves.

In contrast, when using raw data, every set of tissue-specific genes exhibits at least some moderately strong co-expressions in non-target tissues. This suggests that for some tissues, qsmooth normalization may help mitigate spurious correlations for tissue-specific genes. However, this normalization may also mitigate correlations within target tissues as well. For example, the qsmooth data appear to weaken the co-expressions of spleen-specific genes in spleen samples themselves, mostly among highly expressed genes. The same behavior occurs in thyroid-specific genes, which likewise have weaker co-expressions in thyroid samples when using qsmooth data compared to raw data.

We display the empirical p-values from the permutation tests of tissue-specific genes within blocks in Figure 3.7a. Every tissue has a majority of p-values equal to or near zero for both target and non-target samples, suggesting the presence of many significant spurious correlations for every set of tissue-specific genes. However, certain sets of tissue-specific genes, particularly those for the spleen and skin, more consistently yield extremely low p-values for non-target samples. The t-statistics in Figure 3.7b reinforce this finding. The t-statistics are generally extreme for both target and non-target samples for every set of genes. However, the t-statistics for blocks of spleen- and skin-specific genes are relatively extreme in non-target samples compared to those for other tissues.

As with the results for Y chromosome genes in Section 3.3.1, we do not perform a correction for the p-values reported here because many, if not all, p-values are exactly zero in every tissue. We display the percentage of empirical p-values equal to zero for each set of tissue-specific genes in Table 3.3.

3.4 Discussion and Conclusion

Researchers commonly use RNA-seq data to study co-expression among genes, but these data may contain spurious correlations in addition to true co-expressions of biological interest. It is important for investigators to be cognizant of these spurious correlations, to identify them, and to adjust their data as necessary. We have explored the presence of these spurious correlations in the 6.0 release of the GTEx data set for two different types of genes: those on the Y chromosome and those specific

Table 3.3: The percentage of empirical p-values that are equal to zero for the permutation tests corresponding with blocks of 10 tissue-specific genes, when using $\log_2$ qsmooth expression values.

Tissue	Target Tissue	Other Tissues
Kidney Cortex	77.2	70.4
Minor Salivary Gland	78.7	83.6
Spleen	57.0	98.5
Liver	99.2	84.6
Pancreas	85.5	91.5
Stomach	100	83.0
Thyroid	94.8	73.6
Lung	65.0	68.0
Skin	93.3	100

to a particular tissue. We have done this both for raw expression values and values which have been normalized using qsmooth, a common normalization procedure for heterogeneous data sets.

As part of the identification and characterization of these spurious correlations, an investigator must both choose a measure of co-expression among genes and a hypothesis test to assess the significance of that co-expression. We have presented three different measures of co-expression. We advocated for using the median co-expression in blocks of genes when sorted by their median expression because spurious correlations appear to be more prominent among genes of comparable expression levels. To assess the significance of these measures, we proposed permutation tests that can yield both empirical p-values and t-statistics, which may be useful for comparing results when many empirical p-values are exactly zero.

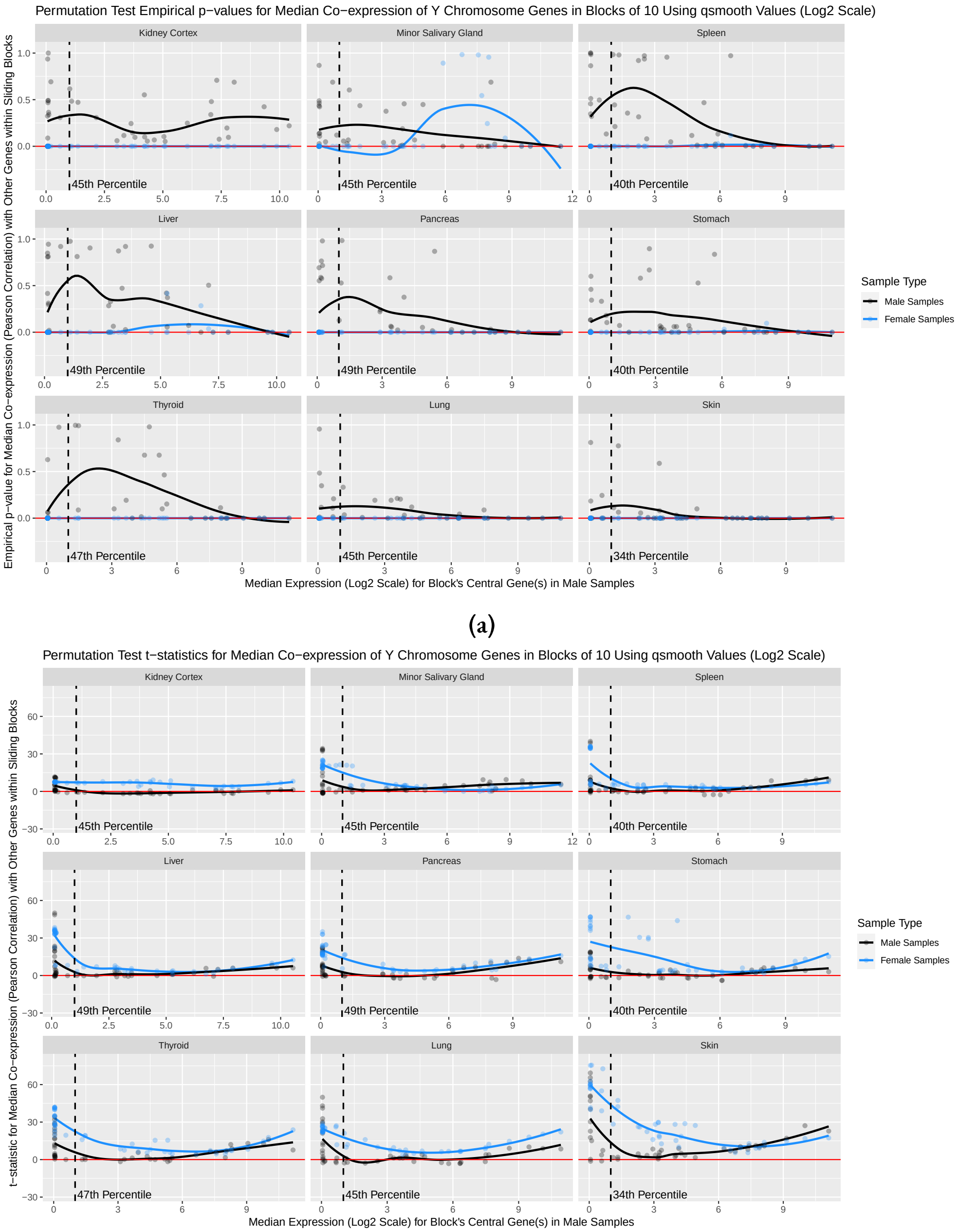
We first focused on analyzing the co-expression of Y chromosome genes because these genes should objectively have no co-expression in female samples. It is possible for technical issues associated with sequencing, such as mismapping, to induce these spurious correlations. If these spurious correlations are weak and rare in a given experiment, then they may be of relatively little concern. However, across genes on the Y chromosome, we find clear evidence of statistically significant correlations in female samples and that this result is largely consistent across various tissues. Often, these co-expression measures are stronger and more significant than those for male samples, for which these co-expression measures should capture true signal. Additionally, it appears that qsmooth normalization exacerbates these spurious correlations relative to the raw data. Thus, these spurious correlations may not only arise due to technical issues with sequencing procedures, but also because of normalization procedures that take place during analysis.

We also analyzed tissue-specific genes, which should co-express in their target tissue but less so in other tissues. Results vary more by tissue for these genes than for Y chromosome genes. This suggests that both the genes and tissues analyzed in an experiment are important for determining the presence of spurious correlations. Using permutation tests, we find that every set of tissue-specific genes has highly significant spurious correlations. While some of these correlations are relatively weak in signal, they are weak across thousands of non-target samples, making them extreme compared to what would be likely under the null in which no spurious correlations exist. Additionally, the impact of qsmooth normalization on the spurious correlations appears to differ by tissue. qsmooth normalization slightly exacerbates the spurious correlations in spleen-specific genes but mitigates the presence of spurious correlations in genes specific to the other tissues. Despite this, many of these spurious correlations for tissue-specific genes are still highly significant and indicate that the data require further adjustment.

It is worth noting that a limitation of analyzing tissue-specific genes is how a researcher defines them. While we use an objective and established definition [77], other investigators may use different definitions [37]. Therefore, it is possible that "tissue-specific" genes could have true co-expressions among tissue to which they are not specific, depending on one's definition. Considering this, genes on the Y chromosome provide a more objective setting in which spurious correlations can be assessed.

Our analysis suggests that investigators need to check their data for spurious correlations and adjust their data as necessary. While we do not present methods for adjusting data in detail, it is important to note that approaches will typically be context-dependent. When non-zero expression values are definitively known to be spurious, such as those for Y chromosome genes in female samples, an investigator could consider a simple approach like eliminating the corresponding read counts. However, this same approach would not necessarily be appropriate when handling tissue-specific genes which, depending on their definition, could truly express in samples other than their target tissue. In such cases, an investigator may need to use a more sophisticated approach, such as Count Adjustment to Improve the Modeling of Association-based Networks (CAIMAN) [37], which uses an expectation-maximization (EM) algorithm [22] to distinguish expressed and non-expressed genes within each tissue to determine which counts to eliminate.

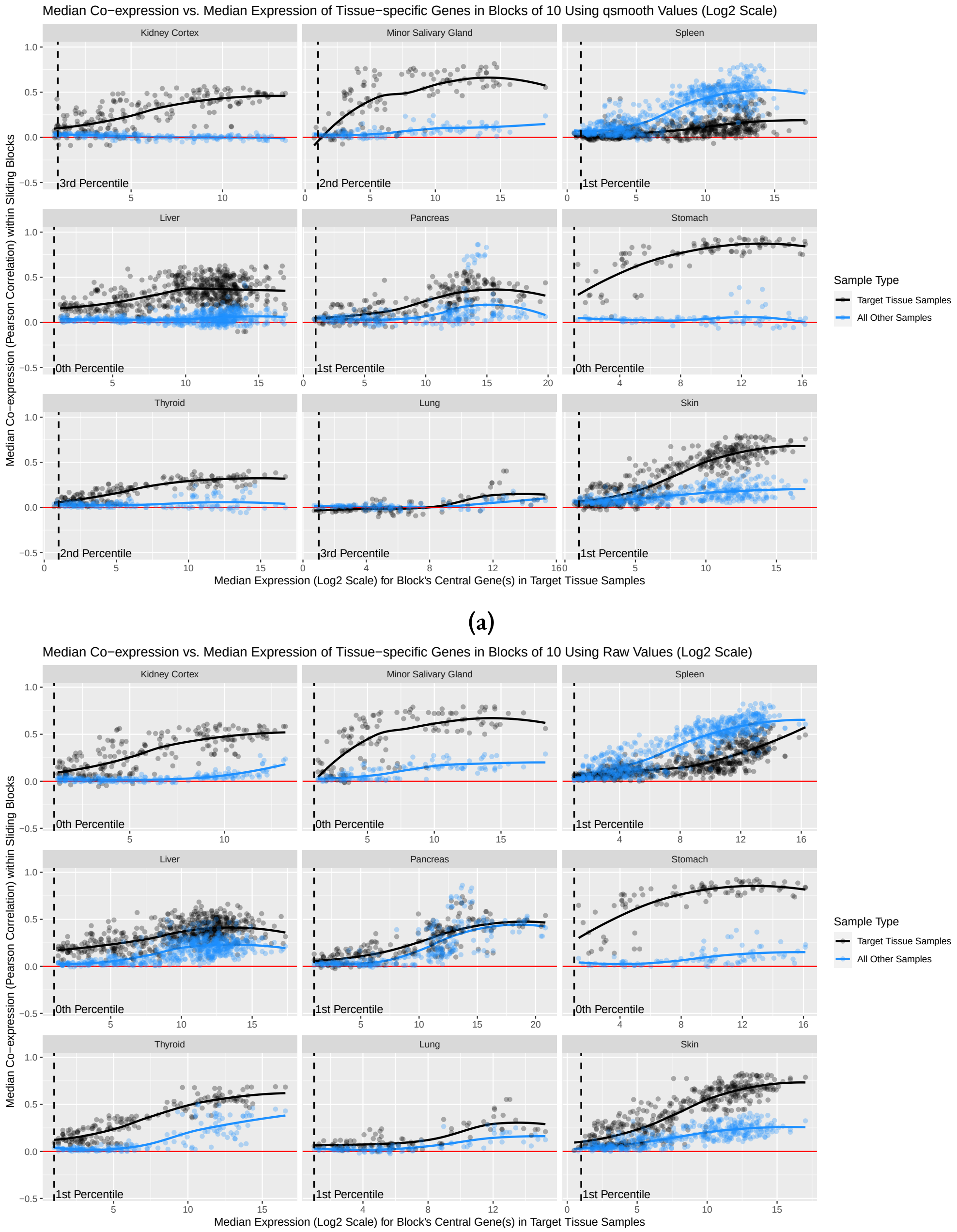
Our findings strongly suggest that spurious correlations arise for various sets of genes in different tissues. Investigators need to utilize appropriate measures and statistical tests to assess the presence of these spurious correlations. These steps are vital to avoiding false positives in any co-expression analysis. Additionally, researchers must give careful consideration to any normalization procedure they use, as it may either mitigate or exacerbate spurious correlations depending on the context.



(a)

(b)

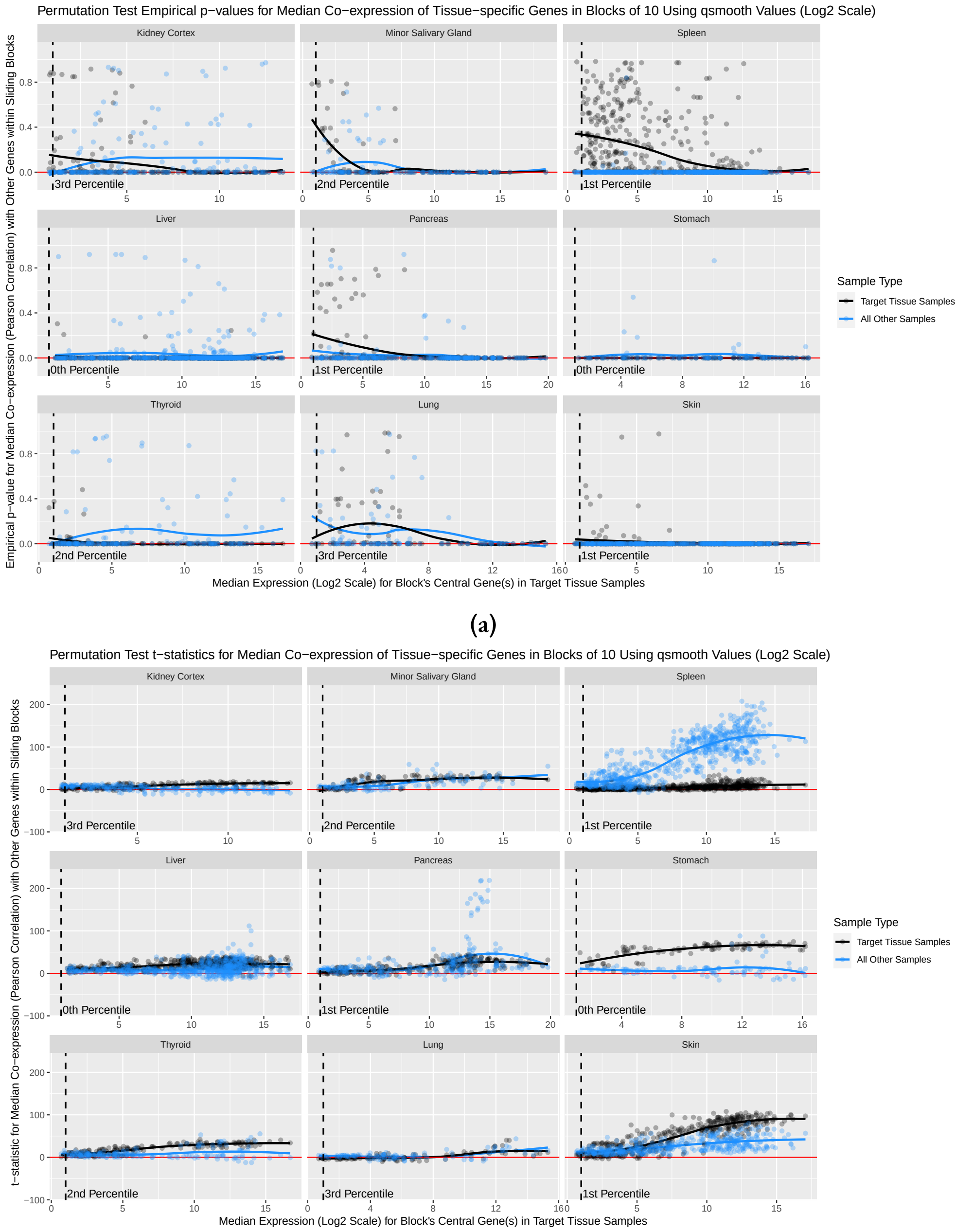
Figure 3.5: Permutation test results for $\rho^{(t)}_{\text{med},B_b}$ across Y chromosome genes for blocks of size $M = 10$, using qsmooth expression values on the $\log_2$ scale. Results include (a) empirical p-values and (b) t-statistics. Genes are sorted based on their median expression in male samples for the given tissue, but results and loess trends are otherwise stratified by sex. The dashed line and corresponding percentile indicate how many blocks' central genes have a median expression less than or equal to 1 for that tissue among male samples.



(a)

(b)

Figure 3.6: $\rho^{(t)}_{\text{med},B_b}$ across tissue-specific genes for blocks of size $M = 10$, using (a) qsmooth expression values and (b) raw expression values, both on the $\log_2$ scale. Genes are sorted based on their median expression in the target tissue samples, but results and loess trends are otherwise stratified by target tissue status. The dashed line and corresponding percentile indicate how many blocks' central genes have a median expression less than or equal to 1 in the target tissue.



(a)

(b)

Figure 3.7: Permutation test results for $\rho^{(t)}_{\text{med},B_b}$ across tissue-specific genes for blocks of size $M = 10$, using qsmooth expression values on the $\log_2$ scale. Results include (a) empirical p-values and (b) t-statistics. Genes are sorted based on their median expression in the target tissue samples, but results and loess trends are otherwise stratified by target tissue status. The dashed line and corresponding percentile indicate how many blocks' central genes have a median expression less than or equal to 1 in the target tissue.

A

Supplemental Material for Automated Selection of Changepoints using Empirical P-values and Trimming (ASCEPT)

A.1 Review of Offline Changepoint Detection

Researchers have created various methods for offline changepoint detection over the years. We briefly review the most relevant approaches here. In addition, [80] contains a more extensive review.

In our application for identifying changepoints in mHealth data, we are concerned with performing offline changepoint detection for an unknown number of changepoints that primarily reflect mean-shifts in a time series. This is a common scenario for changepoint analysis and appears reasonable for mobile health research in particular.

In offline changepoint detection, the goal is typically to perform an optimization. In many cases, one will make a parametric assumption about the data, such as assuming normality. Additionally, all observations between two changepoints, which form a "segment," will be assumed to follow the same distribution, while those in segments separated by changepoints may follow different distributions, such as normal distributions with different means. Many detection algorithms identify changepoints by minimizing a cost function (e.g., negative log-likelihood) subject to a penalty for introducing additional changepoints to prevent overfitting. There are methods that provide approximate, or locally optimal, results as well as those that provide exact, or globally optimal, results. While approximate methods do not guarantee a globally optimal result, they typically offer lower computational complexities.

One of the most popular approximate methods is binary segmentation. Binary segmentation effectively considers splitting a time series of observations, $y_1, \dots, y_T$ for times $t = 1, \dots, T$, into two subsegments by identifying a changepoint at time τ. To do this, the method first defines a cost function, $\mathcal{C}(\cdot)$, and sets $\tau = \text{argmin}_{t \in \{1,\dots,T\}}[\mathcal{C}(y_1, \dots, y_t) + \mathcal{C}(y_{t+1}, \dots, y_T)]$. Here, the cost function may be something like the negative log-likelihood, if assuming a parametric model. If one wishes to detect multiple changepoints, then one can run this minimization again on each subsegment, one from $t = 1$ to $t = \tau$ and the other from $t = \tau + 1$ to $t = T$. This process repeats until some stopping criterion is met. The primary advantage of this approach is its relatively low computational complexity of $\mathcal{O}(n \log n)$ when considering a series of n observations [45].

Other approximate approaches have built off of binary segmentation. These include Circular Binary Segmentation (CBS) [61], which allows for detection of two changepoints at a time, and Wild Binary Segmentation (WBS) [25], which randomly draws and checks segments. Though CBS is approximate, ASCEPT uses similar principles. For instance, CBS generates empirical p-values to iteratively assess potential changepoints, retaining those found to be significant. It then prunes or trims, the set of significant changepoints to remove those within linear trends. ASCEPT follows comparable principles but uses different implementations at each step.

There is also a number of exact methods for multiple changepoint detection. However, these generally suffer from relatively high computational complexities compared to approximate methods.

For instance, the Segment Neighborhood method [13] has $\mathcal{O}(mn^2)$ complexity for a time series of length n with m changepoints. Likewise, the Optimal Partitioning algorithm has $\mathcal{O}(n^2)$ computational complexity [39]. The method that we consider to be the state-of-the-art is Pruned Exact Linear Time (PELT), a modified version of the Optimal Partitioning algorithm that is capable of running in $\mathcal{O}(n)$ time under certain assumptions [45]. Consider detecting m changepoints, $\tau_1, \dots, \tau_m$, with $1 \leq \tau_1 < \dots < \tau_m \leq n-1$. We define $\tau_0 = 0, \tau_{m+1} = n$ for the purpose of segmenting all of the data. For a cost function, $\mathcal{C}(\cdot)$, PELT performs the minimization:

$$\min_{m,\tau_1,\dots,\tau_m} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{\tau_{i-1}+1}, \dots, y_{\tau_i})\right] + \beta f(m) \tag{A.1}$$

where $f(m)$ is a penalty based on the number of changepoints and β is a multiplier on the penalty. PELT is often used with a penalty that is linear in the number of changepoints, $\beta f(m) = \beta m$. Under this condition, we can equivalently write Equation 1 as:

$$\min_{m,\tau_1,\dots,\tau_m} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{\tau_{i-1}+1}, \dots, y_{\tau_i}) + \beta\right] \tag{A.2}$$

PELT solves this optimization problem using dynamic programming in a similar manner to Optimal Partitioning [39], but is able to obtain its considerable speed-up by pruning the space over which it searches for changepoints. Namely, consider the scenario where the cost function is defined to be the negative log-likelihood associated with a segment. Likewise consider indices t and s where $t < s < T$, letting $\mathcal{T}_t$ denote the set of possible changepoints to be detected over indices 1,...,t and likewise for $\mathcal{T}_s$. In the case where:

$$\left[\min_{m,\mathcal{T}_t} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{\tau_{i-1}+1}, \dots, y_{\tau_i}) + \beta\right]\right] + \mathcal{C}(y_t, \dots, y_s) \geq \left[\min_{m,\mathcal{T}_s} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{\tau_{i-1}+1}, \dots, y_{\tau_i}) + \beta\right]\right] \tag{A.3}$$

t cannot be the last optimal changepoint prior to T [45]. Under certain regularity conditions, notably that the expected number of changepoints increases linearly with n, this approach can achieve a complexity of $\mathcal{O}(n)$. In the worst case, PELT has the same computational complexity as Optimal Partitioning, $\mathcal{O}(n^2)$.

The main difficulty with using PELT is the specification of the penalty constant, β. Selecting β is often non-intuitive. To help with this, the Changepoints for a Range of PenaltieS (CROPS) algorithm offers an efficient approach for running PELT under many different values of β. In particular, CROPS identifies all of the different sets of changepoints detected as one varies β between a chosen β_{min} and β_{max} [34]. CROPS takes advantage of the fact that many different penalty constants will yield the same results under PELT. For instance, if a chosen β yields the set of changepoints T, then increasing or decreasing β by a small amount will often not lead to PELT detecting fewer or more changepoints. Using CROPS, one needs to run PELT a maximum of $m(\beta_{\text{min}}) - m(\beta_{\text{max}}) + 2$ times where $m(\beta)$ refers to the number of changepoints detected under penalty constant β.

Running CROPS on PELT allows an investigator to explore the results from PELT under many different penalties. However, this approach still suffers from some practical challenges. For example, CROPS gives an investigator the results of many runs of PELT but does not provide any indication as to which set of changepoints is the "best" set among those runs. The investigator has to manually determine which set is the most appropriate for their data. Thus, we need an approach for selecting an optimal set among those presented by CROPS. This is especially difficult to formalize when investigating multiple time series, such as what we encountered in our analysis of mHealth data from the Precision VISSTA study. There is clearly a need for a rigorous approach for selecting a final set of changepoints in this context. This is the primary motivation for ASCEPT.

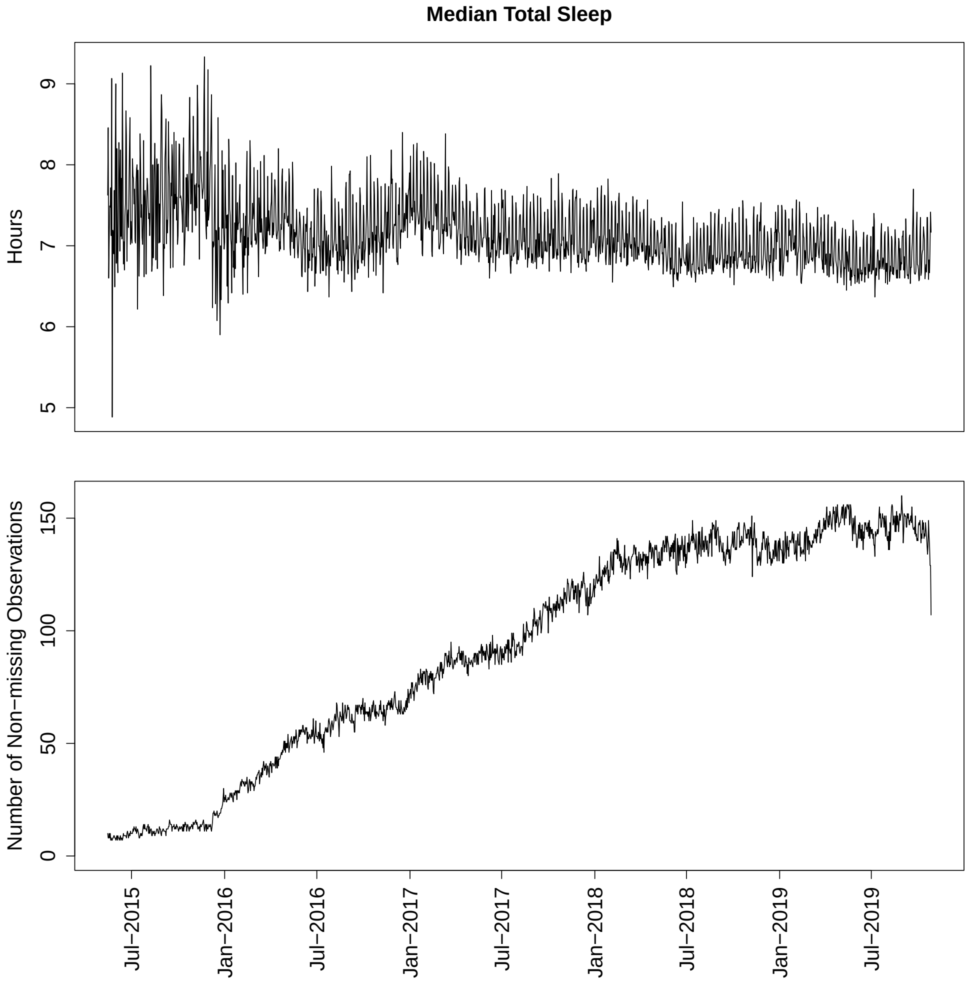


Figure A.1: The daily median total sleep from the Precision VISSTA study and the corresponding number of contributing observations each day.

A.2 Additional Results for Various Trimming Thresholds

In the main manuscript, we present the results of ASCEPT when using a trimming threshold of 1.2. However, it is important to note that our specific results depended on this selected threshold value. We investigated which changepoints ASCEPT retained or trimmed when varying the trimming threshold for the simulated time series data (Supplementary Figure A.2). We found that any trimming threshold between 1.13 and 1.20 inclusive yielded the same final set of changepoints while a trimming threshold greater than 1.20 trimmed out the changepoints from Stage 1 of ASCEPT at indices 699 and 700, thereby introducing false negatives. Decreasing the trimming threshold below 1.13 resulted in ASCEPT retaining multiple changepoints initially detected within the seasonal pattern between indices 401 and 600 inclusive, thereby introducing nuisance changepoints. Overall, this analysis shows that, while results are fairly robust across multiple trimming thresholds, it is important to choose an appropriate value in order to avoid either removing or retaining too many changepoints.

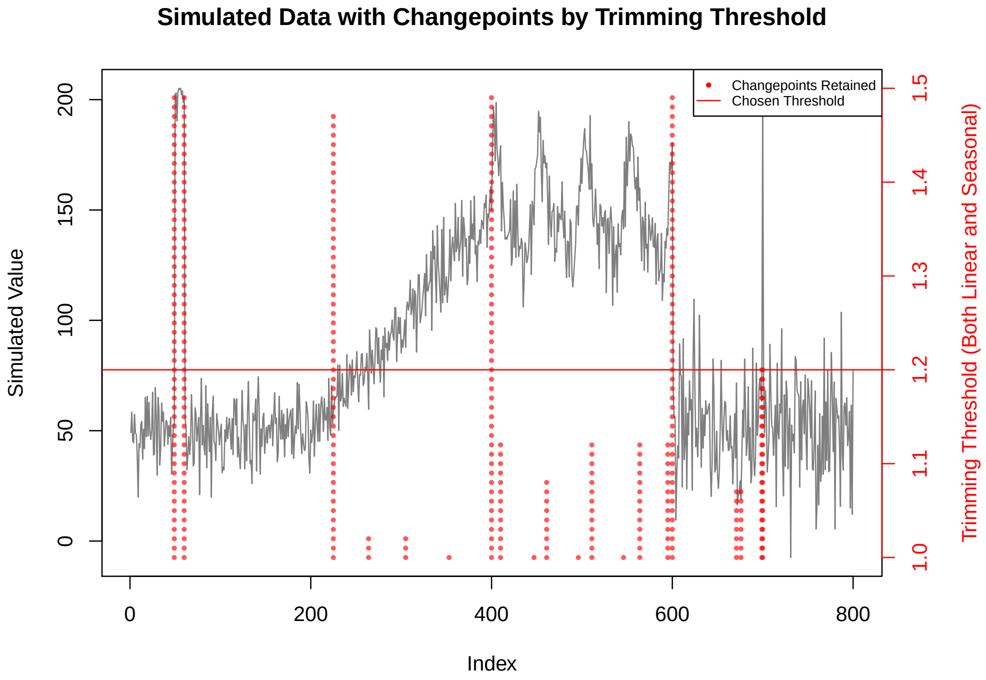


Figure A.2: The simulated time series with ASCEPT changepoints initially detected, using a 0.01 significance level and 10,000 Monte Carlo simulations, trimmed at various thresholds. All changepoints are retained at a threshold of 1, and all are removed by a threshold of 1.5. Thresholds between 1.13 and 1.2 inclusive all yield the same results as a threshold of 1.2, as used in the main manuscript.

A.3 Additional Precision VISSTA Results for ASCEPT and CBS

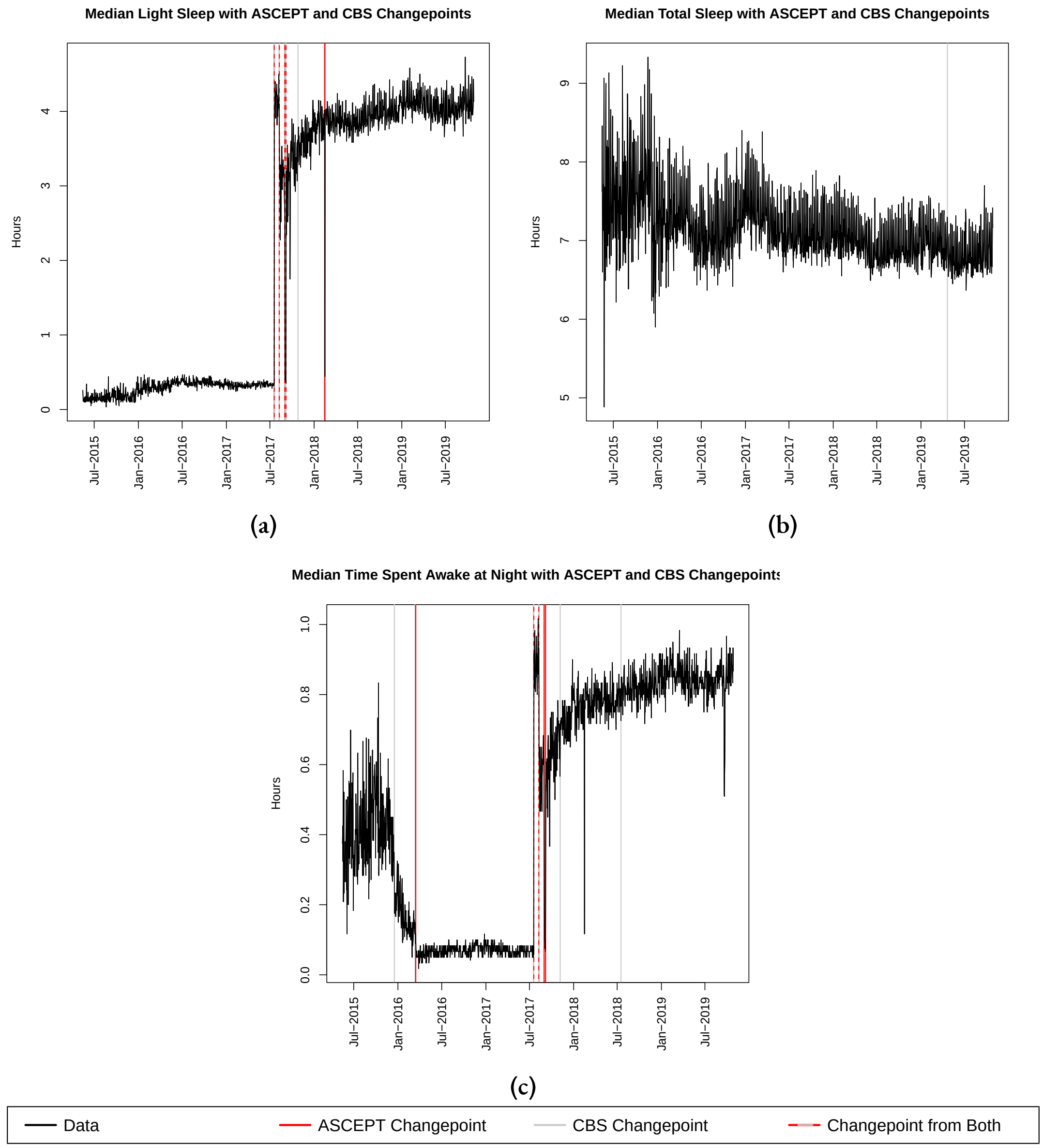
In Supplementary Figures A.3 and A.4, we present the results for both ASCEPT and CBS on different variables from the Precision VISSTA study, excluding those in Figure 1.5 of the main text. Across the different variables, we found that ASCEPT generally outperformed CBS at identifying mean-shifts in the data, especially those lasting only one day, and at trimming changepoints within linear and seasonal trends.

While ASCEPT performed well when applied to these various time series, we identified one exception when investigating the awake variable, depicted in Supplementary Figure A.3c. Here, both ASCEPT and CBS missed four relevant changepoints. In the case of ASCEPT, reducing the trimming threshold to 1.15 resulted in the method capturing two of these changepoints. Interestingly, the behavior of this variable was nearly identical to the times woken variable, on which ASCEPT performed well (see Figure 1.5c in the main text). This indicates that small changes in a series can sometimes yield fairly different results in the final set of identified changepoints. We note that changing the trimming threshold to 1.15 also introduced several nuisance changepoints in the series of times woken, emphasizing the importance of considering multiple trimming thresholds.

A.4 Additional Results for Segment Correction

The main text's segment correction analysis used a fitting threshold of 1.75. A linear or harmonic regression was deemed the best fit to a segment only if the ratio of the constant fit's RMSE to the best corresponding linear regression or harmonic regression's RMSE was greater than this fitting threshold. Supplementary Figures A.5 and A.6 show the results when using 1.50 and 1.25 as fitting thresholds, respectively.

The results did not change appreciably when performing segment correction using the ASCEPT-identified changepoints. The only difference was that under fitting thresholds of 1.50 and 1.25, the segment



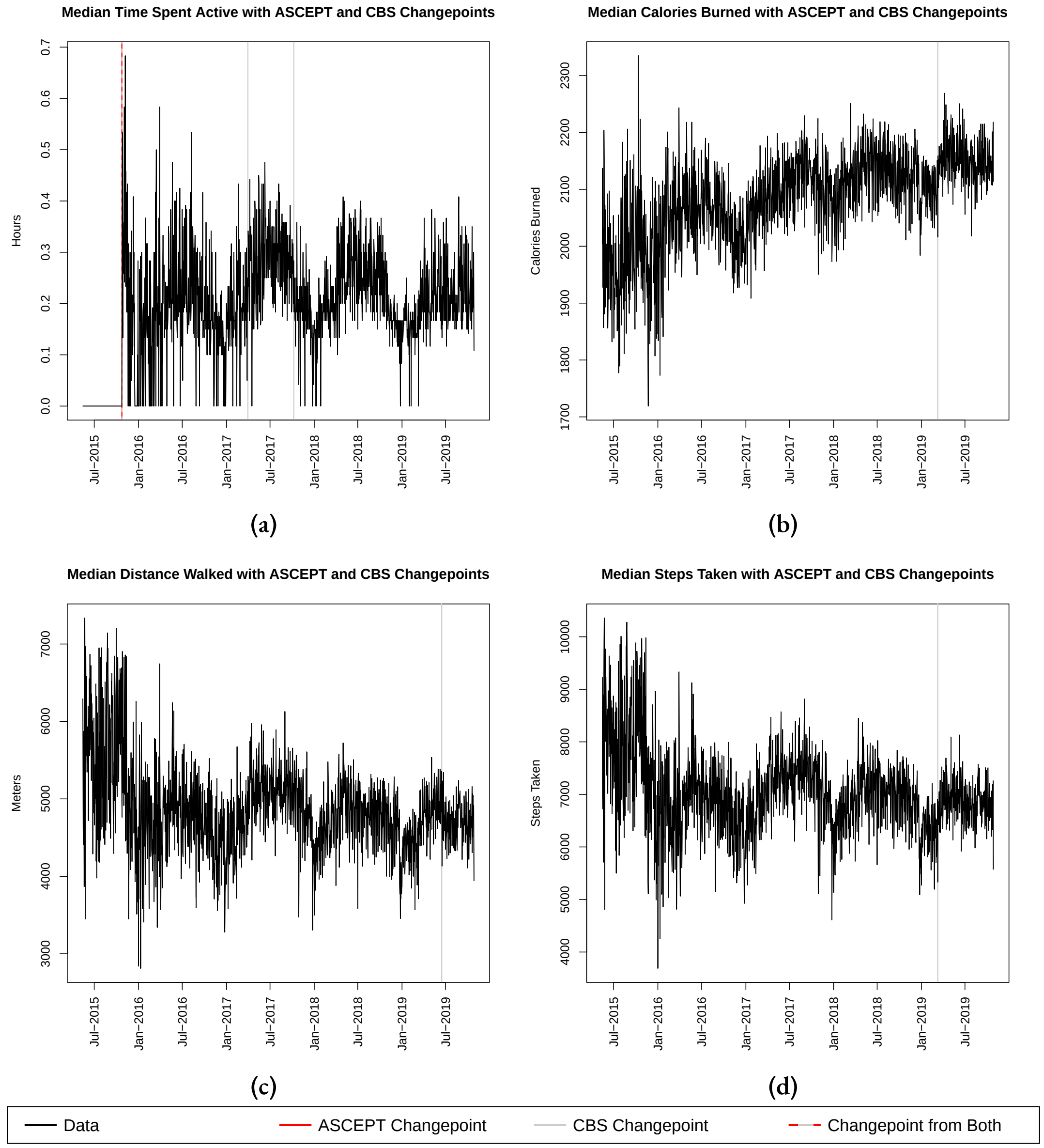
(a)

(b)

(c)

Figure A.3: Comparison of ASCEPT with CBS for (a) median light sleep, (b) median total sleep, and (c) median time awake at night.

from indices 50 to 60 was incorrectly identified to be best fit with a harmonic regression, rather than a constant fit. This segment was therefore transformed slightly differently than it was in Figure


Median Time Spent Active with ASCEPT and CBS Changepoints

Median Calories Burned with ASCEPT and CBS Changepoints

(a)

(b)

Median Distance Walked with ASCEPT and CBS Changepoints

Median Steps Taken with ASCEPT and CBS Changepoints

(c)

(d)

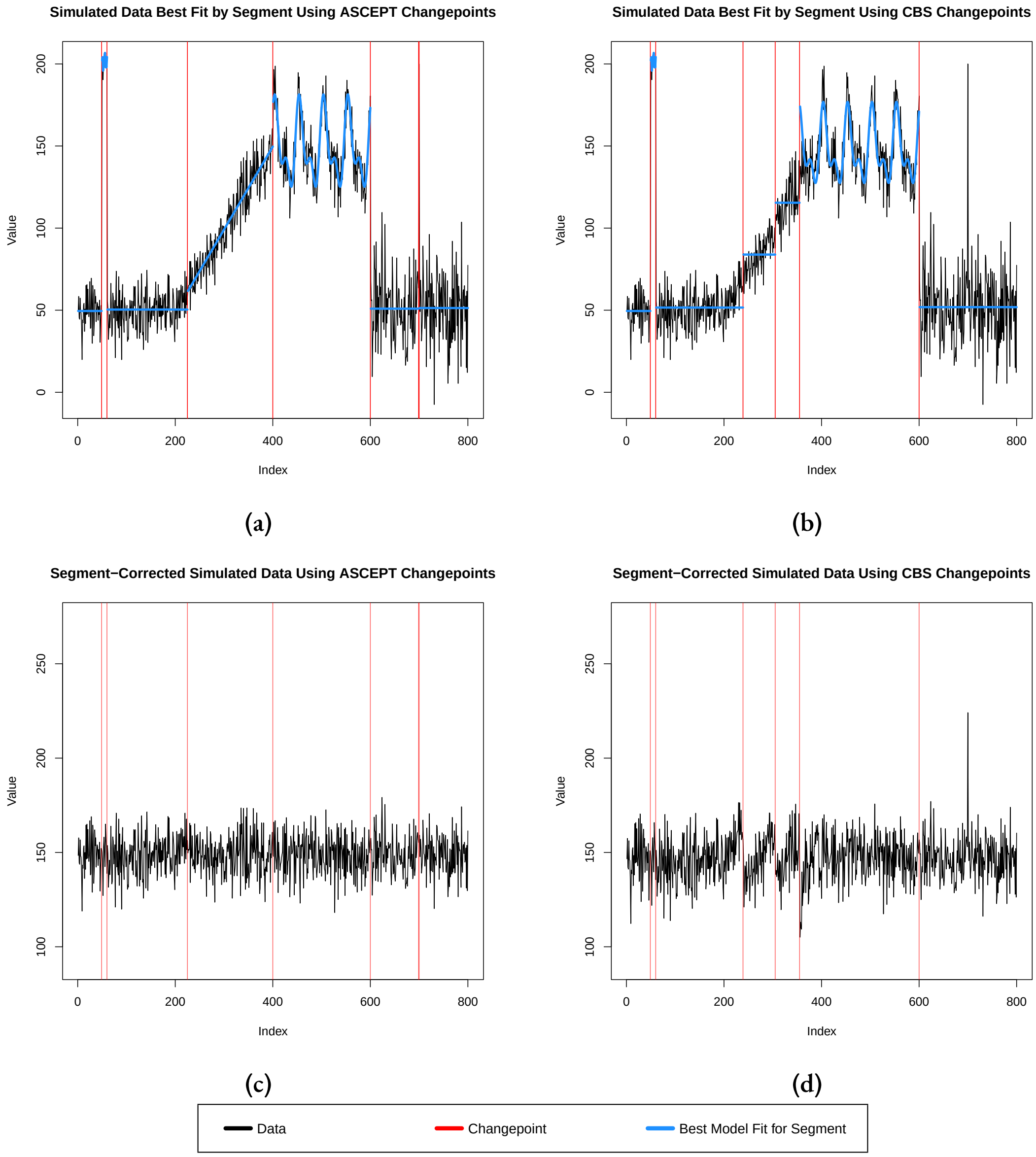
Figure A.4: Comparison of ASCEPT with CBS for (a) median time active, (b) median calories burned, (c) median distance walked, and (d) median steps.

1.6c. Despite this change, the transformed series under ASCEPT changepoints still appeared to be normally distributed noise without any mean-shifts. Supplementary Figures A.5a, A.5c, A.6a, and

A.6c show these results.

For CBS, the linear and seasonal trends were more appropriately modeled using the smaller fitting thresholds, as shown in Supplementary Figures A.5b, A.5d, A.6b, and A.6d. In particular, along the segment corresponding to the seasonal trend, the best fit was now a harmonic regression. Since all segments are scaled to match the residual standard error of this chosen reference segment, the transformed series in Supplementary Figures A.5d and A.6d had smaller spreads than that shown in Figure 1.6d, where the best fit for this segment was identified as a constant trend. However, there were still some issues with the segment correction due to CBS' misidentification of the relevant changepoints. There were clear residual mean-shifts, including linear trends between indices 201 and 400 and the single-point segment at index 700.

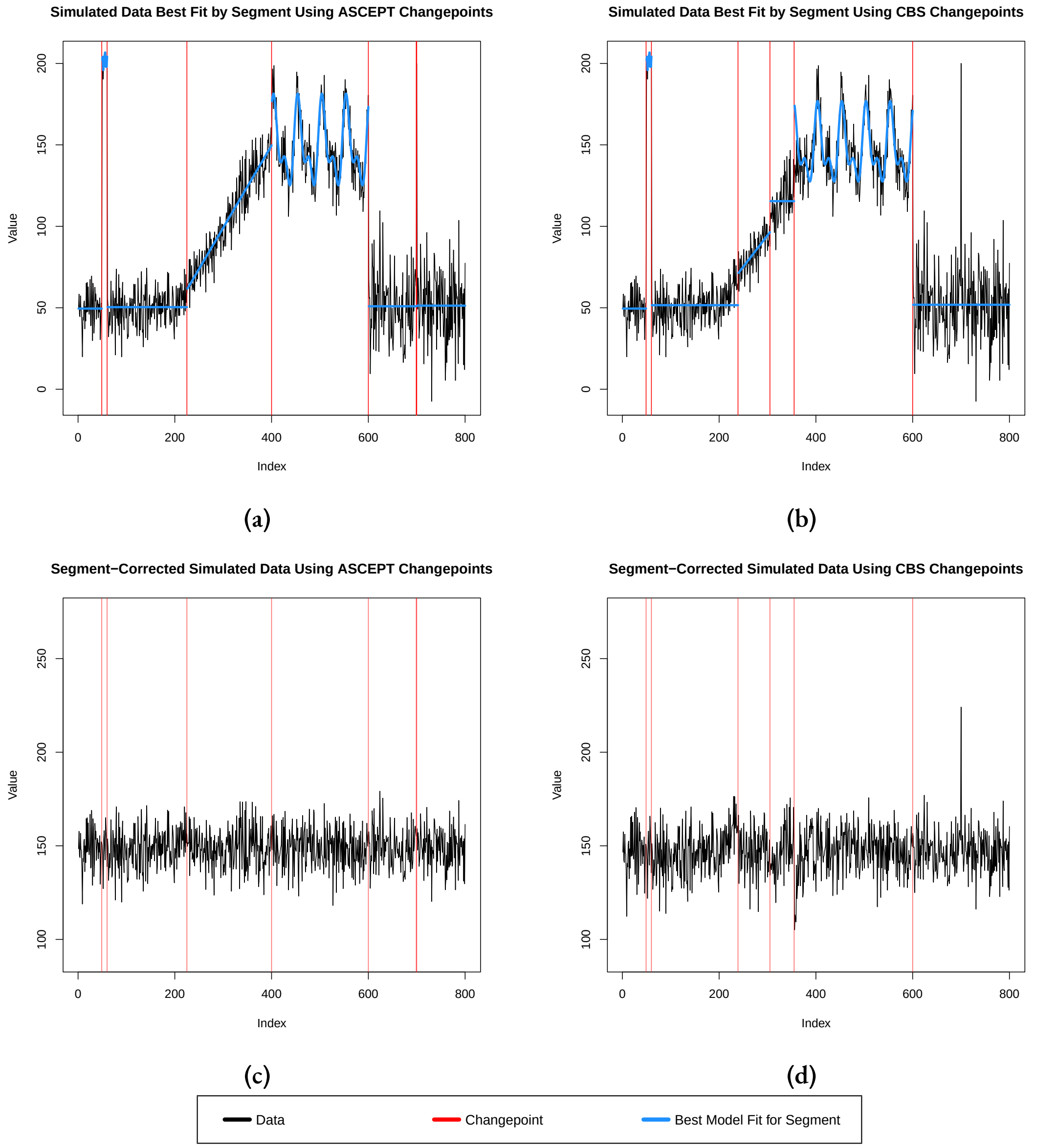
Overall, we found that, while this correction procedure was somewhat sensitive to the chosen fitting threshold, the accurately identified ASCEPT changepoints were more robust to the choice of threshold and yielded more ideal downstream results compared to the less accurate CBS changepoints.



(a) (b)

(c) (d)

Figure A.5: The results of performing segment correcting using a 1.50 fitting threshold. (a) The best model fits using ASCEPT changepoints. (b) The best model fits using CBS changepoints. (c) The corrected series using ASCEPT changepoints. (d) The corrected series using CBS changepoints.



(a)

(b)

(c)

(d)

Figure A.6: The results of performing segment correction using a 1.25 fitting threshold. (a) The best model fits using ASCEPT changepoints. (b) The best model fits using CBS changepoints. (c) The corrected series using ASCEPT changepoints. (d) The corrected series using CBS changepoints.

B

Supplemental Material for Detection and Statistical Inference for Differential Binding Sites

B.1 Additional Details for Data Generation and Processing

B.1.1 Additional Details for the Simulated Data

We used a modified version of the simulation study presented in [53]. The study implemented two different styles of simulations. One represents relatively narrow or sharp binding events associated with TFs. The other represents relatively broad and *complex DB events* associated with histones and HMs.

For TFs, we simulated 20,000 binding sites that were 10,000-20,000 base pairs apart on a single chromosome. Each binding event consisted of a single peak that was 200 base pairs wide, corresponding with a binding site in the center of the peak and an extension of 100 base pairs in either direction to reflect the average fragment length. Of these sites, 1,000 were DB sites and the remaining were non-DB, or null, sites. For every sample in experimental group g, the count at site k was sampled from a negative binomial distribution with mean μ_{kg} and dispersion factor φ_k. Table B.1 lists the values of μ_{k1} and μ_{k2} for different sites. The means for DB sites were balanced as to avoid the need for normalization. Dispersion factors were sampled from an inverse scaled chi squared distribution with scaling factor $\tau^2 = \frac{1}{4}$ and degrees of freedom $\nu = 20$, $\varphi_k \sim \frac{\frac{1}{4} \cdot 20}{\chi^2_{20}} = \frac{5}{\chi^2_{20}}$. The negative binomial count was distributed across the 200 base pair interval according to a Beta(2,2) distribution to provide a smooth binding profile.

Table B.1: Means for samples in each experimental group for the TF simulations.

Site Type	μ_{k1}	μ_{k2}	Number of Sites
DB	112	16	10
DB	84	16	10
DB	56	16	480
DB	16	112	10
DB	16	84	10
DB	16	56	480
Non-DB	36	36	19,000

For HMs or complex DB events, we again simulated 20,000 binding sites that were 10,000-20,000 base pairs apart on a single chromosome. Again, 1,000 were DB sites and the remaining were non-DB, or null, sites. Each binding event was 1,000 base pairs wide and contained three overlapping subintervals, each 500 base pairs wide and equally spaced. For every sample, the counts for every subinterval at site i were sampled from a negative binomial distribution with mean 30 and dispersion factor φ_k. Counts for the three subintervals within the same sample at a given site were sampled with correlation. The correlation between the first and second subintervals and between the second and third subintervals was 0.5. The correlation between the first and third subinterval was 0.25. To create DB sites, the read counts for one or two subintervals for every sample in one of the experimental groups were reduced to zero. The samples in the other group were left unaltered. For 500 of the DB sites, counts in the first group were eliminated, while in the other 500 DB sites, counts in the second group were eliminated. The balanced removal avoided the need for normalization. Dispersion factors were sampled in the same manner as for the TF simulations. The counts for every subinterval were distributed across the subinterval according to a Beta(2,2) distribution. The fragment length was again set to 100 base pairs.

For both TF and HM simulations, background enrichment was added by sampling counts for 2,000 base pair wide bins that partition the genome. The count for bin b was sampled from a negative binomial with mean μ_b and dispersion factor φ_b. μ_b was sampled from a Unif(10,25) and was the same across all samples. φ_b was sampled in the same manner as for the dispersion factors in TF and HM sites. The counts for each bin were uniformly distributed across the bin.

We made multiple adjustments to the simulations. We were interested in comparing the performance of different pipelines in the presence of relatively high and realistic biological variability. Therefore, we adjusted the sampling of dispersion factors from an inverse chi squared distribution with no scaling, $\varphi_k \sim \frac{1}{\chi^2_{20}}$, to one with scaling to closely match dispersion factors estimated from the real H3K4me3 ChIP-seq study [24]. This increased both the mean and variance of dispersion factors. As a result, the mean and variance of the biological variances in simulated sites also increased.

However, increasing variability also increased the difficulty of the simulations. To keep the TF simulations tractable, we increased the means used for sampling counts. The original means for DB sites were 15 and 45, rather than 16 and 56, 84, or 112. Likewise, the mean for non-DB sites was 30 rather than 36. We set 40 DB sites to have exceptionally high means of 84 or 112 so that the pipelines could more easily identify some candidate regions as highly significant. Without these relatively high means, pipelines produced very few candidate regions with reported FDRs below 0.2-0.4, which impacted inference results.

We introduced correlation into the HM simulations to match expectations for real experiments more closely. The original simulation assumed counts for subintervals within the same sample and site to be independent. The correlation of 0.5 between consecutive subintervals and the correlation of 0.25 between the first and third subinterval were chosen based on autocorrelation values observed between windows along the genome in the H3K4me3 ChIP-seq study [24].

Some background sites induced moderately strong and consistent DB signals by random chance. Therefore, we reduced the upper bound of the uniform distribution used to select means for background counts from 50 to 25. While the background enrichment still added noise to the simulation study, this reduction helped restrict the comparison of the different methods to the binding sites alone.

We included a larger experimental setup with 6 replicates in each group (*6 vs. 6*), in addition to the original *2 vs. 2* experiment, to explore how results varied with sample size. We performed five simulations for each experimental setup for each of the TF and HM simulations.

The original simulation study included a third simulation style, in which dispersion factors were constant across all TF binding sites. However, we excluded this simulation since we are primarily interested in the scenario where biological variability itself varies across sites.

B.1.2 Additional Details for the H3K4me3 Data

The original data were obtained from the Gene Expression Omnibus using accession GSE68952 [24]. Reads were aligned to the hg19 human reference genome using Bowtie 2 (v2.3.5.1) in single-end mode [19, 49]. SAMtools (v1.10) was used to sort and index the BAM files [21]. No deduplication was performed.

B.2 Additional Details for Methods

B.2.1 Reading, Filtering, and Normalizing

DB analysis begins with indexed and sorted BAM files reflecting the locations of a TF or HM along a genome. The experiment must contain at least two replicates for each of the two groups, $g \in \{1, 2\}$. An investigator needs to first perform several initial steps common to DB approaches: reading in counts from the BAM files, filtering genomic intervals with low counts, and normalizing across samples. DBFinder is largely agnostic to the decisions made regarding these stages of the analysis. However, below we lay out some common procedures and recommendations for each of these. More detailed discussions and comparisons of different filtering and normalization approaches can be found in [52, 81, 59].

The current implementation of DBFinder reads BAM files using csaw's sliding window approach [54]. However, an investigator may use any method that slides a window of fixed genomic length, typically on the order of tens or low hundreds of base pairs, along the genome in order to count fragments overlapping with these windows. As the window slides, it shifts by a fixed amount, often no larger than the window width such that neighboring windows overlap.

The investigator can then filter windows with relatively low counts across samples. This eliminates windows that likely do not have any DB to avoid unnecessary computation. One may choose to simply filter based on a chosen threshold for the sum of counts within a window across samples. Alternately, one may choose a more advanced approach, such as comparing window-specific counts to average background abundance [54].

One should then normalize the counts to account for possible biases and to ensure that counts are comparable across samples. By default, DBFinder uses the trimmed mean of M values (TMM) normalization [67] procedure on large bins across the genome, as implemented in csaw and edgeR [54, 52, 66]. However, we also transform counts using Anscombe's variance-stabilizing log transformation [12]. Let $\hat{\varphi}$ denote the estimate of the common dispersion across the experiment's count assay [68]. When we normalize raw counts, we add a value of $\frac{1}{2\hat{\varphi}}$ to each count. edgeR accounts for the normalized library sizes of different samples when adding this value to the raw counts when obtaining counts per million [66]. We then log-transform these normalized values using the natural log, yielding y_{ijr} as used in the manuscript.

B.2.2 Pipelines and Software for Detecting Differential Binding Sites

We compare DBFinder with two popular pipelines for detecting DB sites: csaw (v1.24.3) [54] and DiffBind (v3.0.15) [78, 69]. DBFinder is available as an R package at `https://github.com/matthewquinnl/DBFinder`. We performed analyses using R (v4.0.4) and Python 3 (v3.7.4) [63, 7]. We provide details on the parameter values used for each pipeline in the subsequent sections.

B.2.2.1 csaw

csaw uses a sliding window approach to scan over the genome, recording read counts within each window. For TF simulations, we set the window to be 10 base pairs wide and to shift 10 base pairs at a time. For HM simulations and the analysis of H3K4me3, we set the window to be 150 base pairs wide and to shift 10 base pairs at a time. The fragment length is set to 100 base pairs for all simulations and is estimated as 127 base pairs for the analysis of H3K4me3. We do not deduplicate when counting reads. We filter out windows that average fewer than 5 counts per sample in all analyses. For the analysis of H3K4me3, we additionally filter out reads with a mapping quality score below 10 and reads in areas specified by ENCODE's blacklist (version 3) for the hg19 assembly [3, 9]. The steps up to this point are also used for DBFinder.

csaw then filters again by removing windows with low enrichment relative to the background, based on large background bins. These bins are 2,000 base pairs wide for the simulations and 10,000 base pairs wide for the analysis of H3K4me3. In the analysis of H3K4me3, we additionally normalize using edgeR's TMM normalization and 10,000 base pair wide background bins. The csaw procedure for the original simulation used a filtering approach based on relative abundance whose implementation has since changed. Therefore, we updated its relative abundance filtering in accordance with csaw's user guide [52]. For all simulations, we set it such that windows must have a fold change enrichment of two over global background to be incorporated into candidate regions. For the real data analysis of H3K4me3, we set this threshold to a fold change of four because a fold change of two does not remove any additional windows beyond the initial filtering step.

csaw then uses edgeR's quasi-likelihood F-test [66, 56] to perform the DB analysis, obtaining a p-value for the DB effect associated with each window individually. Windows within 100 base pairs of each other are merged into candidate regions. Regions are broken up if they exceed 5,000 base pairs in width. The p-values of the windows within a given candidate region are aggregated using Simes' method [75] in order to obtain a *combined p-value* for that candidate region. Then, the FDRs are calculated for the candidate regions using the Benjamini-Hochberg (BH) procedure [15].

B.2.2.2 DiffBind

DiffBind, unlike a sliding window approach, requires predefined regions of interest to be provided for DB analysis. An investigator typically passes peaks detected by a ChIP-seq peak caller, such as MACS/MACS2 [88], HOMER [35], or SICER [84] into DiffBind for this purpose. For all analyses, we use MACS2 (v2.2.7.1) as the peak caller which reads with a fragment length of 100 base pairs for the simulations and 127 base pairs for the analysis of H3K4me3. DiffBind then reads counts for the peaks, using the same fragment lengths as for MACS2. We do not deduplicate when counting reads. For the analysis of H3K4me3, we additionally filter out reads with a mapping quality score below 10 and reads in areas specified by ENCODE's blacklist (version 3) for the hg19 assembly [3, 9].

DiffBind merges peaks that overlap within samples and across at least two samples into overall *consensus peaks*, which serve as candidate regions with possible DB. The consensus peaks are centered around consensus summits, which are then extended upstream and downstream by a specified amount. For TF simulations, they were extended by 200 base pairs in either direction, yielding 400 base pair wide consensus peaks. For HM simulations and H3K4me3, they were extended by 500 base pairs in either direction, yielding 1,000 base pair wide consensus peaks. The original simulation used 200 base pair extensions for both TF and HM sites, but we chose a larger extension for HM sites so that its consensus peaks are more comparable in size to the true HM binding events. In the analysis of H3K4me3, we additionally normalize using edgeR's TMM normalization and 10,000 base pair wide background bins. The pipeline then tests these consensus peaks for DB using either DESeq2 [51] or edgeR [66, 56]. For consistency with csaw, we use edgeR's quasi-likelihood F-test.

B.2.2.3 DBFinder

DBFinder starts with counting and initial filtering procedures identical to those for csaw for all analyses. For TF simulations, we set the window to be 10 base pairs wide and to shift 10 base pairs at a time. For HM simulations and the analysis of H3K4me3, we set the window to be 150 base pairs wide and to shift 10 base pairs at a time. The fragment length is set to 100 base pairs for all simulations and is estimated as 127 base pairs for the analysis of H3K4me3. We do not deduplicate when counting reads. We filter out windows that average fewer than 5 counts per sample in all analyses. For the analysis of H3K4me3, we additionally filter out reads with a mapping quality score below 10 and reads in areas specified by ENCODE's blacklist (version 3) for the hg19 assembly [3, 9].

For the simulations, we apply Anscombe's variance-stabilizing log transformation to the raw counts since no normalization is necessary. In the analysis of H3K4me3, we normalize using edgeR's TMM normalization and 10,000 base pair wide background bins. We apply Anscombe's variance-stabilizing log transformation as discussed in the manuscript and Appendix B.2.1 to account for the normalization.

For all analyses, we then identify candidate regions using a target of $20,000 \pm 250$ regions, regardless of the direction of the DB effect in each region. We required candidate regions to contain at least three windows minimum in all analyses. For the simulations with two samples in each group, we perform the permutation test using all balanced permutations. For the simulations with six samples in each group and for the analysis of H3K4me3, we perform the permutation test using five randomly selected balanced permutations. We calculate p-values for both the estimated DB effects and the corresponding moderated t-statistics. We then calculate FDRs for the candidate regions using the BH procedure [15].

B.2.3 Calculation of ROC Curves and FDRs for the Simulations

We slightly adjust the calculation of receiver operating characteristic (ROC) curves and FDRs compared to that done for the original simulation [54]. Instead of checking for any amount of overlap between candidate regions and binding events, we require an overlap of 25 base pairs for a candidate region to catch the corresponding binding site, whether it be DB or non-DB.

Additionally, for HM simulations, we count non-DB portions of larger DB events as non-DB events themselves. As discussed in Section 2.2.1 and Appendix B.1.1, events in the HM simulations contain three subintervals. However, in complex DB events, only one or two of these subintervals will contain DB. Therefore, for every DB site, except for those where the first and third subintervals are both DB, there is a portion of the complex DB event that is actually non-DB. The original simulation only accounted for the DB subintervals of complex DB events in the calculation of ROC curves. That is, a candidate region could only be rewarded for overlapping with a DB event, even if it largely overlapped with the non-DB portion of that event as well. This approach rewards large and imprecise candidate regions that overlap with DB subintervals, regardless of what else they include.

Instead, we record the portions of complex DB events that are non-DB and account for them when calculating specificity for ROC curves. This rewards candidate regions with greater precision on the base pair-level. This adjustment does not impact or penalize FDRs because we can only count a single candidate region as a true positive or false positive, not both. Therefore, when calculating observed FDRs, we take one minus the proportion of significant candidate regions that overlap with a true DB event, regardless of whether it also overlaps with a non-DB event.

Similarly, when two subintervals are DB in a complex DB event, we count the DB subintervals separately when calculating sensitivity, instead of merging them. This ensures that candidate regions are only rewarded as identifying two DB subintervals in the same event if they overlap sufficiently with both subintervals. If these DB subintervals are merged, a candidate region only needs to capture one in order to be marked as catching both. While the subintervals are part of the same event, they reflect different randomly drawn counts and DB signals based on sampling variability. Therefore, we consider it important to treat them as distinct since a pipeline could reasonably model them separately or assign distinct p-values to each subinterval.

Lastly, we average the ROCs and observed FDRs over the five simulations for each experimental setup. For ROCs, we average the sensitivity and specificity at each reported FDR threshold across the simulations. When comparing how pipelines control FDRs, we average the observed FDR at each reported FDR threshold across the simulations.

References

[1] Community. https://community.fitbit.com/. Accessed: 2020-06-06.

[2] What's changed in the latest Fitbit device update? https://help.fitbit.com/articles/en_US/Help_article/1372. Accessed: 2020-06-06.

[3] (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414), 57–74. https://sites.google.com/site/anshulkundaje/projects/blacklists.

[4] (2017a). Alta HR Firmware Release - 26.62.6. https://community.fitbit.com/t5/Alta-HR/Alta-HR-Firmware-Release-26-62-6/td-p/2119538. Accessed: 2020-06-06.

[5] (2017b). Charge 2 Sleep Stages. https://community.fitbit.com/t5/Charge-2/Charge-2-Sleep-Stages/td-p/1907433. Accessed: 2020-06-11.

[6] (2017c). Received Classic Sleep rather than Sleep Stages. https://community.fitbit.com/t5/Blaze/RESOLVED-9-3-Received-Classic-Sleep-rather-than-Sleep-Stages/td-p/2174227. Accessed: 2020-06-11.

[7] (2022). *Python Language Reference*. Python Software Foundation.

[8] Amaratunga, D. & Cabrera, J. (2001). Analysis of data from viral DNA microchips. *Journal of the American Statistical Association*, 96(456), 1161–1170.

[9] Amemiya, H. M., Kundaje, A., & Boyle, A. P. (2019). The ENCODE blacklist: Identification of problematic regions of the genome. *Scientific Reports*, 9(1). https://sites.google.com/site/anshulkundaje/projects/blacklists.

[10] Anders, S. & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11(10).

[11] Andonegui-Elguera, S. D., Zamora-Fuentes, J. M., Espinal-Enríquez, J., & Hernández-Lemus, E. (2021). Loss of long distance co-expression in lung cancer. *Frontiers in Genetics*, 12.

[12] Anscombe, F. J. (1948). The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3-4), 246–254.

[13] Auger, I. & Lawrence, C. (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, (pp. 39–54).

[14] Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1–48.

[15] Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1), 289–300.

[16] Bolstad, B., Irizarry, R., Astrand, M., & Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2), 185–193.

[17] Bottomly, D., Kyler, S. L., McWeeney, S. K., & Yochum, G. S. (2010). Identification of beta-catenin binding regions in colon cancer cells using ChIP-Seq. *Nucleic Acids Research*, 38(17), 5735–5745.

[18] Chung, A., Gotz, D., Kappelman, M., Mentch, L., Glass, K., & Gehlenborg, N. (2019). Precision VISSTA: Enabling Precision Medicine through the Development of Quantitative and Visualization Methods. http://precisionvissta.web.unc.edu/. Accessed: 2020-3-14.

[19] Church, D. M., Schneider, V. A., Graves, T., Auger, K., Cunningham, F., Bouk, N., Chen, H.-C., Agarwala, R., McLaren, W. M., Ritchie, G. R., Albracht, D., Kremitzki, M., Rock, S., Kotkiewicz, H., Kremitzki, C., Wollam, A., Trani, L., Fulton, L., Fulton, R., Matthews, L., Whitehead, S., Chow, W., Torrance, J., Dunn, M., Harden, G., Threadgold, G., Wood, J., Collins, J., Heath, P., Griffiths, G., Pelan, S., Grafham, D., Eichler, E. E., Weinstock, G., Mardis, E. R., Wilson, R. K., Howe, K., Flicek, P., & Hubbard, T. (2011). Modernizing reference genome assemblies. *PLoS Biology*, 9(7), e1001091.

[20] Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., & Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology*, 17(1).

[21] Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., & Li, H. (2021). Twelve years of SAMtools and BCFtools. *GigaScience*, 10(2).

[22] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1–38.

[23] den Berge, K. V., Hembach, K. M., Soneson, C., Tiberi, S., Clement, L., Love, M. I., Patro, R., & Robinson, M. D. (2019). RNA sequencing data: Hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, 2(1), 139–173.

[24] Dong, X., Tsuji, J., Labadorf, A., Roussos, P., Chen, J.-F., Myers, R. H., Akbarian, S., & Weng, Z. (2015). The role of h3k4me3 in transcriptional regulation is altered in huntington's disease. *PLOS ONE*, 10(12), e0144398.

[25] Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *Ann. Statist.*, 42(6), 2243–2281.

[26] GOWER, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4), 325–338.

[27] GTEx Consortium (2013). The genotype-tissue expression (GTEx) project. *Nature Genetics*, 45(6), 580–585.

[28] GTEx Consortium (2015). The genotype-tissue expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235), 648–660.

[29] guo Zhou, X., liang Huang, X., yuan Liang, S., mei Tang, S., kao Wu, S., tong Huang, T., nan Mo, Z., & yan Wang, Q. (2018). Identifying miRNA and gene modules of colon cancer associated with pathological stage by weighted gene co-expression network analysis. *OncoTargets and Therapy*, Volume 11, 2815–2830.

[30] Haas, B. E., Horvath, S., Pietiläinen, K. H., Cantor, R. M., Nikkola, E., Weissglas-Volkov, D., Rissanen, A., Civelek, M., Cruz-Bautista, I., Riba, L., Kuusisto, J., Kaprio, J., Tusie-Luna, T., Laakso, M., Aguilar-Salinas, C. A., & Pajukanta, P. (2012). Adipose co-expression networks across finns and mexicans identify novel triglyceride-associated genes. *BMC Medical Genomics*, 5(1).

[31] Haque, A., Engel, J., Teichmann, S. A., & Lönnberg, T. (2017). A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9(1).

[32] Harrison, P. (2015). Anscombe's 1948 variance stabilizing transformation for the negative binomial distribution is well suited to rna-seq expression data.

[33] Harvey, D. I., Leybourne, S. J., & Taylor, A. M. R. (2011). Testing for unit roots and the impact of quadratic trends, with an application to relative primary commodity prices. *Econometric Reviews*, 30(5), 514–547.

[34] Haynes, K., Eckley, I. A., & Fearnhead, P. (2017). Computationally efficient changepoint detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26(1), 134–143.

[35] Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H., & Glass, C. K. (2010). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular Cell*, 38(4), 576–589.

[36] Hicks, S. C., Okrah, K., Paulson, J. N., Quackenbush, J., Irizarry, R. A., & Bravo, H. C. (2017). Smooth quantile normalization. *Biostatistics*, 19(2), 185–198.

[37] Hsieh, P.-H., Lopes-Ramos, C. M., Sandve, G. K., Glass, K., & Kuijjer, M. L. (2021). Adjustment of spurious correlations in co-expression measurements from RNA-sequencing data.

[38] Illumina (2020). Precise analysis of DNA-protein binding sequences. https://www.illumina.com/techniques/sequencing/dna-sequencing/chip-seq.html. Accessed: 2020-3-14.

[39] Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., & Tsai, T. T. (2005). An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2), 105–108.

[40] Kang, H., Lee, J., & Yu, S. (2016). Differential co-expression networks using RNA-seq and microarrays in alzheimer's disease. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*: IEEE.

[41] Kelli, H. M., Witbrodt, B., & Shah, A. (2017). The future of mobile health applications and devices in cardiovascular health. *European Medical Journal Innovations*, (pp. 92–97).

[42] Kennedy, B., Deatherage, D., Gu, F., Tang, B., Chan, M., Nephew, K., Huang, T., & Jin, V. (2011). ChIP-seq Defined Genome-Wide Map of TGFbeta/SMAD4 Targets: Implications with Clinical Outcome of Ovarian Cancer. *PloS One*, 6(7).

[43] Khatoon, Z., Figler, B., Zhang, H., & Cheng, F. (2014). Introduction to RNA-seq and its applications to drug discovery and development. *Drug Development Research*, 75(5), 324–330.

[44] Killick, R. & Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal of Statistical Software*, 58(3), 1–19.

[45] Killick, R., Fearnhead, P., & Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500), 1590–1598.

[46] Korthauer, K., Chakraborty, S., Benjamini, Y., & Irizarry, R. A. (2018). Detection and accurate false discovery rate control of differentially methylated regions from whole genome bisulfite sequencing. *Biostatistics*, 20(3), 367–383.

[47] Kosecki, D. (2017). New Fitbit Features Deliver Data Previously Only Available Through a Sleep Lab. https://blog.fitbit.com/sleep-stages-and-sleep-insights-announcement/. Accessed: 2020-06-11.

[48] Kumar, S., Nilsen, W. J., Abernethy, A., Atienza, A., Patrick, K., Pavel, M., Riley, W. T., Shar, A., Spring, B., Spruijt-Metz, D., Hedeker, D., Honavar, V., Kravitz, R., Lefebvre, R. C., Mohr, D. C., Murphy, S. A., Quinn, C., Shusterman, V., & Swendeman, D. (2013). Mobile health technology evaluation: The mhealth evidence workshop. *American Journal of Preventive Medicine*, 45(2), 228 – 236.

[49] Langmead, B. & Salzberg, S. L. (2012). Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9(4), 357–359.

[50] Lee, Y., Park, D., & Iyer, V. R. (2017). The ATP-dependent chromatin remodeler chd1 is recruited by transcription elongation factors and maintains h3k4me3/h3k36me3 domains at actively transcribed and spliced genes. *Nucleic Acids Research*, 45(12), 7180–7190.

[51] Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15, 550.

[52] Lun, A. (2021). The csaw book. http://bioconductor.org/books/3.13/csawBook/.

[53] Lun, A. T. & Smyth, G. K. (2014). De novo detection of differentially bound regions for ChIP-seq data using peaks and windows: controlling error rates correctly. *Nucleic Acids Research*, 42(11), e95–e95.

[54] Lun, A. T. L. & Smyth, G. K. (2016). csaw: a Bioconductor package for differential binding analysis of ChIP-seq data using sliding windows. *Nucleic Acids Res.*, 44(5), e45.

[55] Lynch, G., Guo, W., Sarkar, S. K., & Finner, H. (2017). The control of the false discovery rate in fixed sequence multiple testing. *Electronic Journal of Statistics*, 11(2), 4649 – 4673.

[56] McCarthy, D. J., Chen, Y., & Smyth, G. K. (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10), 4288–4297.

[57] Mei, S., Qin, Q., Wu, Q., Sun, H., Zheng, R., Zang, C., Zhu, M., Wu, J., Shi, X., Taing, L., Liu, T., Brown, M., Meyer, C. A., & Liu, X. S. (2016). Cistrome data browser: a data portal for ChIP-seq and chromatin accessibility data in human and mouse. *Nucleic Acids Research*, 45(D1), D658–D662.

[58] Motallebipour, M., Ameur, A., Bysani, M. S. R., Patra, K., Wallerman, O., Mangion, J., Barker, M. A., McKernan, K. J., Komorowski, J., & Wadelius, C. (2009). Differential binding and co-binding pattern of FOXA1 and FOXA3 and their relation to h3k4me3 in HepG2 cells revealed by ChIP-seq. *Genome Biology*, 10(11), R129.

[59] Nakato, R. & Sakata, T. (2021). Methods for ChIP-seq analysis: A practical workflow and advanced applications. *Methods*, 187, 44–53.

[60] Namani, A., Liu, K., Wang, S., Zhou, X., Liao, Y., Wang, H., Wang, X. J., & Tang, X. (2019). Genome-wide global identification of NRF2 binding sites in A549 non-small cell lung cancer cells by ChIP-Seq reveals NRF2 regulation of genes involved in focal adhesion pathways. *Aging*, 11(24), 12600–12623.

[61] Olshen, A., Venkatraman, E. S., Lucito, R., & Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4), 557–572.

[62] Paulson, J. N., Chen, C.-Y., Lopes-Ramos, C. M., Kuijjer, M. L., Platig, J., Sonawane, A. R., Fagny, M., Glass, K., & Quackenbush, J. (2017). Tissue-aware RNA-seq processing and normalization for heterogeneous and sparse data. *BMC Bioinformatics*, 18(1).

[63] R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[64] Rider, P. R. (1960). Variance of the median of small samples from several special populations. *Journal of the American Statistical Association*, 55(289), 148–150.

[65] Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7), e47–e47.

[66] Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1), 139–140.

[67] Robinson, M. D. & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3), R25.

[68] Robinson, M. D. & Smyth, G. K. (2007). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2), 321–332.

[69] Ross-Innes, C. S., Stark, R., Teschendorff, A. E., Holmes, K. A., Ali, H. R., Dunning, M. J., Brown, G. D., Gojis, O., Ellis, I. O., Green, A. R., Ali, S., Chin, S.-F., Palmieri, C., Caldas, C., & Carroll, J. S. (2012). Differential oestrogen receptor binding is associated with clinical outcome in breast cancer. *Nature*, 481, –4.

[70] Samuel-Cahn, E. (1996). Is the simes improved bonferroni procedure conservative? *Biometrika*, 83(4), 928–933.

[71] Sarkar, S. K. & Chang, C.-K. (1997). The simes method for multiple hypothesis testing with positively dependent test statistics. *Journal of the American Statistical Association*, 92(440), 1601–1608.

[72] Seshan, V. E. & Olshen, A. (2019). *DNAcopy: DNA copy number data analysis*. R package version 1.60.0.

[73] Shen, L., Shao, N.-Y., Liu, X., Maze, I., Feng, J., & Nestler, E. J. (2013). diffReps: Detecting differential chromatin modification sites from ChIP-seq data with biological replicates. *PLoS ONE*, 8(6), e65598.

[74] Silva, B. M., Rodrigues, J. J., de la Torre Díez, I., López-Coronado, M., & Saleem, K. (2015). Mobile-health: A review of current state in 2015. *Journal of Biomedical Informatics*, 56, 265 – 272.

[75] Simes, R. J. (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3), 751–754.

[76] Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 1–25.

[77] Sonawane, A. R., Platig, J., Fagny, M., Chen, C.-Y., Paulson, J. N., Lopes-Ramos, C. M., DeMeo, D. L., Quackenbush, J., Glass, K., & Kuijjer, M. L. (2017). Understanding tissue-specific gene regulation. *Cell Reports*, 21(4), 1077–1088.

[78] Stark, R. & Brown, G. (2011). *DiffBind: differential binding analysis of ChIP-Seq peak data*.

[79] Steinhauser, S., Kurzawa, N., Eils, R., & Herrmann, C. (2016). A comprehensive comparison of tools for differential ChIP-seq analysis. *Briefings in Bioinformatics*, 17(6), 953–966.

[80] Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167, 107299.

[81] Tu, S. & Shao, Z. (2017). An introduction to computational tools for differential binding analysis with ChIP-seq data. *Quantitative Biology*, 5(3), 226–235.

[82] van Dam, S., Võsa, U., van der Graaf, A., Franke, L., & de Magalhães, J. P. (2017). Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in Bioinformatics*, (pp. bbw139).

[83] VonHandorf, A., Zablon, H. A., Biesiada, J., Zhang, X., Medvedovic, M., & Puga, A. (2021). Hexavalent chromium promotes differential binding of CTCF to its cognate sites in euchromatin. *Epigenetics*, 16(12), 1361–1376.

[84] Xu, S., Grullon, S., Ge, K., & Peng, W. (2014). Spatial clustering for identification of ChIP-enriched regions (SICER) to map regions of histone methylation patterns in embryonic stem cells. *Methods in Molecular Biology*, (pp. 97–111).

[85] Yin, W., Mendoza, L., Monzon-Sandoval, J., Urrutia, A. O., & Gutierrez, H. (2021). Emergence of co-expression in gene regulatory networks. *PLOS ONE*, 16(4), e0247671.

[86] Zhang, J., Wu, Y., Jin, H.-Y., Guo, S., Dong, Z., Zheng, Z.-C., Wang, Y., & Zhao, Y. (2018). Prognostic value of sorting nexin 10 weak expression in stomach adenocarcinoma revealed by weighted gene co-expression network analysis. *World Journal of Gastroenterology*, 24(43), 4906–4919.

[87] Zhang, Y., Lin, Y.-H., Johnson, T. D., Rozek, L. S., & Sartor, M. A. (2014). PePr: a peak-calling prioritization pipeline to identify consistent or differential peaks from replicated ChIP-seq data. *Bioinformatics*, 30(18), 2568–2575.

[88] Zhang, Y., Liu, T., Meyer, C., Eeckhoute, J., Johnson, D., Bernstein, B., Nusbaum, C., Myers, R., Brown, M., Li, W., & Liu, S. (2008). Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, 9(R137).

[89] Zheng, R., Wan, C., Mei, S., Qin, Q., Wu, Q., Sun, H., Chen, C.-H., Brown, M., Zhang, X., Meyer, C. A., & Liu, X. S. (2018). Cistrome data browser: expanded datasets and new tools for gene regulatory analysis. *Nucleic Acids Research*, 47(D1), D729–D735.