Stony Brook University

On miRNA-mRNA network extraction and ultra-fast nucleotide barcodes clustering algorithm

A Dissertation Presented

by

Lu Zhao

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

In

Applied Mathematics and Statistics

Stony Brook University

August 2016

The Graduate School

Lu Zhao

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Professor Wei Zhu
Department of Applied Mathematics and Statistics

Assistant Professor Pei Fen Kuan
Department of Applied Mathematics and Statistics

Assistant Professor Yi Gao
Department of Biomedical Informatics

Professor Wadie F. Bahou
School of Medicine

This dissertation is accepted by the Graduate School

Nancy Goroff
Interim Dean of the Graduate School

Abstract of the Dissertation

On miRNA-mRNA network extraction and ultra-fast nucleotide barcodes clustering

Lu

Doctor of Philosophy

in

Applied Mathematics and Statistics

Stony Brook University

2016

This thesis consists of two topics: (1) discovery of microRNA/mRNA regulatory networks on essential thrombocytosis (ET), and (2) a novel ultrafast clustering algorithm to count nucleotide barcode and amplicon reads with errors.

The objective of the first study is to discover miRNA-mRNA regulatory networks related to ET, a chronic myeloproliferative disorder with an unregulated surplus of platelets. Complications of ET include stroke, heart attack, and formation of blood clots. While the genetic basis of ET has been studied to some extent, no direct diagnostic test is available to date. In this study, we aim to identify novel ET-related miRNA-mRNA regulatory networks through comparisons of transcriptomes between healthy control and ET patients. Four network discovery algorithms have been employed, including (a) Pearson correlation network, (b) sparse supervised canonical correlation analysis (sparse sCCA), (c) sparse partial correlation network analysis (SPACE), and, (d) (sparse) Bayesian network analysis – all through a combination of data-driven and knowledge-based analyses. The result predicts a close relationship between 8 miRNAs (including miR-9, miR-490-5p, miR-490-3p, miR-182, miR-34a, miR-196b, miR-34b*, miR-181a-2*) and a 9-mRNA set (including CAV2, LAPTM4B, TIMP1, PKIG, WASF1, MMP1,

ERVH-4, NME4, HSD17B12). The majority of the identified variables have been linked to hematologic function by a sizable number of studies. Furthermore, it is observed that the selected mRNAs are high relevant to ET disease. The study will shed light on understanding the etiology of ET.

The objective of the second study is to develop an ultrafast and accurate clustering algorithm and software to detect barcodes, certain DNA sequences, and their abundances from raw next-generation barcode sequencing (bar-seq) data. Although bar-seq use has been quickly growing, the computational pipelines for its analyses have not been well developed. Available methods are slow and often result in over-clustering artifacts that group distinct barcodes together. Here, we developed a software package called Bartender, which employs a divide-and-conquer strategy for fast implementation and a modified two-sample proportion test for cluster merging. Additionally, Bartender includes a "multiple time point" mode that matches barcode clusters between different clustering runs for seamless handling of time course data. For both simulated and real data, Bartender clusters millions of unique barcodes in a few minutes at high accuracy (>99.9%), and is ~100-fold faster than previous methods.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RNA - Ribonucleic acid

miRNA - microRNA

mRNA – message RNA

ET - essential thrombocytosis

SPACE -- Sparse Partial Correlation Estimation

BN -- Bayesian Network

SCCA – Sparse Canonical Correlation Analysis

sSCCA – Sparse Supervised Canonical Correlation Analysis

NGS -- Next Generation Sequencing

BLAST -- Basic Local Alignment Search Tool

CV -- Coefficient of Variation

UMI -- Unique Molecular Identifiers

Preface

This dissertation is an original, unpublished and independent work by the author, Lu Zhao, and consists of two self-contained papers. The numbered lists of Figures, Tables and Bibliographies are in a combined format.

The thesis includes the mathematical model, algorithm and selected results accomplished during the Ph.D. study of the author at Stony Brook University, New York. It first discusses the discovery of microRNA/messenger RNA regulatory networks on essential thrombocytosis, through a thorough comparison of four existing miRNA-mRNA network extraction methods. It also presents a novel ultrafast clustering algorithm to count nucleotide barcode and amplicon reads.

Part I miRNA-mRNA regulatory network analyses

1 Background

Platelets are anucleate blood cells that play an important role in haemostasis and thrombosis. Thrombocytosis is a disorder of platelet overproduction in the blood, which has two major forms: essential/primary thrombocytosis (ET) and reactive/secondary thrombocytosis (RT). Essential thrombocytosis is caused by a chronic myeloproliferation with an unregulated surplus of platelets attributed to a malfunction in the body's feedback system. Complications of ET include stroke, heart attack, and formation of blood clots. To date, the genetic basis of ET is still under full investigation and no direct diagnostic tests are available (Gnatenko, Cupit et al. 2005).

Messenger RNA (mRNA) is an RNA molecule that is transcribed from a DNA template. It embodies the genetic information and acts as the template in the process of protein synthesis (Kozak Mar. 1983). microRNA (miRNA) is a single-stranded 21 to 23 nucleotide RNA molecule that binds to mRNAs through a complementary pairing to the 3'-untranslated region (UTR) of mRNAs (Edelstein and Bray 2011) and subsequently regulates their translation or stability (Edelstein, McKenzie et al. 2013). Increasing evidence demonstrates that miRNAs play important roles in various biological processes, through regulating expression levels of their target genes.

Many computational methods have been developed to study interactions between miRNA and mRNA, which are largely based on two types of methods: one is computer-based method that uses the sequence complementarities of miRNA and its mRNA targets to build *in silicon* interaction databases, including MiRBase (Griffiths-Jones, Saini et al. 2008), TargetScan (Lewis, Burge et al. 2005, Grimson, Farh et al. 2007) and so on; the second one is data-based method that examines expression profiles of miRNAs and mRNAs for negative correlations. For example,

GenMiR++ (Huang, Babak et al. 2007, Huang, Morris et al. 2007) and HOCTAR (Gennarino, Sardiello et al. 2009) predicts the interaction between miRNA and mRNA by integrating the expression profiling and sequence-based recognition software. Several other methods that are based on solely on expression profile have also been published. Jayaswal (Jayaswal, Lutherborrow et al. 2011) developed a two-stage procedure that first clusters each expression data for miRNA and mRNA and then identify significant miRNA-mRNA relationship using t-test. Li (Li, Gill et al. 2011) proposed a method to find a set of differentially expressed miRNAs and mRNAs via Partial Least Squares Regression. It is very challenging to build causal relationship using observational data. Le (Le, Liu et al. 2013) designed a algorithm to uncover the causal regulatory relationship between miRNAs and mRNAs, using expression profiles of miRNAs and mRNAs without taking into consideration the previous target information. It is based on Intervention calculus when the Directed Acrylic Graph (DAG) is absent (IDA) (Maathuis, Kalisch et al. 2009).

While all above methods focus on uncovering interaction between individual miRNA and mRNA, there is a growing body of literature showing that multiple miRNAs are coordinated by forming cohesive groups to collectively regulate one or more mRNAs (Boross, Orosz et al. 2009). The complex regulatory network formed between a group of miRNAs and a group of mRNAs acts as a vital force in catering similar functioning miRNAs and mRNAs together, and may provide better understandings on robust and potent miRNA-mRNA regulatory modules (MMRMs) (Masud Karim, Liu et al. 2016).

In this study, we explore the potential miRNA/mRNA regulatory networks associated to ET based on a 43-member cohort, through a combination of data-driven and knowledge-based analyses. Three classes of correlation network analyses methods, namely, the Pearson correlation

network, the sparse canonical correlation network, and the sparse partial correlation network have been implemented, compared and finally, integrated for a more reliable miRNA-mRNA pathway. This pathway was subsequently examined for its biological functionalities through an Ingenuity Pathway Analysis. Additionally, we have also applied a sparse Bayesian Network paradigm, the A* Lasso, to compare with the three correlational network analysis methods.

2 Methods

2.1 Pearson Correlation, Partial Correlation, and the Sparse Partial Correlation Network

Correlation is a widely used concept to describe how two random variables are related to each other. The first correlation measurement was created by Francis Galton, Karl Pearson's mentor, in 1888 to relate measurements under different conditions (Stigler 1989). The degree of correlation is measured by a correlation coefficient, usually denoted as ρ (pho) for a population and by r for a sample. Another well-known measurement is Pearson product-moment correlation coefficient (or Pearson's r), which was introduced by Karl Pearson to measure the linear dependence between two variables (Pearson 1920). The value of Pearson's r always lies between -1 and 1. The geometrical interpretation of Pearson's r can be considered as the cosine of the angle between two vectors in the Euclidean space, each pointing from the variable mean to the origin point (Fisher 1924). By Fisher's transformation, correlation coefficient would follow approximately a normal distribution.

A high correlation between two variables in a network may be indicative of three potential situations: (1) direct interaction, (2) indirect interaction, or (3) regulation through a third common variable (Figure 1). To explore the functional mechanisms between biological

molecules, most investigators would be primarily interested in the direct. For this purpose, the partial correlation coefficient (Yule 1907, Pearson 1915, Fisher 1924) has been designed.

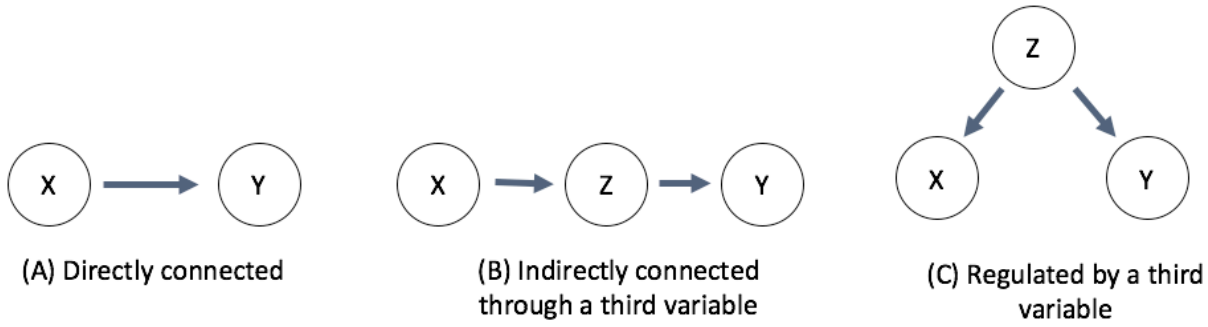


Figure 1 Potential relationships between two variables with high correlation coefficient
In the situations depicted in the center and right-hand side of the figure, the partial correlation coefficients of X and Y given Z would be zero (assuming ideal experimental conditions). Hence partial correlation coefficient prevents false positives due to indirect, rather than direct effects between two variables.

The partial correlation coefficient measures the degree of dependence between two random variables (e.g. miRNA and mRNA expressions) while controlling for the effects from other variables. For example, the correlation $r_{ij.p}$ between variables X_i and X_j conditioning on X_p is the correlation between the parts of X_i and X_j that are uncorrelated with X_p. Under the normality assumption, a partial correlation coefficient of zero between two random variables means that they are conditionally independent given the remaining variables. Given a group of random variables, we are typically interested in the following questions for each pairwise analysis: (1) measuring the strength of a relationship (i.e. the magnitude of a partial correlation coefficient); (2) determining whether a relationship is significant; and (3) comparing the relationship of the same pair of variables between different groups.

Given p continuous random variables $\{X_i, i = 1, 2, \dots, p\}$. from n samples, we can denote the set of measurements/data as

$$X = (X_1, X_2, \ldots\ldots, X_P)^T \in \mathbb{R}^{n\times p} \qquad \textit{Equation 1-1}$$

Here the rows of the matrix represent the samples and the columns the variables. Within each column (variable), the data are centered to the column mean. For any two random variables X_i and X_j, we denote the set of all other variables as $X_{-(i,j)}$, that is,

$$X_{-(i,j)} = X \setminus \{X_i, X_j\} = \{X_k, 1 \le k \ne i, j \le p\} \qquad \textit{Equation 1-2}$$

where X_i and $X_j \in \mathcal{R}^n$ are the ith and jth columns of X, and $X_{-(i,j)} \in \mathbb{R}^{n\times(p-2)}$ is the matrix obtained from X by deleting its ith and jth columns. Without loss of generality, we assume that $i < j$. When the sample size (n) is larger than the number of variables (p), the standard estimate of partial correlation coefficient of X_i and X_j while controlling the effects of variables in the remaining variable set of $X_{-(i,j)}$, can be calculated using three different methods.

The first method is through "matrix inversion" (Schafer and Strimmer 2005) and can be accomplished in a computation time of $\mathcal{O}(n^3)$. Denoting the covariance matrix of X as $\Sigma = (\sigma_{ij})_{p*p}$, it can be further decomposed into the variance components σ_i^2 and the Pearson correlation matrix $P = (r_{ij})_{p*p}$. Since the data are column-centered, the covariance matrix Σ is estimated as

$$\widehat{\Sigma'}_{p*p} = (X\text{-}EX)^T(X\text{-}EX) = X^T X \qquad \textit{Equation 1-3}$$

Here ^{T} denote the transpose of a matrix and $X^T X$ is the inner product of X itself, that is, the sum of squares of all elements in X. The standard unbiased estimate of Σ is then given by

$$\hat{\Sigma}_{p^*p} = \frac{1}{n\text{-}1}\hat{\Sigma'}_{p^*p} = \frac{1}{n\text{-}1}X^T X \qquad \textit{Equation 1-4}$$

In the setting of $n > p$, the above p-by-p matrix is symmetric and positive-semi definite. If $\hat{\Sigma}$ is invertible, we denote the precision (or concentration) matrix $\hat{\Omega}$ as the inverse of $\hat{\Sigma}$ such that

$$\hat{\Omega} = \left(\hat{\omega}_{ij}\right)_{p^*p} = \hat{\Sigma}^{\text{-}1}$$

Therefore, an unbiased estimate of the partial correlation coefficient of X_i and X_j giving $X_{\text{-}(i,j)}$ is given by

$$\hat{\rho}_{ij} = -\frac{\hat{\omega}_{ij}}{\sqrt{\hat{\omega}_{ii}\hat{\omega}_{jj}}} \qquad \textit{Equation 1-5}$$

Another simple way to compute partial correlation coefficient is through the least square regression. Consider the two linear regression models

$$X_i = X_{\text{-}(i,j)}\beta^{(i)} + \varepsilon_i = \sum_{k \neq i,j} \beta_k^{(i)} X_k + \varepsilon_i \qquad \textit{Equation 1-6}$$

$$X_j = X_{\text{-}(i,j)}\beta^{(j)} + \varepsilon_j = \sum_{k \neq i,j} \beta_k^{(j)} X_k + \varepsilon_j \qquad \textit{Equation 1-7}$$

where ε_i and ε_j are i.i.d. random noises. The intercept term is not included in neither model because all variables are centered. The least square estimates of $\beta_k^{(i)}$ and $\beta_k^{(j)}$ can be obtained as follows:

$$\hat{\beta}^{(i)} = \left(\hat{\beta}_1^{(i)}, \hat{\beta}_2^{(i)}, \ldots, \hat{\beta}_{i\text{-}1}^{(i)}, \hat{\beta}_{i+1}^{(i)}, \ldots, \hat{\beta}_{j\text{-}1}^{(i)}, \hat{\beta}_{j+1}^{(i)}, \ldots, \hat{\beta}_p^{(i)}\right) \qquad \textit{Equation 1-8}$$

$$= \arg \min_{\beta \in \mathcal{R}^{p-2}} \left\| X_i - X_{-(i,j)}\beta \right\|^2$$

$$\hat{\beta}^{(j)} = \left(\hat{\beta}_1^{(j)}, \hat{\beta}_2^{(j)}, \ldots, \hat{\beta}_{i-1}^{(j)}, \hat{\beta}_{i+1}^{(j)}, \ldots, \hat{\beta}_{j-1}^{(j)}, \hat{\beta}_{j+1}^{(j)}, \ldots, \hat{\beta}_p^{(j)}\right)$$

$$= \arg \min_{\beta \in \mathcal{R}^{p-2}} \left\| X_j - X_{-(i,j)}\beta \right\|^2 \qquad \textit{Equation 1-9}$$

$Here\ \|a\|_2^2 = \sum_j a_j^2$ is the L_2 norm, indicating the sum of squared elements of the matrix. The corresponding regression residuals are

$$\widehat{R_i} = X_i - X_{-(i,j)}\hat{\beta}^{(i)} = X_i - \sum_{k \neq i,j} \hat{\beta}_k^{(i)} X_k \qquad \textit{Equation 1-10}$$

$$\widehat{R_j} = X_j - X_{-(i,j)}\hat{\beta}^{(j)} = X_j - \sum_{k \neq i,j} \hat{\beta}_k^{(j)} X_k \qquad \textit{Equation 1-11}$$

The Pearson correlation between the residuals is a measurement of the strength of the relationship between X_i and X_j with the linear effect of $X_{-(i,j)}$ removed and can thus represent the partial correlation between the two variables of interest.

A third way to estimate the partial correlation coefficient also relates to the least square regression problem (Peng, Wang et al. 2009). Constructing p linear regression models

$$X_i = X_{-(i)}\beta^{(i)} + \varepsilon = \sum_{k \neq i} \beta_k^{(i)} X_k + \varepsilon, i = 1, 2, \ldots, p \qquad \textit{Equation 1-12}$$

where ε are i.i.d. disturbance terms, the least square estimate of the regression coefficient vector is calculated as

$$\hat{\beta}^{(i)} = \left(\hat{\beta}_1^{(i)}, \hat{\beta}_2^{(i)}, \ldots, \hat{\beta}_{i-1}^{(i)}, \hat{\beta}_{i+1}^{(i)}, \ldots, \hat{\beta}_p^{(i)}\right) = \arg\min_{\beta \in \mathcal{R}^{p-1}} \left\| X_i - X_{-(i)}\beta \right\|^2$$
$$= \left(X_{-(i)}^T X_{-(i)}\right)^{-1} X_{-(i)}^T X_i, \quad for\ i = 1, 2, \ldots, p \qquad \text{Equation 1-13}$$

The sample partial correlation coefficient is then estimated as

$$\hat{\rho}_{ij} = sign\left(\hat{\beta}_j^{(i)}\right)\sqrt{\hat{\beta}_j^{(i)}\hat{\beta}_i^{(j)}} \qquad \text{Equation 1-14}$$

Given $n > p$, the two coefficient vectors $\hat{\beta}_j^{(i)}$ and $\hat{\beta}_i^{(j)}$ always have the same sign and thus the term of square root in the above equation is well-defined. Based on the above formula, the process of searching for non-zero partial correlation coefficients is equivalent to the model selection problem under the regression setting.

To tackle the high-dimension-low-sample size problem, a new loss function was proposed by employing sparse regression techniques. The loss function contains two parts. One part represents the regression part (Equation 1-15).

$$L_n(\Theta, \Lambda, \mathbf{Y}) = \frac{1}{2}\left(\sum_{i=1}^{p} w_i \left\| Y_i - \sum_{j \neq i} \beta_{ij} Y_j \right\|^2\right) \qquad \text{Equation 1-15}$$

Where $\Theta = \left(\rho_{12}, \cdots, \rho_{(p-1)p}\right)^T$; $\Lambda = diag\left(\Sigma^{-1}\right) = \{\sigma^{ii}\}_{i=1}^{p}$; $\beta_{ij} = \rho_{ij}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}$; $\boldsymbol{Y} = \{\boldsymbol{Y^k}\}_{k=1}^{n}$; Lastly, $\boldsymbol{w}$ are nonnegative weights. The other part is the ℓ_1 penalty on partial correlations Θ (Equation 1-16),

$$\boldsymbol{J(\Theta) = \lambda\|\Theta\|_1 = \lambda \sum_{1 \le i < j \le p} \rho_{ij}} \qquad \text{Equation 1-16}$$

The partial correlations Θ is estimated by minimizing a penalized loss function

$$\boldsymbol{\mathcal{L}_n}((\Theta, \Lambda, \mathbf{Y}) = L_n(\Theta, \Lambda, \mathbf{Y}) + \boldsymbol{J}(\boldsymbol{\Theta}) \quad \textit{Equation 1-17}$$

An *active-shooting* algorithm was developed to solve the loss function efficiently

The distribution of the sample partial correlation for continuous variables was first studied by Fisher (Fisher 1924). Assuming the original data of all variables coming from a multivariate Gaussian distribution, it states that the random sampling distribution of a partial correlation coefficient controlling k variables, is exactly that of a total correlation coefficient with k fewer degrees of freedom. Thus, we can test the null hypothesis that the population partial correlation coefficient equals to zero via an F-test (Equation 1-18).

$$F = \frac{\hat{\rho}_{ij}^2}{1 - \hat{\rho}_{ij}^2} * (N\text{-}k\text{-}2) \sim F_{1,N\text{-}k\text{-}2} \quad \textit{Equation 1-18}$$

Here N is the sample size and k is the number of variables being controlled. Similarly, the Fisher's z-transformation can also be used:

$$Z = \frac{1}{2} \ln\left(\frac{1 + \hat{\rho}_{ij}}{1 - \hat{\rho}_{ij}}\right) \dot{\sim} N(0, 1) \quad \textit{Equation 1-19}$$

There is no exact test for the equality between two partial correlation coefficients. For data with sufficiently large sample size, some methods of approximation are known. Again the most widely used approach is Fisher's z-transformation (Fisher 1914). To compare two population partial correlation coefficients $\rho_{ij}^{(1)}$ and $\rho_{ij}^{(2)}$ conditioning on k variables, draw an independent sample from each of the population with sample size n_1 and n_2 respectively, and

calculate the sample partial correlation coefficients $\hat{\rho}_{ij}^{(1)}$ and $\hat{\rho}_{ij}^{(2)}$. The test statistics of the null hypothesis that the two population partial correlation coefficients $\rho_{ij}^{(1)}$ and $\rho_{ij}^{(2)}$ are equal is

$$Z = \frac{{}^{1}/_{2}\left(\ln\left(\frac{1+\hat{\rho}_{ij}^{(1)}}{1-\hat{\rho}_{ij}^{(1)}}\right) - \ln\left(\frac{1+\hat{\rho}_{ij}^{(2)}}{1-\hat{\rho}_{ij}^{(2)}}\right)\right)}{\sqrt{\frac{1}{n_1\text{-}k\text{-}3} + \frac{1}{n_2\text{-}k\text{-}3}}} \dot{\sim} N(0,1) \qquad \textit{Equation 1-20}$$

As noted previously, the bootstrap resampling method (Efron 1994) is a widely used non-parametric alternative for data failed to meet the normality assumption. To determine whether two partial correlation coefficients are equal, one could perform either of the following: (1) to generate a bootstrap confidence interval for each sample partial correlation coefficient and see whether the two confidence intervals overlap; or (2) to bootstrap the difference between two partial correlation coefficients and see whether the bootstrap confidence interval of the difference contains zero.

Pradhan (Pradhan 2009) proposed a two-level regression method to convert the test to that of a regression coefficient for the aim of comparing the strength of a partial correlation between two groups (for example, diseased and normal), in a network. The binary grouping covariate is denoted as $G = \{0,1\}$. In the first step, two residual terms (prediction errors) $\widehat{R_i}$ and $\widehat{R_j}$ are obtained via linear regressions of X_i and X_j on $X_{\text{-}(i,j)}$ respectively.

$$X_i = X_{\text{-}(i,j)}\beta^{(i)} + \varepsilon_i \qquad \textit{Equation 1-21}$$

$$X_i = X_{\text{-}(i,j)}\beta^{(i)} + \varepsilon_i \qquad \textit{Equation 1-22}$$

$$\widehat{R_\iota} = X_i - X_{-(i,j)}\hat{\beta}^{(i)} = X_i - \sum_{k \neq i,j} \hat{\beta}_k^{(i)} X_k \qquad \textit{Equation 1-23}$$

$$\widehat{R_\jmath} = X_j - X_{-(i,j)}\hat{\beta}^{(j)} = X_j - \sum_{k \neq i,j} \hat{\beta}_k^{(j)} X_k \qquad \textit{Equation 1-24}$$

The test of the Pearson correlation coefficient between two residuals features the same significance to that of the slope coefficient in the linear regression model of $\widehat{R_\iota}$ and $\widehat{R_\jmath}$.

$$\widehat{R_\iota} = a_0 + a_1\widehat{R_\jmath} + \varepsilon \qquad \textit{Equation 1-25}$$

Or equivalently,

$$\widehat{R_\jmath} = c_0 + c_1\widehat{R_\iota} + \tau \qquad \textit{Equation 1-26}$$

Integrating the covariate G into the above regression models brings about the second stage models:

$$\widehat{R_\iota} = a_0 + a_1\widehat{R_\jmath} + \varepsilon = a_0 + (b_0 + b_1 G)\,\widehat{R_\jmath} + \varepsilon = a' + b_0\widehat{R_\jmath} + b_1 G\widehat{R_\jmath} + \varepsilon$$

$$\widehat{R_\jmath} = c_0 + c_1\widehat{R_\iota} + \tau = c_0 + (d_0 + d_1 G)\,\widehat{R_\iota} + \tau = c' + d_0\widehat{R_\iota} + d_1 G\widehat{R_\iota} + \tau$$

Therefore, the significance of coefficients b_1 and d_1, (that is, the average p-value from tests of significance for b_1 and d_1) represents the significance of the covariate effect G on the partial correlation coefficient between X_i and X_j controlling for $X_{-(i,j)}$.

In 2008, Friedman, Hastie, and Tibshirani proposed an L1-penalized precision matrix estimation, the graphical Lasso, for learning partial correlation structures for a multivariate normal distribution. A different but yet equivalent perspective is on learning a precision matrix

through the estimation of a set of joint sparse linear regression models, as described in (Peng et al. 2009) mentioned previously in this work. The R-code based on their approach, SPACE, has been applied to our dataset and compared to other correlational network methods, and subsequently integrated with, the sparse canonical correlation method (sCCA) introduced below.

2.2 Sparse Canonical Correlation Analysis

Introduced by Hotelling in 1936, (the first) canonical correlation between two variable sets looks for the weighted combination of all variables within each variable set such that the correlation of the two combinations is maximized. The weighted combinations are called canonical variables or components. Considering an matrix $X \in \mathbb{R}^{n\times p}$ and an matrix $Y \in \mathbb{R}^{n\times q}$. Without loss of generality, we assume $p < q$. Canonical correlation analysis (CCA) (Hotelling 1936) seeks coefficient vectors $\boldsymbol{u} \in \mathbb{R}^{p\times 1}$ and $\boldsymbol{v} \in \mathbb{R}^{q\times 1}$, such that the correlation between the linear combinations $\omega = X\boldsymbol{u}$ and $\xi = Y\boldsymbol{v}$ is maximized, i.e.

$$\max_{\boldsymbol{u},\boldsymbol{v}} Corr(\omega, \xi) = \max_{\boldsymbol{u},\boldsymbol{v}} \frac{\boldsymbol{u}^{'}\Sigma_{XY}\boldsymbol{v}}{\sqrt{\boldsymbol{u}^{'}\Sigma_{XX}\boldsymbol{u}}\sqrt{\boldsymbol{v}^{'}\Sigma_{YY}\boldsymbol{v}}}$$

where Σ_{XX}, Σ_{YY}, and Σ_{XY} are the variance for X, Y, and the covariance for X and Y, respectively. It is attained by the canonical variate pairs

$$\omega = X\boldsymbol{u} = e^{'}\Sigma_{XX}^{-\frac{1}{2}}X; \qquad \xi = Y\boldsymbol{v} = f^{'}\Sigma_{YY}^{-\frac{1}{2}}Y$$

with e and f from the singular value decomposition (SVD) of a matrix K given by

$$K = \Sigma_{XX}^{-\frac{1}{2}}\Sigma_{XY}\Sigma_{YY}^{-\frac{1}{2}} = eDf^{'}$$ (Parkhomenko, Tritchler et al. 2007).

In canonical correlation analysis, all variables are included in the linear combinations, yet for genetic data obtained via microarray studies or other high throughput methods, the number of variables usually surpasses tens of thousands, exceeding the number of study subjects. Thus the fitted linear combinations may not be easily interpreted and the application of standard algorithms may fail. These problems may be solved by introducing sparse loadings in the canonical components. Motivated by this idea, sparse canonical correlation analysis (SCCA) has been firstly proposed in 2007 (Parkhomenko, Tritchler et al. 2007) and has been extended and widely applied in the genetic area. The idea of SCCA in the field of genetics is consistent with the belief that only a small section of genes is expressed under a certain condition.

Based on the foundation of SCCA, Witten and Tibshirani (Witten and Tibshirani 2009) have further presented "sparse supervised canonical correlation analysis (sparse sCCA)", targeting on finding the sparse linear combinations of the two variable sets that are correlated with each other and also associated with the trait of interest. Still an matrix $X \in \mathbb{R}^{n\times p}$ and an matrix $Y \in \mathbb{R}^{n\times q}$, and assuming that the columns of X and Y have been standardized with mean 0 and standard deviation 1. Suppose in addition we have a categorical outcome vector $z \in \mathbb{R}^n$. The estimates of canonical vectors are defined as

$$\max_{u,v} u^T X^T Y v, \text{ subject to}$$

$$\|u\|^2 \le 1, \|v\|^2 \le 1, P_1(u) = \|u\|_1 \le c_u, P_2(v) = \|v\|_1 \le c_v,$$

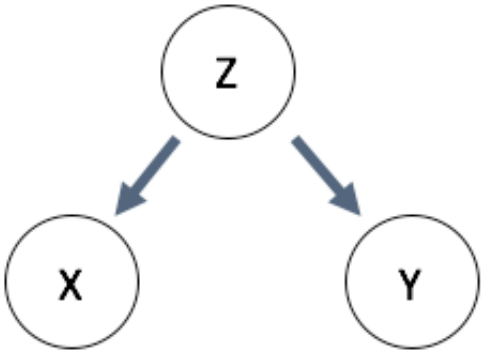
$$u_j = 0 \; \forall \, j \notin Q_u, v_j = 0 \; \forall \, j \notin Q_v,$$

where P_1 and P_2 are convex penalty functions; c_u and c_v are assumed to be $1 \leq c_u \leq \sqrt{p}$ and $1 \leq c_v \leq \sqrt{q}$; Q_u and Q_v are the sets of variables with highest univariate association with the outcome z in X and Y, respectively; the threshold for variables to be included in Q_u and Q_v can either be fixed or be defined as a tuning parameter. The vectors u and v are obtained using an iterative algorithm with soft-thresholding. We have performed this sparse sSCCA method on our genetic data set to investigate whether the expression of miRNA would have a significant effect on that of genes and vice versa.

2.3 Sparse Bayesian Network Analysis

The fundamental structure among a series of random variables is depicted by their joint probability distribution. Probabilistic graphical models are used to describe the conditional independence or dependence structure implied by the joint distribution with a graph-induced decomposition of the joint density function. A Bayesian Network (BN), a branch of probabilistic graphical model, is a probabilistic graphical model defined over a DAG G with a set of $p = |V|$ nodes $V = \{v_1, \cdots, v_2\}$. In such a graph or network, a node is a random variable, and an edge between two nodes indicates certain stochastic association. The probability model associated with G in a Bayesian network factorizes as $p(X_1, \cdots, X_p) = \prod_{j=1}^{p} p(X_j | Pa(X_j))$, where $p(X_j | Pa(X_j))$ is the conditional probability distribution for X_j given its parents $Pa(X_j)$ with directed edges from each node in $Pa(X_j)$ to X_j in G. For Gaussian random variables, this is equivalent to a zero partial correlation: $\rho_{XY.Z} = 0$, as illustrated in Figure 2. This provides certain insight into the relationship between the Bayesian network and the partial correlation network in

that, the partial correlation, by controlling all other variables except the two targeting variables, should in general be more conservative than the Bayesian network.



(C) Regulated by a third variable

Figure 2 Conditional independence in a Bayesian network

A recently published paper (Jing Xiang 2013) presented an algorithm entitled A* lasso, for learning a Sparse Bayesian Network structure for continuous variables in a high-dimensional space. Compared to the common two-stage inference methods, A* lasso is a single stage method that recovers the optimal sparse Bayesian network structure by solving a single optimization problem with A* search algorithm that uses lasso in its scoring system. The A*lasso method assumes continuous random variables and uses a linear regression model for the conditional probability distribution of each node $X_j = Pa(X_j)^*\beta_j + \epsilon$, where $\beta_j = \{\beta_{jk}^{'} s for\ X_k \in Pa(X_j)\}$ is the vector of unknown parameters to be estimated from data and ϵ is the noise distributed as $N(0,1)$. The BN's structure and parameters are obtained by minimizing the negative log likelihood of data with sparsity enforcing L_1 penalty (Equation 1-27).

$$\min_{\beta_1,\cdots,\beta_p} \sum_{j=1}^{p} \left\| x_j - x_{-j}^{'} \beta_j \right\|_2^2 + \lambda \sum_{j=1}^{p} \left\| \beta_j \right\|_1 \; s.t. G \in DAG. \qquad \textit{Equation 1-27}$$

Here X_{-j} represents all columns of $\boldsymbol{X}$ excluding x_j, assuming all other variables are candidate parents of node v_j. This lasso optimization problem can be solved efficiently with the shooting algorithm (Fu 1998) if the acyclicity constraint is ignored, which is the most challenge part of the BN inference procedure. A heuristic scheme of A* lasso is proposed to prune search space when learning the Bayesian network structure by exploring a scoring algorithm based on lasso score generated by the shooting algorithm (Fu 1998).

$$f(Q_s) = g(Q_s) + h(Q_s) \qquad \textit{Equation 1-28}$$

Here Q_s is the set of variables for which the ordering has been determined. And $g(Q_s)$ is the accumulated cost for reaching the Q_s state (Equation 1-29).

$$g(Q_s) = \sum_{v_j \in Q_s} LassoScore(v_j | \prod_{<v_j}^{Q_s} h(Q_s)) \qquad \textit{Equation 1-29}$$

$Here\ h(Q_s)$ is the estimated cost of reaching the goal stat from the current state (Equation 1-30).

$$g(Q_s) = \sum_{v_j \in V \setminus Q_s} LassoScore(v_j | V \setminus v_j) \qquad \textit{Equation 1-30}$$

Furthermore, the Lasso Score is defined in Equation 1-31

$$LassoScore(v_j | V \setminus v_j) = \min_{\beta_j} \left\| x_j - x_{-j}^{'} \beta_j \right\|_2^2 + \lambda \sum_{j=1}^{p} \left\| \beta_j \right\|_1 \qquad \textit{Equation 1-31}$$

On top of the heuristic scheme, A* lasso further reduces the search space by limiting the size of intermediate search path via a size-limited priority queue that orders the promising intermediate search paths via the above scoring scheme. The combined strategy gives the A* lasso great advantage in efficiency over the common DP algorithms, which makes it scalable for high-dimension data, such as the miRNA and mRNA interaction problem in our study. This is why we have applied the A* lasso algorithm to our dataset and compared it with the sCCA and the SPACE (see result please).

2.4 Overview of the Pipeline

We proposed a novel pipeline for extracting miRNA and mRNA interaction network by combining the sCCA and the SPACE methods. Our pipeline is designed for small/moderate sample size with large number of miRNAs and mRNAs. In order to extract meaningful insights from small/moderate datasets, the pipeline selects most relevant miRNAs and mRNAs that has the largest canonical correlation via sCCA and then finds the links between these selected miRNAs and mRNAs through the SPACE method, where the latter would compute the pair-wise partial correlation coefficient conditioned on other features.

There are four steps in the pipeline. First, the differentially expressed (DE) miRNAs and mRNAs are selected via commonly used packages, such as limma (Altschul, Gish et al. 1990) or SAM (Chu, Li et al. 2001). Second, a subset of miRNAs and mRNAs are selected by performed the sparse sCCA method on the pooled DE miRNAs and mRNAs. In the third step, the pair-wise partial correlations are calculated by performing SPACE on the pooled DE miRNAs and mRNAs. Lastly, only the links that connects the selected miRNAs and mRNAs by sCCA are kept and added to the sCCA result.

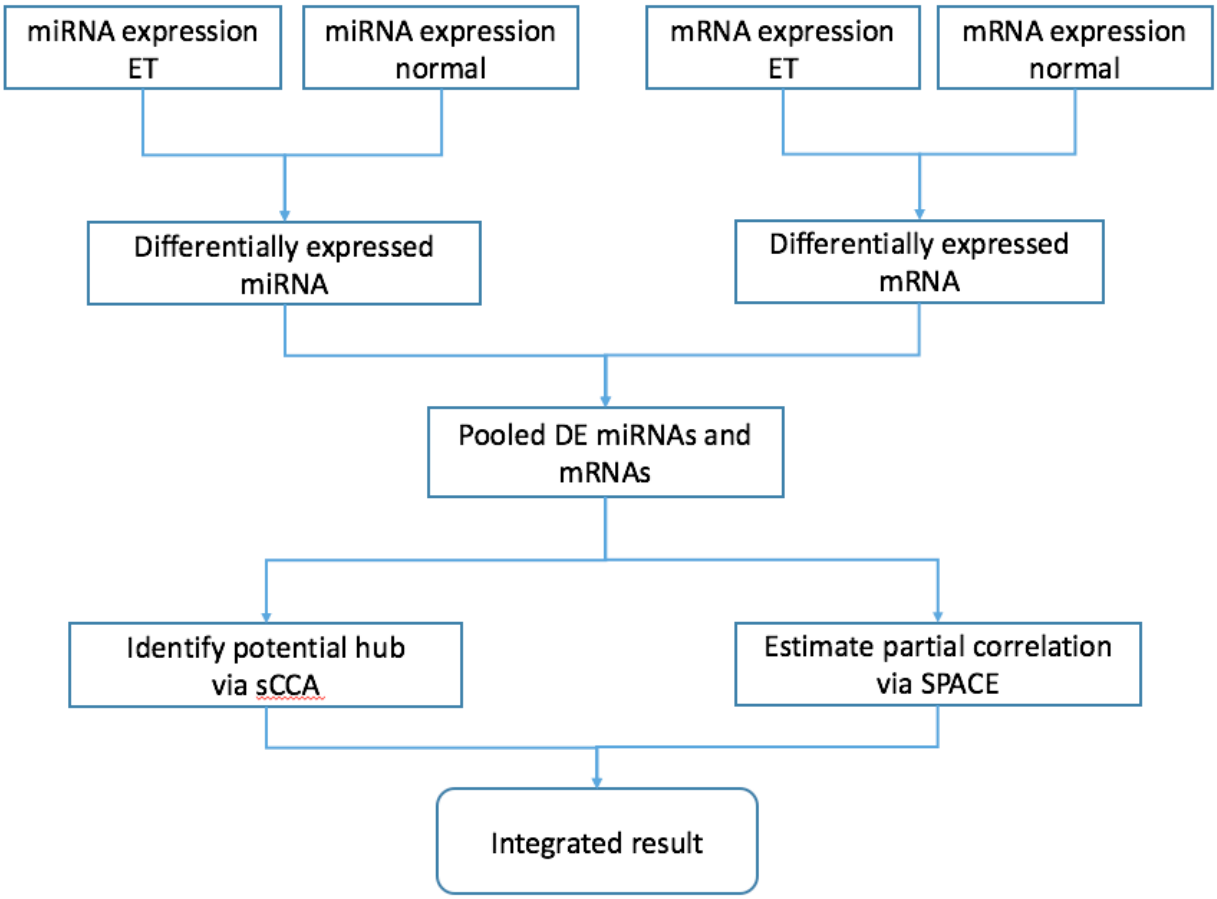


Figure 3 Pipeline for extracting the data-based miRNA and mRNA interaction network

3 Data analysis and result

3.1 Data processing

Our data included two data sets: 354 platelet-specific mRNA data from the custom array and 939 miRNA data from the Agilent microarray (Santa Clara, CA), which are paired with each other from 13 patients with essential thrombocytosis (ET) disease and 30 control subjects (**Error! Reference source not found.**).

Table 1 Data structure

	ET	Control	Total
# of Paired Subjects	13	30	43
	miRNA	**mRNA**	**Subject**
Total # of Items	939	354	43

Data were preprocessed and analyzed using R 3.2.2 with the Bioconductor packages (http://www.bioconductor.org/). Figure 4 presents the flow chart of the entire data processing process, before the ensuing network analysis.

There are 8 (3 ET, 4 NO, 1 RT) samples with technical replicates. The values of these samples are reset by the mean value of the sample replicates. The original miRNA data set was filtered by two steps: The first step is to filter out miRNAs with less than 30% non-absent cells in both groups. Next, miRNAs with more than 40% missing values in the sample sets were also dropped out. For the mRNA data, the proportion of missing expression data in the sample set for each mRNA was calculated and those with 50% or more absent data have been excluded. In addition, potential outliers were checked and filtered with a criterion of 3 standard deviations from the mean expression value. In both data sets, quantile normalization was applied to correct between-array variation (Pradervand, Weber et al. 2009), followed by the K-nearest neighbors algorithm for imputing missing expression data.

After data filtering and processing, there are totally 327 platelet-specific mRNAs and 396 miRNAs left. Linear models for microarray data (limma)(Smyth 2005) was used to picked up

DE miRNAs and mRNAS. A total of 61 miRNAs and 19 mRNAs are selected at the significant level 0.01.

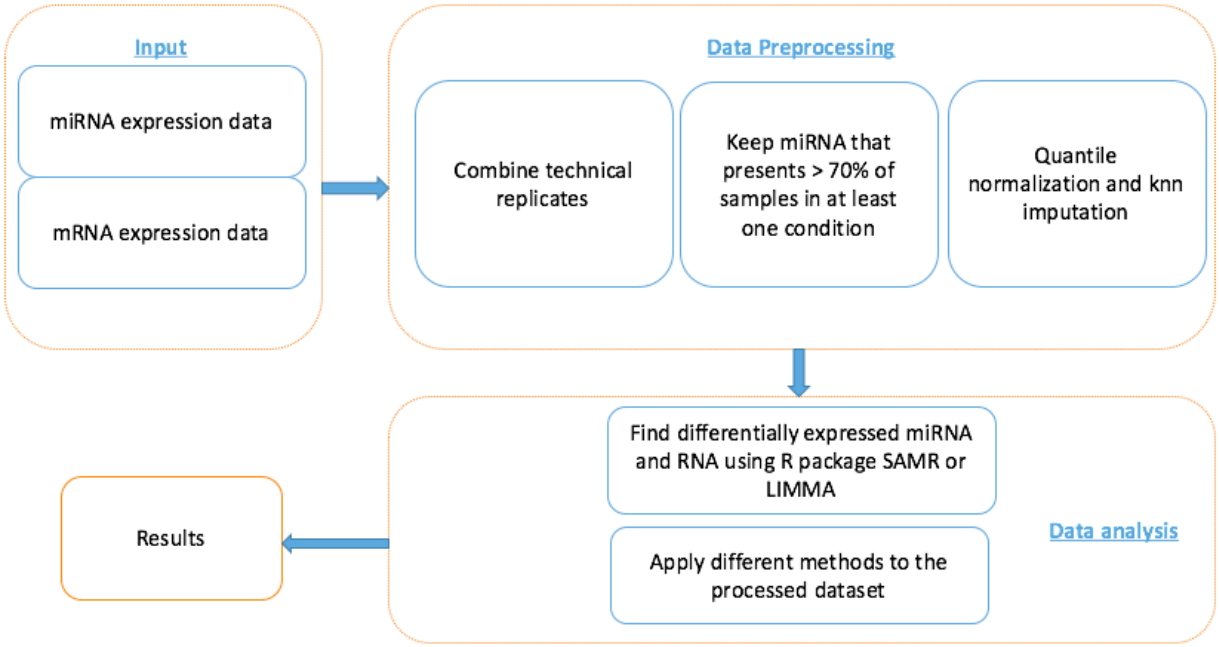


Figure 4 Flow chart of data processing before the network analysis

3.2 Result

With the 61 selected miRNAs as one variable set, the 19 mRNAs as the other, and the vector of subject disease status as a binary outcome vector, we applied three methods (sCCA, SPACE and A* lasso) to the differentially expressed data sets (miRNA and mRNA).

On the Pearson correlation analysis, the pair-wise Pearson correlation coefficient is calculated using psych package and 3200 non-zero coefficients are identified. It covers all links from the results of SPACE and A* lasso, which indicates Pearson correlation generates much more false positives than other methods. Therefore, we decided focus on the results of other three methods.

On the sSCCA method, the miRNA and mRNA subsets are selected with the penalty 0.3 on vector u and 0.5 on vector v. As discussed previously, vector u restricts the number of selected miRNA, while vector v does the same to the mRNA. In the result, 8 miRNAs stand out with 9 corresponding mRNAs (Table 2).

Table 2 Selected miRNAs and mRNAs by sSCCA

miRNA	**weight**	**mRNA**	**weight**
Has-miR-9	0.609	WASF1	0.797
Has-miR-182	-0.537	TIMP1	0.442
Has-miR-490-3p	0.439	CAV2	0.244
Has-miR-490-5p	0.289	HSD17B12	0.232
Has-miR-196b	-0.171	NME4	0.157
Has-miR-34a	0.165	ERVH-4	-0.154
Has-miR-34b*	0.089	MMP1	0.066
Has-miR-181a-2*	-0.045	LAPTM4B	0.049
		PKIG	0.038

Figure 5 visualizes the weights in the loadings of the first canonical coefficient of selected miRNAs and mRNAs. Here red or green node represents positive or negative weight in vector u and v. The node size represents the absolute value of weight.

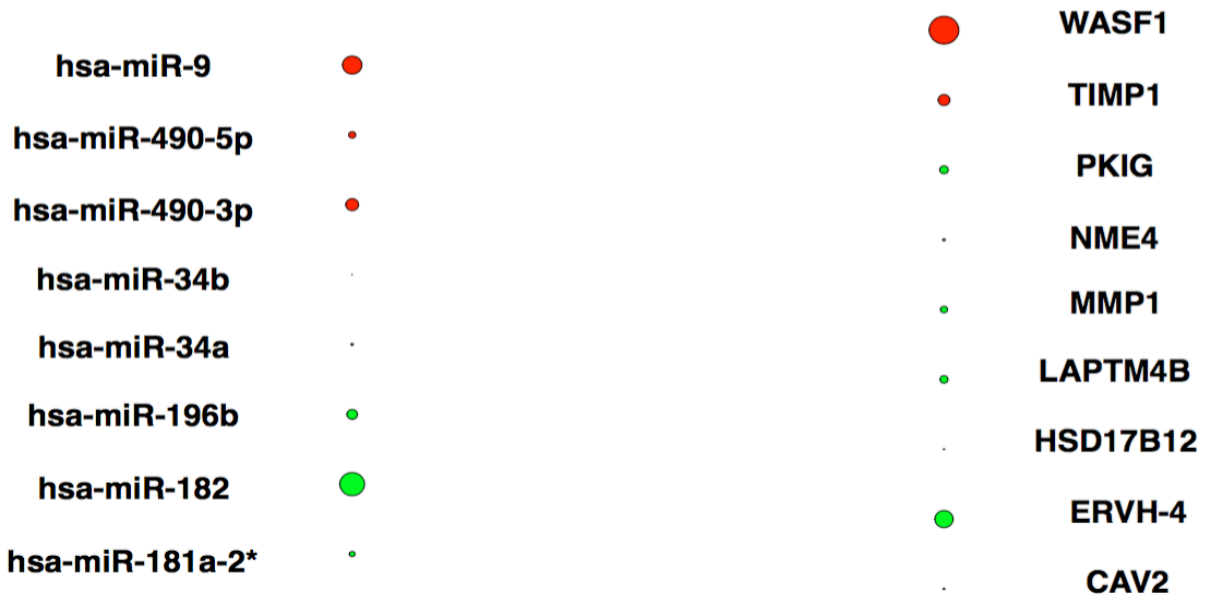


Figure 5 Bipartite plot of the sSCCA result

SPACE is a sparse method, which has one tuning parameter that controls the L_1 penalty on Lasso regression. The value is set as 0.5765849 as calculated by Equation 1-32.

$$L_1 = \frac{\Phi(1 - \frac{\alpha}{2 * p^2})}{\sqrt{n}} \qquad \textit{Equation 1-32}$$

Here n is the sample size (43), p is the number of features (80), and α is a constant (1).

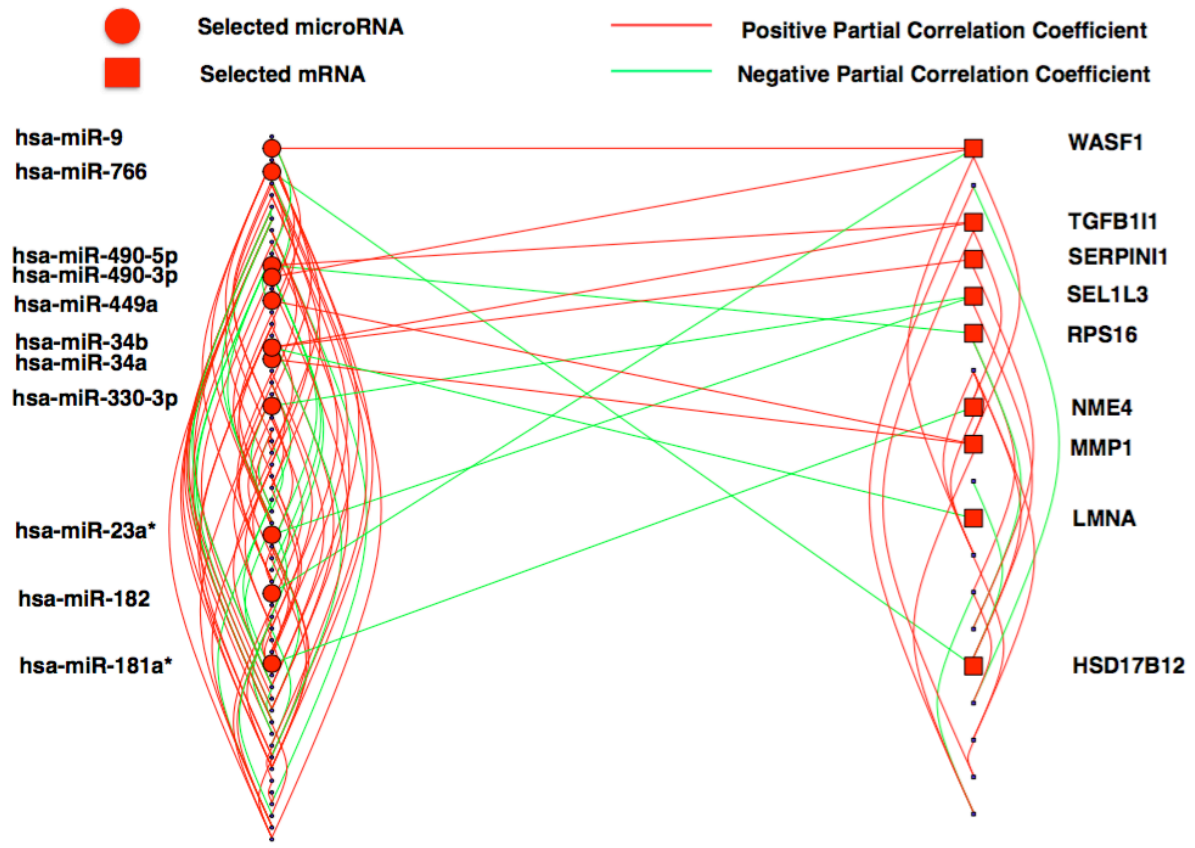


Figure 6 Bipartite plot of the SPACE result

Here red circles represent miRNAs that have direct connection with the mRNAs, while the red squares denote the mRNAs that have direct link with the miRNAs.In addition, red and green lines represent positive or negative partial correlations between the pairs.

Figure 6 illustrates the SPACE interaction network emphasizing the interactions between miRNAs and mRNAs. Those miRNAs and mRNAs that have direct link with the other side are labeled. The network is connected and there is no isolated node. Within group links accounts for most of the edges of the network and suggests that interaction within group is more common than that between groups. There are only 14 (14/165) direct links between miRNAs and mRNAs.

On the result from the A* lasso algorithm, there are two critical parameters. One is the L_1 penalty on Lasso regression. We choose 0.2 (recommended value) as the L_1 value. The other parameter is the queue size that limits the search depth. In order to obtain close-to optimal structure, 3000 is chosen for this option. Since all mRNAs have direct link with miRNAs, the

names are not listed in the figure (Figure 7) , A* lasso shows the same pattern with SPACE result, that is, within group interaction is more common than the between group interaction.

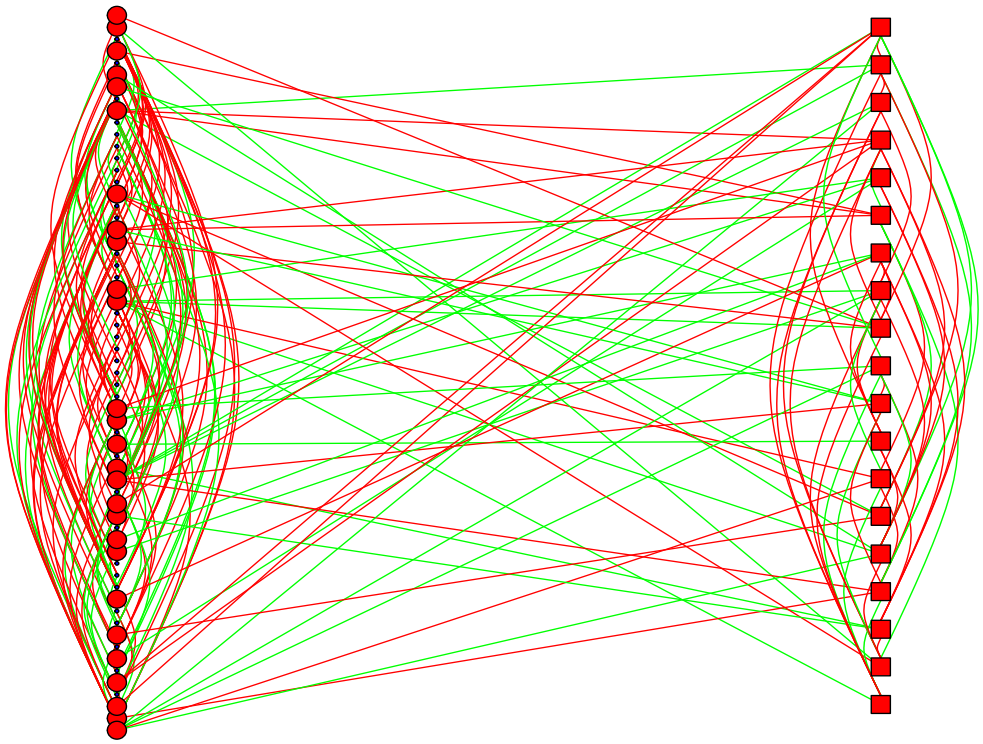


Figure 7 Network generated by A lasso*

As discussed in the method section, A* lasso identified 306 links that covers 192 out of 250 links from SPACE result, which is consistent with our expectation that SPACE should be more conservative than A* lasso considering the methodology differences. Since it is very hard to interpret a network with too many links and nodes, we integrate the SPACE and A* lasso result with result from sCCA by only keeping the selected miRNAs, mRNAs and the corresponding links from SPACE and A* lasso method (Figure 8) respectively. Figure 8 compares

the integrated results using SPACE and A* Lasso method with sSCCA. The interaction within the selected mRNAs are strikingly consistent both on links and the value signs except A* Lasso has more links. Two miRNA and mRNA interactions are overlapped. One is the link between has-miR-182 and WASF1. The other is the link between has-mir-34a and MMP1 gene (miRNA).

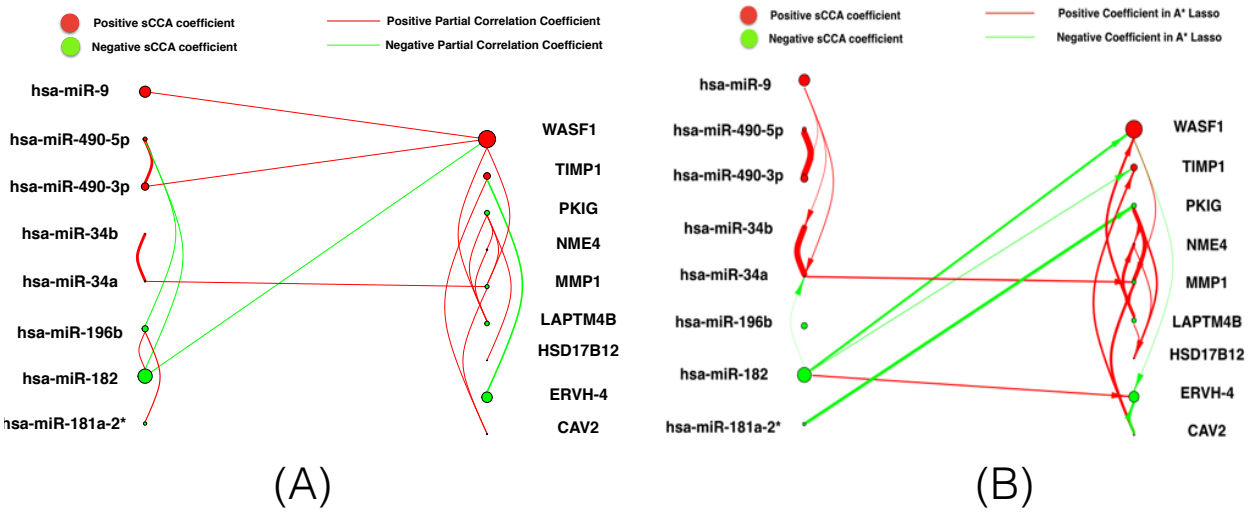


Figure 8 Integrated results

(A) is the integrated result between sSCCA and SPACE. (B) is the integrated result between sSCCA and A Lasso method. The arrow is added back on figure (B)*

To render the results more comprehensive, the expression value of those selected miRNAs and mRNAs are presented by Table 3 and Table 4. All selected miRNAs and mRNAs are differentially expressed with very small p-values.

Table 3 Quantile normalized expression of selected miRNAs

miRNAs	Normal		ET		Adjusted p-value
	mean	std	mean	std	
Has-miR-9	2.6443	0.7643	4.8766	1.6094	<0.0001
hsa-miR-490-5p	1.2040	1.0495	4.7451	2.3452	<0.0001
hsa-miR-182	3.8207	0.7794	2.0378	1.3189	<0.0001
hsa-miR-34a	6.0485	0.7324	8.2727	0.9558	<0.0001
hsa-miR-490-3p	1.1563	1.1951	5.3272	2.3615	<0.0001
hsa-miR-196b	6.1403	0.3886	5.2337	0.5812	<0.0001
hsa-miR-34b*	2.1224	0.9340	4.5740	0.9960	<0.0001
hsa-miR-181a-2*	5.5544	0.4086	4.4720	0.7544	<0.0001

Table 4 Quantile normalized expression of selected mRNAs

Genes (miRNA)	Normal		ET		Adjusted p-value
	mean	std	mean	std	
CAV2	1.2077	0.7195	2.1356	0.7285	<0.0001
LAPTM4B	0.7808	0.3000	1.7317	1.2278	<0.0001
TIMP1	0.8762	0.7070	2.6453	1.5047	<0.0001
PKIG	0.7111	0.2330	1.4218	0.3989	<0.0001
WASF1	0.7378	0.2937	2.8658	1.7532	<0.0001
MMP1	0.9536	0.8535	2.3396	1.4260	<0.0001
ERVH-4	1.3802	0.7963	0.6395	0.4932	<0.0001
NME4	0.7495	0.3800	1.8671	1.3119	<0.0001
HSD17B12	0.8473	0.6015	2.2129	0.9899	<0.0001

3.3 Comparison between SPACE and A* Lasso

Since results between SPACE and A* Lasso have significant overlap, this section will explore the overlapped part in more details. There are 108 out of 192 shared links that have the same sign in both results (Figure 9). Among the 108 links, there are 7 links with the direct interactions between miRNA and mRNA. Although all links could not be supported by previous studies, most of miRNA and RNAs highlighted are reported to be relevant to ET disease. For example, two encoded protease/protease inhibitors – *MMP1* (Matrix Metallopeptidase 1) and *SERPINI1* (Serpin Peptidase Inhibitor, Clade I (Neuroserpin), Member 1) are a class of proteins well-associated in tumor invasiveness and cancer metastases and have both been detected as over-expressed in ET group comparing to normal (Gnatenko, Cupit et al. 2005, Saito and Bunnett 2005). *MMP1* has been demonstrated to be related to inflammation in several studies (Brassart, Fuchs et al. 2001, Herouy, Mellios et al. 2001, Zhang, Cai et al. 2003, Andonovska, Dimova et al. 2008). Moreover, members in the matrix Metallopeptidase family have been indicated to be involved in the migration and invasion of leukemia cell (*MMP-2*) (He, Cao et al. 2009); and to mediate megakaryocyte transendothelial migration and proplatelet formation (*MMP-9*) (Lane, Dias et al. 2000).

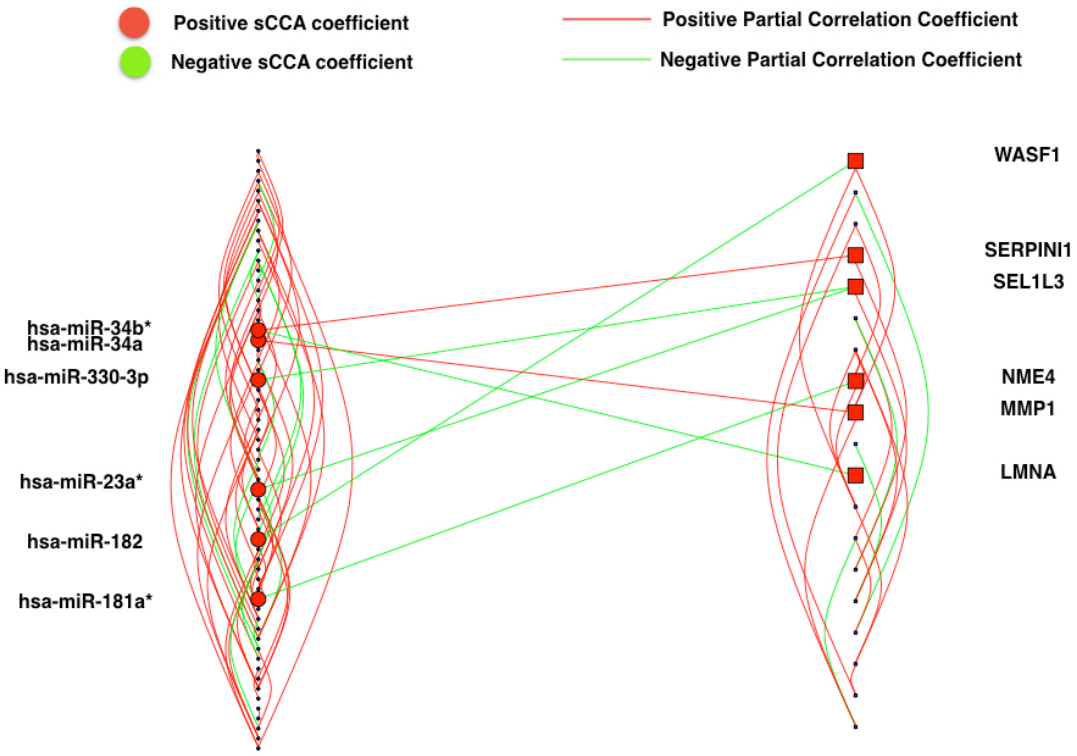


Figure 9 Overlaped links between SPACE and A lasso*

4 Discussion and Future work

4.1 Discussion

As shown in Table 3, the miRNA *hsa-miR-34a and has-miR-182* identified by sparse sSCCA expresses has been previously shown to express aberrantly in polycythemia vera (PV) granulocytes (Bruchova, Merkerova et al. 2008) and is one of the miRNA members that expressed most differentially among the ET, RT, and control groups (Xu, et al. 2012).

The platelet-expressed gene *HSD17B12*, standing for "Hydroxysteroid (17-β) Dehydrogenase 12", has been claimed previously to be associated with the distinction of ET platelets from the normal ones (Gnatenko, Cupit et al. 2005). The encoded protease/protease inhibitors – *MMP1* (Matrix Metallopeptidase 1) is a class of proteins well-associated in tumor invasiveness and cancer metastases and have both been detected over-expressed in the ET group comparing to the normal (Gnatenko, Cupit et al. 2005, Saito and Bunnett 2005). *MMP1* has been demonstrated to be related to inflammation in several studies (Brassart, Fuchs et al. 2001, Herouy, Mellios et al. 2001, Zhang, Cai et al. 2003, Andonovska, Dimova et al. 2008). Moreover, members in the matrix Metallopeptidase family have been indicated to be involved in the migration and invasion of leukemia cell (*MMP-2*) (He, Cao et al. 2009); and to mediate megakaryocyte transendothelial migration and proplatelet formation (*MMP-9*) (Lane, Dias et al. 2000).

CAV2 (Caveolin 2 and *NME4* (NME/NM23 Nucleoside Diphosphate Kinase 4) have been inferred to associate with tumors, metastasis, and multiple types of cancer. *TIMP1* (TIMP Metallopeptidase Inhibitor 1) has been found to be highly related to tumors, cancer metastasis

and inflammation. *WASF1* (WAS Protein Family, Member 1) has been predicted as a potential target of hsa-miR-34a by a popular miRNA target prediction tools, TargetScan (http://www.targetscan.org/), which predicts regulatory targets using conserved complementary (Lewis, Burge et al. 2005). Moreover, the WAS protein family has been shown to be related to nucleosome and chromatin assembly, performing an important role in gene transcription that may regulate megakaryocytopoiesis and/or proplatelet formation (Schulze and Shivdasani 2004). A recent study in class prediction models of ET included a member from this family, *WASF3*, as one of the biomarkers segregating ET, RT and the normal groups (Gnatenko, Zhu et al. 2010). Although instead of *WASF1*, the study pointed to *WASF3*, our result would implicate a specific role *WASF1* plays in the classification and prediction models for ET and normal cohorts.

In conclusion, our data analysis on miRNA and mRNA data has predicted a close relationship between 8 miRNAs (including miR-9, miR-490-5p, miR-490-3p, miR-182, miR-34a, miR-196b, miR-34b*, miR-181a-2*) and a 9-mRNA set (including CAV2, LAPTM4B, TIMP1, PKIG, WASF1, MMP1, ERVH-4, NME4, HSD17B12). A majority of the identified variables have been linked to hematologic function by a sizable number of studies. Furthermore, it is observed that the selected mRNAs are high relevant to ET disease. On the other side, the networks identified by SPACE and A* lass method has a significant overlap, which suggests that the two methods give comparable results even they use different methodology to identify the network. To be specific, the two overlapped links (miR-182--WAFS1 and miR-34a--MMP1) are highly worthy targets for biological validation since all four mRNAs/mRNAs involved have been shown to be related with the ET disease by various studies. All together it alludes that the identified mRNA set might be considered as a contributor in the regulatory mechanism of the ET disease -- with the 8 selected miRNAs playing a modulating role on that mRNA set. We also

used Ingenuity Pathway Analysis (IPA) software to find the confirmed links between 8-miRNAs and 9-mRNAs. IPA predicts that miR-9 and miR-196b have interaction with NME4, which is related with several pathways, such as Salvage Pathways of Pyrimidine Ribonucleotides, Pyrimidine Ribonucleotides De Novo Biosynthesis and Pyrimidine Ribonucleotides Interconversion and Pyrimidine Deoxyribonucleotides De Novo Biosynthesis I. It also links hsa-miR-34a/has-miR-34b* with WASF1 and relates these two links with multiple pathways including Actin Cytoskeleton Signaling, Actin Nucleation by ARP-WASP Complex, Epithelial Adherens Junction Signaling, Rac Signaling, Regulation of Actin-based Motility by Rho, RhoA Signaling,RhoGDI Signaling and Signaling by Rho Family GTPases.

4.2 Future work

Part II an ultrafast clustering algorithm to count barcode and amplicon reads.

1 Introduction

1.1 Next generation sequencing technology

Next Generation Sequencing (NGS) Technology provides an inexpensive, high resolution, genome-wide sequence readout as an endpoint to applications ranging from chromatin immunoprecipitation, mutation mapping and polymorphism discovery to noncoding RNA discovery(Mardis 2008). With great advantages over the old sequencing methods, NGS now becomes the main stream method that is used to extract the genetic information in most of biology fields, where it provides the ability to answer the questions in unimaginable speed. The Table 5 is a brief comparison between NGS and traditional sequence platform Sanger sequencing.

Table 5 Gains and pains of NGS

GAINS	PAINS
NGS provides a much cheaper and higher throughput alternative to sequencing DNA than traditional Sanger sequencing. Whole small genomes can now be sequenced in a day.	NGS, although much less costly in time and money in comparison to first-generation sequencing, is still too expensive for many labs. NGS platforms can cost more than $ 100,000 in start-up costs, and individual sequencing reactions can cost upward of $ 1,000 per genome.
High-throughput sequencing of the human genome facilitates the discovery of genes and regulatory elements associated with disease	Inaccurate sequencing of homopolymer regions (spans of repeating nucleotides) on certain NGS platforms, including the Ion Torrent PGM, and short-sequencing read lengths (on average 200 500 nucleotides) can

	lead to sequence errors.
Targeted sequencing allows the identification of disease-causing mutations for diagnosis of pathological conditions.	Data analysis can be time-consuming and may require special knowledge of bioinformatics to garner accurate information from sequence data.

Error rates of individual NGS reads are higher than Sanger sequencing. To addressing the accuracy issue, redundancy of sequence coverage is the primary method to enhance the reliability of down-stream analysis on NGS data. The presence of multiple reads in the same location is used to confirm the accuracy of the base calls in the applications where accuracy is critical. NGS also provides the quality values for each sequenced base pair. The quality values calculated during NGS base calling provide important information for alignment, assembly, and variant analysis. Although the calculation of quality varies between platforms, the calculations are all related to the historically relevant phred score, introduced in 1998 for Sanger sequence data. The phred score quality value, q, uses a mathematical scale to convert the estimated probability of an incorrect call, e, to a log scale: $q = -10 * \log(e)$ Miscall probabilities of 0.1(10%),0.01(1%) and 0.001(0.1%) yield phred scores 10, 20 and 30. Quality values an important tool for rejecting low quality reads, trimming low quality bases, improving alignment accuracy, and determining consensus sequence and variation calls (Li, Ruan et al. 2008).

1.2 High resolution lineage track with random barcode

This section will briefly introduce the basic concepts, the development of this area and latest experimental approach.

Lineage tracking refers the idea of tracking the descendants of a small number of founders and was first used in genealogy, which has been studied for centuries. Lineage tracking was first studied in mathematical way by a problem proposed by Francis Galton. The problem came from the concern that aristocratic surnames were becoming extinct. Francis Galton formulized this problem in mathematical way and published it in Educational Times. Rev. H. W. Watson answered this problem with a solution and they started to investigate this problem together later on. This story ended up with a new area named branching process and their work still forms the basis of this area nowadays.

Lineage tracking was introduced to biology area in 19th centuries and was widely used to study the development. More recently, lineage track was adapted to study experimental evolution dynamics using fluorescent lineage tagging(Hegreness, Shoresh et al. 2006). However, fluorescent lineage tagging does not have enough resolution to handle large population and could not be applied to evolution dynamics of large populations. One approach to increase the number of unique lineages is to use genetic markers and track their relative abundance via NGS. Since NGS is inexpensive and high throughput, sequencing-based lineage tracking is a platform that is cheap, scalable, and, in theory, provides an unlimited number of unique lineage tags(Blundell and Levy 2014). To generate large number of lineage tags in sequence based lineage tracking, the short engineered DNA sequence called DNA barcode, is the key component. A DNA barcode is a stretch of bases at a known location in the genome or on a plasmid, whose relative frequency within a population can be measured by microarrays, and more recently quantitative sequencing. If each nucleotide position in DNA barcode is generated pure randomly, then the possible combination of a fixed length DNA barcode could easily reach to millions (e.g. a 26mer random barcode could have about 10^{16} possible combinations) and make it possible to keep track large

populations simultaneously. In the early stage, constructing DNA barcode is slow and labor intensive. Then a random DNA barcode was developed to and is generated by a lentivirus insertion system (Robinson, Chen et al. 2014), which is much efficient DNA barcode construction system. Although construction random DNA barcode is easy, a high quality barcode library usually takes much more to construct. First, the barcode length should be long enough such that sequence error will not result in misidentification in but also should be short as much as possible in order to reduce computational challenge in downstream analysis. Second, truncated barcode might be erroneously generated and could dramatically reduce the library complexity. A fixed "spacer" could be inserted into adjacent random region to remove most of truncated barcodes. In addition, "spacer" also adds the benefit for the downstream analysis by serving as anchor to locate the barcode, therefore, simplifying the barcode extraction. Another challenge is to integrate the DNA barcode with genome at neutral location, which should be considered carefully. There are a couple of feasible methods, each of which has its own pros and cons. The most popular way is to use Plasmids containing viral recombinases such as Cre-loxP (Austin, Ziese et al. 1981) and the lambda phage recombination system. This approach could combine the barcode to a specific sit of genome with high efficiency (Figure 10).

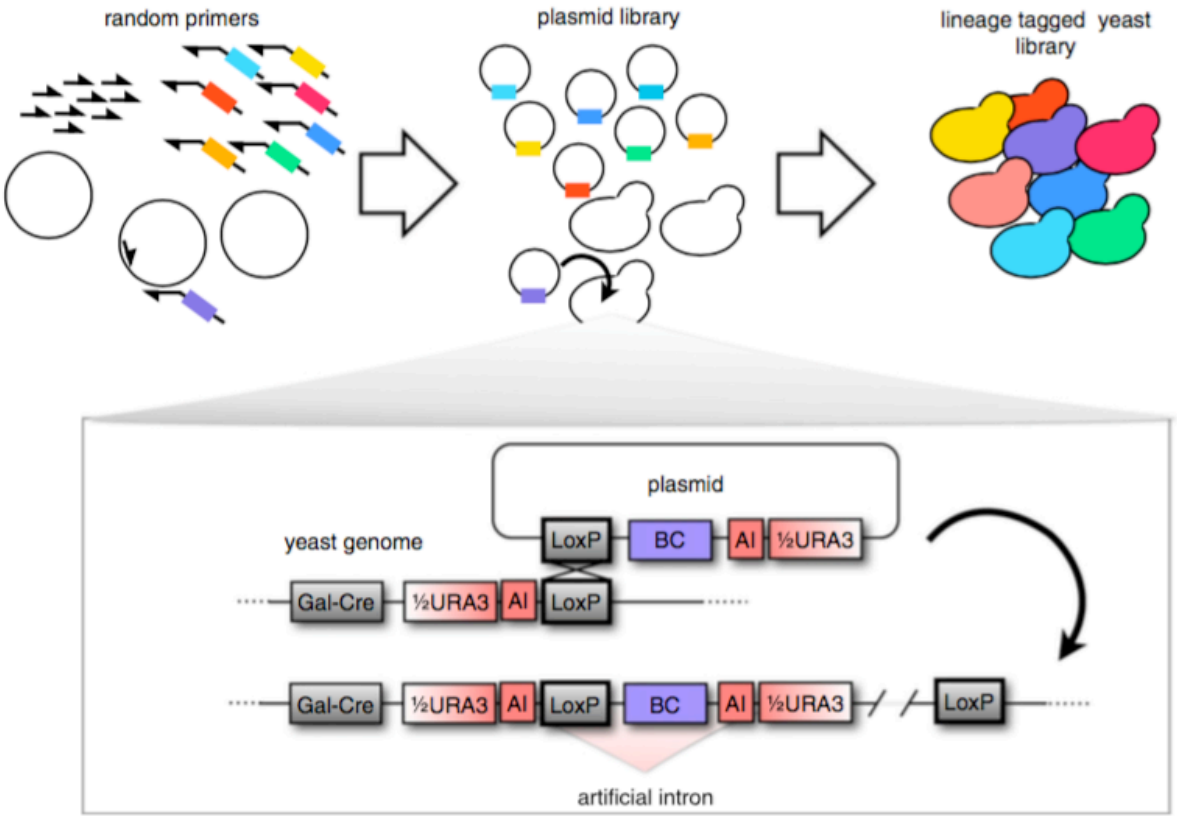


Figure 10 Site-specific genome integration

Sequences containing random barcodes (indicated by different colors) are amplified by PCR and ligated into a plasmid. The plasmid library is integrated into a specific neutral location in the yeast genome using the high-efficiency viral Cre-LoxP recombination system. Plasmid integration com- pletes a URA3 selectable marker that is interrupted by an artificial intron containing the barcode and one loxP site. Barcodes are maintained during the evolution by continued selection for the URA3 marker (Blundell and Levy 2014).

Now the study that uses high-throughput sequencing of nucleotide barcodes to assay large numbers of cell lineages, genotypes or perturbations in complex cell pools forms a new area named bar-seq. It generally works by growing a pool of barcoded cells under selective conditions, amplifying extracted barcodes using common primers, and sequencing barcode amplicons to quantify relative barcode frequencies in the cell pool. This approach was first applied on the *Saccharomyces cerevisiae* deletion collection, which was designed such that each individual deletion strain is marked with a unique barcode (Winzeler, Shoemaker et al. 1999, Giaever, Chu et al. 2002). Bar-seq of these deletion strains have been used, for example, to

detect gene products that are drug targets (Smith, Vijaykrishna et al. 2009), and those that are important for surviving starvation (Gresham, Boer et al. 2011), and heat stress (Gibney, Lu et al. 2013). Barcoded deletion collections have subsequently been generated in a number of bacteria and yeasts, allowing for similar massively parallel functional profiling (Han, Xu et al. 2010, Hobbs, Astarita et al. 2010, Noble, Treadwell et al. 2010, Schwarzmuller, Ma et al. 2014). Analogously, a growing number of studies in mammalian cells sequence pseudo-barcodes: short nucleotide sequences such as shRNAs or sgRNAs, that serve as both the cell-specific perturbation and the unique cell identifier for short-read sequencing (Schlabach, Luo et al. 2008, Silva, Rowntree et al. 2008, Bassik, Lebbink et al. 2009, Sims, Mendes-Pereira et al. 2011, Wang, Wei et al. 2014, Wong, Choi et al. 2015).

In addition to the above approaches where the sequence of the barcode or pseudo-barcode is known *a priori*, more recent bar-seq studies employ barcodes of random unknown sequence. As one example, sequencing of the DNA that happens to be adjacent to a transposon in random transposon insertion libraries has been used to functionally profile gene disruptions when systematic deletion collections are unavailable (Gawronski, Wong et al. 2009, van Opijnen, Bodi et al. 2009, Carette, Raaben et al. 2011, Brutinel and Gralnick 2012). A second example is the insertion of random barcode libraries into genomes to serve as neutral markers for lineage tracking studies over the course of development, evolution or cancer progression (Lu, Neff et al. 2011, Blundell and Levy 2014, Bhang, Ruddy et al. 2015, Levy, Blundell et al. 2015, Nguyen, Pellacani et al. 2015). In these studies, random barcodes are generated from primers with random bases, which are first inserted into vectors and then into cell genomes.

Despite the widespread use of bar-seq, computational pipelines for handling the resulting sequencing data have not been well developed. For barcodes of known sequence, the primary concern is mapping reads that may contain PCR or sequencing errors to the known barcodes. One naïve strategy would be to ignore reads that do not exactly match any putative barcode. However, given that some barcodes in the pool may be more prone to PCR or sequencing errors (Meyerhans, Vartanian et al. 1990, Goren, Ozsolak et al. 2010, Gundry and Vijg 2012, Schmitt, Kennedy et al. 2012), this strategy could introduce counting biases. A more sensible strategy currently employed is to compare each read to the set of putative barcodes by calculating the Hamming(Hamming 1950) or Levenshtein distance(Levenshtein 1966), thereby discovering the best match (Gresham, Boer et al. 2011). However, this strategy is extremely computationally expensive and computational cost grows at least quadratically with the number of putative barcodes. Additionally, *a priori* errors in the set of known barcodes (likely due to errors that occurred during Sanger sequencing of barcoded clones) have been found to be common (Smith, Vijaykrishna et al. 2009), meaning that unexpected barcodes that are present in the pool may be missed. To find these, an unbiased barcode detection strategy that does not depend on prior information would be needed.

For random barcode libraries, an additional computational problem is discovering the true barcodes in the pool. That is, reads that identify a true barcode must be differentiated from reads that contain PCR or sequencing errors. Because sequences representing an exact match to a true barcode are likely to be sequenced at much higher frequencies than those with errors, one approach would be to ignore reads below a predefined frequency threshold and treat all other reads as true barcodes. However, in cell pools with a skewed barcode frequency distribution, less abundant true barcodes will fall below the threshold, while errors from extremely abundant

barcodes rise above it. If each barcode in the pool is expected to be distant in sequence space from all other barcodes (e.g. >3 mismatches), a second approach is to cluster reads by their sequence similarities (Lu, Neff et al. 2011, Bhang, Ruddy et al. 2015, Levy, Blundell et al. 2015). Here, reads that are within one or two mismatches of each other are grouped together, with the most abundant sequence likely to be the true barcode and others being errors. This approach, however, is currently extremely computationally expensive because each unique read must be compared against each other unique read, with the number of unique reads often exceeding 10^6 in a typical Illumina HiSeq run. To avoid calculating all pairwise Hamming or Levenshtein distances, we have previously used a semi-pairwise BLAST strategy to cluster reads (Altschul, Gish et al. 1990, Levy, Blundell et al. 2015). However, even this strategy is computationally expensive and may often falsely merge distinct barcodes that are close in sequence space (see Results).

Here, we developed Bartender, an ultrafast and unbiased clustering algorithm for identifying and counting barcodes and pseudo-barcodes from short read sequencing data. We use a divide-and-conquer strategy that first identifies high-quality seeds and then iterates through each seed to sort short reads into different bins for parallel processing. Reads within each bin are clustered using a computationally efficient greedy clustering algorithm. Instead of merging clusters solely on sequence similarity (Hamming or Levenshtein distance), Bartender uses additional information of the cluster size to prevent over-merging. Our algorithm includes handling of unique molecular identifiers (UMIs), which, if included in the sequencing reads, allow an investigator to detect and remove PCR duplicates (Kivioja, Vaharautio et al. 2012, Levy, Blundell et al. 2015). Additionally, we include a "multiple time point" mode, which uses cluster information from adjacent time points to minimize the impact of read errors on barcode

trajectories. Compared to previous methods, we find Bartender to be orders of magnitude faster and more accurate for routine processing of barcodes and pseudo-barcodes.

1.3 Method based on BLAST

To the best of our knowledge, there is no general clustering algorithm for bar-seq data. Most research groups developed their own in-house methods to count lineage tag. Due to the most recent and relative efficient one is described in (Levy, Blundell et al. 2015), which uses the semi-pairwise BLAST strategy (this method will be referred as BLAST in the remaining sections) to group similar unique reads to identify the true lineages and corresponding frequency. There are three major steps that are critical for keeping a result as accurate as possible and reducing the computation.

The first step is to find the size threshold to split unique reads into putative barcode (high frequency) and low frequency unique reads, most of which are barcode variations due to sequence and PCR errors. To determine the gating threshold, one has to know the underlying experiment design (e.g the expected number of barcode and the distribution of unique reads) and has a good estimation on sequence parameters including sequence depth, sequence error. In the paper (Levy, Blundell et al. 2015), the gating threshold is set to 10 based on their own experiments.

Once the unique reads are divided into two lists, the second step is to group similar high frequent sequences by performing a pairwise comparison within the high frequency list, where the similarity is defined in terms of e-value in BLAST package. Two clusters are joined if any member (sequence) presents in both cluster. Clustering joining continues until the number of clusters is stable.

In the last step, sequences with less than the pre-specified size threshold are then matched to the clustering list obtained in the second step by BLAST using the same criteria.

One challenge posed by this method is how to define the similarity using the e-value in BLAST. E-value in this problem is related with many factors, i.e. the barcode library size, barcode length and sequence errors. There is no explicit formula or guideline for the choice of e-value, which make this method difficult to tune once the data changes.

Presumably each cluster represents one barcode lineage and the cluster size is treated as the lineage size. BLAST calculates the center by majority voting at each nucleotide position and use the center to be representative of the lineage tag.

2 Methods

This chapter covers Bartender in full details. Each section describes one component of Bartender, including the Bartender extractor, the clustering algorithm and multiple sample mode.

Barcode or pseudo-barcode lengths typically range between 10 and 30 nucleotides. However, with the advent of highly complex long oligonucleotide libraries (LeProust, Peck et al. 2010). Synthesis of high-quality libraries of long (150mer) oligo-nucleotides by a novel depurination controlled process. Variable regions may sometimes exceed 100 nucleotides(Goodman, Church et al. 2013, Kosuri, Goodman et al. 2013). Current tools become cumbersome for analysis of these longer barcodes for two major reasons. First, longer barcodes will accumulate more errors and result in more unique reads that must be clustered. Second, longer barcodes will slow similarity comparisons between unique reads because more nucleotide positions must be accounted for. To address these problems, an ultrafast and accurate package, named Bartender, was built with the capability and speed to handle arbitrary barcode lengths.

First, an extractor tool was developed that will quickly pull the variable region from FASTQ/FASTA files using a user-defined pattern and read quality threshold (see below). Second, the Bartender clustering tool was designed to improve speed and accuracy by exploring several strategies (below). We devised a statistics test to add another constrain on merging operation between reads or clusters in the clustering process. This test statistic incorporates the sequence similarity and the barcode size information into the two-sample proportion test by assuming that the small cluster derives from the large cluster by sequence error under the null hypothesis. The primary speedup comes from a divide-and-conquer binning strategy that greatly reduces the number of comparisons (Figure 13). Lastly, a multiple sample mode was developed to track the

lineages along the time point.

2.1 Bartender Extractor

Bartender extractor is a simple command-line tool that is designed to extract barcode from the raw reads. This tool was developed under the barcode temple design philosophy presented by (Blundell and Levy 2014, Levy, Blundell et al. 2015). Basically it assumes that each read has one barcode and the barcode is preceded and followed by a flank sequence, which help to locate the barcode. The barcode template could have multiple random regions interleaved by spacers (optional). To deal with sequence error, at most one mismatch in each flank region is allowed. Synthesis of oligonucleotides with random regions can often result in some variants with missing or additional random bases. To account for these errors, the extractor allows the user to assign random nucleotide regions of variable lengths during extraction.

The extractor tool also calculates the combined PCR and sequencing error rate by examining the frequency of errors at user-defined invariant regions of the amplicons. These error rates are useful for downstream clustering. Along the extraction process, the extractor also keeps track the total number of error bases identified in the flank region and estimate the sequence error using the percentage of error bases in the flank region in the last step. Since it allows one mismatch in flank region, the specified flank sequence should not be too long in order to obtain relative accurate error estimation given only one-mistmatch is allowed and we recommend 5bps right next to the barcode region from both end separately. Although the sequence error estimated might have bias given the fact that the sequence error is not uniform across all positions, it could give some insight on the data and serves as a baseline to compare the sequence error estimated from the clustering result by Bartender.

The extractor transfers the fed barcode template into regular expression and extracts the matched subsequence in each read in the raw file (FASTQ or FASTA format). To filter out low quality barcode, a user defined quality threshold is used to filter out those barcode, whose average quality is below the threshold (only applicable to FASTQ format). The matched sequence (flank region removed) and the corresponding line number in raw file forms the ultimate output file, which could be the input of the clustering component. To remove the PCR effects, users need to extract the UMIs from the raw reads by themselves, replace the line number with the UMIs at same line of the raw file and use this updated file as the input for clustering algorithm.

2.2 Clustering algorithm

2.2.1 General overview

The Bartender clustering tool utilizes a number of strategies to improve speed. The primary speedup comes from a divide-and-conquer binning strategy that greatly reduces the number of comparisons (Figure 11). Briefly, Bartender surveys the variable regions for nucleotide positions with the greatest entropy (most variability) and generates a set of non-contiguous seeds (3-8 nucleotides) ranked by total entropy (Figure 13). These seeds are then used to split unique reads into bins, with each read within a bin having an identical seed. Comparisons are only performed between reads within each bin. Similar clusters are merged (the merging criteria are described later) and these merged clusters are then used with the next seed for further merging. By using seeds with the greatest entropy first, Bartender creates the maximum number of bins at the first clustering step (when the most clusters exist) and thereby minimizes the number of comparisons necessary. Bartender continues clustering with additional seeds. In most cases, however, the number of comparisons needed drops dramatically between

rounds. A secondary speedup comes from prioritizing comparisons to the largest clusters within each bin. These large clusters are the most likely to be "true" barcodes and have a better chance of matching to smaller clusters that are more likely to be a sequence that contains a PCR or sequencing error. A last source of increased speed comes from the use of a computationally efficient comparison metric that is described in more detail later. Bartender also estimates the sequence error using high quality and relative large clusters after the clustering process. The two estimated sequence error could be compared and help users roughly evaluate how the Bartender perform on the data and tune the parameters e.g distance threshold and z-value to generate "better" results

Bartender's accuracy stems from a new statistical test schema that uses both nucleotide sequence and cluster size information to prevent over-merging; its speed stems from a novel binning strategy and a computationally efficient greedy clustering algorithm. Bartender includes handling of both UMIs and time course data, and promises to be a useful tool for a large number of diverse applications. Here is a high level introduction to our method and the following sections will explain the critical components one by one.

Before clustering, identical barcode reads are grouped together to form a list of unique read sequences, each is associated with a count. Unique Molecular Identifiers (UMIs) for each read, which can later be used to detect and remove PCR duplicates, can be stored with each cluster. Using this list, a set of overlapping 3-8 base pair seeds are selected (depending on how parameters are set). Seeds are not necessarily contiguous and are chosen such that nucleotide positions with the most entropy (variance) appear in the first seed, those with slightly less entropy are added to the second seed, and so on (Figure 13). The total number of seeds is determined by the seed length and the number of selected positions. For each seed, clusters

(initially the set of unique reads) are partitioned into bins, each of which has a unique seed sequence. Next, each bin is clustered independently using the greedy clustering approach that searches for sequence similarity across all nucleotides of the barcode. Then clusters are merged within each bin and these are used as the starting point for clustering with the next seed. Each step is discussed in more detail below. PCR duplicates are next removed by searching for identical UMIs within each cluster. Lastly, three data files are output: the consensus cluster sequence (the barcode) and its counts, the quality of each cluster, and a map between each unique read which cluster it belongs in.

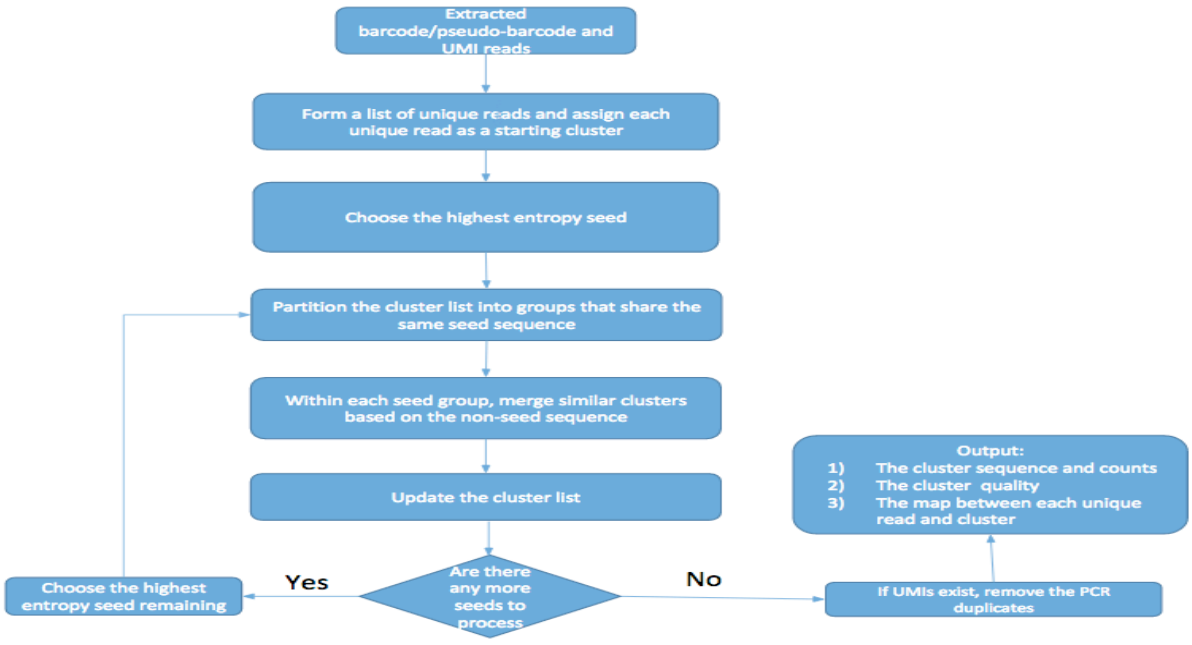


Figure 11 A schematic flowchart of the Bartender clustering algorithm

Figure 12 illustrates the detail logic of the clustering algorithm. The algorithm takes a list of extracted barcode and UMIs associated with each barcode. And it outputs a list of clusters with quality and mapping information between unique reads and clusters.

```
Data: F : Barcode frequency table
Result: C : A list of clusters
Initialize C with F by setting each unique barcode as a cluster itself;
Identify a list of seeds S using the position weight matrix of F;
for s in S do
    size_threshold ← trimmed mean of the cluster size in C ;
    l ← [] ;
    shatter C into buckets B using s ;
    for b in B do
        lb ← ClusterSingleBucket(b, size_threshold) ;
        append lb to l ;
    end
    C ← l ;
end                                                          s
```

Figure 12 Main procedure of clustering algorithm

The input is the tabulated extracted barcode frequency table. The output is a list of clusters. The logic is very simple. It first identifies the high diversity nucleotide positions and form a list of seeds using these selected positions. Then it group similar barcode/clusters using each seed. And the algorithm stops when there is no change between adjacent round or all seeds are processed.

There are three critical components in this clustering algorithm. The first one is the binning strategy. The second one is the test statistics used to distinguish barcode lineages that are similar in sequence space. The last one is the greedy clustering algorithm used in each bucket after binning step. Before introducing these components, some necessary notations need to be defined.

2.2.2 Dissimilarity measures between reads and clusters

Suppose there are N unique reads and let r_i Be the ith unique read with a frequency of f_i. For two distinct reads (r_i And r_j), we use Hamming distance (Hamming 1950) as the dissimilarity metric to defines their distance, which is given as follows:

$$d(r_i, r_j) = \sum_{k=1}^{L} I[r_i(k) \neq r_j(k)] \qquad \text{Equation 2-1}$$

where $r_i(k)$ is the nucleotide at the kth position of sequence r_i, and $I(x)$ is an indicator function such that

$$I(x) = \begin{cases} 1 & \text{if x is true} \\ 0 & \text{if x is false} \end{cases}$$

For a particular barcode cluster C that may contain several unique reads, we define its size to be:

$$S(C) = \sum_{all\ r_i \in C} f_i \qquad \textit{Equation 2-2}$$

The centroid of the cluster C is defined as a sequence $c = [c_1 c_2 \dots c_L]$, where $c_k = \underset{b \in \{A,C,G,T\}}{\operatorname{argmax}} \sum_{all\ r_i \in C} f_i * I(r_i(k) == b)$, for $k = 1, \dots, L$. That is, the centroid consists of nucleotides that are most frequent at each position. Since a cluster usually corresponds to a barcode, the centroid can also be viewed as an estimate of the corresponding barcode.

For two distinct clusters C_x, C_y, let c_x, c_y denote their centroids. We use *Hamming distance* between the two centroids as the dissimilarity metric to define the distance between the two clusters, which is given as follows:

$$d(C_x, C_y) = d(c_x, c_y) \qquad \textit{Equation 2-3}$$

The position weight matrix PWM_{4*L} of cluster C is defined by

$$P_{ij} = \frac{\sum_{r_k \in C} (r_{k,j} == i) * f_k}{S(C)} \qquad \textit{Equation 2-4}$$

Where A, C, G, T is encoded as 0,1,2,3 respectively.

2.2.3 Seeds selection and binning

The most computationally expensive task of this problem is a large number of pairwise comparisons between barcode sequences. To minimize the pairwise computation, we partition

barcode clusters by seeds and only perform pairwise comparisons within each bucket. Thus, choosing seeds that break barcode clusters into a certain number of bins will decrease the number of pairwise comparisons and thereby improve speed. To select seed nucleotide positions with the greatest potential to maximize effectiveness of binning, the entropy value for each nucleotide position is calculated first. For N unique reads, we first calculate the frequency of each nucleotide at each position: $f_{b,k} = \sum_{i=1}^{N} f_i * I[r_i(k) == b]$, where $b \in \{A, C, G, T\}$ and $k = 1, \ldots, L$. At each position k, the four nucleotides are dichotomized into one group with the most frequent nucleotide (the major allele) and another group with the other three nucleotides (the other alleles). The relative frequencies of the major and other alleles are then calculated and denoted as $p_k = \frac{\max_{b \in \{A,C,G,T\}} f_{b,k}}{\sum_{b \in \{A,C,G,T\}} f_{b,k}}$ and $1 - p_k$, respectively. The entropy at position k is defined as:

$$E[k] = -p_k log_2(p_k) - (1 - p_k) log_2(1 - p_k) \qquad \textit{Equation 2-5}$$

Seeds are then selected based on the order of the entropy values, and positions with larger entropies are used in earlier iterations of binning. Nucleotide positions with too low entropies below a pre-specified (or user-defined) threshold are excluded from seed selection.

Next, clusters, based on their centroids, are sorted into different bins based on selected seeds. The binning process is demonstrated in Figure 13. Each bin contains all clusters containing the same seed sequence. There are two major computational advantages of binning by seeds. First, unnecessary pairwise comparisons between distant clusters are dramatically reduced because only clusters within the same bin will be compared. Second, each bin can be processed independently, making it easy to parallelize the algorithm on multiple-core computers. Intuitively, longer seed generates more bins and the number of clusters in each bin decreases

which will reduce the number of pair-wise comparison. So seed length is a critical parameter that balances the speed and accuracy of the clustering algorithm. That is, longer seeds increase speed by reducing pairwise comparisons, but are more likely to leave spurious barcodes ungrouped, thereby increasing false positives. By default, seed length is set to be five (see Figure 13 for details), and Bartender iterates through the seed position list using a sliding window with a size of the seed length and slide one nucleotide at a time.

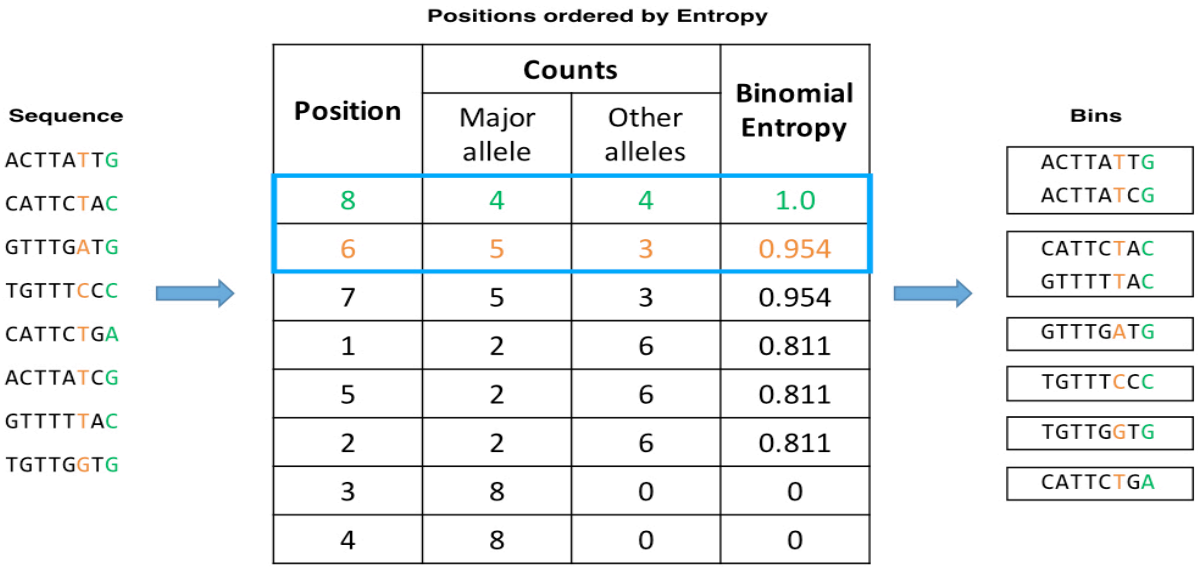


Figure 13 Binning strategy

A toy example of seed selection on eight unique sequences with a seed size of two. The entropy value at each position is calculated and positions are sorted in descending order, with ties broken arbitrarily. Here, positions eight and six have the highest entropy and are used as the first seed, breaking the sequences into six bins (right). The next seed is positions are by default positions six and seven.

2.2.4 Statistical test schema in clustering algorithm

When two clusters are close in distance, they may arise from three possibilities: (1) one cluster is a true barcode cluster and another is a cluster with error reads, (2) both clusters are errorneous reads derived, and (3) both clusters are true barcode clusters. To avoid over-merging

in the second case, we use a modified two-sample proportion test to determine if two clusters would be merged or not.

Let C_1 and C_2 denote the two clusters being tested, with centroids of c_1 and c_2, respectively. The distance between C_1 and C_2 is $d(C_1, C_2) = d(c_1, c_2) = d$. Let C_3 denote the new cluster when C_1 and C_2 are merged, with a new centroid c_3. The sizes of C_1 and C_2 are S(C_1) and $S(C_2)$, and the size of C_3 is $S(C_3) = S(C_1) + S(C_2)$. Let $e_{i,3}, i = 1,2$ be the cumulative number of base pair errors of reads in C_i with respect to the new cluster C_3:

$$e_{i,3} = \sum_{all\ r_k \in C_i} d(r_k, c_3) * f_i \qquad \textit{Equation 2-5}$$

Let $p_{i,3}$ be the error rate of cluster C_i with respect to cluster C_3, which is defined as:

$$p_{i,3} = \frac{e_{i,3}}{L*S(C_i)} \qquad \textit{Equation 2-7}$$

The idea is that if one barcode cluster indeed originates from errors of another barcode cluster, the size of the error cluster should be much smaller than that of the true barcode cluster. This two error rates are different from each other in most cases. This test statistic incorporates the size information, which amplifies the error rate by the cluster size (the smaller one). That is, the larger the smaller cluster is, the more significant of this test will be.

The hypotheses can be formulated as

$$H_0: p_{1,3} = p_{2,3} \quad vs \quad H_1: p_{1,3} \neq p_{2,3}$$

The test statistic is given by:

$$T = \frac{p_{1,3} - p_{2,3}}{\sqrt{\frac{p_{1,3}(1-p_{1,3})}{L^{*}S(C_1)} + \frac{p_{2,3}(1-p_{2,3})}{L^{*}S(C_2)}}}$$ *Equation 2-8*

We reject H_0 for large T value. T becomes large under two scenarios: 1) the distance between cluster centroids is large regardless of the relative cluster sizes, and 2) a small but consistent difference between two sufficiently large cluster centroids. This allows us to merge closely-spaced clusters that are likely caused by errors but avoid merging-closely spaced clusters that are likely to each represent a true barcode. Since true barcodes close in sequence space accounts a very small portion of the barcode library, the test should be very conservative on rejecting the null hypotheses, which could be achieved by choosing a very small p-value or a very large critical values.

This test schema becomes very sensitive when the number base pairs within both clusters are sufficiently large. That is, any small difference between two sufficiently large clusters will make the test significant and the merge operation will be rejected, which is the desirable feature used to alleviate merging issues.

2.2.5 Clustering within one bin

Since there may be thousands of barcode clusters within one bin, pairwise comparisons within a bin are still computationally expensive. To further improve the speed, we developed a greedy clustering algorithm that uses cluster frequency information to prioritize comparisons. The rationale is that clusters with high frequencies are more likely to represent a true barcode, while those at low frequencies are more likely to be errors. We take advantage of this by first splitting clusters into high- and low-frequency bins based on a threshold, the mean cluster size

that is derived from the empirical distribution of the cluster sizes. Since error reads are expected to greatly outnumber reads with true barcode sequences, the mean cluster size should partition the majority of error-containing sequences (and a minority of true barcodes) into the low-frequency group. Then for each cluster in the low-frequency group, all similar high-frequency clusters are picked up by matching it against to the high-frequency group. There are three scenarios on the number of picked high-frequency clusters. If no high-frequency cluster got picked, this low-frequency cluster is added to the high-frequency cluster group, as they are likely to be true barcode clusters; if it only picks one large cluster, the low-frequency cluster is then merged into this high-frequency cluster. If this low-frequency cluster picks more than one high-frequency clusters, the low-frequency cluster will be merged with the closest high-frequency cluster. If there are multiple high-frequency clusters are equally close to this low-frequency cluster, then it will be merged into the largest high-frequency cluster and tie break arbitrarily. A pairwise comparison among those picked high-frequency clusters is performed and merge those large clusters if they are similar and pass the statistical test (see Statistical test schema in clustering algorithm). A rigorous description is given by Figure 14.

```
Data:
      C : a list of clusters;
      T : a size threshold used to split clusters;
      D : distance threshold.
Result: R : A list of clusters
S ← [] // clusters with size smaller than or equal to T;
L ← [] // clusters with size larger or equal than T;
// split the cluster into two groups based on cluster size ;
for c in C do
    if c.size ≤ T then
        Append c to S ;
    end
    else
        Append c to L ;
    end
end
Sort S in descending order;
R ← [] ;
for each s in S do
    pc ← [];
    // Find all big clusters that are similar with s and add it to pc;
    for each l in L do
        if d(l, s) < D then
            Append l to pc ;
        end
    end
    if pc is not empty then
        sort pc by cluster size in descending order;
        merge s with the first cluster in pc;
        while pc is not empty do
            u = pc[1];
            remove the first element form pc;
            for c in pc do
                if d(u, c) < D and shouldMerge(u, c) then
                    merge u and c;
                    remove c from pc;
                end
            end
            Append u to R;
        end
    end
    else
        Append sc to R
    end
end
```

Figure 14 Greedy algorithm within single bucket

Figure 14 Greedy algorithm within single bucket *illustrates the clustering process in each bucket. This module takes a sub list of clusters that belongs to the bucket and outputs a list of updated clusters. This module is called by the main procedure for clustering individual bins.*

2.3 Merging clusters across multiple samples

A growing number of studies perform time course bar-seq experiments that require tracking barcode lineages over time (Gresham, Boer et al. 2011, Levy, Blundell et al. 2015). In some cases, lineages will persist at low frequencies or be driven over time to extinction. These sorts of scenarios present additional challenges for Bartender clustering. As discussed above, a small cluster is more likely to result in a difference between the true barcode sequence and the sequence Bartender calls. These cluster center errors will result in some barcodes that reach low frequencies apparently "disappearing" at some time points and then "reappearing" at others, greatly impacting interpretation of the lineage trajectory. One work-around we previously employed (Levy, Blundell et al. 2015), was to pool reads from all time points, cluster the big pool to build a map between each unique read and a barcode cluster, and then reconstruct the counts at each time point from this map. However, this approach is extremely computationally expensive and can easily exceed the memory limitations of even a powerful desktop computer.

To solve this problem, we include a "multiple time point" mode in Bartender. This feature allows time points to be clustered independently, thereby reducing computational overhead and allowing for parallelization. First, the clustering result of each time point should be obtained by applying the Bartender to the raw data. Once barcode clusters are obtained for each time point; we next match cluster centroids across all time points. Sequencing or PCR errors, especially in reads that form a small barcode cluster, may result in centroids that do not match between time points. For time course data, where many barcode lineages may be driven to extinction, these mismatches may result in the observation that some lineages prematurely disappear from the pool, as errors in small clusters become increasingly likely to result in miscalled centroid. For data with multiple time points, these errors may make Bartender miss

crucial non-zero data points. To solve these problems, we load the clustering results in the order specified by user and assume that each listed result is the descendent of the previous time point. This strategy uses prior information to assign the results across time points (Figure 15) and handles several subtleties observed in real data.

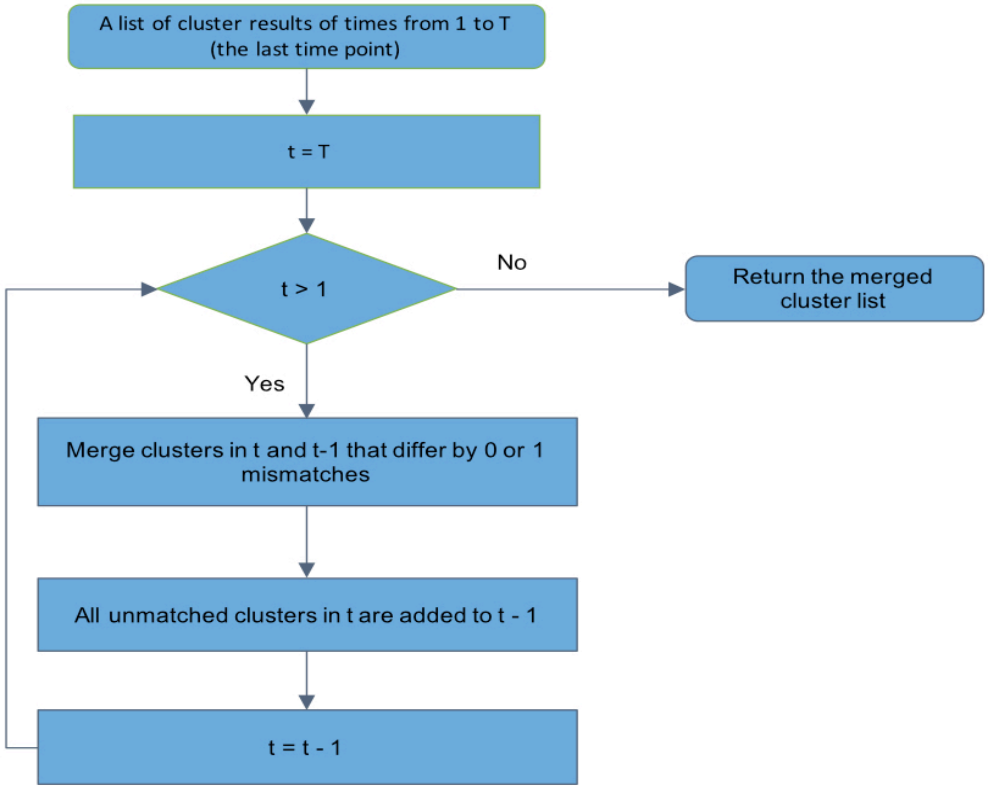


Figure 15 The schematic flowchart of multiple sample mode

A flowchart of cluster merging across multiple time points. The merging strategy starts from a list of clustering results of each time point. Clusters from the latest available time point are merged to identical clusters from the previous time point. Clusters without an identical match merged with the most appropriate (see the main text) cluster within one mismatch. Other unmatched clusters will be kept only if its size is larger than the user specified size threshold.

By comparing barcode clusters with the previous time point, exact matches are merged into the previous condition. The remaining unmerged barcode clusters of current time point fall into two primary categories. One is unmerged barcode variations due to PCR and sequence error. The second are those true barcodes, which is not (correctly) identified in the previous time point.

To handle these two cases, Bartender searches for all barcode clusters with a centroid that is one mismatch away from an unmatched cluster centroid in the previous condition. If a match exists, the cluster is merged with the closest cluster in sequence space (Equation 2-1) if the merge operation passes the statistic test. If there are multiple one mis-matched clusters, then it will be merged with the cluster that has the least significant test result. If there is no match and the size of unmatched barcode cluster is larger than the user defined threshold (most likely be the second category), it will be added to the current condition. The size of each barcode cluster is recorded for all time points, with "zeros" included when the cluster is absent at any time point. The final result contains a list clusters, each of which keeps the size information along the merging process. The overall cluster quality and position weight matrix across all time points (conditions) are also included in the final result. The detail logic is presented by Figure 16 and Figure 17.

Data:
L : a list of list of clusters from multiple time points;
T : a size cutoff used to filter small size clusters;
Result: R : A list of clusters
$l \leftarrow L.size()$;
$R \leftarrow L[l]$;
while $l > 0$ **do**
  $l \leftarrow l - 1$;
  $R \leftarrow$ mergeTimePoint($L[l]$, R, T);
end

Figure 16 Main procedure of merging strategy

The algorithm takes into a list of clustering results. The results should be logically ordered either by condition or time. Then it merge the results in backward manner by repeatedly calling procedure that merges two adjacent time points, which is described in Figure 17.

```
Data:
      L1 : a list of clusters of previous time point;
      L2 : a list of clusters of current time point;
      T : a size cutoff used to filter small size clusters;
Result: R : A list of clusters
R ← [];
for each c1 in L1 do
    if There is a cluster c2 in L2 has same center with c1 then
        c1.merge(c2);
        Append c1 to R;
        Remove c1 from L1;
        Remove c2 from L2;
    end
end
//For those remaining clusters in L1 and L2;
// merge all pairs of clusters in L1 and L2, which are one bp away from
each other;
for each c2 in L2 do
    lsim ← all clusters that are one bp away with c2;
    cl ← the largest cluster in lsim;
    cl.merge(c2);
    Append cl to R;
    Remove cl from L1;
    Remove c2 from L2;
end
Insert L1 to R;
// Remove low frequency clusters in L2;
for each c2 in L2 do
    if c2.size() ≥ T then
        Append c2 to R;
    end
end
```

Figure 17 Procedure of merging two adjacent time points

Basically, it assumes that the lineages in current time point is the subset of the previous time point. To correct the mis-call center due to PCR or sequence error, it first find all exact matched clusters by their center under Hamming distance metric. Then it finds all clusters that are one-mismatch away and combine them. For those unmatched clusters with size larger then the user specified size threshold, they will be kept and should be checked in more detail after the mergint step.

2.4 Miscellaneous

Bartender measures the similarity between reads/clusters using Hamming distance (see above) instead of Levenshtein distance for two reasons. First, the time complexity of computing Levenshitein distance is quadratic with respect to barcode length, which make it a computational burden for long barcode sequence given that the dominant computation step of this problem is

barcode sequence comparison. In contract, the computation of Hamming distance metrics grows linearly with the barcode sequence length and has much more advantage for long barcode sequence although it is only applicable for sequences with same length. Second, insertion or deletions occur in much lower chance than mismatches in biological process or sequence stage. So the accuracy gains by using Levenshitein similarity metric is very limited for bar-seq data. Theoretically, Bartender will slightly underestimate the lineage size if indels happen to this lineage in the biological process or other experimental steps. However, we did not observe this in the comparison on the real data between BLAST and Bartender (see

Performance on real data).

Unique molecular identifiers (UMIs) are additional, usually random, sequences that are added to template molecules before PCR that allow an investigator to detect and remove PCR duplicates and thereby improve the accuracy of amplicon counting (Kivioja, Vaharautio et al. 2012, Levy, Blundell et al. 2015). Bartender allows a user to attach a UMI sequence to each barcode prior to clustering: the user must simply generate a comma separated file the contains one barcode and one UMI on each line. Following clustering, Bartender will search for identical UMIs within each cluster and report counts that include or exclude repeated UMIs (putative PCR duplicates). We note here that UMI length must be carefully considered as part of the experimental design, and provide more some general guidelines in the Discussion.

To better evaluate clustering result, Bartender provides a cluster quality measurement using the entropy metric. For each cluster, the position weight matrix is first generated based on all unique reads that belong to it. Then the largest binomial entropy value in each position is treated as the cluster quality and saved to the output file. This measurement directly gauges the

most diversity nucleotide position. And this diversity could help users to check if this cluster could potential contain more than one true barcodes. Bartender also estimates the sequence error by leveraging the cluster result. It first selects all high quality clusters by the cluster quality value (e.t the value is less than 0.8 (the majority nucleotide accounts at least 95% in each position)). Then it adds up all error bases and non-error bases in all clusters. The sequence error is the ratio of this to numbers.

3 Experiments and validations

To characterize bartender performance and accuracy, three simulation datasets and one real datasets are applied to Bartender with different settings. This chapter is organized into four sections. The first section characterizes the Bartender's efficiency using three datasets. Accuracy comparisons are presented using one simulation data and one real dataset in the second section. The third section demonstrates the effectiveness of multiple sample mode using a multiple time points simulation datasets. Lastly, an experimental approach was designed and carried out to study the impact of sequence depth on Bartender's accuracy.

3.1 Bartender is flexible and ultrafast

To measure the effect of these improvements, we compared Bartender clustering speed to the fastest existing clustering method, which performs comparisons between unique reads by BLAST (see Methods and (Levy, Blundell et al. 2015)). We simulated 100,000 barcodes of a number of different realistic barcode lengths (26, 38, and 64 nucleotides), while keeping the number of reads (~10M) and the combined PCR and sequencing error rate (2%) constant (Methods). The frequency distribution follows exponential distribution with mean 100. On a desktop computer, Bartender performed the clustering in under two minutes in all cases, at least two orders of magnitude faster than BLAST. One tunable parameter that affects Bartender performance is the seed length used to partition unique reads prior to performing pairwise comparisons (Methods). Longer seeds will reduce the pairwise comparisons (increase speed) by creating more bins, but, if error rates are high, may result in under-merging because similar clusters might never find themselves in the same bin. By testing Bartender performance across different seed lengths, we find that longer seeds do indeed increase speed, however increases are marginal above 5 (Figure 2B). We next performed these same speed tests on a typical laptop

computer, and found that performance is only moderately diminished (< 2-fold slower). Taken together, these results indicate Bartender can be used to process most realistic barcode libraries on most computers in just a few minutes. Other parameters in Bartender are default value, such as z-value is 5.

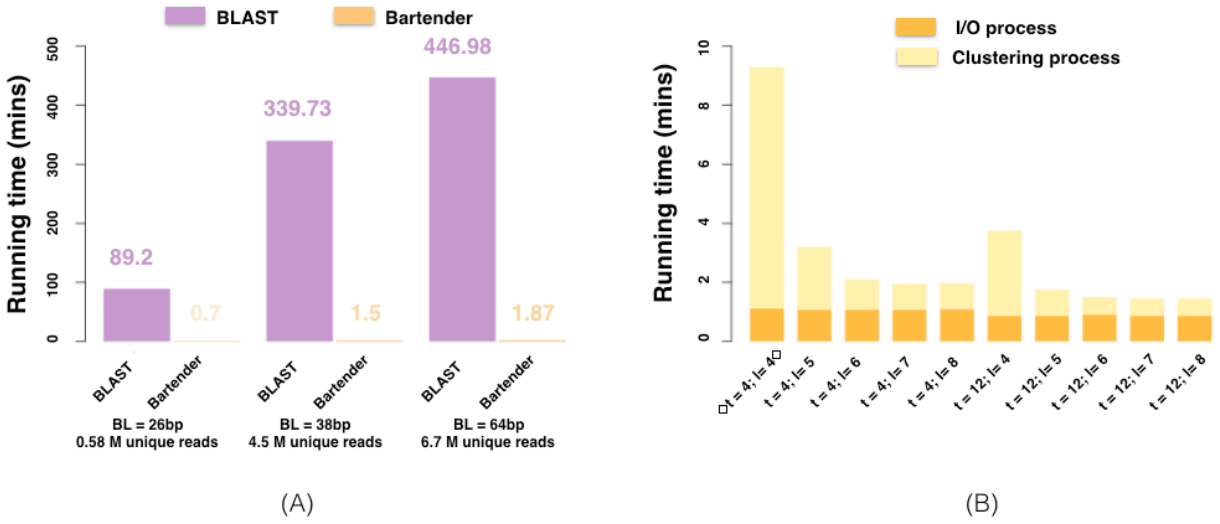


Figure 18 Bartender speed

(A) Running time for Bartender and BLAST on simulated data of barcodes of different lengths (BLs), using 12 processors (t = 12). Bartender was performed with a seed length of 5 (l = 5). B) Bartender performance using a variable number of processors (t) and seed lengths (l). Input/output (I/O) time is shown in orange and clustering time in shown in yellow. For t = 12, a desktop equipped with 3.5 GHz 6-core intel xeon E5, 64 GB memory was used. For t =4, a laptop equipped with 3.0 GHz Intel core i7 and 16GB memory was used.

3.2 Bartender prevents over-clustering

3.2.1 Performance on simulation data

To evaluate Bartender performance, we generated a simulated dataset that includes sampling noise from DNA extraction and sequencing and a 2% sequencing error rate (see Methods) (Levy, Blundell et al. 2015). Similar to the barcode design in our previous work(Levy, Blundell et al. 2015), 100,000 barcodes were generated, consisting of four random 5mers,

separated by two constant dimers. To simulate non-uniform barcode distributions that can be found in real data, simulated barcode frequencies were exponential ($\lambda = 0.01$) before introduction of sampling or sequencing errors.

The raw data has 4,562,582 unique reads with 38 base pairs (Table 6). To make the data close to real dataset, 6 base pairs flank region is added to both ends. There are 6 spacers and 20 random positions (Figure 19 (b))Averagely a true barcode loses 53.5 percent of size due to the sequence error (Figure 19 (c)). The data set is analyzed by bartender with the following options. Distance threshold is 3. The seed length is 6 and the overlapped base pairs between adjacent seeds are 1. Z value is 5(default). Since extremely low frequent barcodes are very hard to recover, the result only considers true barcodes with frequency larger than 3. The BLAST use e-value 0.01 to assess the barcode similarity, which is equivalent to 2-mismatch in Levenshtein distance.

Table 6 Simulation parameters

Parameters	Value
Number of Lineages	100,000
Barcode length	38
Barcode size distribution	Exponential distribution with λ=0.01
Number of reads	~10,000,000
Sequence error	0.02

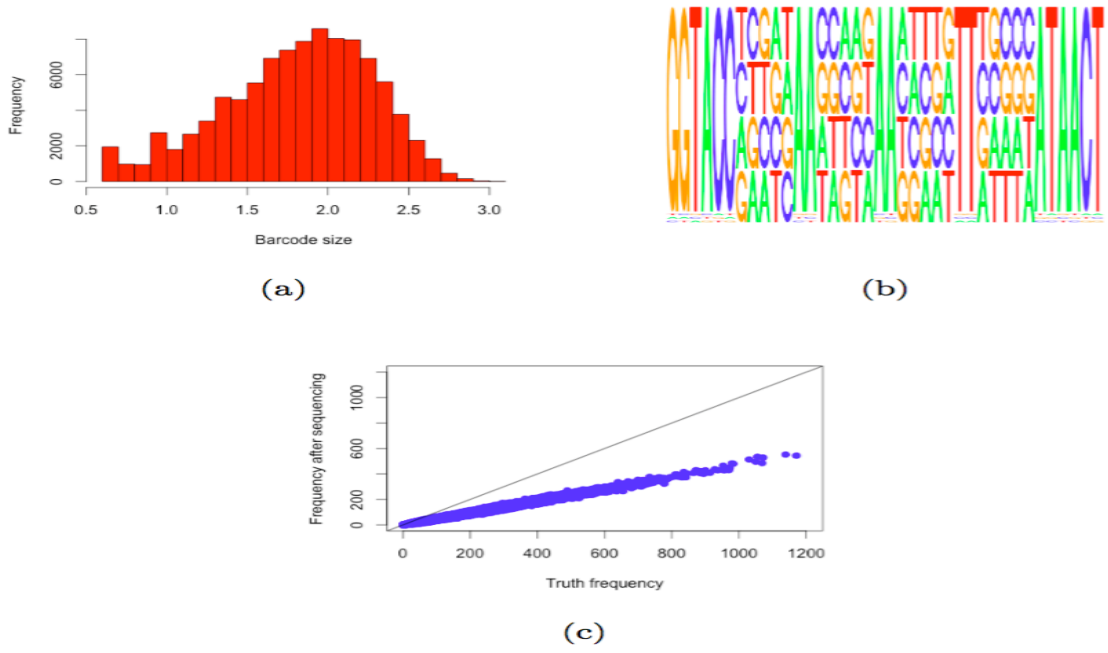


Figure 19 Summary of simulation data.

(a) is the histogram of the true barcode size in log in base 10 scale. The majority of barcoded has about 100 size, which is consistent with the simulation setting. (b) is the position matrix of the raw sequence data. (c) is the scatter plot between true size and frequency after sequencing. It shows that the size lost is proportional to the true barcode size.

The in-house method (BLAST) and bartender is applied to the simulation data. The best results from both in-house BLSST and Bartender are presented in Figure 20. For this comparison, clusters with less than three reads were ignored because these low-frequency clusters are more likely to be erroneous and derived from another cluster in the pool. Using the remaining 96610 clusters, the two methods found nearly the same number of clusters (96517 for Bartender, 95537 for BLAST), however Bartender generally estimated the true count with greater accuracy (Figure 20). In particular, many BLAST clusters were too large, indicating that two closely-spaced clusters were erroneously merged (Figure 20, green triangles). BLAST over-merging resulted in a higher number of false negatives (Bartender = 93, BLAST = 1073), and marginally less false positives (Bartender = 94, BLAST = 53), with Bartender errors exclusively occurring in small clusters. We note that over-merging is not a problem that is unique to BLAST. All previous

methods use a similarity threshold (e.g. 2 mismatches) to merge similar reads, but not cluster size information (unique to Bartender), and will therefore be subject to the same over-merging artifacts as BLAST unless the method under-clusters the data. The sequence error estimated by Bartender on this simulated dataset is 0.184, which is close to the theoretical value (0.2). Furthermore, BLAST takes about 339.73 minutes and bartender takes about 1.53 minutes, which is 122 times faster than blast.

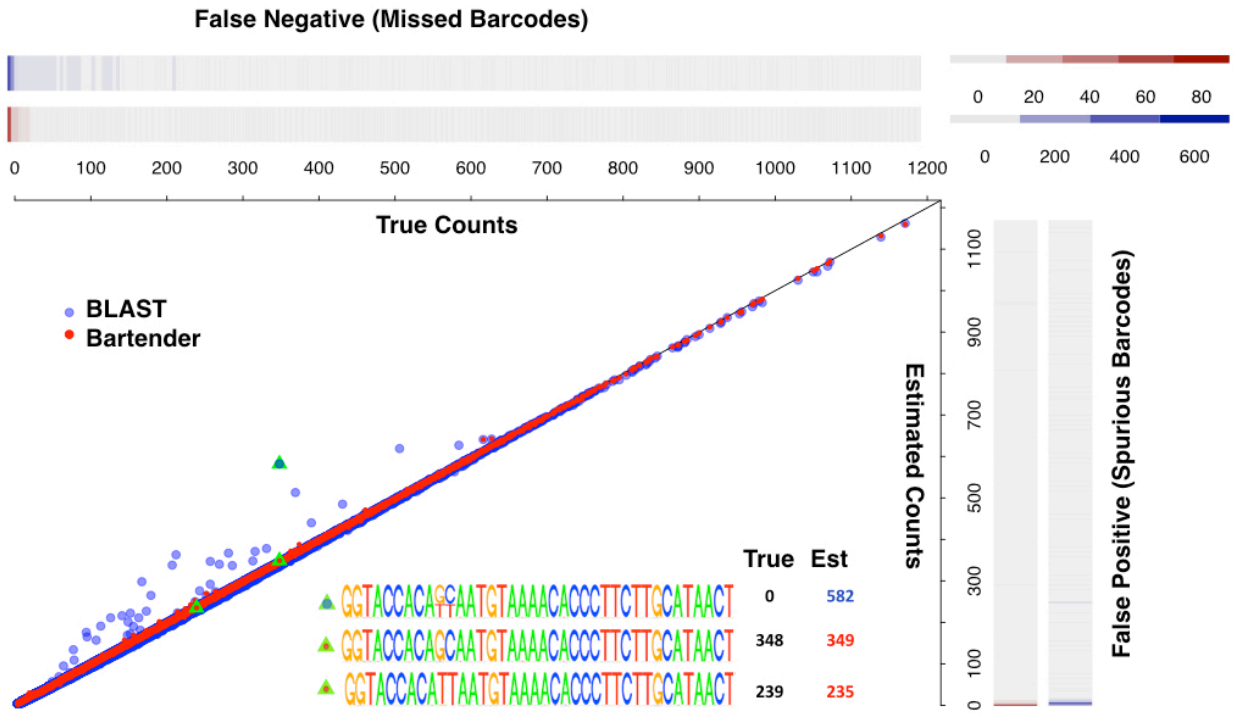


Figure 20 Comparison of BLAST and Bartender on simulation data

Comparison of Bartender and BLAST on simulated data. A scatter plot of the estimated and true counts of each barcode cluster for Bartender (red) and BLAST (blue). Over-clustering by BLAST but not Bartender results in some unique barcodes being merged. Green triangles and inset of position weight matrices show a representative example of BLAST over-merging of barcodes that are two mismatches away from each other. Heatmaps show the rate of false positives and false negatives for Bartender (red) and BLAST (blue).

To make the comparison complete, the barcode frequency distributions recovered by blast and bartender are compared to the true distribution respectively. The two-sample Kolmogorov-Smirnov(KS) test indicates that the frequency distribution from the blast method is significantly different from the truth with test statistics 0.013966 and p-value less than 0.0001, while we cannot reject the null hypothesis that the frequency distribution estimated by bartender is consistent with the truth given the test statistics is 0.0029 and p-value equals 0.8167.

3.3 Performance on real data

To compare Bartender and BLAST performance on real sequencing data, we next clustered published data of a high-complexity barcode library (~0.5M barcodes, 20 random bases, 26 total bases) that has been sequenced deeply (~136M reads) to generate ~3M unique barcode reads (Levy, Blundell et al. 2015). This barcode library is generated from a random primer and, in contrast with simulated data, some barcodes are by chance longer or shorter than the expected 26 bases. Bartender uses a comparison metric based on the Hamming distance that does not allow comparisons between barcodes of different lengths, and instead clusters barcodes of different lengths independently (see

Miscellaneous). In contrast, BLAST uses a comparison metric based on the Levenshtein distance that allows barcodes of different lengths to be compared directly. This difference complicates comparisons between Bartender and BLAST, because BLAST cluster centers (the consensus nucleotide at each position) can be influenced by "frame-shifts" where reads with an additional or missing base at one position causes a misalignment in many other positions. Therefore, to maximize the overlaps between Bartender and BLAST results, we redefine the BLAST cluster center as the most frequent unique read within each cluster. The two methods discovered 488,983 overlapping barcode clusters, with 2602 additional clusters identified by

Bartender and 22 by BLAST (Figure 21 (A)). The 2602 Bartender-only clusters typically contained numerous reads (2301 clusters with more than 10 reads), indicating that these are likely to be true barcodes that BLAST erroneously merged with another barcode (Figure 21 (B)). In contrast, the 22 BLAST-only clusters were only present at low frequencies, suggesting they likely arose from read errors. The overlapped clusters between BLAST and Bartender (Figure 21 (B)) reveals that the two methods generally correlate well (Figure 21 (B)and Figure 22, Pearson' correlation = 0.9723). For many barcodes, however, BLAST estimated higher counts than Bartender, echoing findings on simulated data (Figure 20), and suggesting BLAST over-merging. To investigate this possibility, we examined these outliers and found that, in most cases, BLAST merged two or more distinct barcodes together. A representative example of BLAST over-merging is shown in Figure 21(4C) and Figure 21(4D) (grey triangles): several barcodes with similar counts that are close in sequence space can be distinguished by Bartender but not BLAST. Similar speed improvements found with simulated data are found with this real data set (Bartender = 5.4 minutes, BLAST = 337 minutes, 4 threads) and the detailed running time of Bartender is given by Figure 23. The sequence error estimated by Bartender is 0.033, which is very close to the estimated error by Bartender extractor (0.036).

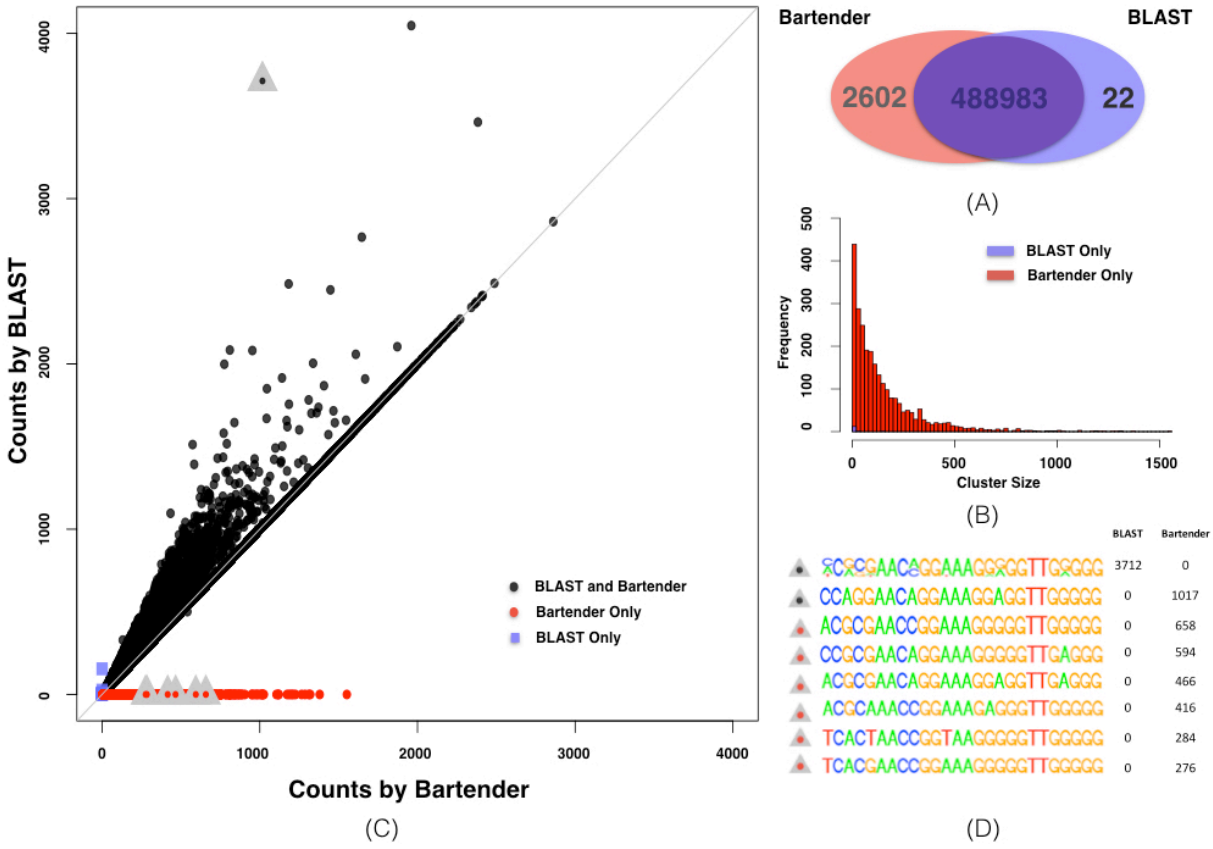


Figure 21 Performance on real data

(A) summarizes the overlap between blast and bartender results. Bartender results almost covers blast result (only 22 low frequency clusters left from blast). And about 2600 extra clusters are identified by bartender. The frequency distribution of non-overlapped clusters/barcodes is shown in (B). The 22 clusters presented only in blast result are in low frequency and only one reaches 150. Bartender's unique clusters has about 1737 clusters that the size is over 50. The unique clusters by bartender mainly comes from two categories. The first category contains all true barcodes that are missed by blast due to aggressive merging. The second category is consisted of barcode variations that could not be grouped to the true barcodes either due to indels errors or merging errors made by bartender. (C) zooms into the cluster size ranging between 0 to 4000, which covers most of the clusters. The scatter plot has similar pattern with simulation data (figure 3). One representative cluster (in grey triangle) is selected to illustrate the difference between blast and bartender on how similar barcodes are treated in figure (D). Blast merges 6 similar barcodes with comparable size, while bartender distinguishes them very well. The first cluster is generated by BLAST, is variable at multiple nucleotide positions, matches the sequence of the first Bartender cluster (second black dot), and incorporates many additional unique clusters that are distinguished by Bartender (red dots). All Bartender clusters display low variation at each nucleotide position.

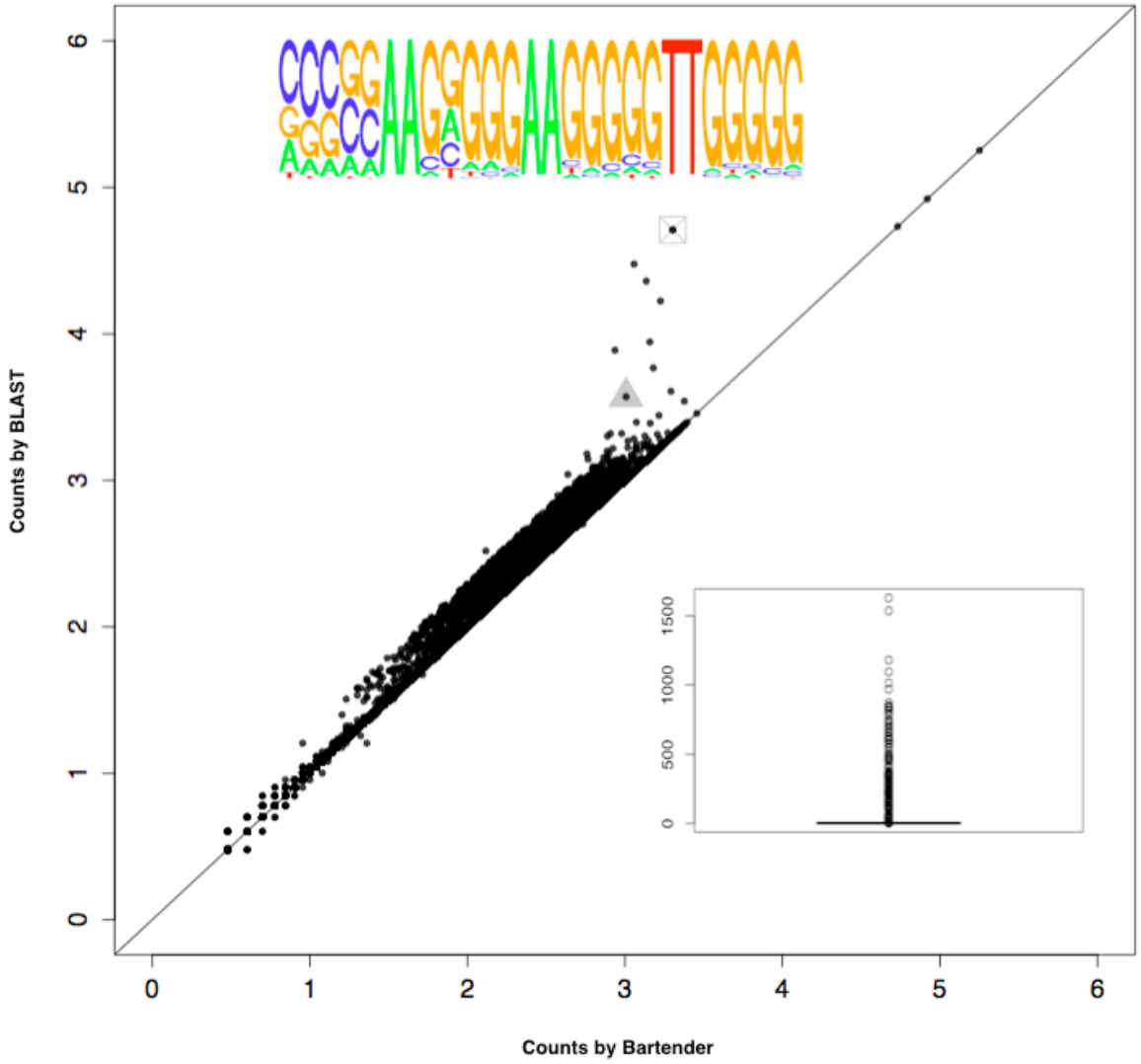


Figure 22 Complete scatter plot on real data

X-axis is the cluster size (log 10 scale) in Bartender and Y-axis is the cluster size (log 10) in BLAST. Most of points lie on the diagonal line. And a certain portion of points lie above the diagonal line, which is similar to what we observed in simulation data (Figure 20). The grey triangle is the one highlighted by Figure 21.

Another more extreme point is selected to show the over-merging errors in BLAST. The position weight matrix shows that there are at least 7 heterogeneous positions. The boxplot below the line summarizes the distribution of unique reads frequency. The most frequent reads in this cluster is only 1630 while the estimated cluster size by BLAST is 51180. There are 115

unique reads that the frequency is above 100. To sum up, this giant cluster is primarily consisted of unique barcodes with relative comparable size and no dominant barcode is found.

Since this dataset has about 136M reads, Bartender spent most of time on loading the extracted barcode and tabulate the unique reads. The actually wall-lock time on clustering process takes very small portion of the overall running time (Figure 23). When seed length reaches 5, the efficiency gains by increasing seed length almost disappear, which indicates seed length 5 should be a good parameter for this dataset considering the balance between accuracy and speed discussed in section Seeds selection and binning. The z-value in bartender was set to 5 for all running instance. And the distance is set to 3.

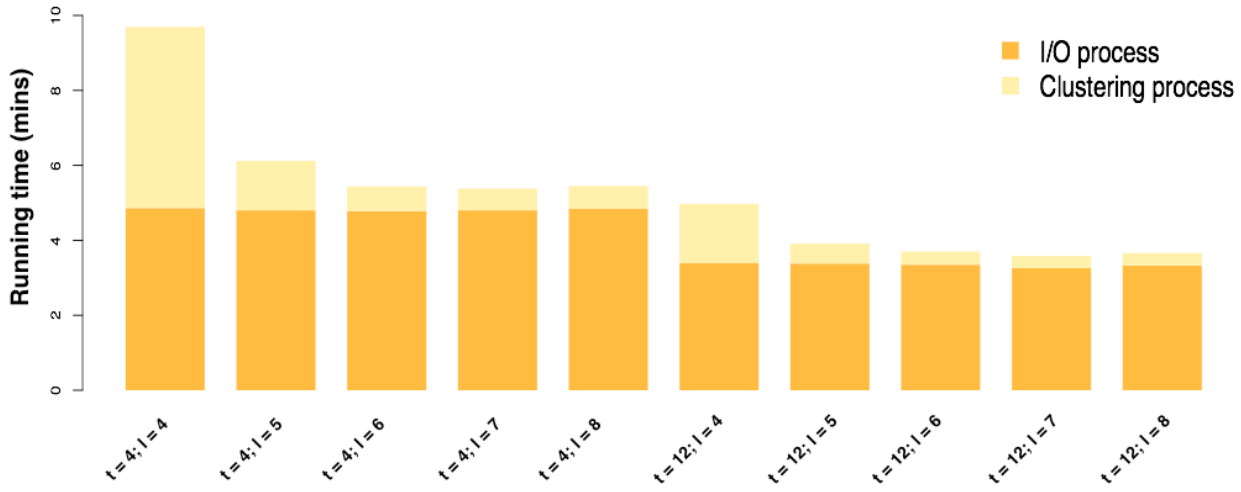


Figure 23 Running time of Bartender on real data

Here, t means number of threads and l means seed length. This experiment runs on the same laptop used in efficiency comparison above (page 64). Two thread settings are explored, which shows that the clustering process indeed becomes faster when seed length is 4.

3.4 Evaluation on multiple sample merging strategy

3.4.1 Simulation Setting

To validate the merging strategy on the multiple time points situation, another simulation dataset is generated and the parameters were adopted from estimates in real data. Since the goal of this experiment is to evaluate the merging strategy, the simulation assumes that there is no mutation occurring at any time point and each lineage tag has a fixed growth rate throughout all generations. No spacer and flank sequence are present in barcode because these constant regions have little impact on the effectiveness of merging strategy. The simulation process mimics the real experiment and sequencing step using two separate stages at each time point. The first stage simulates the cell growth and bottleneck caped by the medium saturation size. In this stage, each lineage grows based on the its growth rate and its actual population size follows the Poisson distribution with mean equals to the normalized frequency caped by the medium saturation size. The second step is sequencing, which is simulated by a Poisson distribution with mean equals to the actual population size. Table 7 shows all relevant parameters in this simulation.

Growth and bottleneck: Growth is performed in discrete generations where the population of cells in the $t+1$ generation $N_{i,t+1}$ follows Poisson distribution with $\lambda = \frac{N_{i,t}{}^{*}(1+g_i)^{*}N_s}{\sum_i^N N_{i,t}{}^{*}(1+g_i)}$ assuming that the whole population has already saturated the whole medium from the initial generation.

Sequence: to simplify the sequence step, the simulation assumes that the number of cell sequenced cell for a specific lineage follows the Poisson distribution with $\lambda = N_{i,t}$. And the sequence process is purely random and only mismatches occurs.

Table 7 Multiple time point simulation parameters

Parameters	Symbol	Value
Number of Lineages	N	100,000
Barcode length	L	20
Generations of evolution	T	113
Number of lineage tags with growth advantage	-	5,000
Initial population size per barcode	-	$Poi(100)$
Sequence error	e	0.02
Distribution of growth rate (fitness coefficient)	-	Truncated Exponential Distribution $\lambda = 0.00083$ Bounded by $(0.05, 0.15)$
The growth rate of lineage i		g_i
Population size of lineage i at time point t	$N_{i,t}$	Poisson distribution with $\lambda = \frac{N_{i,t\text{-}1}*(1+g_i)*N_s}{\sum_i^N N_{i,t\text{-}1}*(1+g_i)}$
Number of actual sequenced cells for lineage i at time point t	$N_{i,t}^s$	Poisson distribution with $\lambda = N_{i,t}$
Saturation population size	N_s	10,000,000
Number of reads at each generation	N_r	~10,000,000

To analyze the multiple time points simulated data, each time point is first processed by bartender. The parameters for each single run is covered by Table 8. Then the merging strategy is used to combine the clustering results. The merging strategy is run with z-value 5 and frequency cutoff 1.

Table 8 Bartender parameters in multiple time points simulation

Parameters	Value
Seed length	5
Overlap between adjacent seeds	4
Hamming distance	3
Z-value	5

Over time, higher fitness lineages will expand and drive lower fitness lineages to low frequencies and eventual extinction. And we observed that all lineages start with roughly same size and all lineage tags without growth advantage begin to die out from the 73rd generation and completely extinguish at 81generation. All 5000 lineages with growth advantage co-exist for a long stretch of generations after 81st generation and we stop at 113th generation, which covers all the interesting evolution stages. To examine the effectiveness of algorithm, the selected generations are 0, 19, 49, 59, 64, 69, 72 - 82, 84, 86, 90, 92, 96, 104 and 112, which zooms in the generations between 72 and 92 to capture the evolution dynamics and how many low frequency lineages are lost in the clustering result.

3.4.2 Experiment results

Bartender detected 100389 barcode clusters including all true barcodes (no false negatives) and 390 (one cell lineage has zero frequency at initial generation due to randomness)

erroneous barcodes (false positives). False positives were only detected in the first time point and no subsequent time points, and would be easily identified and ignored by an investigator. Bartender was extremely accurate at estimating barcode frequencies over time, but accuracy suffered when lineages fell to low frequencies of ~2-3 reads per barcode. To exemplify this point, we sampled the barcode frequencies often during the mass extinction event that begins at ~70 generations in our simulation (Figure 24). During this time, we find that Bartender underestimates the number of barcode lineages that are still present in the pool. As discussed above, these errors are generally due to PCR and sequencing errors in low frequency lineages that cause Bartender to miscall barcode cluster centers. When the number of lineages becomes stable (5000 lineages at 81 generations), Bartender again finds all barcodes that exist in the pool.

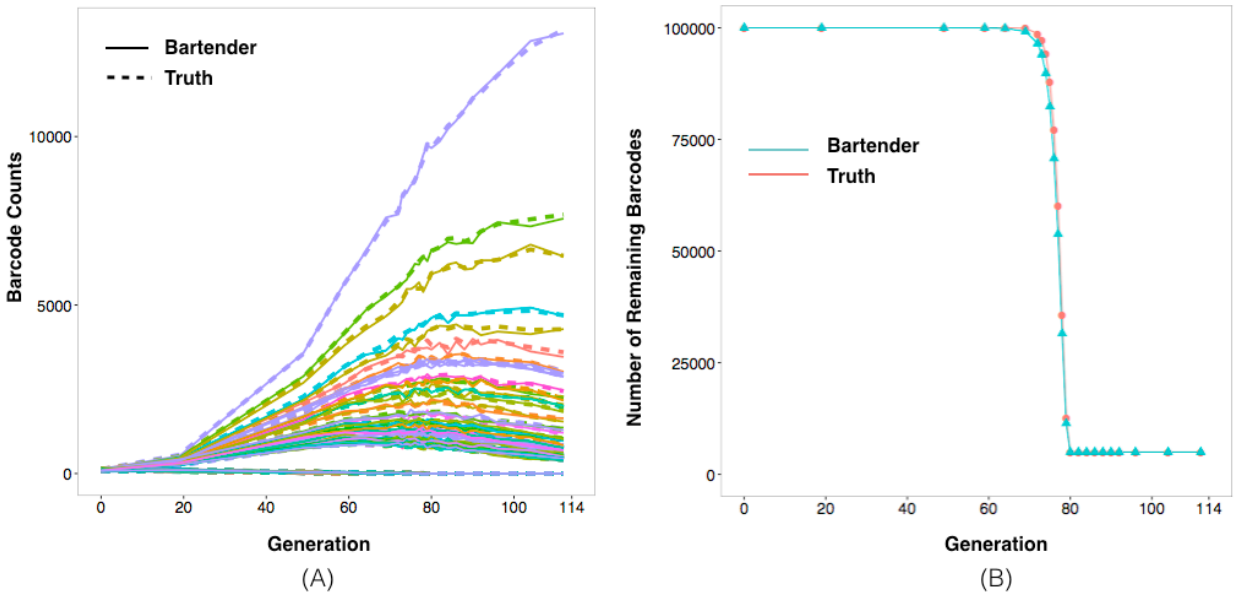


Figure 24 Performance of merging strategy on simulation data

Bartender performance on time course data. A simulation was performed of 100,000 barcoded cells with different fitness coefficients that are evolved in competition for 112 generations (A) Lineage trajectories of 1000 randomly selected barcodes (colors). Solid lines are trajectories estimated by Bartender and dashed lines are the true trajectories. (B) The number of barcodes present in the pool at a count greater than 1 (red) and the number detected by Bartender (turquoise). At ~70 generations, lineages without a growth advantage begin to go extinct. During the following mass extinction, Bartender slightly underestimates the number of barcodes present.

3.5 Bartender errors and sequencing depth

Barcodes with low sequencing depths are difficult to accurately count for several reasons. First, sampling noise (derived from which molecules happen to be sequenced) scales with the square root of the expected number of reads. That is, barcodes with less reads will have high coefficients of variation simply because of sampling on the sequencer. Second, small read numbers can often result in an erroneous cluster center (the predicted barcode sequence does not match the true barcode sequence) because PCR or sequencing errors in one read will have a large impact on the predicted center. These errors in small clusters will often result in both false positives and false negatives because reads matching true barcode sequence are missing while those matching a new (erroneous) barcode sequence are present. Third, merging decisions between small clusters become less accurate. As described above, Bartender takes advantage of cluster size information to make merging decisions. An unequal frequency between clusters favors merging because it becomes more likely that the low frequency cluster represents a sequence error of the high frequency cluster. However, when a barcode is present at low frequencies, both true reads and errors are expected to have few counts, so differences in frequency become blunted and less informative.

3.5.1 Experimental approach

This section is trying to give some insights on how to determine the sequence depth such that bartender could adjust the sampling and sequencing error to recover most of the true lineages. To answer this question, the error proportion introduced by bartender with respect to the overall error is measured in both simulation and real data.

To make the problem simpler, this study only considers the three most primary errors including sampling error, sequencing error and error introduced by bartender. The sampling error

refers the size difference of each barcode lineage between the actual sequenced copy number and the theoretically expected sequence copy number. Sequence error refers only include the mismatched introduced in the sequence step. Lastly, error by Bartender is defined as the size difference between the estimated and the actual sequence copy number.

To evaluate these three types of errors, the whole dataset is randomly partitioned into 15 subsets. Bartender is applied to analyze each partition and the result of each partition is matched to the ground truth by the cluster center. Since there is no ground truth for the real data, the clustering result on the whole dataset is used as the truth.

Since each reads are sampled individually, the cluster size in each partition follows the Binomial distribution (Equation 3-1). And the expected barcode count and standard deviation are given by equation (Equation 3-2) and (Equation 3-3).

$$f_p \sim Bin(F_w, \frac{1}{n}) \qquad \textit{Equation 3-1}$$

where F_w is the size of the corresponding cluster in whole dataset.

So the expected cluster size in one partition is given by

$$\mathrm{E}[f_p] = \mu = {}^{F_w}/_{n} \qquad \textit{Equation 3-2}$$

And the standard deviation in one partition is given by

$$\sigma_p = \sqrt{F_w \frac{1}{n} * \frac{n\text{-}1}{n}} \qquad \textit{Equation 3-3}$$

The coefficient of variation (CV) (Equation 3-3) is selected as the error measurement for each error source since it is an error measurement independent of the barcode lineage size. The estimated CV is calculated based on equation (Equation 3-5) for three errors in an accumulative way.

$$CV = \frac{\sigma}{\mu} \qquad \textit{Equation 3-4}$$

$$\widehat{CV} = \frac{s}{\bar{x}} \qquad \textit{Equation 3-5}$$

where $s = \sum_{i=1}^{n} \frac{(f_i - \bar{f})^2}{n-1}$ and $\bar{x} = \bar{f} = \frac{\sum_{i=1}^{n} f_i}{n}$.

f_i is the observed frequency of each cluster in partition i. $\bar{x} = \bar{f} = \frac{\sum_{i=1}^{n} f_i}{n}$

3.5.2 Experiment results

In both cases (Figure 25), Bartender clustering introduced almost no errors for large clusters (>10 reads). Counting errors for these large clusters are due almost exclusively to sampling at the sequencer. For smaller clusters (≤10 reads), the clustering process does introduce additional error for reasons described above. However, the additional error due to clustering is small, especially at the lower (and more realistic) ~0.33% sequencing error rate. Nevertheless, sequencing at a coverage sufficient to read each barcode at least 10 times would virtually eliminate Bartender errors.

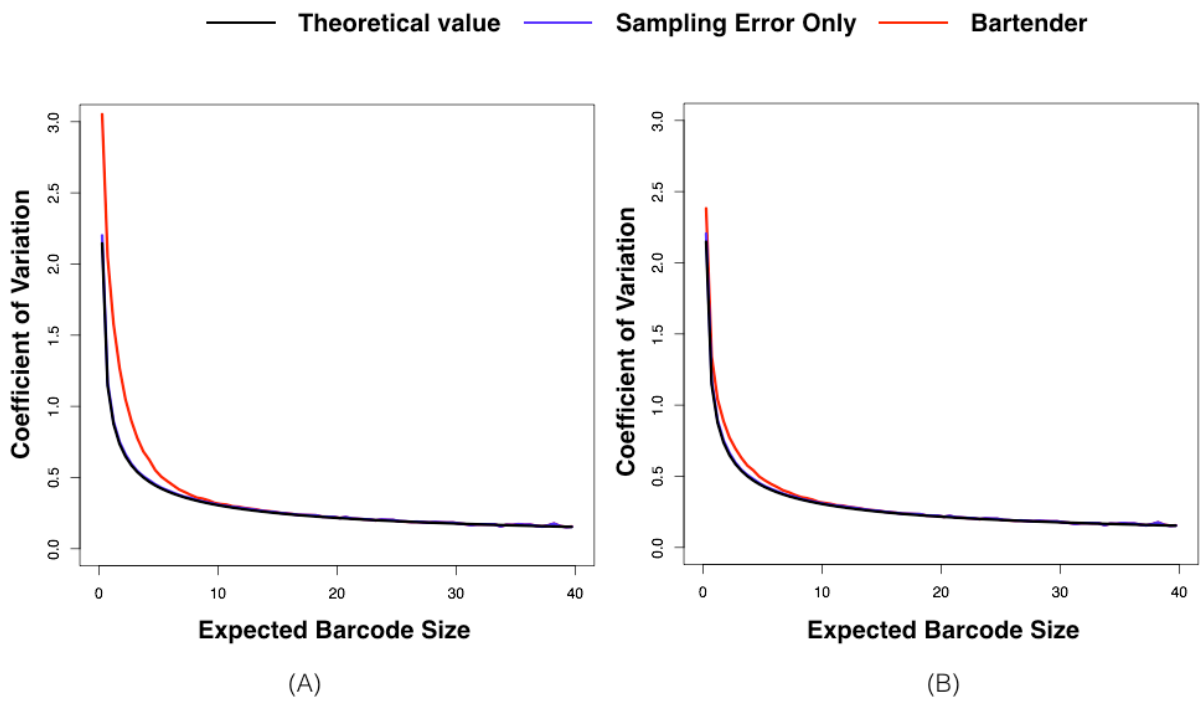


Figure 25 Coefficient of Variations of three types errors on simulation data

X-axis is the expected barcode count in each partition. Y-axis is the Coefficient of variation (CV) calculated by Equation 3-4. Black line is the theoretical CV of the expected barcode size, which follows the Binomial distribution. The blue line is the CV of the realized barcode size after sampling. The red line represents the accumulative error that contains the sampling error, sequencing error and errors introduced by Bartender. The impact of sequencing depth on Bartender performance. A plot of the barcode count by the coefficient of variation (CV) for that count on simulated data with a 2% (A) and 0.33% (B) combined error rate of PCR and sequencing error. The black lines are theoretical values, which follow the Binomial distribution. The blue lines are the CV of sampling (at the sequencer) alone, without sequencing errors or errors introduced by Bartender clustering. The red lines are the CV after running Bartender, and include sampling, sequencing, and clustering errors. All lines are smoothed with window size 0.5. To sum up, larger sequence error and longer barcode poses more challenges on recovering low frequency barcode.

We also test bartender on the real data using the same approach described above. Since there is no ground truth for real data, the clustering result on whole dataset is considered as the "ground truth". Figure 26 shows the results by applying the approach on this dataset. The three lines overlap with each other very well. Bartender does not introduce extra error on top of sampling error, which indicates the real data is in high quality and bartender works well on this dataset. Same bin size and software parameters are applied to the data.

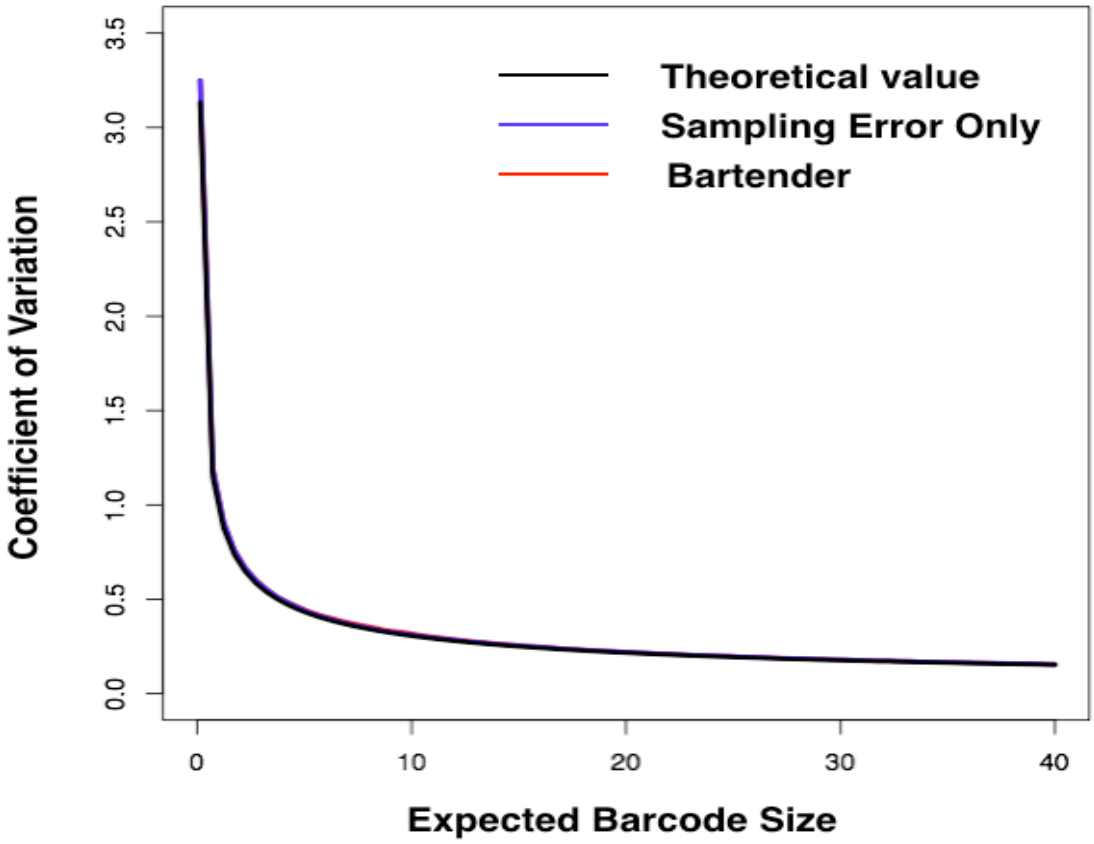


Figure 26 CV plot on real data

The single time point simulation data and the same real dataset is used in this study. This experiment suggests that the idea sequence depth is related with two factors. One is the sequence error. Low sequence error (high quality data) requires less sequence depth in order to have an accurate clustering result from bartender, which means that high quality data saves experiment cost. The other factor is the barcode length. Longer barcode needs larger sequencing depth in order to recover the low frequency barcodes as much as possible based on the fact that the probability that the barcode is not mutated decreases exponentially with respect to barcode length.

4 Discussion and future work

4.1 Discussion

To the best of our knowledge, Bartender is the first general clustering algorithm for accurate and fast counting of barcode and amplicon reads. Bartender's accuracy stems from a new statistical test schema that uses both nucleotide sequence and cluster size information to prevent over-merging; its speed stems from a novel binning strategy and a computationally efficient greedy clustering algorithm. Bartender includes handling of both UMIs and time course data, and promises to be a useful tool for a large number of diverse applications.

Bartender includes three parameters that are tunable for different applications: the sequencing error rate (e), the seed length (l, see Chapter 3), and the merging threshold (z, see Chapter 2). We recommend using the empirically determined e (an output of Bartender extractor) for all applications, and the default seed length (l = 5) for applications where shorter run times are not a priority. The remaining parameter, z, should be set according to the expected coverage per barcode and the barcode library complexity. For low to medium coverage (<500 reads/barcode), we recommend starting with the default setting for z (= 5). However, in some cases, it may be necessary to adjust the z. For example, we have noticed some nucleotide sequences in our random barcode libraries are more prone to PCR or sequencing errors (i.e. errors at that position are greater than e), and, if these errors occur within an abundant barcode, they may cause the same erroneous read to occur at a high frequency and therefore be interpreted as an independent barcode cluster. At extremely high coverage (e.g. >10,000 reads/barcode in a low-complexity barcode library), this problem is amplified because a high error position in any barcode will create this artifact. In this case, we recommend setting z higher. In cases where all barcodes are expected to be distant (averagely 5-6 mismatches from a nearest neighbor), we

recommend disabling the merging threshold (z = -1) to make merging decisions based on cluster distance only, i.e., all sequences within the user specified distance of each other will be merged, since this scheme is unlikely to merge any real barcodes together (over-merging) and will remove the possibility of nucleotide sequences with abnormally high errors causing artifacts. We note that Bartender will only perform well if most barcodes within the pool are sufficiently spaced such that they are at least 3-4 mismatches away from a nearest neighbor (see (Blundell and Levy 2014, Levy, Blundell et al. 2015) for a discussion on random barcode design). In cases where this rule is broken (Bhang, Ruddy et al. 2015), Bartender performance will be encumbered, as will any existing clustering approach.

Bartender speed is mainly due to the fact that it partitions unique reads into bins and restrains sequence comparisons within a bin. We use the entropy at each nucleotide position to prioritize seeds that have the highest probability of creating a maximal number of bins, and thereby minimizing the number of pairwise sequence comparisons. Barcode designs with regions of varying complexity (e.g. (Goodman, Church et al. 2013, Kosuri, Goodman et al. 2013, McKenna, Findlay et al. 2016)) could potentially disrupt this process and greatly slow Bartender, for example, for a barcode that is split into low complexity (e.g. 1000 random variants) and high complexity (e.g. 1M random variants) regions. Because a nucleotide in either region would be expected to have high entropy, Bartender may choose initial seeds that contain only low complexity nucleotides resulting in far fewer bins. One potential solution would be to use a seed selection protocol that considers associations between nucleotides (e.g. information gain or mutual information), however this is not implemented here as this type of design should be rare.

Bartender removes PCR duplicates by simply searching for repeated UMIs within each cluster (exact matches) and removing these from the counts. Because it searches only for exact UMI matches, a PCR or sequencing error that happened to occur in a repeated UMI would not be recognized as a repeat and thus result in over-counting of that cluster. However, the alternative, merging UMI with similar sequences, raises greater problems. Large clusters may contain many UMIs, and, because UMIs are generally short, UMI clustering would erroneously merge many distinct UMIs that are close in sequence. Even using our exact match criterion, it is possible that extremely large barcode clusters will begin to use up all available UMIs resulting in under-counting. For example, a barcode that is read 100,000 times and contains an 8mer UMI (4^8 = ~65,000 possible sequences) will necessarily have UMI repeats even when each sequenced read stems from a unique template molecule. To avoid these problems, we recommend selecting a UMI length that results in at least 10-fold more possible sequences than the largest expected cluster.

The multiple time point mode is designed to generate the trajectory for an evolution process, which naturally put the constraint that the lineage at any time point (generation) is the result of the previous time point (generation) assuming that the diversity of cell lineage does not increase along the biological process. This mode could then be applied to other non-time point data if the data confirms this assumption. To capture the crucial near-zero point (generation), i.e. the extinction point of most lineage tags, and alleviate the miscall error observed in extremely low frequency lineage tag, Bartender matches the cluster of current time point to previous time point by allowing at most one mismatch. For data with general multiple experiment conditions that do not satisfy such subset connection, they need to be handled on a case-by-case basis. If there are only few (2-3) conditions and large overlap in barcodes is expected between conditions,

the multiple time point mode can be directly applied by assigning an arbitrary sequence to conditions. However, in cases where the overlap between conditions is expected to be small, the multiple time point mode will not be beneficial and the regular mode of Bartender can be applied to each condition separately to obtain clustering results for subsequent analyses.

Lastly, based on our experience, Bartender will work well with a standard laptop or desktop computer for most applications. Available memory (RAM) is generally the limiting factor for Bartender processing, and the necessary RAM is a function of the number of unique barcode reads and the barcode length. We recommend 4-6 GB RAM for datasets with less than 1M unique 40mer barcodes, and 8-10 GB RAM for datasets with less than 3M unique 60mer barcodes.

4.2 Future work

Although Bartender is ultrafast and generates accurate result on the simulation data and published real data, there are still some characteristics of bar-seq data is not well handled by Bartender.

For low complexity barcode with extremely high coverage, the test will fail because it is quite easy for a high frequent lineages generate large artifacts and merging these artifacts with its true cluster will always be rejected by the statistical test proposed above. So a more robust statistical test that considering the sequence coverage and barcode complexity library should be developed in order to handle these case. One way to achieve it is to use bootstrap and resize high frequent clusters under the scale that the current statistical test has the most power.

We tested Bartender on some unpublished double barcode data, which refers the paired-end read and both ends have valid barcodes inside. We notice that some positions have much

higher error rate than other position due to the barcode template. So it might be a good idea to give each nucleotide position with different weight based on the sequence error profile when measuring the similarity including the sequence similarity and the test statistic test. However, it is very difficult to estimate the sequence error from the raw data unless the sequence platform could provide these statistics. One thing could be done on Bartender extractor. That is, Bartender extractor could calculate the error rate on several discrete positions on fixed region by allowing mismatches on fixed regions. And the smoothing error rate between positions interval could be used as the error rate for the positions between those positions with estimated error rate.

As discussed above, picking up high diversity seeds is very challenge problem, which is quite computational expensive to find the optimal seed sequence. The goal is to find a set of positions in the raw sequence such that the raw sequence could be distributed into buckets as even as possible. Mathematically, it could be formulized as an optimization problem, which could be potentially an interesting research direction.

5

Reference

Altschul, S. F., W. Gish, W. Miller, E. W. Myers and D. J. Lipman (1990). "Basic local alignment search tool." Journal of Molecular Biology **215**(3): 403-410.

Andonovska, B., C. Dimova and S. Panov (2008). "Matrix metalloproteinases (MMP-1, -8, -13) in chronic periapical lesions." Vojnosanit Pregl **65**(12): 882-886.

Austin, S., M. Ziese and N. Sternberg (1981). "A novel role for site-specific recombination in maintenance of bacterial replicons." Cell **25**(3): 729-736.

Bassik, M. C., R. J. Lebbink, L. S. Churchman, N. T. Ingolia, W. Patena, E. M. LeProust, M. Schuldiner, J. S. Weissman and M. T. McManus (2009). "Rapid creation and quantitative monitoring of high coverage shRNA libraries." Nat Methods **6**(6): 443-445.

Bhang, H. E., D. A. Ruddy, V. Krishnamurthy Radhakrishna, J. X. Caushi, R. Zhao, M. M. Hims, A. P. Singh, I. Kao, D. Rakiec, P. Shaw, M. Balak, A. Raza, E. Ackley, N. Keen, M. R. Schlabach, M. Palmer, R. J. Leary, D. Y. Chiang, W. R. Sellers, F. Michor, V. G. Cooke, J. M. Korn and F. Stegmeier (2015). "Studying clonal dynamics in response to cancer therapy using high-complexity barcoding." Nat Med **21**(5): 440-448.

Blundell, J. R. and S. F. Levy (2014). "Beyond genome sequencing: lineage tracking with barcodes to study the dynamics of evolution, infection, and cancer." Genomics **104**(6 Pt A): 417-430.

Boross, G., K. Orosz and I. J. Farkas (2009). "Human microRNAs co-silence in well-separated groups and have different predicted essentialities." Bioinformatics **25**(8): 1063-1069.

Brassart, B., P. Fuchs, E. Huet, A. J. Alix, J. Wallach, A. M. Tamburro, F. Delacoux, B. Haye, H. Emonard and W. Hornebeck (2001). "Conformational dependence of collagenase (matrix metalloproteinase-1) up-regulation by elastin peptides in cultured fibroblasts." Journal of Biological Chemistry **276**(7): 5222-5227.

Bruchova, H., M. Merkerova and J. T. Prchal (2008). "Aberrant expression of microRNA in polycythemia vera." Haematologica **93**(7): 1009-1016.

Brutinel, E. D. and J. A. Gralnick (2012). "Anomalies of the anaerobic tricarboxylic acid cycle in Shewanella oneidensis revealed by Tn-seq." Mol Microbiol **86**(2): 273-283.

Carette, J. E., M. Raaben, A. C. Wong, A. S. Herbert, G. Obernosterer, N. Mulherkar, A. I. Kuehne, P. J. Kranzusch, A. M. Griffin, G. Ruthel, P. Dal Cin, J. M. Dye, S. P. Whelan, K. Chandran and T. R. Brummelkamp (2011). "Ebola virus entry requires the cholesterol transporter Niemann-Pick C1." Nature **477**(7364): 340-343.

Chu, G., J. Li, B. Narasimhan, R. Tibshirani and V. Tusher (2001). "Significance Analysis of Microarrays Users Guide and Technical Document."

Edelstein, L. C. and P. F. Bray (2011). "MicroRNAs in platelet production and activation." Blood **117**(20): 5289-5296.

Edelstein, L. C., S. E. McKenzie, C. Shaw, M. A. Holinstat, S. P. Kunapuli and P. F. Bray (2013). "MicroRNAs in platelet production and activation." J Thromb Haemost **11 Suppl 1**: 340-350.

Efron, B., and Tibshirani, R. J. (1994). "An Introduction to the Bootstrap." CRC press **57**.

Fisher, R. A. (1914). "Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population." Biometrika **10**: 507-521.

Fisher, R. A. (1924). "The Distribution of the Partial Correlation Coefficient." Metron **3**: 4.

Fisher, R. A. (1924). "The Distribution of the Partial Correlation Coefficient." Metron **3**: 329-332.
Fu, W. J. (1998). "Penalized Regressions: The Bridge versus the Lasso." Journal of Computational and Graphical Statistics **7**(3): 397-416.
Gawronski, J. D., S. M. Wong, G. Giannoukos, D. V. Ward and B. J. Akerley (2009). "Tracking insertion mutants within libraries by deep sequencing and a genome-wide screen for Haemophilus genes required in the lung." Proc Natl Acad Sci U S A **106**(38): 16422-16427.
Gennarino, V. A., M. Sardiello, R. Avellino, N. Meola, V. Maselli, S. Anand, L. Cutillo, A. Ballabio and S. Banfi (2009). "MicroRNA target prediction by expression analysis of host genes." Genome Res **19**(3): 481-490.
Giaever, G., A. M. Chu, L. Ni, C. Connelly, L. Riles, S. Veronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. Andre, A. P. Arkin, A. Astromoff, M. El-Bakkoury, R. Bangham, R. Benito, S. Brachat, S. Campanaro, M. Curtiss, K. Davis, A. Deutschbauer, K. D. Entian, P. Flaherty, F. Foury, D. J. Garfinkel, M. Gerstein, D. Gotte, U. Guldener, J. H. Hegemann, S. Hempel, Z. Herman, D. F. Jaramillo, D. E. Kelly, S. L. Kelly, P. Kotter, D. LaBonte, D. C. Lamb, N. Lan, H. Liang, H. Liao, L. Liu, C. Luo, M. Lussier, R. Mao, P. Menard, S. L. Ooi, J. L. Revuelta, C. J. Roberts, M. Rose, P. Ross-Macdonald, B. Scherens, G. Schimmack, B. Shafer, D. D. Shoemaker, S. Sookhai-Mahadeo, R. K. Storms, J. N. Strathern, G. Valle, M. Voet, G. Volckaert, C. Y. Wang, T. R. Ward, J. Wilhelmy, E. A. Winzeler, Y. Yang, G. Yen, E. Youngman, K. Yu, H. Bussey, J. D. Boeke, M. Snyder, P. Philippsen, R. W. Davis and M. Johnston (2002). "Functional profiling of the Saccharomyces cerevisiae genome." Nature **418**(6896): 387-391.
Gibney, P. A., C. Lu, A. A. Caudy, D. C. Hess and D. Botstein (2013). "Yeast metabolic and signaling genes are required for heat-shock survival and have little overlap with the heat-induced genes." Proc Natl Acad Sci U S A **110**(46): E4393-4402.
Gnatenko, D. V., L. D. Cupit, E. C. Huang, A. Dhundale, P. L. Perrotta and W. F. Bahou (2005). "Platelets express steroidogenic 17beta-hydroxysteroid dehydrogenases. Distinct profiles predict the essential thrombocythemic phenotype." THROMBOSIS AND HAEMOSTASIS-STUTTGART- **94**(2): 412.
Gnatenko, D. V., L. D. Cupit, E. C. Huang, A. Dhundale, P. L. Perrotta and W. F. Bahou (2005). "Platelets express steroidogenic 17beta-hydroxysteroid dehydrogenases. Distinct profiles predict the essential thrombocythemic phenotype." Thromb Haemost **94**(2): 412-421.
Gnatenko, D. V., W. Zhu, X. Xu, E. T. Samuel, M. Monaghan, M. H. Zarrabi, C. Kim, A. Dhundale and W. F. Bahou (2010). "Class prediction models of thrombocytosis using genetic biomarkers." Blood **115**(1): 7-14.
Goodman, D. B., G. M. Church and S. Kosuri (2013). "Causes and effects of N-terminal codon bias in bacterial genes." Science **342**(6157): 475-479.
Goren, A., F. Ozsolak, N. Shoresh, M. Ku, M. Adli, C. Hart, M. Gymrek, O. Zuk, A. Regev, P. M. Milos and B. E. Bernstein (2010). "Chromatin profiling by directly sequencing small quantities of immunoprecipitated DNA." Nat Methods **7**(1): 47-49.
Gresham, D., V. M. Boer, A. Caudy, N. Ziv, N. J. Brandt, J. D. Storey and D. Botstein (2011). "System-level analysis of genes and functions affecting survival during nutrient starvation in Saccharomyces cerevisiae." Genetics **187**(1): 299-317.
Griffiths-Jones, S., H. K. Saini, S. van Dongen and A. J. Enright (2008). "miRBase: tools for microRNA genomics." Nucleic Acids Res **36**(Database issue): D154-158.

Grimson, A., K. K. Farh, W. K. Johnston, P. Garrett-Engele, L. P. Lim and D. P. Bartel (2007). "MicroRNA targeting specificity in mammals: determinants beyond seed pairing." Mol Cell **27**(1): 91-105.
Gundry, M. and J. Vijg (2012). "Direct mutation analysis by high-throughput sequencing: from germline to low-abundant, somatic variants." Mutat Res **729**(1-2): 1-15.
Hamming, R. W. (1950). "Error Detecting and Error Correcting Codes." Bell System Technical Journal **29**(2): 147-160.
Han, T. X., X. Y. Xu, M. J. Zhang, X. Peng and L. L. Du (2010). "Global fitness profiling of fission yeast deletion strains by barcode sequencing." Genome Biol **11**(6): R60.
He, Y., L. Cao, M. Yang, M. Zhao, Y. Yu and W. Xu (2009). "[Role of WAVE1 in K562 leukemia cells invasion and its mechanism]." Zhonghua xue ye xue za zhi= Zhonghua xueyexue zazhi **30**(4): 237-241.
Hegreness, M., N. Shoresh, D. Hartl and R. Kishony (2006). "An equivalence principle for the incorporation of favorable mutations in asexual populations." Science **311**(5767): 1615-1617.
Herouy, Y., P. Mellios, E. Bandemir, S. Dichmann, P. Nockowski, E. Schöpf and J. Norgauer (2001). "Inflammation in stasis dermatitis upregulates MMP-1, MMP-2 and MMP-13 expression." Journal of dermatological science **25**(3): 198-205.
Hobbs, E. C., J. L. Astarita and G. Storz (2010). "Small RNAs and small proteins involved in resistance to cell envelope stress and acid shock in Escherichia coli: analysis of a bar-coded mutant collection." J Bacteriol **192**(1): 59-67.
Hotelling, H. (1936). "Relations between two sets of variates." Biometrika **28**(3-4): 321-377.
Huang, J. C., T. Babak, T. W. Corson, G. Chua, S. Khan, B. L. Gallie, T. R. Hughes, B. J. Blencowe, B. J. Frey and Q. D. Morris (2007). "Using expression profiling data to identify human microRNA targets." Nat Methods **4**(12): 1045-1049.
Huang, J. C., Q. D. Morris and B. J. Frey (2007). "Bayesian inference of MicroRNA targets from sequence and expression data." J Comput Biol **14**(5): 550-563.
Jayaswal, V., M. Lutherborrow, D. D. Ma and Y. H. Yang (2011). "Identification of microRNA-mRNA modules using microarray data." BMC Genomics **12**: 138.
Jing Xiang, S. K. (2013). A* Lasso for Learning a Sparse Bayesian Network

Structure for Continuous Variables. Advances in neural information processing systems**:** 2418--2426.
Kivioja, T., A. Vaharautio, K. Karlsson, M. Bonke, M. Enge, S. Linnarsson and J. Taipale (2012). "Counting absolute numbers of molecules using unique molecular identifiers." Nat Methods **9**(1): 72-74.
Kosuri, S., D. B. Goodman, G. Cambray, V. K. Mutalik, Y. Gao, A. P. Arkin, D. Endy and G. M. Church (2013). "Composability of regulatory sequences controlling transcription and translation in Escherichia coli." Proc Natl Acad Sci U S A **110**(34): 14024-14029.
Kozak, M. (Mar. 1983). "Comparison of Initiation of Protein Synthesis in procaryotes, eucaryotes, and organelles." MICROBIOLOGICAL REVIEWS **47**(1): 1-45.
Lane, W. J., S. Dias, K. Hattori, B. Heissig, M. Choy, S. Y. Rabbany, J. Wood, M. A. Moore and S. Rafii (2000). "Stromal-derived factor 1–induced megakaryocyte migration and platelet production is dependent on matrix metalloproteinases." Blood **96**(13): 4152-4159.
Le, T. D., L. Liu, A. Tsykin, G. J. Goodall, B. Liu, B. Y. Sun and J. Li (2013). "Inferring microRNA-mRNA causal regulatory relationships from expression data." Bioinformatics **29**(6): 765-771.

LeProust, E. M., B. J. Peck, K. Spirin, H. B. McCuen, B. Moore, E. Namsaraev and M. H. Caruthers (2010). "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process." Nucleic Acids Res **38**(8): 2522-2540.
Levenshtein, V. I. (1966). "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." Soviet Physics Doklady **10**.
Levy, S. F., J. R. Blundell, S. Venkataram, D. A. Petrov, D. S. Fisher and G. Sherlock (2015). "Quantitative evolutionary dynamics using high-resolution lineage tracking." Nature **519**(7542): 181-186.
Lewis, B. P., C. B. Burge and D. P. Bartel (2005). "Conserved Seed Pairing, Often Flanked by Adenosines, Indicates that Thousands of Human Genes are MicroRNA Targets." Cell **120**(1): 15-20.
Li, H., J. Ruan and R. Durbin (2008). "Mapping short DNA sequencing reads and calling variants using mapping quality scores." Genome Res **18**(11): 1851-1858.
Li, X., R. Gill, N. G. Cooper, J. K. Yoo and S. Datta (2011). "Modeling microRNA-mRNA interactions using PLS regression in human colon cancer." BMC Med Genomics **4**: 44.
Lu, R., N. F. Neff, S. R. Quake and I. L. Weissman (2011). "Tracking single hematopoietic stem cells in vivo using high-throughput sequencing in conjunction with viral genetic barcoding." Nat Biotechnol **29**(10): 928-933.
Maathuis, M. H., M. Kalisch and P. Bühlmann (2009). "Estimating high-dimensional intervention effects from observational data." The Annals of Statistics **37**(6A): 3133-3164.
Mardis, E. R. (2008). "The impact of next-generation sequencing technology on genetics." Trends Genet **24**(3): 133-141.
Masud Karim, S. M., L. Liu, T. D. Le and J. Li (2016). "Identification of miRNA-mRNA regulatory modules by exploring collective group relationships." BMC Genomics **17 Suppl 1**: 7.
McKenna, A., G. M. Findlay, J. A. Gagnon, M. S. Horwitz, A. F. Schier and J. Shendure (2016). "Whole organism lineage tracing by combinatorial and cumulative genome editing." Science.
Meyerhans, A., J.-P. Vartanian and S. Wain-Hobson (1990). "DNA recombination during PCR." Nucleic Acids Research **18**(7): 1687-1691.
Nguyen, L. V., D. Pellacani, S. Lefort, N. Kannan, T. Osako, M. Makarem, C. L. Cox, W. Kennedy, P. Beer, A. Carles, M. Moksa, M. Bilenky, S. Balani, S. Babovic, I. Sun, M. Rosin, S. Aparicio, M. Hirst and C. J. Eaves (2015). "Barcoding reveals complex clonal dynamics of de novo transformed human mammary cells." Nature **528**(7581): 267-271.
Noble, M., J. R. Treadwell, S. J. Tregear, V. H. Coates, P. J. Wiffen, C. Akafomo and K. M. Schoelles (2010). "Long-term opioid management for chronic noncancer pain." Cochrane Database Syst Rev(1): CD006605.
Parkhomenko, E., D. Tritchler and J. Beyene (2007). "Genome-wide sparse canonical correlation of gene expression with genotypes." BMC Proc **1 Suppl 1**: S119.
Pearson, K. (1915). "On the partial correlation ratio." Proceedings of the Royal Society of London. Series A **91**(632): 492-498.
Pearson, K. (1920). "Notes on the history of correlation. Being a paper read to the Society of Biometricians and Mathematical Statisticians, June 14, 1920." Biometrika **13**: 25-45.
Peng, J., P. Wang, N. Zhou and J. Zhu (2009). "Partial Correlation Estimation by Joint Sparse Regression Models." J Am Stat Assoc **104**(486): 735-746.
Pradervand, S., J. Weber, J. Thomas, M. Bueno, P. Wirapati, K. Lefort, G. P. Dotto and K. Harshman (2009). "Impact of normalization on miRNA microarray expression profiling." RNA **15**(3): 493-501.

Pradhan, K. (2009). Partial Correlation Analysis in Functional Brain Imaging Studies. PhD, Stony brook Univeristy.
Robinson, D. G., W. Chen, J. D. Storey and D. Gresham (2014). "Design and analysis of Bar-seq experiments." G3 (Bethesda) **4**(1): 11-18.
Saito, T. and N. W. Bunnett (2005). "Protease-activated receptors." Neuromolecular medicine **7**(1-2): 79-99.
Schafer, J. and K. Strimmer (2005). "An empirical Bayes approach to inferring large-scale gene association networks." Bioinformatics **21**(6): 754-764.
Schlabach, M. R., J. Luo, N. L. Solimini, G. Hu, Q. Xu, M. Z. Li, Z. Zhao, A. Smogorzewska, M. E. Sowa, X. L. Ang, T. F. Westbrook, A. C. Liang, K. Chang, J. A. Hackett, J. W. Harper, G. J. Hannon and S. J. Elledge (2008). "Cancer proliferation gene discovery through functional genomics." Science **319**(5863): 620-624.
Schmitt, M. W., S. R. Kennedy, J. J. Salk, E. J. Fox, J. B. Hiatt and L. A. Loeb (2012). "Detection of ultra-rare mutations by next-generation sequencing." Proc Natl Acad Sci U S A **109**(36): 14508-14513.
Schulze, H. and R. A. Shivdasani (2004). Molecular mechanisms of megakaryocyte differentiation. Seminars in thrombosis and hemostasis, Copyright© 2004 by Thieme Medical Publishers, Inc., 333 Seventh Avenue, New York, NY 10001, USA.
Schwarzmuller, T., B. Ma, E. Hiller, F. Istel, M. Tscherner, S. Brunke, L. Ames, A. Firon, B. Green, V. Cabral, M. Marcet-Houben, I. D. Jacobsen, J. Quintin, K. Seider, I. Frohner, W. Glaser, H. Jungwirth, S. Bachellier-Bassi, M. Chauvel, U. Zeidler, D. Ferrandon, T. Gabaldon, B. Hube, C. d'Enfert, S. Rupp, B. Cormack, K. Haynes and K. Kuchler (2014). "Systematic phenotyping of a large-scale Candida glabrata deletion collection reveals novel antifungal tolerance genes." PLoS Pathog **10**(6): e1004211.
Silva, S. S., R. K. Rowntree, S. Mekhoubad and J. T. Lee (2008). "X-chromosome inactivation and epigenetic fluidity in human embryonic stem cells." Proc Natl Acad Sci U S A **105**(12): 4820-4825.
Sims, D., A. M. Mendes-Pereira, J. Frankum, D. Burgess, M. A. Cerone, C. Lombardelli, C. Mitsopoulos, J. Hakas, N. Murugaesu, C. M. Isacke, K. Fenwick, I. Assiotis, I. Kozarewa, M. Zvelebil, A. Ashworth and C. J. Lord (2011). "High-throughput RNA interference screening using pooled shRNA libraries and next generation sequencing." Genome Biol **12**(10): R104.
Smith, G. J., D. Vijaykrishna, J. Bahl, S. J. Lycett, M. Worobey, O. G. Pybus, S. K. Ma, C. L. Cheung, J. Raghwani, S. Bhatt, J. S. Peiris, Y. Guan and A. Rambaut (2009). "Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic." Nature **459**(7250): 1122-1125.
Smyth, G. K. (2005). "limma: Linear Models for Microarray Data." 397-420.
Stigler, S. M. (1989). "Francis Galton's Account of the Invention of Correlation." Statistical Science **4**: 7.
van Opijnen, T., K. L. Bodi and A. Camilli (2009). "Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms." Nat Methods **6**(10): 767-772.
Wang, T., J. J. Wei, D. M. Sabatini and E. S. Lander (2014). "Genetic screens in human cells using the CRISPR-Cas9 system." Science **343**(6166): 80-84.
Winzeler, E. A., D. D. Shoemaker, A. Astromoff, H. Liang, K. Anderson, B. Andre, R. Bangham, R. Benito, J. D. Boeke, H. Bussey, A. M. Chu, C. Connelly, K. Davis, F. Dietrich, S. W. Dow, M. El Bakkoury, F. Foury, S. H. Friend, E. Gentalen, G. Giaever, J. H. Hegemann, T. Jones, M. Laub, H. Liao, N. Liebundguth, D. J. Lockhart, A. Lucau-Danila, M. Lussier, N.

M'Rabet, P. Menard, M. Mittmann, C. Pai, C. Rebischung, J. L. Revuelta, L. Riles, C. J. Roberts, P. Ross-MacDonald, B. Scherens, M. Snyder, S. Sookhai-Mahadeo, R. K. Storms, S. Veronneau, M. Voet, G. Volckaert, T. R. Ward, R. Wysocki, G. S. Yen, K. Yu, K. Zimmermann, P. Philippsen, M. Johnston and R. W. Davis (1999). "Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis." Science **285**(5429): 901-906.
Witten, D. M. and R. J. Tibshirani (2009). "Extensions of sparse canonical correlation analysis with applications to genomic data." Statistical applications in genetics and molecular biology **8**(1): 1-27.
Wong, A. S., G. C. Choi, A. A. Cheng, O. Purcell and T. K. Lu (2015). "Massively parallel high-order combinatorial genetics in human cells." Nat Biotechnol **33**(9): 952-961.
Yule, G. U. (1907). "On the theory of correlation for any number of variables, treated by a new system of notation." Proceedings of the Royal Society of London. Series A **79**(529): 182-193.
Zhang, B. B., W. M. Cai, H. L. Weng, Z. R. Hu, J. Lu, M. Zheng and R. H. Liu (2003). "Diagnostic value of platelet derived growth factor-BB, transforming growth factor-beta1, matrix metalloproteinase-1, and tissue inhibitor of matrix metalloproteinase-1 in serum and peripheral blood mononuclear cells for hepatic fibrosis." World J Gastroenterol **9**(11): 2490-2496.