

東京大学大学院新領域創成科学研究科
メディカル情報生命専攻

平成 27 年度

博士論文

Probabilistic models for haplotype assembly and
differentiation analysis

(確率モデルに基づくハプロタイプアセンブリ法と細胞分化機序解析法の開発)

指導教員 木立 尚孝 准教授

松本 拡高

Probabilistic models for haplotype assembly and differentiation analysis

Thesis by

Hiroataka Matsumoto

In Partial Fulfillment of the Requirements
for the Degree of Science

Submitted to

Department of Computational Biology and Medical Sciences
Graduate School of Frontier Sciences
the University of Tokyo

2016

(Defended January 19, 2016)

© 2016

Hiroataka Matsumoto

All Rights Reserved

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor, Associate Professor. Hisanori Kiryu for the very polite guidance and the continuous support over 5 years since I was an ignorant undergraduate student. Thanks to his direction, I can grope about in the darkness of study life.

I would like to thank Professor. Kiyoshi Asai for many advice not limited to research, especially at academic meetings. I also would like to thank all of professors of Department of Bioinformatics and Systems Biology, and Department of Computational Biology for gracious and rigorous education.

My sincere thanks also goes to Ms. Yoko Nomura, who supports paperwork, health care, and so on. She is an unsung hero in the true sense of the word.

I thank my friends Dr. Haruka Ozaki, Mr. Tsukasa Fukunaga, and Ms. Risa Kawaguchi for daily discussions. These stimulating discussions inspired me and I got many idea from them. I also thank the members of the student room for giving me such a fun time whilst research.

I also would like to thank JSPS for financial support because I cannot dedicate in my study without their support.

Finally, I would like to express my especial gratitude to my parents.

Abstract

The advancement of experimental technologies have enabled remarkable progress in molecular biology in recent years. However, the advancements in computational methods as well as sequencing technologies are essential to the progress of molecular biology. Molecular biological data frequently contains a mixture of multiple states and is hence heterogeneous, and computational methods are powerful tools to elucidate biological tasks from such heterogeneous data. In this research, we accomplish the following two tasks, which cannot be investigated easily from experimental data, by developing computational methods. Firstly, we developed a computational method to infer individual haplotypes from sequencing data. Next, we developed a computational method to analyze single-cell expression dynamics during cellular differentiation.

Development of a probabilistic model for haplotype assembly

Haplotype information is useful for various genetic analyses, including genome-wide association studies. Determining haplotypes experimentally is difficult and there are several computational approaches that infer haplotypes from genomic data. Among such approaches, single individual haplotyping or haplotype assembly, which infers two haplotypes of an individual from aligned sequence fragments, has been attracting considerable attention. To avoid incorrect results in downstream analyses, it is important not only to assemble haplotypes as long as possible but also to provide means to extract highly reliable haplotype regions. Although there are several efficient algorithms for solving haplotype assembly, there are no efficient method that allow for extracting the regions assembled with high confidence. Therefore, we develop a probabilistic model, called MixSIH, for solving the haplotype assembly problem. Based on the optimized model, a quality score is defined, which we call the ‘minimum connectivity’ (MC) score, for each segment in the

haplotype assembly. By using the MC scores, our algorithm can extract highly accurate haplotype segments. We also show evidence that an existing experimental dataset contains chimeric read fragments derived from different haplotypes, which significantly degrade the quality of assembled haplotypes. Therefore, we developed a method to detect chimeric fragments. The basis of our method is that a chimeric fragment would correspond to an artificial recombinant haplotype and would, therefore, differ from biological haplotypes in the population. We applied our method to two dilution-based sequencing datasets and the accuracy of assembled haplotypes increased significantly after removing chimeric fragment candidates.

Development of a probabilistic model for differentiation analysis

The advancement of single-cell technologies will shed light on the elucidation of the mechanism of differentiation. To fully analyze single-cell data, a novel computational method is necessary. There are several methods which use dimension reduction approach and reconstruct differentiation trajectory on the latent space to analyze single-cell expression data along differentiation. Although these approach will be useful to extract the properties of differentiations, these methods have several problems such as the absence of standard in the selection of the axis. In this research, we developed a novel method SCOUP to analyze single-cell expression data along differentiation by representing the expression dynamics with Ornstein-Uhlenbeck process. In our evaluation, SCOUP can infer the degree of differentiation of a cell (pseudo-time) with high accuracy comparing to previous methods, especially for single-cell RNA-seq. We evaluated the cell lineage estimation and SCOUP can estimate more accurately than previous method, especially for cells at an early stage of bifurcation. To understand cell fate decision mechanisms, it is important to analyze cells immediately after bifurcation. We also developed a novel correlation calculation to analyze gene regulatory relationship while removing the spurious correlation. Thus, SCOUP will be a promising approach to analyze single-cell expression data during cellular differentiation and to elucidate regulatory mechanism of differentiation.

Contents

Acknowledgments	iv
Abstract	v
1 General Introduction	1
1.1 Driving force of molecular biology in recent years	1
1.2 Heterogeneity of biological data	2
1.3 Probabilistic model and machine learning methods in bioinformatics	2
1.4 A new paradigm in bioinformatics	3
2 MixSIH: a mixture model for single individual haplotyping	5
2.1 Introduction	5
2.2 Methods	9
2.2.1 Algorithms and implementation	9
2.2.1.1 Notation	9
2.2.1.2 Mixture model	10
2.2.1.3 The minimum connectivity score	11
2.2.1.4 Variational bayesian inference	12
2.2.1.5 Inferring haplotypes	13
2.2.1.6 Possible extensions of the model	13
2.2.2 Datasets and data processing	14
2.2.2.1 Dataset generation	14
2.2.2.2 Potential chimeric fragments	18
2.3 Results and discussion	18

2.3.1	Comparison of pairwise accuracies	18
2.3.2	Effects of potential chimeric fragments	19
2.3.3	Incorporation of the trio-based data	20
2.3.4	Spatial distribution of MC values	21
2.3.5	Dependency of MC values on the fragment parameters	21
2.3.6	Optimality of inferred parameters	23
2.3.7	Comparison of running times	23
2.4	Conclusions	24
2.5	Supplementary text	26
2.5.1	Difference Between Our Model and Existing Models	26
2.5.2	Variational Bayes Expectation Maximization Algorithm	26
2.5.3	Iterative Twist Operations to Avoid Sub-optimal Solutions	27
2.5.3.1	Optimality of Inferred Parameters	28
2.5.4	Comparison of Accuracy Measures	29
2.5.5	Potential Chimeric Fragments	31
3	Integrating dilution-based sequencing and population genotypes for single individual	
	haplotyping	32
3.1	Background	32
3.2	Methods	35
3.2.1	Notation	35
3.2.2	Statistical phasing method	36
3.2.3	Chimeric fragment detection model	36
3.2.4	Cluster length and heterozygous calls for detecting chimeric fragment	38
3.2.5	Recovering SNP fragments from CF candidates	38
3.2.6	Mixture model for SIH	39
3.2.7	CF detection based on trio-based haplotypes	40
3.2.8	Dataset and data processing	40
3.2.9	Accuracy measure for CF detection	42
3.2.10	Accuracy measure for SIH	43

3.3	Results and discussion	43
3.3.1	Detection of chimeric fragments	43
3.3.2	SIH accuracy after removing suspicious CFs by using CSP	46
3.3.3	Assembled haplotype block size	47
3.3.4	Comparison of MixSIH and PHASE	50
3.4	Conclusions	52
3.5	Supplementary text	54
3.5.1	Cluster length and heterozygous calls	54
3.5.1.1	Evaluation of heterozygous calls in a reads cluster	54
3.5.1.2	ROC curves of heterozygosity evaluation	54
3.5.1.3	Distribution of length of reads clusters	55
3.5.2	Effects of changing various parameters	55
3.5.2.1	Impact of changing sliding window width on accuracy and running time	55
3.5.2.2	Effect of error rate α	57
3.5.2.3	Effect of the number of individual genotypes	57
3.5.3	Recovering SNP fragments from CF candidates	57
3.5.4	Calculation of SNP fragment error rate	59
3.5.5	Comparison for Duitama's SNP fragments	60
3.5.5.1	The number of NFs and CFs of Duitama's SNP fragments	60
3.5.5.2	SIH accuracy of Duitama's SNP fragments after removing suspicious CFs by using CSP	60
3.5.6	Precision of MixSIH and PHASE	61
4	SCOUP: a probabilistic model based on the Ornstein–Uhlenbeck process to analyze single-cell expression data during differentiation	64
4.1	Introduction	64
4.2	Methods	67
4.2.1	Ornstein-Uhlenbeck process	67
4.2.2	OU process for single lineage differentiation	68

4.2.3	Sufficient statistic for OU processes	69
4.2.4	EM algorithm	70
4.2.5	Mixture OU process for multi-lineage differentiation	71
4.2.6	Initialization of time parameter	72
4.2.7	Dimension reduction approach	73
4.2.8	Correlation between genes	73
4.2.9	Dataset	75
4.2.9.1	single-cell qPCR for single-lineage differentiation	75
4.2.9.2	single cell qPCR for bifurcation	75
4.2.9.3	Single-cell RNA-seq for single-lineage differentiation	76
4.2.10	Accuracy measure	76
4.2.10.1	Pseudo-time evaluation	76
4.2.10.2	Lineage evaluation	77
4.3	Results and discussion	77
4.3.1	Validation of parameter optimization	77
4.3.2	Validation of pseudo-time estimation	78
4.3.3	Validation of cell lineage estimate	81
4.3.4	Clustering genes	84
4.3.5	Correlation analysis	85
4.4	Conclusions	89
4.5	Supplementary text	91
4.5.1	Limit of a discrete time OU process	91
4.5.1.1	Transformation into the multivariate normal distribution	91
4.5.1.2	Derivation of mean vector and variance-covariance matrix	92
4.5.1.3	The complete log-likelihood	95
4.5.1.4	Derivation of the sufficient statistic	96
4.5.1.5	Derivation of F_{ss}	96
4.5.1.6	Derivation of F_{ss+1}	96
4.5.1.7	Derivation of F_s	97
4.5.2	Derivation of Q function	97

4.5.2.1	Derivation of $2F_{ss} - 2F_{ss+1} + X_0^2 + X_N^2$	98
4.5.2.2	Derivation of $X_0^2 - X_N^2 - 2\theta^* X_0 + 2\theta^* X_N + \lambda^* (-2F_{ss} + F_{ss+1} + 2\theta^* F_s - N\theta^{*2})$	99
4.5.2.3	Q function	100
4.5.3	Parameter optimization	101
4.5.3.1	Optimization of θ	101
4.5.3.2	Optimization of α	101
4.5.3.3	Optimization of σ^2	101
4.5.3.4	Optimization of t	102
4.5.4	Mixture OU process for multi-lineage differentiation	103
4.5.4.1	Parameter optimization	105
4.5.4.2	Optimization of θ_{gk}	105
4.5.4.3	Optimization of α_g	105
4.5.4.4	Optimization of σ_g^2	105
4.5.4.5	Optimization of t_c	106
4.5.4.6	Optimization of π_k	106
4.5.5	Expected value of S_{cg}	106
4.5.6	The marginal log-likelihood	107
4.5.7	A procedure of parameter optimization	107
4.5.8	Validation of parameter optimization method	109
4.5.9	Cell lineage estimation with Gaussian mixture model	113
4.5.10	Annotated pairs in the top 1,000 C_{Raw} and C_{Std} values	114

List of Figures

- 2.1 An illustration of single individual haplotyping (SIH). The input data for SIH are the SNP fragments (B) which are extracted from the heterozygous alleles in aligned DNA fragments (A). SIH algorithms (C) reconstruct the original haplotypes (D) from the SNP fragments. 8
- 2.2 Consistency of pair sites. A. a. We assume that the two true haplotypes are the sequences of all 0 and all 1. b. Inferred haplotypes contain switch errors indicated by the arrows: (i) a consistent pair, (ii) an inconsistent pair, and (iii) if there are an uncontrolled number of switch errors between a pair, the probabilities of being consistent or inconsistent are both 0.5. B. The example of the case that switch error rate is not suitable to evaluate the quality of the segment. The consistency of a reconstructed haplotype which has single switch error in the middle (top) is high than a reconstructed haplotype which has single switch error located at an end of the segment, but switch error rate cannot distinguish these situations. Two contiguous switch errors, which are caused by sequencing error or genotyping error and do not disrupt the consistency between front and back parts, are regarded as twice of a single switch error in switch error rate (bottom). 17
- 2.3 Precision curves based on the consistent pair counts. The x -axis represents the number of predicted pairs in log scale. The arrows indicate the MC thresholds. The accuracies are computed for the simulation dataset (A), and the real dataset (B): \square no assembly; \circ MixSIH; \triangle ReFHap; $+$ FastHare; \times DGS. In the simulation, we set $M = 2000$ and repeated the experiment 10 times for each algorithm; average values are plotted. 19

2.4	The precisions of the algorithms for the dataset in which fragments with chimerity greater than 10 are removed. For comparison, the precisions of MixSIH for the original dataset are also shown as diamonds.	20
2.5	Spatial distribution of MC and LD. A. A colored density plot of the MC values and the number of fragments. The x -axis represents the coordinates of heterozygous sites. The actual locations of the sites in genome coordinates are shown by thin black diagonal lines and the black horizontal line represents a 10-11 megabase region of chromosome 20. The upper densities represent the connectivity values. The lower densities represent the number of fragments spanning the pair sites. B. A colored density plot of the precisions (upper) and the absolute normalized linkage disequilibrium $ D' $ (lower) for the same region.	22
2.6	Dependency of the lowest MC value with precision ≥ 0.95 for coverage c , fragment length $[l_1, l_2]$, and error rate e . The experiments were repeated 10 times, and the average values are plotted.	23
2.7	Increase of log likelihood values for each iteration. The dotted line represents the approximate maximal log likelihood; the solid line, the changes of the optimized log likelihood for each twist operation; the broken line, the connectivity values at the positions that the optimizing parameters are twisted.	24
2.8	The running times of the tested algorithms. The x -axis is the number of sites. The y -axis is the running time in seconds. Both are displayed on a logarithmic scale.	25
2.9	Increase of log likelihood values for each iteration. The dotted line represents the approximate maximal log likelihood; the solid line, the changes of the optimized log likelihood for each twist operation; the broken line, the connectivity values at the positions that the optimizing parameters are twisted.	28
2.10	Chimerity distribution of the real dataset.	31
2.11	The precisions for the original dataset (\circ) and the datasets in which the fragments with chimerity greater than 5 (\triangle), 10 (+), and 30 (\times) are removed.	31

- 3.1 An illustration of dilution-based sequencing. (i) The DNA fragments are separated into multiple low-concentration dilutions. (ii) After sequencing and mapping an aliquot, mapped reads form clusters which correspond to DNA fragments. (iii) Clusters are merged into read fragments and result in natural fragments (a), (b) and a chimeric fragment (c). Chimeric fragments are produced when short reads derived from multiple DNA fragments are regarded as one cluster. 35
- 3.2 Comparison of CSP density distributions for NFs and CFs. (A) and (B) are the distributions of Kaper's data and Duitama's data, respectively. 44
- 3.3 The ROC curves of CSP, cluster length, and total heterozygosity for classification of CFs and NFs. The ROC curves are obtained by changing the threshold of CSP, cluster length, total heterozygosity, respectively. There is a region that the data point of the ROC curve of total heterogeneity for Kaper's data is absent, and hence, the ROC curve is supplemented (shown as gray line). (A) and (B) correspond to Kaper's data and Duitama's data, respectively. 45
- 3.4 The Venn diagrams of CFs detected by CSP, length, and total heterozygosity. The number in each cell is the number of CFs in the corresponding category. The threshold for CF detection of each measure was set so that the 1-specificity was under 0.1. (A) and (B) correspond to Kaper's data and Duitama's data, respectively. 46
- 3.5 Precision curves based on consistent pair counts. The x -axis represents the number of predicted pairs on a log scale. MC of MixSIH was changed from 0 to 10. The accuracies of the data filtered with cluster length and heterozygous calls (filtered) (filled point symbols) and the further filtered data, in which fragments with CSP > 7 are removed (filtered+CSP) (empty point symbols), are shown for Kaper's data (A) and Duitama's data (B): \circ MixSIH; \triangle ReFHap; \square FastHare; \diamond DGS. 48

3.6	Comparison of MC scores and maximum PHASE probabilities (A) and (B) correspond to Kaper's data and Duitama's data, respectively. The x -axis represents $\ln(1.001 - \max P)$, where $\max P$ is the maximum PHASE probability and we use 1.001 to deal with $\max P = 1.0$. The y -axis represents the MC score of MixSIH. Data are randomly selected 1000 times from chromosome 1. The vertical dotted line corresponds to the maximum PHASE probability above which the precision of PHASE is over 0.9, and the horizontal dotted line corresponds to the MC value above which precision of MixSIH is over 0.9.	51
3.7	The ROC curves of total heterozygosity, average heterozygosity, and maximum heterozygosity for classification of CFs and NFs. A and B correspond to Kaper's data and Duitama's data, respectively.	55
3.8	The distribution of cluster length. The x -axis represents the length of reads cluster and the y -axis represents the number of SNP fragments which are correspond to the each reads cluster.	56
3.9	AUC values and running times for various values of W	56
3.10	AUC values for various numbers of individuals.	58
3.11	Precision curves based on consistent pair counts for Duitama's SNP fragments (A) and our processed Duitama's data (B). The x -axis represents the number of predicted pairs on a log scale. MC of MixSIH was changed from 0 to 10. The accuracies of the original data (filled point symbols) and the processed data (empty point symbols), in which fragments with $CSP > 7$ are removed, are shown: \circ MixSIH; \triangle ReFHap; \square FastHare; \diamond DGS.	61
3.12	The consistent pair precision of (A) MC value and (B) maximum PHASE probability (maxP) for 5 SNPs regions.	63

- 4.1 The conceptual diagrams of the OU process (A) and SCOUP for multi-lineage differentiation (B). (A) The OU process represents a variable (i.e., expression of a gene g in a cell c) moving toward attractor (θ_g) with Brownian motion. The value at time t satisfies the normal distribution (see “Methods”). (B) Each lineage has distinct attractor (θ_{g1} and θ_{g2}), and the lineage of a cell c is represented with latent value Z_c . The expression of gene g in cell c is described with the mixture OU process. 67
- 4.2 Validation of parameter estimation of SCOUP for simulation data. (A) and (B) is the comparison between the estimated values and true values for pseudo-time (t) and θ_g , respectively. The outlier whose estimated value exceeds the boundary of drawing area is visualized in the border with a red circle for visualization. (C) is the log-likelihood curve with respect to t_c of a cell. The optimized t_c is indicated with x-max. 78
- 4.3 The histograms of pseudo-time estimates produced by each method for Kouno’s data (1) without additional noise. The histograms are drawn for each experimental time point with different colors. The pseudo-time values inferred by SCOUP over 1.0 are integrated into 1.0 for visualization. The pseudo-time values inferred by Monocle and TSCAN are normalized so that maximum = 1.0. 79
- 4.4 PIS of each method applied to Kouno’s data (1). The x -axis represents the noise level (ϵ) (see “Methods”) and the y -axis represents the degree of inconsistency between the pseudo-time and experimental time (PIS). Each method is distinguished by color: red, SCOUP; yellow, SP; green, Monocle; and blue, TSCAN. We compared the PIS of Monocle for different parameters *max_components*, which correspond to dimensions. The solid and dotted lines correspond to *max_components* = 2 and 3, respectively. 81
- 4.5 PCA of cells of Kouno’s data based on gene expression. The cell colors indicate the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with Monocle (right). The color for SCOUP is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for Monocle is defined by estimated states: black, state 1 (pre-bifurcation); red, state 2; and blue, state 3. The color of each state is defined so that they are consistent among each plots. 83

4.6	PCA of cells of Moignard’s data based on gene expression. The cell colors represent the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with Monocle (right). The color for the genuine lineage is defined by the annotation of the cell; yellow, HF; red, 4SG; and purple, 4SFG ⁻ . The color for the SCOUP analysis is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for the Monocle analysis is defined by estimated states; black, state 1 (pre-bifurcation); red, state 2; and blue, state 3. We determined the color of each state so that they are consistent among each plot.	83
4.7	Overall trend in standardized expression patterns along pseudo-time for each group. This plot is drawn with the plot_clusters function in the Monocle package.	85
4.8	The correlation network based on C_{Raw} (A) and C_{Std} (B) for genes in group 1. There are a total of 93 and 107 genes in the C_{Raw} and C_{Std} network, respectively. The width of each edge represents the magnitude of an expression correlation between the two genes, and color represents the sign, green for a positive correlation and red for a negative correlation. To improve clarity, correlations with an absolute value lower than 0.55 (0.25) are not shown for C_{Raw} (C_{Std}) network.	88
4.9	Comparison between the estimated values and true values: (A) for pseudo-time (t), (B) for θ_g , (C) for α_g , (D) for σ_g^2 , (E) for mean, (F) for variance. The outlier whose estimated value exceeds the boundary of drawing area is visualized in the border with a red circle for visualization.	111
4.10	The log-likelihood curve with respect to t_c of a cell. The optimized t_c is indicated with x-max.	111
4.11	The log-likelihood surface with respect to α_g and θ_g of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (α_g, θ_g) is indicated with x-max.	112
4.12	The log-likelihood surface with respect to σ_g^2 and θ_g of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (σ_g^2, θ_g) is indicated with x-max.	112

4.13 The log-likelihood surface with respect to α_g and σ_g^2 of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (α_g, σ_g^2) is indicated with x-max. 112

4.14 PCA of cells of Kouno’s data based on gene expression. The cell colors indicate the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with mclust (right). The color for SCOUP is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for mclust is defined by expectation of latent values; black, 0.5; red, 0.0; and blue, 1.0. The color of each state is consistent among plots. 113

4.15 PCA of cells of Moignard’s data based on gene expression. The cell colors represent the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with mclust (right). The color for the genuine lineage is defined by the annotation of the cell; yellow, HF; red, 4SG; and purple, 4SFG⁻. The color for the SCOUP analysis is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for the mclust analysis is defined by expectation of latent values; black, 0.5; red, 0.0; and blue, 1.0. We determined the color of each state so that they are consistent among each plot. 114

List of Tables

3.1	The switch error rate (%) of each SIH algorithm for data (filtered) and data (filtered+CSP). MC of MixSIH is set to 10. (A) and (B) correspond to Kaper's data and Duitama's data, respectively.	47
3.2	QAN50 (kb) of each SIH algorithm data (filtered) and data (filtered+CSP), in which fragment with $CSP > 7$ are removed. MC of MixSIH is set to 10. (A) and (B) correspond to Kaper's data and Duitama's data, respectively.	49
3.3	The number of the SNP fragments which cover the certain range of the numbered of SNPs (before SIH) and the number of haplotype blocks which contain the certain range of the number of phased SNPs (after SIH) for Kaper's data (A) and Duitama's data (B) (Note that a SNP can be contained in multiple SNP fragments and the haplotype blocks do not overlap each other). The first row defines the range of the number of SNPs.	49
3.4	The numbers of regions for each of the areas which are defined by the precision of MixSIH and PHASE: (A) Kaper's data and (B) Duitama's data. The rows and columns represent the accuracy of MixSIH and PHASE, respectively. The numbers in parentheses are the numbers of regions remaining after regions which contain sites that lack SNP fragments have been removed.	52
3.5	AUC values for each α	57
3.6	The numbers of all fragments, NFs, and CFs after performing each process on Kaper's data (A) and Duitama's data (B).	59
3.7	The number of NFs and CFs of Duitama's SNP fragments (A) and our processed Duitama's data (B).	60

4.1	PIS for each method applied to Shalek's data. Each row represents the method, and each column represents the kind of stimulation for differentiation. NA means that Monocle did not work well.	81
4.2	Mean AUC values for cell lineage estimates using each method for Kouno's data (2).	82
4.3	The number of top 5,000 genes, top 1,000 genes in each group. The total number are not equal to 5000 and 1000 because the response curves for a few genes could not be calculated.	85
4.4	The top three GO terms for each group. The third column shows the negative logarithm of the Bonferroni-adjusted p -value.	86
4.5	The top three transcription factors and their related genes for group 1. The left and right tables correspond to $\bar{C}_{\text{Raw}}(i, 1)$ and $\bar{C}_{\text{Std}}(i, 1)$, respectively. The first column of each table contains the rank of the absolute difference of expression between 1-h cells and 6-h cells, and the second column lists the gene names. The third column contains the $\bar{C}_{\text{Raw}}(i, 1)$ or $(\bar{C}_{\text{Std}}(i, 1))$ of the candidate genes.	86
4.6	The annotated pairs in the top 1,000 C_{Raw} and C_{Std} values. The left and right tables correspond to C_{Raw} and C_{Std} , respectively. The first column of each table contains the TF names and the second column lists the target gene names. The third column contains the C_{Raw} or C_{Std} of the pairs.	115

Chapter 1

General Introduction

1.1 Driving force of molecular biology in recent years

The advancement of experimental equipment such as sequencing instruments and the development of experimental techniques such as ChIP-seq and RNA-seq have enabled remarkable progress in molecular biology in recent years. For example, the ENCODE (Encyclopedia of DNA Elements) project, an international collaboration of research groups that investigates comprehensive experiments using such technologies, has yielded many insights [1]. Such progress illustrates the importance of advancements in experimental technologies for the future of molecular biology. However, advancements in computational methods as well as sequencing technologies are essential to the progress of molecular biology. Without advancements in bioinformatics algorithms for genome assembly and read mapping, biological analysis using a high volume of sequenced reads would be impossible.

The development of computational methods is indispensable to the progress of molecular biology. In this research, we accomplish the following two tasks, which cannot be investigated easily from experimental data, by developing computational methods. Firstly, we developed a computational method to infer individual haplotypes from sequencing data. Next, we developed a computational method to analyze single-cell expression dynamics during cellular differentiation.

1.2 Heterogeneity of biological data

Molecular biological data frequently contains a mixture of multiple states and is hence heterogeneous. For example, various subtypes exist in a tumor [2], and genome sequencing data of a tumor sample therefore contains the individual reads from different subtypes. This heterogeneity disrupts the accurate recognition of mutations in each subtype. Even if a sample is collected from normal tissue, it may contain multiple cell types and the expression data from that sample is then the average of the heterogeneous sample. Moreover, the human genome is diploid, containing two homologous haplotypes. Therefore, genome sequencing data comprise a mixture of reads with different haplotype origins, which makes distinguishing the origin of a read and estimating haplotypes a significant challenge. Because molecular biological data exhibit heterogeneity at various scales, computational methods are necessary to infer the original states. The first theme of this research is the development of a computational method to reconstruct haplotypes from sequencing data by inferring the origin of individual reads.

However, there have been attempts to overcome the problem of heterogeneity through experimental approaches. Single-cell sequencing technologies could solve some of these heterogeneity problems. For example, single-cell qPCR and single-cell RNA-seq provide single-cell resolution expression data and hence overcome the problem of averaged expression for bulk sample expression assays. Because of the novel properties of single-cell data, a novel computational method is necessary to fully utilize these data. The second theme of this research is the development of a novel computational approach to analyzing single-cell expression data for differentiation.

1.3 Probabilistic model and machine learning methods in bioinformatics

In bioinformatics, there are several computational methods for data analysis that are based on probabilistic models and machine learning methods. Because of the flexibility of these approaches, we can integrate several conditions into a single probabilistic model. Moreover, there are several parameter optimization methods for probabilistic models within the framework of machine learning, which makes it possible to analyze huge amounts of data. Therefore, methods that use probabilistic

models and machine learning are efficient approaches for analyzing complicated biological data. In particular, many efficient probabilistic models based on mixture models have been developed to overcome heterogeneity. For example, RNA-seq data contain a mixture of reads of different isoform origins and inferences of isoform abundance are complicated because an individual read can be mapped to multiple isoforms. The Cufflinks [3] deals with this problem by considering a mixture model that generate reads from one of the isoform depending on the iso form abundance and estimates expression level from optimized parameters.

In addition, a method based on probabilistic model can be applied to the analysis of different source data. For example, inference of taxonomic composition from metagenomic data can be considered the inference of isoform abundance from transcriptome data, and a mixture model like Cufflinks has been developed for metagenomic analyses [4]. Thus, probabilistic model-based methods will be progressively more important in bioinformatics.

In this research, we construct methods for haplotype assembly and for differentiation analysis based on a probabilistic model and machine learning approach. In a probabilistic model, data are usually regarded not as an input, but as an output. Accordingly, it is important to properly represent the experimental process by which data are generated in developing a probabilistic model. It is also important that parameters capture biological meaning in probabilistic models so that biological results can be directly interpreted from optimized parameters. From this perspective, we developed computational methods for the two aforementioned tasks.

1.4 A new paradigm in bioinformatics

Because of the advancement of experimental technologies, we can now obtain huge biological datasets, such as nucleotide sequence datasets. "Big data" advances data-driven science (rather than hypothesis-driven science) and increases the importance of bioinformatics. Although a large portion of recent biological knowledge has been generated through the power of bioinformatics, current bioinformatics research programs usually just analyze biological data as requested by an experimental researcher. However, biology is also advanced through progress in computational research, for example, through the development of probabilistic models to elucidate biological problems and by proposing optimum experimental designs for subsequent analyses. This approach is regarded as an

integration of hypothesis-driven science and data-driven science, and will become a new paradigm of bioinformatics. The second objective of this research is to delve into biological problems from the perspective of the computational approach by fully utilizing single-cell data, and we hope this will be a harbinger of a new paradigm in bioinformatics.

Chapter 2

MixSIH: a mixture model for single individual haplotyping

2.1 Introduction

Human somatic cells are diploid and contain two homologous copies of chromosomes, each of which is derived from either paternal or maternal chromosomes. The two chromosomes differ at a number of loci and the most abundant type of variation is single nucleotide polymorphism (SNP). Most current research does not determine the chromosomal origin of the variations and uses only genotype information for the analyses. However, haplotype information is valuable for genome-wide association studies (GWAS) [5] and for analyzing genetic structures such as linkage disequilibrium, recombination patterns [6], and correlations between variations and diseases [7].

Let us consider a simple example to demonstrate the importance of haplotype information. Suppose that in a gene coding region, there are two SNP loci, each of which has an independent deleterious mutation in either one of the two homologous chromosomes. If both of the two deleterious mutations are located on the same chromosome, the other chromosome can produce normal proteins. On the other hand, if each chromosome contains either one of the two deleterious mutations, the cells cannot produce normal proteins. It is not possible to distinguish these two cases with only genotype information.

There is a group of algorithms for haplotype inference that statistically construct a set of haplotypes from population genotypes [8–12] Review see [13]. These algorithms have been developed in

response to technological advances such as SNP arrays that efficiently measure personal genotypes at a genomic scale. The algorithms infer haplotype blocks based on the assumption that the variety of combinations of alleles is very limited. Therefore, these algorithms fail to identify correct haplotypes in regions with low linkage disequilibrium (LD) where there are frequent recombination events. These algorithms also cannot identify spontaneous mutations. These difficulties are partially resolved by using genotypes of pedigrees. However, family data are not always available, and furthermore, they cannot determine the haplotypes of the loci at which all the family members have the same genotype.

Another group of algorithms is single individual haplotyping (SIH) or haplotype assembly. These algorithms infer the two haplotypes of an individual from sequenced DNA fragments [14–21]. These algorithms take as input the read fragments that are aligned to the reference genome, and output the two assembled haplotypes (Figure 2.1). The algorithms utilize the fact that each read fragment is derived from either one of two chromosomes, though the observed data are a mixture of fragment data from both the chromosomes. If a read fragment spans two or more heterozygous loci, the haplotype can be determined for these sites from the co-occurrence of alleles in the fragment. Two read fragments are determined to originate from the same chromosome if they overlap at a region that has at least one heterozygous locus, and they have the same alleles at these loci. In this case, we obtain a larger haplotype-resolved region by merging the two fragments. The SIH problem is complicated because the fragment data contain many inconsistent fragments caused by sequencing or mapping error.

SIH algorithms did not attract much attention until recently, since the read fragments of next-generation sequencing experiments are not long enough to span multiple heterozygous loci, which exist at only one in one kilo-base on average [22], and the Sanger sequencing that produces long read fragments is too expensive to be conducted at a genomic scale. However, this situation is changing rapidly with the advent of real-time single-molecule sequencing technologies, which are able to sequence DNA fragments as long as 50 kilo-bases [23], and with the development of a novel experimental technique called ‘fosmid pool-based next-generation sequencing’ [17, 24, 25], which randomly assigns a bar-code to each read cluster that is derived from the same region in the same chromosome. Because of these advances in experimental techniques, SIH has emerged as one of the most promising approaches for analyzing the haplotype structures of diploid organisms.

The haplotype information which contains errors is likely to lead to wrong results in downstream analyses. For example, in detecting the recombination events from the parent-offspring haplotypes [26], the haplotyping errors are regarded as recombination events by mistake. Another example is that haplotyping errors considerably decrease the detection power of amplified haplotypes in cancer [27] and fetus haplotypes [28]. To use haplotype information in downstream analyses while avoiding such harmful influence of haplotyping errors, it is important not only to assemble haplotypes as long as possible but also to provide means to extract highly reliable haplotype regions. In the statistical haplotype phasing, reliable haplotype regions are determined by selecting the blocks of limited haplotype diversity and level of LD [29–31]. Although there are many algorithms for SIH, none of these algorithms can provide confidence scores to extract reliable haplotype regions.

The algorithms for SIH are classified into two strategies; most of the previous algorithms use deterministic strategies [14–17, 19, 21] but a few take a probabilistic modeling approach [18, 20]. The deterministic algorithms usually include solving the MAX-CUT problem of graph theory [32] in their computational procedures in order to partition the set of the input fragments into two groups representing the two haplotypes. Because these algorithms are designed to optimize only a certain global score function that measures the number of inconsistent fragments and do not model the fragments and haplotypes themselves, it is difficult to produce confidence scores for each region of the assembled haplotypes.

On the other hand, the probabilistic approaches of Kim [18] and Li [20] assume that each observed fragment is sampled from one of the two unobserved haplotypes. Unlike the deterministic approaches, probabilistic models allow the computation of various expected values and confidence values from the Bayesian posterior distributions. For example, Kim [18] and Li [20] defined a confidence value for the haplotype reconstruction of each segment of SNP loci. Unfortunately, those researchers chose a model structure for which the exact computation of the likelihood is extremely computationally intensive. Because the complexity of this summation is exponential in the number of SNP sites, only the posterior probabilities of the haplotypes for neighboring loci are considered. The complete haplotypes are reconstructed by connecting plausible haplotypes of neighboring pairs according to their posterior probabilities. Hence, their approach cannot take into account the full information of fragments that span three or more SNP loci. Their confidence scores for haplotype

2.2 Methods

2.2.1 Algorithms and implementation

2.2.1.1 Notation

Throughout the paper, we denote the number of elements of any set A by $|A|$, and the direct product set $\underbrace{A \times \cdots \times A}_n$ by $A^{\otimes n}$. Let $X = \{1, 2, \dots, M\}$ be the set of SNP loci, and $\mathcal{H} = \{0, 1\}$ be the two haplotypes. It is convenient to introduce a *phase vector* $\Phi = \varphi_1 \cdots \varphi_M$. The pair $\varphi_j = (\varphi_{j0}, \varphi_{j1})$ is referred to as *phase*, and represents the two alleles of haplotype 0 and 1 at site j , respectively. Because the haplotype assembly problem is trivial for homozygous sites, and because it is usually much easier to determine the genotype than to determine the haplotypes, it is often convenient to restrict the SNP loci X to heterozygous sites. Furthermore, if sequence-specific sequencing errors are not considered, it is convenient to use a simple binary representation of alleles; we randomly assign 0 to one of the two alleles at each heterozygous site j , and 1 to the other allele. In this case, the set of alleles is denoted by $\Sigma = \{0, 1\}$, and the set of possible phases is denoted by $\Delta = \{(0, 1), (1, 0)\}$. We assume this binary representation throughout the paper.

Let $F = \{f_i | i = 1, \dots, N\}$ be the set of input fragments which are supposed to be aligned to the reference genome, and each fragment f_i takes value $f_{ij} \in \Sigma$ at locus $j \in X$ if a nucleotide is aligned and equal to one of two alleles, and $f_{ij} = \emptyset$ if fragment f_i is unaligned, gapped, ambiguous, or a base different from the two alleles, at site j . For any subset $X' \subseteq X$, we say fragment f_i *spans* the sites X' if $f_{ij} \neq \emptyset$ for all $j \in X'$. We refer to the subset of X spanned by fragment f as $X(f)$. We say fragment f_i *covers* site j if there exists a pair of spanning two different (possible non consecutive) SNP sites $j_1, j_2 \in X(f_i)$ such that $j_1 < j \leq j_2$. The set of fragments that cover site j is denoted by $F^c(j)$. Further, we refer to the set of all the possible haplotypes for sites $X(f_i)$ as $\Delta(f_i) = \Delta^{\otimes |X(f_i)|}$.

The SIH problem takes a set of aligned SNP fragments F as input and outputs a hidden phase vector Φ (Figure 2.1). Because the SIH problem does not associate the inferred haplotypes \mathcal{H} with the real paternal and maternal chromosomes, the switched configuration $\bar{\Phi} = \bar{\varphi}_1 \cdots \bar{\varphi}_M$, $\bar{\varphi}_j = (\varphi_{j\bar{0}}, \varphi_{j\bar{1}})$ with $\bar{0} = 1$ and $\bar{1} = 0$, must be regarded as a completely equivalent prediction. Therefore, SIH has no meaning if there is only one heterozygous site, and it is only meaningful if one considers co-occurrences of alleles on the same haplotype for two or more heterozygous sites.

2.2.1.2 Mixture model

We model the probabilistic distribution of the observed fragments F by

$$P(F|\Theta) = \sum_{H \in \mathcal{H}^{\otimes N}} \prod_{i=1}^N \sum_{\Phi^{(i)} \in \Delta(f_i)} P(f_i|h_i, \Phi^{(i)}) p^m(h_i) P(\Phi^{(i)}),$$

$$P(\Phi^{(i)}) = \prod_{j \in X(f_i)} p_j^\Phi(\varphi_j^{(i)}),$$

where Θ represents a set of parameters defined later, $\Phi^{(i)} \in \Delta(f_i)$ represents a partial haplotype reconstruction over the sites $X(f_i)$ spanned by fragment f_i , $H = h_1 \dots h_N$ where $h_i \in \mathcal{H}$ represents the haplotype origin of fragment f_i , $p^m(h)$ is the mixture probability of haplotype $h \in \mathcal{H}$, and $p_j^\Phi(\nu)$ is the probability that phase $\nu \in \Delta$ is instantiated at site j . We define the probability of emitting fragment f_i from haplotype h_i given a fixed phase vector $\Phi^{(i)}$ as follows.

$$P(f_i|h_i, \Phi^{(i)}) = \prod_{j \in X(f_i)} p^e(f_{ij}|\varphi_{jh_i}^{(i)})$$

where,

$$p^e(\sigma|\sigma') = \begin{cases} (1 - \alpha) & \text{for } \sigma = \sigma' \\ \alpha & \text{for } \sigma \neq \sigma' \end{cases}$$

is the probability that we observe $\sigma \in \Sigma$ when the true allele is $\sigma' \in \Sigma$ and α represents the sequence error rate which we assume is independent of fragments and positions.

We take α as a fixed constant because it is better estimated from other resources rather than from only the bases at the SNP sites. For example, we may estimate α by using the all the read sequences or by using information from other dedicated studies about sequencing and mapping errors. In the following, we use $\alpha = 0.1$ unless otherwise mentioned and the dependency of the α is described in the Additional file. We further assume the mixture probabilities are equal, $p^m(0) = p^m(1) = 0.5$, as they often converge to around 0.5. Therefore, the parameter set Θ that needs to be optimized consists only of the set of phase probabilities: $\Theta = \{\theta_{j\nu}\} = \{p_j^\Phi(\nu)\}$.

Let $\mathcal{I}_{ihj\nu}$ be the indicator function that is one if fragment f_i is derived from haplotype h , $X(f_i)$ includes j , and the haplotypes have phase ν at site j , and that is zero otherwise. $\mathcal{I}_{ihj\nu}$ is uniquely determined if the haplotype origins $H = \{h_i | i = 1, \dots, N\}$ and phase vectors $\Psi = \{\Phi^{(i)} | i = 1, \dots, N\}$ of fragments F are specified. Then the marginalized likelihood $P(F|\Theta)$ is given by

$$P(F|\Theta) = \sum_{H, \Psi} P(F, H, \Psi|\Theta),$$

$$\log(P(F, H, \Psi|\Theta)) = N \log(0.5) +$$

$$\sum_{i=1}^N \sum_{h \in \mathcal{H}} \sum_{j \in X(f_i)} \sum_{\nu \in \Delta} \mathcal{I}_{ihj\nu} [\mu_{ihj\nu} + \log \theta_{j\nu}],$$

$$\mu_{ihj\nu} = \log(p^e(f_{ij} | \nu_h)).$$

We explain the difference between our model and the models of Kim [18] and Li [20] in the Additional file.

2.2.1.3 The minimum connectivity score

As described above, the two haplotypes \mathcal{H} in the SIH problem have no particular identity and it is not possible to predict which of them converges to the actual paternal or maternal chromosome. In relation to this, the likelihood function $P(F, H, \Psi|\Theta)$ has a symmetry between the switched configurations: $P(F, \bar{H}, \bar{\Psi}|\bar{\Theta}) = P(F, H, \Psi|\Theta)$, where $\bar{H} = \{\bar{h}_i | i = 1, \dots, N\}$ and $\bar{\Psi} = \{\bar{\Phi}^{(i)} | i = 1, \dots, N\}$ represent the configuration that all the haplotype origins of the fragments are exchanged, and $\bar{\Theta} = \{\bar{\theta}_{j\nu}\}$, $\bar{\theta}_{j\nu} = \theta_{j\bar{\nu}}$ are the switched phase probabilities. Therefore, the marginal likelihood $P(F|\Theta) = \sum_{H, \Psi} P(F, H, \Psi|\Theta)$ is symmetric for the two parameter sets: $P(F|\bar{\Theta}) = P(F|\Theta)$.

Suppose that the probabilistic model is optimized for two segments of SNP sites between which there are no connecting fragments, then the association of the haplotypes $\{0, 1\}$ to the true paternal and maternal chromosomes are selected at random for each segment. Even if there are several connecting fragments, the associations in each segment are determined almost randomly if the number of connecting fragments is not sufficient or there are many conflicting fragments. Such sites often cause switch errors. We define the connectivity at site j_0 as a log ratio of the marginal log

likelihoods:

$$\text{connectivity}(j_0) = \log \left(\frac{P(F|\Theta)}{P(F|\Theta')} \right) = \log \left(\frac{P(F^c(j_0)|\Theta)}{P(F^c(j_0)|\Theta')} \right)$$

where $\Theta' = \{\theta'_{j\nu}\}$ with $\theta'_{j\nu} = \theta_{j\nu}$ for $j < j_0$ and $\theta'_{j\nu} = \bar{\theta}_{j\nu}$ for $j \geq j_0$. The second equality follows from the symmetry of $P(F|\Theta)$ described above, and shows that only the fragments covering site j_0 are necessary to compute the connectivity of site j_0 . The connectivity measures the resilience of the assembly result against swapping the two haplotypes 0 and 1 in the right part $j = j_0, \dots, M$ of the sites. We refer to this change of parameters $\Theta \rightarrow \Theta'$ as *twisting the parameters at site j_0* .

For each pair of sites (j_1, j_2) ($j_1 < j_2$), we define the minimum connectivity (MC) score as

$$\text{MC}(j_1, j_2) = \min_{j_1 < j \leq j_2} \text{connectivity}(j) .$$

We extract confidently assembled regions by selecting the pairs (j_1, j_2) with high MC values. From the above definition, it is obvious that if the MC value is higher than a given threshold for some pair (j_1, j_2) , then all the pairs inside range $[j_1, j_2]$ have MC values higher than the threshold. In this sense, $\text{MC}(j_1, j_2)$ can be considered as defined on the range $[j_1, j_2]$.

2.2.1.4 Variational bayesian inference

We use the VBEM algorithm to optimize the parameters Θ [33]. We approximate the Bayesian posterior distribution $P(H, \Psi, \Theta|F)$ with factorized variational functions $Q(H, \Psi, \Theta) = Q^{H\Psi}(H, \Psi) \cdot Q^\Theta(\Theta)$ such that the Kullback-Leibler divergence $KL_{H\Psi\Theta}(Q(H, \Psi, \Theta)||P(H, \Psi, \Theta|F))$ between the two distributions is minimized. The solution to this optimization problem has the form

$$Q^{H\Psi}(H, \Psi) = \frac{1}{Z_{H\Psi}} \exp \left(\sum_{i=1}^N \sum_{h \in \mathcal{H}} \sum_{j \in X(f_i)} \sum_{\nu \in \Delta} \mathcal{I}_{ihj\nu} \log(\beta_{ihj\nu}) \right) ,$$

$$Q^\Theta(\Theta) = \prod_{j=1}^M \text{Dir}(\theta_j | \lambda_j) ,$$

where $Z^{H\Psi}$ is a normalization constant, $\beta_{ihj\nu}$ and $\lambda_{j\nu}$ represent the hyperparameters that specify the posterior distributions, and $\text{Dir}(\theta_j|\lambda_j)$ is the Dirichlet probability distribution of $|\Delta|$ parameters. Because $Q^{H\Psi}(H, \Psi)$ and $Q^\Theta(\Theta)$ are connected through the dependencies among the hyperparameters, they cannot be found simultaneously. Therefore, we optimize $\beta_{ihj\nu}$ and $\lambda_{j\nu}$ by an iterative method.

In our model, the parameters often converge to sub-optimal solutions, because switch errors existing in the sub-optimal configurations are not removed by gradual parameter changes. Therefore, we apply a heuristic procedure that re-runs the VBEM several times with twisted parameter configurations after every convergence:

- 1) Do VBEM and calculate the connectivities for all the sites.
- 2) Do another VBEM with a parameter set Λ that is twisted at a site with low connectivity.
- 3) Repeat until convergence.

Here, the twist of hyperparameters $\Lambda = \{\lambda_{j\nu}\}$ is defined similarly to that of parameters $\Theta = \{\theta_{j\nu}\}$. We describe the details of this procedure in the Additional file.

2.2.1.5 Inferring haplotypes

We set $p_j^\Phi(\nu)$ to the posterior mean estimate of $\theta_{j\nu}$ with respect to the converged posterior distribution:

$$p_j^\Phi(\nu) = \int d\Theta \theta_{j\nu} Q_\Theta(\Theta) = \frac{\lambda_{j\nu}}{\sum_{\nu'} \lambda_{j\nu'}} .$$

We select the phase ν at site j for which this $p_j^\Phi(\nu)$ is the highest. We limit the predicted haplotype segments to the regions with high MC values.

2.2.1.6 Possible extensions of the model

In this paper, we consider only the binary representation of heterozygous sites. We also constrain the error rate to be constant throughout the sequence. However, some of these constraints are easily removed. We can include homozygous sites and four nucleotide alleles by expanding the phase set Δ . For example, the phase set of a multi-allelic variant is represented like

$\Delta = \{(A,C),(A,G),(C,A),(C,G),(G,A),(G,C)\}$. We can even include small structural variations if they can be represented by additional allele symbols and the phase set of a structural variant is represented such as $\Delta_1 = \{(A,-),(-,A)\}$ for indel and $\Delta_2 = \{("AC","ACAC"),("ACAC","AC")\}$ for short tandem repeats. With these extensions, the accuracy of genotype calling of multi-allelic variants from sequencing data might be improved by considering haplotypes simultaneously [34] and the accuracy and the recall of the haplotype region might be improved because all variant sites add information to infer the derivation of the fragments. Furthermore, we can make the error probability matrix $p^e(\sigma|\sigma')$ dependent on the alleles of each fragment, which may be useful for incorporating the quality scores of sequenced reads.

2.2.2 Datasets and data processing

2.2.2.1 Dataset generation

Simulation data were created through a strategy similar to the one reported by Geraci [35]. We first generated M binary heterozygous phase vectors and then we generated SNP fragments by replicating each haplotype c times and randomly dividing them into subsequences of length between l_1 and l_2 . We then randomly flipped the binary values of the fragments from 0(1) to 1(0) with probability e . In the following, we use $M = 1000$, $c = 5$, $l_1 = 3$, $l_2 = 7$ and $e = 0.1$ unless otherwise mentioned.

For the real data, we used the dataset of Duitama's work [17], who conducted fosmid pool-based next-generation sequencing for HapMap trio child NA12878 from the CEU population. NA12878 had about 1.65×10^6 heterozygous sites on autosomal chromosome and the haplotypes of about 1.36×10^6 sites were determined by a trio-based statistical phasing method [22]. In the fosmid pool-based next-generation sequencing, the diploid genomic DNA was fragmented into pieces of length about 40 kilo-bases, and partitioned into 32 pools with low concentration, so that the fragments were long enough to span several heterozygous sites and each pool rarely contained homologous chromosomal regions of different haplotypes. Each pool was sequenced separately using a next-generation sequencer and the read data were mapped onto the reference genome. Since a read cluster in which the reads were close to each other and had the same pool origin were supposed to originate from the same DNA fragment, the alleles observed in the same cluster were merged into

a SNP fragment. Duitama [17] converted the fragment data to a binary representation by collecting only the alleles of the heterozygous sites determined by the 1000 genomes project. The coverage of the data was about 3.03. We used the trio-based data and the sequencing data in binary format for our experiment.

The normalized linkage disequilibrium D' for the CEU population was downloaded from the HapMap Project [6].

We compared our MixSIH software with ReFHap [17], FastHare [21], DGS [19], which were implemented by Duitama [17], and HapCUT [15]. We selected these algorithms because they have been shown to be superior to other algorithms [17].

For the comparison of the runtimes, we generated simulation data with $M = 100, 200, 500, 1000$. We repeated the measurement 10 times for each M and the average runtimes are reported here. The computations were performed on a cluster of Linux machines equipped with dual Xeon X5550 processors and 24 GB RAM.

As described in the introduction, our algorithm is focusing on extracting the reliable haplotype regions. To examine whether we have succeeded in extracting the reliable haplotype regions, an accuracy measure which evaluates the quality of the piecewise haplotype regions is needed. However, existing accuracy measures are designed to compare the efficiency between the algorithms and are not suitable for evaluating the quality of the piecewise haplotype regions.

Let $\Phi^{(t)}$ be the true haplotypes, and Φ be inferred haplotypes. Because the inferred haplotypes Φ are sets of partially assembled haplotype segments $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_B)$ where each of Φ_b is independently predicted, the accuracy measures have to be applicable for such predictions.

Many previous papers used the Hamming distance to measure the quality of assembled haplotypes [35]:

$$R(\Phi_0) = 1 - \frac{1}{2M} \min \left[D(\Phi_0, \Phi^{(t)}), D(\Phi_0, \bar{\Phi}^{(t)}) \right],$$

$$D(\Phi, \Phi') = \sum_{j=1}^M \sum_{h \in \mathcal{H}} I(\varphi_{jh} = \varphi'_{jh}),$$

where Φ_0 represents a fully assembled haplotype prediction and $I(a = b)$ represents the indicator function which assumes 1 if $a = b$ and 0 otherwise. A simple modification of the above formula to

the partially assembled haplotype segments might be

$$R'(\Phi) = 1 - \frac{1}{2M} \sum_{b=1}^B \min [D(\Phi_b, \Phi_b^{(t)}), D(\Phi_b, \bar{\Phi}_b^{(t)})] .$$

However, this definition is inconvenient because the minimization is applied for each segment and this accuracy measure can always be improved just by breaking a segment into smaller pieces at random positions.

The switch error rate [17] is another measure used for comparing SIH algorithms. A switch error is defined by the inconsistency between $\bar{\Phi}$ and $\Phi^{(t)}$ at neighboring heterozygous sites: $(\varphi_j, \varphi_{j+1}) = (\varphi_j^{(t)}, \bar{\varphi}_{j+1}^{(t)})$ or $(\bar{\varphi}_j^{(t)}, \varphi_{j+1}^{(t)})$. The switch error rate is defined by the total number of switch errors divided by the total number of neighboring pairs of heterozygous sites in all the segments. Although the switch error rate is useful for comparing different algorithms, it does not reflect the global influence of switch errors. Figure 2.2(B) shows the example of the case that the switch error rate is not suitable to evaluate the quality of the segments. A single switch error in the middle of a reconstructed haplotype segment has a greater influence on downstream analyses such as detecting amplified haplotypes [27] than a switch error located at an end of the segment (top and middle of Figure 2.2(B)). Two contiguous switch errors, which are likely to be caused by sequencing error or genotyping error, do not disrupt the consistency between front and back parts of the haplotype segments. However, such two contiguous switch error disrupt twice in terms of switch error rate (bottom of Figure 2.2(B)).

Here, we propose another simple accuracy measure based on the pairwise consistency of the prediction with the true haplotypes. This pairwise consistency score is inspired by the D' -measure of linkage disequilibrium where the statistical correlations among population genomes are measured for pair sites. Similarly to the switch error, a pair of heterozygous sites j and j' ($j < j'$) is defined as consistent if $(\varphi_j, \varphi_{j'}) = (\varphi_j^{(t)}, \varphi_{j'}^{(t)})$ or $(\bar{\varphi}_j^{(t)}, \bar{\varphi}_{j'}^{(t)})$, and inconsistent otherwise. A pair (j, j') in a haplotype segment is consistent if there is no switch error in range $[j, j']$ and inconsistent if there is one switch error in the segment. If there are uncontrolled number of switch errors in range $[j, j']$, the probabilities that pair (j, j') is consistent or inconsistent are both 0.5, which is equivalent to selecting a random phase at each site (Figure 2.2(A)). For each haplotype segment, we count the consistent and inconsistent pairs. The total numbers of consistent and inconsistent pairs over all the

2.2.2.2 Potential chimeric fragments

The processed sequence data derived from fosmid pool-based next-generation sequencing might contain chimeric fragments if a pool contains DNA fragments derived from the same region of different chromosomes and reads with different chromosomal origins are merged into a single SNP fragment. By using the trio-based haplotypes, we compute the ‘chimerity’ of each SNP fragment f by measuring the change of its likelihood after breaking it into two pieces:

$$\text{chimerity}(f) = -\log \left(\frac{\max_{h \in \mathcal{H}} P_0(f|h)}{\max_{j \in X(f), h \in \mathcal{H}} P_0(f_{\leq j}|h) P_0(f_{> j}|\bar{h})} \right),$$

$$P_0(f|h) = (1 - \alpha_0)^{n(f,h)} \alpha_0^{|X(f)| - n(f,h)},$$

where $n(f, h)$ is the number of sites at which the fragment f matches with the true haplotype h , $f_{\leq j}$ and $f_{> j}$ represent the left and right parts of fragment f divided at site j , and $\alpha_0 = 0.028$ is the empirical sequence error rate computed by comparing the true haplotypes and all the SNP fragments. We removed potential chimeric fragments with chimerity higher than a given threshold. We recomputed the accuracies for this removed dataset and compared them with those for the original dataset.

2.3 Results and discussion

2.3.1 Comparison of pairwise accuracies

We examined whether MixSIH can extract the accurate haplotypes regions by using MC. Figure 2.3 shows the accuracies derived from counting the consistent pairs. The x -axis is the number of predicted pairs (CP+IP) and the y -axis is the precision (CP/(CP+IP)). We have also shown the accuracy for the prediction without the haplotype assembly where the phase of each pair is determined by majority voting of spanning fragments. Figure 2.3A shows that the precisions of all the algorithms are around 0.5-0.6 at recall ~ 1.0 , indicating that there are many switch errors in the predictions and the quality of assembled haplotypes are not much different from picking phases randomly. By increasing the MC threshold, the precision of MixSIH improves rapidly and becomes

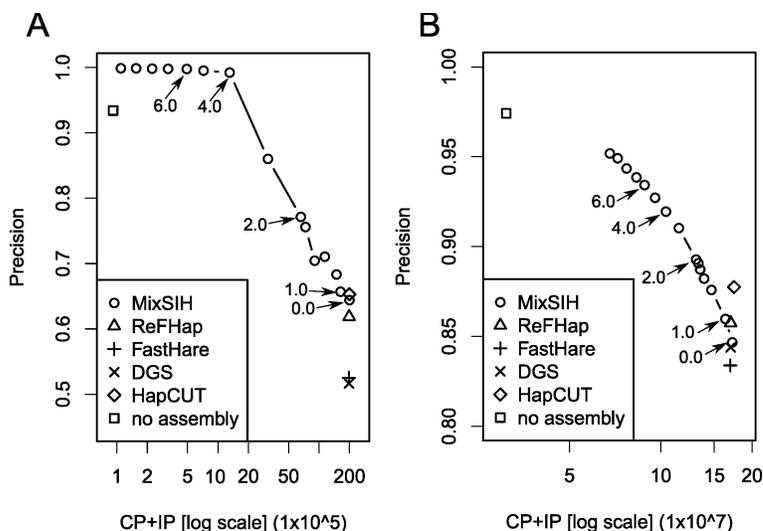


Figure 2.3. Precision curves based on the consistent pair counts. The x -axis represents the number of predicted pairs in log scale. The arrows indicate the MC thresholds. The accuracies are computed for the simulation dataset (A), and the real dataset (B): \square no assembly; \circ MixSIH; \triangle ReFHap; $+$ FastHare; \times DGS. In the simulation, we set $M = 2000$ and repeated the experiment 10 times for each algorithm; average values are plotted.

close to one around $MC = 4$ at recall 0.07. The recall of unassembled haplotypes is about 0.005 with precision 0.93, which is 20 times smaller than the recall 0.1 of MixSIH at the same precision. For the real dataset, the precision of the algorithms is around 0.85 at recall ~ 1.0 , which is much higher than the precision for the simulation dataset. This is because there are many small fragment clusters for which the correct haplotypes are easily predicted. The accuracy of MixSIH can still be improved with precision up to 0.95 at the expense of deleting about 3/5 of weakly supported pairs from the prediction. However, it does not reach the precision of unassembled haplotype prediction. We discuss this issue in the next subsection.

2.3.2 Effects of potential chimeric fragments

Inspecting the switch errors in the prediction for the real dataset, we found that there are potential chimeric fragments that have a considerable effect on the pairwise accuracies. A chimeric fragment is defined as a fragment whose left and right parts match different chromosomes very well. Such fragments can occur in fosmid pool-based next-generation sequencing data. We show the chimerity distribution in the Additional file. We computed the accuracy of MixSIH for a fragment dataset in which the fragments with chimerity higher than a given threshold are removed. We

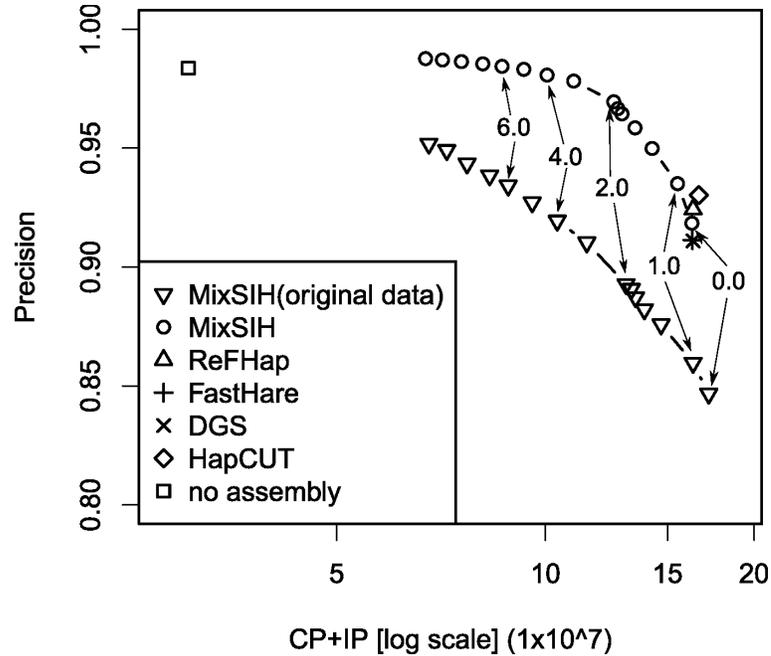


Figure 2.4. The precisions of the algorithms for the dataset in which fragments with chimerity greater than 10 are removed. For comparison, the precisions of MixSIH for the original dataset are also shown as diamonds.

experimented with several chimerity thresholds and we found that the accuracy improves with decreasing chimerity thresholds and saturated at about chimerity threshold 10, which corresponded to the case that only 1.65% (4,482/271,184) of the fragments were removed. We show the accuracies for different chimerity thresholds in the Additional file. We also show that the fragments whose chimerity is over 10 are indeed chimeric in the Additional file. Figure 2.4 shows the precision curves for the dataset of removed fragments. The accuracies are considerably higher for this dataset, and the precision now reaches that of the unassembled prediction at recall 0.5 with MC threshold 6.0. We also show the effects of chimeric fragments on simulation data in the Additional file.

These results suggest that more careful data processing to avoid spurious chimeric fragments is necessary to obtain high-quality haplotype assembly.

2.3.3 Incorporation of the trio-based data

Although the trio-based statistical phasing method can determine most of the phases of the sites, there still exist SNP sites whose phases cannot be determined by this method. SIH is capable of determining the phases which are not determined by the trio-based data, and we can obtain more

complete haplotypes data by combining both of the SIH-based data and the trio-based data. To examine how many phases of the sites can be determined anew by combining both of the SIH-based data and the trio-based data, we devise a method that combines both information to determine the phases (see the Additional file). By using this method, about 82% (237,950/291,466) of the phases of the sites which are undetermined by trio-based data could be determined anew and totally about 97% (1,601,381/1,654,897) of the phases could be determined by both the methods. This result suggests that almost all of the phases of the sites can be determined by using both of the SIH-based data and the trio-based data.

2.3.4 Spatial distribution of MC values

Figure 2.5A shows an example of the spatial distribution of the MC values for the real dataset. The regions that are densely covered tend to have large MC values. On the other hand, the MC values are low in chromosomal regions with sparse heterozygous sites because few fragments span two or more sites. Figure 2.5B shows the density plot of MC values which are converted to the corresponding precisions using the graph of Figure 2.5B, and the absolute normalized linkage disequilibrium $|D'|$. SIH can accurately infer the haplotypes in many regions with low linkage disequilibrium, but there are also regions with reduced precision and high $|D'|$ values. This suggests that the accuracy of predictions might be improved by using both pieces of information.

2.3.5 Dependency of MC values on the fragment parameters

Figure 2.6 shows the dependency of MC values on the quality of the input dataset. In these figures, the minimal MC threshold that achieves precision ≥ 0.95 (y -axis) is plotted for different fragment length ranges $[l_1, l_2]$ (three panels), coverages c (three lines), and error rates e (x -axis). They show that the MC threshold must be increased to obtain high-quality assembly for low-coverage, highly erroneous data, while it has a minor dependence on the typical fragment length. However, the overall scale of the MC threshold changes relatively moderately and it is bounded above at $MC = 6$ for the tested cases. We also calculated the dependency of MC values on the input dataset which include chimeric fragments and the results were almost the same (see the Additional file). Hence we set the default MC threshold to 6.0 in our software.

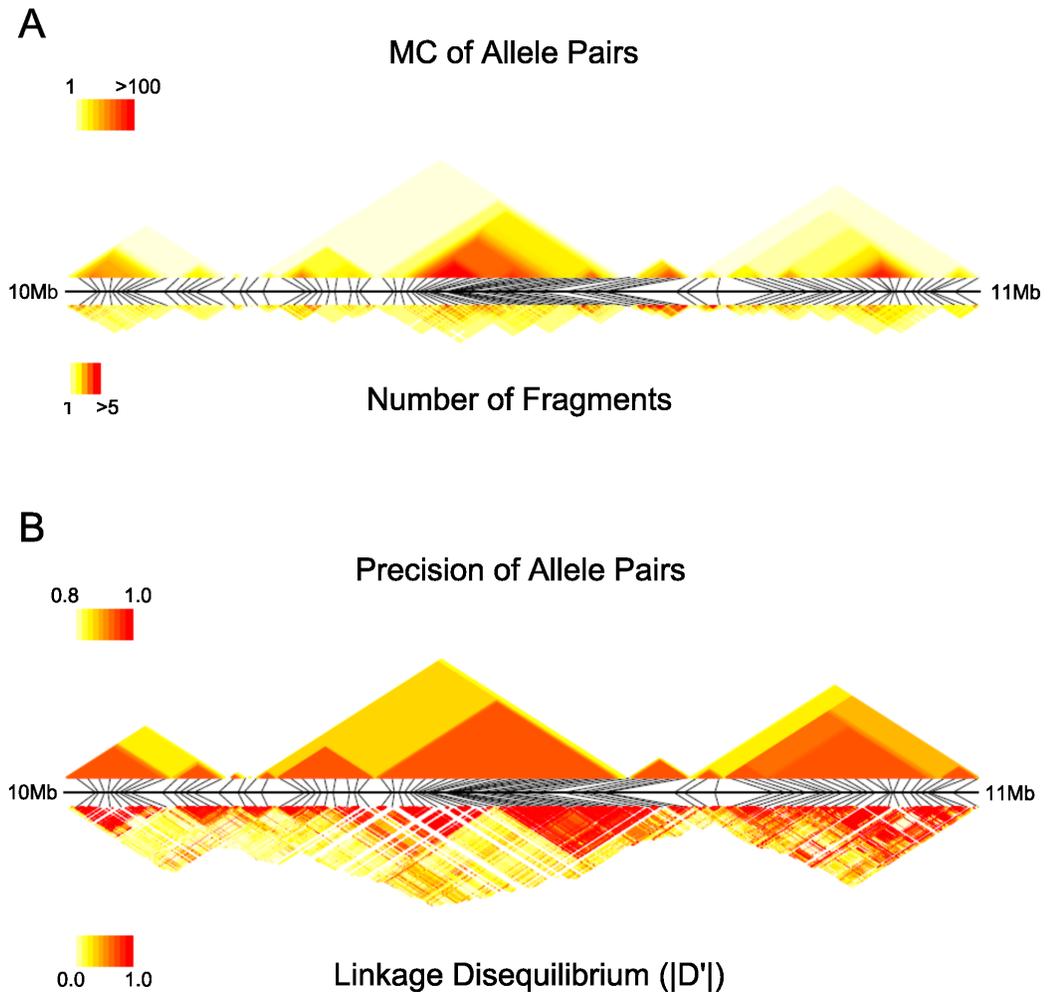


Figure 2.5. Spatial distribution of MC and LD. A. A colored density plot of the MC values and the number of fragments. The x -axis represents the coordinates of heterozygous sites. The actual locations of the sites in genome coordinates are shown by thin black diagonal lines and the black horizontal line represents a 10-11 megabase region of chromosome 20. The upper densities represent the connectivity values. The lower densities represent the number of fragments spanning the pair sites. B. A colored density plot of the precisions (upper) and the absolute normalized linkage disequilibrium $|D'|$ (lower) for the same region.

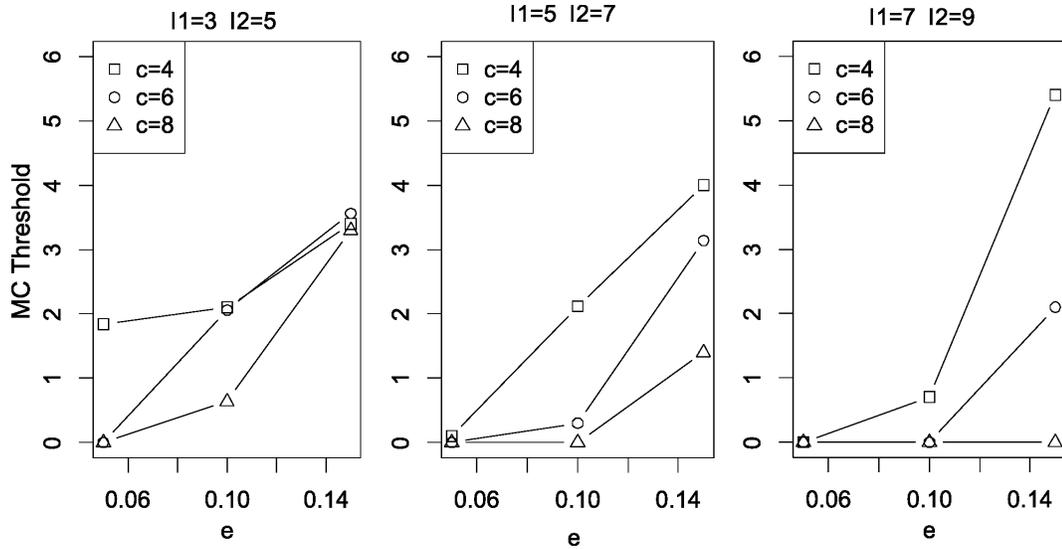


Figure 2.6. Dependency of the lowest MC value with precision ≥ 0.95 for coverage c , fragment length $[l_1, l_2]$, and error rate e . The experiments were repeated 10 times, and the average values are plotted.

2.3.6 Optimality of inferred parameters

We use a heuristic method for parameter optimization to avoid sub-optimal solutions. To test whether the optimized parameters actually reach the global optimum, we compared the log likelihood of the optimized parameters with the approximate maximal log likelihood obtained by optimizing the parameters with an initial condition in which the optimal solution falls into the set of true haplotypes; we add one to the Dirichlet parameters for the true phase probability: that is, $\lambda_{j\nu} = \lambda_{j\nu}^{(0)} + 1$ if $\nu = \varphi_j^{(t)}$ and $\lambda_{j\nu} = \lambda_{j\nu}^{(0)}$ otherwise, where $\lambda_{j\nu}^{(0)}$ is hyperparameters of the Dirichlet distribution and $\varphi_j^{(t)}$ is the true phase at site j . Figure 2.7 shows the changes of the log likelihood for each twist operation. It also shows the connectivity values at the sites where the parameters Λ are twisted. The log likelihood increases monotonically and reaches the approximate maximal likelihood after 50 twist iterations. The connectivity values also increase monotonically in most cases. The figure implies that the parameters converge to the global optimum upon repeating the twist operation.

2.3.7 Comparison of running times

Figure 2.8 shows the runtimes of the test programs. Bansal released the faster version of HapCUT recently, so we calculated the runtimes of both latest and previous version of HapCUT. Our

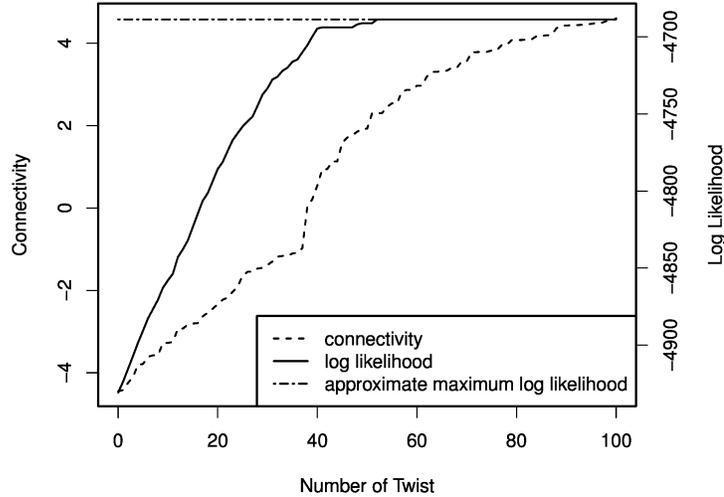


Figure 2.7. Increase of log likelihood values for each iteration. The dotted line represents the approximate maximal log likelihood; the solid line, the changes of the optimized log likelihood for each twist operation; the broken line, the connectivity values at the positions that the optimizing parameters are twisted.

method applies the VBEM algorithm repeatedly and hence is rather slow. It is comparative to HapCUT(previous versoin) and about 10-fold slower than both ReFHap and HapCUT(latest versoin), and from 50-fold to 500-fold slower than both FastHare and DGS. Considering that NA12878 has about 1.23×10^5 heterozygous sites on chromosome 1, it is roughly estimated that MixSIH takes about 15 days to finish haplotyping for the data whose connected component includes all heterozygous sites, and MixSIH is still manageable for such chromosome-wide data.

2.4 Conclusions

With advances in sequencing technologies and experimental techniques, single individual haplotyping (SIH) has become increasingly appealing for haplotype determination in recent years. In this paper, we have developed a probabilistic model for SIH (MixSIH) and defined the minimal connectivity (MC) score that can be used for extracting accurately assembled haplotype segments. We have introduced a new accuracy measure, based on the pairwise consistency of the inferred haplotypes, which is intuitive and easy to calculate but nevertheless avoids some of the problems of existing accuracy measures. By using the MC scores our algorithm can extract highly accurate haplotype segments. We have also found evidence that there are a small number of chimeric fragments in an

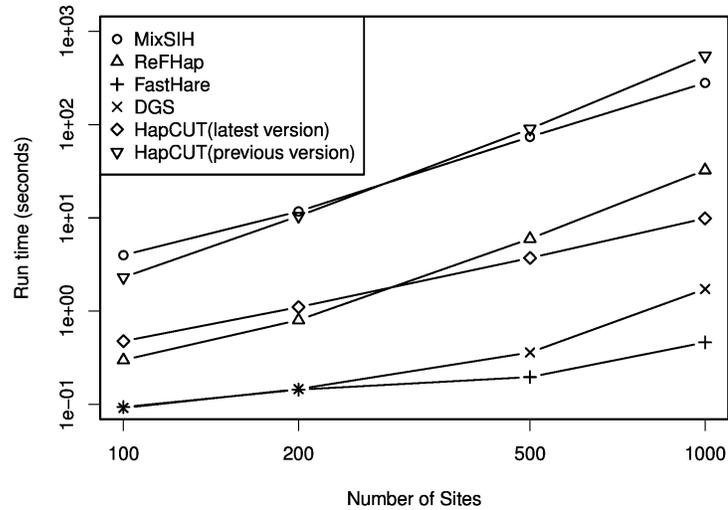


Figure 2.8. The running times of the tested algorithms. The x -axis is the number of sites. The y -axis is the running time in seconds. Both are displayed on a logarithmic scale.

existing dataset from fosmid pool-based next-generation sequencing, and these fragments considerably reduce the quality of the assembled haplotypes. Therefore, a better data processing method is necessary to avoid creating chimeric fragments.

Our program uses only read fragment data derived from an individual. However, it is expected that more powerful analyses could be made by combining SIH algorithms with statistical haplotype phasing methods that use population genotype data. An interesting possibility would be to construct a unified probabilistic model that infers the haplotypes on the basis of both kinds of data.

2.5 Supplementary text

2.5.1 Difference Between Our Model and Existing Models

There are a number of differences between our model and those of [18] and [20]. Our model takes a ‘mixture model’ approach: each fragment is emitted independently of the other fragments and a partial phase vector $\Phi^{(i)} \in \Delta(f_i)$ is independently drawn for each fragment f_i :

$$P(F|\Theta) = \sum_{H \in \mathcal{H}^{\otimes N}} \prod_{i=1}^N \sum_{\Phi^{(i)} \in \Delta(f_i)} P(f_i|h_i, \Phi^{(i)}) p^m(h_i) P(\Phi^{(i)})$$

On the other hand, [18] and [20] take a ‘hidden variables’ approach: the model first draws a full-length phase vector Φ , then all the fragments are emitted from this common phase vector Φ :

$$P(F|\Theta) = \sum_{\Phi \in \Delta^{\otimes M}} P(\Phi) \sum_{H \in \mathcal{H}^{\otimes N}} \prod_{i=1}^N P(f_i|h_i, \Phi) p^m(h_i)$$

Although their model might look somewhat more natural, since the fragments are actually derived from the fixed true chromosomes, the computation of the likelihood function is quite costly; we need either to traverse all the $|\Delta|^M$ -phase patterns (where $|\Delta|$ is the number of possible phases at each site), or to traverse all the $2^{|F^c(j)|}$ -patterns for assigning haplotype origins $h_i \in \mathcal{H}$ to covering fragments $f_i \in F^c(j)$ for each site j . Therefore, it is impractical to use their model to compute a likelihood for genome-scale data. On the other hand, our model considers only one fragment at a time and the complexity of the likelihood computation is only $|\Delta| \times \sum_{i=1}^N |X(f_i)|$. Although our model loses some complicated correlations among fragments, it still takes into account the allele co-occurrences within each fragment.

2.5.2 Variational Bayes Expectation Maximization Algorithm

We set the prior probabilities for parameters Θ to be those of the Dirichlet distribution with hyperparameters $\Lambda^{(0)} = \{\lambda_{j\nu}^{(0)}\}$:

$$P(\Theta) = \prod_{j=1}^M \text{Dir}(\theta_j | \lambda_j^{(0)})$$

$$\text{Dir}(\theta_j | \lambda_j^{(0)}) = Z(\lambda_j^{(0)})^{-1} \prod_{\nu} (\theta_{j\nu})^{\lambda_{j\nu}^{(0)}}$$

$$Z(\lambda_j^{(0)}) = \left[\prod_{\nu} \Gamma(\lambda_{j\nu}^{(0)}) \right] / \Gamma(\sum_{\nu} \lambda_{j\nu}^{(0)}),$$

where $\Gamma(x)$ is the gamma function, and we set $\lambda_{j\nu}^{(0)} = 0.5$ for all j and ν .

The solutions for $Q^{H\Psi}(H, \Psi)$ and $Q^{\Theta}(\Theta)$ have the form

$$Q^{H\Psi}(H, \Psi) = \frac{1}{Z^{H\Psi}} \exp \left(\sum_{i=1}^N \sum_{h \in \mathcal{H}} \sum_{j \in X(f_i)} \sum_{\nu \in \Delta} \mathcal{I}_{ihj\nu} \log(\beta_{ihj\nu}) \right),$$

$$Q^{\Theta}(\Theta) = \prod_{j=1}^M \text{Dir}(\theta_j | \lambda_{j\nu}),$$

where $Z^{H\Psi}$ represents a normalization constant and $\beta_{ihj\nu}$ and $\lambda_{j\nu}$ are the hyperparameters that specify the posterior distributions. Because $Q^{H\Psi}(H, \Psi)$ and $Q^{\Theta}(\Theta)$ are dependent on each other through the dependencies among the hyperparameters, they cannot be found simultaneously. Therefore, we optimize $\beta_{ihj\nu}$ and $\lambda_{j\nu}$ by repeating two computational procedures, called VBE and VBM.

In the VBE step, we calculate the expectations

$$\gamma_{ihj\nu} = \sum_{H\Psi} \mathcal{I}_{ihj\nu} Q^{H\Psi}(H, \Psi) = \gamma_{ih}^{(1)} \gamma_{ihj\nu}^{(2)},$$

$$\gamma_{ih}^{(1)} = \frac{\prod_{j \in X(f_i)} (\sum_{\nu \in \Delta} \beta_{ihj\nu})}{\sum_{h'} \prod_{j \in X(f_i)} (\sum_{\nu \in \Delta} \beta_{ih'j\nu})},$$

$$\gamma_{ihj\nu}^{(2)} = \frac{\beta_{ihj\nu}}{\sum_{\nu' \in \Delta} \beta_{ihj\nu'}}.$$

In the VBM step, we update the Dirichlet parameters $\lambda_{j\nu}$ and then compute expectation $w_{j\nu}$ as well as $\beta_{ihj\nu}$:

$$\lambda_{j\nu} = \lambda_{j\nu}^{(0)} + \sum_{i=1}^N \sum_{h \in \mathcal{H}} \gamma_{ihj\nu},$$

$$w_{j\nu} = \int d\Theta \log(\theta_{j\nu}) Q^{\Theta}(\Theta) = \psi(\lambda_{j\nu}) - \psi(\sum_{\nu} \lambda_{j\nu}),$$

$$\beta_{ihj\nu} = p^e(f_{ij} | \nu_h) \exp(w_{j\nu}).$$

2.5.3 Iterative Twist Operations to Avoid Sub-optimal Solutions

We optimize the parameters as follows.

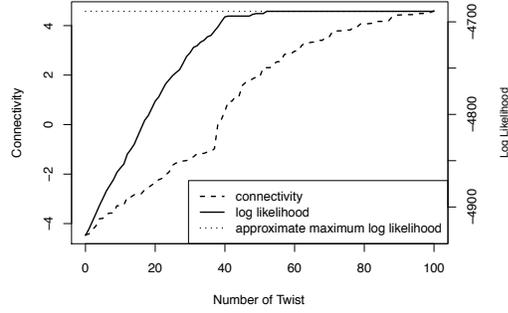


Figure 2.9. Increase of log likelihood values for each iteration. The dotted line represents the approximate maximal log likelihood; the solid line, the changes of the optimized log likelihood for each twist operation; the broken line, the connectivity values at the positions that the optimizing parameters are twisted.

- 1) Set $\lambda_{k\nu}^{(0)} = 0.5$ for all k and ν and initialize Λ with $\lambda_{k\nu} = \lambda_{k\nu}^{(0)} + r_{k\nu}$. Here, $r_{k\nu}$ are random numbers sampled from the uniform distribution in the range $[0.0, 0.1]$. They are necessary to avoid the symmetric point of the likelihood function. Let S be the empty set, and set $\text{score} = -\infty$ and $\Lambda_1 = \Lambda$.
- 2) Do variational Bayes expectation maximization [33] with initial parameter Λ_1 until the parameters converge or the number of iterations exceeds a given limit (100). Let score' and Λ' denote the converged likelihood and converged parameter set, respectively.
- 3) If $\text{score} < \text{score}'$ then set $\text{score} = \text{score}'$, $\Lambda = \Lambda'$.
- 4) Select the site j out of sites $X \setminus S$ that has the smallest connectivity c_j with respect to the model Λ .
- 5) Add j to S if j has already been selected once in the previous iterations.
- 6) Set $\Lambda_1 = \Lambda$ and twist Λ_1 at site j . (The concept of ‘twisting’ is described in ‘The Minimum Connectivity Score’ subsection in the main paper.)
- 7) If $c_j > 7.0$ or $X = S$, then terminate, otherwise go to step 2.

2.5.3.1 Optimality of Inferred Parameters

We use a heuristic method for parameter optimization to avoid sub-optimal solutions. To test whether the optimized parameters actually reach the global optimum, we compared the log like-

likelihood of the optimized parameters with the approximate maximal log likelihood obtained by optimizing the parameters with an initial condition in which the optimal solution falls into the set of true haplotypes; we add one to the Dirichlet parameters for the true phase probability: that is, $\lambda_{j\nu} = \lambda_{j\nu}^{(0)} + 1$ if $\nu = \varphi_j^{(t)}$ and $\lambda_{j\nu} = \lambda_{j\nu}^{(0)}$ otherwise, where $\varphi_j^{(t)}$ is the true phase at site j . Figure 2.9 shows the changes of the log likelihood for each twist operation. It also shows the connectivity values at the sites where the parameters Λ are twisted. The log likelihood increases monotonically and reaches the approximate maximal likelihood after 50 twist iterations. The connectivity values also increase monotonically in most cases. The figure implies that the parameters converge to the global optimum upon repeating the twist operation.

2.5.4 Comparison of Accuracy Measures

Because of the equivalence of predictions between the switched haplotypes as explained above, measuring the difference between $\Phi^{(t)}$ and Φ is nontrivial. Many previous papers used the Hamming distance to measure the quality of assembled haplotypes [35]:

$$R(\Phi) = 1 - \frac{1}{2M} \min \left[D(\Phi, \Phi^{(t)}), D(\Phi, \bar{\Phi}^{(t)}) \right],$$

$$D(\Phi, \Phi') = \sum_{j=1}^M \sum_{h \in \mathcal{H}} I(\varphi_{jh} = \varphi'_{jh}),$$

where $I(a = b)$ represents the indicator function which assumes 1 if $a = b$ and 0 otherwise. This definition is not appropriate when we consider the accuracy of multiple, partially resolved haplotype segments. For example, there is no way for the SIH algorithms to relate the haplotypes of chromosome 1 to those of chromosome 2 because there is no fragment that overlaps with both the chromosomes. It is also impossible for any SIH algorithm to relate the haplotypes of two consecutive regions if there is no fragment that overlaps with both regions. Furthermore, we wish to extract confidently assembled sub-regions using the minimum connectivity thresholds. Therefore, it is desirable for the accuracy measures to allow comparisons on the set of partially assembled haplotype segments. We now consider a simple extension of the Hamming distance measure. Let $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_B)$ be the set of partially assembled haplotype segments with M total sites,

then a simple modification of the above formula might be

$$R'(\Phi) = 1 - \frac{1}{2M} \sum_{b=1}^B \min [D(\Phi_b, \Phi_b^{(t)}), D(\Phi_b, \bar{\Phi}_b^{(t)})] .$$

However, this definition is inconvenient because the minimization is applied for each segment and this accuracy measure can always be improved just by breaking a segment into smaller pieces at random positions.

The switch error rate [17] is another measure used for comparing SIH algorithms. A switch error is defined by the inconsistency between Φ and $\Phi^{(t)}$ at neighboring heterozygous sites: $(\varphi_j, \varphi_{j+1}) = (\varphi_j^{(t)}, \bar{\varphi}_{j+1}^{(t)})$ or $(\bar{\varphi}_j^{(t)}, \varphi_{j+1}^{(t)})$. The switch error rate is defined by the total number of switch errors divided by the total number of neighboring pairs of heterozygous sites in all the segments. Although the switch error rate is useful for comparing different algorithms, it does not reflect the global influence of switch errors. For example, a single switch error in the middle of a reconstructed haplotype segment has a greater influence on downstream analyses, through incorrect prediction of allele co-occurrences, than a switch error located at an end of the segment.

There are other measures, such as the minimum number of entries to correct (MEC) [36], the adjusted N50 (AN50) and its variants S50, N50 [37], and the quality adjusted N50 (QAN50). Apart from QAN50, these measures do not use the true haplotypes and there is no guarantee that the correct haplotypes have a higher score than incorrect ones. The procedure to compute the QAN50 score is complex and can be roughly described as follows. First the prediction is broken into smaller segments that do not contain any switch errors. For each segment an adjusted length score, which is the segment length in the reference genome multiplied by the proportion of heterozygous sites inside of the segment, is assigned. The segments are sorted in order of decreasing adjusted length scores and AN50 is defined as the threshold score such that half of heterozygous sites are covered by segments with scores greater than the threshold. Although this measure accounts for both the quality and segment sizes of the reconstruction, the complex interactions between inhomogeneity of the SNP density and fragment coverage seem to make it difficult to understand the practical utility of SIH algorithms by using their QAN50 scores.

In comparison to the switch error rate, which cannot account for genotyping errors in homozygous sites, the pairwise consistency score works without modification in the cases where homozy-

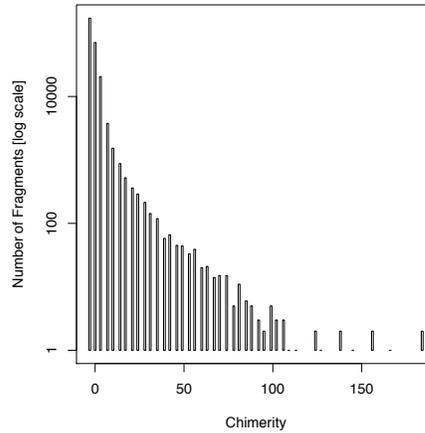


Figure 2.10. Chimerity distribution of the real dataset.

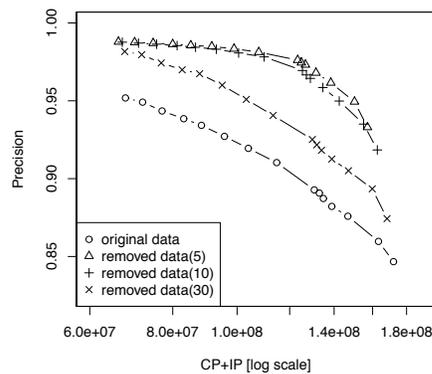


Figure 2.11. The precisions for the original dataset (\circ) and the datasets in which the fragments with chimerity greater than 5 (Δ), 10 ($+$), and 30 (\times) are removed.

gous sites are included in the prediction space. Furthermore, although the notion of pairwise consistency is applicable to haplotype segments that are not made up of simple contiguous sites, the definition of a switch error for such segments is somewhat ambiguous.

2.5.5 Potential Chimeric Fragments

Figure 2.10 shows the chimerity distribution of real data [17], which indicates that only a small proportion of the data has high chimerity. Figure 2.11 shows the accuracies for different chimerity thresholds, which suggests that the improvement of the accuracies saturates at around chimerity threshold 10.

Chapter 3

Integrating dilution-based sequencing and population genotypes for single individual haplotyping

3.1 Background

Advances in experimental techniques for DNA sequencing and genotyping have made it possible to determine many individual human genomes and detect variations, such as single nucleotide polymorphisms (SNPs) [22, 38]. This has brought about great progress in genome analyses, such as genome-wide association studies (GWAS) [39], inference of population structure [40], and expression phenotypes [41]. However, most technologies give only genotype information and most current research does not determine the haplotype origin of the variations. Haplotypes contain more detailed information than genotypes and are valuable for GWAS [5], and for analyzing genetic structures such as linkage disequilibrium, recombination patterns [38], and correlations between variations and diseases [7].

Determining haplotypes experimentally is difficult, and there are three main computational approaches for haplotype inference. The first approach is the statistical phasing method, which infers population haplotypes from population genotypes using statistical computation [8–11, 13]. Algorithms for statistical phasing have been developed in response to technological advances for genotyping, and its basis is that the diversity of haplotypes is limited, and there are conserved haplo-

types [42]. Because of the strategy, statistical phasing does not work well in chromosomal regions which exhibit several different haplotypes, particularly regions of low linkage disequilibrium. This approach is also weak in inferring long haplotypes because the complexity of population haplotypes increases exponentially according to the number of SNPs.

In the second approach, haplotypes are inferred from genotypes of pedigrees. For example, a child's haplotypes are determined from the genotypes of child and its parents (trio-based haplotyping). The origin of child's alleles can be determined if only one of the parents has the same alleles. However, the haplotypes of sites at which all family members have the same genotype cannot be determined and, furthermore, family genotype data are not always available. In addition, neither the statistical phasing method nor this approach can identify spontaneous mutations.

The third approach uses DNA sequencing data and is called single individual haplotyping (SIH) or haplotype assembly [14–21, 43]. SIH utilizes the fact that each sequenced read is derived from only one of the haplotypes. If a read spans two or more heterozygous sites, the haplotype can be determined from the co-occurrence of alleles in the read. Two reads are determined to originate from the same chromosome if they overlap at a region that has at least one heterozygous site, and they have the same alleles at these sites.

SIH did not attract much attention until recently, since it needed long DNA sequencing reads that spanned multiple heterozygous sites, and obtaining such reads quickly and economically was difficult. However, this situation is changing rapidly with the advent of new experimental techniques, such as fosmid pool-based next-generation sequencing [17, 24, 25], long read fragment technology [44], and dilution-amplification-based sequencing [45] that can produce virtual long reads. In these methods, long DNA fragments are separated into distinct low-concentration aliquots which each contain less than one fragment per chromosomal region. After sequencing an aliquot with a next-generation sequencer and mapping short reads, clusters are formed in which the reads are close to each other. A cluster corresponds to a long DNA fragment and is supposed to be derived from a single haplotype. Thus, virtual long reads can be obtained by merging the short reads in a cluster (see Figure 3.1).

Although such experimental techniques are sophisticated, they have the problem of producing chimeric fragments whose left and right parts match different chromosomes very well. Because long DNA fragments are separated into aliquots randomly, there are cases where an aliquot has some long

DNA fragments derived from the same region of different chromosomes and, consequently, reads with different chromosomal origins are regarded as one cluster and merged into a single fragment (see Figure 3.1). In the process of developing MixSIH [43], which is the first SIH algorithm that can evaluate the reliability of a haplotype region, we have shown that such chimeric fragments significantly decrease the accuracy of SIH. This is because the chimeric fragments can lead to opposite haplotypes between right and left of haplotype regions.

In our previous study we detected chimeric fragments under the condition that parents genotypes were given. However, independence from pedigree data is one of the advantages of SIH and, therefore, it is common to assume that pedigree genotypes are not available. Even if pedigree genotypes are available, there are regions whose haplotypes are not determined from pedigree genotypes and the chimeric fragments in such regions cannot be detected with the previous method. The length of a reads cluster and heterozygous calls in a reads cluster were also used for detecting chimeric fragments [17]. The length of a reads cluster which correspond to a chimeric fragment will be larger than that of most reads clusters because reads with different long DNA fragment origins are regarded as one cluster and merged into one fragment. In addition, if there are some heterozygous SNPs in an overlapped region where reads with different haplotype origins coexist, these SNPs will show heterozygous calls in a reads cluster. Although some chimeric fragments will be detected with cluster length and heterozygous calls, considerable number of chimeric fragments will be left behind because of the dispersion of the cluster lengths, and non-detection of the heterozygous calls in the overlapped regions due to the lack of coverage and absence of heterozygous SNPs. For these reasons, chimeric fragment detection method which does not depend on pedigree genotypes and can detect chimeric fragments which are overlooked by the cluster length and the heterozygous calls is necessary to obtain high quality assembled haplotypes.

In this paper, we propose a general method to detect chimeric fragments without using pedigree genotypes. Our method is based on the assumption that chimeric fragments are derived artificially and differ from the biological conserved haplotypes in the population. Under this assumption, we use population genotypes to evaluate inconsistency between virtual long read and inferred haplotypes.

Previous researches showed that the quality of haplotype inference will increase by integration of SIH and statistical phasing [46–48]. These approach basically consider the SNP fragments

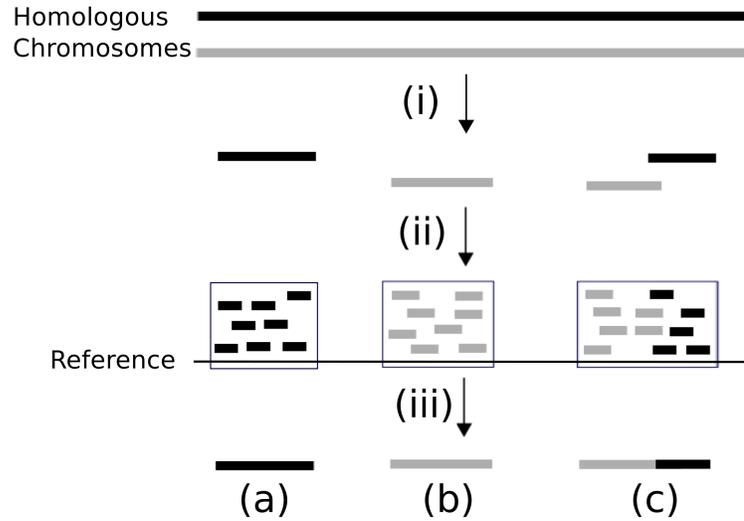


Figure 3.1. An illustration of dilution-based sequencing. (i) The DNA fragments are separated into multiple low-concentration dilutions. (ii) After sequencing and mapping an aliquot, mapped reads form clusters which correspond to DNA fragments. (iii) Clusters are merged into read fragments and result in natural fragments (a), (b) and a chimeric fragment (c). Chimeric fragments are produced when short reads derived from multiple DNA fragments are regarded as one cluster.

as reliable information sources and use population haplotypes to supplement inferred individual haplotypes. Therefore, these approaches will not be suitable for preventing the effect of chimeric fragments, which are unreliable and can lead to incorrect haplotypes. Our research presents the importance of considering chimeric fragments in SIH and proposes a novel strategy for integration by focusing on the process of dilution-based sequencing.

We applied our method to two real datasets and showed that the chimeric fragments could be detected with high accuracy. Moreover, we compared the accuracy of multiple SIH algorithm for before and after removing chimeric fragments candidates. We found that accuracy of assembled haplotypes improved considerably after chimeric fragment candidates were removed using our method. In addition, we found that SIH algorithm successfully inferred long haplotypes and showed the usefulness of SIH.

3.2 Methods

3.2.1 Notation

Throughout the paper, we denote chimeric fragment as CF, and natural fragment as NF.

Because SIH is trivial for homozygous sites and because it is usually much easier to determine the genotype than to determine the haplotypes, we focus on heterozygous sites and represent heterozygous alleles by a simple binary representation. Fragments from which the homozygous sites have been removed are called SNP fragments. SNP fragments are represented by $F = \{f_i | i = 1, \dots, N\}$, and fragment f_i takes value $f_{ij} \in \{0, 1\}$ at site j if f_i covers the site. We denote the set of sites which f_i covers by $X(f_i)$.

3.2.2 Statistical phasing method

In this paper, we describe a method to detect CFs by using statistical phasing. The statistical phasing method estimates haplotypes from population genotype data based on the fact that the diversity of local haplotypes is low.

Here, we use the software PHASE (version 2.1.1) with default settings for statistical phasing [10, 11]. PHASE infers haplotypes of the specified set of SNPs S using the Gibbs sampling method which samples each individual in a random order, updates the individual haplotypes under the assumption that all the other haplotypes are given, and repeats this process. PHASE outputs several candidate haplotypes and their probabilities for each individual. In detecting CFs, we are interested in the individual haplotypes of the individual who is the target of SIH and denote the set of candidate haplotypes for the individual by $H^{(p)} = \{H_i^{(p)} | i = 1, \dots, M\}$, where M is the number of candidates and $H_i^{(p)}$ is composed of the haplotype pair $H_{i0}^{(p)}$ and $H_{i1}^{(p)}$. $H_{ij}^{(p)}$ is composed of the set of $H_{ijk}^{(p)}$ ($k \in S$) which represent the binary allele of the haplotype $H_{ij}^{(p)}$ at site k . We denote the probability of $H_i^{(p)}$ for the individual by $P(H_i^{(p)})$.

3.2.3 Chimeric fragment detection model

We model probabilities that a fragment f_i is NF ($P^n(f_i)$) and CF ($P^c(f_i)$), and develop an indicator for detecting CF with these probabilities. Upon the calculation of the NF and CF probabilities of a fragment, we obtain $H^{(p)}$ and $P(H_i^{(p)})$ by running PHASE for $S = X(f_i)$.

The NF probability of fragment f_i is composed of the probability of the individual haplotypes and the probability of the SNP fragment given the haplotypes:

$$P^n(f_i) = \sum_{j=1}^M P(H_j^{(p)}) P^n(f_i | H_j^{(p)}),$$

$$\begin{aligned}
P^n(f_i|H_j^{(p)}) &= \frac{1}{2} \left(P^n(f_i|H_{j0}^{(p)}) + P^n(f_i|H_{j1}^{(p)}) \right) , \\
P^n(f_i|H_{jk}^{(p)}) &= \prod_{l \in X(f_i)} P(f_{il}|H_{jkl}^{(p)}) , \\
P(f_{il}|H_{jkl}^{(p)}) &= \begin{cases} (1 - \alpha) & \text{for } f_{il} = H_{jkl}^{(p)} \\ \alpha & \text{for } f_{il} \neq H_{jkl}^{(p)} , \end{cases}
\end{aligned}$$

where α is a error term to deal with sequencing and PHASE errors. In this paper, we use $\alpha = 0.01$ (the effect of changing α is described in the Additional file).

The CF probability is similar to the NF probability, but the probability of SNP fragments given haplotypes is slightly different. $P^c(f_i|H_{jk}^{(p)})$ is calculated by assuming that left and right parts of f_i are derived from different haplotypes in a haplotype pair:

$$\begin{aligned}
P^c(f_i) &= \sum_{j=1}^M \left(P(H_j^{(p)}) \frac{1}{2} \sum_{k=0}^1 P^c(f_i|H_{jk}^{(p)}) \right) , \\
P^c(f_i|H_{jk}^{(p)}) &= \sum_{l \in X(f_i)} \left(\prod_{m \leq l} P(f_{im}|H_{jkm}^{(p)}) \prod_{m > l} P(f_{im}|H_{j\bar{k}m}^{(p)}) \right) ,
\end{aligned}$$

where $\bar{0} = 1$ and $\bar{1} = 0$. Although we assume that the CF changes the origin of haplotype only once, it is possible that a CF changes the derivation many times over. However, such a CF would be rare and the CF probability given above would, in such a situation, approximate the result obtained by marginalizing over the switched sites.

Using these probabilities, we would like to define an indicator that evaluates the degree of artificiality of a recombinant SNP fragment which we will call the ‘chimerity based on statistical phasing’ (CSP). In principle, we would like to use

$$\text{CSP}(f_i) = \ln P^c(f_i) - \ln P^n(f_i) .$$

However, because the number of possible haplotypes and their combinations increase exponentially and the running time of PHASE increases according to SNP fragment size, we use a sliding-window approach to calculate CSP if the size of a SNP fragment is over sliding window

width:

$$\text{CSP}(f_i) = \max_{\beta \in X'(f_i)} \left(\ln P^c(f_i^{(\beta, \beta+W-1)}) - \ln P^n(f_i^{(\beta, \beta+W-1)}) \right),$$

where $f_i^{(\beta, \beta+W-1)}$ is the partial fragment of f_i which starts from the β th site and whose size is W . $X'(f_i)$ is $X(f_i)$ in which $X(f_i^{(\gamma, \gamma+W-1)})$ is removed, where $f_i^{(\gamma, \gamma+W-1)}$ is the rightmost partial fragment. W is the sliding window width and we use $W = 5$ for the default setting (see the Additional file for the effect of changing W). In the process of sliding window calculation, $H^{(p)}$ and $P(H_i^{(p)})$ are obtained by running PHASE for $S = X(f_i^{(\beta, \beta+W-1)})$.

We detect the CF candidates in a set of SNP fragments by selecting the SNP fragments whose CSP are larger than a threshold.

3.2.4 Cluster length and heterozygous calls for detecting chimeric fragment

In the previous research, the length of a reads cluster and heterozygous calls in a reads cluster were used for filtering CFs [17]. Because a CF is produced when two long DNA fragments are regarded as one reads cluster, the length of reads cluster (cluster length) which corresponds to a CF tends to be larger than that of reads clusters which corresponds to NFs. Therefore, CFs can be detected by selecting the SNP fragments whose cluster length are over than a threshold. Moreover, if there are some heterozygous SNPs in a overlapped region and there are enough coverage, reads in a reads cluster will show heterozygosity. Because there are several evaluation for heterozygous calls in a reads cluster, we used three measure, the total number of reads which cover minority allele (total heterozygosity), maximum of the rate of the minority allele (maximum heterozygosity), and average of the rate of the minority allele (average heterozygosity) (see the Additional file for the detailed definition). We compare the performance of CSP with that of methods based on cluster length and heterozygosity.

3.2.5 Recovering SNP fragments from CF candidates

The CSP method might regard NFs as CF candidates when the NFs differ from population haplotypes due to rare variants or spontaneous recombination. To recover such NFs from CF candidates, we use coverage data. Because CFs are produced when an aliquot happens to contain some DNA

fragments which cover the same regions, CFs would be distributed randomly. Therefore, if there are many CF candidates that cover the same regions, they would be NFs. We, therefore, recover the CF candidates which fulfill a coverage threshold condition. However, CFs might be accidentally located in a high coverage region and, therefore, we run SIH for recovered SNP fragments, calculate the chimerity based on inferred haplotypes, and remove SNP fragments whose chimerity is larger than a threshold. The detailed process and results are shown in the Additional file.

3.2.6 Mixture model for SIH

We have previously developed a mixture model for SIH (MixSIH) [43]. Our model provides a confidence score for haplotype regions, and we could extract reliable haplotype blocks using this confidence score.

Here, we give a brief explanation of MixSIH. The probability distribution of the observed SNP fragments F were modeled by parameter Θ , which represents the phase of each site. $P(F|\Theta)$ can be represented by the indicator function that represents the haplotype origin of fragments. We used the VBEM algorithm to optimize Θ with the indicator function, and inferred haplotypes from optimized Θ .

In SIH, the associations in each segment are almost random if the number of connecting fragments is not sufficient or there are many conflicting fragments. Such sites often cause switch errors and, therefore, we need a method to evaluate the reliability of the connection of the haplotypes. With the optimized parameters, we defined the connectivity at site j_0 as a ratio of the marginal log likelihoods:

$$\text{connectivity}(j_0) = \frac{P(F|\Theta)}{P(F|\Theta')},$$

where Θ' correspond to a recombinant of Θ at site j_0 . The connectivity measures the resilience of the assembly result against swapping the two haplotypes at site j_0 .

We extended the idea of connectivity to give a confidence score for a region. For the region $[j_1, j_2]$ ($j_1 < j_2$), we defined the minimum connectivity (MC) score as

$$\text{MC}(j_1, j_2) = \min_{j_1 < j \leq j_2} \text{connectivity}(j).$$

We can extract reliable assembled blocks by selecting regions with high MC values.

3.2.7 CF detection based on trio-based haplotypes

We defined the chimerity used to detect CF by using trio-based haplotypes in our previous research and use this indicator to define the true dataset.

$$\text{chimerity}(f_i) = -\ln \left(\frac{\max_{j=0,1} P^t(f_i | H_j^{(t)})}{\max_{\substack{j=0,1 \\ k \in X(f_i)}} P^t(f_{i,\leq k} | H_j^{(t)}) P^t(f_{i,>k} | H_j^{(t)})} \right)$$

$$P^t(f_i | H_j^{(t)}) = \prod_{k \in X(f_i)} P_0(f_{ik} | H_{jk}^{(t)}),$$

$$P_0(f_{ik} | H_{jk}^{(t)}) = \begin{cases} (1 - \alpha_0) & \text{for } f_{ik} = H_{jk}^{(t)} \\ \alpha_0 & \text{for } f_{ik} \neq H_{jk}^{(t)}, \end{cases}$$

where $H^{(t)} = (H_0^{(t)}, H_1^{(t)})$ is the pair of true haplotypes which are determined by trio-based haplotyping, $f_{i,\leq k}$ and $f_{i,>k}$ represent the left and right parts of fragment f_i divided at site k , and α_0 is the sequence error rate term. We define a CF as being an SNP fragment whose chimerity is over a threshold.

3.2.8 Dataset and data processing

For the sequencing data, we used the data from Kaper et al. [45] and Duitama et al. [17]. Kaper and coworkers diluted and distributed long DNA fragments into physically distinct aliquots, while Duitama and coworkers partitioned long DNA fragments into distinct low-concentration aliquots using fosmid clones. The aliquots were sequenced using next-generation sequencers. After mapping the short reads onto the reference genome, short reads formed clusters in which the reads were close to each other. Each cluster corresponded to a long DNA fragment and was supposed to originate from the same haplotypes and, therefore, the alleles observed in a cluster could be merged into a SNP fragment. In the above procedure, CFs would be produced because an aliquot might contain some long DNA fragments derived from the same region of a different chromosome, and reads with different chromosomal origins might be merged into a single SNP fragment (Figure 3.1).

Both groups conducted analyses of the HapMap trio child NA12878 from the CEU population

[38]. NA12878 had about 1.65×10^6 heterozygous sites on an autosomal chromosome and the haplotypes of about 1.36×10^6 sites were determined by a trio-based phasing method [22].

We aligned Kaper's data and Duitama's data to a human reference genome (hg18) using bowtie (version 1.0.0) and bfast (version 0.7.0), respectively. We identified read clusters that corresponded to long DNA fragments by using the targetcut function of SAMtools (version 0.1.19), and converted the clusters into SNP fragments by majority decision at the alleles of the heterozygous sites determined by the 1000 genomes project [22]. SNP fragments whose sizes were below 1 were discarded. Accordingly, 323,734 and 212,351 of SNP fragments were found for Kaper's data and Duitama's data, respectively. The average SNP fragment size in Kaper's (Duitama's) data was about 8.8 (22.6), and the average coverage of SNP fragments was 1.7 (2.9).

Next, we implemented filtering step for the reads cluster data to filter CFs by using the cluster length and heterozygous calls. This step is based on the preprocessing method proposed by previous research [17]. The reads cluster were divided into multiple reads clusters at the SNPs which show heterozygous calls. The heterozygous call was defined so that either one of the following two conditions were satisfied: (1) the number of reads which contain minority allele is larger than half the average coverage of the aliquot; (2) the number of reads which contain minority allele is larger than half of the number of reads which contain majority allele. The reads cluster which is significantly large ($>30\text{kb}$ for Kaper's data and $>45\text{kb}$ for Duitama's data) are divided into multiple reads cluster so that each cluster length is below threshold (30kb and 45kb, respectively). Accordingly, 346,417 and 436,543 of SNP fragments were found for Kaper's data and Duitama's data, respectively. The average SNP fragment size in Kaper's (Duitama's) data was about 8.0 (10.2), and the average coverage of SNP fragments was 1.7 (2.7). Hereafter, we designate this procedure as *filtering*.

In addition, we also used the original SNP fragments data of Duitama's data which was downloaded from <http://owww.molgen.mpg.de/~genetic-variation/SIH/data/>. We designate this dataset as Duitama's SNP fragments.

For statistical phasing, we used CEU population genotypes downloaded from the 1000 genomes project. To exclude the bias of related genotypes, the parents genotypes were removed. In total, 61 genotypes including NA12878 itself were used for PHASE. The influence of the number of individuals is discussed in the Additional file.

For SIH, we used ReFHap [17], FastHare [21], and DGS [19], which were implemented by Duitama [17] in addition to MixSIH.

3.2.9 Accuracy measure for CF detection

To evaluate the detection of CFs by CSP, we defined true NFs and CFs by using chimerity. A true CF was defined to be an SNP fragment which satisfies $\text{chimerity} \geq 2 \ln(\alpha_0/(1 - \alpha_0))$, and a true NF was an SNP fragment which satisfies $\text{chimerity} < 2 \ln(\alpha_0/(1 - \alpha_0))$. However, the chimerity of fragments for which haplotypes of the region could not be determined by trio-based haplotyping could not be calculated. For this reason, we removed such fragments from the evaluation. We defined sensitivity and specificity as the proportion of CFs which are detected and the proportion of the NFs which are detected by mistake, respectively.

Based on the chimerity threshold, the number of NFs and CFs in Kaper’s data (before filtering) are 283,270 and 6,864, respectively, while the number of NFs and CFs in Duitama’s data (before filtering) are 188,928 and 13,063, respectively. After filtering with cluster length and heterozygous calls, the number of NFs and CFs in Kaper’s data are 304,423 and 3,830, respectively, while the number of NFs and CFs in Duitama’s data are 384,857 and 6,381 respectively. The results of Duitama’s SNP fragments are shown in the Additional file.

The CF rate of Duitama’s data (before filtering) (6.5%) is larger than that of Kaper’s data (before filtering) (2.4%) because Duitama’s experimental approach tends to contain long DNA fragments from the same regions in a single aliquot, which results in CFs. Kaper separated long DNA fragments into 196 aliquots so that each aliquot would have a low concentration while Duitama separated fragments into 32 aliquots. Moreover, the DNA fragments in Duitama’s data are longer than those of Kaper’s data and the longer the DNA fragments are, the higher the probability that the DNA fragments overlap.

Although it is better for SIH to have fewer CFs, one cannot say unconditionally that Kaper’s data is better than Duitama’s data. This is because longer DNA fragments result in longer SNP fragments which are useful for assembling haplotypes. Moreover, from the perspective of efficiency and cost, separating long DNA fragments in more aliquots is difficult. For these reasons, each of the experimental approaches has merits and demerits.

3.2.10 Accuracy measure for SIH

To evaluate the accuracy of the partially assembled haplotype, we defined a pairwise accuracy measure in previous research [43]. Let $H^{(t)}$ be the true haplotypes, and $\hat{H} = (\hat{H}_1, \hat{H}_2, \dots, \hat{H}_B)$ be the inferred haplotypes blocks. A pair of heterozygous sites j and j' ($j < j'$) was defined as consistent if $(\hat{H}_{i,j}, \hat{H}_{i,j'}) = (H_{0,j}^{(t)}, H_{0,j'}^{(t)})$ or $(H_{1,j}^{(t)}, H_{1,j'}^{(t)})$, and inconsistent otherwise, where $\hat{H}_{i,j}$ represents the allele of the j th locus belonging to the i th haplotype segment. For each haplotype block, we count the consistent and inconsistent pairs. The total numbers of consistent and inconsistent pairs over all the haplotype blocks are denoted by CP and IP, respectively. We defined *precision* by $CP/(CP + IP)$. The detailed explanation is shown in previous research [43].

We also used other two accuracy measures, switch error rate and QAN50 [17]. The switch error rate is defined as the frequency of switch errors which are inconsistency between inferred haplotypes and true haplotypes. The QAN50 is remodeled from N50 so that it takes consistency between inferred haplotypes and true haplotypes into account. In short, prediction is divided into smaller haplotype blocks that do not contain any switch errors, and QAN50 is N50 of divided inferred haplotypes with some adjustments.

3.3 Results and discussion

3.3.1 Detection of chimeric fragments

We compared the CSP density distributions for NFs and CFs of the data before filtering (Figure 3.2). The CSP of CFs shows a tendency to be larger than that of NFs. This result suggests that the CFs are regarded as artificial recombinant haplotypes and hence differ from the biological haplotypes which exist in the population. There are peaks in the CSP density distributions at 4.6 and 9.2. These peaks correspond to SNP fragments which are inconsistent with statistically phased haplotypes and are consistent when the SNP fragment changes the derivation to another haplotype. The CSP is around 4.6 ($\approx -\ln(\alpha/(1-\alpha))$) when a SNP fragment changes the haplotype origin at the first site from the end, and the CSP is around 9.2 ($\approx -2\ln(\alpha/(1-\alpha))$) when a SNP fragment changes the haplotype origin at the second site from the end. For $W=5$, the CSP of CFs which are inconsistent with statistically phased haplotypes is expected to be around 9.2 because in that case the SNP fragment is recombinant at the second site from the end in the sliding window calculation.

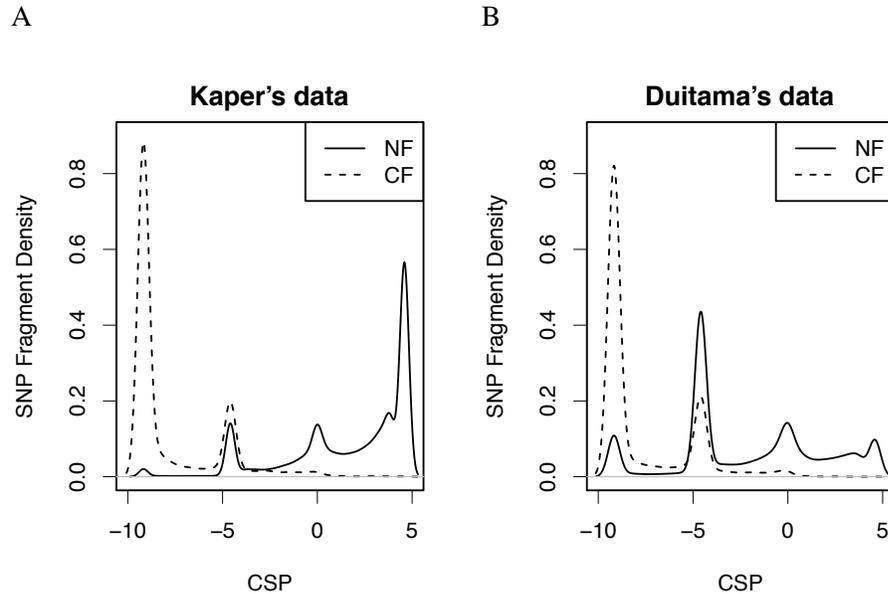


Figure 3.2. Comparison of CSP density distributions for NFs and CFs. (A) and (B) are the distributions of Kaper's data and Duitama's data, respectively.

Actually, 74.1% (71.9%) of CFs in Kaper's (Duitama's) data are between CSP=7 and CSP=12, and 1.5% (9.7%) of NF are within the same bounds. The peak at 4.6 is likely to be caused by sequencing and statistical phasing errors.

Figure 3.3 shows the ROC curves of CSP, cluster length, and total heterozygosity for each dataset before filtering. The ROC curves of maximum heterozygosity and average heterozygosity are inferior to that of total heterozygosity, and are shown in the Additional file. The area under the curve (AUC) of CSP for Kaper's data is 0.97 and the AUC for Duitama's data is 0.88. These values are higher than those of cluster length (0.71 for Kaper's data and 0.85 for Duitama's data) and total heterozygosity (0.80 for Kaper's data and 0.82 for Duitama's data). The AUC values of cluster length are lower than that of CSP, especially in the case of Kaper's data, and this is because the cluster length of NFs and CFs overlap significantly (see the Additional File for the distribution of cluster length of NFs and CFs). The AUC values of total heterozygosity are lower than that of CSP and this is because there are considerable CFs which do not show heterozygosity due to the lack of coverage and absence of heterozygous SNPs in overlapped regions. Moreover, sequencing error will disturb to distinguish NFs and CFs because sequencing errors in NFs will bring heterozygous calls and such NFs might be regarded as CFs by mistake. These results show the high performance of the

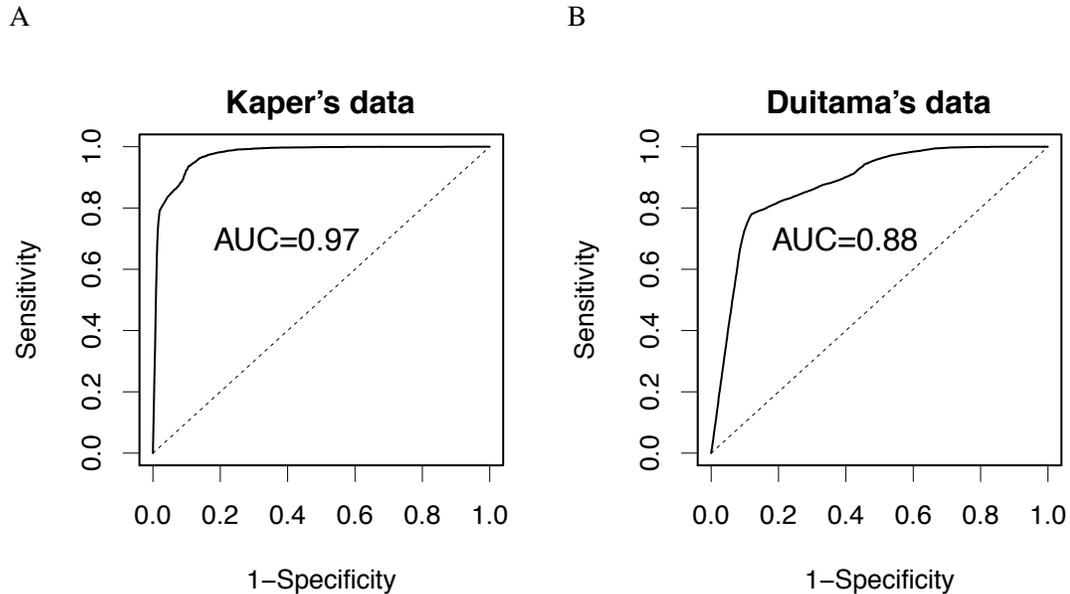


Figure 3.3. The ROC curves of CSP, cluster length, and total heterozygosity for classification of CFs and NFs. The ROC curves are obtained by changing the threshold of CSP, cluster length, total heterozygosity, respectively. There is a region that the data point of the ROC curve of total heterogeneity for Kaper's data is absent, and hence, the ROC curve is supplemented (shown as gray line). (A) and (B) correspond to Kaper's data and Duitama's data, respectively.

detection of CFs using CSP, regardless of the experimental conditions. The difference between the AUC values of CSP for each dataset might be caused by the error rate in SNP fragments; The SNP fragment error rate of Duitama's data is 4.0% and that of Kaper's data is 1.2% (see the Additional file for the SNP fragment error rate calculation).

Figure 3.4 shows the Venn diagrams of CFs detected by CSP, cluster length, and total heterozygosity for each dataset. The threshold of each measure was set so that (1-specificity) was under 0.1. In Kaper's data, the number of CFs which were detected with CSP was largest, and about 94% of CFs which were detected with either cluster length or total heterozygosity were also detected with CSP. In Duitama's data the number of CFs which were detected with CSP was slightly lower than that of CFs detected with cluster length, but about 14% of CFs detected with CSP were detected with neither cluster length nor total heterozygosity. These results also show that CSP is an effective indicator for detecting CFs which are detected with neither cluster length nor heterozygosity. Since there are significant number of CFs which are detected only with cluster length and heterozygosity calls, we compare the SIH accuracies of the SNP fragments that are filtered with cluster length and heterozygous calls with those of the SNP fragments that are further filtered with CSP, and examined

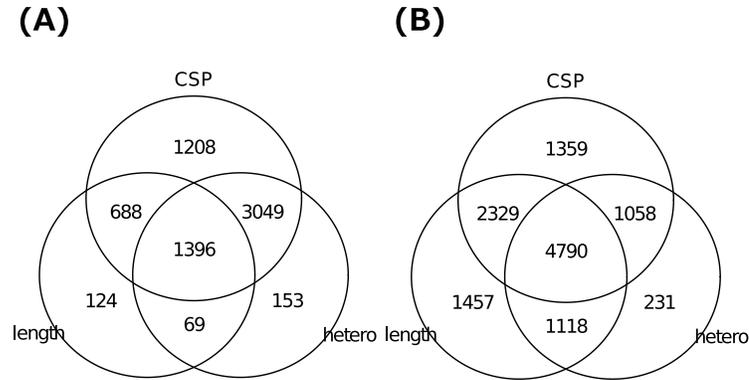


Figure 3.4. The Venn diagrams of CFs detected by CSP, length, and total heterozygosity. The number in each cell is the number of CFs in the corresponding category. The threshold for CF detection of each measure was set so that the 1-specificity was under 0.1. (A) and (B) correspond to Kaper's data and Duitama's data, respectively.

the usefulness of CSP in SIH in the following section.

3.3.2 SIH accuracy after removing suspicious CFs by using CSP

We defined a CF candidate as a SNP fragment whose CSP was larger than 7, and removed these from SNP fragments. We hereafter represent the SNP fragments filtered with cluster length and heterozygous calls as "filtered", and the SNP fragments further filtered with CSP as "filtered+CSP".

The CSP threshold was determined so that many CFs were removed while avoiding a high false-positive rate; many CFs had a CSP of around 9.2 and there were many NFs with around $\text{CSP} = 4.6$ (Figure 3.2). With this procedure, 1.6% (5,375/346,417) of Kaper's data and 3.8% (16,715/436,543) of Duitama's data were removed. The removed fragment rate for Duitama's data was higher than that for Kaper's data because Duitama's data would contain more CFs because of the experimental approach (see Section 2.8 for a detailed explanation).

Figure 3.11 shows the accuracies of MixSIH, ReFHap, FastHare, and DGS for each dataset: filtered with cluster length and heterozygous calls (filtered); further filtering with CSP (filtered+CSP). The precision of MixSIH increased from about 0.972 to 0.985 at $(\text{CP}+\text{IP}) = 1.5 \times 10^7$ for Kaper's data, and increased from about 0.950 to 0.966 at $(\text{CP}+\text{IP}) = 5.0 \times 10^7$ for Duitama's data. The precision of other algorithm increased likewise. In addition, the precision for Duitama's SNP fragments also increased after removing CFs candidates with CSP (shown in the Additional file). Thus,

CSP increases SIH accuracy by removing CF candidates which would have a serious influence.

In addition, (CP+IP) for Duitama's data is larger than that for Kaper's data because the SNP fragment size and coverage are larger. The precision of Kaper's data is higher because it contains fewer CFs and the SNP error rate is lower; the decrease of (CP+IP) is lower for the same reason.

Table 3.1 and Table 3.2 show the switch error rate and the QAN50 of each algorithm for each dataset, respectively. In these analyses, MC of MixSIH were set to 10. The switch error rate improved after removing suspicious CFs in all conditions. This result is consistent with the result based on pairwise accuracy measure and shows the usefulness of removing CFs with CSP. Switch error rates of MixSIH were lowest in all conditions and this suggests that MixSIH succeeds to extract reliable haplotype regions with MC values.

The QAN50 also improved after removing suspicious CFs in all conditions excluding the QAN50 of MixSIH at MC=10. The QAN50 of MixSIH at MC=10 were lowest in those of other algorithm and did not improve after removing CF candidates. This is because QAN50 does not contain the penalty of connecting wrong haplotypes and will improve just by connecting two haplotypes blocks randomly with probability 0.5, and is inappropriate to evaluate extracting reliable haplotypes.

From these results, we concluded that CSP is an efficient indicator to improve SIH accuracy by removing suspicious CFs.

Table 3.1. The switch error rate (%) of each SIH algorithm for data (filtered) and data (filtered+CSP). MC of MixSIH is set to 10. (A) and (B) correspond to Kaper's data and Duitama's data, respectively.

		MixSIH	ReFHap	FastHare	DGS
(A)	filtered	0.67	1.54	1.59	1.73
	filtered+CSP	0.52	1.22	1.28	1.38
(B)	filtered	2.75	3.22	3.28	3.47
	filtered+CSP	2.13	2.77	2.84	3.03

3.3.3 Assembled haplotype block size

We examined the size distribution of assembled haplotype blocks. The haplotypes were inferred from each dataset in which the fragments with CSP larger than 7 were removed. Table 3.3 shows the number of haplotype blocks that contain the certain range of the number of phased SNPs for each

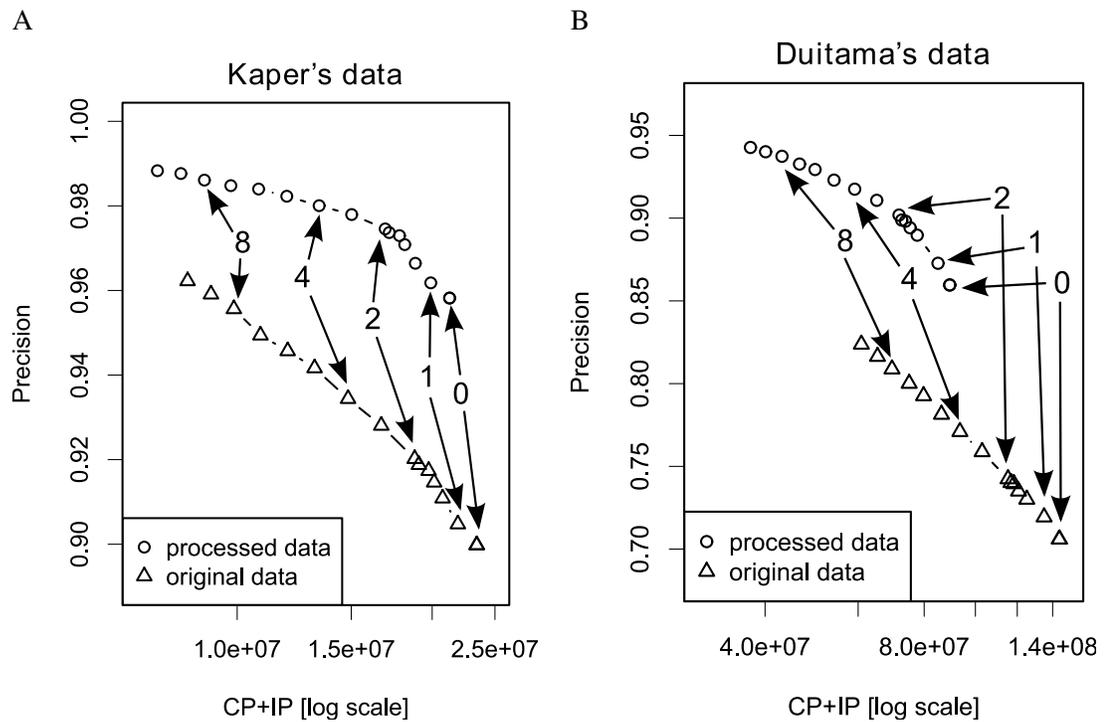


Figure 3.5. Precision curves based on consistent pair counts. The x -axis represents the number of predicted pairs on a log scale. MC of MixSIH was changed from 0 to 10. The accuracies of the data filtered with cluster length and heterozygous calls (filtered) (filled point symbols) and the further filtered data, in which fragments with CSP > 7 are removed (filtered+CSP) (empty point symbols), are shown for Kaper's data (A) and Duitama's data (B); \circ MixSIH; \triangle ReFHap; \square FastHap; \diamond DGS.

Table 3.2. QAN50 (kb) of each SIH algorithm data (filtered) and data (filtered+CSP), in which fragment with CSP > 7 are removed. MC of MixSIH is set to 10. (A) and (B) correspond to Kaper’s data and Duitama’s data, respectively.

		MixSIH	ReFHap	FastHare	DGS
(A)	filtered	16.6	27.3	27.1	26.8
	filtered+CSP	16.6	27.5	27.4	27.2
(B)	filtered	32.7	69.2	68.4	67.7
	filtered+CSP	32.5	70.4	70.0	68.6

dataset. For comparison, the number of SNP fragments that cover the certain range of the number of SNPs are also shown.

The averages of haplotype block size are about 19.2 and 42.6 for Kaper’s data and Duitama’s data, and they are larger than the averages of SNP fragment size (8.0 and 10.2, respectively). Moreover, the number of haplotype blocks that contain more than 100 SNPs are larger than the number of SNP fragments for both dataset. These results suggest that MixSIH succeeds to assemble haplotypes from SNP fragments. 1.8% and 12.9% of haplotype blocks in Kaper’s data and Duitama’s data contain more than 100 phased SNPs, and the ratio of phased SNPs in such long haplotype blocks to total SNPs are about 13.1% and 53.8%, respectively. This result suggests that SIH is able to determine long haplotypes which are not determined by statistical phasing.

In addition, the haplotype blocks in Duitama’s data tend to be longer than those of Kaper’s data because the SNP fragment size and coverage are larger. This result shows that SIH will be able to infer longer haplotypes in accordance with improvements of sequencing technologies.

Table 3.3. The number of the SNP fragments which cover the certain range of the numbered of SNPs (before SIH) and the number of haplotype blocks which contain the certain range of the number of phased SNPs (after SIH) for Kaper’s data (A) and Duitama’s data (B) (Note that a SNP can be contained in multiple SNP fragments and the halotype blocks do not overlap each other). The first row defines the range of the number of SNPs.

		-10	11-20	21-50	51-100	101-200	201-
(A)	before SIH	261,537	65,429	18,894	540	16	1
	after SIH	28,631	10,503	11,186	3,998	923	72
(B)	before SIH	291,495	92,104	49,092	3,652	192	8
	after SIH	15,273	4,037	6,039	4,882	3,267	1,202

3.3.4 Comparison of MixSIH and PHASE

The strong and weak points of SIH and statistical phasing will differ because they use different information for inferring haplotypes. For example, SIH cannot infer haplotype regions which lack SNP fragments because of sequencing and mapping difficulties. Statistical phasing is weak in determining haplotype regions where linkage disequilibrium values are high and there are multiple haplotypes in population. To investigate these differences, we compared the reliabilities of MixSIH and PHASE.

We selected 10,000 regions in chromosome 1 randomly so that each region had five SNP sites and the haplotypes of the regions were determined by trio-based haplotyping. We used Kaper's data (filtered) and Duitama's data (filtered) for SIH in this section. Figure 3.6 shows the MC value and the maximum probability of the PHASE for each region. The x -axis is $\ln(1.001 - \max P)$, where $\max P$ is the maximum haplotypes probability of PHASE for the region. We used 1.001 to deal with the case that $\max P = 1.0$. The vertical dotted line corresponds to the maximum probability above which the precision of PHASE is over 0.9, and the horizontal dotted line corresponds to the MC value above which precision of MixSIH is over 0.9 (see the Additional file for the calculation of precision).

Table 3.4 shows the number of regions for each division created by the previously noted dotted lines. In Duitama's data, the rates in upper left division and lower right division are 8.4% and 22.2%, respectively. This result suggests that there are chromosomal regions for which SIH successfully infers the haplotypes and statistical phasing fails, and vice versa. The rate in the lower right division of Duitama's data decreases from 22.2% to 14.1% when we remove the regions which contain sites that lack SNP fragments. This result suggests that many regions where SIH does not work are the result of a lack of SNP fragments.

Moreover, the rate in the upper divisions for Kaper's data and Duitama's data are 39.3% and 70.9%, respectively. The rate for Duitama's data is larger than that for Kaper's data because SNP fragment size and coverage are larger. This result suggests that SIH results will be improved just by getting larger and more SNP fragments.

In summary, there are regions where either SIH or statistical phasing can infer the haplotypes for these data. In the case of SIH, a shortage of data is likely to be the main reason for inference failure. For this reason, the performance of SIH will increase with advances in sequencing techniques.

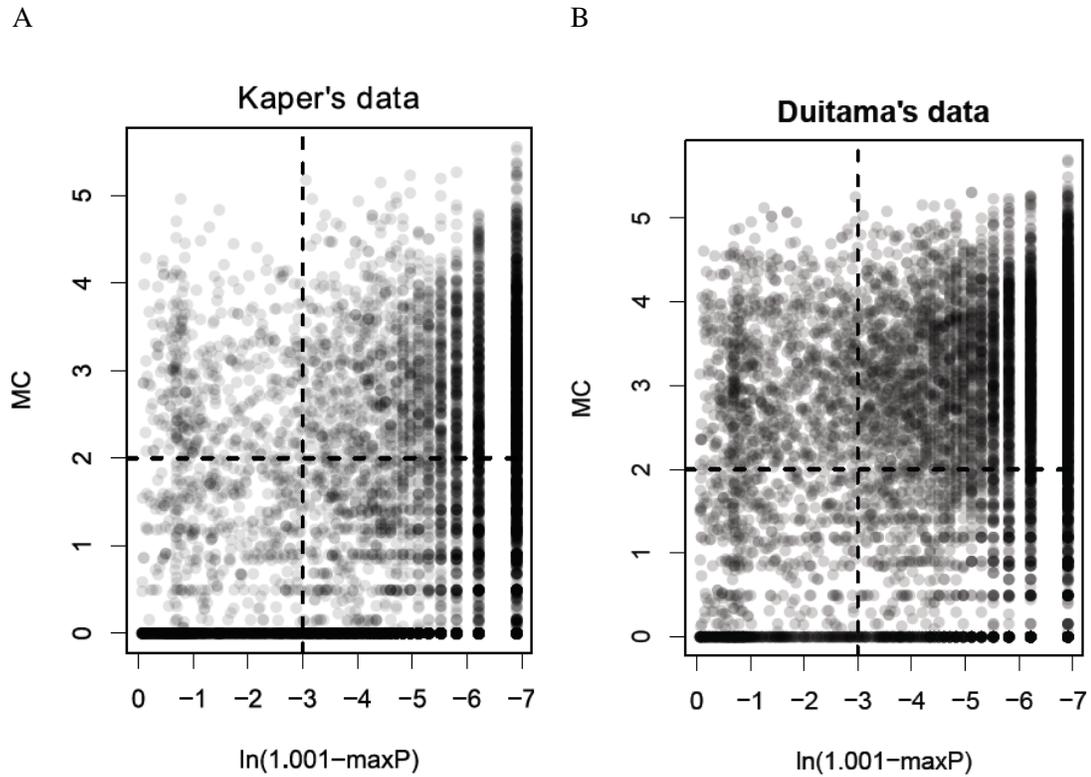


Figure 3.6. Comparison of MC scores and maximum PHASE probabilities (A) and (B) correspond to Kaper's data and Duitama's data, respectively. The x -axis represents $\ln(1.001 - \max P)$, where $\max P$ is the maximum PHASE probability and we use 1.001 to deal with $\max P = 1.0$. The y -axis represents the MC score of MixSIH. Data are randomly selected 1000 times from chromosome 1. The vertical dotted line corresponds to the maximum PHASE probability above which the precision of PHASE is over 0.9, and the horizontal dotted line corresponds to the MC value above which precision of MixSIH is over 0.9.

Table 3.4. The numbers of regions for each of the areas which are defined by the precision of MixSIH and PHASE: (A) Kaper’s data and (B) Duitama’s data. The rows and columns represent the accuracy of MixSIH and PHASE, respectively. The numbers in parentheses are the numbers of regions remaining after regions which contain sites that lack SNP fragments have been removed.

A

	PHASE < 0.9	PHASE \geq 0.9
MixSIH \geq 0.9	433 (366)	3,499 (2,792)
MixSIH < 0.9	1,096 (251)	4,972 (988)

B

	PHASE < 0.9	PHASE \geq 0.9
MixSIH \geq 0.9	842 (749)	6,250 (5,337)
MixSIH < 0.9	687 (390)	2,221 (1,061)

3.4 Conclusions

In this paper, we have developed a general method to detect chimeric fragments (CFs) on the assumption that CFs correspond to an artificially recombinant haplotype and differ from the biological haplotypes in the population. Based on this assumption, we developed natural fragment (NF) and CF probabilities of a fragment which use the result of statistical phasing. The NF probability calculates the consistency between a fragment and statistically inferred haplotypes. The CF probability also calculates the consistency, but it assumes that left and right parts of the fragment are derived from different haplotypes in a haplotype pair. With these probabilities, we developed an indicator CSP which evaluates the degree of chimerity by calculating the logarithmic difference.

We applied CSP to two sequencing datasets, Kaper’s data and Duitama’s data [17, 45]. The CSP of CFs tends to be lower than that of NFs. Moreover, there are a lot of CFs at around possible largest value. These results support the propriety of our model. The high AUC values of CSP (0.97 for Kaper’s data and 0.88 for Duitama’s data) also shows that CSP is a highly efficient measure to detect CFs. The AUC values of CSP are higher than that of measures based on cluster length and heterozygosity. Moreover, there are significant number of CFs which are only detected with CSP. These results suggests the usefulness of CSP for detecting CFs.

We then compared the accuracies of MixSIH before and after removing the chimeric fragment candidates detected using CSP. The accuracies of MixSIH increased significantly after removing

CFs. From these results, we conclude that CSP is a useful method for detecting CFs and improving SIH accuracy, regardless of the type of dilution-based sequencing.

In addition, we analyzed the results of MixSIH. The assembled haplotype blocks contain a lot of long haplotype blocks and this supports the capability of SIH that SIH can determine long haplotypes. We also compared the performance of MixSIH and statistical phasing method (PHASE). At the moment, the number of correctly inferred regions of PHASE is larger than that of MixSIH. However, lack of SNP fragments is the main reason for failure of SIH and, therefore, the importance of SIH and our method will increase in accordance with the advance of sequencing technologies.

In the future the amount of dilution-based sequencing data will increase, and our approach will be an important strategy not only for SIH but also for many other types of analysis, such as the detection of novel recombinant events.

3.5 Supplementary text

3.5.1 Cluster length and heterozygous calls

3.5.1.1 Evaluation of heterozygous calls in a reads cluster

To detect CF by using the heterozygous calls in a reads cluster, we defined three measurements to evaluate the heterozygosity of SNP fragment f_i .

Firstly, we defined the total number of reads which cover minority allele (total heterozygosity) as follows:

$$\sum_{j \in X(f_i)} \min(n(r_{i,j} = 0), n(r_{i,j} = 1)) ,$$

where $n(r_{i,j} = 0)$ and $n(r_{i,j} = 1)$ are the number of reads, which are contained in a reads cluster which corresponds to a SNP fragment f_i and whose base at j -th locus are major allele and minor allele, respectively.

Secondary, we defined maximum of the rate of the minority allele (maximum heterozygosity) as follows:

$$\max_{j \in X(f_i)} \frac{\min(n(r_{i,j} = 0), n(r_{i,j} = 1))}{n(r_{i,j} = 0) + n(r_{i,j} = 1)} .$$

Thirdly, we defined average of the rate of the minority allele (average heterozygosity) as follows:

$$\frac{1}{|X(f_i)|} \sum_{j \in X(f_i)} \frac{\min(n(r_{i,j} = 0), n(r_{i,j} = 1))}{n(r_{i,j} = 0) + n(r_{i,j} = 1)} .$$

With these measurements, we detected CFs candidates by selection the fragments whose values are larger than a threshold.

3.5.1.2 ROC curves of heterozygosity evaluation

Figure 3.7 shows the ROC curves of total heterozygosity, maximum heterozygosity, and average heterozygosity. In Kaper's data, ROC curves stops at sensitivity is around 0.7. This is because there are many CFs which do not show heterozygous, and this could be caused when the coverage

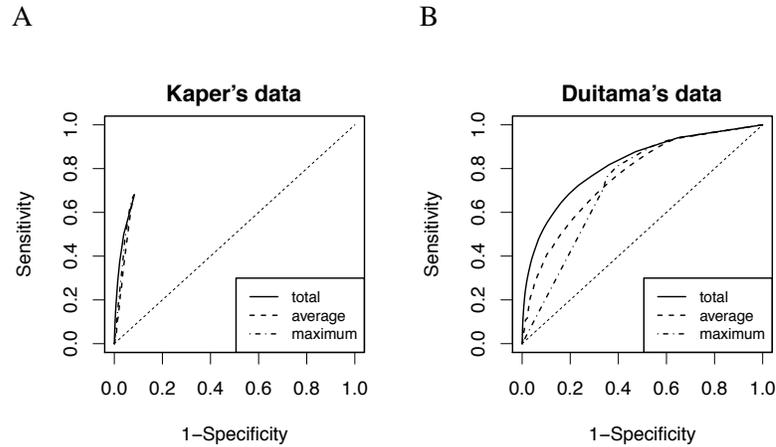


Figure 3.7. The ROC curves of total heterozygosity, average heterozygosity, and maximum heterozygosity for classification of CFs and NFs. A and B correspond to Kaper's data and Duitama's data, respectively.

is low and only one origin of reads which derived from the same haplotype exist. In Duitama's data, the ROC curve of maximum heterozygosity and averaged heterozygosity are below that of total heterozygosity. This is because maximum and average heterozygosity overestimate the effect of sequencing error. Therefore, we concluded that total heterozygosity is appropriate to evaluate heterozygosity in a reads cluster.

3.5.1.3 Distribution of length of reads clusters

Figure 3.8 shows the distribution of the length of reads clusters for each dataset. The length of reads cluster which correspond to CFs tend to be larger because reads with different long DNA fragments origins are merged into one reads cluster. Although the cluster length of CFs tend to be larger than that of NFs, there are considerable overlapping between NFs and CFs, especially in the Kaper's data.

3.5.2 Effects of changing various parameters

3.5.2.1 Impact of changing sliding window width on accuracy and running time

PHASE takes time to deal with a long SNP fragment because the number of possible haplotypes and their combinations increases exponentially. We defined a sliding window calculation to reduce the running time for long fragments. Because the sliding window width would affect the result, we examined the impact of sliding window width (W) on accuracy and running time. We used SNP

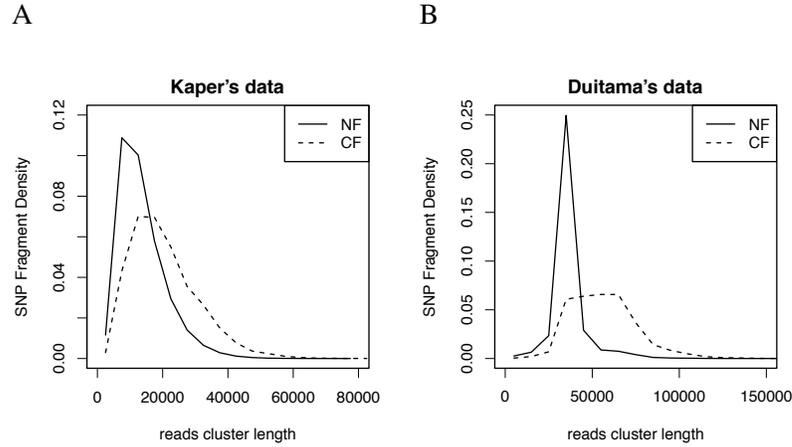


Figure 3.8. The distribution of cluster length. The x -axis represents the length of reads cluster and the y -axis represents the number of SNP fragments which correspond to the each reads cluster.

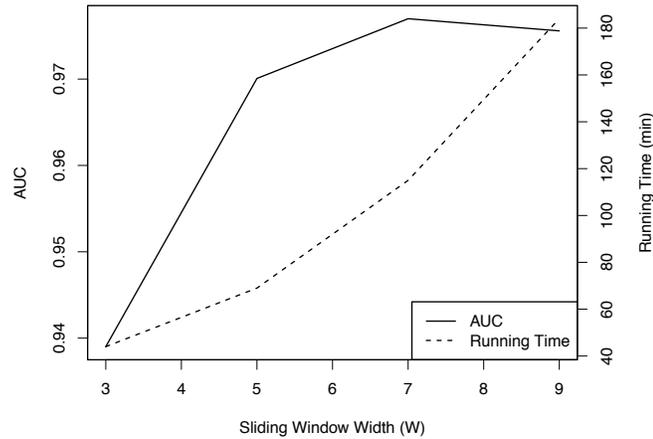


Figure 3.9. AUC values and running times for various values of W .

fragments of chromosome 1 from Kaper's data for the AUC calculation, and used 100 randomly generated SNP fragments of size 30 for the running time calculation.

Figure 3.9 shows the AUC values and running times for $W=3, 5, 7, 9$. AUC increases roughly in line with the increase of W . This is because the difference between haplotypes becomes clearer when we consider more SNPs. However, difference between AUC values for $W=3$ and $W=5$ is larger than that for $W=5$ and $W=7$, which suggests that AUC would roughly saturate for low W . Running time also increases with increasing W . This is because the possible haplotypes and combinations of haplotypes increase exponentially as W increases. In view of these accuracy and running time results, we use $W=5$ as the default setting.

3.5.2.2 Effect of error rate α

We included an error term in CSP to represent sequencing and PHASE errors. To examine the effect of the error rate parameter α , we calculated AUC values for various values of α . We used chromosome 1 from Kaper's data and Duitama's data for the AUC calculation. Table 3.5 shows the AUC values for each α . The AUC for $\alpha = 0.0$ is lowest because CSP with $\alpha = 0.0$ cannot deal with the inconsistency between inferred haplotypes and the context of a fragment which is caused by the sequencing and PHASE errors. The AUC values for $0.001 \leq \alpha \leq 0.1$ are almost equal. These results suggest that including α in CSP is important but the absolute value of α is unimportant. Based on these results, we use $\alpha=0.01$ as the default value.

Table 3.5. AUC values for each α

	Kaper's data	Duitama's data
$\alpha=0.0$	0.724	0.683
$\alpha=0.001$	0.970	0.879
$\alpha=0.01$	0.970	0.878
$\alpha=0.1$	0.969	0.882

3.5.2.3 Effect of the number of individual genotypes

The accuracy of PHASE should increase with the number of individual genotypes. To examine the effect of changing the number of individual genotypes, we calculated the AUC of CF detection using chromosome 1 from Kaper's data and selecting $N=5, 10, 20, 40, 60$ individuals randomly from 60 unrelated individuals in the CEU population. We ran PHASE for randomly selected genotypes and the NA12878 genotype, and calculated AUC using the result of PHASE (Figure 3.10). AUC increases with the number of individuals. However, the rate of increase slows when the number of individuals increases. This suggests that detecting CFs which are located in multiple haplotype regions or contain sequencing errors, is difficult regardless of the number of individuals.

3.5.3 Recovering SNP fragments from CF candidates

CSP might regard NFs as CF candidates when NFs differ from population haplotypes because of rare variants or spontaneous recombination. As CFs are generated because an aliquot occasionally

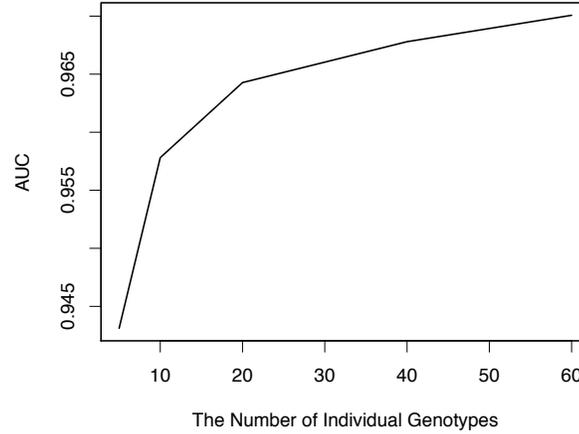


Figure 3.10. AUC values for various numbers of individuals.

contains multiple DNA fragments which cover the same region, CFs would be distributed randomly. Therefore, if there are many CF candidates which cover the same region, they would be misidentified NFs. Because some CFs remain with only the threshold coverage, we removed fragments using a SIH-based measure. The detailed process is as follows:

- 1) Calculate the coverage of CF candidates for each heterozygous site.
- 2) Exclude sites whose coverage is lower than 3 and recover the SNP fragments which correspond to the remaining sites (P1).
- 3) Run MixSIH for recovered SNP fragments.
- 4) Calculate the chimerity-like measure ‘SIH-chimerity’

$$\text{SIH-chimerity}(f) = -\ln \left(\frac{\max_{i=0,1} P^t(f|\hat{H}_i)}{\max_{i=0,1, j \in X(f)} P^t(f_{\leq j}|\hat{H}_i) P^t(f_{> j}|\hat{H}_i)} \right),$$

where $\hat{H} = (\hat{H}_0, \hat{H}_1)$ is the pair of haplotypes which are inferred by MixSIH.

- 5) Remove the fragments which satisfy $\text{SIH-chimerity} \geq 2 \ln(\alpha_0/(1 - \alpha_0))$ (P2).

Table 3.6 shows the numbers of all fragments, NF, and CF before and after recovery. The numbers of all fragments are larger than sums of NFs and CFs because trio-based haplotyping is partial and the chimerity of fragments which cover unphased regions cannot be calculated. The rates of CF for Kaper’s data are 61.2%, 2.5%, and 2.1%, and the rates of NF for Duitama’s data

Table 3.6. The numbers of all fragments, NFs, and CFs after performing each process on Kaper’s data (A) and Duitama’s data (B).

A				B			
	Before	P1	P2		Before	P1	P2
All	5,375	290	288	All	16,715	4,151	4,045
NF	1,924	236	235	NF	10,699	2,759	2,692
CF	3,030	6	5	CF	4,875	897	858

are 31.3%, 24.5%, and 24.2%. For both of datasets, the rates of CF decrease and we successfully recover NFs from CF candidates with high precision. The recovered fragments rates are 4.4% ($235/5,375$) and 16.1% ($2,692/16,715$) for Kaper’s data and Duitama’s data, respectively. The rate of recovered fragments for Duitama’s data is larger than that for Kaper’s data because the coverage of Duitama’s data is higher than that of Kaper’s data. High coverage might result in a larger CF rate in recovered fragments for Duitama’s data.

In summary, NFs could be recovered from the CFs candidates by using the coverage information and SIH based chimerity. The coverage threshold should be determined according to the purpose of the analysis because there is a tradeoff between sensitivity and specificity.

3.5.4 Calculation of SNP fragment error rate

The SNP fragment error rate was calculated by comparison with the results of trio-based haplotyping. Because we were interested in the SNP fragment errors which were caused by sequencing and mapping errors, and CFs might disrupt the error rate calculation, we used only SNP fragments whose chimerity was under $2 \ln(\alpha_0/(1 - \alpha_0))$ for the calculation. The SNP fragment error rate is

$$\frac{\sum_{i=1}^N \min_{j=0,1} \left(\sum_{k \in X'(f_i)} I(f_{ik} \neq H_{jk}^{(t)}) \right)}{\sum_{i=1}^N |X'(f_i)|},$$

where $X'(f_i)$ is the set of sites which are covered by f_i and whose phases are determined by trio-based haplotyping, $|X'(f_i)|$ is the number of sites in $X'(f_i)$, and $I(f_{ik} \neq H_{jk}^{(t)})$ is 1 when f_{ik} is inconsistent with reference haplotype $H_{jk}^{(t)}$ and 0 otherwise.

Table 3.7. The number of NFs and CFs of Duitama’s SNP fragments (A) and our processed Duitama’s data (B).

	NF	CF
(A)	245,772	8,247
(B)	384,857	6,381

3.5.5 Comparison for Duitama’s SNP fragments

3.5.5.1 The number of NFs and CFs of Duitama’s SNP fragments

The number of NFs and CFs of Duitama’s SNP fragments are 245,772 and 8,247, respectively, while the number of NFs and CFs of our processed Duitama’s data are 384,857 and 6,381, respectively (Table 3.7). The number of NFs of Duitama’s SNP fragments is lower than that of our data. This difference could be caused by the mapping tools, the reads cluster detection algorithm, and the filtering step. We used bfast for mapping SOLiD reads instead of BioScope which was used by Duitama et al. because the original bfast paper suggested that bfast has robustness against the sequence variants, and BioScope was not easily available. We used the targetcut function of the SAMtools which was used by Kaper et al. for reads cluster detection because the source code of cluster detection used by Duitama et al. was not open.

Concerning that the number of CFs of our data is lower than that of Duitama’s SNP fragments, our processing method turns out to be more strict processing method. Some reads clusters will be divided into smaller reads clusters with the strict processing method, and this results in the increase of the number of NFs.

The SIH accuracy was shown to decrease with the presence of CFs. Therefore, our processing method which generates less CFs will be better than Duitama’s processing method in terms of SIH accuracy.

3.5.5.2 SIH accuracy of Duitama’s SNP fragments after removing suspicious CFs by using CSP

The SNP fragments data, in which long reads cluster and heterozygous calls are already filtered, is open by Duitama’s group and we examined the pairwise accuracies of original Duitama’s SNP fragments and processed Duitama’s SNP fragments, in which fragments with $CSP > 7$ are removed

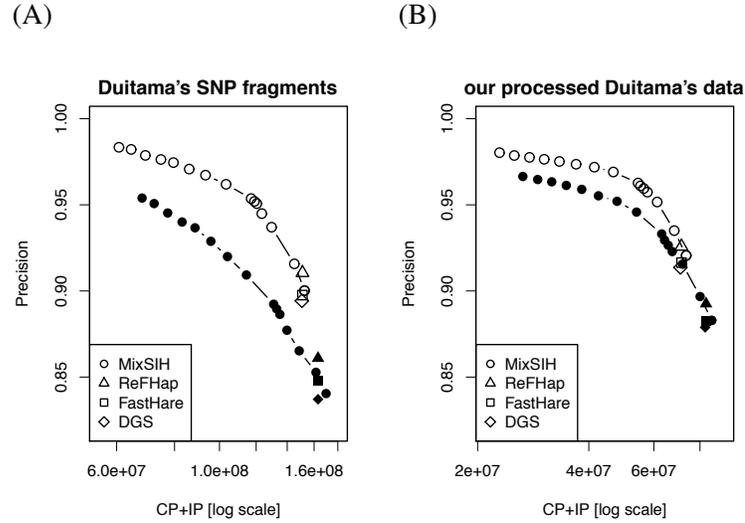


Figure 3.11. Precision curves based on consistent pair counts for Duitama's SNP fragments (A) and our processed Duitama's data (B). The x -axis represents the number of predicted pairs on a log scale. MC of MixSIH was changed from 0 to 10. The accuracies of the original data (filled point symbols) and the processed data (empty point symbols), in which fragments with $CSP > 7$ are removed, are shown: \circ MixSIH; \triangle ReFHap; \square FastHare; \diamond DGS.

(Figure 3.11 (A)). For comparison, the pairwise accuracies our processed Duitama's data that are already shown in the main text are shown again (Figure 3.11 (B)). With the CSP filtering procedure, 4.6% (12,364/271,184) of Duitama's SNP fragments were removed. The precision of MixSIH increased from 0.875 to 0.925 at $(CP+IP) = 1.4 \times 10^8$. The precision of other algorithm increased likewise. Thus, CSP is an efficient measure to detect the CFs which are undetected with cluster length and heterozygous calls, and useful for improving SIH accuracy.

In addition, $(CP+IP)$ for Duitama's SNP fragments is larger than that for our processed Duitama's data, while the precision of each algorithm for Duitama's SNP fragments are lower than those for our data. These differences are caused by the difference of the processing methods (as discussed in the above section). With the strict processing method, the length of SNP fragments become smaller owing to the division of the reads cluster, and hence the length of assembled haplotypes is smaller. On the other hand, the strict processing method generates less CFs and the precision of assembled haplotypes increase.

3.5.6 Precision of MixSIH and PHASE

The precision of MixSIH was calculated as follows.

- 1) Select 10,000 regions in chromosome 1 randomly such that each region has five SNP sites and the haplotypes of the regions are determined by trio-based haplotyping.
- 2) Calculate MC values for each region.
- 3) Calculate the precision for MC value, which is defined by

$$CP_{mc}/(CP_{mc} + IP_{mc}) ,$$

where mc is the target MC value, and CP_{mc} and IP_{mc} are the number of consistent pairs and inconsistent pairs in the regions for which MC value satisfy $mc \leq MC < mc + 0.5$.

Figure 3.12(A) shows the precision for each dataset. In our evaluation, MixSIH precisions are over 0.90 for $MC \geq 1.5$.

The precision for each $\ln(1.001 - \max P)$, where $\max P$ is the maximum PHASE probability, was calculated as follows.

- 1) Run PHASE for the 10,000 selected regions.
- 2) Examine the best haplotypes and its probability ($\max P$) for each region.
- 3) Calculate the precision for $\ln(1.001 - \max P)$, which is defined by

$$CP_p/(CP_p + IP_p) ,$$

where p is the target $\ln(1.001 - \max P)$, and CP_p and IP_p are the number of consistent pairs and inconsistent pairs in the regions for which $\max P$ satisfy $p - 0.5 < \ln(1.001 - \max P) \leq p$.

In our evaluation, PHASE precision is more than 0.90 for $\ln(1.001 - \max P) \leq -2.5$.

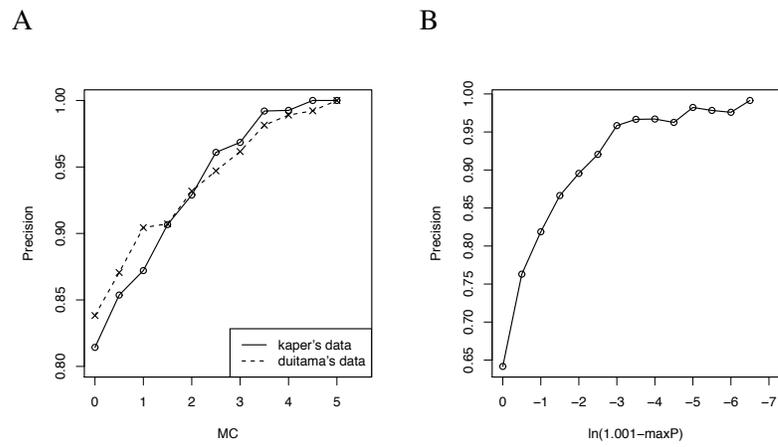


Figure 3.12. The consistent pair precision of (A) MC value and (B) maximum PHASE probability (maxP) for 5 SNPs regions.

Chapter 4

SCOUP: a probabilistic model based on the Ornstein–Uhlenbeck process to analyze single-cell expression data during differentiation

4.1 Introduction

Conventional analyses of bulk cells, such as bulk transcriptome analyses, are based on the averaged data of an ensemble of cells and cannot reveal the states of individual cells. Therefore, such analyses cannot distinguish cell types due to the effect of averaging across all cells in a sample, unless each cell lineage is divided in advance by using prior knowledge, such as marker genes. Additionally, bulk transcriptome during differentiation is usually the ensemble of the cells of different degrees of differentiation and information regarding changes in cellular state is smeared. Accordingly, the accurate investigation for gene expression dynamics and regulatory relationships among genes during differentiation are difficult.

With the advent of single-cell technologies, such as single-cell RNA-seq, quantification of the comprehensive states of individual cells is possible [49]. Using single-cell technologies, investigations of cellular states and its transition processes, such as the classification and identification of cell

types [50–52], reconstruction of cell lineages [53, 54], and embryonic development [55, 56], have made remarkable progress. Single-cell data is also useful for elucidating cell fate decision mechanisms of multi-lineage differentiation from a single progenitor cell type [57, 58]. Thus, single-cell technologies have the power to shed light on differentiation in particular [59, 60].

To fully analyze the single-cell expression data during differentiation, novel computational methods are necessary [59, 61]. First, ordering of the cells based on expression data so that the order represents the trajectory of differentiation is necessary to investigate gene expression dynamics and regulatory mechanisms. Although experimental time can be used for ordering cells, even cells derived from the same time-point can exhibit different degrees of differentiation [62]. Moreover, computational ordering method is often useful to reconstruct the differentiation process from *in vivo* snap-shot data, which contains cells at distinct stages of differentiation [53]. Second, estimating the lineage of the cells is necessary to investigate multi-lineage differentiation. Although the expression of marker genes will be useful to classify cell lineages, the prior knowledge of marker genes is limited. Therefore, a lineage estimation method without prior knowledge is necessary to fully analyze the mechanisms of cell fate decisions.

To order cells without prior knowledge, several methods have been developed [62–64]. These methods use dimension reduction techniques, such as principal component analysis (PCA), and reconstruct the differentiation path in reduced space using several approaches, such as minimum spanning tree (MST) and principal curves. Each cell is projected onto the reconstructed path and the degree of differentiation of a cell (in pseudo-time) is defined by the projected position on the path. To estimate cell lineage from expression data, a few methods, which use the same framework, have been developed. Monocle [62], a dimension reduction-based approach, estimates the lineage of each cell by estimating multiple paths in reduced space and assigning each cell to one of the paths. These approaches are powerful tools to reconstruct the differentiation process without prior knowledge, and the development of such computational methods will help reveal the mechanisms of differentiation in conjunction with the advancement of single-cell technologies.

However, pseudo-time estimation and cell lineage estimation based on dimension reduction have several problems. For example, interpreting the biological meaning of the path in reduced space is difficult. Additionally, the position in reduced space is affected by noise and gene expression that is irrelevant to differentiation, and the results can therefore change significantly in a subsequent

analysis. Moreover, deterministic approaches, such as applications of MST in reduced space, cannot quantify the subtle differences among cells and are inadequate to estimate the lineages of cells at an early stage of bifurcation, which are important for analyzing cell fate decisions. Hence, we developed another approach based on stochastic processes.

In this research, we developed a novel method SCOUP (a probabilistic model to analyze Single-Cell expression data during differentiation with Ornstein–Uhlenbeck Process). SCOUP describes the dynamics of gene expression throughout differentiation directly, including pseudo-time and cell fate of individual cells. SCOUP is based on the Ornstein–Uhlenbeck (OU) process, which represents a variable moving toward an attractor with Brownian motion. In the case of differentiation, an attractor is regarded as a stable expression pattern of a gene after differentiation, and hence, an OU process is appropriate to describe expression dynamics throughout differentiation. Because OU processes suppose only a single attractor and cannot represent multi-lineage differentiation, we expand the typical OU process into a mixture OU process by representing the cell fate of each cell and lineage-specific expression patterns with latent values and different attractors, respectively. We compared the accuracy of pseudo-time estimates from SCOUP with those of previous methods using time-series scqPCR and scRNA-seq, and SCOUP was superior to previous methods in almost all conditions. We also evaluated the cell lineage estimation using scqPCR data in which cells exhibit multi-lineage differentiation. SCOUP successfully estimated cell lineage more accurately than Monocle, especially for cells at an early stage of bifurcation. In addition, SCOUP represents each gene expression dynamic directly and can be applied to various downstream analyses. As an example, we developed a novel correlation calculation method for elucidating regulatory relationships among genes. We normalized data based on the optimized parameters in our model, which assumes the conditional independency among genes, and calculated correlations within normalized data, and this method detected covariance that cannot be explained by the model alone. We applied this method to scRNA-seq data and detected a candidate of key regulator for differentiation and clusters in a correlation network which were not detected with conventional correlation analysis. Thus, SCOUP is a promising approach for further single-cell analysis and available at <https://github.com/hmatsu1226/SCOUP>.

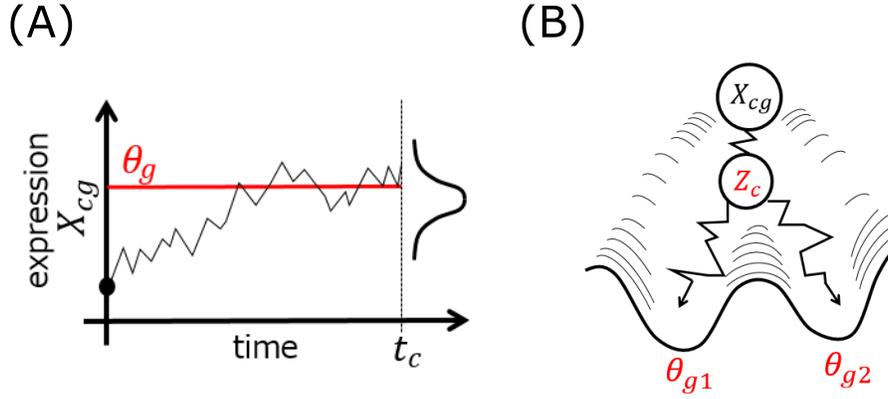


Figure 4.1. The conceptual diagrams of the OU process (A) and SCOUP for multi-lineage differentiation (B). (A) The OU process represents a variable (i.e., expression of a gene g in a cell c) moving toward attractor (θ_g) with Brownian motion. The value at time t satisfies the normal distribution (see “Methods”). (B) Each lineage has distinct attractor (θ_{g1} and θ_{g2}), and the lineage of a cell c is represented with latent value Z_c . The expression of gene g in cell c is described with the mixture OU process.

4.2 Methods

4.2.1 Ornstein-Uhlenbeck process

Let X_t be an OU process. X_t satisfies the following stochastic differentiation equation:

$$dX_t = -\alpha(X_t - \theta)dt + \sigma dW_t,$$

where α , θ , σ , and W_t denote the strength of relaxation toward the attractor, the value of the attractor, the strength of noise, and “white noise,” respectively. If the initial value is given by X_0 , the value at time t (X_t) satisfies the following normal distribution:

$$\begin{aligned} &P(X_t|X_0, \alpha, \sigma^2, \theta, t) \\ &= \mathcal{N}\left(X_t|e^{-\alpha t}X_0 + (1 - e^{-\alpha t})\theta, \frac{\sigma^2(1 - e^{-2\alpha t})}{2\alpha t}\right). \end{aligned}$$

This OU process represents a variable moving toward attractor θ with Brownian motion (Figure 4.1A) and has been used to describe adaptive evolution of a quantitative trait along phylogenetic tree [65], for example. This OU process suits the modeling of gene expression dynamics throughout differentiation by considering that θ , α , and σ represent specific expression patterns after differentiation, the speed of expression change, and level of noise, respectively. In this research, we extended

the OU process for single-cell expression data and developed a parameter optimization method.

4.2.2 OU process for single lineage differentiation

We developed a probabilistic model for single lineage differentiation. Hereinafter, we denote the number of cells, the number of genes, the cell index, and the gene index as C , G , c , and g , respectively. We assume that expression in each cell is independent and that the total probability $P(E|\Phi, T)$, where E is the expression data of all cells and genes and Φ is the set of parameters, is the product of cell probabilities. Each cell has a degree of differentiation progression parameter (i.e., pseudo-time) t_c . We also assume that each gene follows its OU process independently and has parameters α_g , σ_g^2 , and θ_g . Therefore, a cell probability is the product of gene expression probability $P(E_{cg}|\Phi_g, t_c)$, where E_{cg} is the expression data of gene g in cell c . Thus, the probability of single lineage differentiation is given by

$$\begin{aligned} P(E|\Phi, T) &= \prod_{c=1}^C \prod_{g=1}^G P(E_{cg}|\Phi_g, t_c) \\ &= \prod_{c=1}^C \prod_{g=1}^G \int dS_{cg} P_{\text{ou}}(E_{cg}|S_{cg}, \Phi_g, t_c) P(S_{cg}), \end{aligned}$$

where $\Phi_g = (\alpha_g, \sigma_g^2, \theta_g)$, $\Phi = \{\Phi_g | g = 1, \dots, G\}$, $T = \{t_c | c = 1, \dots, C\}$, S_{cg} is the expression of gene g in cell c at $t = 0$, and P_{ou} is a probability distribution based on an OU process and described by the following normal distribution:

$$\begin{aligned} &P_{\text{ou}}(E_{cg}|S_{cg}, \Phi_g, t_c) \\ &= \mathcal{N}\left(E_{cg} | e^{-\alpha_g t_c} S_{cg} + (1 - e^{-\alpha_g t_c}) \theta_g, \frac{\sigma_g^2 (1 - e^{-2\alpha_g t_c})}{2\alpha_g}\right). \end{aligned}$$

$P(S_{cg})$ is the initial distribution of a gene and is given by a normal distribution as follows:

$$P(S_{cg}) = \mathcal{N}(S_{cg} | \mu_{0g}, \sigma_{0g}^2).$$

In this research, we assume that μ_{0g} and σ_{0g}^2 are known because the expression data of progenitor cells are generally obtained.

4.2.3 Sufficient statistic for OU processes

Like a continuous Markov model for nucleotide evolution [66], the continuous OU process can be regarded as the limit of a discrete time OU process. $P_{\text{ou}}(E_{cg}|S_{cg}, \Phi_g, t_c)$ can be described as follows:

$$\begin{aligned} P_{\text{ou}}(E_{cg}|S_{cg}, \Phi_g, t_c) &= \lim_{N \rightarrow \infty} P_N(X_{cgN}|X_{cg0}, \Phi_g, t_c) \\ P_N(X_{cgN}|X_{cg0}, \Phi_g, t_c) &= \int dX_{cg} \prod_{s=1}^N P_{\text{ou}}(X_{cgs}|X_{cgs-1}, \Phi_g, t_c/N) \\ P(X_{cg}|\Phi_g, t_c) &= \prod_{s=1}^N P_{\text{ou}}(X_{cgs}|X_{cgs-1}, \Phi_g, t_c/N) P(X_{cg0}), \end{aligned}$$

where $X_{cg} = \{X_{cgs}|s = 0, \dots, N\}$ represents a path such that X_{cg0} and X_{cgN} satisfy S_{cg} and E_{cg} , respectively. In this model, we assume S_{cg0} is fixed and consider X_{cg} as $X_{cg} \in \{X_{cgs}|s = 1, \dots, N\}$ for simplicity (see supplementary text for the calculations related to S_{cg0}). Accordingly, we consider the likelihood of X_{cg} as follows:

$$P(X_{cg}|S_{cg}, \Phi_g, t_c) = \prod_{s=1}^N P_{\text{ou}}(X_{cgs}|X_{cgs-1}, \Phi_g, t_c/N).$$

According to the expansion of the above likelihood, the log-likelihood of X_{cg} is described as follows (see supplementary text for detailed calculation). Here, we abbreviate the indexes c and g and represent X_{cg} and X_{cgs} as X and X_s for simplicity.

$$\begin{aligned} l(X) &= \sum_{s=1}^N \ln P_{\text{ou}}(X_s|X_{s-1}, \Phi_g, t_c/N) \\ &= -\frac{N}{2} \ln \frac{\alpha}{\pi\sigma^2(1 - e^{-2\alpha t})} \\ &\quad - \frac{N}{2t\sigma^2} \left(2 \left(\sum_{s=1}^{N-1} X_s^2 - \sum_{s=0}^{N-1} X_s X_{s+1} \right) + X_0^2 + X_N^2 \right) \\ &\quad + \frac{\alpha}{2\sigma^2} \left(X_0^2 - X_N^2 - 2\theta X_0 + 2\theta X_N \right) \\ &\quad + \frac{\alpha^2 t}{2N\sigma^2} \left(-2 \sum_{s=1}^{N-1} X_s^2 + \sum_{s=0}^{N-1} X_s X_{s+1} + 2\theta \sum_{s=1}^{N-1} X_s - N\theta^2 \right) \\ &\quad + \mathcal{O}(1/N). \end{aligned}$$

Accordingly, we can calculate the log-likelihood by using the following statistics $\sum_{s=1}^{N-1} X_s^2$, $\sum_{s=0}^{N-1} X_s X_{s+1}$, and $\sum_{s=1}^{N-1} X_s$.

The expected values of the above statistics are sufficient for parameter optimization. The posterior probability $P(X_1 \dots X_{N-1} | X_N, X_0)$ is regarded as the multivariate normal distribution, and the expectation of X_s and X_s^2 can be calculated from the mean and variance–covariance matrix of the multivariate normal distribution. However, the expansion of the posterior probability gives only the $(N-1) \times (N-1)$ precision matrix, and we must therefore calculate the inverse of the matrix to obtain the variance–covariance matrix. Although we cannot use numerical methods to solve the inverse of the precision matrix because we consider N as the limit for infinite, we can solve for the inverse matrix analytically by using the tridiagonal property of the precision matrix [67]. By hand calculation, we showed that the expected values of these statistics were able to be solved analytically. For example, the expected value of one of the statistics is as follows:

$$\sum_{s=1}^{N-1} \langle X_s \rangle = \frac{X_0 + X_N - 2\theta}{\sinh \alpha t} \sum_{s=1}^{N-1} \sinh \left(s \frac{\alpha t}{N} \right) + (N-1)\theta + \mathcal{O}(1/N).$$

The detailed calculation is described in the supplementary text.

4.2.4 EM algorithm

We employed a parameter optimization using an expectation–maximization (EM) algorithm. When the likelihood function contains unobserved variables, an EM algorithm can be used for parameter optimization. The EM algorithm runs E step and M step iteratively and finds parameters that satisfy the local maximum of the marginal likelihood function. In the E step, we calculate the expectation of a specific statistic with current parameters. In the M step, we calculate the expected log-likelihood function (Q function) and optimize parameters so that they maximize the Q function. In our model, the path $X_{cg1} \dots X_{cgN-1}$ is unobserved, and the Q function is as follows:

$$\mathcal{Q}((\Phi, T), (\Phi^{\text{old}}, T^{\text{old}})) = \prod_c \prod_g \int dX_{cg1:N-1} P(X_{cg1:N-1} | X_{cgN}, X_{cg0}, \Phi_g^{\text{old}}, t_c^{\text{old}}) l(X_{cg}),$$

where $X_{cg1:N-1} = (X_{cg1}, X_{cg2}, \dots, X_{cgN-1})$.

The Q function can be expanded analytically with an expected value of the statistic described in the previous section. Thus, we can optimize Φ_g by solving

$dQ/d\theta_g = 0, dQ/d\alpha_g = 0, dQ/d\sigma_g^2 = 0$, which results in the following equations:

$$\begin{aligned}\theta_g^* &= \theta_g + \frac{1}{\sum t_c} \sum_c \frac{2(X_{cgN} - e^{-\alpha_g t_c} X_{cg0} - (1 - e^{-\alpha_g t_c})\theta_g)}{\alpha_g(1 + e^{-2\alpha_g t_c})} \\ \alpha_g^* &= \frac{\sum_c (-t_c \sigma_g^2 - (X_{cg0} - \theta_g)^2 + (X_{cgN} - \theta_g)^2)}{\sum_c Z_c^{\alpha_g}} \\ \sigma_g^{*2} &= \frac{1}{C} \sum_c \frac{2\alpha_g}{1 - e^{-2\alpha_g t_c}} (X_{cgN} - e^{-\alpha_g t_c} X_{cg0} - (1 - e^{-\alpha_g t_c})\theta_g)^2,\end{aligned}$$

where Z^α is explained in the supplementary text. The pseudo-time variable t_c cannot be optimized analytically, and we therefore solve t_c to satisfy $dQ/dt_c = 0$ by Newton's method.

In cases, X_{cg0} is also unobserved, so we must calculate the expected value of X_{cg0} . As such, we calculate the expected values, including the expected value of X_{cg0} and X_{cg0}^2 , in the E step and optimize parameters with the above equation in the M step. The detailed optimization process and calculation are described in the supplementary text.

We validated our parameter optimization method with simulation data and confirmed that SCOUP succeeded to optimize parameters so that the marginal likelihood was maximized (see supplementary text).

4.2.5 Mixture OU process for multi-lineage differentiation

We also extended the single lineage model to a mixture model in order to consider multi-lineage differentiation, such as bifurcation (Figure 4.1B). We assume that the number of lineages is known and given by K and that each lineage has a different attractor θ_{gk} . The fate of a cell c is unknown and is represented with the latent value Z_c , which is 1 of K representations. With this latent value, the mixture OU process is given by

$$\begin{aligned}P(E_c, S_c) &= \sum_{k=1}^K \pi_k \prod_{g=1}^G P_{\text{ou}}(E_{cg} | S_{cg} \alpha_g, \sigma_g, \theta_{gk}, t_c) P(S_{cg}) \\ P(E_c, S_c, Z_c) &= \prod_{k=1}^K \pi_k^{Z_{ck}} \prod_{g=1}^G (P_{\text{ou}}(E_{cg} | \alpha_g, \sigma_g, \theta_{gk}, t_c) P(S_{cg}))^{Z_{ck}},\end{aligned}$$

where π_k is the probability of lineage k .

Here, Z_c is an unobserved value, and we maximize the marginal likelihood with the EM algorithm. As described in the previous section, we must calculate the expectation of the unobserved

value to calculate the Q function. The posterior probability of Z_c and the expectation of Z_c (γ_{ck}) are described as follows:

$$P(Z_c|E_{cg}, S_{cg},) \propto \prod_{k=1}^K \left(\pi_k^{Z_{ck}} \prod_{g=1}^G P_{\text{ou}}(E_{cg}|S_{cg}, \theta_{gk}, t_c)^{Z_{ck}} \right)$$

$$\gamma_{ck} = E[Z_{ck}] = \frac{\pi_k \prod_{g=1}^G P_{\text{ou}}(E_{cg}|S_{cg}, \theta_{gk}, t_c)}{\sum_{k'} \pi_{k'} \prod_{g=1}^G P_{\text{ou}}(E_{cg}|S_{cg}, \theta_{gk'}, t_c)}.$$

By using the above equation and previous description, we can calculate the Q function analytically. We optimize

$$\pi_k = \frac{\sum_c \gamma_{ck}}{\sum_c \sum_{k'} \gamma_{ck'}}$$

by solving $dQ/d\pi_k = 0$. Other parameters are optimized likewise using the single lineage model. Accordingly, we calculate the expected values of variables, such as γ_{ck} and S_{cg0} , in the E step and update parameters in the M step.

The lineage of a cell is estimated from the expectation of the latent value of a cell (γ_c). SCOUP can quantify the certainty of the estimated lineage of a cell from the value of γ_c .

4.2.6 Initialization of time parameter

Our method might converge to undesirable local optima if T is initialized randomly. For example, estimated pseudo-time might be inferred in the reverse order of differentiation. To avoid undesirable local optima, rough initialization of T is effective. Although experimental time will be useful for initialization, such data are not always available. For example, experimental time does not exist for expression data of an in vivo snap-shot sample [53]. Therefore, an initialization method that does not depend on experimental time is necessary. Here, we explain our initialization method based on a dimension reduction approach.

we developed dimension reduction approach for pseudo-time initialization, called SP (pseudo-time calculation based on Shortest Path from the root cell in the MST). Firstly, we added the mean of the initial distribution ($\mu \in \{\mu_{g0}|g = 1..G\}$) to expression data and regarded it as an initial point for the pseudo-time calculation. Next, we performed PCA, constructed MSTs in the reduced space, searched for the shortest path from an initial point using Prim's algorithm, and regarded the weight

of the shortest path as the pseudo-time. In this paper, we set the dimensionality of the PCA to two and used this pseudo-time for the initialization of our method.

4.2.7 Dimension reduction approach

In this section, we explain the previous pseudo-time estimation methods based on a dimension reduction approach.

Monocle [62] constructs a MST in reduced space, searches for the longest path in the MST, and estimates pseudo-time along the longest path. We added the mean of the initial distribution data and regarded it as an initial point for the pseudo-time calculation. We used all genes in a dataset as marker genes and the other parameters of Monocle were set to default values, unless otherwise specified.

TSCAN [64] performs model-based clustering in reduced space, connects clusters, and estimates pseudo-time by projecting cells onto the connected path. Although TSCAN can infer an order of clusters, it cannot regard a point as an initial point. Therefore, we compared the accuracy of outputted pseudo-time with reversed pseudo-time and defined the pseudo-time of TSCAN as the superior one. Because TSCAN failed to output pseudo-time of partial cells when we set a high number of clusters, we set the number of clusters to three in this research.

In this paper, we compared the performance of SCOUP with those of above dimension reduction-based methods in addition to SP.

4.2.8 Correlation between genes

We also proposed a novel correlation function between two genes. Although we assume the conditional independence among genes to represent gene dynamics, we can detect the regulatory relationship between genes by calculating the covariance. Our correlation function quantifies the covariance between genes that is not explained by our model.

For time-series data, a ordinal correlation coefficient will be high even if two variables only have similar time-trend. For example, any two independent genes that are upregulated in accordance with differentiation exhibit a high correlation. In the case of the detection of interactions between genes, it is most appropriate to remove the influence of time-trend. To remove this trend effect, the expression data at a specific experimental time point is often used to calculate the correlation. However,

this approach is insufficient to remove the time effect resulting from the difference between the experiment time and the progression of cells. Accordingly, the trend effect is best removed by using cells within a specific pseudo-time span for calculation. Although this analysis will remove the trend effect, the number of cells that are used for the calculation decreases owing to the limit of the span of pseudo-time and precise calculation will therefore be difficult.

To overcome this problem, we developed a novel correlation function based on our probabilistic model. As described in the section on "OU process for single lineage differentiation" and the supplementary text, the probabilistic distribution of the expression of a gene g at time t (X_{tg}) is described as follows:

$$P(X_{tg}|\Phi_g, t_c) = \int dS_g P_{ou}(X_{tg}|S_g, \Phi_g, t)P(S_g) = \mathcal{N}(X_{tg}|\mu_{tg}, \sigma_{tg}^2),$$

where

$$\begin{aligned}\mu_{tg} &= e^{-\alpha_g t} \mu_{0g} + (1 - e^{-\alpha_g t}) \theta_g \\ \sigma_{tg}^2 &= \frac{\sigma_g^2 (1 - e^{-2\alpha_g t})}{2\alpha_g} + e^{-2\alpha_g t} \sigma_{0g}^2.\end{aligned}$$

As such, we can remove the time dependency by standardizing the time-dependent mean and variance as follows:

$$Z_{cg} = \frac{E_{cg} - \mu_{t_{cg}}}{\sigma_{t_{cg}}^2}.$$

We calculated the correlation coefficient for the above standardized values. This correlation function can detect gene pairs that exhibit interactions that are unexplained by the model, which assume the conditional independence among genes.

The above standardization assumes a single normal distribution and is not suitable for multi-lineage model. However, $\max_k \gamma_{ck}$ of most cells, which we analyzed, were about 1.0, and hence, most cells would be assigned to one of the lineage. Therefore, the standardization will be effective by assigning a cell to a relevant lineage. In addition, correlation of each lineage will be calculated by dividing cells into each lineage in advance.

4.2.9 Dataset

4.2.9.1 single-cell qPCR for single-lineage differentiation

We used the time-series single-cell qPCR dataset produced by Kouno’s group [68] from THP-1 human myeloid monocytic leukemia cells differentiating into macrophages. They investigated the expression of 45 transcription factors by 120 single cells at each eight time point (0, 1, 6, 12, 24, 48, 72, and 96 h) after phorbol myristate acetate stimulation.

To evaluate the estimated pseudo-time in many conditions, we constructed a dataset, (Kouno’s data (1)) follows. We added noise to raw expression data as described below to investigate the effect of noise in pseudo-time estimation. We added noise to raw expression data E_{cg} by adding $\bar{E}_g \times U_R[0, \epsilon]$, where \bar{E}_g is the mean expression of a gene and $U_R[0, \epsilon]$ is a uniform random number from 0 to ϵ . We produced 20 replicates for each ϵ (noise level), and validated the pseudo-time of each method for each noise level.

We also constructed another dataset, (Kouno’s data (2)), to validate lineage estimation by adding 45 pseudogenes that exhibit various expression patterns among lineages. We initially selected 60 cells randomly from 120 cells at a given time point. The expression $E_{cg'}$ of a pseudogene g' by the selected cells is equal to raw expression ($E_{cg'} = E_{cg}$). For the remaining cells, we inverted the raw expression in relation to the initial mean ($E_{cg'} = -2E_{cg} + \mu_{0g}$). We also added noise as mentioned above in regard to Kouno’s data (1). Because Monocle cannot accept negative values, we incremented the values by a minimum of 1 to make the expression positive.

The initial distribution (μ_{0g} and σ_{0g}^2) was calculated from 0-h cells.

4.2.9.2 single cell qPCR for bifurcation

To validate the lineage estimation in real data, we used a dataset produced by Moignard’s group [69]. They investigated the single-cell qPCR results for 46 transcription factors throughout hematopoietic development from embryonic day (E) 7.0 to E8.5 in mouse embryos. These data include a lineage bifurcation between E7.75 and E8.25; at this time, head fold (HF) cells differentiate into putative blood and endothelial populations, which are distinguished as either GFP⁺ cells (4SG) or Flk1⁺GFP⁻ cells (4SFG⁻). We used the expression profiles of HF, 4SG, and 4SFG⁻ and investigated whether SCOUP and Monocle can classify 4SG and 4SFG⁻ using only their expres-

sion profiles. We regarded any cell in which the expression of more than half of the genes was not detected as an outlier, and these outliers were removed. In addition, we randomly selected 1,000 cells because Monocle did not seem to work correctly for a large number of cells. These screening procedures left 398 HF cells, 342 4SG cells, and 260 4SFG⁻ cells. The initial distribution was calculated from HF cells.

4.2.9.3 Single-cell RNA-seq for single-lineage differentiation

We also investigated the stimulation time-series single-cell RNA-seq dataset (at 0, 1, 4, and 6 h) for primary mouse bone-marrow-derived dendritic cells that was produced by Shalek's group [70]. This dataset contains data for three different time series corresponding to each of the different stimulation methods: lipopolysaccharide (LPS), viral-like double-stranded RNA (PIC), and synthetic mimic of a bacterial lipopeptides (PAM). First, we converted transcripts per million (TPM) to $\log(\text{TPM} + 1)$ and defined this value as gene expression. Next, we removed outlier cells so that each cell in the dataset contained more than 4,000 genes with detectable levels of expression; this left 281 LPS cells, 224 PAM cells, and 159 PIC cells. Third, we calculated the absolute difference in mean gene expression between the 1-h cells and 6-h cells for each stimulation. We extracted the top 1,000 genes in descending order of this difference for each stimulation and used these genes for pseudo-time estimation. We also added unstimulated cells (outlier cells were removed through a procedure like that described above, leaving 85 cells) to the LPS, PAM, and PIC data and regarded these cells as 0-h data. The initial distribution was calculated from unstimulated cells.

4.2.10 Accuracy measure

4.2.10.1 Pseudo-time evaluation

To evaluate the accuracy of pseudo-time estimated from each method, we regarded experimental time as genuine time and calculated the rate of inconsistency between pseudo-time and experimental time. By using the accuracy measure of TSCAN as a reference, we evaluated the inconsistency by calculating the rate of cell pairs whose pseudo-time ordering was inconsistent with experimental-

time ordering, and we defined the pseudo-time inconsistency score (PIS) as follows:

$$\text{PIS} = \frac{\sum_{(i,j) \in (t_i^{(e)} < t_j^{(e)})} I(t_i > t_j)}{\sum_{(i,j) \in (t_i^{(e)} < t_j^{(e)})} (I(t_i < t_j) + I(t_i > t_j))},$$

where $t_c^{(e)}$ and t_c are respectively the experimental time and pseudo-time of cell c . $I(t_i < t_j)$ is an indicator function that takes the value 1 if the conditional expression is true.

4.2.10.2 Lineage evaluation

We evaluated the performance of lineage estimation by SCOUP and Monocle by comparing the cell lineage annotation of each cell. The annotation of a cell from simulation data is obvious and that of Moignard's data is given by 4SG or 4SFG⁻ in accordance with GFP⁺ or Flk1⁺GFP⁻. SCOUP estimates a cell lineage based on the expectation of the posterior probability of cell fate (γ_{ck}). We classified cells into one of two lineages on the basis of whether γ_{ck} exceeded a threshold. We calculated the precision and recall for each threshold and calculated the area under the curve (AUC) value. Monocle also can estimate cell lineage by setting the parameter *num_paths* to 2, thereby outputting the state of a cell as either state1 (pre-bifurcation), state2 (one lineage), or state3 (another lineage). Monocle is a deterministic method and cannot distinguish subtle differences. Therefore, we regard that state1, state2, and state3 belong to one lineage with probabilities 0.5, 1.0, and 0.0, respectively. We calculated the AUC value for Monocle in the same way.

4.3 Results and discussion

4.3.1 Validation of parameter optimization

We validated our parameter optimization method with simulation data. We generated simulation data from the normal distribution based on the OU process by varying the parameters. The number of genes and cells are set to 500 and 100, respectively.

Firstly, we compared the values of estimated parameters with those of true parameters (Figure 4.2A,B). The values of estimated time and estimated θ_g are highly correlated with those of true values (r^2 are 0.94 and 0.96, respectively). The values of estimated mean and variance of the OU process are also highly correlated with those of true mean and variance (0.99 and 0.94, respec-

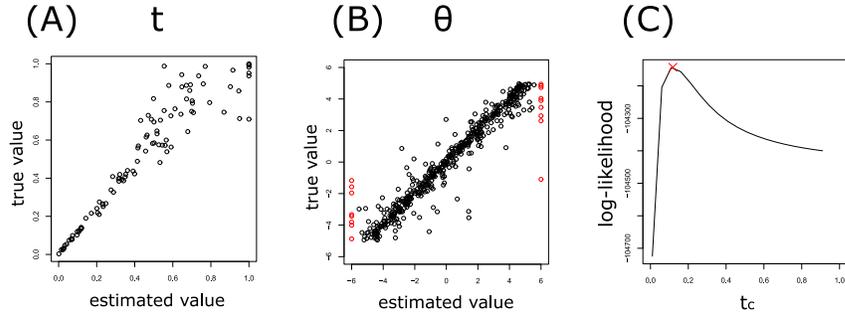


Figure 4.2. Validation of parameter estimation of SCOUP for simulation data. (A) and (B) is the comparison between the estimated values and true values for pseudo-time (t) and θ_g , respectively. The outlier whose estimated value exceeds the boundary of drawing area is visualized in the border with a red circle for visualization. (C) is the log-likelihood curve with respect to t_c of a cell. The optimized t_c is indicated with x-max.

tively), and hence, SCOUP succeeded to reconstruct the original probabilistic distribution with high accuracy (the details are described in the supplementary text).

Next, we investigated that the log-likelihood of optimized parameters was higher than those of varied parameters. Figure 4.2C is the example of the log-likelihood curve with respect to time parameter of a cell (t_c), and the value of optimized t_c is drawn with x-mark. The log-likelihood of the optimized t_c was located in the top of the log-likelihood curve. We also verified that the optimized parameters were located in the top of the log-likelihood surface in regards to other parameters (the details are described in the supplementary text). Thus, SCOUP can optimize the parameters correctly.

4.3.2 Validation of pseudo-time estimation

In this section, we compared the accuracy of the pseudo-time of each method: SCOUP, our method; SP, pseudo-time estimation based on shortest path in the MST in reduced space; Monocle, dimension reduction-based method that reconstruct differentiation path by the longest path in the MST; TSCAN, dimension reduction-based method that reconstruct differentiation path by running model-based clustering and connecting clusters. For pseudo-time evaluation, we used Kouno's data (1) and the Shalek's data.

Figure 4.3 shows the histograms of pseudo-time inferred by each method for Kouno's data (1) without additional noise ($\epsilon = 0$). The histograms are drawn for each experimental time point. Al-

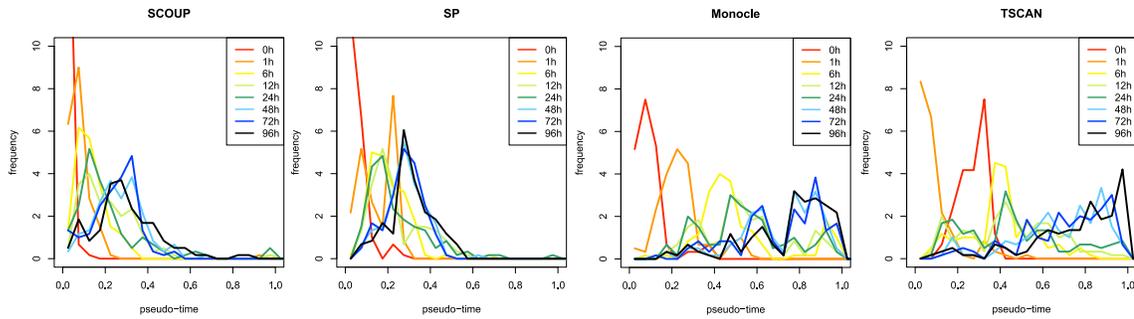


Figure 4.3. The histograms of pseudo-time estimates produced by each method for Kouno’s data (1) without additional noise. The histograms are drawn for each experimental time point with different colors. The pseudo-time values inferred by SCOUP over 1.0 are integrated into 1.0 for visualization. The pseudo-time values inferred by Monocle and TSCAN are normalized so that maximum = 1.0.

though the pseudo-time trends of each method are roughly consistent with experimental time order, each method shows distinctive characteristics. In most cases, the orders of pseudo-time produced by TSCAN for 0-h cells and 1-h cells are reversed. The orders might be reversed in the process of assigning cells to clusters or ordering clusters. In SP, the pseudo-time of the portion of cells is large and that of the remaining cells is relatively small. This is because a portion of the cells must be outliers and are therefore located far from other cells in reduced space. The outliers cause long paths in the MSTs and affect other pseudo-time estimates through normalization. Monocle seems to successfully order cells. In SCOUP, the pseudo-times of 0-h cells are relatively concentrated at $t = 0.0$ as compared to the other methods. The pseudo-time of 0-h cells based on dimension reduction approaches is dispersed because 0-h cells tend to scatter in reduced space owing to the dispersion of expression and noise. In contrast, SCOUP contains a noise term in the model and estimates pseudo-time from the trend of total gene expression, which removes the influence of noise. Because 0-h cells are progenitor cells and belong to a steady state before differentiation, it is appropriate to consider the pseudo-time of 0-h cells as approximately 0. Thus, SCOUP successfully identified the initial steady state.

Next, we quantitatively evaluated the accuracy of pseudo-time estimated by each method for Kouno’s data (1) based on the pseudo-time inconsistency score (PIS) (Figure 4.4). The PISs of SCOUP were superior to those of other methods under most conditions. This demonstrates that SCOUP can estimate pseudo-time well, even from noisy data. Under one condition, the PIS of Monocle was superior to that of SCOUP, and SCOUP was the second best. This can be because

SCOUP does not describe the differentiation process completely. For example, SCOUP cannot represent variable attractors, such as transient patterns, and dimension reduction-based methods might be able to accommodate such expression patterns. In future work, we will extend SCOUP to represent such dynamics.

We also investigated the effect of the number of dimensions of reduced space for pseudo-time estimation in Monocle. We set the argument of Monocle *max_components*, which corresponds to the number of dimensions, to 2 and 3 and denote Monocle analyses with each configuration as Monocle(2) and Monocle(3), respectively. Across all conditions, Monocle(3) was inferior to Monocle(2). This is because the third dimension of reduced space represents something unrelated to differentiation. Without prior knowledge, it is difficult to set a proper number of dimensions, and pseudo-time can be erroneous under an improper number of dimensions. Although SCOUP is based on a dimension reduction approach in the process of pseudo-time initialization, we verified that the pseudo-time estimated from different numbers of dimensions (i.e., 2 and 3) converged to almost same value in this dataset ($r^2 = 0.94$ for $\epsilon = 0.0$). Even if the estimated pseudo-times of SCOUP differ, we can infer appropriate pseudo-times by selecting the model with the highest likelihood.

Next, we evaluated the pseudo-time of each method as inferred from Shalek’s data. The PIS of each method is shown in Table 4.1. Across all conditions, the PISs of SCOUP were superior to those of other methods. Unlike qPCR, RNA-seq provides comprehensive gene expression profiles and contains the expression of genes that are largely unrelated to differentiation. SCOUP can omit the effect of such genes by reducing the weight of their influence automatically in pseudo-time optimization. In contrast, the positions of cells in reduced space will be affected and the pseudo-time will vary with the presence of such genes. Moreover, the dispersion of RNA-seq is higher than that of qPCR, which influences the analyses.

The PISs of PIC and PAM were higher than those of LPS. This will be because the numbers of PIC and PAM cells were lower than that of LPS. It is difficult to reconstruct differentiation trajectories from a small number of samples. In particular, it is important to obtain cells distributed evenly throughout the differentiation process in order to reconstruct trajectories with high accuracy.

In summary, SCOUP estimated pseudo-time with high accuracy, especially for RNA-seq data. Moreover, SCOUP successfully identified the initial state which was difficult to be detected with dimension reduction-based approaches. In addition, SCOUP is based on a probabilistic model, and

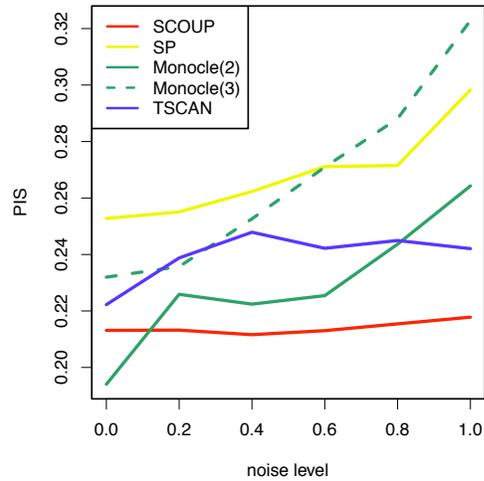


Figure 4.4. PIS of each method applied to Kouno’s data (1). The x -axis represents the noise level (ϵ) (see “Methods”) and the y -axis represents the degree of inconsistency between the pseudo-time and experimental time (PIS). Each method is distinguished by color: red, SCOUP; yellow, SP; green, Monocle; and blue, TSCAN. We compared the PIS of Monocle for different parameters $max_components$, which correspond to dimensions. The solid and dotted lines correspond to $max_components = 2$ and 3, respectively.

Table 4.1. PIS for each method applied to Shalek’s data. Each row represents the method, and each column represents the kind of stimulation for differentiation. NA means that Monocle did not work well.

	LPS	PIC	PAM
SCOUP	0.03	0.12	0.12
SP	0.14	0.32	0.17
Monocle(2)	NA	0.38	NA
Monocle(3)	0.18	0.45	0.32
TSCAN	0.17	0.27	0.24

hence can evaluate proper pseudo-time by using likelihood. Thus, SCOUP has advantages over dimension reduction-based methods in pseudo-time estimation.

4.3.3 Validation of cell lineage estimate

In this section, we evaluate the accuracy of cell lineage estimation from single-cell expression data containing lineage bifurcation.

First, we validated SCOUP and Monocle with simulation data (Kouno’s data (2)). Table 4.2 shows the mean AUC values of each method for each condition. The AUC values for SCOUP were

Table 4.2. Mean AUC values for cell lineage estimates using each method for Kouno’s data (2).

	$\epsilon = 0.0$	$\epsilon = 0.5$	$\epsilon = 1.0$
SCOUP	0.99	0.99	0.99
Monocle	0.98	0.97	0.95

higher than those for Monocle in every condition. Figure 4.5 summarizes cells in the space of the first two PCs for expression data with $\epsilon = 1.0$. The color of each cell represents its genuine cell lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with Monocle (right). Both methods estimated cell lineage with high accuracy for cells that were sufficiently separated in PCA space. This result suggests that estimating the lineage of a cell whose expression pattern has changed sufficiently after bifurcation is not difficult using these methods. However, Monocle was not able to estimate cell lineage correctly for cells whose expression pattern did not change sufficiently after bifurcation. In contrast, SCOUP successfully quantified the certainty of lineage of such cells and estimated their lineages with fairly high accuracy (Table 4.2). To understand cell fate decision mechanisms, it is important to analyze cells immediately after bifurcation. Therefore, SCOUP, which can quantify the certainty of estimated cell lineage and accurately estimate the lineage of cells that have just undergone bifurcation, will be useful for investigations of cell fate decision mechanisms.

Next, we investigated cell lineage estimation using Moignard’s data. The Moignard’s data includes the lineage bifurcation as follows; head fold (HF) cells differentiate into putative blood and endothelial populations, which are distinguished as either GFP^+ cells (4SG) or $\text{Flk1}^+\text{GFP}^-$ cells (4SFG⁻). SCOUP was able to distinguish cells of 4SFG⁻ and 4SG almost completely correctly (AUC value = 1.0). The AUC value for Monocle was 0.81. Figure 4.6 shows cells in the space of the first two PCs and the colors of cells indicate the genuine cell lineage (left), the lineage estimated using our method (middle), and the lineage using Monocle (right). The lineage estimation using SCOUP were highly consistent with cell annotations, while Monocle incorrectly regarded a non-negligible number of 4SFG⁻ cells as 4SG cells. This tendency of Monocle did not change when we changed the dimension number parameter (*max_components*). In contrast with simulation data, which were produced based on symmetric bifurcation, real data likely show complicated bifurcation patterns, and hence, a deterministic approach, such as MST in reduced space, might be inadequate

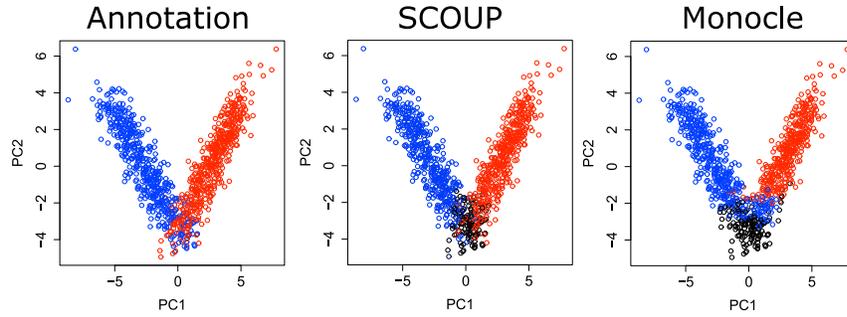


Figure 4.5. PCA of cells of Kouno's data based on gene expression. The cell colors indicate the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with Monocle (right). The color for SCOUP is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for Monocle is defined by estimated states: black, state 1 (pre-bifurcation); red, state 2; and blue, state 3. The color of each state is defined so that they are consistent among each plots.

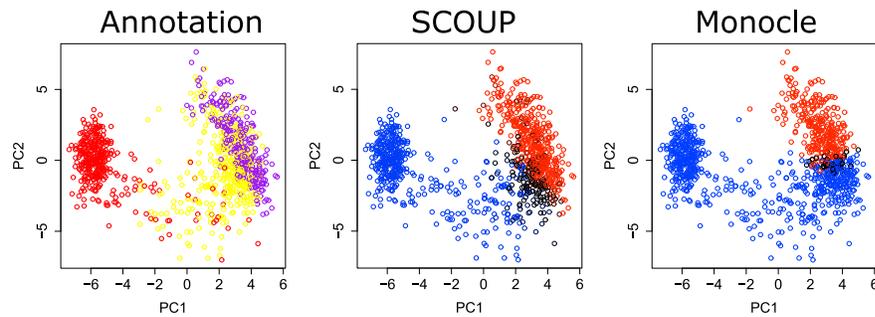


Figure 4.6. PCA of cells of Moignard's data based on gene expression. The cell colors represent the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with Monocle (right). The color for the genuine lineage is defined by the annotation of the cell; yellow, HF; red, 4SG; and purple, 4SFG⁻. The color for the SCOUP analysis is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for the Monocle analysis is defined by estimated states; black, state 1 (pre-bifurcation); red, state 2; and blue, state 3. We determined the color of each state so that they are consistent among each plot.

to capture bifurcation.

The results described above show that SCOUP is superior to Monocle with respect to cell lineage estimation for both simulated and real data. SCOUP can capture subtle differences in cells immediately after bifurcation and will be a powerful method for investigations of cell fate decision mechanisms.

We also investigated cell lineage estimation with Gaussian mixture model (GMM) implemented in mclust package [71]. The AUC values for mclust were inferior to those of SCOUP, and mclust was not able to estimate cell lineage correctly for cells at an early stage of bifurcation (see supplementary text for AUC values and PCA plots of mclust). This is because mclust does not have time

parameters in the model and will work well only for cells whose expression pattern has sufficiently changed after bifurcation. Moreover, GMM fitted to the position in which large number of cells exist for Moignard’s data. Therefore, GMM is inadequate to estimate the path of bifurcation in the condition that cells are unevenly distributed. Thus, it is important to take time parameters into account to estimate the path of differentiation and cell lineage.

4.3.4 Clustering genes

We grouped genes for Shalek’s data based on expression patterns along pseudo-time estimated with SCOUP. Hereafter, we used the data for LPS stimulation because the number of LPS cells is largest in Shalek’s data. In this analysis, we investigated the top 5,000 genes by the clustering method implemented in Monocle. Monocle regards the expression pattern as a function of pseudo-time and calculates a smooth response curve based on generalized additive models. Then, Monocle defines the distance between two genes as $1 - \rho_{xy}/2$, where ρ is the Pearson correlation coefficient of standardized response curves, and groups genes with K-medoids clustering. In this analysis, we set the number of clusters as 6 and the overall trend in expression pattern for each cluster and the number of genes in each cluster are shown in Figure 4.7 and Table 4.3.

We performed gene ontology (GO) enrichment analyses for genes in each group with DAVID [72, 73], and the top three GO terms (ordered by p -value) for each cluster are shown in Table 4.4. The cells of Shalek’s data are differentiated into dendritic cells, and immune response genes were upregulated (groups 1 and 2). Genes in groups 4 and 5 were downregulated and were enriched for the cell cycle GO term, consistent with previous research [74]. In this previous study, increased energy usage was also detected. In our analysis, genes related to energy usage were enriched in groups 3 and 6, which show a transient upregulation. Thus, we can classify gene function based on expression patterns along pseudo-time and the landscape of gene regulation can be characterized by investigating differences in these patterns. For example, although both groups 1 and 2 exhibited an upregulation, its timing was later for group 2 than group 1. The GO term related to “antigen” was enriched only in group 2, and this might reflect a different regulatory cascade during differentiation. We also calculated KEGG pathway enrichment for genes of group 1 and group 2, respectively. Group 2 did not include the term of KEGG pathway whose Benjamin-adjusted p -value was less than 10^{-5} , whereas the term “Toll-like receptor signaling pathway” was the most significantly enriched

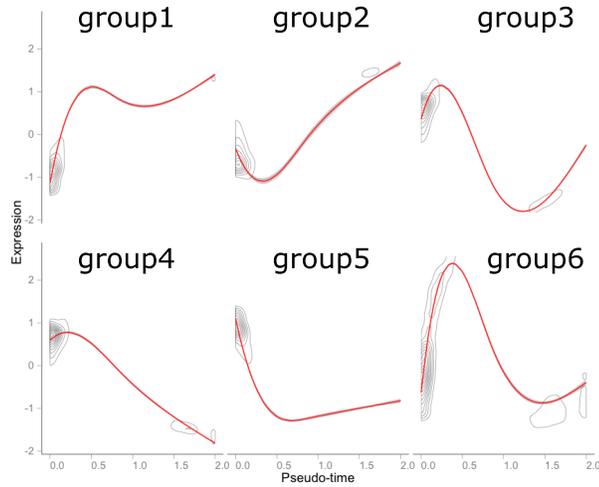


Figure 4.7. Overall trend in standardized expression patterns along pseudo-time for each group. This plot is drawn with the `plot_clusters` function in the Monocle package.

Table 4.3. The number of top 5,000 genes, top 1,000 genes in each group. The total number are not equal to 5000 and 1000 because the response curves for a few genes could not be calculated.

	group					
	1	2	3	4	5	6
total	867	403	958	1354	778	599
top-1000 gene	260	81	177	291	76	111

in group 1 and Benjamin-adjusted p -value was 6.5×10^{-7} . This data is the RNA-Seq of LPS stimulated bone-marrow derived dendritic cells and LPS is known to activate “Toll-like receptor signaling pathway” at first which cause the up-regulation of “antigen processing and presentation” a little late [75]. Our result is consistent with such mechanisms. Thus, investigations of expression patterns along pseudo-time can elucidate the regulatory machinery involved in differentiation.

4.3.5 Correlation analysis

In this research, we propose a novel correlation analysis by using standardization based on SCOUP to detect covariance that cannot explained by the model that assumes the conditional independence among genes alone, and investigated the regulatory relationships among genes using correlations within raw expression data or standardized expression data. Hereafter, we refer to the correlations within raw data and standardized data as C_{Raw} and C_{Std} , respectively. We first investi-

Table 4.4. The top three GO terms for each group. The third column shows the negative logarithm of the Bonferroni-adjusted p -value.

group	GO term	$-\log_{10}(p)$
1	immune response	22.9
	defense response	11.4
	response to wounding	7.0
2	antigen processing and presentation	5.5
	immune response	3.8
	antigen processing and presentation of exogenous antigen	3.3
3	generation of precursor metabolites and energy	5.1
	protein localization	4.8
	establishment of protein localization	3.2
4	cell cycle	9.6
	cell division	7.9
	ribonucleoprotein complex biogenesis	7.7
5	translation	6.7
	M phase of mitotic cell cycle	3.2
	cell cycle	2.9
6	generation of precursor metabolites and energy	11.5
	protein transport	5.6
	establishment of protein localization	5.5

Table 4.5. The top three transcription factors and their related genes for group 1. The left and right tables correspond to $\overline{C}_{\text{Raw}}(i, 1)$ and $\overline{C}_{\text{Std}}(i, 1)$, respectively. The first column of each table contains the rank of the absolute difference of expression between 1-h cells and 6-h cells, and the second column lists the gene names. The third column contains the $\overline{C}_{\text{Raw}}(i, 1)$ or $\overline{C}_{\text{Std}}(i, 1)$ of the candidate genes.

rank	Gene Symbol	C_{Raw}	rank	Gene Symbol	C_{Std}
5	<i>Ifit1</i>	0.46	313	<i>Sqstm1</i>	0.076
6	<i>Ifi205</i>	0.44	45	<i>Ifih1</i>	0.071
17	<i>Ifi204</i>	0.43	5	<i>Ifit1</i>	0.071

gated whether the target genes of a transcription factor (TF) can be predicted under the assumption that the expression of a TF and its target genes are highly correlated. The list of TFs and their target genes was downloaded from the Integrated Transcription Factor Platform (ITFP) [76], a database containing 71 TFs and 648 pairs of TFs and target genes in the top 1,000 genes. We calculated the C_{Raw} and C_{Std} values between 71 TFs and the remaining 929 genes and extracted from the top 1,000 positively correlated pairs of TFs and genes according to each correlation method. The top 1,000 C_{Raw} and C_{Std} values contained correlations of 24 and 27 annotated pairs, respectively (see supplementary text for the list of detected annotated pairs), and the probabilities of capturing these annotated pairs by random sampling are $p < 6.2 \times 10^{-5}$ and $p < 2.8 \times 10^{-6}$, respectively. This

suggests that target genes of a specific TF can be predicted from a correlation analysis of single-cell expression data.

Only three annotated pairs were common between the 24 C_{Raw} correlation values and the 27 C_{Std} correlation values, which indicates that different regulatory relationships were detected when analyzing raw and standardized expression data. Analysis of standardized expression data revealed correlations that were not explained by the model that assumes the conditional independence among genes, whereas raw expression data analysis revealed correlations produced from similar expression patterns during differentiation. Thus, our novel correlation analysis method can deliver new insights that are not detected by conventional correlation methods.

Next, we aimed to detect a key regulator of each group by using the two correlation methods. We downloaded the candidates of key regulator TFs and their related genes from the Riken Transcription Factor Database (TFdb) [77] and FANTOM5 SSTAR [78] as well as TF data from ITFP. In this analysis, 117 genes of the annotated TFs and their related genes were contained in top 1,000 gene and were considered as key regulator candidates. We calculated the C_{Raw} (and C_{Std}) values between each candidate and genes in a group, and calculated the average C_{Raw} (C_{Std}) value of the candidate for the group. We denote these values as $\bar{C}_{\text{Raw}}(i, j)$ and $\bar{C}_{\text{Std}}(i, j)$, where i is the index of a candidate and j is the index of a group. We assumed the key regulator of the group is highly correlated with genes in the group and investigated to detect the key regulators by extracting the candidates of high $\bar{C}_{\text{Raw}}(i, j)$ or $\bar{C}_{\text{Std}}(i, j)$. There were few differences between $\bar{C}_{\text{Raw}}(i, j)$ and $\bar{C}_{\text{Std}}(i, j)$ for groups 3 and 6 because our standardization was inadequate to deal with the transient patterns found in these groups. The difference between $\bar{C}_{\text{Raw}}(i, 1)$ and $\bar{C}_{\text{Std}}(i, 1)$ was largest among all groups, and therefore we focused on group 1 hereafter.

Table 4.5 shows the top three candidates according to $\bar{C}_{\text{Raw}}(i, 1)$ and $\bar{C}_{\text{Std}}(i, 1)$, respectively. The $\bar{C}_{\text{Raw}}(i, 1)$ candidates are basically the genes which have large absolute expression differences between 1-h cells and 6-h cells. The large absolute expression difference can bring about high spurious correlation due to the similar expression trends during differentiation. Thus, C_{Raw} is likely to be influenced by spurious correlation and therefore is inadequate to detect the key regulator. As for $\bar{C}_{\text{Std}}(i, 1)$, *Sqstm1* is the top rank. The absolute expression difference rank of *Sqstm1* is 313 of 1,000 genes and the $\bar{C}_{\text{Raw}}(i, 1)$ rank of *Sqstm1* is 29 of 117 candidates. *Sqstm1*, which is also called p62, has been suggested to be a key intracellular target of innate defense regulator peptides [79] and

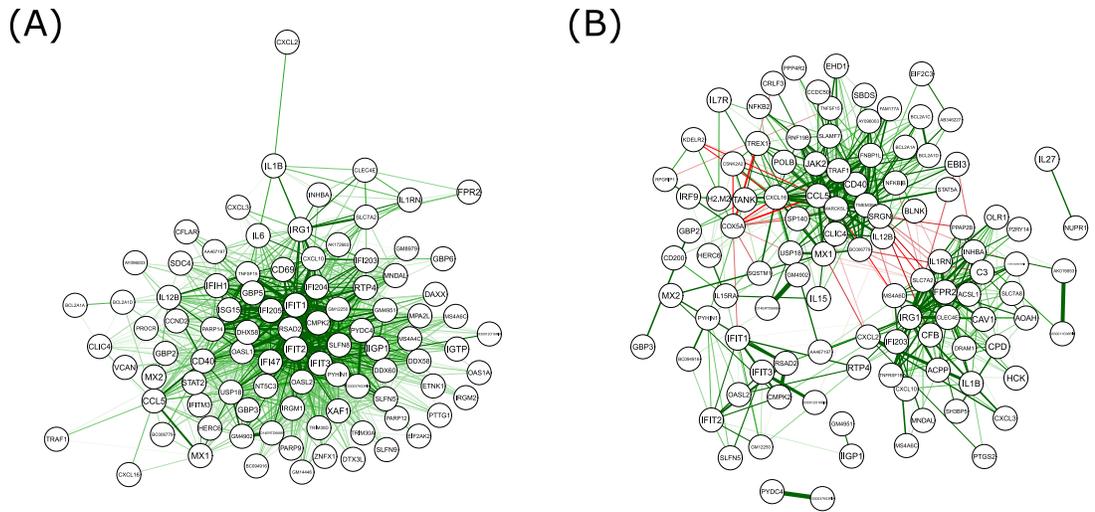


Figure 4.8. The correlation network based on C_{Raw} (A) and C_{Sid} (B) for genes in group 1. There are a total of 93 and 107 genes in the C_{Raw} and C_{Sid} network, respectively. The width of each edge represents the magnitude of an expression correlation between the two genes, and color represents the sign, green for a positive correlation and red for a negative correlation. To improve clarity, correlations with an absolute value lower than 0.55 (0.25) are not shown for C_{Raw} (C_{Sid}) network.

is therefore an important key factor for the immune response. Thus, our correlation method was able to detect a key factor that was difficult to detect by conventional correlation method and is a powerful tool for elucidating gene regulatory networks.

Next, we investigated the correlation network for all genes in group 1 based on both the correlation methods. We omitted the genes with maximum of C_{Raw} (C_{Sid}) values lower than 0.6 (0.3) to improve visibility. Figure 4.8 show the correlation networks based on C_{Raw} (Figure 4.8A) and C_{Sid} (Figure 4.8B). In the C_{Raw} network, the correlations of most of the gene pairs are positive because of spurious correlations over time, and most of the genes are therefore positively connected with each other. In contrast, the C_{Sid} network mainly consists of two clusters, and there are a considerable number of negative correlations between the genes of different clusters. We assumed that each cluster is regulated by distinct regulatory mechanisms and investigated the differences of genes between two clusters. Hereafter, we focus on the chemokine genes ($CXCL2$, $CXCL3$, $CXCL10$, $CXCL16$, and $CCL5$), which are a family of small cytokines or proteins secreted by cells and are known to be involved in immune response [80]. In the C_{Sid} network, $CXCL2$, $CXCL3$, and $CXCL10$ belong to one cluster, while $CXCL16$ and $CCL5$ belong to another cluster. Although $CXCL16$ belongs to

the same CXC gene family, as *CXCL2*, *CXCL3*, and *CXCL10*, it has properties that distinguish it from other CXC chemokine genes. For example, *CXCL2*, *CXCL3*, and *CXCL10* are located in the proximal chromosomal region (5qE2, 5qE2, and 5qE3, respectively), while *CXCL16* is located on another chromosome (11qB4) [81]. Further, although *CCL5* belongs to a different gene family (the CC gene family), *CCL5* is located proximal to *CXCL16* (11qB5). The up-regulation of chemokine genes located in the proximal region has been suggested in breast cancer [82], and our correlation analysis also suggests that chemokine genes in located in the proximal region (*CXCL2*, *CXCL3*, and *CXCL10*) are regulated by different mechanisms than are *CXCL16* and *CCL5*. Thus, each clusters in the C_{Std} network is likely to be regulated by region-dependent mechanisms, and examining correlations among standardized gene expression profiles is a useful approach to elucidate regulatory networks that works by controlling for the effect of trends over time.

4.4 Conclusions

The advancement of single-cell technologies will enable the elucidation of many biological processes, such as differentiation. The development of a novel computational method is necessary to fully analyze single-cell data. We developed a novel method, SCoup, to analyze single-cell expression data for differentiation. Unlike previous methods, which use dimension reduction approaches and reconstruct differentiation trajectories in reduced space, SCoup describes gene expression dynamics during differentiation directly, including pseudo-time and cell fate. We evaluated pseudo-time using SCoup and previous methods based on the consistency between pseudo-time and experimental time and showed that the SCoup results were superior to those of other methods for almost all conditions. We also compared the accuracy of cell lineage estimation using SCoup and Monocle, and showed that SCoup can estimate cell lineages with high accuracy, even for the cells at an early stage of bifurcation. SCoup is based on a probabilistic model and can be extended to many applications. In this research, we developed a novel correlation analysis method based on SCoup. It calculates the covariance that cannot be explained by a model, which assumes the conditional independence among genes, alone. We applied this method to scRNA-seq, and detected the candidate of key regulator of differentiation and the clusters in the correlation network which were not detected with conventional correlation analysis. In future work, we plan to extend our model

to consider transient expression patterns and to estimate complicated cell lineages. In addition, we will develop a multivariate OU process to estimate gene regulatory networks more directly.

4.5 Supplementary text

4.5.1 Limit of a discrete time OU process

In this section, we consider limit of a discrete time OU process. Because both of genes and cells are supposed to be independent in SCoup, we forget the index of gene and cell, and consider a general OU process in this section. We represent observed value as E and initial value as S , and $X = \{X_s | s = 0, \dots, N\}$ is a path such that $X_N = E$ and $X_0 = S$. As mentioned in the main text, a OU process can be regarded as limit of a discrete time OU process:

$$\begin{aligned} P_{\text{ou}}(E|S, \alpha, \sigma^2, t) &= \lim_{N \rightarrow \infty} P_N(X_N|X_0, \alpha, \sigma^2, t) \\ P_N(X_N|X_0, \alpha, \sigma^2, t) &= \int dX \prod_{s=1}^N P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma^2, t/N) \\ P(X|\alpha, \sigma^2, t) &= \prod_{s=1}^N P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma^2, t/N) P(X_0), \end{aligned}$$

where the interval of integration is the all paths which satisfies $X_0 = S$ and $X_N = E$.

Hereafter, we assume X_0 is given and re-define X as $X \in \{X_s | s = 1, \dots, N\}$ for simplification. The calculation of the case that X_0 is unobserved is given in the after section. In this case, the complete likelihood is given by

$$P(X|S, \alpha, \sigma, t) = \prod_{s=1}^N P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma^2, t/N).$$

4.5.1.1 Transformation into the multivariate normal distribution

In this section, we transform the product of the transition probability $P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma^2, t/N)$ as the multivariate normal distribution. In the case of OU process, the transition probability is calculated with the normal distribution as follows:

$$P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma^2, t/N) = \sqrt{\frac{1}{2\pi V}} \exp\left(-\frac{1}{2V}(X_s - BX_{s-1} - (1-B)\theta)^2\right),$$

where

$$V = \frac{\sigma^2(1 - e^{-2\alpha t/N})}{2\alpha}, \quad B = e^{-\alpha t/N}.$$

The complete likelihood $P(X|S, \alpha, \sigma^2, t)$ is equal to the following multivariate normal distribution:

$$\begin{aligned} P(X|S, \alpha, \sigma^2, t) &= \prod_{s=1}^N P_{\text{ou}}(X_s|X_{s-1}, \alpha, \sigma, t/N) \\ &= \frac{\sqrt{|\Lambda|}}{(\sqrt{2\pi})^{N-1}} \exp\left(-\frac{1}{2}(X_{-N} - \mu)^T \Lambda (X_{-N} - \mu)\right) \\ &= \mathcal{N}(X_{-N}|\mu, \Lambda^{-1}), \end{aligned}$$

where $X_{-N} \in \{X_s|s = 1, \dots, N-1\}$ and Λ is $(N-1) \times (N-1)$ matrix and μ is $(N-1)$ dimension vector and satisfy following equations.

$$\Lambda_{i,j} = \begin{cases} \frac{1+B^2}{V} & (i = j) \\ -\frac{B}{V} & (j = i + 1 \text{ or } j = i - 1) \\ 0 & (\text{otherwise}) \end{cases}$$

$$\sum_{j=1}^{N-1} \Lambda_{i,j} \mu_j = \begin{cases} \frac{B}{V} X_0 + \frac{(1-B)^2}{V} \theta & (i = 1) \\ \frac{(1-B)^2}{V} \theta & (1 < i < N - 1) \\ \frac{B}{V} X_N + \frac{(1-B)^2}{V} \theta & (i = N - 1) \end{cases}$$

From above equation, μ_j can be calculated as follows:

$$\mu_j = \frac{B}{V} X_0 \Lambda_{1,j}^{-1} + \frac{(1-B)^2}{V} \theta \sum_{i=1}^{N-1} \Lambda_{i,j}^{-1} + \frac{B}{V} X_N \Lambda_{N-1,j}^{-1}.$$

4.5.1.2 Derivation of mean vector and variance-covariance matrix

As mentioned in the next section, the expectation of X_s , $X_s X_{s+1}$, and X_s^2 are necessary to optimize parameters. These expectations can be calculated from the mean vector and variance-covariance matrix of the multivariate normal distribution. Because we consider a limit of a discrete time OU process, we cannot use numerical calculation and have to solve analytically. In this section, we derivate the mean vector and the variance-covariance matrix.

Firstly, we derivate the variance-covariance matrix. To simplify this, we define Λ' so that

$\Lambda = -V^{-1}B\Lambda'$. We also define following variable:

$$\begin{aligned} -B^{-1} - B &= -(e^{\alpha t/N} + e^{-\alpha t/N}) = -2 \cosh(\lambda) \\ \lambda &= \alpha t/N, \end{aligned}$$

and Λ' is represented as follows:

$$\Lambda'_{i,j} = \begin{cases} -2 \cosh \lambda & (i = j) \\ 1 & (j = i + 1 \text{ or } j = i - 1) \\ 0 & (\text{otherwise}) \end{cases}$$

It is shown that the inversion of symmetric tridiagonal matrix can be calculated analytically [67].

By using this, we can derive the inversion of Λ' and Λ as follows:

$$\begin{aligned} [\Lambda']_{i,j}^{-1} &= -\frac{\cosh(N - |j - i|)\lambda - \cosh(N - i - j)\lambda}{2 \sinh \lambda \sinh N\lambda} \\ [\Lambda]_{i,j}^{-1} &= \frac{V}{B} \frac{\cosh(N - |j - i|)\lambda - \cosh(N - i - j)\lambda}{2 \sinh \lambda \sinh N\lambda}. \end{aligned}$$

Next, we substitute Λ^{-1} and derive the mean μ_j .

$$\mu_j = V^{-1}BX_0\Lambda_{1,j}^{-1} + V^{-1}(1 - B)^2\theta \sum_{i=1}^{N-1} \Lambda_{i,j}^{-1} + V^{-1}BX_N\Lambda_{N-1,j}^{-1}$$

Firstly, we solve the first member $V^{-1}BX_0\Lambda_{1,j}^{-1}$.

$$\begin{aligned} V^{-1}BX_0\Lambda_{1,j}^{-1} &= \frac{\cosh(N - j + 1)\lambda - \cosh(N - j - 1)\lambda}{2 \sinh \lambda \sinh N\lambda} X_0 \\ &= \frac{\sinh(N - j)\lambda}{\sinh N\lambda} X_0 \end{aligned}$$

Secondly, we solve third member $V^{-1}BX_N\Lambda_{N-1,j}^{-1}$.

$$\begin{aligned} V^{-1}BX_N\Lambda_{N-1,j}^{-1} &= \frac{\cosh(j + 1)\lambda - \cosh(-j + 1)\lambda}{2 \sinh \lambda \sinh N\lambda} X_N \\ &= \frac{\sinh j\lambda}{\sinh N\lambda} X_N \end{aligned}$$

Lastly, we solve $\sum_{i=1}^{N-1} \Lambda_{i,j}^{-1}$.

$$\begin{aligned}
& \sum_{i=1}^{N-1} \Lambda_{i,j}^{-1} \\
&= \frac{V}{2B \sinh \lambda \sinh N\lambda} \sum_{i=1}^{N-1} (\cosh(N - |j - i|)\lambda - \cosh(N - i - j)\lambda) \\
&= \frac{V}{2B \sinh \lambda \sinh N\lambda} \left(\sum_{i=1}^j (\cosh(N - j + i)\lambda - \cosh(N - i - j)\lambda) \right. \\
&\quad \left. + \sum_{i=j+1}^{N-1} (\cosh(N + j - i)\lambda - \cosh(N - i - j)\lambda) \right)
\end{aligned}$$

Here, we use following equation to calculate above formula.

$$\begin{aligned}
& \sum_{i=1}^j (\cosh(N - j + i)\lambda - \cosh(N - i - j)\lambda) \\
&= \frac{2}{(1 - e^\lambda)(1 - e^{-\lambda})} (\sinh(N - j)\lambda \sinh j\lambda + \sinh(N - j)\lambda \sinh(j + 1)\lambda + \sinh(N - j)\lambda \sinh \lambda) \\
& \sum_{i=j+1}^{N-1} (\cosh(N + j - i)\lambda - \cosh(N - i - j)\lambda) \\
&= \frac{2}{(1 - e^\lambda)(1 - e^{-\lambda})} (\sinh j\lambda \sinh(N - j - 1)\lambda - \sinh j\lambda \sinh(N - j)\lambda + \sinh j\lambda \sinh \lambda)
\end{aligned}$$

The sum of the above equations becomes

$$\frac{2}{(1 - e^\lambda)(1 - e^{-\lambda})} (\sinh \lambda (\sinh(N - j)\lambda + \sinh j\lambda) + \sinh j\lambda \sinh(N - j - 1)\lambda - \sinh(j + 1)\lambda \sinh(N - j)\lambda)$$

By using following equation,

$$\sinh j\lambda \sinh(N - j - 1)\lambda - \sinh(j + 1)\lambda \sinh(N - j)\lambda = -\sinh \lambda \sinh N\lambda,$$

the sum can be described as follows:

$$\frac{2 \sinh \lambda}{(1 - e^\lambda)(1 - e^{-\lambda})} (\sinh(N - j)\lambda + \sinh j\lambda - \sinh N\lambda).$$

Therefore, second member $V^{-1}(1-B)^2 \frac{V}{2B \sinh \lambda \sinh N\lambda} \theta \sum \Lambda$ becomes

$$\begin{aligned} & \frac{(1-B)^2}{2B \sinh N\lambda} \left(\frac{2 \sinh \lambda}{(1-e^\lambda)(1-e^{-\lambda})} (\sinh(N-j)\lambda + \sinh j\lambda - \sinh N\lambda) \right) \theta \\ &= \frac{1}{\sinh N\lambda} (-\sinh(N-j)\lambda - \sinh j\lambda + \sinh N\lambda) \theta. \end{aligned}$$

Thus, the mean becomes

$$\begin{aligned} \mu_j &= \frac{\sinh(N-j)\lambda}{\sinh N\lambda} X_0 + \frac{\sinh j\lambda}{\sinh N\lambda} X_N + \frac{\theta}{\sinh N\lambda} (-\sinh(N-j)\lambda - \sinh j\lambda + \sinh N\lambda) \\ &= \frac{(X_0 - \theta) \sinh(N-j)\lambda + (X_N - \theta) \sinh j\lambda}{\sinh N\lambda} + \theta. \end{aligned}$$

4.5.1.3 The complete log-likelihood

The complete log-likelihood of X given X_0 is given by

$$\begin{aligned} l(X) &= -\frac{N}{2} \ln \frac{V}{\pi} - \frac{V^{-1}(1+B^2)}{2} \sum_{s=1}^{N-1} X_s^2 + V^{-1}B \sum_{s=0}^{N-1} X_s X_{s+1} + V^{-1}(1-B)^2 \theta \sum_{s=1}^{N-1} X_s \\ &\quad - \frac{V^{-1}B^2}{2} X_0^2 - V^{-1}B(1-B)\theta X_0 - \frac{V^{-1}}{2} X_N^2 + V^{-1}(1-B)\theta X_N - \frac{1}{2} \sum_{s=1}^N V^{-1}(1-B)^2 \theta^2 \\ &= -\frac{N}{2} \ln \frac{\alpha}{\pi \sigma^2 (1-e^{-2\alpha t})} - \frac{N}{2t\sigma^2} \left(2 \left(\sum_{s=1}^{N-1} X_s^2 - \sum_{s=0}^{N-1} X_s X_{s+1} \right) + X_0^2 + X_N^2 \right) \\ &\quad + \frac{\alpha}{2\sigma^2} \left(X_0^2 - X_N^2 - 2\theta X_0 + 2\theta X_N + \frac{\alpha t}{N} \left(-2 \sum_{s=1}^{N-1} X_s^2 + \sum_{s=0}^{N-1} X_s X_{s+1} + 2\theta \sum_{s=1}^{N-1} X_s - N\theta^2 \right) \right) \\ &\quad + \mathcal{O}(1/N), \end{aligned}$$

and Q function is given by

$$\begin{aligned} \mathcal{Q} &= -\frac{N}{2} \ln \frac{V}{\pi} - \frac{V^{-1}(1+B^2)}{2} \sum_{s=1}^{N-1} \langle X_s^2 \rangle + V^{-1}B \sum_{s=0}^{N-1} \langle X_s X_{s+1} \rangle + V^{-1}(1-B)^2 \theta \sum_{s=1}^{N-1} \langle X_s \rangle \\ &\quad - \frac{V^{-1}B^2}{2} X_0^2 - V^{-1}B(1-B)\theta X_0 - \frac{V^{-1}}{2} X_N^2 + V^{-1}(1-B)\theta X_N - \frac{1}{2} N V^{-1}(1-B)^2 \theta^2 \\ &= -\frac{N}{2} \ln \frac{\alpha^*}{\pi \sigma^{*2} (1-e^{-2\alpha^* t})} - \frac{N}{2t^* \sigma^{*2}} (2F_{ss} - 2F_{ss+1} + X_0^2 + X_N^2) \\ &\quad + \frac{\alpha^*}{2\sigma^{*2}} \left(X_0^2 - X_N^2 - 2\theta^* X_0 + 2\theta^* X_N + \frac{\alpha^* t^*}{N} \left(-2F_{ss} + F_{ss+1} + 2F_s - N\theta^{*2} \right) \right) \\ &\quad + \mathcal{O}(1/N), \end{aligned}$$

where F_{ss}, F_{ss+1}, F_s is $\sum_{s=1}^{N-1} \langle X_s^2 \rangle, \sum_{s=0}^{N-1} \langle X_s X_{s+1} \rangle, \sum_{s=1}^{N-1} \langle X_s \rangle$, respectively.

4.5.1.4 Derivation of the sufficient statistic

The likelihood function depends on X only through $X_s^2, X_s X_{s+1}$, and X_s , these are the sufficient statistic for the model. In this section, we derive the these statistic analytically.

4.5.1.5 Derivation of F_{ss}

Firstly, we solve the expectation of X_s^2 .

$$\sum_{s=1}^{N-1} \langle X_s^2 \rangle = \sum_{s=1}^{N-1} \Lambda_{ss}^{-1} + \sum_{s=1}^{N-1} \mu_s^2$$

The first member is

$$\begin{aligned} \sum_{s=1}^{N-1} \Lambda_{ss}^{-1} &= \frac{V}{2B \sinh \lambda \sinh N\lambda} \sum (\cosh N\lambda - \cosh(N-2s)\lambda) \\ &\simeq \frac{\sigma^2}{2\alpha \sinh N\lambda} \left(N \cosh N\lambda - \frac{1}{\lambda} \sinh N\lambda - \lambda \sinh N\lambda \right) \end{aligned}$$

The second member is

$$\begin{aligned} \sum_{s=1}^{N-1} \mu_s^2 &= \sum \left(\frac{(X_0 - \theta) \sinh(N-s)\lambda + (X_N - \theta) \sinh s\lambda}{\sinh N\lambda} + \theta \right)^2 \\ &= \frac{(X_0 - \theta)^2 + (X_N - \theta)^2}{\sinh^2 N\lambda} \sum \sinh^2 s\lambda + \frac{2(X_0 - \theta)(X_N - \theta)}{\sinh^2 N\lambda} \sum \sinh s \sinh(N-s)\lambda \\ &\quad + \frac{2\theta(X_0 + X_N - 2\theta)}{\sinh N\lambda} \sum \sinh s\lambda + (N-1)\theta^2 \end{aligned}$$

4.5.1.6 Derivation of F_{ss+1}

Secondly, we calculate the following equation:

$$\sum_{s=0}^{N-1} \langle X_s X_{s+1} \rangle = \sum_{s=1}^{N-2} \Lambda_{ss+1}^{-1} + \sum_{s=1}^{N-2} \mu_s \mu_{s+1} + \mu_1 X_0 + \mu_{N-1} X_N.$$

The first member is

$$\begin{aligned} \sum_{s=1}^{N-2} \Lambda_{ss+1}^{-1} &= \frac{V}{2B \sinh \lambda \sinh N\lambda} \sum (\cosh(N-1)\lambda - \cosh(N-2s-1)\lambda) \\ &\simeq \frac{\sigma^2}{2\alpha \sinh N\lambda} \left(N \cosh N\lambda - \frac{1}{\lambda} \sinh N\lambda - N\lambda \sinh N\lambda - \frac{1}{2}\lambda \sinh N\lambda + \frac{N\lambda^2}{2} \cosh N\lambda \right). \end{aligned}$$

The remainder is

$$\begin{aligned} &\sum_{s=1}^{N-2} \mu_s \mu_{s+1} + \mu_1 X_0 + \mu_{N-1} X_N \\ &= \sum_{s=0}^{N-1} \left(\frac{(X_0 - \theta) \sinh(N-s)\lambda + (X_N - \theta) \sinh s\lambda}{\sinh N\lambda} + \theta \right) \\ &\quad \times \left(\frac{(X_0 - \theta) \sinh(N-s-1)\lambda + (X_N - \theta) \sinh(s+1)\lambda}{\sinh N\lambda} + \theta \right) \\ &= \frac{(X_0 - \theta)^2 + (X_N - \theta)^2}{\sinh^2 N\lambda} \left(\cosh \lambda \sum_{s=1}^{N-1} \sinh^2 s\lambda + \frac{\sinh \lambda}{2} \sum_{s=1}^{N-1} \sinh 2\lambda \right) \\ &\quad + \frac{(X_0 - \theta)(X_N - \theta)}{\sinh^2 N\lambda} \left(2 \cosh \lambda \sum_{s=1}^{N-1} \sinh s\lambda \sinh(N-s)\lambda + \sinh N\lambda \sinh \lambda \right) \\ &\quad + \frac{\theta(X_0 + X_N - 2\theta)}{\sinh N\lambda} \left(2 \sum_{s=1}^{N-1} \sinh s\lambda + \sinh N\lambda \right) + N\theta^2. \end{aligned}$$

4.5.1.7 Derivation of F_s

Lastly, we calculate F_s .

$$\begin{aligned} \sum_{s=1}^{N-1} \langle X_s \rangle &= \sum \left(\frac{(X_0 - \theta) \sinh(N-s)\lambda + (X_N - \theta) \sinh s\lambda}{\sinh N\lambda} + \theta \right) \\ &= \frac{X_0 + X_N - 2\theta}{\sinh N\lambda} \sum_{s=1}^{N-1} \sinh s\lambda + (N-1)\theta \end{aligned}$$

4.5.2 Derivation of Q function

As mentioned in the above section, Q function is

$$\begin{aligned} \mathcal{Q} &= -\frac{N}{2} \ln \frac{\alpha^*}{\pi \sigma^{*2} (1 - e^{-2\alpha^* t})} - \frac{N}{2t^* \sigma^{*2}} (2F_{ss} - 2F_{ss+1} + X_0^2 + X_N^2) \\ &\quad + \frac{\alpha^*}{2\sigma^{*2}} \left(X_0^2 - X_N^2 - 2\theta^* X_0 + 2\theta^* X_N + \frac{\alpha^* t^{*2}}{N} (-2F_{ss} + F_{ss+1} + 2F_s - N\theta^{*2}) \right) + \mathcal{O}(1/N). \end{aligned}$$

In the following section, we calculate the Q function.

4.5.2.1 Derivation of $2F_{ss} - 2F_{ss+1} + X_0^2 + X_N^2$

Firstly, we calculate the member which is related to $\sum \Lambda_{ss} - \sum \Lambda_{ss+1}$.

$$\begin{aligned} 2 \sum \Lambda_{ss} - 2 \sum \Lambda_{ss+1} &\simeq \frac{\sigma^2}{2\alpha \sinh N\lambda} (2N\lambda \sinh N\lambda - \lambda \sinh N\lambda - N\lambda^2 \cosh N\lambda) \\ &= t\sigma^2 - \lambda \left(\frac{\sigma^2}{2\alpha} + \frac{t\sigma^2 \cosh N\lambda}{2 \sinh N\lambda} \right) \end{aligned}$$

Secondary, we calculate the member which is related to $(X_0 - \theta)^2 + (X_N - \theta)^2$.

$$\begin{aligned} &\frac{1}{\sinh^2 N\lambda} (2 \sum \sinh^2 s\lambda - 2 \cosh \lambda \sum \sinh^2 s\lambda - \sinh \lambda \sum \sinh 2\lambda) \\ &\simeq -1 + \lambda \left(\frac{\cosh N\lambda}{2 \sinh N\lambda} + \frac{N\lambda}{2 \sinh^2 N\lambda} \right) \end{aligned}$$

Thirdly, we calculate the member which is related to $(X_0 - \theta)(X_N - \theta)$.

$$\begin{aligned} &\frac{1}{\sinh^2 N\lambda} \left(4 \sum \sinh s\lambda \sinh(N-s)\lambda - 4 \cosh \lambda \sum \sinh s\lambda \sinh(N-s)\lambda + 2 \sinh N\lambda \sinh \lambda \right) \\ &\simeq -\frac{\lambda}{\sinh^2 N\lambda} (\sinh N\lambda + N\lambda \cosh N\lambda) \end{aligned}$$

Fourthly, we calculate the member which is related to $\theta(X_0 + X_N - 2\theta)$.

$$\frac{1}{\sinh N\lambda} \left(4 \sum \sinh s\lambda - 4 \sum \sinh s\lambda - 2 \sinh N\lambda \right) = -2$$

Lastly, we calculate the member which is related to θ^2 .

$$2(N-1) - 2N = -2$$

With above calculation results, $2F_{ii} - 2F_{ii+1} + X_0^2 + X_N^2$ can be derived as follows:

$$\begin{aligned} &t\sigma^2 - \lambda \left(\frac{\sigma^2}{2\alpha} + \frac{N\lambda\sigma^2 \cosh N\lambda}{2\alpha \sinh N\lambda} \right) - (X_0 - \theta)^2 - (X_N - \theta)^2 - 2\theta(X_0 + X_N - 2\theta) - 2\theta^2 + X_0^2 + X_N^2 \\ &+ \lambda \left(\frac{\cosh N\lambda}{2 \sinh N\lambda} + \frac{N\lambda}{2 \sinh^2 N\lambda} \right) ((X_0 - \theta)^2 + (X_N - \theta)^2) \end{aligned}$$

$$\begin{aligned}
& - \frac{\lambda}{\sinh^2 N\lambda} (\sinh N\lambda + N\lambda \cosh N\lambda) (X_0 - \theta)(X_N - \theta) \\
= & t\sigma^2 - \lambda \left(\frac{\sigma^2}{2\alpha} - \frac{N\lambda\sigma^2 \cosh N\lambda}{2\alpha \sinh N\lambda} \right) \\
& + \lambda \left(\frac{\cosh N\lambda}{2 \sinh N\lambda} + \frac{N\lambda}{2 \sinh^2 N\lambda} \right) ((X_0 - \theta)^2 + (X_N - \theta)^2) \\
& - \frac{\lambda}{\sinh^2 N\lambda} (\sinh N\lambda + N\lambda \cosh N\lambda) (X_0 - \theta)(X_N - \theta).
\end{aligned}$$

4.5.2.2 Derivation of $X_0^2 - X_N^2 - 2\theta^* X_0 + 2\theta^* X_N + \lambda^*(-2F_{ss} + F_{ss+1} + 2\theta^* F_s - N\theta^{*2})$

In this section, we calculate the coefficient of $\frac{\alpha^*}{2\sigma^{*2}}$ which is $X_0^2 - X_N^2 - 2\theta^* X_0 + 2\theta^* X_N + \lambda^*(-2F_{ss} + F_{ss+1} + 2\theta^* F_s - N\theta^{*2})$. In this calculation, we disregard the $1/N$ order terms.

At first, we calculate the member related to F_{ss} and F_{ss+1} . The member related to Λ is calculated as follows:

$$\begin{aligned}
-2\lambda^* \sum \Lambda_{ss} + \lambda^* \sum \Lambda_{ss+1} & \simeq -\frac{\lambda^* \sigma^2}{2\alpha \sinh N\lambda} \left(N \cosh N\lambda - \frac{1}{\lambda} \sinh N\lambda \right) \\
& = \frac{\lambda^*}{\lambda} \left(\frac{\sigma^2}{2\alpha} - \frac{t\sigma^2 \cosh N\lambda}{2 \sinh N\lambda} \right).
\end{aligned}$$

The member related to $(X_0 - \theta)^2 + (X_N - \theta)^2$ is

$$\begin{aligned}
& - \frac{\lambda^*}{\sinh^2 N\lambda} \left(2 \sum \sinh^2 s\lambda - \cosh \lambda \sum \sinh^2 s\lambda - \frac{\sinh \lambda}{2} \sum \sinh 2\lambda \right) \\
& \simeq \frac{\lambda^*}{\lambda} \left(-\frac{\cosh N\lambda}{2 \sinh N\lambda} + \frac{N\lambda}{2 \sinh^2 N\lambda} \right).
\end{aligned}$$

The member related to $(X_0 - \theta)(X_N - \theta)$ is

$$\begin{aligned}
& - \frac{\lambda^*}{\sinh^2 N\lambda} \left(4 \sum \sinh i\lambda \sinh(N-s)\lambda - 2 \cosh \lambda \sum \sinh s\lambda \sinh(N-s)\lambda - \sinh N\lambda \sinh \lambda \right) \\
& \simeq \frac{\lambda^*}{\lambda} \frac{1}{\lambda \sinh^2 N\lambda} (\sinh N\lambda - N\lambda \cosh N\lambda).
\end{aligned}$$

The member related to $\theta(X_0 + X_N - 2\theta)$ is

$$- \frac{\lambda^*}{\sinh N\lambda} \left(4 \sum \sinh s\lambda - 2 \sum \sinh s\lambda - \sinh N\lambda \right) \simeq -2 \frac{\lambda^* \cosh N\lambda - 1}{\lambda \sinh N\lambda}.$$

The member related to $\theta^*(X_0 + X_N - 2\theta)$ is

$$\lambda^* \frac{2}{\sinh N\lambda} \sum \sinh s\lambda \simeq 2 \frac{\lambda^* \cosh N\lambda - 1}{\lambda \sinh N\lambda}.$$

And the member related to $\theta^2, \theta^*\theta, \theta^{*2}$ is

$$\lambda^* \left((-2(N-1) + N)\theta^{*2} + 2(N-1)\theta\theta^* - N\theta^2 \right) \simeq -N\lambda^*(\theta^* - \theta)^2.$$

Therefore, $X_0^2 - X_N^2 - 2\theta^*X_0 + 2\theta^*X_N + \lambda^*(-2F_{ss} + F_{ss+1} + 2\theta^*F_s - N\theta^{*2})$ becomes as follows:

$$\begin{aligned} & (X_0 - \theta^*)^2 - (X_N - \theta^*)^2 + \frac{\lambda^*}{\lambda} \left(-\frac{\cosh N\lambda}{2 \sinh N\lambda} + \frac{N\lambda}{2 \sinh^2 N\lambda} \right) ((X_0 - \theta)^2 + (X_N - \theta)^2) \\ & + \frac{\lambda^*}{\lambda} \frac{1}{\sinh^2 N\lambda} (\sinh N\lambda - N\lambda \cosh N\lambda) (X_0 - \theta)(X_N - \theta) + 2 \frac{\lambda^* \cosh N\lambda - 1}{\lambda \sinh N\lambda} (\theta^* - \theta)(X_0 + X_N - 2\theta) \\ & - N\lambda^*(\theta^* - \theta)^2 + \frac{\lambda^*}{\lambda} \left(\frac{\sigma^2}{2\alpha} - \frac{t\sigma^2 \cosh N\lambda}{2\alpha \sinh N\lambda} \right), \end{aligned}$$

4.5.2.3 Q function

The standardization term is approximated as follows:

$$\frac{2\alpha^*}{1 - e^{-2\alpha^*t/N}} \simeq \frac{N}{t} \frac{1}{1 - \frac{\alpha^*t}{N}} \simeq \frac{N}{t} + \alpha^*.$$

By using the calculation results so far, the Q function is described as follows:

$$\begin{aligned} \mathcal{Q} = & \frac{N}{2} \ln \left(\frac{N}{t^*} + \alpha^* \right) - \frac{N}{2} \ln \sigma^{*2} - \frac{Nt\sigma^2}{2t^*\sigma^{*2}} + \frac{\alpha^*}{2\sigma^{*2}} \left(\frac{\alpha t}{\alpha^*t^*} + \frac{\alpha^*t^*}{\alpha t} \right) \frac{\sigma^2}{2\alpha} + \frac{\alpha^*}{2\sigma^{*2}} \left(\frac{\alpha t}{\alpha^*t^*} - \frac{\alpha^*t^*}{\alpha t} \right) \frac{t\sigma^2 \cosh N\lambda}{2 \sinh N\lambda} \\ & + \frac{\alpha^*}{2\sigma^{*2}} \left(\frac{\alpha t}{\alpha^*t^*} + \frac{\alpha^*t^*}{\alpha t} \right) \left(-\frac{\cosh N\lambda}{2 \sinh N\lambda} ((X_0 - \theta)^2 + (X_N - \theta)^2) + \frac{1}{\sinh N\lambda} (X_0 - \theta)(X_N - \theta) \right) \\ & + \frac{\alpha^*}{2\sigma^{*2}} \left(\frac{\alpha t}{\alpha^*t^*} - \frac{\alpha^*t^*}{\alpha t} \right) \left(-\frac{N\lambda}{2 \sinh^2 N\lambda} ((X_0 - \theta)^2 + (X_N - \theta)^2) + \frac{N\lambda \cosh N\lambda}{\sinh^2 N\lambda} (X_0 - \theta)(X_N - \theta) \right) \\ & + \frac{\alpha^*}{2\sigma^{*2}} \frac{\alpha^*t^*}{\alpha t} (\theta^* - \theta) \left(2 \frac{\cosh N\lambda - 1}{\sinh N\lambda} (X_0 + X_N - 2\theta) \right) - \frac{\alpha^{*2}t^*}{2\sigma^{*2}} (\theta^* - \theta)^2 \\ & + \frac{\alpha^*}{2\sigma^{*2}} ((X_0 - \theta^*)^2 - (X_N - \theta^*)^2) + \mathcal{O}(1/N). \end{aligned}$$

4.5.3 Parameter optimization

We optimize parameters by solving $dQ/d\theta^* = 0$, $dQ/d\alpha^* = 0$, $dQ/d\sigma^{*2} = 0$, and $dQ/dt^* = 0$. In this research, we optimize all parameters independently.

4.5.3.1 Optimization of θ

We solve $dQ/d\theta^* = 0$.

$$dQ/d\theta^* = \frac{\alpha}{2\sigma^2} \left(2 \frac{\cosh \alpha t - 1}{\sinh \alpha t} (X_0 + X_N - 2\theta) - 2\alpha(\theta^* - \theta) - 2(X_0 - X_N) \right)$$

So, θ^* which satisfies $dQ/d\theta^* = 0$ is

$$\begin{aligned} \theta^* &= \theta + \frac{1}{\alpha t} \left(\frac{\cosh \alpha t - 1}{\sinh \alpha t} (X_0 + X_N - 2\theta) - (X_0 - X_N) \right) \\ &= \theta + \frac{2}{\alpha t(1 + e^{-2\alpha t})} \left(X_N - e^{-\alpha t} X_0 - (1 - e^{-\alpha t})\theta \right). \end{aligned}$$

4.5.3.2 Optimization of α

α^* which satisfies $dQ/d\alpha^* = 0$ is

$$\alpha^* = \frac{-\sigma^2 t + (X_N - \theta)^2 - (X_0 - \theta)^2}{Z^\alpha},$$

where

$$\begin{aligned} Z^\alpha &= \frac{\sigma^2}{\alpha} \left(\frac{1}{\alpha} - \frac{t \cosh \alpha t}{\sinh \alpha t} \right) + \frac{2}{\alpha} \left(\frac{\alpha t}{2 \sinh^2 \alpha t} - \frac{\cosh \alpha t}{2 \sinh \alpha t} \right) ((X_0 - \theta)^2 + (X_N - \theta)^2) \\ &\quad + \frac{2}{\alpha} \left(\frac{1}{\sinh \alpha t} - \frac{\alpha t \cosh \alpha t}{\sinh^2 \alpha t} \right) (X_0 - \theta)(X_N - \theta). \end{aligned}$$

4.5.3.3 Optimization of σ^2

We solve $dQ/d\sigma^{*2} = 0$.

$$\begin{aligned} \sigma^{*2} &= \sigma^2 - \frac{1}{N} \left(\sigma^2 + 2\alpha \left(-\frac{\cosh \alpha t}{2 \sinh \alpha t} ((X_0 - \theta)^2 + (X_N - \theta)^2) + \frac{1}{\sinh \alpha t} (X_0 - \theta)(X_N - \theta) \right) \right. \\ &\quad \left. + \alpha((X_0 - \theta)^2 - (X_N - \theta)^2) \right) \end{aligned}$$

The highest order term results in $\sigma^{*2} = \sigma^2$ which means σ^2 will not change. Therefore, we optimize in regards to second highest term.

$$\sigma^2 = \frac{2\alpha}{1 - e^{-2\alpha t}} \left(X_N - e^{-\alpha t} X_0 - (1 - e^{-\alpha t})\theta \right)^2$$

When we regard above σ^2 as σ^{*2} , $dQ/d\sigma^{*2}$ will be zero up to second highest order if σ^2 is converged sufficiently.

4.5.3.4 Optimization of t

We solve $dQ/dt^* = 0$. Because t^* is related to α and σ , we consider $\alpha'^* = t^*\alpha$, $\sigma'^*2 = t^*\sigma^2$.

$$\begin{aligned} \frac{dQ}{d\alpha'^*} &= \frac{1}{2} + \frac{1}{2\alpha t} - \frac{1}{2} \frac{\sinh \alpha t}{\cosh \alpha t} + \frac{1}{t\sigma_g^2} (D^{(1)} - D^{(2)}) + \frac{1}{2t^*\sigma^2} ((X_0 - \theta)^2 - (X_N - \theta)^2) \\ \frac{d\alpha'^*}{dt^*} \frac{dQ}{d\alpha'^*} &= \frac{\alpha}{2} + \frac{1}{2t} - \frac{\alpha \sinh \alpha t}{2 \cosh \alpha t} + \frac{\alpha}{t\sigma^2} (D^{(1)} - D^{(2)}) + \frac{\alpha}{2t^*\sigma^2} ((X_0 - \theta)^2 - (X_N - \theta)^2), \end{aligned}$$

and

$$\begin{aligned} \frac{d\sigma'^*2}{dt^*} \frac{dQ}{d\sigma'^*2} &= \frac{N}{2} \frac{1}{t^{*2}} (t - t^*) - \frac{1}{4t^{*2}} \left(t + \alpha t^2 \frac{\sinh \alpha t}{\cosh \alpha t} + \frac{2\alpha t}{\sigma^2} (D^{(1)} + D^{(2)}) \right) - \frac{\alpha}{2t^*\sigma^2} ((X_0 - \theta)^2 - (X_N - \theta)^2) \\ &\quad - \frac{1}{4t^2} \left(t - \alpha t^2 \frac{\sinh \alpha t}{\cosh \alpha t} + \frac{2\alpha t}{\sigma^2} (D^{(1)} - D^{(2)}) \right), \end{aligned}$$

where

$$\begin{aligned} D^{(1)} &= -\frac{\cosh N\lambda}{2 \sinh N\lambda} ((X_0 - \theta)^2 + (X_N - \theta)^2) + \frac{1}{\sinh N\lambda} (X_0 - \theta)(X_N - \theta) \\ D^{(2)} &= -\frac{N\lambda}{2 \sinh^2 N\lambda} ((X_0 - \theta)^2 + (X_N - \theta)^2) + \frac{N\lambda \cosh N\lambda}{\sinh^2 N\lambda} (X_0 - \theta)(X_N - \theta) \end{aligned}$$

So, dQ/dt^* is described as follows:

$$\begin{aligned} \frac{dQ}{dt^*} &= \frac{d\alpha'^*}{dt^*} \frac{dQ}{d\alpha'^*} + \frac{d\sigma'^*2}{dt^*} \frac{dQ}{d\sigma'^*2} \\ &= \frac{N}{2} \frac{1}{t^{*2}} (t - t^*) \\ &\quad - \frac{1}{4t^{*2}} \left(t + \alpha t^2 \frac{\sinh \alpha t}{\cosh \alpha t} + \frac{2\alpha t}{\sigma^2} (D^{(1)} + D^{(2)}) \right) \\ &\quad - \frac{1}{4t^2} \left(-t + \alpha t^2 \left(-2 + \frac{\sinh \alpha t}{\cosh \alpha t} \right) + \frac{2\alpha t}{\sigma^2} (-D^{(1)} + D^{(2)}) \right). \end{aligned}$$

The highest order term results in $t^* = t$ and we consider second highest term. In the case that $dQ/dt^* = 0$ for second highest order, the following equation consists.

$$t + \alpha t^2 \frac{\sinh \alpha t}{\cosh \alpha t} + \frac{2\alpha t}{\sigma^2} (D^{(1)} + D^{(2)}) = t - \alpha t^2 \left(-2 + \frac{\sinh \alpha t}{\cosh \alpha t} \right) + \frac{2\alpha t}{\sigma^2} (D^{(1)} - D^{(2)})$$

Above equation becomes as follows:

$$\alpha t \left(1 - \frac{\cosh \alpha t}{\sinh \alpha t} \right) - 2 \frac{\alpha}{\sigma^2} E = 0.$$

Unfortunately, t cannot be solved analytically and we use Newton's method.

$$\begin{aligned} f(t) &= \alpha t \left(1 - \frac{\cosh \alpha t}{\sinh \alpha t} \right) - 2 \frac{\alpha}{\sigma^2} D^{(2)} \\ \frac{df(t)}{dt} &= \alpha \left(1 - \frac{\cosh \alpha t}{\sinh \alpha t} \right) + \frac{\alpha^2 t}{\sinh^2 \alpha t} - 2 \frac{\alpha^2}{\sigma^2} \frac{1}{\sinh^3 \alpha t} \left(-\frac{1}{2} (\sinh \alpha t - 2\alpha t \cosh \alpha t) ((X_0 - \theta)^2 + (X_N - \theta)^2) \right. \\ &\quad \left. + (\cosh \alpha t \sinh \alpha t - \alpha t (1 + \cosh^2 \alpha t)) (X_0 - \theta) (X_N - \theta) \right) \\ t_{n+1} &= t_n - \frac{f(t_n)}{f'(t_n)} \end{aligned}$$

We optimize t iteratively by using above equation.

4.5.4 Mixture OU process for multi-lineage differentiation

We denote the number of cell, gene, and lineage by C , G , and K , respectively. We also denote the index of cell, gene, and lineage by c , g , and k , respectively. We assume each lineage has different attractor θ_{gk} and the likelihood is given by

$$\begin{aligned} P(E|S, \Theta, T) &= \prod_{c=1}^C \prod_{g=1}^G P(E_{gc} | S_{gc}, \theta_g, t_c) \\ &= \prod_{c=1}^C \left(\sum_{k=1}^K \pi_k \prod_{g=1}^G P(E_{gc} | S_{gc}, \theta_{gk}, t_c) \right) \\ &= \prod_{c=1}^C \left(\sum_{k=1}^K \pi_k \prod_{g=1}^G \sum_{X_{gc} \in (E_{gc}, S_{gc})} \prod_{s=1}^N P(X_{gcs} | X_{gcs-1}, \theta_{gk}, t_c/N) \right), \end{aligned}$$

where π_k is the probability of lineage k .

With the latent value Z_c which is 1 of K representation and indicates the cell fate,

$P(X, Z|E, S, \Theta, T)$ and $P(Z|E, S, \Theta, T)$ are given by

$$P(X, Z|E, S, \Theta, T) \propto \prod_{c=1}^C \prod_{k=1}^K \left(\pi_k^{Z_{ck}} \prod_{g=1}^G \prod_{s=1}^N P(X_{gcs} | X_{gcs-1}, \theta_{gk}, t_c/N)^{Z_{ck}} \right)$$

$$P(Z|E, S, \Theta, T) \propto \prod_{c=1}^C \prod_{k=1}^K \left(\pi_k^{Z_{ck}} \prod_{g=1}^G P(E_{gc} | S_{gc}, \theta_{gk}, t_c)^{Z_{ck}} \right).$$

So, γ_{ck} , which is the expectation of posterior probability of Z_{ck} is represented as follows:

$$\gamma_{ck} = \mathbf{E}[Z_{ck}] = \frac{\pi_k \prod_{g=1}^G P(E_{gc} | S_{gc}, \theta_{gk}, t_c)}{\sum_{k'} \pi_{k'} \prod_{g=1}^G P(E_{gc} | S_{gc}, \theta_{gk'}, t_c)}.$$

To avoid overfitting, we added pseudo-count and re-defined γ_{ck} as follows:

$$\gamma_{ck} := \frac{\gamma_{ck} + 0.01}{\sum_{k'} (\gamma_{ck'} + 0.01)} = \frac{\gamma_{ck} + 0.01}{1 + 0.01 \times K}.$$

Hereafter, we denote $\ln P(X_{cg} | S_{cg}, Z_{ck} = 1)$ by l_{gck} , and l_{gck} is described as follows:

$$l_{gck} = \ln \left(\prod_{i=1}^N P(X_{gcs} | X_{gcs-1}, \theta_{gk}, t_c/N) \right)$$

$$= -\frac{N}{2} \ln \frac{V_g}{\pi} - \frac{V_g^{-1}(1+B_g^2)}{2} \sum_{s=1}^{N-1} X_{gcs}^2 + V_g^{-1} B_g \sum_{s=1}^{N-2} X_{gcs} X_{gcs+1} + V_g^{-1} (1-B_g)^2 \theta_{gk} \sum_{s=1}^{N-1} X_{gcs}$$

$$+ V_g^{-1} B_g X_{gc0} X_{gc1} + V_g^{-1} B_g X_{gcN} X_{gcN-1} - \frac{V_g^{-1} B_g^2}{2} X_{gc0}^2 - V_g^{-1} B_g (1-B_g) \theta_{gk} X_{gc0}$$

$$- \frac{V_g^{-1}}{2} X_{gcN}^2 + V_g^{-1} (1-B_g) \theta_{gk} X_{gcN} - \frac{1}{2} \sum_{i=1}^N V_g^{-1} (1-B_g)^2 \theta_{gk}^2.$$

And the Q function of l_{gck} (\mathcal{Q}_{gck}) is

$$\mathcal{Q}_{gck} = \frac{N}{2} \ln \left(\frac{N}{t_c^*} + \alpha_g^* \right) - \frac{N}{2} \ln \sigma_g^{*2} - \frac{N t \sigma_g^2}{2 t_c^* \sigma_g^{*2}}$$

$$+ \frac{\alpha_g^*}{2 \sigma_g^{*2}} \left(\frac{\alpha_g t_c}{\alpha_g^* t_c^*} + \frac{\alpha_g^* t_c^*}{\alpha_g t_c} \right) \frac{\sigma_g^2}{2 \alpha_g} + \frac{\alpha_g^*}{2 \sigma_g^{*2}} \left(\frac{\alpha_g t_c}{\alpha_g^* t_c^*} - \frac{\alpha_g^* t_c^*}{\alpha_g t_c} \right) \frac{t_c \sigma_g^2 \cosh N \lambda_g}{2 \sinh N \lambda_g}$$

$$+ \frac{\alpha_g^*}{2 \sigma_g^{*2}} \left(\frac{\alpha_g t_c}{\alpha_g^* t_c^*} + \frac{\alpha_g^* t_c^*}{\alpha_g t_c} \right) D_{gck}^{(1)} + \frac{\alpha_g^*}{2 \sigma_g^{*2}} \left(\frac{\alpha_g t_c}{\alpha_g^* t_c^*} - \frac{\alpha_g^* t_c^*}{\alpha_g t_c} \right) D_{gck}^{(2)}$$

$$+ \frac{\alpha_g^*}{2 \sigma_g^{*2}} \frac{\alpha_g^* t_c^*}{\alpha_g t_c} (\theta_{gk}^* - \theta_{gk}) \left(2 \frac{\cosh N \lambda_{gc} - 1}{\sinh N \lambda_{gc}} (X_{gc0} + X_{gcN} - 2 \theta_{gk}) \right)$$

$$-\frac{\alpha_g^{*2} t_c^*}{2\sigma_g^{*2}} (\theta_{gk}^* - \theta_{gk})^2 + \frac{\alpha_g^*}{2\sigma_g^{*2}} ((X_{gc0} - \theta_{gk}^*)^2 - (X_{gcN} - \theta_{gk}^*)^2) + \mathcal{O}(1/N),$$

where

$$D_{gck}^{(1)} = -\frac{\cosh N\lambda_{gc}}{2 \sinh N\lambda_{gc}} ((X_{gc0} - \theta_{gk})^2 + (X_{gcN} - \theta_{gk})^2) + \frac{1}{\sinh N\lambda_{gc}} (X_{gc0} - \theta_{gk})(X_{gcN} - \theta_{gk})$$

$$D_{gck}^{(2)} = -\frac{N\lambda_{gc}}{2 \sinh^2 N\lambda_{gc}} ((X_{gc0} - \theta_{gk})^2 + (X_{gcN} - \theta_{gk})^2) + \frac{N\lambda_{gc} \cosh N\lambda_{gc}}{\sinh^2 N\lambda_{gc}} (X_{gc0} - \theta_{gk})(X_{gcN} - \theta_{gk}).$$

Thus, the complete Q function is

$$\mathcal{Q} = E_{Z,X} [\ln P(X, Z, E|S, \Theta, T)] = \sum_c \sum_k \left(\gamma_{ck} \ln \pi_k + \sum_g \gamma_{ck} \mathcal{Q}_{gck} \right) + \mathcal{O}(1/N).$$

4.5.4.1 Parameter optimization

The optimization equation is derived by solving the differentiation of the Q function likewise the parameter optimization of single gene, cell, and lineage model.

4.5.4.2 Optimization of θ_{gk}

$$\theta_{gk}^* = \theta_{gk} + \frac{\sum_c \gamma_{ck} \frac{2}{\alpha_g(1+e^{-\alpha_g t_c})} (X_{gkN} - e^{-\alpha_g t_c} X_{gk0} - (1 - e^{-\alpha_g t_c}) \theta_{gk})}{\sum_c \gamma_{ck} t_c}$$

4.5.4.3 Optimization of α_g

$$\alpha_g^* = \frac{\sum_c \sum_k \gamma_{ck} \left(-t_c \sigma_g^2 - (X_{gc0} - \theta_{gk})^2 + (X_{gcN} - \theta_{gk})^2 \right)}{\left(\sum_c \sum_k \gamma_{ck} Z_{cgk}^\alpha \right)}$$

4.5.4.4 Optimization of σ_g^2

$$\sigma_g^{*2} = \frac{1}{C} \sum_c \sum_k \gamma_{gck} \frac{2\alpha_g}{1 - e^{-2\alpha_g t_c}} \left(X_{gkN} - e^{-\alpha_g t_c} X_{gk0} - (1 - e^{-\alpha_g t_c}) \theta_{gk} \right)^2$$

4.5.4.5 Optimization of t_c

We optimize t_c so that it satisfies following equation.

$$\sum_k \sum_g \gamma_{ck} \left(\alpha_g t_c \left(1 - \frac{\cosh \alpha_g t_c}{\sinh \alpha_g t_c} \right) - 2 \frac{\alpha_g}{\sigma_g^2} D_{gck}^{(2)} \right) = 0$$

We used Newton's method as follows:

$$\begin{aligned} f(t_c) &= \sum_k \sum_g \gamma_{ck} \left(\alpha_g t_c \left(1 - \frac{\cosh \alpha_g t_c}{\sinh \alpha_g t_c} \right) - 2 \frac{\alpha_g}{\sigma_g^2} D_{gck}^{(2)} \right) \\ \frac{df(t_c)}{dt_c} &= \sum_k \sum_g \gamma_{ck} \left(\alpha_g \left(1 - \frac{\cosh \alpha_g t_c}{\sinh \alpha_g t_c} \right) + \frac{\alpha_g^2 t_c}{\sinh^2 \alpha_g t_c} \right. \\ &\quad \left. - 2 \frac{\alpha_g^2}{\sigma_g^2} \frac{1}{\sinh^3 \alpha_g t_c} \left(-\frac{1}{2} (\sinh \alpha_g t_c - 2\alpha_g t_c \cosh \alpha_g t_c) ((X_{cg0} - \theta_{gk})^2 + (X_{cgN} - \theta_{gk})^2) \right. \right. \\ &\quad \left. \left. + (\cosh \alpha_g t_c \sinh \alpha_g t_c - \alpha_g t_c (1 + \cosh^2 \alpha_g t_c)) (X_{cg0} - \theta_{gk})(X_{cgN} - \theta_{gk}) \right) \right) \\ t_{cn+1} &= t_{cn} - \frac{f(t_{cn})}{f'(t_{cn})}. \end{aligned}$$

4.5.4.6 Optimization of π_k

$$\pi_k = \frac{\sum_c \gamma_{ck}}{\sum_c \sum_{k'} \gamma_{ck'}}$$

4.5.5 Expected value of S_{cg}

Thus far, we assume $S_{cg}(X_{cg0})$ is given. However, it is unobserved practically and we have to calculate expected value of S_{gc} .

$$\begin{aligned} P(S_{cg}|E_{cg}) &= \frac{P(E_{cg}|S_{cg})P(S_{cg})}{P(E_{cg})} \\ &\propto \mathcal{N}(E_{cg} | e^{-\alpha t} S_{cg} + (1 - e^{-\alpha_g t_c}) \theta_{gk}, V_{gc}) \mathcal{N}(S_{cg} | \mu_{0g}, \sigma_{0g}^2) \\ &\propto \mathcal{N}\left(S_{cg} \left| \frac{V'^{-1} \mu' + \sigma_{0g}^{-2} \mu_{0g}}{V'^{-1} + \sigma_{0g}^{-2}}, \frac{1}{V'^{-1} + \sigma_{0g}^{-2}} \right.\right), \end{aligned}$$

where

$$\begin{aligned}\mu' &= e^{\alpha_g t_c} E_{cg} + (1 - e^{\alpha_g t_c}) \theta_{gk} \\ V' &= e^{2\alpha_g t_c} V_{gc} \\ V_{gc} &= \frac{\sigma_g^2 (1 - e^{-2\alpha_g t_c})}{2\alpha_g}.\end{aligned}$$

Therefore, the expected values related to X_{cg0} are given by

$$\begin{aligned}E[X_{cg0}^2] &= \left(\frac{V'^{-1} \mu' + \sigma_{0g}^{-2} \mu_{0g}}{V'^{-1} + \sigma_{0g}^{-2}} \right)^2 + \frac{1}{V'^{-1} + \sigma_{0g}^{-2}} \\ E[X_{cg0}] &= \frac{V'^{-1} \mu' + \sigma_{0g}^{-2} \mu_{0g}}{V'^{-1} + \sigma_{0g}^{-2}},\end{aligned}$$

and we just substitute the above $E[X_{cg0}]$ and $E[X_{cg0}^2]$ into X_{cg0} and X_{cg0}^2 , respectively, for parameter optimization.

4.5.6 The marginal log-likelihood

S_{gc} is not observed and we have to marginalize over S_{gc} to calculate the marginal log-likelihood, and it is described as follows:

$$\begin{aligned}\int dS_{gc} N(E_{gc} | e^{-\alpha t} S_{gc} + (1 - e^{-\alpha t}) \theta_{gk}, V_g) N(S_{gc} | \mu_{0g}, \sigma_{0g}^2) \\ = N(E_{gc} | e^{-\alpha t} \mu_{0g} + (1 - e^{-\alpha t}) \theta_{gk}, V + e^{-2\alpha t} \sigma_{0g}^2).\end{aligned}$$

Therefore, the log-likelihood of E is

$$l(E) = \sum_c \log \left(\sum_k \pi_k \prod_g N(E_{cg} | e^{-\alpha t} \mu_{0g} + (1 - e^{-\alpha_g t_c}) \theta_{gk}, V_g + e^{-2\alpha_g t_c} \sigma_{0g}^2) \right).$$

4.5.7 A procedure of parameter optimization

In this research, we used following parameter optimization procedure to avoid sub-optimal solutions.

Firstly, we initialized pseudo-time t_c based on dimension reduction approach and $\alpha_g, \sigma_g^2, \theta_g$ by

using the model of $K = 1$.

- 1) Add the mean of initial distribution ($\mu_0 = \{\mu_{0g}|g = 0, \dots, G\}$) as a root to single-cell expression data, and perform principal component analysis.
- 2) Calculate minimum spanning tree with Prim's algorithm on D dimensional latent space and calculate the shortest path from root (μ_0) to a cell c and define the standardized of the total weight of the shortest path as the initial value of t_c (In this research, we set $D = 2$ unless we refer).
- 3) Set the initial value of θ_g to the mean of the expression ($\theta_g = \frac{1}{C} \sum_{c=1}^C E_{gc}$).
- 4) Initialize α_g, σ_g^2 (In this research, we set $\alpha_g = 5.0, \sigma_g^2 = 1.0$).
- 5) Optimize $\alpha_g, \sigma_g^2, \theta_g$ with SCOUP of $K = 1$ by EM algorithm with 10 iterations.

Secondary, we optimize parameter of mixture model. Because parameters fell into a sub-optimal solution which shows wrong order of cells when we optimized all parameter simultaneously, we optimized parameters except t_c at first.

- 1) Initialize θ_{gk} with θ_g calculated by above procedure.
- 2) Initialize the expectation of a latent value of a cell (γ_{ck}) randomly and calculate other statistic, and optimize α_g, σ_g^2 , and θ_{gk} with M-step.
- 3) Run E-step to calculate γ_{ck} and other statistic.
- 4) Run M-step to optimize α_g, σ_g^2 , and θ_{gk} .
- 5) Return to step 3 until the number of iterations reach m_1 (In this research, we set $m_1 = 1,000$).

Lastly, we optimize all parameters.

- 1) Run E-step
- 2) Run M-step. We optimize t_c at first, and optimized other parameters after that.

- 3) Stop the parameter optimization if $|e^{-\alpha_g^* t_{\max}} \theta_{gk}^* - e^{-\alpha_g t_{\max}} \theta_{gk}|$, $|\frac{\sigma_g^{*2}(1-e^{-2\alpha_g^* t_{\max}})}{2\alpha_g^*} - \frac{\sigma_g^2(1-e^{-2\alpha_g t_{\max}})}{2\alpha_g}|$, and $|t_c^* - t_c|$ are under ϵ (In this research, we set $\epsilon = 0.01$). We used these values to check convergence because these are meaningful values and α_g and σ_g^2 can change together so that the likelihood does not change (see next section).
- 4) Stop the parameter optimization if the number of iterations reach m_2 (In this research, we set $m_2 = 10,000$ and verified that parameters are converged before m_2 iterations under most conditions).
- 5) Return to step 1.

α_g can be very large and small which is meaningless to estimate accurately and we set lower and upper bounds to $\alpha_{\min} = 0.1, \alpha_{\max} = 100$ and set $\alpha_g = \alpha_{\min}$ (α_{\max}) if α_g^* is under (over) α_{\min} (α_{\max}). σ_g^2 can be significantly small and we set the lower bound similar way ($\sigma_{\min}^2 = 0.1$). We also set the bounds of pseudo-time so that the lower bound is $t_{\min} = 0.001$ and the upper bound is $t_{\max} = 2.0$. For pseudo-time (t_c) optimization, we stop the Newton's method if $|t_{cn+1} - t_{cn}|$ is lower than ϵ or the number of iteration reach 100. The solution of Newton's method t_c can be incorrect value and we set the new parameter of t_c to the time whose likelihoods are highest in the three case: t_c is old time; t_c is the solution of Newton's method; $t_c = t_{\min}$; and $t_c = t_{\max}$.

4.5.8 Validation of parameter optimization method

In this section, we validated the parameter optimization method with simulation data. We generated simulation data with $C = 100, G = 500$, and $K = 1$, and each parameter was sampled so that $t_c = U_R[0, 1]$, $\alpha_g = U_R[0.1, 10]$, $\sigma_g^2 = U_R[0.1, 100]$, and $\theta_g = U_R[-5, 5]$, where $U_R[a, b]$ is a uniform random number from a to b. All of the initial distributions of the gene were $N(X_{cg0}|0, 1.0)$ and X_{cg0} was sampled from the distribution. To complete the scale of the parameters, we set $t_{\max} = 1.0$. With above conditions, we sampled the expression data from the normal distribution based on OU process, and applied this simulation data for SCoup.

Firstly, we compared the values of estimated parameters with those of true parameters (Figure 4.9(A,B,C,D)). The values of estimated time is highly correlated with true values ($r^2 = 0.94$). The difference between estimated time and true time becomes large for large t_c . This is because the distribution of OU-process becomes stationary distribution for t sufficiently large, and the fluc-

tuation of the value of optimized t_c will be large for such condition. The values of estimated θ_g is also highly correlated with true values ($r^2 = 0.96$). But estimated θ_g of a few genes are different from true values significantly. This will be because the influence of θ_g on the distribution is significantly small when $\alpha_g \simeq 0$, and hence, the value of θ_g is unstable in such condition. The values of estimated α_g and σ_g^2 are highly correlated with true values (r^2 is 0.79 and 0.77, respectively), but estimated α_g and σ_g^2 of some genes are significantly larger than true values. This is because that the distribution of different α_g and σ_g^2 will be almost equal for the gene of $\theta_g \simeq \mu_{0g}$ as long as the balance between α_g and σ_g^2 are kept, and the estimated absolute values will be unstable. Then, we investigated the value of mean ($e^{-\alpha_g t} \theta_g$) and variance ($\sigma_g^2(1 - e^{-2\alpha_g t})/2\alpha_g$) of OU process at time $t = 1$ (Figure 4.9(E,F)). The values of estimated mean and variance are highly correlated with those of true mean and variance (0.99 and 0.94, respectively), and hence, SCOUP succeed to reconstruct the original probabilistic distribution with high accuracy.

Next, we investigated that the log-likelihood of optimized parameters is higher than those of varied parameters. Figure 4.10 is the example of the log-likelihood curve with respect to time parameter of a cell (t_c), and the value of optimized t_c is drawn with x-mark. The log-likelihood of the optimized t_c is located in the top of the log-likelihood curve. As shown in Figure 4.10(C), the log-likelihood are almost equal $0.5 < t_c$ because the distribution will almost be equal to the stationary distribution for large t_c . Figure 4.11 is the example of the log-likelihood surface with respect to α_g and θ_g of a gene. The log-likelihood of the optimized α_g and θ_g are located in the top of the log-likelihood surface. Figure 4.12 is also the example of the log-likelihood surface with respect to σ_g^2 and θ_g of a gene. The log-likelihood of the optimized σ_g^2 and θ_g are located in the top of the log-likelihood surface. Figure 4.13 is also the example of the log-likelihood surface with respect to θ_g and σ_g^2 of a gene. Although the log-likelihood of the optimized α_g and σ_g^2 are located in the top of the log-likelihood surface, the log-likelihood are almost equal for $\sigma_g^2 \simeq 20 \times \alpha_g$ regarding a gene (Figure 4.13(C)). This is because that the distribution of different α_g and σ_g^2 will be almost equal in some conditions as mentioned in the above paragraph.

Although the values of optimized parameters have potential to be unstable in some conditions, the mean and variance of OU process are stable and the log-likelihood of optimized parameters are highest. Therefore, we conclude that SCOUP succeed to optimize parameters.

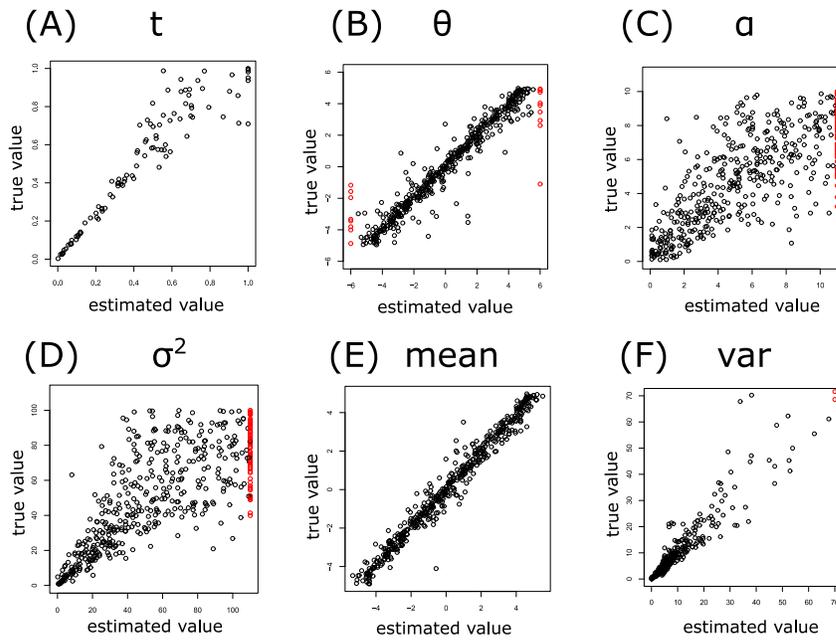


Figure 4.9. Comparison between the estimated values and true values: (A) for pseudo-time (t), (B) for θ_g , (C) for α_g , (D) for σ_g^2 , (E) for mean, (F) for variance. The outlier whose estimated value exceeds the boundary of drawing area is visualized in the border with a red circle for visualization.

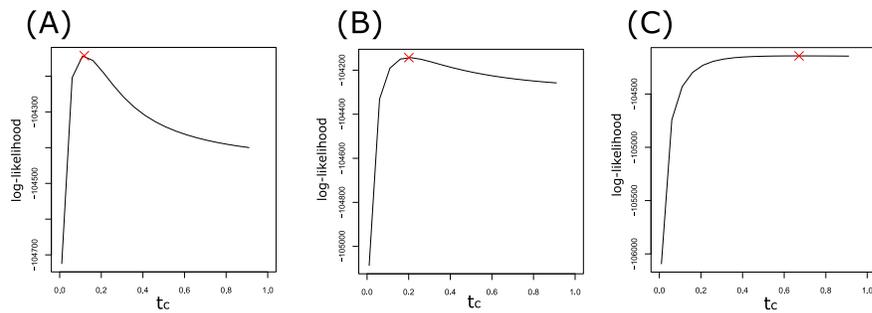


Figure 4.10. The log-likelihood curve with respect to t_c of a cell. The optimized t_c is indicated with x-max.

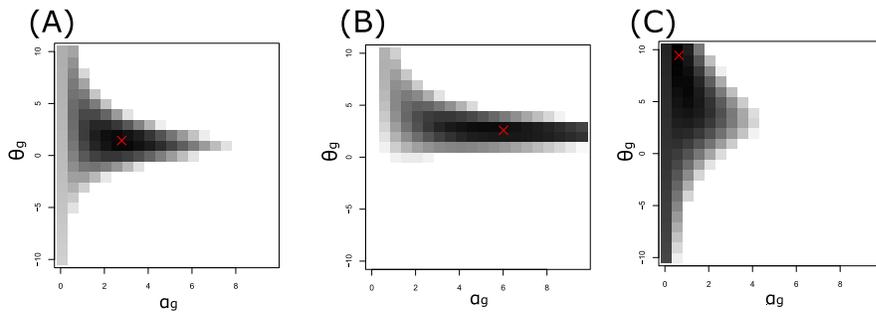


Figure 4.11. The log-likelihood surface with respect to α_g and θ_g of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (α_g, θ_g) is indicated with x-max.

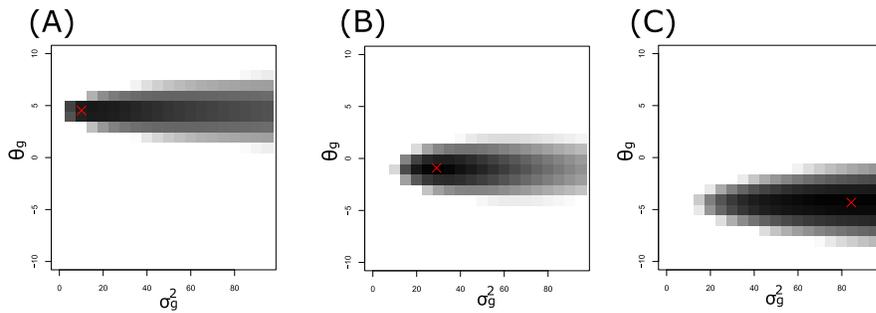


Figure 4.12. The log-likelihood surface with respect to σ_g^2 and θ_g of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (σ_g^2, θ_g) is indicated with x-max.

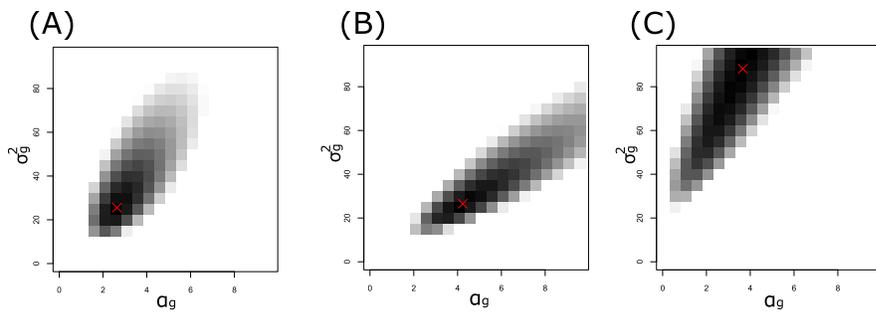


Figure 4.13. The log-likelihood surface with respect to α_g and σ_g^2 of a gene. The color of a pixel represents the log-likelihood and black represents the highest log-likelihood. The optimized (α_g, σ_g^2) is indicated with x-max.

4.5.9 Cell lineage estimation with Gaussian mixture model

We estimated cell lineage with Gaussian mixture model (GMM) implemented in `mclust` package [71]. The AUC values of `mclust` for Kouno's data(2) ($\epsilon = 0.0$) and Moignard's data are 0.86 and 0.96, respectively, and both of them are inferior to those of SCOUP (0.99 and 1.0). Figure 4.14 and 4.15 show cells of Kouno's data and Moignard's data in the space of first two PCs and the colors of cells indicate the genuine cell lineage (left), the lineage estimated using SCOUP (middle), and the lineage using `mclust` (right). GMM cannot estimate cell lineage correctly for cells at an early stage of bifurcation because GMM does not count the time axis and will work well only for cells whose expression pattern changes sufficiently after bifurcation. Moreover, `mclust` seems to overfit the 4SG cells around $(-6, 0)$ in the PCA space, and failed to distinguish a portion of 4SG cells for Moignard's data (Figure 4.15). This is because GMM will fit to the position in which large number of cells exist, and GMM cannot estimate the path of bifurcation in the condition that cells are unevenly distributed. Therefore, it is important to count time axis to analyze expression for cells during bifurcation.

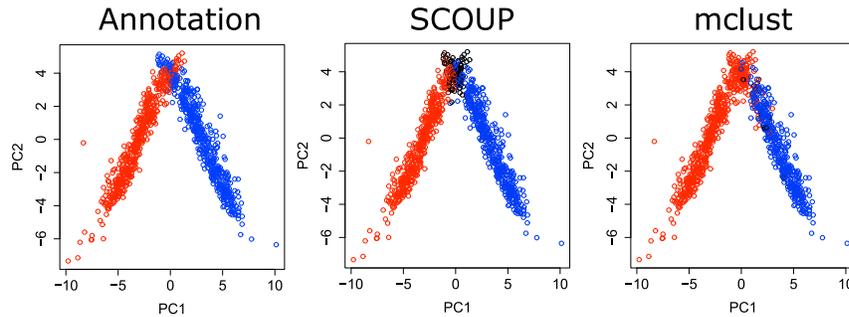


Figure 4.14. PCA of cells of Kouno's data based on gene expression. The cell colors indicate the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with `mclust` (right). The color for SCOUP is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for `mclust` is defined by expectation of latent values; black, 0.5; red, 0.0; and blue, 1.0. The color of each state is consistent among plots.

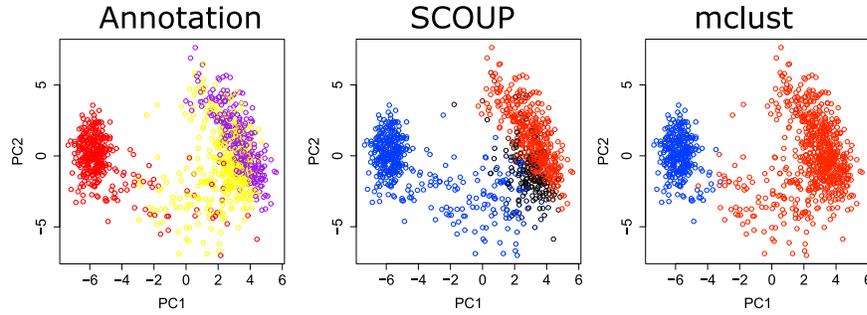


Figure 4.15. PCA of cells of Moignard’s data based on gene expression. The cell colors represent the genuine lineage (left), lineage estimated with SCOUP (middle), and lineage estimated with mclust (right). The color for the genuine lineage is defined by the annotation of the cell; yellow, HF; red, 4SG; and purple, 4SFG⁻. The color for the SCOUP analysis is defined by γ_{c0} ; black, 0.5; red, 0.0; and blue, 1.0. The color for the mclust analysis is defined by expectation of latent values; black, 0.5; red, 0.0; and blue, 1.0. We determined the color of each state so that they are consistent among each plot.

4.5.10 Annotated pairs in the top 1,000 C_{Raw} and C_{Std} values

As described in the main manuscript, we investigated whether the target genes of a transcription factor (TF) can be predicted under the assumption that the expression of a TF and its target genes are highly correlated. The top 1,000 C_{Raw} and C_{Std} values contained correlations of There are 24 and 27 annotated pairs in the top 1,000 C_{Raw} and C_{Std} values, respectively (Table 4.6). Only three annotated pairs (UHRF1, RRM2), (MCM5, RRM2), and (MCM4, RRM2), were common between the 24 C_{Raw} correlation values and the 27 C_{Std} correlation values.

Table 4.6. The annotated pairs in the top 1,000 C_{Raw} and C_{Std} values. The left and right tables correspond to C_{Raw} and C_{Std} , respectively. The first column of each table contains the TF names and the second column lists the target gene names. The third column contains the C_{Raw} or C_{Std} of the pairs.

TF	target gene	C_{Raw}	TF	target gene	C_{Std}
IFIT1	RTP4	0.761	UHRF1	RRM2	0.666
IFIT1	IFI47	0.760	MCM5	RRM2	0.644
IFIT1	OASL2	0.746	MCM4	RRM2	0.557
IFI205	IFI47	0.702	MCM5	CDCA8	0.489
IRF7	IFI47	0.699	ATAD2	RRM2	0.486
IRF7	OASL2	0.694	MCM3	RRM2	0.482
UHRF1	RRM2	0.681	UHRF1	CDCA8	0.480
IFI205	RTP4	0.681	CENPA	TOP2A	0.476
IFIT3	MPA2L	0.681	MCM5	TOP2A	0.470
IFIT1	USP18	0.680	MCM3	2810417H13RIK	0.464
IFI205	PYHIN1	0.658	HMGB2	TOP2A	0.447
IFI203	RTP4	0.655	MCM3	TOP2A	0.447
MCM5	RRM2	0.653	PLD4	FCGR2B	0.445
IRF7	USP18	0.651	MCM5	MAD2L1	0.437
PARP14	IFI47	0.649	H2AFZ	2810417H13RIK	0.434
IFIH1	IFI47	0.645	MCM4	TOP2A	0.432
IFIH1	RTP4	0.641	ATAD2	2810417H13RIK	0.422
IFIH1	OASL2	0.637	MCM4	CDCA8	0.421
IFIT1	IGTP	0.619	MCM5	DNAJC9	0.421
IFI205	MPA2L	0.605	ATAD2	TOP2A	0.413
IRF7	RTP4	0.593	MCM5	DTYMK	0.408
IFI205	GBP2	0.592	MCM4	DTYMK	0.408
PARP9	USP18	0.587	H2AFZ	TOP2A	0.404
MCM4	RRM2	0.584	MCM4	ANP32E	0.402
			MCM4	LBR	0.402
			UHRF1	CKS1B	0.387
			MCM3	ANP32E	0.383

Bibliography

- [1] The ENCODE Project Consortium: An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**(7414), 57–74 (2012)
- [2] Marusyk, A., Almendro, V., Polyak, K.: Intra-tumour heterogeneity: a looking glass for cancer? *Nat. Rev. Cancer* **12**(5), 323–334 (2012)
- [3] Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J., Pachter, L.: Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* **28**(5), 511–515 (2010)
- [4] Meinicke, P., Asshauer, K.P., Lingner, T.: Mixture models for analysis of the taxonomic composition of metagenomes. *Bioinformatics* **27**(12), 1618–1624 (2011)
- [5] Schaid, D.J.: Evaluating associations of haplotypes with traits. *Genet. Epidemiol.* **27**, 348–364 (2004)
- [6] The International HapMap Consortium: A second generation human haplotype map of over 3.1 million SNPs. *Nature* **449**, 851–861 (2007)
- [7] Tewhey, R., Bansal, V., Torkamani, A., Topol, E.J., Schork, N.J.: The importance of phase information for human genomics. *Nat. Rev. Genet.* **12**, 215–223 (2011)
- [8] Clark, A.G.: Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol.* **7**, 111–122 (1990)
- [9] Excoffier, L., Slatkin, M.: Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.* **12**, 921–927 (1995)

- [10] Stephens, M., Smith, N.J., Donnelly, P.: A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.* **68**, 978–989 (2001)
- [11] Stephens, M., Donnelly, P.: A comparison of bayesian methods for haplotype reconstruction from population genotype data. *Am. J. Hum. Genet.* **73**, 1162–1169 (2003)
- [12] Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R.: MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet. Epidemiol.* **34**(8), 816–834 (2010)
- [13] Browning, S.R., Browning, B.L.: Haplotype phasing: existing methods and new developments. *Nat. Rev. Genet.* **12**, 703–714 (2011)
- [14] Bansal, V., Halpern, A.L., Axelrod, N., Bafna, V.: An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Res.* **18**, 1336–1346 (2008)
- [15] Bansal, V., Bafna, V.: HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics* **24**, 153–159 (2008)
- [16] Chen, Z., Fu, B., Schweller, R., Yang, B., Zhao, Z., Zhu, B.: Linear time probabilistic algorithms for the singular haplotype reconstruction problem from SNP fragments. *J. Comput. Biol.* **15**, 535–546 (2008)
- [17] Duitama, J., McEwen, G.K., Huebsch, T., Palczewski, S., Schulz, S., Verstrepen, K., Suk, E.K., Hoehe, M.R.: Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of Single Individual Haplotyping techniques. *Nucleic Acids Res.* **40**, 2041–2053 (2012)
- [18] Kim, J.H., Waterman, M.S., Li, L.M.: Diploid genome reconstruction of *Ciona intestinalis* and comparative analysis with *Ciona savignyi*. *Genome Res.* **17**, 1101–1110 (2007)
- [19] Levy, S., *et al.*: The diploid genome sequence of an individual human. *PLoS Biol.* **5**, 254 (2007)
- [20] Li, L.M., Kim, J.H., Waterman, M.S.: Haplotype reconstruction from SNP alignment. *J. Comput. Biol.* **11**, 505–516 (2004)

- [21] Panconesi, A., Sozio, M.: Fast Hare: a fast heuristic for single individual SNP haplotype reconstruction. In: WABI'04, pp. 266–277 (2004)
- [22] The 1000 Genomes Project Consortium: An integrated map of genetic variation from 1,092 human genomes. *Nature* **491**(7422), 56–65 (2012)
- [23] Eid, J., *et al.*: Real-time DNA sequencing from single polymerase molecules. *Science* **323**, 133–138 (2009)
- [24] Kitzman, J.O., Mackenzie, A.P., Adey, A., Hiatt, J.B., Patwardhan, R.P., Sudmant, P.H., Ng, S.B., Alkan, C., Qiu, R., Eichler, E.E., Shendure, J.: Haplotype-resolved genome sequencing of a Gujarati Indian individual. *Nat. Biotechnol.* **29**, 59–63 (2011)
- [25] Suk, E.K., McEwen, G.K., Duitama, J., Nowick, K., Schulz, S., Palczewski, S., Schreiber, S., Holloway, D.T., McLaughlin, S., Peckham, H., Lee, C., Huebsch, T., Hoehe, M.R.: A comprehensively molecular haplotype-resolved genome of a European individual. *Genome Res.* **21**, 1672–1685 (2011)
- [26] Coop, G., Wen, X., Ober, C., Pritchard, J.K., Przeworski, M.: High-resolution mapping of crossovers reveals extensive variation in fine-scale recombination patterns among humans. *Science* **319**(5868), 1395–1398 (2008)
- [27] Dewal, N., Hu, Y., Freedman, M.L., Laframboise, T., Pe'er, I.: Calling amplified haplotypes in next generation tumor sequence data. *Genome Res.* **22**(2), 362–374 (2012)
- [28] Kitzman, J.O., Snyder, M.W., Ventura, M., Lewis, A.P., Qiu, R., Simmons, L.E., Gammill, H.S., Rubens, C.E., Santillan, D.A., Murray, J.C., Tabor, H.K., Bamshad, M.J., Eichler, E.E., Shendure, J.: Noninvasive whole-genome sequencing of a human fetus. *Sci Transl Med* **4**(137), 137–76 (2012)
- [29] Gabriel, S.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E.S., Daly, M.J., Altshuler, D.: The structure of haplotype blocks in the human genome. *Science* **296**(5576), 2225–2229 (2002)

- [30] Zhang, K., Deng, M., Chen, T., Waterman, M.S., Sun, F.: A dynamic programming algorithm for haplotype block partitioning. *Proc. Natl. Acad. Sci. U.S.A.* **99**(11), 7335–7339 (2002)
- [31] Anderson, E.C., Novembre, J.: Finding haplotype block boundaries by using the minimum-description-length principle. *Am. J. Hum. Genet.* **73**(2), 336–354 (2003)
- [32] Karp, R.M.: Reducibility among combinatorial problems. In: *Complexity of Computer Computation*, pp. 85–103 (1972)
- [33] Attias, H.: Inferring parameters and structure of latent variable models by variational Bayes. In: *UAI'99*, pp. 21–30 (1999)
- [34] Zhi, D., Wu, J., Liu, N., Zhang, K.: Genotype calling from next-generation sequencing data using haplotype information of reads. *Bioinformatics* **28**(7), 938–946 (2012)
- [35] Geraci, F.: A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem. *Bioinformatics* **26**, 2217–2225 (2010)
- [36] He, D., Choi, A., Pipatsrisawat, K., Darwiche, A., Eskin, E.: Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics* **26**, 183–190 (2010)
- [37] Lo, C., Bashir, A., Bansal, V., Bafna, V.: Strobe sequence design for haplotype assembly. *BMC Bioinformatics* **12 Suppl 1**, 24 (2011)
- [38] The International HapMap 3 Consortium: Integrating common and rare genetic variation in diverse human populations. *Nature* **467**(7311), 52–58 (2010)
- [39] Ozaki, K., Ohnishi, Y., Iida, A., Sekine, A., Yamada, R., Tsunoda, T., Sato, H., Sato, H., Hori, M., Nakamura, Y., Tanaka, T.: Functional SNPs in the lymphotoxin-alpha gene that are associated with susceptibility to myocardial infarction. *Nat. Genet.* **32**(4), 650–654 (2002)
- [40] Pritchard, J.K., Stephens, M., Donnelly, P.: Inference of population structure using multilocus genotype data. *Genetics* **155**(2), 945–959 (2000)
- [41] Stranger, B.E., Forrest, M.S., Dunning, M., Ingle, C.E., Beazley, C., Thorne, N., Redon, R., Bird, C.P., de Grassi, A., Lee, C., Tyler-Smith, C., Carter, N., Scherer, S.W., Tavare, S., De-

- loukas, P., Hurles, M.E., Dermitzakis, E.T.: Relative impact of nucleotide and copy number variation on gene expression phenotypes. *Science* **315**(5813), 848–853 (2007)
- [42] Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., Lander, E.S.: High-resolution haplotype structure in the human genome. *Nat. Genet.* **29**(2), 229–232 (2001)
- [43] Matsumoto, H., Kiryu, H.: MixSIH: a mixture model for single individual haplotyping. *BMC Genomics* **14 Suppl 2**, 5 (2013)
- [44] Peters, B.A., Kermani, B.G., Sparks, A.B., Alferov, O., Hong, P., Alexeev, A., Jiang, Y., Dahl, F., Tang, Y.T., Haas, J., Robasky, K., Zaranek, A.W., Lee, J.H., Ball, M.P., Peterson, J.E., Perazich, H., Yeung, G., Liu, J., Chen, L., Kennemer, M.I., Pothuraju, K., Konvicka, K., Tsoumpko-Sitnikov, M., Pant, K.P., Ebert, J.C., Nilsen, G.B., Baccash, J., Halpern, A.L., Church, G.M., Drmanac, R.: Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells. *Nature* **487**(7406), 190–195 (2012)
- [45] Kaper, F., Swamy, S., Klotzle, B., Munchel, S., Cottrell, J., Bibikova, M., Chuang, H.Y., Kruglyak, S., Ronaghi, M., Eberle, M.A., Fan, J.B.: Whole-genome haplotyping by dilution, amplification, and sequencing. *Proc. Natl. Acad. Sci. U.S.A.* **110**(14), 5552–5557 (2013)
- [46] He, D., Han, B., Eskin, E.: Hap-seq: an optimal algorithm for haplotype phasing with imputation using sequencing data. *J. Comput. Biol.* **20**(2), 80–92 (2013)
- [47] He, D., Eskin, E.: Hap-seqX: expedite algorithm for haplotype phasing with imputation using sequence data. *Gene* **518**(1), 2–6 (2013)
- [48] Yang, W.Y., Hormozdiari, F., Wang, Z., He, D., Pasaniuc, B., Eskin, E.: Leveraging reads that span multiple single nucleotide polymorphisms for haplotype inference from sequencing data. *Bioinformatics* **29**(18), 2245–2252 (2013)
- [49] Kolodziejczyk, A.A., Kim, J.K., Svensson, V., Marioni, J.C., Teichmann, S.A.: The technology and biology of single-cell RNA sequencing. *Mol. Cell* **58**(4), 610–620 (2015)
- [50] Buettner, F., Natarajan, K.N., Casale, F.P., Proserpio, V., Scialdone, A., Theis, F.J., Teichmann, S.A., Marioni, J.C., Stegle, O.: Computational analysis of cell-to-cell heterogeneity in single-

- cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.* **33**(2), 155–160 (2015)
- [51] Grun, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., van Oudenaarden, A.: Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* **525**(7568), 251–255 (2015)
- [52] Zeisel, A., Munoz-Manchado, A.B., Codeluppi, S., Lonnerberg, P., La Manno, G., Jureus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., Rolny, C., Castelo-Branco, G., Hjerling-Leffler, J., Linnarsson, S.: Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**(6226), 1138–1142 (2015)
- [53] Treutlein, B., Brownfield, D.G., Wu, A.R., Neff, N.F., Mantalas, G.L., Espinoza, F.H., Desai, T.J., Krasnow, M.A., Quake, S.R.: Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature* **509**(7500), 371–375 (2014)
- [54] Burns, J.C., Kelly, M.C., Hoa, M., Morell, R.J., Kelley, M.W.: Single-cell RNA-Seq resolves cellular complexity in sensory organs from the neonatal inner ear. *Nat Commun* **6**, 8557 (2015)
- [55] Xue, Z., Huang, K., Cai, C., Cai, L., Jiang, C.Y., Feng, Y., Liu, Z., Zeng, Q., Cheng, L., Sun, Y.E., Liu, J.Y., Horvath, S., Fan, G.: Genetic programs in human and mouse early embryos revealed by single-cell RNA sequencing. *Nature* **500**(7464), 593–597 (2013)
- [56] Yan, L., Yang, M., Guo, H., Yang, L., Wu, J., Li, R., Liu, P., Lian, Y., Zheng, X., Yan, J., Huang, J., Li, M., Wu, X., Wen, L., Lao, K., Li, R., Qiao, J., Tang, F.: Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. Struct. Mol. Biol.* **20**(9), 1131–1139 (2013)
- [57] Guo, G., Huss, M., Tong, G.Q., Wang, C., Li Sun, L., Clarke, N.D., Robson, P.: Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Dev. Cell* **18**(4), 675–685 (2010)
- [58] Moignard, V., Gottgens, B.: Transcriptional mechanisms of cell fate decisions revealed by single cell expression profiling. *Bioessays* **36**(4), 419–426 (2014)

- [59] Trapnell, C.: Defining cell types and states with single-cell genomics. *Genome Res.* **25**(10), 1491–1498 (2015)
- [60] Semrau, S., van Oudenaarden, A.: Studying Lineage Decision-Making In Vitro: Emerging Concepts and Novel Tools. *Annu. Rev. Cell Dev. Biol.* **31**, 317–345 (2015)
- [61] Stegle, O., Teichmann, S.A., Marioni, J.C.: Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.* **16**(3), 133–145 (2015)
- [62] Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N.J., Livak, K.J., Mikkelsen, T.S., Rinn, J.L.: The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32**(4), 381–386 (2014)
- [63] Marco, E., Karp, R.L., Guo, G., Robson, P., Hart, A.H., Trippa, L., Yuan, G.C.: Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proc. Natl. Acad. Sci. U.S.A.* **111**(52), 5643–5650 (2014)
- [64] Ji, Z., Ji, H.: TSCAN: Tools for Single-Cell ANalysis (2015). R package version 1.8.0
- [65] Cressler, C.E., Butler, M.A., King, A.A.: Detecting Adaptive Evolution in Phylogenetic Comparative Analysis Using the Ornstein-Uhlenbeck Model. *Syst. Biol.* **64**(6), 953–968 (2015)
- [66] Kiryu, H.: Sufficient statistics and expectation maximization algorithms in phylogenetic tree models. *Bioinformatics* **27**(17), 2346–2353 (2011)
- [67] Hu, G.Y., O’Connell, R.F.: Analytical inversion of symmetric tridiagonal matrices. *J.Phys.A* **29**(7), 1511–1513 (1996)
- [68] Kouno, T., de Hoon, M., Mar, J.C., Tomaru, Y., Kawano, M., Carninci, P., Suzuki, H., Hayashizaki, Y., Shin, J.W.: Temporal dynamics and transcriptional control using single-cell gene expression analysis. *Genome Biol.* **14**(10), 118 (2013)
- [69] Moignard, V., Woodhouse, S., Haghverdi, L., Lilly, A.J., Tanaka, Y., Wilkinson, A.C., Buetner, F., Macaulay, I.C., Jawaid, W., Diamanti, E., Nishikawa, S., Piterman, N., Kouskoff, V., Theis, F.J., Fisher, J., Gottgens, B.: Decoding the regulatory network of early blood development from single-cell gene expression measurements. *Nat. Biotechnol.* **33**(3), 269–276 (2015)

- [70] Shalek, A.K., Satija, R., Shuga, J., Trombetta, J.J., Gennert, D., Lu, D., Chen, P., Gertner, R.S., Gaublot, J.T., Yosef, N., Schwartz, S., Fowler, B., Weaver, S., Wang, J., Wang, X., Ding, R., Raychowdhury, R., Friedman, N., Hacohen, N., Park, H., May, A.P., Regev, A.: Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510**(7505), 363–369 (2014)
- [71] Fraley, C., Raftery, A.E., Murphy, T.B., Scrucca, L.: *mclust* Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. Technical Report **597** (2012)
- [72] Huang, d.a.W., Sherman, B.T., Lempicki, R.A.: Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat Protoc* **4**(1), 44–57 (2009)
- [73] Huang, d.a.W., Sherman, B.T., Lempicki, R.A.: Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* **37**(1), 1–13 (2009)
- [74] Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A.K., Slichter, C.K., Miller, H.W., McElrath, M.J., Prlic, M., Linsley, P.S., Gottardo, R.: MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16**, 278 (2015)
- [75] Watts, C., West, M.A., Zaru, R.: TLR signalling regulated antigen presentation in dendritic cells. *Curr. Opin. Immunol.* **22**(1), 124–130 (2010)
- [76] Zheng, G., Tu, K., Yang, Q., Xiong, Y., Wei, C., Xie, L., Zhu, Y., Li, Y.: ITFP: an integrated platform of mammalian transcription factors. *Bioinformatics* **24**(20), 2416–2417 (2008)
- [77] Kanamori, M., Konno, H., Osato, N., Kawai, J., Hayashizaki, Y., Suzuki, H.: A genome-wide and nonredundant mouse transcription factor database. *Biochem. Biophys. Res. Commun.* **322**(3), 787–793 (2004)
- [78] Lizio, M., Harshbarger, J., Shimoji, H., Severin, J., Kasukawa, T., Sahin, S., Abugessaisa, I., Fukuda, S., Hori, F., Ishikawa-Kato, S., Mungall, C.J., Arner, E., Baillie, J.K., Bertin, N., Bono, H., de Hoon, M., Diehl, A.D., Dimont, E., Freeman, T.C., Fujieda, K., Hide, W.,

- Kaliyaperumal, R., Katayama, T., Lassmann, T., Meehan, T.F., Nishikata, K., Ono, H., Rehli, M., Sandelin, A., Schultes, E.A., 't Hoen, P.A., Tatum, Z., Thompson, M., Toyoda, T., Wright, D.W., Daub, C.O., Itoh, M., Carninci, P., Hayashizaki, Y., Forrest, A.R., Kawaji, H.: Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biol.* **16**, 22 (2015)
- [79] Yu, H.B., Kielczewska, A., Rozek, A., Takenaka, S., Li, Y., Thorson, L., Hancock, R.E., Guarna, M.M., North, J.R., Foster, L.J., Donini, O., Finlay, B.B.: Sequestosome-1/p62 is the key intracellular target of innate defense regulator peptide. *J. Biol. Chem.* **284**(52), 36007–36011 (2009)
- [80] Esche, C., Stellato, C., Beck, L.A.: Chemokines: key players in innate and adaptive immunity. *J. Invest. Dermatol.* **125**(4), 615–628 (2005)
- [81] Zlotnik, A., Yoshie, O., Nomiya, H.: The chemokine and chemokine receptor superfamilies and their molecular evolution. *Genome Biol.* **7**(12), 243 (2006)
- [82] Bieche, I., Chavey, C., Andrieu, C., Busson, M., Vacher, S., Le Corre, L., Guinebretiere, J.M., Burlinon, S., Lidereau, R., Lazennec, G.: CXC chemokines located in the 4q21 region are up-regulated in breast cancer. *Endocr. Relat. Cancer* **14**(4), 1039–1052 (2007)