Computational studies in epigenomics using histone modification data

THÈSE N^{O} 6327 (2014)

PRÉSENTÉE LE 28 AOÛT 2014
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE BIOLOGIE COMPUTATIONNELLE ET BIOINFORMATIQUE
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nishanth Ulhas NAIR

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. B. Moret, Dr Ph. Bucher, directeurs de thèse
Prof. S. Hannenhalli, rapporteur
Prof. F. Naef, rapporteur
Prof. W. Stafford Noble, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2014

गुरुर्ब्रह्मा गुरुर्विष्णुर्गुरुर्देवो महेश्वरः ।
gururbrahmā gururviṣṇurgururdevo maheśvaraḥ |
गुरुरेव परब्रह्म तस्मै श्रीगुरवे नमः ॥
gurureva parabrahma tasmai śrīgurave namaḥ ||

— verse 32 in Sri Guru Gita (authored by sage Vyasa)

To my wonderful mother and father.
To my beloved wife.
To my teachers, advisors, mentors.
To the great Masters, to whom I can but humbly bow.

Acknowledgements

To say I have had a very blessed student life would be an understatement. Although as long as one is in the world of academics one is always a student, I understand that with the completion of my PhD, I may not get an other opportunity to be a formal student of science and technology again. There are a countless people who I am deeply indebted to throughout my life without who I would not have such a wonderful student life. I would like to thank all of them and name a few below.

First and foremost I would like to thank my PhD advisor Prof. Bernard M.E. Moret for making me a part of the Laboratory of Computational Biology and Bioinformatics (LCBB) lab. Bernard is one of the best advisors anyone can get and I am very lucky to have him as my advisor. I don't think any advisor gives more freedom to his/her students than Bernard does. Bernard gave me a lot of freedom and at the same time guided me well throughout my PhD with a lot of compassion as well as patience. Many times I found him more as a friend than an advisor and I had many non-technical discussions with him too... especially about politics, governance, and society. I am grateful to Bernard for making me a part of this lab even though I had a non-CS and non-Biology background. In fact Bernard was kind enough to come all the way to India just to attend my wedding! Thank you Bernard for all that you have done for me.

Most students are lucky to have one good advisor... but I had two! Dr. Philipp Bucher is my other PhD advisor and he has been a truly amazing. He is a person who is full of ideas and every time I met him I learned something new. He has been very patiently and compassionately helping me throughout the duration of my PhD. Thank you Philipp for everything.

The Guru-Shishya parampara (teacher-disciple tradition) is well known in India. The word "Gu" means darkness while the word "Ru" means dispeller — dispelling darkness. Guru is thus the one who removes your ignorance.

My mother Mrs. Vimala V.N. Pillai — who I call Amma — is my first Guru. Amma's infinite compassion, her love and dedication towards my well-being and education is primarily responsible for me to complete my PhD. She has almost super-human levels of tolerance towards the most adverse situations and her dedication and sincerity to everyone she meets is a constant source of inspiration for me. She is also one of the most intelligent and insightful people that I know and I constantly look up to her for advice and protection. I don't think I could have asked for a better mother than her. I feel extremely blessed to be her son.

My father Mr. D. Ullas — who I call Achha — has been very loving and compassionate towards me. Achha has got an extraordinary energy and is full of good humour. He has struggled a lot to give me a good education and life and has been very supportive of my career decisions.

With immense gratitude I thank Amma and Achhan for all that they have done for me and for being my parents.

My brother Nitin has always been my well-wisher. We used to play a lot together when we were kids, and had a lot of fun doing so. He has been constantly supportive of me, and been my well-wisher... and I thank him very much for it.

I am also deeply grateful to my grandmother Mrs. Janaky Amma (whom I fondly call Ammooma) for her blessings and love towards me.

I would also like to thank my other relatives who have helped and advised me and my family a lot along the way. In particular I would like to express my deep love and affection to my cousins Ashwin Dinesh (fondly called Achhu) and Nandana.

Ever since my childhood, when it came to my education, things somehow by magic seemed to work, as if the Universe was aligning itself in a way such that I get a fine education. I was very lucky to get into a great school like St. Joseph's Boys' High School in Bangalore. It is one of the oldest and finest schools in the country. My parents struggled a lot and put me into this school although it was beyond their means. A person named Shaji whom I have never met played a role in getting me admission into this school. I thank him for that.

Studying at St. Joseph's Boys' High School gave me an excellent exposure to the opportunities outside. I had a very good time in the school and learnt a lot from my teachers like Mrs. Sandhya Raman, Mr. Uday Kumar, Dr. David Chatterjee and others. One of my seniors in school, the extremely bright Dr. Sriram Sankararaman has been a constant source of inspiration for me, and it was partly because of him that I chose to pick computational biology as my area of research.

I was also very fortunate to study at BASE which is a very fine coaching institute for IIT-JEE exams. Although I did not make it into the IITs, I learnt a lot by going there and my love for physics, organic chemistry, and mathematics developed quite a bit during my time there. The teachers at BASE and the founder of the institute Dr. H.S. Nagaraja (often called HSN) were people of great dedication and sincerity.

I would like to thank my B.E. friends — Mushaffa, Manjunath, Karthik, Srinivas for their friendship during my undergraduate days.

Probably the finest moment of my academic life came when I got admission into Indian Institute of Science (IISc), Bangalore for my Masters degree. IISc is one of the oldest and probably the finest graduate level research university in India, and I was very lucky to study at the Electrical Communication Engineering department which is one of the best departments at IISc.

I had a great advisor in Prof. T.V. Sreenivas during my study at IISc. TVS sir was not just a great advisor technically, he also taught me a lot about life. We used have wonderful philosophical discussions and I am extremely grateful to him for everything that he has done for me.

At IISc, I also had an opportunity to meet some excellent friends and met some amazing people. To name a few, I would like to thank Sreeram, Basavaraj, Sudha, Anoop and his wife Anusha, Dinesh, Arun, Manasa, Nadir, Narayana, Venkat, Saikat, Mrugesh, Sriram etc. to name a few. I am very grateful to Mrugesh for getting me interested in yoga and spirituality by giving me the book “The Power of Now” written by Eckhart Tolle, which is the first book on spirituality that I read.

I got an opportunity to work in a wonderful place like Microsoft Research India (MSRI) for a year as soon as I finished my Masters degree at IISc. I would like to thank my advisor Dr. Navin Goyal for his support. I also thank Kalika Bali, Dr. Amitava Das, and Prof. Nagasuma R. Chandra who advised me during my stay there. I also got some good friends — like Cohan, Santhosh, Meena, Sneha. Through my interaction with some of them I also got to meet Lakshmi Dinesh who turned out be an excellent friend over all these years.

I remember longing to do a PhD ever since my childhood days and I was extremely delighted to get an admission into a world class institute like EPFL. EPFL is a wonderful place for research and study. I have had a very beautiful time during my stay in Switzerland. I was lucky to be a part of the Laboratory of Computational Biology and Bioinformatics (LCBB), which was an excellent place to be, thanks to Bernard and all my colleagues there. I would like to thank lab mates Avinash, Xiuwei, Wei, Yu, Vaibhav, Mingfu, Cristina, Min, Yann, Krister, Slavica, Olga, Ana, Jelena, Paulina, Laura, Hermina, Aleksandra, Alisa for all their friendship and support.

I was lucky to collaborate with some exceptionally bright people during my PhD days, other than my advisors. My collaboration started with Avinash Das Sahu, who is one of the brightest and smartest person that I have met ; and I doubt if my PhD would have happened without his support and immense help. I also had good collaborations with Dr. Yu Lin, Dr. Sunil Kumar, Ana Manasovska, Paulina Grnarova, Jelena Antic, Prof. Judy A Brusslan, Prof. Matteo Pellegrini, Dr. Vaibhav Rajan, Laura Hunter, Mingfu Shao. I thank them all.

I had an opportunity to mentor a few extremely bright students — Ana, Paulina, Jelena, and Laura. I thank all of them for giving me this opportunity. In fact I am proud to say that Ana, Paulina, Jelena got their first publication through their collaboration with me.

I am grateful to Prof. Sridhar Hannenhalli, Prof. William Stafford Noble, Prof. Emre Telatar, Prof. Felix Naef for being a part of my PhD thesis jury committee. Along with my two advisors Bernard and Philipp, they carefully read my thesis and giving me valuable feedback. I would like to thank them for that.

My life in Switzerland would not have been so great if it wasn’t for some amazing friends I had. In specific I would like to thank my best friends Dr. Abhishek Tewari, Avinash Das Sahu, and Dr. Rammohan Narendula who I would like to keep in touch throughout my life. Abhishek

has been my closest friend throughout my PhD days. He had joined at the same time as I did, and he stayed in the opposite door studio in the same building that I lived during most of my PhD days. We cooked together on most days and we had excellent discussions on Indian politics, society, philosophy, spirituality etc. In fact Abhishek, Ram, and I used to take active interest in Indian politics and participate in several ways. My life here would have been boring if not for Abhishek's genuine friendship and compassion. I am deeply indebted to him for that. Avinash is an incredible person who lives a very interesting life. He is truly brilliant and very good at many things. Other than helping me a lot of research and academic subjects, I also learnt a lot about Indian society and culture, politics and many other things from him. He also helped a lot during any personal difficulty I had. I thank him for that. Ram became my very good friend during my last 2-3 years of PhD. Our "tea with Ram" sessions everyday used to be a delightful time, when many of us would meet for tea in Ram's lab library and discuss many things — especially Indian politics. Ram was always known for getting amazingly cheap offers and deals. He is an avid traveler, and I traveled a lot with him around Switzerland.

I had many other wonderful friends at EPFL. Rajasunder Chandran (Raja) has been a great friend who is deeply passionate about what he does, and he is very sincere, innocent, and compassionate person. I also enjoyed the friendship of Rahul VV who helped me quite a bit. Some of the other friends I would like to thank are Suri, Saket, Vandana, Megha, Pankaj, Rishikesh, Arnab, Aviinash, Krishnan, Devika among many others.

I am grateful to Devika for starting the Gita self-study group and it is something I have immensely enjoyed and learnt a lot over the last few years. It was truly amazing that few of us — mainly Devika, Ram, Abhishek, Raja, Giovanni, Krishnan among many others met once a week almost every week for the last two years or so for an hour of Gita class. We studied the Bhagavad Gita Home Study (BGHS) course written by Swami Dayananda Saraswati. So far we finished more than 800 pages of a two-volume book which is around 1800 pages. Giovanni, a Christian priest who I met during the class, has been a great friend. Giovanni is also a scholar on many Indian texts and I learnt a lot from him. He organized some wonderful travels to Florence and Rome, and I learnt a lot from him during these travels.

EPFL and Switzerland has offered a lot to me. Other than good education and research facilities, EPFL offers a very good PhD stipend which helped me and my family a lot financially. Coming to Switzerland was very good for me health wise too — I did not suffer from the allergies which I often suffered from in India. I cannot but be awed my the beauty of Switzerland — the gorgeous mountains and lakes. I am equally awed by the countless Swiss engineers and workers who made the roads, tunnels, railways in Switzerland to make these mountains so accessible for us. In fact Amma who is currently in Switzerland visiting me remarked "God and man has strived in equal capacity to make Switzerland what it is" (this is an approximate translation of her remark made in Malayalam).

I made some great friends via Facebook like Vinay Vaidya and Dafne (both of whom I have never met), and I learnt a lot from my interactions with them. My dear friend Dr. Sreeram Kannan played a very important role in my life, especially as a spiritual guide. He introduced

me to the “Inner Engineering” program designed by Sadhguru. I thank him very much for everything.

During my PhD, I had an opportunity to attend a semester long program “Mathematical and Computational Approaches in High Throughput Genomics,” organized by Institute of Pure and Applied Mathematics (IPAM), University of California, Los Angeles. I would like to thank IPAM for organizing such a great program. I had a great time in Los Angeles, and an opportunity to collaborate with wonderful researchers like Prof. Matteo Pellegrini. I also got some good friends like Dr. Greg Ryslik.

I would like to thank my in-laws Mrs. T.V. Meera Nambiar (who I call Mummy), Mr. E.K. Krishnan Nambiar (who I call Daddy) for all their love and support ; and for allowing me to be Asha’s husband. I would also like to thank Asha’s brother Mr. Anu Krishna for all his support and compassion.

I am deeply grateful to my lovely wife Dr. Asha Krishna. We had an arranged marriage, and did not really know each other well before the wedding. We got married on January 8th, 2014 ; and the last few months has been among the best in my life. Asha has been the most loving and compassionate wife. I consider myself fortunate to receive her love and affection, and consider myself lucky to be her husband. Thank you muthee for everything.

I humbly bow to the great Masters of Yoga to whom I am ever grateful. In India, among all human relationships, the relationship between a Guru (spiritual Master) and Shishya (student/seeker) is considered the highest. Because it is the only relationship which is truly one sided — in that a Guru only gives without expecting anything in return from the student except the student’s growth, while a student only takes without having anything to offer in return. I learnt a lot from studying the teachings of great Masters of Yoga (and Advaita Vendanta or nondualism) like Bhagavan Sri Ramana Maharshi, Sri Nisargadatta Maharaj, Eckhart Tolle, among many others, and by reading some of the commentary of the Sashtras by Swami Dayananda Saraswati. I am deeply grateful to all the ancient Gurus who walked on this planet all the way upto the first Guru or the Adiguru — Adiyogi Shiva — who is worshiped as Dakshinamurthy, the South-facing Jnana aspect of Shiva. I guess I am blessed to be born in India which has an extraordinary spiritual tradition. And I am most grateful to Sadhguru Jaggi Vasudev for being my Master. Sadhguru came into my life when I was looking for a living Master.

Finally I thank the Greatest Guru of them all — Life.

Lausanne, August 28, 2014 Nishanth Ulhas Nair

Abstract

Epigenetic factors like histone modifications are known to play an important role in gene regulation and cell differentiation. Recently, thanks to advances in technologies like ChIP-Seq which is a high-throughput, high resolution, and low cost technology for studying histone modifications and transcription factors, we have large amounts of data available. Therefore computational techniques become important for studying and interpreting this data.

In this thesis, we have focused on primarily building computational methods to analyze and study ChIP-Seq histone modification data. The work can be divided into two broad topics : (a) to process ChIP-Seq data computationally and to identify regions of biological interest ; (b) to use processed data for higher-level analysis to study problems in cell differentiation and evolution of cell types, based on phylogenetic approaches.

In the first topic, this thesis makes a contribution by addressing two problems : (i) We propose a two-stage statistical method, called *ChIPnorm*, to normalize ChIP-Seq data, and to find differential regions in the genome, given two libraries of histone modifications of different cell types. We show that our method removes most of the bias in the data and also provides a normalization that enables direct comparison of values between the two cell types. We show that our method outperforms the state of the art techniques in literature. (ii) We propose *probabilistic partitioning* methods to discover significant patterns in ChIP-Seq data. Our methods work on the principle of expectation-maximization, is simple and flexible, and takes into account signal magnitude, shape, strand orientation, and shifts. It runs in linear time and gives improved results on the state of the art techniques especially when used on sparse data.

In the second topic, we try to provide a link between the fields of epigenomics and evolution. We introduce the concept of *cell-type trees* based on the principles of phylogenetic inference on ChIP-Seq histone modification data. These cell-type trees are precisely defined and algorithmic techniques are designed to infer these trees from the data. In the process, we develop new data representation techniques and also a peak-finder to help us build good cell-type trees. We obtain biologically meaningful results and show that cell-type trees have the potential to study cell differentiation and the evolution of cell types across species.

Key words : epigenomics, epigenetics, histone modifications, ChIP-Seq, cell-type trees, evolution, phylogeny, evolution of cell types, ChIPnorm, probabilistic partitioning, expectation maximization, phylogenetic trees.

Résumé

Les facteurs épigénétiques, tels que les modifications des histones, joueent un rôle important dans la régulation des gènes et la différenciation cellulaire. Le développement récent de la technologie ChIP-Seq, une méthode de laboratoire de haut débit, haute précision, et faible coût, a mis à disposition des chercheurs de grandes quantité de données pour létude des modifications de l'histone et des facteurs de transcription. Le développment de techniques de calcul pour analyser ces données prend donc une place importante dans la recherche.

Dans cette thèse, nous présentons de telles techniques de calcul pour l'analyse des données ChIP-Seq sur la modification des histones. Le travail couvre deux thèmes principaux : (a) les méthodes requises pour traiter les données brutes de ChIP-Seq et d'y identifier les régions d'intérêt dans les sciences de la vie ; et (b) des méthodes pour analyser ces régions à plus haut niveau pour approfondir nos connaissances dans le domaine de la différenciation cellulaire et l'évolution des types de cellules.

Sous le premier thème, cette thèse contribue des solutions à deux problèmes. La première est une méthode statistique en deux étapes, ChIPnorm, pour normaliser les données de ChIP-Seq et s'en servir pour déceler des régions différentielles dans le génome, étant données deux bibliothèques de modifications des histones de différents types cellulaires. Notre méthode supprime la plupart des biais dans les données et permet une comparaison directe des valeurs entre les deux types cellulaires ; nous montrons qu'elle surpasse l'état de l'art dans ce domaine. La deuxième est une méthode de partitionnement probabiliste pour découvrir des modèles intèressants dans les données de ChIP-Seq. Notre méthode fonctionne sur le principe de l'espérance-maximisation, est simple et flexible, et prend en compte l'amplitude and la forme du signal, l'orientation des chaînes, et le déphasage. Il tourne en temps linéaire et surpasse l'état de l'art, en particulier pour les données éparses.

Sous le second thème, nous étudions les liens entre l'épigenomique et l'évolution. Nous introduisons le concept d'arbres cellulaires, construits selon les principes de l'inférence phylogénétique sur la base de données de modification des histones. Nous donnons une définition précise de ces arbres aussi bien que des algorithmes pour les construire à partir des données ChIP-Seq. Ces algorithmes utilisent de nouvelles représentations de ces données et un nouveau moteur de recherche pour les pics. Nous obtenons des résultats biologiquement pertinents et démontrons que les arbres cellulaires offrent un outil valable dans l'étude de la différenciation cellulaire et de l'évolution des types de cellules.

Mots clefs :
l'épigénétique, les modifications des histones, ChIP-Seq, arbres cellulaires, évolution, phylogénie, évolution des types de cellules, ChIPnorm, partitionnement probabiliste, espérance-maximisation

Contents

Acknowledgements **v**

Abstract (English/Français) **xi**

List of figures **xvii**

List of tables **xxiii**

1 Introduction **1**

2 Background **5**

2.1 Biological concepts 5

2.1.1 Epigenetics and Histone Modifications 5

2.1.2 Phylogenetic trees 6

2.1.3 Developmental biology and Evolution 6

2.2 Experimental technology and data 8

2.2.1 ChIP-Seq technology 8

2.2.2 Common data formats 9

2.2.3 Data availability 10

2.2.4 Data visualization 10

2.3 Computational Methods 10

2.3.1 Data representation 10

2.3.2 Statistical measures 12

2.3.3 Normalization techniques 13

2.3.4 Expectation-maximization algorithm 13

2.3.5 Phylogenetic tree building methods 14

2.3.6 Minimum spanning tree 15

3 ChIPnorm: a statistical method for normalizing and identifying differential regions in histone modification ChIP-Seq libraries **17**

3.1 Introduction 17

3.2 Methods 19

3.2.1 ChIPnorm scheme 21

3.2.2 Stochastic noise ε 21

3.2.3 Local genomic bias ν 23

3.2.4 Quantile normalization 24
3.2.5 The complete ChIPnorm method summarized 27
3.3 Experimental Design 28
3.4 Results 29
3.4.1 Comparative Analysis 30
3.4.2 Correlation with gene expression 37
3.4.3 Bivalent region analysis 38
3.4.4 Differential enriched regions along protein-coding genes 40
3.4.5 Using ChIPnorm in analyzing histone modification data from Arabidopsis 40
3.4.6 Software Availability 42
3.5 Conclusion 42

4 Probabilistic partitioning methods to find significant patterns in ChIP-Seq data 43
4.1 Introduction 43
4.2 Methods 46
4.2.1 Expectation-Maximization (EM) algorithm 46
4.2.2 Modified “Shape-Only” EM algorithm 46
4.2.3 Variations - with shift and flip 47
4.2.4 Seeding and initialization strategies 48
4.3 Results and Discussion 49
4.3.1 On simulated data 49
4.3.2 On real ChIP-Seq data 53
4.4 Conclusion 62

5 Cell-type trees 63
5.1 Introduction 63
5.2 Methods 65
5.2.1 Model of differentiation for histone marks 65
5.2.2 Data representation techniques 65
5.2.3 Phylogenetic analysis 67
5.2.4 On the inference of ancestral nodes 68
5.2.5 Peak-finding 68
5.3 Results and Discussion 69
5.3.1 Code 85
5.4 Conclusions 85

6 Further study using cell-type trees 87
6.1 Introduction 87
6.2 Lifting: inferring ancestral nodes 87
6.2.1 Algorithm for lifting 87
6.2.2 Results and Discussion 88
6.3 Using normalized raw data 91
6.3.1 Evaluation of different data representations 91

6.4 Comparing human and mouse cell types 93
6.5 Conclusion 94

7 Conclusions 97

A Appendix for Chapter 4 99
A.1 *R* code for probabilistic partitioning 99
A.1.1 Expectation-Maximization step for basic ChIP-partitioning algorithm . . 99
A.1.2 Complete algorithm for partitioning with random seeds 100
A.1.3 Iterative partitioning - standard version for two classes 100
A.1.4 Iterative EM partitioning with untrained flat class 101
A.1.5 Variations of the EM algorithm - shape-based partitioning 101
A.1.6 Shape-based partitioning with flips 102
A.1.7 Shape-based partitioning with shifts 102
A.1.8 Shape-based partitioning with flips and shifts 103
A.2 *R* code for generating simulated data 104
A.2.1 Two classes without shifts and flips 104
A.2.2 Two classes with flips 104
A.2.3 Four classes 105
A.2.4 Two classes characterized by co-localizing peaks of different width . . . 106

B Appendix for Chapter 5 107

Bibliography 121

Curriculum Vitae 123

List of Figures

2.1 *Example of a phylogenetic tree (source:* http://sites.duke.edu/dukeresearch/2011/11/14/being-the-shy-kid-may-have-its-ben
and adapted from http://phylogenous.files.wordpress.com/2011/01/treea.png*).* 6

2.2 *Example of cell-differentiation among blood cells (source:* Wikipedia*). The arrows indicate a tree structure.* 7

2.3 *The various steps in the ChIP-Seq procedure are shown (source* Wikipedia*).* 9

2.4 *Profile of a ChIP-Seq data library. Signal above some threshold are classified as peaks.* 11

2.5 *Profile of two ChIP-Seq data libraries* $L1$ *and* $L2$*. Differential regions in the genome are marked.* . 11

3.1 *Histone modification profile as seen in the UCSC genome browser [61]. Tracks 1 (red) and 2 (green) show the H3K27me3 modifications for ES and NP cells, respectively. Track 3 shows the differentially enriched regions found by our ChIPnorm method. Track 4 shows the differentially enriched regions found by the ChIPDiff method [125]. In tracks 3 and 4, red indicates differential enrichment in ES cells and green indicates differential enrichment in NP cells. Track 5 shows the UCSC genes.* . . . 20

3.2 *Overview of ChIP-Seq process. We see how we can get the ChIP-Seq library, input DNA control, and the random distribution (null hypothesis).* 22

3.3 *An example of random distribution (amplified binomial distribution with* $\alpha = 2$*) is shown in red and the estimated real distribution of the original ChIP-Seq data for one chromosome is also shown (blue). ABD — amplified binomial distribution. Data probability — estimated real distribution. The figure is zoomed into a small region for clarity.* 23

3.4 *Iterative Normalization of input DNA. (a) before first iteration. (b) after first iteration, post removal of outliers.* 24

3.5 *The new inverse cumulative distribution function on the modified libraries (after stage 1). On the* x *axis is the percentile, on the* y *axis are the bin values.* 26

3.6 *The schematic diagram of the ChIPnorm method. In the first stage one we find the enriched-significant bins by removing various kinds of errors in the data. In the second stage we normalize the two ChIP-Seq libraries and find differentially enriched bins.* 27

3.7 *Enrichment level of bins with respect to gene density in a 1 Mbp region. The* x *axis indicates lowest to highest gene density. (a) The* y *axis indicates the total number of H3K27me3 ChIP-Seq fragments divided by the number of Mbp regions (counts per megabase) found in each gene density. (b) The plots are re-normalized so that the* y *axis range is same for both ES and NP cell data. We see that the enrichment level of ChIP-Seq data increases with respect to gene density for both ES and NP cells.* 31

3.8 *Enriched bins with respect to gene density in 1 Mbp region. The plots are normalized. x axis 1 to 10 indicates lowest to highest gene density, while y axis 0 to 1 indicates minimum to maximum average number of differentially enriched bins for both ES and NP cells. Blue line indicates ES differentially enriched bins and red line indicates NP differentially enriched bins. The number of enriched bins per 1 Mbp should increase with gene density.* 32

3.9 *Plot of* $\frac{sensitivity}{(sensitivity + error)}$ *over five different threshold values. (a) NP is differentially over-expressed compared to ES, (b) ES is differentially over-expressed compared to NP.* 34

3.10 *Robustness studies: sensitivity and error analysis for ChIPnorm by fixing the fold-change threshold* $\tau = 3$ *and varying the bin size from 200 bp to 2000 bp in steps of 200 bp. Data from Mikkelsen* et al. *2007 [85].* 35

3.11 *Two ROC curves are shown for various methods. (a) first ROC: Class 1 - four-fold NP differentially over-expressed genes compared to ES; Class 0: rest of the genes. (b) second ROC: Class 1 - four-fold ES differentially over-expressed genes compared to NP; Class 0: rest of the genes.* 37

3.12 *Gene profile according to expression and histone modifications. Genes are grouped in (A-E) according to increasing ratio of expression level in ES cells and NP cells. Within each groups, genes are classified into 4 types. Type 2 genes have differential histone enrichment in ES cells in their promoter regions and type 4 genes have differential enrichment in NP cells. (a) Percentage of type 2 genes is decreasing, while percentage of type 4 genes is increasing along the group (A-E). (b) Percentage of type 2 genes is increasing, while percentage of type 4 genes is decreasing along the group (A-E).* . . 38

3.13 *Bivalent regions in genes: in 333 selected genes and in UCSC known genes. Genes are attributed to classes according to the presence of modifications in ES and NP cells. The "A-B" notation in the labels indicates the presence of modification of type "A" in ES cells and modification of type "B" in NP cells. (*) marked labels have bivalent domains in ES cells.* 39

3.14 *Bivalent gene profile vs. expression data. Genes are grouped in (A-E) according to increasing ratio of expression level in ES cells and NP cells. Each bar shows the percentage of genes with the corresponding "A-B" modifications (as listed in the box), "A" for modifications in ES cells and "B" for modifications in NP cells. It is seen that there is a strong over-representation of the K4+K27-K4 transition (yellow) in the genes class, which is strongly up-regulated in NP cells.* 40

3.15 *Feature aggregate plot of the differential/non-differential regions identified by the ChIPnorm method. Each row corresponds to a region from ChIPnorm and each column corresponds to a genomic feature of protein coding GENCODE genes. Curve inside a cell represents the relative frequency of overlap between the ChIPnorm identified regions and the genomic feature of GENCODE genes, when compared to a similar relative frequency of overlap that would occur by random chance. Figure created using Segtools [18].* 41

4.1 ***Simulated data without shifts or flips.*** *Shows the data and the patterns found using the the K-Means clustering method, CAGT methods, and the probabilistic partition method (shape-based without shift or flips). Sub-figures* $a1$, $b1$, $c1$, $d1$, *and* $e1$ *are for* $f = 50$ *and* $a2$, $b2$, $c2$, $d2$, *and* $e2$ *are for* $f = 1$. *Red is class 1 (class* $c1$ *in Table 4.1) and cyan is class 2 (class* $c2$ *in Table 4.1).* 50

4.2 ***Simulated data with flips.*** *Data (4000 samples) consist of two classes characterized by Gaussian-shaped patterns. Each class is represented by two subsets of 1000 samples, one showing the underlying pattern in native, the other one in reversed (flipped) orientation. Sub-figures* $a1$*,* $b1$*,* $c1$*, and* $d1$ *are for* $f = 50$ *and* $a2$*,* $b2$*,* $c2$*, and* $d2$ *are for* $f = 1$*.* $b1$ *and* $b2$ *are aggregation plots of the same data but with all samples presented in their native orientation. It can be seen that the probabilistic partitioning method (shape-based) using flips captures the actual data patterns at high (*$f = 50$*) and low (*$f = 1$*) coverage. The CAGT (correlation) method works well for* $f = 50$ *only. Red and cyan colors correspond to classes c1 and c2 in Table 4.3, respectively.* 53

4.3 ***Results from experiment with four classes.*** *(a)-(e) Aggregation plots (APs) for simulated data. (f)-(i) Probability weighted APs of samples as classified by the probabilistic partitioning algorithm.* 54

4.4 ***Experiment with two classes characterized by co-localizing Gaussian peaks.*** *Shown are aggregation plots (APs) for the two classes in the data, and for the rediscovered classes obtained with different clustering algorithms. Note that CAGT (correlation) reports only one class. The absence of a second class is reflected by the horizontal red dashed line at height zero in subfigure (c).* 55

4.5 ***H3K4me1 and H3K4me3 histone modification data.*** *H3K4me1 and H3K4me3 data mixed together and separated using the K-means, CAGT (correlation), CAGT (Euclidean), and probabilistic partitioning approach (non-shape and shape-based). Red is for the class which represents H3K4me1 and cyan is for H3K4me3 for figures (b) to (h). In the figures, each class is normalized so that the maximum value is 1 for the sake of clarity for each class. Only for sub-figure (b) we normalize using a global maximum of H3K4me1 and H3K4me3.* 56

4.6 ***Partitioning of nucleosome positioning patterns in human promoters.*** *All curves are drawn to the same scale. Probabilistic partitioning reveals strong oscillatory patterns for subclasses of promoters which partially cancel each other out when mixed together.* 57

4.7 ***Aggregation plots (APs) for various genomic features in different promoter classes.*** *(a) APs for different features superposed for the same class. (b) Same feature for different classes in one plot. Position zero corresponds to the transcription start site.* 58

4.8 ***Shape-based peak evaluation with shifting.*** *The figure illustrates the effects of probabilistic partitioning on a CTCF peak list provided by ENCODE in terms of motif enrichment. (a) Probabilistic partitioning with shifting. (b) Partitioning based on original p-values. Method details: CTCF binding motifs where identified by scanning the DNA sequence around peak centers with the JASPAR matrix MA0139.1 at a P-value threshold of* 10^{-5}*. The percentage of sequences containing a CTCF motif is plotted in a sliding window of 50 bp. The numbers in parentheses indicate the sizes of the peak lists. For fair comparison, the threshold for partitioning with the original P-values was chosen such as to match the numbers of good and bad peak obtained with probabilistic partitioning. The motif enrichment profile for the complete peak list (dotted line) is included in both graphs.* . . . 60

4.9 ***Motif enrichment profiles for IDR-selected ChIP-Seq peak lists.*** *Peak sets were selected at an IDR of 1%.* 61

5.1 *An example of an interesting region as defined by overlap representation is shown. The dotted horizontal lines represents a portion of the genome. Each row stands for a separate ChIP-Seq library (L1-L5). The dark lines represent peak regions. 1 and 0 is the data representation for each library.* . 67

5.2 *ChIPnormSig method to identify significant regions (which are used as peak regions) from histone modification ChIP-Seq library.* . . . 69

5.3 *Cell-type tree on H3K4me3 data (ENCODE peaks) using only one replicate: (a) windowing representation, (b) overlap representation.* . . . 71

5.4 *Using maximum parsimony (TNT software) on H3K4me3 data (ENCODE peaks) using only one replicate (overlap representation).* . . . 73

5.5 *Cell-type tree on H3K4me3 data (ENCODE peaks, using all replicates): (a) windowing representation, (b) overlap representation.* . . . 74

5.6 *Cell-type tree on H3K27me3 data (ENCODE peaks), using all replicates and overlap representation.* 74

5.7 *Cell-type tree using windowing representation on H3K4me1 data.* . . . 75

5.8 *Cell-type tree using windowing representation on H3K9me3 data.* . . . 75

5.9 *Cell-type tree using windowing representation on H3K27ac data.* . . . 76

5.10 *Cell-type tree on H3K4me3 data (ENCODE peaks, using all replicates) on peaks with negative log p-value* $\geq$ *10: (a) windowing representation, (b) overlap representation.* . . . 77

5.11 *Overlap representation using overlapping peaks (between replicates) which have an IDR value* ≤ 0.025 *(output of IDR program). Since the distance between replicates is 0, the two replicates of each cell type are represented in one label.* . . . 78

5.12 *H3K4me3 data (ENCODE peaks), overlap representation on peaks with negative log p-value* $\geq$ *10. Replicate leaves are removed and replaced by their parent.* . . . 79

5.13 *Cell-type tree on H3K4me3 data (using one replicate) using (a) windowing representation (b) overlap representation. Peaks generated by ChIPnormSig method (FDR 5%).* . . . 80

5.14 *Cell-type tree on H3K4me3 data (using all replicates) using (a) windowing representation (b) overlap representation. Peaks generated by ChIPnormSig method (FDR 5%).* . . . 80

5.15 *Cell-type tree on H3K4me3 data (using profile data representation) using Overlap representation: (a) ENCODE peaks, (b) ChIPnormSig peaks.* . . . 81

5.16 *Cell-type trees to study evolution of cell types - a constructed example is shown in this figure. Tree* $T2$ *- current species* $S2$*. Tree* $T1$ *- ancestral species* $S1$*.* ST *- sub-tree.* . . . 84

6.1 ***Part of lifting procedure****. An example showing a small portion of the lifting algorithm is shown. The edge between node A and node 2 is the smallest among the considered edges. Therefore node A is lifted upwards to its ancestor and it is made the new leaf.* . . . 88

6.2 ***Lifting procedure for ancestral node reconstruction****. Shows cell-type tree for H3K4me3 data using overlap representation. The internal nodes are labelled using green edges to one of the cell-types in the leaf node (by lifting procedure).* . . . 89

6.3 ***Lifting procedure for ancestral node reconstruction****. Shows cell-type tree for H3K4me3 data using overlap representation after lifting process. (Figure is not drawn to scale.)* . . . 90

6.4 ***Minimum spanning tree****. Minimum spanning tree for H3K4me3 data using overlap representation (the edges are not drawn to scale).* . . . 91

6.5 ***Cell-type tree using windowing representation.*** *Data representation using: (a) Peak data, (b) Raw data, (c) Mean normalization of raw data, (d) Mean normalization of raw data (for each chromosome), (e) Standard score normalization of raw data (for each chromosome).* . . . 95

6.6 ***Human and mouse cell types***. *Cell-type tree using overlap representation on human and mouse cell types. Peak data considered only on orthologous regions of human and mouse.* 96

6.7 ***Human and mouse cell types by removing some peaks***. *The number of peaks available for mouse where much more than for human. Therefore the number of peaks used in mouse data were reduced so as to reduce the bias in the results. (Peak data considered only on orthologous regions of human and mouse.) (a) Deleting peaks in mouse data whose negative log p-value is smaller than 50. (b) Divide the genome into windows of 1 Mbp. Then remove those windows if the length of the peak region in either human or mouse data was 50% more than the other (comparing hESC T0 and mE14).* 96

B.1 *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 1 and rest of the cell types (62 replicates) have all 0 (one error allowed at most on both sides) using all replicate ENCODE peak data.* 108

B.2 *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 0 and rest of the cell types (62 replicates) have all 1 (one error allowed at most on both sides) using all replicate ENCODE peak data.* 109

B.3 *For H3K27me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 1 and rest of the cell types (13 replicates) have all 0 (one error allowed at most on both sides) using all replicate ENCODE peak data.* 110

B.4 *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells for day 0, 2, 5, 9, 14 have a pattern 01000. ENCODE peaks are used and only regions which have identical values of all the replicates for each cell type are considered.* 111

B.5 *For H3K27me3 data: gene enrichment analysis containing regions of the genome where ES cells for day 0, 2, 5, 9, 14 have a pattern 01000. ENCODE peaks are used and only regions which have identical values of all the replicates for each cell type are considered.* 112

List of Tables

3.1 *Sensitivity analysis percentages using various methods (data from Mikkelsen* et al. *2007 [85]). Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The parameters of all the methods (except for MACS as explained in the text) were adjusted so that all of them give almost the same percentage (≈* 15%*) of experiment "sensitivity (ES K27-enriched)".* 33

3.2 *Values of the various thresholds (T1, T2, T3, T4, T5) used for the various methods used in Figure 3.9.* 34

3.3 *Sensitivity analysis for human ES and GM12878 cells (replicate 1 data from ENCODE Broad database) percentages using various methods. Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The parameters of all the methods (except for rank normalization as explained in the text) were adjusted so that all of them give almost the same percentage (≈* 11%*) of experiment "sensitivity (HES K27-enriched)".* 35

3.4 *False-positive rate (FPR) analysis for human ES cells (H3K27me3 data from ENCODE Broad database) for the two replicates. We see the percentage of false positive using various methods. Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The thresholds used are same as those in Table 3.3.* 37

4.1 *Results for simulated data without shifts or flips. Model accuracy is expressed as Pearson correlation coefficient between original and rediscovered patterns/classes. The percentage of samples attributed to a class is shown in parentheses. The classes c1 and c2 correspond to the red and cyan curves in Figure 4.1, respectively.* 51

4.2 *Classification error (in percentage) between the discovered patterns and their data classes.* 51

4.3 *Results for simulated data with flips. Model accuracy is expressed as Pearson correlation coefficient between the original and rediscovered patterns/classes. The percentage of samples attributed to a class is shown in parentheses. The classes c1 and c2 correspond to the red and cyan curves in Figure 4.2, respectively.* 52

4.4 *Model accuracy (represented by Pearson correlation) and classification error between the discovered patterns and their data classes for the various methods. The time (in seconds) taken for each of the methods is also shown. Real data for H3K4me1 and H3K4me3 around TSS regions are mixed (34741 samples in each dataset with each sample containing 99 bins). The percentage of each class is shown in brackets. H3K4me1 and H3K4me3 stand for the 2 datasets. (Values are rounded to the fourth decimal place for model accuracy and two decimal places for classification error.)* 54

5.1 *Cell types, short description, and general group for H3K4me3, H3K27me3, H3K4me1, H3K9me3, H3K27ac data. For details see the ENCODE website [101]. The mark ✓ shows the usage of that cell type for that particular histone mark.* 71

5.2 **Statistics for cell-type trees on H3K4me3 data.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. Rows correspond to various methods (WM: windowing; OM: overlap; TP: top peaks with threshold of 10). The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the NJ tree from the Hamming distance between the leaves. ENCODE means peaks from ENCODE data is used while ChIPnormSig means peaks from ChIPnormSig program is used. (one replicate) means only one replicate for each cell type is used, (all replicates) means all available replicates (1, 2, or 3) for each cell type is used, (profile) means a profile representation created using all replicates for each cell type is used. MP - maximum parsimony using TNT software.* 72

5.3 **Statistics for cell-type trees on H3K4me3 data using top peaks.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. Rows correspond to various methods (OM: overlap; TP: top peaks). threshold x means that the peaks which have a (negative) log p-value ≥ x is used. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the neighbor-joining (NJ) tree from the Hamming distance between the leaves. ENCODE peak data is used. (all replicates) means all available replicates (1, 2, or 3) for each cell type is used. We can see that the method is robust to various kinds of threshold used.* 77

5.4 **Statistics for cell-type trees on H3K4me3 data using IDR analysis.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the NJ tree from the Hamming distance between the leaves. Rows correspond to various methods OM (two replicates): overlap representation on all cell-types which have exactly two replicates (on all available peaks). OM-IDR (two replicates), (threshold x): Overlap representation used on overlapping peaks (between replicates) which have an IDR value ≤ x (output of IDR program). Since these are overlapping peaks the SR ratio is always 0 given the nature of overlap representation. ENCODE peak data is used. (two replicates) means all two replicates for each cell type is used (34 cell-types, 68 libraries). We considered data containing only two replicates for this work.* 78

5.5 **Analysis on paths.** *Table shows the number of different types of changes across various days of ES cells. Dx means day x of ES cell type. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column.* 83

5.6 **Analysis on paths.** *Table shows the number of different types of changes for ES, HUVEC, and HBMEC cell types. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column. NA - not applicable (because data for the cell-type is not available).* 84

5.7 **Analysis on paths.** *Table shows the number of different types of changes for ES, WI38, AG04550, HPF cell types. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column. NA - not applicable (because data for the cell-type is not available).* 85

6.1 R **ratio.** *The R ratios for different kinds of data representation are shown for 37 cell-types of H3K4me3 data, each cell-type compared with every other. The frequency of occurrence with a small range of R ratio is shown for all the data representations.* 92

6.2 **Statistics for cell-type trees on H3K4me3 data for different data representation.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades (fraction values indicate only one of the two replicates fall in that clade); the total number of cells of that type is in the top row. Fractions are used when only one of the two replicates fall into the clades. Rows correspond to various data representation techniques. Windowing representation is used for all. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the neighbor-joining tree from the Hamming distance between the leaves. Only peaks from the ENCODE data are used.* 93

1 Introduction

Epigenetics has been usually defined as the study of mitotically and/or meiotically heritable changes in gene function that cannot be explained by changes in DNA sequence [103]. One of the main aims of epigenetics is to elucidate how genetic information encoded in the DNA sequence and non-genetic aspects like how the DNA is packaged in the nucleus jointly control gene expression [14]. Epigenetic factors are known to play an important role in cell differentiation and in cancer. Some of the common well-known epigenetic factors are DNA methylation and histone modifications. Other epigenetic factors may include even small RNA molecules which are shown to be the reason for the progeny of the unicellular organism *Paramecium tetraurelia* to always retain the parental mating type (called even (E) and odd (O)), although their offsprings all start development with identical, mixed genomes [111].

Histones are proteins that package the DNA into nucleosomes [72]. These histones are subjected to various types of modifications like methylation, citrullination, acetylation, phosphorylation, SUMOylation, ubiquitination, and ADP-ribosylation, which alter their interaction with the DNA and nuclear proteins, thereby influencing gene transcription and genomic function. These modifications form an important category of epigenetic changes, changes that help us understand why various types of cells exhibit very different behaviors in spite of their shared genome. Thus the study of histone modifications is crucial to the understanding of genomic function.

ChIP-Seq (immunoprecipitation combined with high-throughput DNA sequencing), also known as ChIP-sequencing, is a recent technology which has become the main approach for capturing histone modifications and transcription factors bound to the DNA, due to its high throughput, high resolution, and low cost [7, 59, 80]. A ChIP-Seq experiment produces a large number of sequence tags that are mapped to the genome thereby resulting in a genome-wide profile of tag counts. Since there is increasing amount of data on histone modifications available due to this technology, computational approaches are very important to study such high-throughput data. Therefore given the importance of histone modifications as an important epigenetic mark and given the reasonably large amount of data available, we decided to study histone modification ChIP-Seq data and its effect on cell-differentiation/evolution in more detail.

Quite a few computational problems relating to histone modifications or ChIP-Seq data have been addressed in literature. One of the important problems is the issue of peak-calling or peak-finding. Basically a peak is a region of the genome where there are a higher number of ChIP-Seq fragments compared to the control data, or some fixed threshold, or compared to what is expected by chance. The presence of a peak is often seen as the presence of a histone mark and therefore peaks become an important preprocessing step for further biological study. Several peak finders have been described in literature, like MACS [131], FindPeaks [39], PeakSeq [102] etc. A detailed evaluation of peak-finders can be found in [124].

One another important problem is to identify differential regions of the genome given two histone modification ChIP-Seq data libraries (possibly belonging to two different cell types). This problem is important to help us understand the role of histone marks in differentiating one cell type from an other, especially given that all cells in one individual organism have almost the same genome. The problem turns out to be surprisingly difficult, even in simple pairwise comparisons, because of the varying levels of significant noise in different ChIP-Seq data. Some methods like ChIPDiff [125], RSEG [115], etc. try to address this problem.

Another important problem is to discover significant recurring patterns in ChIP-Seq data. Chromatin-signatures are a term used to designate recurrent patterns found in ChIP-Seq-based histone modification maps and other types of chromatin profiling data [56]. It is usually represented as a vector of average tag counts in bins of certain sizes in a collection of larger genomic regions. Identifying these chromatin-signatures have been addressed in literature. Some of the simplest methods for identification of these signatures is to use hierarchical clustering [109] and K-means [79] algorithms. For example seqMINER [129] is a method which organizes data into groups of loci having similar features and it contains an in-built K-means function. More specialized methods like ChromaSig [55], ArchAlign [68], CATCHprofiles [93] and CAGT [67] can be used when recurring patterns are in different orientations or misaligned.

There are many other problems related to histone modifications or ChIP-Seq technology which have been addressed by computational methods in literature. One such problem is make sure ChIP-Seq experiments are reproducible. Therefore it has been recommended to perform at least two biological replicates of each ChIP-Seq experiment and examine the reproducibility of both the reads and identified peaks [69, 102]. To measure the reproducibility at the level of peak calling, an irreproducible discovery rate (IDR) analysis [73] can be applied to the two sets of peaks identified from a pair of replicates. A detailed view of many such problems occurring in ChIP-Seq data has been described in [5]. Also, recently some higher-order analysis like reconstructing gene regulatory networks from ChIP-Seq and other high throughput data has been addressed [126, 15, 50].

This thesis addresses three problems related to histone modification ChIP-Seq data, two of which we have seen above. These three problems can be categorized into two main topics:
(1) **Data processing and pattern discovery:** to find various methods to normalize or remove

noise in the histone modification ChIP-Seq data and to use this to identify interesting regions in the genome which are of biological interest.
(2) **A higher-order analysis linking the fields of cell-differentiation and evolution:** to study the role of histone modifications in cell-differentiation and evolution of cell types by using the processed data.

In the first topic we address two specific problems:

1) *Finding differential regions*: As seen above, the problem is to identify differential regions of the genome given two histone modification ChIP-Seq data libraries. We noticed that the results given by many of the previous approaches were affected by the various levels of signal-to-noise ratio between the two data libraries. To address this problem, we propose a two stage statistical method, called ChIPnorm to normalize the two libraries and find differential regions between them. We compared our approach with the previously reported state of the art techniques and found that the ChIPnorm technique outperforms them. We have described this work in detail in chapter 3.

2) *Discovering significant patterns*: The next problem we address is to find significant recurrent patterns in histone modification data or transcription factor ChIP-Seq data. Identifying such significant recurrent patterns is an important problem for understanding biological mechanisms. To address this problem, we propose a probabilistic partitioning approach, based on an expectation-maximization framework. Our methods take into account signal magnitude, shape, strand orientation, and shifts. We compare our methods with some current methods and demonstrate significant improvements, especially when using sparse data. More details of this work are described in chapter 4.

In the second topic, we attempt to provide a link between the fields of phylogeny and epigenetics. We first discuss the similarities (and dissimilarities) between the cell-differentiation and evolution; and discuss why phylogenetic trees can be used to study cell-differentiation process, when using histone modification data. To achieve this, we introduce the novel concept of cell-type trees and precisely define such trees in the context of histone modifications. We then provide a procedure for building such trees. We show that these cell-type trees give us information about how diverse types of cell types are related. In this process, we propose new data representation techniques, peak-finding techniques, and distance measures for ChIP-Seq data and use these together with standard phylogenetic inference methods to build biologically meaningful cell-type trees. We demonstrate our approach on various kinds of histone modifications for various cell types, also using the datasets to explore various issues surrounding replicate data, variability between cells of the same type, and robustness. We use the results to get some interesting biological findings. We discuss that cell-type trees may be useful in studying the cell-differentiation process. We also discuss how cell-type trees can be used to study the evolution of cell types and discuss that cell differentiation tree often recapitulates the phylogeny of cell types. The details of this work are given in chapter 5.

In chapter 6, we outline a few details of our ongoing/future work on cell-type trees. Three problems are addressed: (1) the inference of ancestral nodes; (2) using of normalized raw data instead of peak data for data representation; (3) study of evolution of cell types using mouse and human data. We outline some basic ideas and the challenges faced in each of these problems.

The background material required for the future chapters are given in chapter 2. We have given more details about the previous work present in literature in the introduction sections of chapters 3-5, for their respective topics. Finally this thesis ends with a conclusion chapter (chapter 7).

The work shown in this thesis has been mainly done by the author by collaborating with many others. The collaborator contributions for each chapter are shown below. Keywords used for collaborator names: NUN - Nishanth Ulhas Nair, BMEM - Bernard M.E. Moret, PB - Philipp Bucher, ADS - Avinash Das Sahu, YU - Yu Lin, SK -Sunil Kumar, AM - Ana Manasovska, PG - Paulina Grnarova, JA: Jelena Antic, JAB - Judy A Brusslan, MP - Matteo Pellegrini.
Chapter 3: For most of this chapter, the experiments and methods were conceived and designed by NUN ADS PB BMEM; Implemented the methods and performed the experiments: NUN ADS; Analyzed the data: NUN ADS PB BMEM; defined the problem statement: PB. The subsection "Using ChIPnorm in analyzing histone modification data from Arabidopsis" was carried out by NUN in close collaboration with MP, JAB and their groups. AM helped in making the ChIPnorm code more user friendly.
Chapter 4: Methods were implemented by NUN PB. Defined the problem statement and main idea of the approach: PB. Experiments were carried out by NUN, SK, PB. Analyzed the data: NUN SK BMEM PB.
Chapter 5: Problem was defined by: NUN PB BMEM. Designed the methods and algorithms: NUN YL BMEM. The experimental design was done by: NUN YL ADS PB BMEM. Implemented the methods: NUN and JA with help from PG. Carried out the experiments: NUN AM JA PG. Analyzed the data: NUN YL PB BMEM.
Chapter 6: All problems were defined by NUN PB BMEM. Experiments and methods were designed for all problems by NUN PB BMEM. YL helped in designing experiment for the lifting problem. Experiments for lifting was implemented, carried out, and analyzed by NUN. For experiments using normalized raw data, experiments were implemented, carried out, and analyzed by NUN JA. Experiments between mouse and human cell types were implemented, carried out, and analyzed by NUN PG.

2 Background

This thesis addresses the twin topics of data processing and finding interesting regions in ChIP-Seq histone modification data (chapters 3 and 4), and doing a higher-level analysis by building cell-type trees to study the role of histone marks in cell-differentiation and evolution of cell types (chapters 5 and 6). In this chapter, we provide some introductory material required for the later chapters.

2.1 Biological concepts

We first look into some of the important biological concepts required for this thesis.

2.1.1 Epigenetics and Histone Modifications

As seen before, epigenetics has been usually defined as the study of mitotically and/or meiotically heritable changes in gene function that cannot be explained by changes in DNA sequence [103]. Some of the most common epigenetic factors are histone modifications and DNA methylation.

Histones are proteins that package the DNA into nucleosomes [72, 38]. There are five major families of histones exist: H1/H5, H2A, H2B, H3, H4 [91, 12]. The core histone are H2A, H2B, H3, and H4, while the linker histones are H1 and H5. Each of the two core histones assemble to form one octomeric nucleosome core. That is, this nucleosome core is formed of two H2A-H2B dimers and a H3-H4 tetramer, forming two nearly symmetrical halves by tertiary structure [77]. 147 base pairs (bp) of DNA wrap around this core particle [77]. There is approximately a 50 bp of DNA, called linker DNA, separating each pair of nucleosomes.

These histone proteins are subjected to post translational modification by enzymes primarily on their N-terminal tails. Some of the various types of chemical modifications are methylation, citrullination, acetylation, phosphorylation, SUMOylation, ubiquitination, and ADP-ribosylation. These modifications influence how tightly the DNA is wrapped around in the nucleus and they influence which genes are transcribed in which way in different cell types, thus allowing various types of cells to exhibit very different behaviors in spite of their shared genome. Some of the well known histone modifications are H3K4me3, H3K27me3, H3K4me1, H3K9me3, H3K27ac. H3K4me3 is usually associated with activation of genes while H3K27me3 is usually associated with repression of genes [85]. Some examples of other activators are H3K27ac [23], H3K4me1 [8] while for repressors it is H3K9me3 [7].

2.1.2 Phylogenetic trees

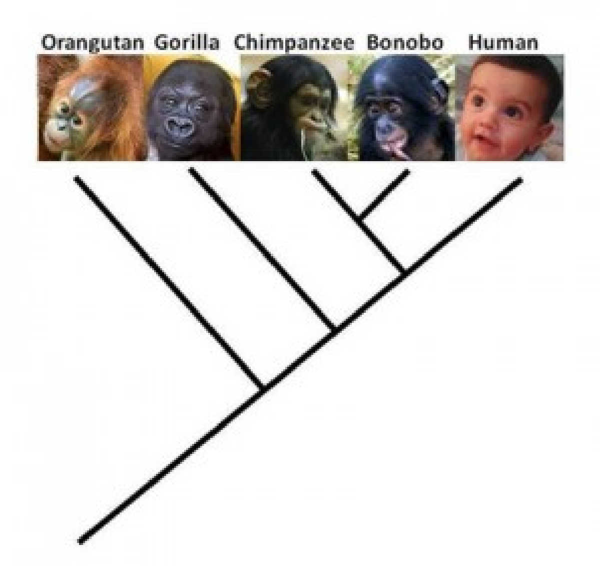


Figure 2.1 – *Example of a phylogenetic tree (source:* http://sites.duke.edu/dukeresearch/2011/11/14/being-the-shy-kid-may-have-its-benefits/ and adapted from http://phylogenous.files.wordpress.com/2011/01/treea.png*).*

Phylogenetics is the study of evolutionary relationships among groups of organisms and phylogenetic trees are used to represent such relationships. There are three commonly used tree building frameworks to build phylogenetic trees using DNA sequence data: distance based, maximum parsimony, and maximum likelihood approaches [40]. Phylogenetic trees build on DNA or protein sequences are usually represented as binary trees, with leaf nodes representing the modern species and the ancestral nodes representing some ancestral species from which the modern species were derived. The branches (edges) of the tree represent number of evolutionary changes or evolutionary time. An example of a phylogenetic tree is shown in Figure 2.1.

2.1.3 Developmental biology and Evolution

In developmental biology, the process by which a less specialized cell becomes a more specialized cell type is called cell differentiation [22]. During the development of a multicellular organism, differentiation occurs numerous times as the organism changes from a simple zygote to a complex system of tissues and cell types. Cell differentiation almost never involves a change in the DNA sequence itself (with a few exceptions). Therefore since all cells in one individual organism have the same genome, epigenetic factors and transcriptional factors play an important role in cell differentiation [71, 75, 76]. For mammals, early development is characterized by the zygote becoming a ball of cells called blastocyst which later becomes embryonic stem cells. Embryonic stem cells are basically undifferentiated biological cells that can differentiate into specialized cells and can divide through mitosis to produce more stem cells [22].

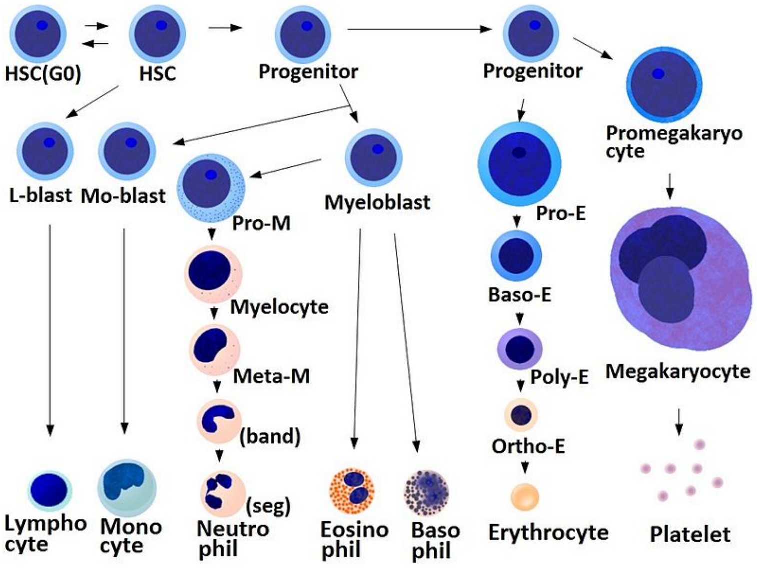


Figure 2.2 – *Example of cell-differentiation among blood cells (source:* Wikipedia*). The arrows indicate a tree structure.*

Cells of a multicellular organism are classified into cell types based on their morphological, physiological or molecular characteristics. Since cell differentiation transforms less specialized cell types into more specialized ones and since most specialized cells of one organ cannot be converted into specialized cells of some other organ, the paths of differentiation together form a tree, in many ways similar to the phylogenetic trees used to represent evolutionary histories. An example of tree structure among differentiation of blood cells is shown in Figure 2.2. In contrast to cell differentiation, evolution deals with changes taking place in the genome across species.

We outline some of the similarities and differences between the fields of cell-differentiation and evolution. Some similarities are as follows: (1) In cell differentiation, more specialized cells are evolved from less specialized cells, while in evolution, present-day species have evolved from some ancestral species — so a tree representation is possible in both cases. (2) The observed changes in the epigenetic state are inheritable, again much as mutations in the genome are (although, of course, through very different mechanisms and at very different scales). As histone modifications are replicated in the cell-differentiation process we can assume not all histone modification marks are perfectly copied from parent to child, due to the stochastic nature of chemical reactions. Therefore there could be stochastic changes in epigenetic features. (Some plausible mechanisms of inheritance of histone modifications marks are given in [81, 132, 98].) (3) Epigenetic traits are subject to stochastic changes, just like genetic mutations. Some of the differences are: (1) The time scales may be very different, cell differentiation happens in a few days or weeks, while evolutionary time scales can be large (millions of years in the case of evolution of mammals). (2) While phylogenetic analysis places all the modern data at the leaves of a tree, with regard to cell differentiation some of the less specialized cell types may continue to exist in the organism and hence they could be ancestral nodes and not leaf nodes. (3) There are well defined models of evolution of DNA sequences for phylogeny while the models of differentiation for mutations in epigenetic states across cell types are not so well defined.

In cell differentiation, the program of mutational events is itself the result of evolution, so as observed by Arendt [3], the cell differentiation tree often recapitulates the phylogeny of cell types. Thus keeping the similarities and differences between the fields of evolution and cell differentiation in mind, we used phylogenetic methods for the analysis of cell types using histone modification data to study cell differentiation. We call the trees we build as cell-type trees (details given in chapter 5).

We also discuss the role of evolution of cell types and how cell-type trees can be used to study this phenomenon in chapter 5. The number of cell types in various organisms do vary. We discuss in chapter 5, how cell-type trees can be used to study the evolution of cell types when comparing the current species to some ancestral species.

2.2 Experimental technology and data

2.2.1 ChIP-Seq technology

The current technology to capture histone modifications is chromatin immunoprecipitation (ChIP), which uses an antibody to isolate DNA fragments in contact with histones that carry a specific modification or transcription factors. ChIP-chip, ChIP-PET, and ChIP-SAGE are some of the ChIP-based technologies used for the study of histone modifications or transcription factor binding in genomic regions [57, 63, 123]. Thanks to advances in sequencing technologies, ChIP-Seq has become the main approach for capturing histone modifications and transcription factors, due to its high throughput, high resolution, and low cost [7, 59, 80, 66]. In the ChIP-Seq process, the sequence of one end of the DNA fragment is read to provide a tag which is then mapped to an assembled genome to determine the location of the DNA fragment. Various steps of the ChIP-Seq process are shown in Figure 2.3.

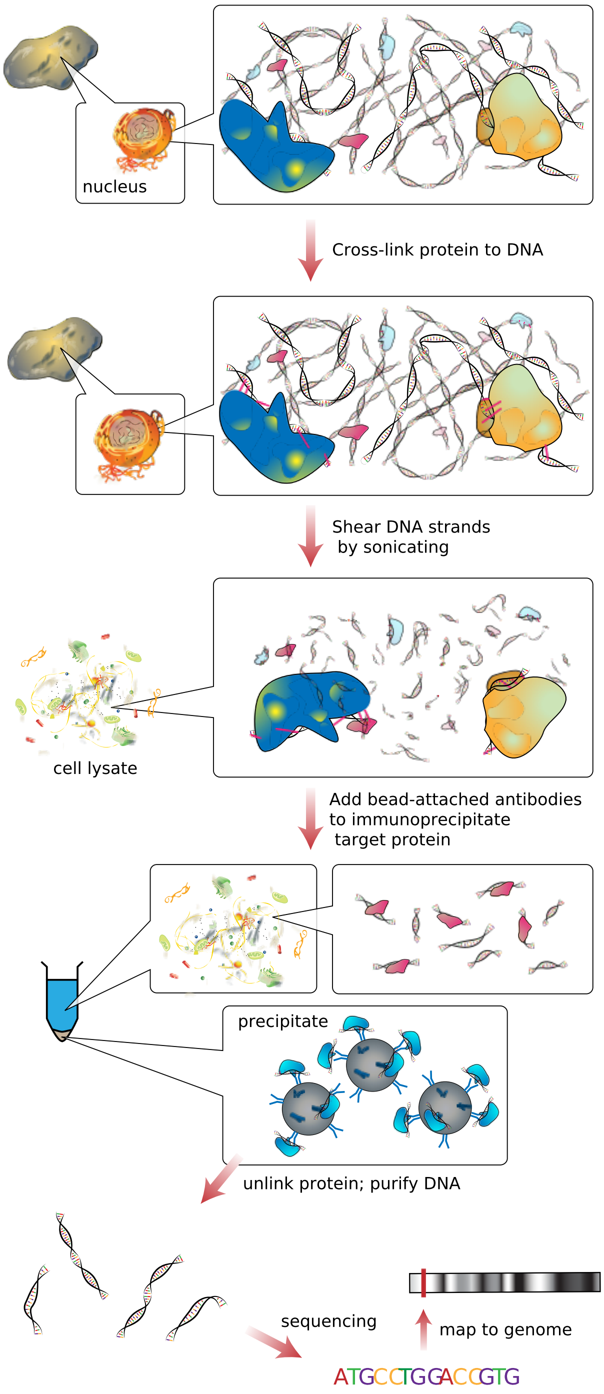


Figure 2.3 – *The various steps in the ChIP-Seq procedure are shown (source* Wikipedia*).*

2.2.2 Common data formats

The ChIP-Seq process gives us a collection of fragments whose ends are sequenced (tags) and mapped back into the genome, the data we get is usually represented as a collection of fragments with chromosome no., starting and ending position in the chromosome (base pair resolution). The common data formats are BED or BAM.

The first three required BED fields are:
chrom - The name of the chromosome (e.g. chr1, chrY).
chromStart - The starting position of the feature in the chromosome.
chromEnd - The ending position of the feature in the chromosome.
There are 9 additional optional BED fields. The sixth position usually signifies the positive (+) or negative strand (-).

BAM is the compressed binary version of the Sequence Alignment/Map (SAM) format, a compact and index-able representation of nucleotide sequence alignments. More details of the formats are given in the UCSC Genome Browser http://genome.ucsc.edu/FAQ/FAQformat.html.

2.2.3 Data availability

The most common sources of data for histone modifications are ENCODE project [82], Roadmap Epigenomics project, Gene Expression Omnibus (GEO) repository.

2.2.4 Data visualization

The UCSC genome browser is often used to visualize the data [61].

2.3 Computational Methods

2.3.1 Data representation

One of the common ways to represent the ChIP-Seq data, is by dividing the genome into non-overlapping windows (called bins) and collecting, for each bin, a count of the mapped sequence tags that fall within the bin (called bincount). Thus we get a “library", which is simply a list of non-negative integers, each successive integer associated with the next bin. Since the median length of each histone modification ChIP-Seq fragment is about 200 bp [7, 99], one can approximate the center of each fragment by shifting the tag end position by 100 bp (or so) downstream or upstream, according to its orientation on the chromosome, as done in [125]. (We note that for paired-end data the fragment length distribution is known and for single-end data the fragment length distribution can be inferred from an autocorrelation analysis. Knowledge of this distribution may be important for proper normalization and data processing.) We also use the word “library” for other forms of data representation of ChIP-Seq data too.

Now these ChIP-Seq libraries are used for statistical analysis. A hand-drawn example of a

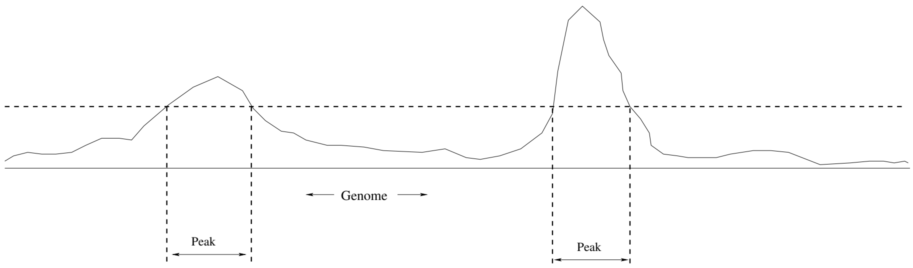


Figure 2.4 – *Profile of a ChIP-Seq data library. Signal above some threshold are classified as peaks.*

ChIP-Seq data library is shown Figure 2.4. The regions of the genome where the signal falls above some threshold are classified as peak regions.

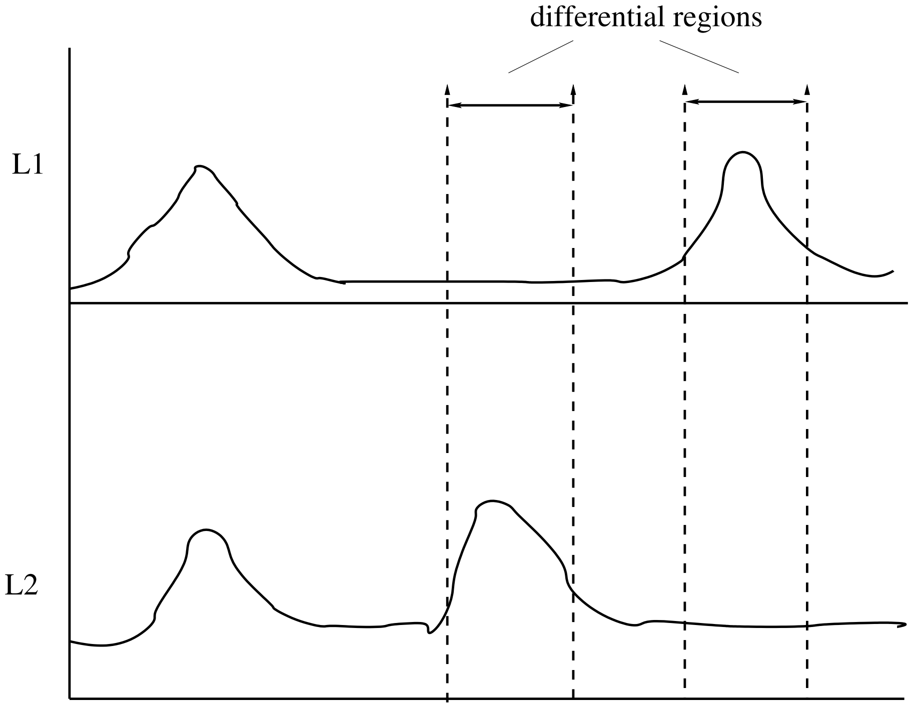


Figure 2.5 – *Profile of two ChIP-Seq data libraries* $L1$ *and* $L2$*. Differential regions in the genome are marked.*

Another hand-drawn example showing two ChIP-Seq libraries $L1$ and $L2$ is shown in Figure 2.5. Regions of the genome where $L1$ is significantly enriched when compared to $L2$ or vice-versa are called differential regions. The problem of detecting such differential regions is what we address in chapter 3.

A real example of a ChIP-Seq library as visualized in the UCSC Genome Browser is shown in Figure 3.1.

2.3.2 Statistical measures

Some of the statistical measures used in this thesis are given below.

Fold change threshold

Fold change thresholds have been used in literature to determine which regions are enriched when comparing ChIP-Seq data from two different cell types. In [125], a fold-change threshold value of 3 was used. We used a fold change threshold for our work in chapter 3.

False discovery rate

False discovery rate (FDR) [9] is a statistical measure which is used in multiple hypothesis testing to correct for multiple comparisons. For the ChIPnorm method we binned the ChIP-Seq data into various bins. FDR estimation is done to find which bins are significantly enriched or not. FDR is estimated as $FDR(t) = \frac{\sum_{\geq t} E_{null}(t)}{\sum_{\geq t} E_{obs}(t)}$ [94], where $E_{obs}(t)$ is the number of observed bins which have bincount values $\geq t$ and $E_{null}(t)$ the number of expected bins which have bincount values $\geq t$. The FDR threshold which is often used in literature is 0.05. More details are given in [94].

p-value

In statistical significance testing, p-value is the probability of obtaining a test statistic result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true [48, 49]. The null hypothesis is rejected if the p-value turns out to be less than a predetermined significance level, often 0.05 [117, 25] or 0.01. p-values are often used to show the significance of ChIP-Seq peaks. The smaller the p-value (or greater the negative log p-value), the more significant the peak is. We use p-values when using the peak data in chapter 5. In the multiple testing problem it may be better to use a tighter p-value threshold so that we do not get a large number of false positives by chance alone. Therefore it is better to adjust the p-values using an optimized FDR approach. the adjusted p-value is called a q-value in literature. More details of this can be found in [118, 119].

IDR

Reproducibility is important for high throughput experiments to be considered reliable. Irreproducible discovery rate (IDR) framework is a unified approach to measure the reproducibility of findings identified from replicate experiments. Unlike the usual scalar measures of reproducibility, the IDR approach creates a curve and quantitatively assesses when the findings are no longer consistent across replicates. The IDR score can be computed at each set of paired replicate ranks and permits the principled setting of thresholds both for assessing reproducibility and combining replicates [73]. The paper [73] reproducibility is described as the extent to which the ranks of the signals are no longer consistent across replicates in decreasing significance. The authors of that paper propose a copula-based graphical tool to visualize the loss of consistency and localize the possible breakdown of consistency empirically. After that reproducibility is quantified by classifying signals into a reproducible and an irreproducible group, using a copula mixture model. Their model then assigns each signal a reproducibility index (which estimates its probability to be reproducible) by jointly modeling the significance of scores on individual replicates and their consistency between replicates. Then based on this index, they define the irreproducible discovery rate (IDR) and a selection procedure, in a fashion analogous to their counterparts in multiple testing, to rank and select signals [73]. IDR is intended to be analogous to FDR [73]. We use some IDR analysis for our work in chapters 4 and 5.

2.3.3 Normalization techniques

Data normalization techniques are very important for us to compare data arising from two different experiments. There are many normalization techniques present in literature, the simplest of those being mean normalization, standard score normalization.

Quantile normalization is one such normalization technique used in statistics, for making two distributions identical in statistical properties. Quantile normalization has often been used in microarray data analysis [1, 16]. One way to do quantile normalization on a test distribution to a reference distribution of the same length is to sort the test distribution and sort the reference distribution; then the highest entry in the test distribution takes the value of the highest entry in the reference distribution, the next highest entry in the reference distribution, and so on, until the test distribution is a perturbation of the reference distribution. If we wish to do quantile normalization on two or more distributions to each other, without a reference distribution, sort as before, then set to the average (usually, arithmetical mean) of the distributions. Now the highest value in all cases becomes the mean of the highest values, the second highest value becomes the mean of the second highest values, and so on. We have adapted the quantile normalization method and used it on ChIP-Seq data (chapter 3).

2.3.4 Expectation-maximization algorithm

The expectation-maximization (EM) algorithm [28] is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables or missing data. The EM algorithm alternates over two steps: guessing a probability distribution over completions of missing data given the current model (known as the E-step) and then re-estimating the model parameters using these completions (known as the M-step). The name 'E-step' comes because one does not usually need to form the probability distribution over completions explicitly, but rather need only to compute 'expected' sufficient statistics over these completions. Also the name 'M-step' comes because model re-estimation can be thought of as 'maximization' of the expected log-likelihood of the data [31].

EM algorithms are widely used in many areas of computational biology [31], like gene expression clustering [30], motif finding [70], haplotype inference problem [37], learning profiles of protein domains [64] and RNA families [34], discovery of transcriptional modules [108], tests of linkage disequilibrium [112], genome segmentation [54], protein identification [92] and medical imaging [27], etc. We have used an EM framework for our probabilistic partitioning approach given in chapter 4.

2.3.5 Phylogenetic tree building methods

Three of the most commonly used phylogenetic tree building framework are: using distance based, maximum parsimony, and maximum likelihood approaches [40]. Some details of each approach are outlined below.

Distance based methods

Distance based methods or distance matrix methods were introduced in [35] and [43]. They were influenced by the clustering algorithms of Sokal and Sneath [113]. The general idea of distance based measures is to calculate a measure of distance between each pair of species, and then find a tree that predicts the observed set of distances as closely as possible [40]. A multiple sequence alignment (MSA) of the DNA or protein sequences are given as input to the distance based algorithms.

One of the most commonly used distance based method is the neighbor-joining algorithm [104]. It takes the distance matrix, specifying the distance between each pair of taxa, as an input. The algorithm starts with completely unresolved tree (star shaped) and iteratively follows a series of steps till the tree is resolved and all branch lengths are known. The simple neighbor-joining method produces unrooted trees. (A rooted phylogenetic tree is a directed tree with a unique node corresponding to the the most recent common ancestor of all the entities at the leaves of the tree. Unrooted tree is tree without a root.) It does not assume a constant rate of evolution (i.e. a molecular clock) across lineages.

Unlike the neighbor-joining method, its relative called the UPGMA (Unweighted Pair Group Method with Arithmetic Mean) which is simple agglomerative (bottom-up) hierarchical clustering method produces rooted trees and requires a constant-rate assumption - that is, it assumes an ultrametric tree in which the distances from the root to every branch tip are equal [114, 40]. Since the UPGMA assumes the molecular clock hypothesis (constant rate of evolution), it is not a well-regarded method for inferring relationships unless this assumption has been tested and justified for the data set being used. The use of neighbor-joining method is thus seen as an advantage as it does not make the molecular clock assumption. The other important feature of neighbor-joining and most other distance based methods is that they are polynomial-time algorithms and are very fast. Neighbor-joining tree also has the property that if the input distance matrix is additive, then the output tree will be a unique correct tree. That is, the distance between any two leaves of the tree is given by the sum of the edges of the path connecting them and it will be same as the corresponding number in the given additive matrix. Also if the distance matrix is “nearly additive”, that is if each entry in the distance matrix differs from the true distance by less than half of the shortest branch length in the tree, then the correctness of the output tree topology given by neighbor-joining method is guaranteed [4]. Although the distance matrix rarely satisfies this condition, it has been seen that neighbor-joining quite often constructs the correct tree topology anyway [84]. We have used neighbor-joining approaches in building cell-type trees in chapters 5 and 6.

Maximum parsimony based methods

Maximum parsimony methods were among the first methods used for inferring phylogenies and their general idea was first described in [35] which declared that the evolutionary tree is to be preferred that involves “the minimum net amount of evolution”. Maximum parsimony predicts the evolutionary tree or trees that minimize the number of steps required to generate the observed variation in the sequences from common ancestral sequences [86]. Therefore this method is also sometimes called minimum evolution method. Maximum parsimony methods are more time consuming and finding the most parsimonious tree is an NP-Hard problem [26]. However given a fixed tree, the Fitch’s algorithm (a dynamic programming approach) can be used to compute the parsimony score and the states of the internal nodes in polynomial time (small parsimony problem) [42]. In some datasets, maximum parsimony methods are found to be better than distance based approaches. The software TNT is a popular parsimony based method to find a phylogenetic tree [47]. We have also used maximum parsimony approaches in building cell-type trees in chapters 5 and 6.

Maximum likelihood based methods

The maximum likelihood method uses underlying probabilistic model on a set of observed sequences and outputs a tree which maximizes the likelihood of the data [121, 40]. The likelihood of the data is basically the conditional probability of producing the data given the model parameters. Usually a substitution model is the probabilistic model used. Maximum likelihood uses a probability of base substitution per evolutionary unit. Maximum likelihood based methods are found to computationally time consuming but usually more accurate than distance based or maximum parsimony based methods [45]. RAxML is a popular method to find phylogenetic tree using a maximum likelihood framework [116].

2.3.6 Minimum spanning tree

Spanning tree of a given connected undirected graph is an acyclic subgraph which connects all the vertices together. Typically there are multiple spanning trees in a single graph. There could be weights assigned to each edge. The sum of the weights of all the edges in that spanning tree gives the weight of the spanning tree. A minimum spanning tree (MST) is a spanning tree with weight less than or equal to the weight of every other spanning tree. Two very common algorithms to find a MST are Prim's algorithm [97] and Kruskal's algorithm [65], both of which are greedy algorithms. We have used MST method in chapter 6.

3 ChIPnorm: a statistical method for normalizing and identifying differential regions in histone modification ChIP-Seq libraries

3.1 Introduction

As we have seen in the earlier chapters, histones are proteins that package the DNA into nucleosomes [72]. These histones are subjected to various types of modifications which alter their interaction with the DNA and nuclear proteins, thereby influencing transcription and genomic function. These modifications along with other epigenetic marks help us understand why various types of cells exhibit very different behaviors in spite of their shared genome. Thus the study of histone modifications, and more particularly of the differential enrichment of these modifications in different cell types, is a crucial tool in the understanding of genomic function. Because of the advances in sequencing technologies, ChIP-Seq has become the main approach for capturing histone modifications, due to its high throughput, high resolution, and low cost [7, 59, 80]. In the ChIP-Seq process, there is first the chromatin immunoprecipitation (ChIP) step, which uses an antibody to isolate DNA fragments in contact with histones that carry a specific modification or transcription factors. Then the sequence of one end of the DNA fragment is read to provide a tag which is then mapped to an assembled genome to determine the location of the DNA fragment.

Genome-wide chromatin maps (using ChIP-Seq technology) for three mouse cell types— embryonic stem (ES) cells, neural progenitor (NP) cells, and embryonic fibroblasts (EF) — have been published [85]. The authors of the paper compared the occurrence of histone-modification sites in promoter regions of the three cell types in a qualitative manner. Subsequently, the first quantitative comparison of two ChIP-Seq libraries using computational techniques appeared [125]; there the authors addressed the problem of finding differential regions given two histone-modification libraries for two different cell types. Their method, ChIPDiff, is based on hidden Markov models. Recently, Taslim *et al.* 2009 [122] proposed a two-step nonlinear normalization method based on locally weighted regression (LOESS) [20] to compare ChIP-Seq data across multiple samples; they modeled the difference using an exponential-normal^K mixture model, then used this fitted model to identify genes associated with differential binding sites. This method first normalizes the data using a locally weighted polynomial least square regression to estimate a LOESS smoother of the mean and variance of the observed data. After normalization they fit a finite mixture model to the normalized data and perform model-based classification to identify genes associated with enrichment regions. The assumption used in the mixture model is that the data comes from three (non-differential, positive- and negative-differential) groups. The non-differential regions are assumed to come from a mixture of K-component normal distributions, where K is unknown and needs to be estimated from the data. The assumption used is that the positive- and negative-differential regions follow exponential and the mirror of exponential distributions, respectively [122]. Using the fitted model they identify genes associated with differential binding sites based on local false discovery rate. However we would like to identify differential regions not just on genes but throughout the genome. Another recent method is RSEG [115]. RSEG identifies epigenomic domains from ChIP-Seq data for histone modifications and is based on hidden Markov model (HMM) framework including the Baum-Welch training and posterior decoding. The method is also able to incorporate a control sample and find genomic regions with differential histone modifications between two samples. From a mathematical viewpoint the problem of finding genomic regions with differential histone modifications between two tissues is not fundamentally different from that of peak finding in ChIP-Seq data using an input control to correct for technical biases. Many such peak finders have been proposed [131]. Some of the other methods present in literature for differential analysis for ChIP-Seq data are DESeq [2], DBChip [74]. The DESeq method assumes the number of reads in a sample that are assigned to a gene can be modeled by a negative binomial distribution, and then the model is fitted to data. This method again works only for finding differential genes and not for all regions in the genome. DBChip is mainly meant for transcription factor ChIP-Seq data. Even after the work done by us in this chapter, some new literature regarding differential analysis has come up in literature. One such method is called MMDiff [107] which detects differential regions taking into account any shape changes of the signals. MMDiff is a multivariate non-parametric approach to testing significant differences in profile patterns between peaks in different conditions and it exploits higher order features in the peak shapes.

A significant impediment to the analysis of ChIP-Seq data is the high level of noise. Noise or systematic distortion can enter at various stages of the procedure: variations in the number of cells used in the experiment, variation in the amount of antibody that attaches to the DNA fragments, tandem repeats, uneven rates of success in sequencing different fragments, etc. The type of histone modification and the cell types can also affect the level of noise. For example, we show in this chapter that histone modification H3K27me3 (K27) in ES cells has less background noise and a better signal-to-noise (S/N) ratio than the same modification in NP cells. In addition, the signal tends to be found mostly in gene-rich regions of the genome. Therefore computational methods may produce many false positives. In the case of modification K27 for ES and NP cells, for instance, false positives are likely in gene-poor regions for NP cells and in gene-rich regions for ES cells. Such bias problems are present in microarray data and many authors have addressed this issue [33]. Similar studies are needed in ChIP-Seq data, as the data characteristics differ [19].

To address the problems of noise and bias in finding differential regions, we propose a two-stage statistical method, called ChIPnorm [88], to remove the noise and the bias from two ChIP-Seq libraries and to normalize the data so as to enable a direct comparison between the two libraries to identify differential regions. Our normalization step is similar to quantile normalization [16]; however we have simplified the method so that it can be readily extended for normalization of more than two libraries. Our method is computationally efficient and can be applied to very large datasets. We use it to analyze ChIP-Seq histone modification data from different types of mouse and human cells, confirming previous findings and making some new observations. We also point to the usefulness of our approach to study histone modification data from Arabidopsis in the context of leaf senescence.

3.2 Methods

To motivate our work, we examine data for histone modification H3K27me3 (K27) in mouse ES and NP cells [85]. Figure 3.1 displays a window of the data mapped onto the mouse genome through the UCSC genome browser [61]. ES data has better S/N ratio as well as more peaks in gene-rich regions than in gene-poor regions (as seen visually from Figure 3.1 and quantitatively from Figure 3.7). These characteristics introduce a bias that must be eliminated before comparing ES data to NP data, as can be seen in the results of the ChIPDiff method [125] in the same figure: most of the differentially NP enriched regions proposed by ChIPDiff fall within gene-poor regions and are almost certainly false positives.

In the following, we use a notation similar to that of Xu *et al.* 2008 [125]. In particular, we assume that the data has been processed by dividing the genome into bins and collecting, for each bin, a count of the mapped sequence tags that fall within the bin. The result is a “library", which is simply a list of non-negative integers, each successive integer associated with the next bin. Let L_a and L_b be two libraries containing the same histone modification for two different cell types—in our example, modification K27 for ES and NP cell types. Let m be the total number of bins in the library and set $Y_a = \{y_{ai} | 1 \le i \le m\}$ and $Y_b = \{y_{bi} | 1 \le i \le m\}$ to be the observed counts of the ChIP-Seq fragments for libraries L_a and L_b respectively, where y_{ai}, respectively y_{bi}, are the sum of the fragments lying in the ith bin. In ChIP-Seq, a tag is retrieved by sequencing one end of the ChIP fragment, and the median length of this fragment is around 200 bp [7, 99]. As was done in Xu *et al.* 2008 [125], we approximate the center of each fragment by shifting the tag end position by 100 bp downstream or upstream, according to its orientation on the chromosome. (We note that instead of using a fixed value like 100 bp to approximate the center of each fragment, one could also infer a proper value separately for

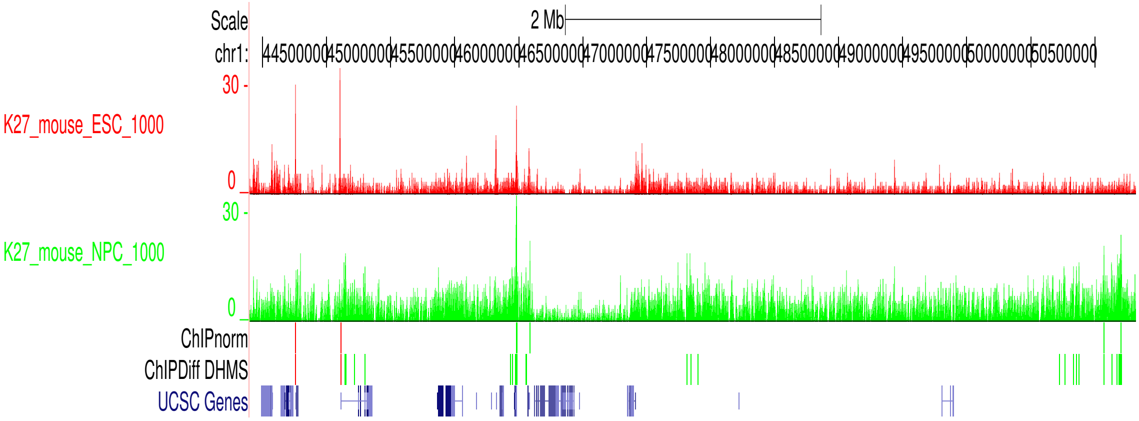


Figure 3.1 – *Histone modification profile as seen in the UCSC genome browser [61]. Tracks 1 (red) and 2 (green) show the H3K27me3 modifications for ES and NP cells, respectively. Track 3 shows the differentially enriched regions found by our ChIPnorm method. Track 4 shows the differentially enriched regions found by the ChIPDiff method [125]. In tracks 3 and 4, red indicates differential enrichment in ES cells and green indicates differential enrichment in NP cells. Track 5 shows the UCSC genes.*

each dataset — however for simplicity we choose a fixed value of 100 bp for this work.) We choose different bin sizes for different types of the histone modifications so as to maximize the discriminative information between the two libraries of the different cell types and minimize the discrimination of the two replicates of the same libraries. We use the spread of the data in the scatter plots as a measure for discriminative information. A lower bin size favors a better spread (away from the diagonal) between the data of the two cell types in the scatter plot. (For robustness studies we also tried using a range of bin sizes, as shown in results section.)

An observed fragment count y_{ai} at the ith bin can be related to the actual number of histone modifications x_{ai} at the ith bin using the following model:

$$\begin{aligned} y_{ai} &= g_a(x_{ai}) + \varepsilon_{ai} + \nu_{ai} \\ y_{bi} &= g_b(x_{bi}) + \varepsilon_{bi} + \nu_{bi} \end{aligned} \tag{3.1}$$

Function g is the (unknown) deterministic function that describes the nonlinear transformation of the actual histone modifications, accounting for the various experimental conditions that may influence the observations in a systematic way. The additive ε term accounts for the stochastic (background) noise introduced by the experimental setup, such as stray fragments from neighbouring modifications. Finally, the parameter ν accounts for local genomic bias, mainly bias due to open chromatin regions and mapability, such noise is common in both the actual ChIP-Seq library and the corresponding control dataset. Naturally, one could choose a stochastic, rather than a deterministic model for g; but since our goal is to detect regions with differential enrichment, and not to produce a detailed predictive model, the deterministic approach suffices.

3.2.1 ChIPnorm scheme

We will address each source of error separately and proceed in two main stages. In the first stage, we address the removal of stochastic background noise and local genomic bias in each library. In the second stage, we address the problem of normalization.

3.2.2 Stochastic noise ε

To solve the problem of stochastic background noise, both Bayesian modelling methods [36] and statistical confidence measure methods have been used. In terms of statistical confidence, the problem reduces to evaluating the probability that a particular bincount (bincount is the total number of DNA fragments captured inside a bin of some size) would occur by chance [94]. We estimate this probability by defining a "null hypothesis", which is a random distribution of bincounts, and then comparing it with the distribution of bincounts of the ChIP-Seq library.

To understand the rationale for choosing the amplified binomial distribution (ABD) as the random distribution for the bincounts of ChIP-Seq library, we must examine the ChIP-Seq process as illustrated in Figure 3.2. The short segments of DNA are treated with a specific antibody to capture a particular histone modification in the genome. The rest of the fragments are washed away. The captured fragments are sequenced by a high-throughput sequencing method, which typically uses PCR amplification of the captured fragments [52] before performing the final base-pair sequencing. The sequenced data is binned to obtain a ChIP-Seq library.

To estimate the null distribution of a ChIP-Seq library, we assume that the fragmented DNA from the sonicated whole cell is treated with an antibody that randomly captures fragments without any specificity. The bincounts of captured fragments then follow a binomial distribution. The captured fragments are amplified, sequenced, and binned to get the null distribution of the ChIP-Seq library. This binned data follows a random distribution, in which each of the fragments following the binomial distribution is amplified; we refer to it as the amplified binomial distribution (ABD). We assume that each fragment is amplified by a constant amplification factor.

An ABD can be defined by two parameters: the total number of fragments N_{bino} in its corresponding binomial distribution and the amplification factor α. We made two assumptions to calculate these parameters for the desired null distribution: (a) the total number of the fragments in the original ChIP-Seq data N_{data} and in the corresponding null distribution $N_{null} = \alpha \times N_{bino}$ are the same, and (b) the total number of bins with zero bincount is the same in the ChIP-Seq library (Z_{data}) and the null distribution. Since the amplification does not change the number of bins with bin-count zero, we can write $Z_{bino} = Z_{data} = \left(1 - \frac{1}{B}\right)^{N_{bino}} B$, where B is total number of bins in the library. Since Z_{data} and B are observed variables, N_{bino} and $\alpha = \frac{N_{data}}{N_{bino}}$ can be evaluated. Now the probability mass function of the random distribution

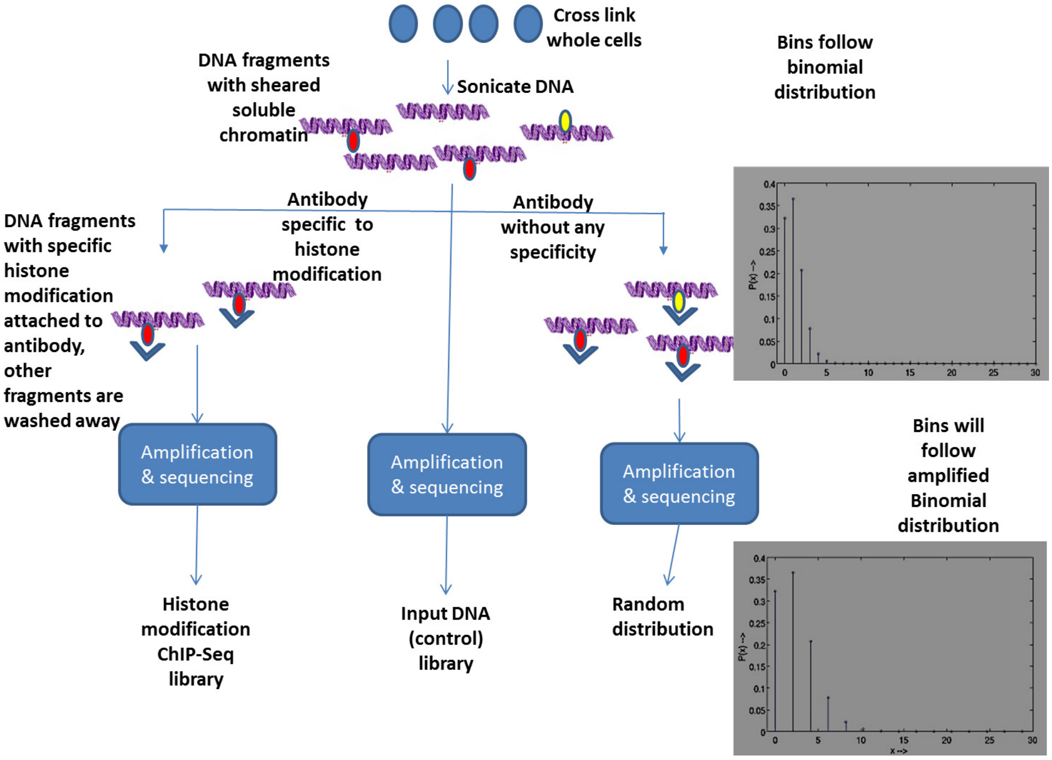

Cross link whole cells
Sonicate DNA
DNA fragments with sheared soluble chromatin
Bins follow binomial distribution
Antibody specific to histone modification
Antibody without any specificity
DNA fragments with specific histone modification attached to antibody, other fragments are washed away
Amplification & sequencing
Amplification & sequencing
Amplification & sequencing
Histone modification ChIP-Seq library
Input DNA (control) library
Random distribution
Bins will follow amplified Binomial distribution
P(x) -->
x -->
0
5
10
15
20
25
30
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4

Figure 3.2 – *Overview of ChIP-Seq process. We see how we can get the ChIP-Seq library, input DNA control, and the random distribution (null hypothesis).*

prior to amplification is

$$P_{random}(t) = \binom{N_{bino}}{t}\left(\frac{1}{B}\right)^t\left(1-\frac{1}{B}\right)^{N_{bino}-t}, \quad 0 \le t \le N_{bino}.$$

The ABD is thus estimated as $P_{ABD}(t) = P_{random}(t/\alpha)$.

The estimated real distribution of the original ChIP-Seq data for one chromosome is shown in Fig. 3.3 (blue). An example of random distribution (amplified binomial distribution with $\alpha = 2$) is also shown (red). The figure is zoomed into a small region for clarity. We can see that since the number of zero bins are same, the probability that bincount is 0 is same for both. The random distribution tends to have a higher probability than real distribution at lower bincounts but a lower probability at higher bincounts.

We calculated the false discovery rate (FDR) for a ChIP-Seq data using its corresponding estimated null distribution (ABD) [94]. We declared bins with FDR ≤ 0.05 as significant bins in the ChIP-Seq library. The value 0.05 is the standard value used in the field [94].

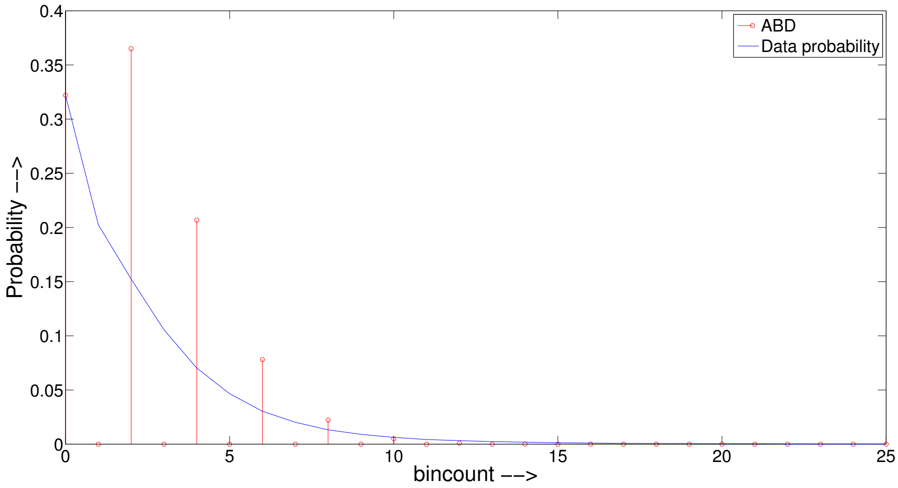


Figure 3.3 – *An example of random distribution (amplified binomial distribution with* $\alpha = 2$*) is shown in red and the estimated real distribution of the original ChIP-Seq data for one chromosome is also shown (blue). ABD — amplified binomial distribution. Data probability — estimated real distribution. The figure is zoomed into a small region for clarity.*

3.2.3 Local genomic bias ν

ChIP-Seq data contains many local genomic biases corresponding to open chromatin regions, over-amplified satellite repeats, GC-rich regions, and unmappable perfect repeat regions. Some of these biases depend on experimental conditions and others may vary among the cell lines in a systematic manner. If such cell-specific biases are not taken care of while comparing ChIP-Seq libraries it will give many false positive differential regions. Some of these biases can be reduced by using an input DNA control library. To find differential regions, we must consider only those regions that are significantly enriched with respect to the input DNA control. The input DNA control are the DNA fragments in the ChIP-Seq experiment prior to the application of the histone specific antibody. To find the enriched bins, we need to normalize the input DNA control with respect to the data.

Yang *et al.* 2002 [127] recommended the use of locally weighted regression (LOESS) normalization. The basic assumption is that the percentage of the differential sites, considered as outliers by LOESS, is small so that these sites do not affect the normalization. However, this is not true in the case of a ChIP-Seq library. In a ChIP-Seq library, the percentage of bins that are differentially enriched relative to the input DNA control can exceed 50%. These differentially enriched bins will affect the LOESS normalization as shown in Figure 3.4(a) and lead to many false negatives. We introduce an iterative normalization to overcome this problem.

In the first stage of iterative normalization, we normalize the DNA control with respect to the data using quantile normalization. For illustration purposes, we first show the LOESS curve of the MA plot in 3.4(a). (MA plot is often used for visual representation of two channel DNA microarray gene expression data [33] which has been transformed onto the M (log ratios)

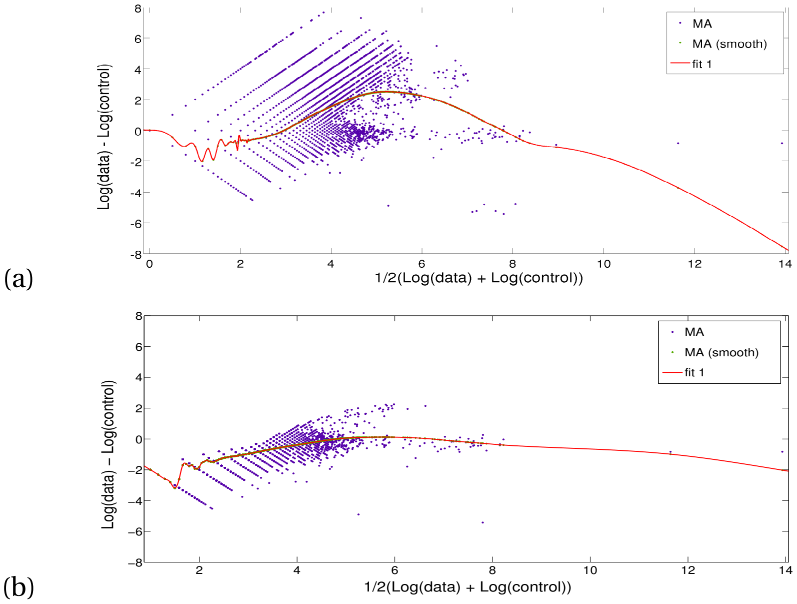


Figure 3.4 – *Iterative Normalization of input DNA. (a) before first iteration. (b) after first iteration, post removal of outliers.*

and A (mean average) scale.) We first classify bins enriched with respect to the DNA control (fold change > 1) after quantile normalization as outliers. These outliers are then removed from further iterations. Figure 3.4(b) shows the LOESS curve after removing the outliers. We see that the LOESS curve is relatively less affected by outliers. In the next iteration, we use non-outliers bins for second quantile normalization to get a more accurate estimate of the normalization function. Once we get this normalization function, we normalize (quantile) all bins. The process of removing outliers and then performing normalization can be repeated to rescue more bins falsely declared as non-enriched. Each bin is declared as enriched if its fold-change value of ChIP-Seq data and quantile normalized control data is > 1. (The LOESS normalization is not used in the ChIPnorm method but is shown here only for illustrating the affects of normalization and removing outliers. Instead quantile normalization is used.)

Bins which are declared both as significant in the ABD approach and as enriched in the iterative approach, are declared as 'enriched-significant'. Bins which are declared as enriched-significant in either of the two libraries are passed on to the second stage, with their original bincount values.

3.2.4 Quantile normalization

Since our first stage removed the majority of bins with low S/N ratios and genomic bias, and since we expect interesting regions to have good S/N ratios, we make the simplifying assumption:

$$\begin{aligned} \varepsilon_{ai} &= \varepsilon_{bi} = 0 \\ \nu_{ai} &= \nu_{bi} = 0; \end{aligned} \tag{3.2}$$

With this assumption the next step is to normalize the data to the same scale so that bin values in the two libraries are comparable. We propose a quantile normalization method (similar to Bolstad *et al.* 2003 [16]) to solve this normalization problem. Quantile normalization assumes that the distribution of the data of the two libraries that are being compared are similar. This may seem problematic because histone modifications change significantly during differentiation. But it is reasonable to assume that their probability distribution of the bincount over the whole genome is similar across different cell types. (This might not be true if one of the two libraries have histone modifications knocked out. This may also not be a valid assumption if the development stage includes a major redistribution of a given histone modification, for example the marking or unmarking of large portions of the genome.) We formulate the problem mathematically as follows. Given two observed data $y_{ai} = g_a(x_{ai})$ and $y_{bi} = g_b(x_{bi})$, find a transformation $f^* = (g_a \circ g_b^{-1})$ such that $y^*_{bi} = f^*(y_{bi}) = g_a(x_{bi})$. We make the following assumptions:

- The actual histone modifications $X_a = \{x_{ai} \mid i \in \{1, m\}\}$ and $X_b = \{x_{bi} \mid i \in \{1, m\}\}$ follow the same distribution, i.e., we have $F_{X_a}(x) = F_{X_b}(x)$.
- Cumulative distribution functions (cdf) F_{X_a} and F_{X_b} are monotonically increasing.
- $g_a()$ and $g_b()$ are monotonically increasing.

The last two conditions imply that F_{Y_a} and F_{Y_b} are also monotonically increasing.

Theorem 1
Any function $\hat{f}\colon y_{bi} \to \hat{y}_{bi}$ satisfies $F_{\hat{Y}_b}(y) = F_{Y_a}(y)$ if and only if we have $\hat{f} = f^*$.

Lemma 1 $\quad F_{Y_b^*}(y) = F_{Y_a}(y)$

Proof (of lemma):
$F_{Y_a}(y) = P(Y_a \leq y) = P(g_a(X_a) \leq y) = P(X_a \leq g_a^{-1}(y)) = F_{X_a}(g_a^{-1}(y))$ and
$F_{Y_b^*}(y) = P(Y_b^* \leq y) = P(g_a(X_b) \leq y) = P(X_b \leq g_a^{-1}(y)) = F_{X_b}(g_a^{-1}(y)) = F_{X_a}(g_a^{-1}(y))$

Proof (of theorem):
only if part: if we have $\hat{f} = f^*$, then from Lemma 1, we also have $F_{\hat{Y}_b}(y) = F_{Y_b^*}(y) = F_{Y_a}(y)$.

if part: if we have $F_{\hat{Y}_b}(y) = F_{Y_a}(y)$, then from Lemma 1, we also have $F_{Y_b^*}(y) = F_{\hat{Y}_b}(y)$. Additionally, as cdfs are assumed to be monotonically increasing, they are one-to-one functions. Hence we can write $\forall i$, $\hat{y}_{bi} = y^*_{bi}$, which in turn implies $\hat{f} = f^*$.

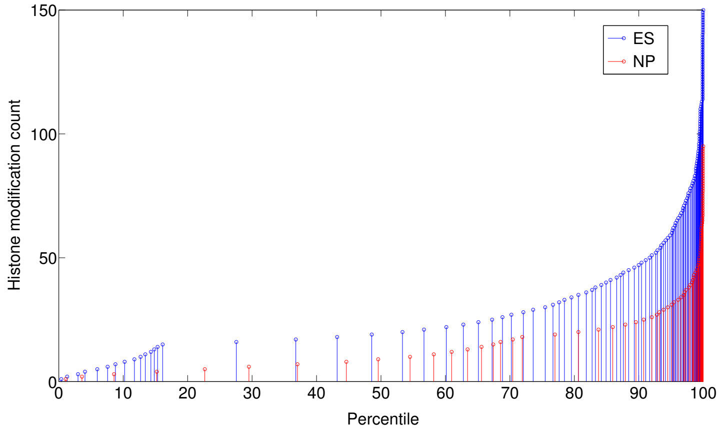


Figure 3.5 – *The new inverse cumulative distribution function on the modified libraries (after stage 1). On the x axis is the percentile, on the y axis are the bin values.*

Theorem 1 states that if (i) we have $F_{X_a}(x) = F_{X_b}(x)$, (ii) the cumulative density functions of X_a and X_b are identical and monotonically increasing, and (iii) $g_a()$ and $g_b()$ are deterministic monotonic increasing functions, then any transformation that meets the conditions of the theorem is our desired transformation $f^* = (g_a \circ g_b^{-1})$.

To find such a transformation, we use the inverse cumulative distribution function (on the modified data after removing noisy bins) of the enrichment level, as shown in Figure 3.5. The x axis of this figure is the percentile while the y axis is the bin values. The figure shows the L_a and L_b bin values plotted against their cumulative percentile. To get the desired transformation of Y_b, we must ensure that the post-transformation data $\hat{Y}_b$ follows the same cdf as Y_a. We fit a spline smoothing function on the bin values of library L_a, then, for all percentile values p, we perform a transformation $\hat{f}: y_b \rightarrow \hat{y}_b$ such that $\hat{y}_b(p) = y_a(p)$. The transformation $\hat{f}$ ensures that the conditions of Theorem 1 are met.

This transformation reduces the problem of comparing two libraries with different probability distributions to the problem of comparing two libraries following the same probability distribution, so that a direct comparison of values can now be used. Since in the second stage we considered bins which were declared as enriched-significant in either of the two libraries L_a and L_b (a union operation), some bins which are not declared as enriched-significant would be present in the second stage too. If both libraries were completely independent events, we would expect 50% of the bins to be enriched-significant, because of the union operation. In effect, we define a bin in library L_b to be differentially enriched for the target modification if (i) observed bin value in libraries L_b lies above the 50% region in the inverse cumulative distribution function and (ii) for some chosen fold change threshold τ (> 1), we have $\hat{y}_{bi} / y_{ai} \geq \tau$. Similarly we can define a bin in library L_a as differentially enriched if the bin value in library L_a lies above the 50% region in the inverse cumulative distribution function and for for some chosen threshold τ (> 1), we have $y_{ai} / \hat{y}_{bi} \geq \tau$. All bins are thus reported as differentially enriched or not. Adjacent bins of the same type of differential enrichment can be grouped together to form differential regions (DHE).

3.2.5 The complete ChIPnorm method summarized

The complete ChIPnorm method is summarized in the Figure 3.6. In the first stage, we identify bins having a significant bincount compared to the estimated random distribution of a ChIP-Seq library as significant bins, by using a false discovery rate (FDR) analysis. We also identify bins of a ChIP-Seq library as enriched bins, if their bincounts are higher than the corresponding bincounts of the normalized input DNA control. Those bins which are both significant w.r.t. null hypothesis and enriched w.r.t. normalized input control DNA are declared as enriched-significant bins. Bins which are declared as enriched-significant in either of these two libraries are passed to the second stage. In the second stage, we normalize the enriched-significant bincounts of the two ChIP-Seq libraries and use a fold change to obtain differentially enriched bins.

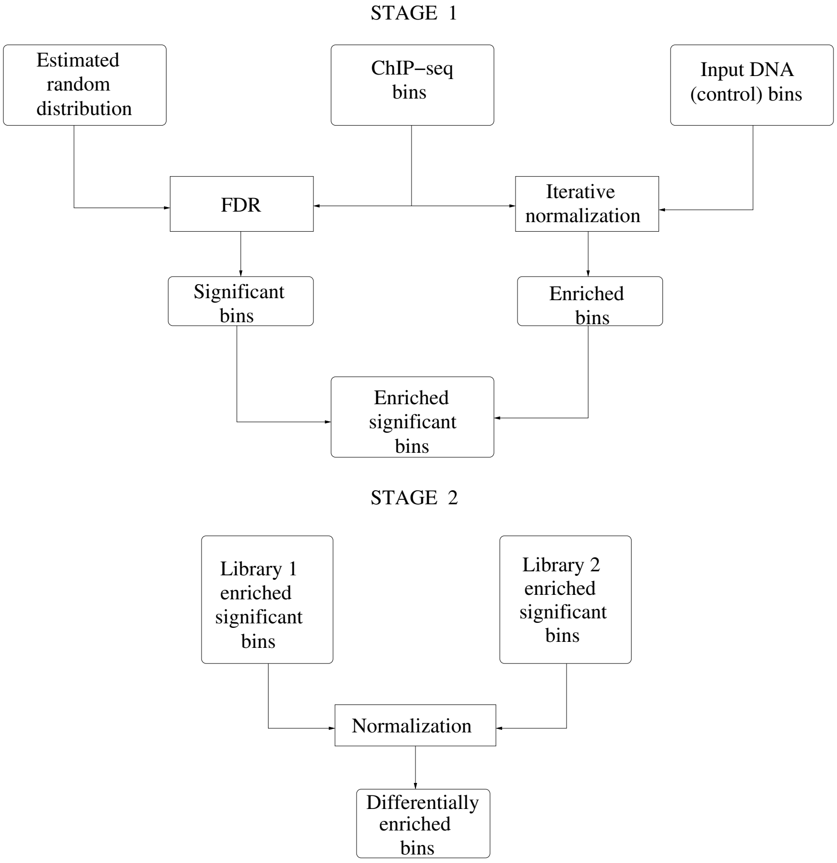

STAGE 1
Estimated random distribution
ChIP–seq bins
Input DNA (control) bins
FDR
Iterative normalization
Significant bins
Enriched bins
Enriched significant bins
STAGE 2
Library 1 enriched significant bins
Library 2 enriched significant bins
Normalization
Differentially enriched bins

Figure 3.6 – *The schematic diagram of the ChIPnorm method. In the first stage one we find the enriched-significant bins by removing various kinds of errors in the data. In the second stage we normalize the two ChIP-Seq libraries and find differentially enriched bins.*

The normalization can also be used to find bins that are enriched in both libraries, thereafter called constitutively highly enriched (CHE). Bins which are above 50% in L_a and L_b in the inverse cumulative distribution function and also below threshold τ are declared as CHE. While differential histone modification enrichment (DHE) regions help us understand why different types of cells behave differently, CHE regions are conserved between the cell types and thus presumably essential to the survival of both types.

The normalization method also facilitates the comparison of more than two libraries. Our method is easily extended to handle multiple types of histone modifications in multiple cell types. Such analyses can give more insight into combinatorial patterns of histone modifications, sometimes referred to as "histone language" [120]. For example, Bernstein *et al.* 2006 [11] hypothesized that a bivalent domain with both H3K27me3 and H3K4me3 modification at the same site plays a crucial functional role in embryonic stem cells. Finally, the ChIPnorm method can be used with any ChIP-Seq data—not just with histone modifications.

3.3 Experimental Design

We carried out a series of experiments with the two libraries for H3K27me3 and H3K4me3 histone modifications (ES and NP cells), including experiments for bias and sensitivity. Since H3K4me3 has sharper peaks than H3K27me3, it needs a finer resolution, and smaller bin sizes are used. Using the replicate data analysis described earlier, we chose a bin size of 1000 bp for H3K27me3 (K27) and a bin size of 200 bp for H3K4me3 (K4). The bin size of 1000 bp for H3K27me3 has also been used previously in literature [125]. We compared ChIPnorm with six other normalization methods: (a) unit mean normalization; (b) quantile normalization; (c) MACS peak finder; (d) ChIPDiff method [125]; (e) rank normalization; and (f) two-stage unit mean normalization. We ran these methods on the H3K27me3 data for ES and NP mouse cells provided by Mikkelsen *et al.* 2007 [85] (with whole cell extract (WCE) control library) and on the H3K27me3 data (of Broad Institute) for ES and GM12878 (replicate 1) from the human ENCODE project [13, 100]. (GM12878 is a lymphoblastoid cell line produced from the blood of a female donor with northern and western European ancestry by EBV transformation [13].) Processing was done on individual chromosomes of the two libraries.

The five methods not yet described are as follows:

- *unit mean normalization*: is the standard Affymetrix scaling method for microarray data [16]. To perform consistent comparison with the ChIPnorm method, we normalized the two libraries to have unit mean using a method similar to the Affymetrix scaling method. To normalize the bin values x_i of a library we calculated its trimmed mean $\bar{x}$ (the mean of the non-zero bins in the library) and then the normalized bin value is set to $x'_i = x_i/\bar{x}$. Finally a threshold (τ) was used to classify bins as differential or not.
- *quantile normalization*: the two libraries are quantile normalized, and a fold change threshold (τ) is used to classify bins as differential or not.
- *MACS peak finder method*: Although MACS is a peak-finding software [131], we use it indirectly to find differential regions as follows: peaks for one library are detected by giving the other library as control, and the bins with peaks are considered as differential regions.

The version of MACS software used is “macs14 1.4.1 20110622”.

- *rank normalization*: the bin values of each of the libraries are sorted separately; the sorted lists are divided into 10 equal partitions, which we define as rank. Finally we compare the values of corresponding ranks at each bin value in both libraries. If the difference between the values is greater than a threshold v then the bin is classified as differential.
- *RSEG method:* RSEG is a recently published method [115] to not only find peaks in histone modifications but to also identify differential regions (rseg-diff) between two histone modification ChIP-Seq libraries.
- *two-stage unit mean normalization*: we removed the noisy bins using the first stage of the ChIPnorm method before applying the unit mean normalization and fold change classification.

For methods unit mean normalization, quantile normalization, rank normalization, two-stage unit mean normalization, ChIPnorm, the fold change ratios where calculated by adding +1 on the numerator and denominator before calculating the ratio so as to avoid the divide-by-zero case. For the sensitivity analysis using the ENCODE data the corresponding gene expression data (RPKM - Reads Per Kilobase of exon model per Million mapped reads) is obtained from the ENCODE Caltech RNA-seq database [13, 100]. To calculate four-fold gene expression ratio for the ENCODE human data, the RPKM values of the two libraries were required to be normalized so that they are comparable to each other. So RPKM values of the library L_b was normalized by dividing it by the sum of all the RPKM values (of all genes) of library L_b and multiplying it by sum of all the RPKM values (of all genes) of library L_a. A small offset value of 5 was added to the RPKM values of each library before taking the fold change ratio so as to avoid division by zero or very small values. This is a common procedure and some other values of offset could also be chosen as it does not bias the results.

For the correlation with gene expression studies and the bivalent analysis studies, we classified the genes from Mikkelsen *et al.* 2007 [85] into the five groups (A-E) based on their increasing log-ratio of their expression levels in ES and NP cells. To ensure that each group has a good representation in terms of the number of genes, we created a histogram of the differential gene expression and divide them into $< -6\sigma$, -6σ to -2σ, -2σ to 2σ, 2σ to 6σ, $> 6\sigma$, from the mean, where σ is the standard deviation.

3.4 Results

Our results are of two kinds. First, we present the characteristics of the ChIPnorm method and compare it to various other methods for normalization. Next, we use the ChIPnorm method to investigate the libraries of various cell types, both to confirm existing findings and to evince new correlations.

3.4.1 Comparative Analysis

Bias with respect to gene density

We now look how the number of histone modifications change with respect to gene density. Basically gene-rich regions are those regions of the genome where there are large number of genes. Even regions of the genome which are upstream and downstream of genes will fall into gene-rich/poor regions depending on the number of genes in that region. Usually the number of histone modifications is comparatively small in long stretches of the genome which have less or no genes (inactive regions). Gene poor regions have a high number of inactive regions. We noticed that most of the earlier methods for comparing two libraries suffer from bias with respect to gene density. In order to study this bias quantitatively, we divided the whole genome (data from Mikkelsen *et al.* 2007 [85]) into regions of size 1 Mbp each. The size of 1 Mbp was chosen so that there are sufficient number of genes within each region and also so that each region is not too big. Each of these regions is then classified into 10 classes according to the number of genes present in that region. Then we compared the number of bins declared enriched by previous methods and by the ChIPnorm method. Histone modifications like H3K27me3 and H3K4me3 mostly occur near the promoter regions of genes. Therefore there should be more differentially enriched regions in gene-rich regions than in gene-poor regions. First we give evidence that there are more histone modifications in gene-rich regions than gene-poor regions. Figure 3.7(a) shows the total number of H3K27me3 ChIP-Seq fragments divided by the number of Mbp regions (counts per megabase) found in each gene density. We see from the figure that there is an increasing trend of ChIP-Seq fragments with gene density. This is true for both ES and NP cells. We also notice that the curve for ES cells is steeper than the curve for NP cells. This shows that NP cells have more background noise than ES cells and a lower signal-to-noise ratio. We think that the positive correlation of H3K27me3 levels and gene density is biologically meaningful: the more genes, the more gene regulatory regions that are potential targets of H3K27me3-mediated repression. From this perspective, the reversal of the trend in the top two gene-rich bins is due to the stronger “signal” (higher number of ChIP-Seq fragments originating from truly H3K27me3-enriched regions) in ES cells.

Now we show the bias of various methods with respect to gene density. Figure 3.8 shows that other methods, namely unit-mean normalization (3.8(a)), quantile normalization (3.8(b)), ChIPDiff (3.8(d)), rank normalization (3.8(e)), and RSEG method (3.8(f)) all follow the trend for ES differentially enriched bins, but show an opposite trend for NP differentially enriched bins (For MACS peak-finder method (3.8(c)) the trend for NP with respect to gene density is not exactly opposite but more random). Because of the increased background noise and lower signal-to-noise ratio in NP data compared to ES data, these methods incorrectly show an increased number of H3K27me3 NP differentially enriched regions in gene-poor regions. This is due to the ineffective normalization techniques. This ineffectiveness is removed by the first stage of the ChIPnorm approach, as it removes the noisy or insignificant bins for each ChIP-Seq library separately. The unit-mean and quantile normalization (Figure 3.8 (a) and (b)) methods shows decreasing trends in gene density for NP cells, but after applying the

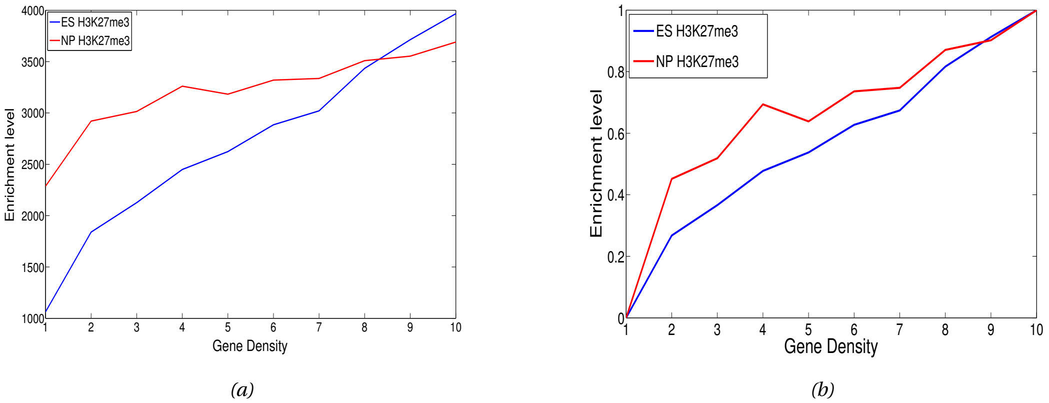


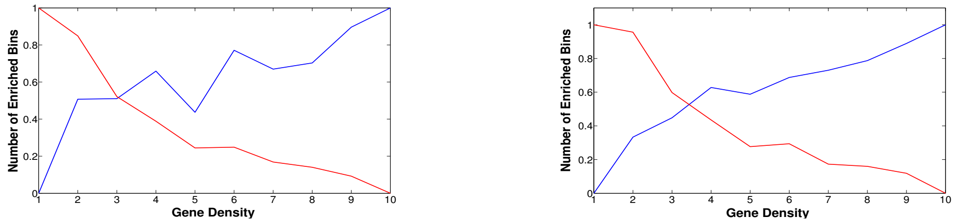
Figure 3.7 – *Enrichment level of bins with respect to gene density in a 1 Mbp region. The x axis indicates lowest to highest gene density. (a) The y axis indicates the total number of H3K27me3 ChIP-Seq fragments divided by the number of Mbp regions (counts per megabase) found in each gene density. (b) The plots are re-normalized so that the y axis range is same for both ES and NP cell data. We see that the enrichment level of ChIP-Seq data increases with respect to gene density for both ES and NP cells.*

first stage of ChIPnorm, this trend is reversed (Figure 3.8 (g) and (h) for two-stage unit-mean normalization and ChIPnorm). This shows the importance of the first stage of the ChIPnorm method, which is common to both these approaches.

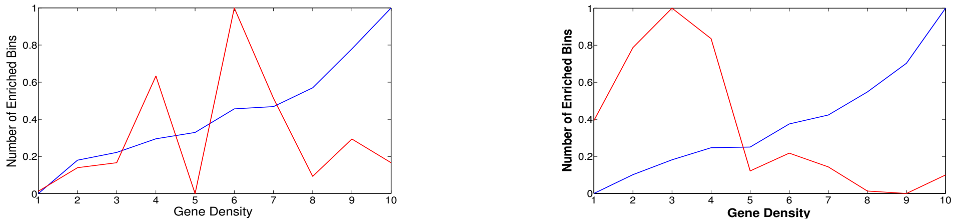
Sensitivity analysis

From the 13,438 genes whose microarray gene expression data is available in Mikkelsen *et al.* 2007 [85], we selected genes for which expression levels are at least four-fold upregulated in ES cells compared to NP cells, and vice-versa. Out of 13,438 genes, 925 genes were at least four-fold differentially over-expressed in NP cells, and 1104 genes were at least four-fold differentially over-expressed in ES cells. We carried out sensitivity analysis on these genes on the various methods. Since K27 is thought to be a gene repressor [7, 85], we expect that there are more ES enriched differential regions in the promoter region (which we define here as ± 1 kbp of the transcription start site (TSS)) of those genes which are over-expressed in NP cells, and more NP enriched differential regions in the promoter region of genes which are over-expressed in ES cells. Note however that we do not expect a high correlation in such a test, as H3K27me3 modification of histone H3 is only one of several mechanisms of gene repression. We are also aware of recent work that questions the exclusively repressive role of H3K27me3 in gene regulation. Young *et al.* 2011 [130] identified a new subclass of H3K27me3-marked genes, which are highly expressed. However, as these genes were reported to have unchanged expression levels between ES and NP cells in the same paper, they are unlikely to interfere with our evaluation protocol.

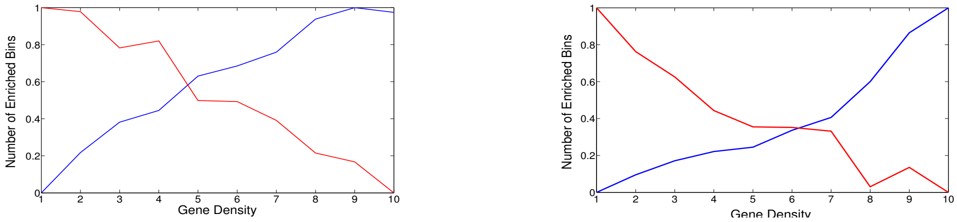
The results of our sensitivity analysis are summarized in Table 3.1 (data from Mikkelsen *et al.* 2007 [85]). The TSS positions were taken from the "knownGene" track of the UCSC genome browser. For each gene all the promoters were considered. Experiment "sensitivity (ES K27-



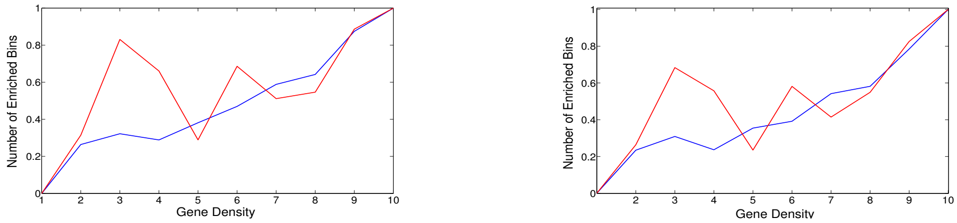
(a) unit mean normalization *(b) quantile normalization*



(c) MACs peak finder *(d) ChIPDiff*



(e) rank *(f) RSEG*



(g) two-stage unit mean normalization *(h) ChIPnorm*

Figure 3.8 – *Enriched bins with respect to gene density in 1 Mbp region. The plots are normalized. x axis 1 to 10 indicates lowest to highest gene density, while y axis 0 to 1 indicates minimum to maximum average number of differentially enriched bins for both ES and NP cells. Blue line indicates ES differentially enriched bins and red line indicates NP differentially enriched bins. The number of enriched bins per 1 Mbp should increase with gene density.*

enriched)" shows the percentage of ES differentially enriched regions around the TSS ± 1 kbp of the 925 genes which are at least four-fold over-expressed in NP cells. Experiment "error (NP K27-enriched)" was done on the same 925 genes as experiment "sensitivity (ES K27-enriched)", but these regions were erroneously declared as NP-enriched instead of being declared as ES-enriched. Likewise, "sensitivity (NP K27-enriched)" and "error (ES K27-enriched)" was determined for the 1104 genes over-expressed in ES cells. To compare the output of the various methods we fixed the parameters so that each of the methods give similar sensitivity for ES K27-enriched regions (≈ 15%). The MACS peak-finder method could not be made to get a sensitivity close to 15% by changing the p-value threshold. We see that all methods give a low "error (NP K27-enriched)" rate which shows that all methods have a good one-sided accuracy. However, we see that for the same thresholds, unit-mean, quantile normalization, MACS peak-finder and rank normalization methods show very small sensitivity for ES differentially over-expressed genes ("sensitivity (NP K27-enriched)") and a corresponding high error rate ("error (ES K27-enriched)"). In fact, we find that the error rates are higher in the quantile normalization method, the MACS peak-finder method, the ChIPDiff method, the RSEG method, than the sensitivity for ES differentially over-expressed genes. This clearly shows a bias towards ES enrichment in promoter regions. However, for the two-stage unit mean and the ChIPnorm method, the problem of bias disappears, as these methods give a higher sensitivity and lower error rate on both the ES and NP differentially over-expressed genes. The reason the ChIPnorm method, improves over the other methods is because the first stage of the ChIPnorm approach, removes the noisy regions, while the second stage which uses quantile normalization, transforms one distribution of one library to that of the other library, thereby reducing the differences in the amplification factors and the SNR of the two libraries. The two-stage unit mean method also works well showing the importance of the first stage that we used, which is common in both these methods.

Table 3.1 – *Sensitivity analysis percentages using various methods (data from Mikkelsen* et al. *2007 [85]). Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The parameters of all the methods (except for MACS as explained in the text) were adjusted so that all of them give almost the same percentage (≈* 15%*) of experiment "sensitivity (ES K27-enriched)".*

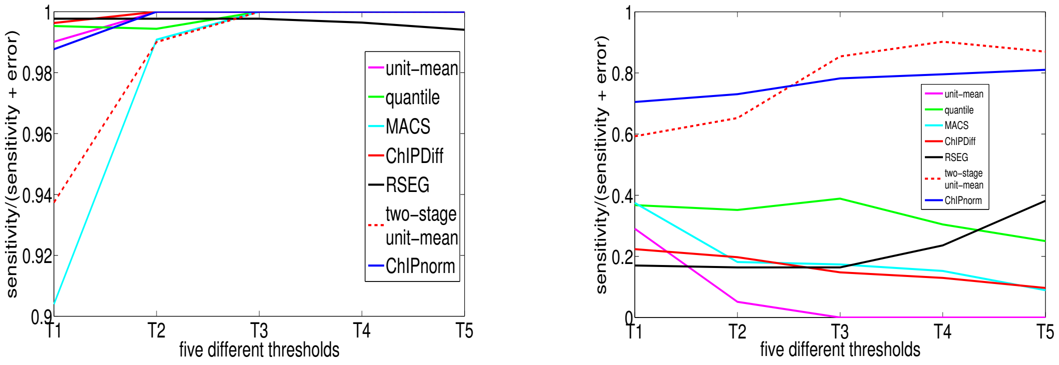
	unit-mean	quantile	MACS	ChIPDiff	rank	RSEG	two-stage unit-mean	ChIPnorm
thresholds	$\tau = 7$	$\tau = 15$	p-val = 10^{-16}	$\tau = 7$	$\nu = 7.5$	cdf = 0.95	$\tau = 1.98$	$\tau = 3$
NP differential (four-fold) expressed: 925 genes								
sensitivity (ES K27-enriched)	14.49	15.24	31.24	14.64	17.62	18.16	15.14	15.14
error (NP K27-enriched)	0	0	0	0.27	6.05	0.11	0	0
ES differential (four-fold) expressed: 1104 genes								
sensitivity (NP K27-enriched)	0	1.27	0.27	0	12.59	1.9	6.88	7.16
error (ES K27-enriched)	1.27	1.99	5.07	0.91	5.43	3.08	1.18	1.99

Since the above sensitivity analysis was done by fixing the sensitivity value to 15%, we tested the various methods over a wide range of thresholds. Figure 3.9 gives the plot of $\frac{\text{sensitivity}}{(\text{sensitivity} + \text{error})}$ over five different threshold values (T1, T2, T3, T4, T5) in increasing order. The actual values of these five thresholds for each method is given in Table 3.2. The higher the value of this ratio, the better the method works, as it shows the error is less. Figure 3.9(a) gives the plots for the case when NP is differentially over-expressed compared to ES while 3.9(b) shows the case when ES is differentially over-expressed compared to NP. We see from the plots that although most methods work well when NP is differentially over-expressed, only the two-stage unit mean normalization and the ChIPnorm method works well when ES is differentially over-expressed compared to NP. In fact the other methods show a ratio less than 0.5 (Figure 3.9(b)), indicating that the error is greater than sensitivity. This clearly indicates that the first stage of the proposed ChIPnorm approach helps remove the problem of bias which happens because of the different signal-to-noise ratios in the two libraries.

In this chapter, we used a bin size of 1000 bp for H3K27me3 data and 200 bp for H3K4me3 data. However we have done robustness studies by varying bin sizes. The sensitivity and error experiments shown in Table 3.1 for the data in Mikkelsen *et al.* 2007 [85] are repeated here

Table 3.2 – *Values of the various thresholds (T1, T2, T3, T4, T5) used for the various methods used in Figure 3.9.*

thresholds	unit-mean	quantile	MACS	ChIPDiff	RSEG	two-stage unit-mean	ChIPnorm
	τ	τ	p-val	τ	cdf	τ	τ
T1	3	11	10^{-2}	1.1	0.25	0.98	2
T2	5	13	10^{-4}	2	0.5	1.48	2.5
T3	7	15	10^{-6}	3	0.75	1.98	3
T4	9	17	10^{-8}	4	0.9	2.48	3.5
T5	11	19	10^{-10}	5	0.95	2.98	4

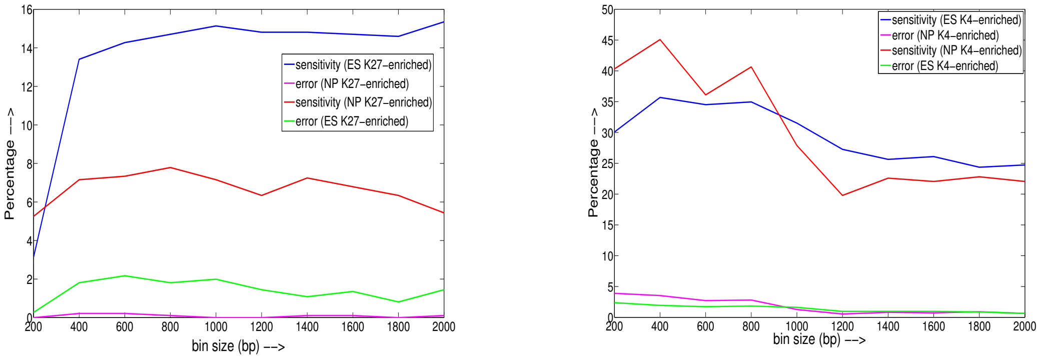


(a) *(b)*

Figure 3.9 – *Plot of* $\frac{sensitivity}{(sensitivity + error)}$ *over five different threshold values. (a) NP is differentially over-expressed compared to ES, (b) ES is differentially over-expressed compared to NP.*

for the ChIPnorm approach by varying the bin sizes for H3K27me3 and H3K4me3 data. The fold change threshold (τ) is fixed at 3 (as done in Table 3.1) for the ChIPnorm method and the bin sizes vary from 200 bp to 2000 bp in steps of 200 bp. We see from Figure 3.10 that the sensitivity and error vary little for bin sizes varying from 400 bp to 2000 bp for H3K27me3 and 200 bp to 800 bp for H3K4me3. (For H3K4me3 data, while the sensitivity drops after 800 bp bin size, so does the error.) This shows that the ChIPnorm method is robust over a wide range of bin sizes and that there is a large range of bin sizes where ChIPnorm works well for both H3K27me3 and H3K4me3 data.

We repeated the sensitivity tests for the human data from the ENCODE Broad database (hg18). We tested the experiments on ES and GM12878 cell lines for H3K27me3 histone modifications [13]. The corresponding gene expression data (RPKM) is from the ENCODE Caltech RNA-seq database [13]. The parameters of all the methods (except rank normalization) were adjusted so that all of them give almost the same percentage ($\approx$ 11%) of experiment “sensitivity (HES K27-enriched)”. For the rank normalization method we could not further change the threshold to get a sensitivity close to 11%. The results are summarized in Table 3.3. We can clearly see that ChIPnorm and the two-stage unit-mean approach outperform all other methods. However the two-stage unit mean approach results vary a lot based on the thresholds. When the threshold τ is changed from 1.4 to 1.89, so as to fix “sensitivity (GM12878



(a) H3K27me3

(b) H3K4me3

Figure 3.10 – *Robustness studies: sensitivity and error analysis for ChIPnorm by fixing the fold-change threshold* $\tau = 3$ *and varying the bin size from 200 bp to 2000 bp in steps of 200 bp. Data from Mikkelsen* et al. *2007 [85].*

K27-enriched)" to approximately the same as that of ChIPnorm approach (i.e. approximately 12%), the results are: "sensitivity (GM12878 K27-enriched)" is 12.07% while the corresponding "error (HES K27-enriched)" is 0.86%; "sensitivity (HES K27-enriched)" reduces to 4.67% (from the earlier 10.69% for $\tau = 1.4$) while the corresponding "error (GM12878 K27-enriched)" value is 0.21%. Therefore the sensitivity of the two-stage unit-mean approach ("sensitivity (HES K27-enriched)") reduced with change in threshold. In fact for this data set, ChIPnorm results varies very little with change in threshold. For example, for ChIPnorm when we change the threshold τ from 3 to 15, the "sensitivity (HES K27-enriched)" values change from 12.71 to 10.07. Similarly the rest of the results do not vary much. This leads us to believe that the second stage quantile normalization approach in ChIPnorm, gives stabler results as it equates the distributions of the two libraries.

Table 3.3 – *Sensitivity analysis for human ES and GM12878 cells (replicate 1 data from ENCODE Broad database) percentages using various methods. Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The parameters of all the methods (except for rank normalization as explained in the text) were adjusted so that all of them give almost the same percentage (≈ 11%) of experiment "sensitivity (HES K27-enriched)".*

	unit-mean	quantile	MACS	ChIPDiff	rank	RSEG	two-stage unit-mean	ChIPnorm
thresholds	$\tau = 5.3$	$\tau = 15.7$	p-val = 10^{-12}	$\tau = 7$	$\nu = 7$	cdf = 0.9	$\tau = 1.4$	$\tau = 10$
GM12878 differential (four-fold) expressed: 1927 genes								
sensitivity (HES K27-enriched)	11.26	11.26	11.26	10.85	17.85	8.27	10.69	11.05
error (GM12878 K27-enriched)	0.10	0.21	0	0.16	5.35	0	0.31	0.16
HES differential (four-fold) expressed: 2908 genes								
sensitivity (GM12878 K27-enriched)	1.55	10.97	1.79	4.09	33.29	0.06	14.65	12.00
error (HES K27-enriched)	1.38	0.93	3.54	1.20	2.27	11.81	1.20	0.55

ROC curves

We next plot receiver operating characteristics (ROC) for H3K27me3 histone modification data (Mikkelsen *et al.* 2007 [85]) to compare the various techniques. Since what is a 'true' differential region is unknown, we used indirect ways of calculating true positives (TP), true negatives (TN), false positives (FP), false negatives (FN), by comparing the results with gene expression data. For the sake of plotting the ROC curves, we convert the 3-sided testing problem (ES differentially enriched, NP differentially enriched, or non-differential) into a two-sided problem. We plot two different ROC curves and define the various parameters keeping in mind that K27 is a repressor.

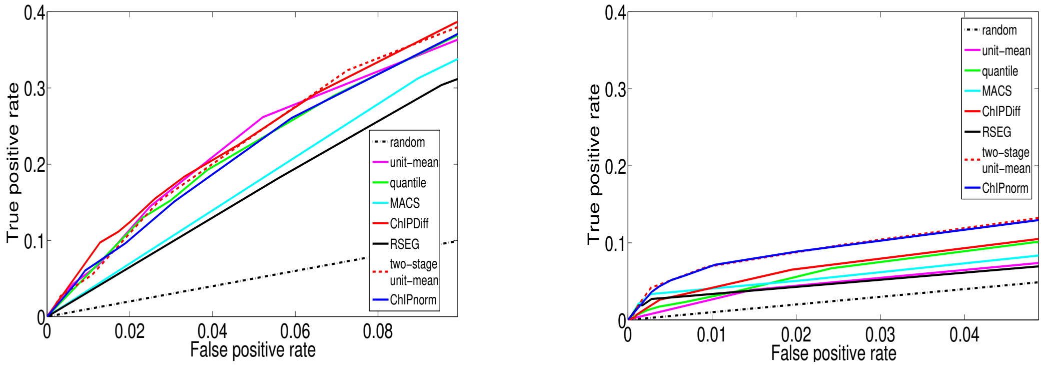
For the first ROC: Class 1 is defined as four-fold NP differentially over-expressed genes compared to ES; Class 0 is rest of the genes.
TP - genes were ES (ChIP-Seq) is declared differentially enriched (i.e. above threshold) in Class 1 (since H3K27me3 is a repressor).
FN - rest of the genes which fall in Class 1.
FP - genes were ES (ChIP-Seq) is declared differentially enriched (i.e. above threshold) in Class 0.
TN - rest of the genes which fall in Class 0.
True positive rate (TPR) = sensitivity = TP/(TP + FN)
False positive rate (FPR) = (1 - specificity) = FP/(FP+ TN)

For the second ROC: Now Class 1 is defined as four-fold ES differentially over-expressed genes compared to NP; Class 0 is rest of the genes. And the rest of the parameters are defined for NP differentially enriched (opposite of the previous case).

The two ROC curves are shown in Figure 3.11. It is important to note that this is only an approximate way of calculating ROCs as gene expression depends on more than just H3K27me3 histone markers but also on many other factors (like other histone modifications, transcription factors, etc.). Therefore a 100% TPR is not necessarily a good result. We varied the thresholds for various methods. In some methods, the values of TPR and FPR do not go beyond a certain value, irrespective of the thresholds. Therefore we show the plots of the regions where maximum value of the FPR exists for all methods. From the figure, it is seen that most methods work well for the first ROC, while for the second ROC curve, ChIPnorm and two-stage unit mean normalization outperforms all other methods, clearly showing the removal of the one sided bias.

False-positive rate

Since the ENCODE Broad database for human histone modifications data [13] has a two replicates for each cell type, we did a false-positive rate study on this data. Basically we ran the methods on H3K27me3 libraries for ES cells for replicates 1 and 2. Since the two libraries used are replicates, we do not expect any differentially enriched regions between the libraries. So any bin declared as differential is considered a false positive. Since the false-positive rate is dependent on the thresholds used, we used the same thresholds as used in Table 3.3 for the various methods. The results are shown in Table 3.4. We see that the false-positive rates are below 1% in all the methods (except rank normalization) and for ChIPnorm it is very close to



(a) first ROC

(b) second ROC

Figure 3.11 – *Two ROC curves are shown for various methods. (a) first ROC: Class 1 - four-fold NP differentially over-expressed genes compared to ES; Class 0: rest of the genes. (b) second ROC: Class 1 - four-fold ES differentially over-expressed genes compared to NP; Class 0: rest of the genes.*

0%.

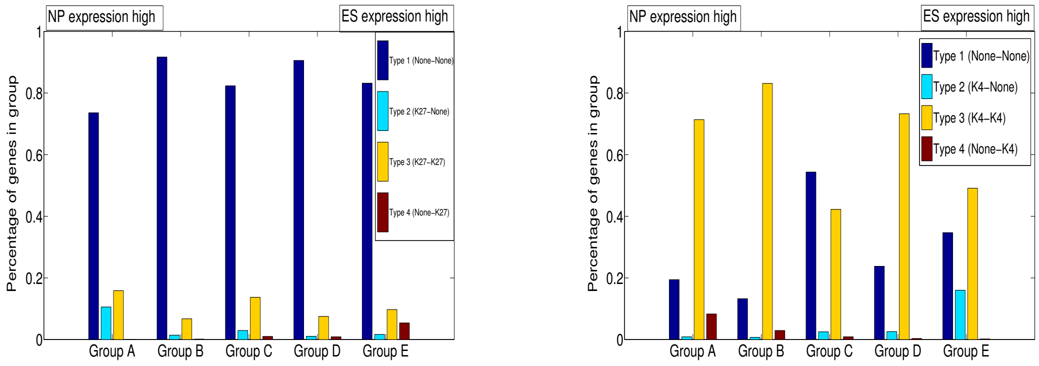
Table 3.4 – *False-positive rate (FPR) analysis for human ES cells (H3K27me3 data from ENCODE Broad database) for the two replicates. We see the percentage of false positive using various methods. Experiments: unit-mean; quantile; MACS peak finder; ChIPDiff; Rank normalization; two-stage unit-mean; ChIPnorm. The thresholds used are same as those in Table 3.3.*

	unit-mean	quantile	MACS	ChIPDiff	rank	RSEG	two-stage unit-mean	ChIPnorm
thresholds	$\tau = 5.3$	$\tau = 15.7$	p-val = 10^{-12}	$\tau = 7$	$\nu = 7$	cdf = 0.9	$\tau = 1.4$	$\tau = 10$
FPR	0.3924	0.0004902	0.0053	0	8.2753	0.4845	0.8652	0.0088

3.4.2 Correlation with gene expression

We analyze the effect of presence of K27 and K4 sites (mouse data from Mikkelsen *et al.* 2007 [85]) on gene expression levels. K4 is associated with activation of genes, while K27 is associated with repression [85]. 13,000 UCSC known genes are used for this purpose. We divide these genes into five groups (A–E) according to the increasing log ratio of their expression levels in ES and NP cells. We take into account the distribution of the number of genes with respect to the log ratio of their expression levels to make the division. This grouping ensures enough representation in each group. Genes in each group are further classified for each K27 or K4 according to the presence of modifications in the promoter (± 1 kbp of TSS) region. These categories are: type 1 genes have neither DHE nor CHE bins in their promoter regions; type 2 genes have at least one DHE bin enriched for ES cells, but not even one CHE or DHE bin with NP enrichment; type 3 genes have at least one CHE bin or at least two bins with opposite enrichments; type 4 genes have at least one DHE bin enriched for NP cells, but no CHE or DHE bin enriched for ES cells.

Figure 3.12(a) shows the percentages of genes in each group for H3K27me3. These percentages decrease from group A to group E, indicating that the number of genes differentially enriched for modification K27 in ES cells decreases at higher levels of differential gene expression. On the other hand, type 4 genes, which are differentially enriched for NP cells, increase from group A to group E. Thus we see clear evidence of negative correlation of K27 with gene expression, confirming the repressive regulation by K27. Similar conclusions can be drawn for K4 from Figure 3.12(b), indicating the positive correlation of K4 with expression levels and thereby confirming its association with the activation of genes.



(a) H3K27me3

(b) H3K4me3

Figure 3.12 – *Gene profile according to expression and histone modifications. Genes are grouped in (A-E) according to increasing ratio of expression level in ES cells and NP cells. Within each groups, genes are classified into 4 types. Type 2 genes have differential histone enrichment in ES cells in their promoter regions and type 4 genes have differential enrichment in NP cells. (a) Percentage of type 2 genes is decreasing, while percentage of type 4 genes is increasing along the group (A-E). (b) Percentage of type 2 genes is increasing, while percentage of type 4 genes is decreasing along the group (A-E).*

3.4.3 Bivalent region analysis

H3K27me3 and H3K4me3 are sometimes present simultaneously at the same promoter [11, 85]. Such bivalent regions may repress the developmental genes in ES cells, while keeping them poised for activation at later stages of development in partially differentiated cells. Thus bivalent regions could play an important role in the maintenance of pluripotency for ES cells. We would therefore expect bivalent regions to be enriched in ES cells as compared to the better differentiated NP cells. We would also expect that bivalent regions in ES cells would preferentially lose the K27 rather than the K4 mark in NP cells.

We applied the ChIPnorm method to investigate these conjectures about bivalent regions in ES cells. First we selected 333 genes from chromosomal regions that are rich in highly conserved noncoding elements (HCNEs) which were previously analyzed by Bernstein *et al.* 2006 [11]. We classified these genes into 16 classes according to the presence or absence of K27 or K4 modifications in one or both of ES and NP cells (data from Mikkelsen *et al.* 2007 [85]). Figure 3.13 shows the representation of each class of genes among these selected HCNE genes.

The “A-B” in a label indicates the presence of modification “A” in ES cells and of modification “B” in NP cells; as before, detected but depleted modifications are treated as if they were

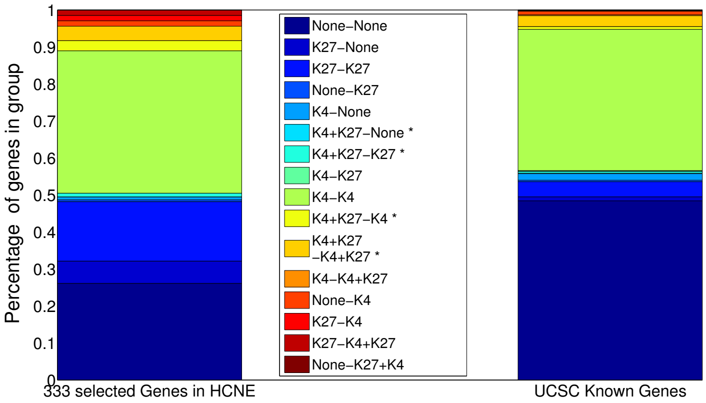


Figure 3.13 – *Bivalent regions in genes: in 333 selected genes and in UCSC known genes. Genes are attributed to classes according to the presence of modifications in ES and NP cells. The "A-B" notation in the labels indicates the presence of modification of type "A" in ES cells and modification of type "B" in NP cells. (*) marked labels have bivalent domains in ES cells.*

absent; labels marked with an asterisk denote bivalent regions. We found that about 20% of these 333 genes had bivalent regions within 1 kb from the promoter in ES cells—a result consistent with Bernstein *et al.* 2006 [11]. We then examined the whole genome for the presence of bivalent regions. Figure 3.13 also shows the representation of such classes among UCSC known genes. Though the percentage of genes with bivalent regions drops to about 10%, it remains surprisingly high, suggesting that bivalent histone marks in ES cells are not confined to key developmental regulators. We also note that bivalent marks specific to NP cells are extremely rare, while bivalent mark present in both cell types occur at an intermediate frequency. Contrary to our expectation, bivalent marks of ES cells do not preferentially lose their K27 mark. A substantial fraction loses the K4 mark instead, which may reflect a transition into a permanently repressed state.

Finally we studied the connection between bivalent regions and gene expression levels. We divided the UCSC known genes in the same A–E groups according to the log ratio of expression levels in ES and NP cells. Within each class we further classified genes into 16 classes according to the presence of one or both histone modifications in ES and NP cells. Figure 3.14 shows a strong over-representation of the K4+K27-K4 transition in the class of genes that are strongly up-regulated in NP cells, indicating that the fate of a bivalent mark indeed influences the expression of the corresponding gene in a progenitor cell. Overall, our findings support the hypothesis of Bernstein *et al.* 2006 [11] that bivalent K4+K27 marks are frequent in ES cells and associated with a temporary repression of genes that need be activated later in development. Our results extend Berstein's hypothesis in that we show that bivalent marks are not confined to HCNE-associated key-regulatory genes, and that a sizable fraction of them transits into a K27-only state possibly reflecting permanent repression. Moreover, using our normalization method, we show that transition of a bivalent state into a K27-only state is a rather frequent event rather than an exception as reported in previous papers (e.g. Cui *et al.* 2009 [24]).

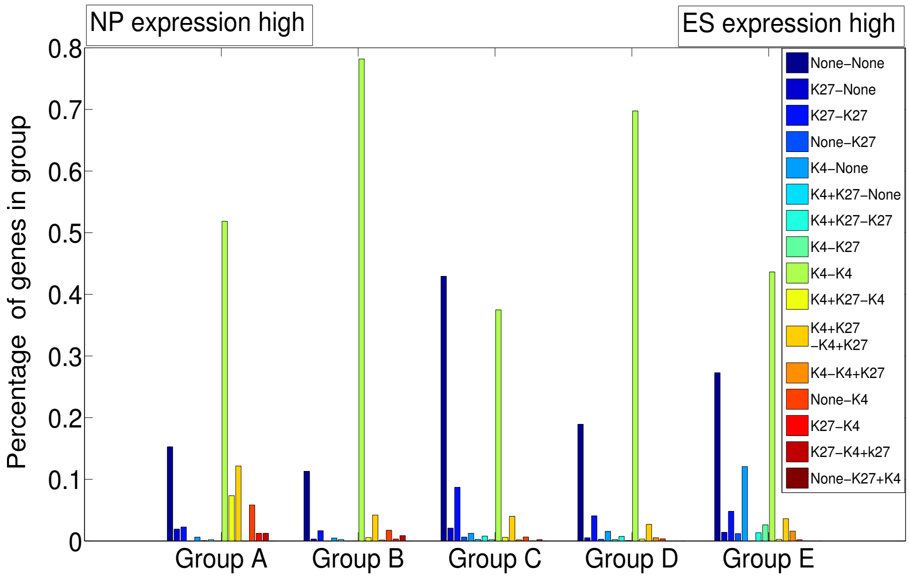


Figure 3.14 – *Bivalent gene profile vs. expression data. Genes are grouped in (A-E) according to increasing ratio of expression level in ES cells and NP cells. Each bar shows the percentage of genes with the corresponding "A-B" modifications (as listed in the box), "A" for modifications in ES cells and "B" for modifications in NP cells. It is seen that there is a strong over-representation of the K4+K27-K4 transition (yellow) in the genes class, which is strongly up-regulated in NP cells.*

3.4.4 Differential enriched regions along protein-coding genes

We analyzed the human ENCODE ES and GM12878 H3K27me3 ChIP-Seq data [13] using Segtools [18]. Figure 3.15 shows the relative enrichment of the regions identified by the ChIPnorm for human ES and GM12878 H3K27me3 ENCODE data along the various genomic features of protein coding genes (GENCODE v3c genes [53]). The regions, which are differentially enriched in ES cells (L1_enrich), are present along the 5' flanking end and initial exons of genes. This shows that these regions are mostly present in the promoter regions of genes. We also see that the regions that are differentially enriched in GM12878 cells (L2_enrich) are also present in the promoter regions of genes. However, regions, which are not differentially enriched. (non_differential) are absent along the promoter regions of genes. Therefore we provide some evidence that most of the promoter regions of protein coding genes have differentially enriched histone-modification sites, and very few non-differential sites.

3.4.5 Using ChIPnorm in analyzing histone modification data from Arabidopsis

Leaf senescence is the orderly dismantling of older tissue that allows recycling of nutrients to developing portions of the plant. It is accompanied by major changes in gene expression [17]. We used the ChIPnorm method to study the effect of H3K4me3 and H3K27me3 on soil-grown mature and naturally senescent Arabidopsis leaves. The ChIPnorm method was used to normalize data sets and identify genomic regions with significant differences in the two histone

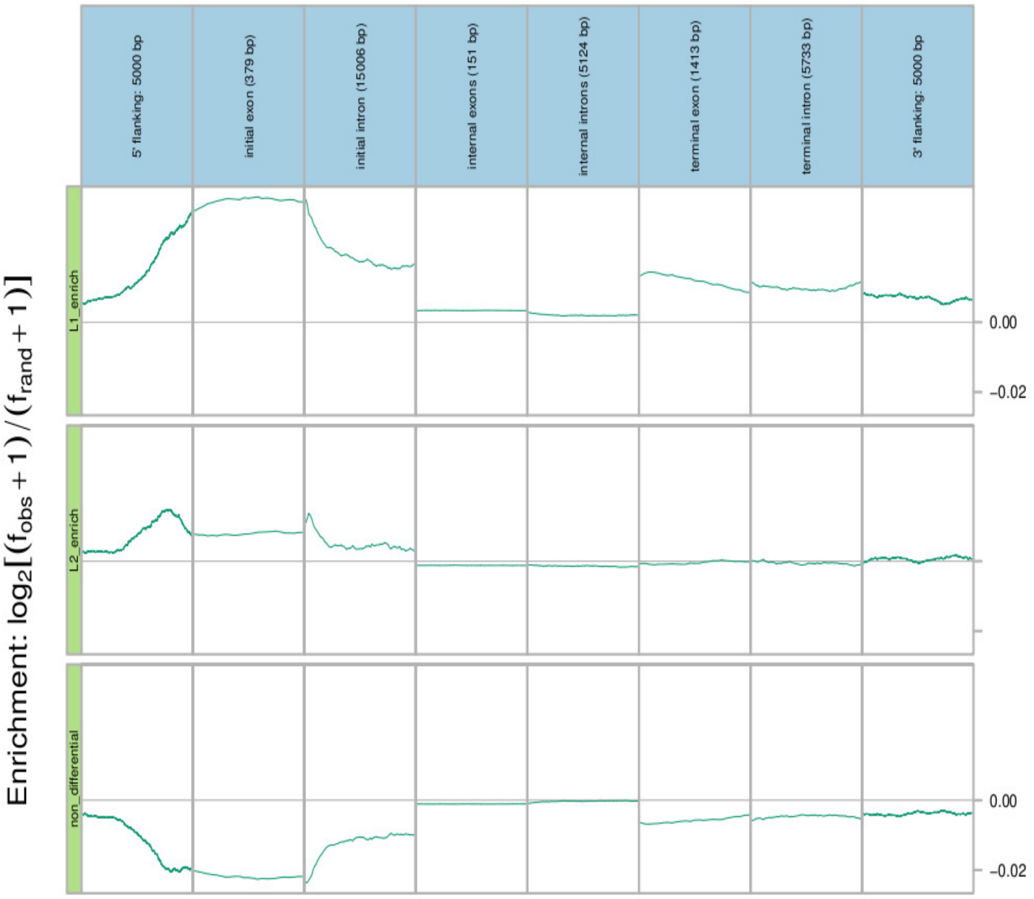


Figure 3.15 – *Feature aggregate plot of the differential/non-differential regions identified by the ChIPnorm method. Each row corresponds to a region from ChIPnorm and each column corresponds to a genomic feature of protein coding GENCODE genes. Curve inside a cell represents the relative frequency of overlap between the ChIPnorm identified regions and the genomic feature of GENCODE genes, when compared to a similar relative frequency of overlap that would occur by random chance. Figure created using Segtools [18].*

methylation patterns. These differences were then correlated with gene expression. We found that genes that showed an increase in the H3K4me3 mark in older leaves were senescence up-regulated, while genes that showed a decrease in the H3K4me3 mark in the older leaves were senescence down-regulated. We also found that for the H3K27me3 modification, genes that lost the H3K27me3 mark in older tissue were senescence up-regulated. From our analysis we found that only a small number of genes gained the H3K27me3 mark, and these were senescence down-regulated. And approximately 50% of senescence up-regulated genes lacked the H3K4me3 mark in both mature and senescent leaf tissue. More details of our work can be found in [17].

3.4.6 Software Availability

The software for this work can be downloaded from http://lcbb.epfl.ch/software.html. The code is mostly written in MATLAB.

3.5 Conclusion

We have presented an approach for the analysis of ChIP-Seq data, with particular emphasis on the discovery of differentially enriched histone-modification sites. The problem of the bias inherent in the comparison of two sets of data with different noise backgrounds is biologically more relevant because such bias shows a false correlation between computationally identified differential regions with gene density. The ChIPnorm removes most of this bias and provides a normalization that enables direct comparison of values. We have conducted experiments that demonstrate that this new approach improves significantly on the state of the art. One of the main reasons that the ChIPnorm method outperforms other methods is because of the first stage of normalization that we proposed. This is the reason when we used the two-stage unit-mean normalization, which used the same first stage as the ChIPnorm method, the results were quite good. We also think that since we used the amplified binomial distribution as the random distribution (instead of using only control data), our results were better than the earlier proposed methods. Finally, we have used our approach to highlight some aspects of K27 modifications in mouse embryonic stem cells and neural progenitor cells, including a so far unnoticed transition of bivalent mark of K4 and K27 in embryonic stem cell to a K27-only state in differentiated cells, possibly reflecting permanent repression of developmental genes. For the human ENCODE H3K27me3 data for ES and GM12878 cells, when we look at protein-coding genes, we provide evidence that most of the promoter regions have differentially enriched histone-modification sites. Also the ChIPnorm approach has also been used to study histone methylation changes associated with leaf senescence in Arabidopsis.

Our approach is not restricted to the identification of differentially enriched sites nor is it limited to pairwise comparisons. A natural next step, therefore, is to apply it to more complex data (multiple cell types with multiple histone modifications, for instance), to verify its efficacy, and to use it to shed light on the complex interactions described in the “histone language".

4 Probabilistic partitioning methods to find significant patterns in ChIP-Seq data

4.1 Introduction

As seen before, a ChIP-Seq experiment produces a large number of sequence tags that are mapped to the genome, resulting is a genome-wide profile of tag counts. A high tag count at a location on the chromosome indicates the presence of a particular protein at that location. The regions enriched in ChIP-Seq tags are diverse in terms of magnitude, shape and orientation [69]. Sequence-specific transcription factors typically produce uniform, narrow Gaussian peaks, while regions enriched in histone modifications tend to show complex multimodal signal distributions.

The term “chromatin signature” has been coined to designate recurrent patterns found in ChIP-Seq-based histone modification maps and other types of chromatin profiling data [56]. A chromatin signature is usually represented by a vector of average tag counts in bins of certain sizes (typically 50bp to 500bp) in a collection of larger genomic regions of sizes 1 kb to 10 kb. Chromatin signatures can be detected by so-called aggregation plots (AP) [58], if precisely mapped experimentally defined anchor points (e.g., transcription start sites, known as TSSs) are available for selection and delineation of the genomic regions of interest. A basic assumption in ChIP-Seq data analysis is that specific chromatin signatures are associated with specific functions. For instance, human promoters are characterized by a nucleosome-free region of about 150bp and a rigidly positioned H3K4me3-marked +1 nucleosome centered 120bp downstream from the TSS [105].

Given a set of genomic regions of constant size around some fixed anchor points (like transcription start sites of genes) we would like to discover recurring patterns or classes (chromatin signatures) in the ChIP-Seq data which arise within these regions. However, discovering a recurring pattern is a difficult problem, especially when precise anchor points are not available.

An effective algorithm must be capable to cope with the following obstacles.

- *Biological inhomogeneity of the samples:* The set of analyzed genomic regions often consists of multiple unknown subclasses, in which case a plot derived from all samples shows the superposition of several different chromatin signatures.
- *Alignment uncertainty:* Precise anchor points are rarely available for delineating genomic regions. Selected chromatin regions first need to be optimally shifted (registered) with respect to each other before an AP can reveal a high-resolution chromatin signature.
- *Asymmetry:* Chromatin signatures associated with directional molecular mechanisms (such as transcription) are usually asymmetrical. However, the orientation of the genomic regions is often unknown. The input count vectors should then be compared with each other in both orientations.
- *Sparse count data:* Certain bins may have very low tag counts, leading to high sampling errors.

The problem of inhomogeneity can be tackled by off-the-shelf clustering and partitioning algorithms. In fact, hierarchical clustering [109] and K-means [79] have been incorporated in several multipurpose computational platforms for ChIP-Seq data analysis. seqMINER [129] offers an in-built K-means function, while ChIPseeker [46] is interfaced with a third-party hierarchical clustering software. However, shifting and flipping is only implemented in specialized programs like ChromaSig [55], ArchAlign [68], CATCHprofiles [93] and CAGT [67]. CATCH-profile is a tool for exhaustive pattern detection in ChIP profiling data. and it implements the CATCH algorithm. In the CATCH algorithm, a hierarchical clustering approach combined with pairwise alignment is used. It keeps a pool of profiles from which it iteratively aligns all pairs and chooses the most similar pair [93]. The exhaustive all-against-all comparison and alignment in the CATCH algorithm makes it computationally expensive. ArchAlign has two methods for aligning regions — the first one is the single-best-pair approach which uses the two regions with the highest similarity as the template pattern to seed the alignment; the second approach, known as seed sampling, is a more comprehensive search of the possible alignment space. ArchAlign performs only shifting and flipping and can find only one single signature. CAGT supports flipping but not shifting. (The problem of optimal shifting is typically solved by exhaustive comparison of all overlapping subregions of a given size from two genomic regions, possibly in both orientations.) ChromaSig, ArchAlign and CATCHprofiles use progressive multiple alignment strategies to assemble similar tag profiles. Since these algorithms have to carry out a large number of pairwise comparisons, they tend to be slow. To overcome this drawback, CAGT applies a two-step divide-and-conquer approach. It first uses the the K-median algorithm (a variant of K-means) to define top-level classes and then runs a hierarchical clustering algorithm on each of these classes in turn. The shifting and clustering functions require some type of distance measure. All of these programs, except ChromaSig, use non-probabilistic measures such as the Euclidean distance or the Pearson correlation coefficient, neither of which does well with low counts per bin. ChromaSig assesses similarity between samples and class membership assuming position-specific Gaussian distributions of the normalized ChIP-Seq signal within a chromatin signature. The use of Gaussian distributions, which seems unnatural for count data, is explained by the fact that ChromaSig was originally designed for ChIP-chip data.

In this chapter, we propose as an alternative approach for finding recurrent patterns in ChIP-Seq data by *probabilistic partitioning* [87]. The underlying principle of this general method is to optimize a mixture model by an Expectation-Maximization (EM) algorithm, a strategy that has already proved effective in finding recurrent DNA motifs in selected genomic regions [78]. A key difference of this method compared to the other clustering methods mentioned is that samples are not deterministically assigned to a single class: rather, their classification status is defined by a vector of class membership probabilities. While EM has long been a standard tool in machine learning, it is a general-purpose method, whose convergence rates and running times depends on the exact formulation of the objective function and the updating formulae. The purpose of this chapter is to demonstrate the merits of EM when applied to ChIP-Seq data and to explain by examples how it can be applied to classification and motif-discovery problems in research on chromatin structure. The probabilistic partitioning approach offers the following advantages.

1. The use of probabilistic distance functions naturally takes into account random sampling variation in low-count data.
2. Probabilistic class assignment allows for accurate characterization of classes even in situations where the classification of individual samples is uncertain.
3. Probabilistic class assignment is flexible and can combine goals, for instance the ranking and prioritizing of ChIP-Seq signal enriched regions based on peak-shape.
4. Shifting and flipping can be implemented in the EM framework via hidden variables.
5. The implementation of probabilistic partitioning is straightforward with existing programming platforms. All algorithms used in this work can be implemented by less than 30 lines of *R* code.
6. *Flexibility:* Methods are readily customized to meet the needs of a particular application. For instance, the switching from a Poisson probabilistic model to a negative binomial model requires only one change in the corresponding *R* code.
7. *Efficiency:* In contrast to most existing methods, the EM algorithm does not require exhaustive pairwise comparisons, so that each iteration runs in time linear in the number of samples.
8. *Transparency and Reproducibility:* Methods can be accurately described in a research paper by reproducing a few lines of *R* code (see the *R* codes given in Appendix A).

Section 4.2 presents in detail several variants of the probabilistic partitioning algorithms. Section 4.3 analyses the performance of these algorithms on carefully chosen examples based on simulated and real ChIP-Seq data and compares its performance with K-means clustering and CAGT.

4.2 Methods

We are given N samples, $S_1, S_2, \ldots, S_N$. Samples are regions of certain sizes around fixed anchor points. These anchor points could be transcription start sites (TSS) of genes or transcription factor binding sites. (For example, a sample could be a region of +/- 1 kbp region around a TSS of a particular gene.) We divide the genome into bins and count the number of ChIP-Seq fragments that fall into each bin to obtain *bin counts*. Thus each sample S_i is an integer vector of length L, $S_i = (s_{i1} s_{i2} \ldots s_{iL})$, where each element s_{il} is a bin count. Bincount vectors of several ChIP-Seq libraries (e.g. different histone marks) may be concatenated in order to partition them together. We assume that the samples originate from a mixture of K different classes, $C_1, C_2, \ldots, C_K$. Each class C_j occurs with characteristic probability $p_j = P(C_j)$ and is further characterized by "profiles" of expected bin counts: $C_j = (c_{j1} c_{j2} \ldots c_{jL})$.

4.2.1 Expectation-Maximization (EM) algorithm

The probability of sample S_i given class C_j is computed as follows:

$$P(S_i|C_j) = \prod_{v=1}^{L} \text{Poisson}(s_{iv}, \lambda = c_{jv}) \tag{4.1}$$

Now, the probability of class C_j given sample S_i is given by:

$$P(C_j|S_i) = \frac{p_j P(S_i|C_j)}{\sum_{b=1}^{K} p_b P(S_i|C_b)} \tag{4.2}$$

Using this probability, we update the classes as follows:

$$c_{jl} = \frac{\sum_{a=1}^{N} P(C_j|S_a) s_{al}}{\sum_{a=1}^{N} P(C_j|S_a)} \tag{4.3}$$

$$p_j = \frac{\sum_{a=1}^{N} P(C_j|S_a)}{N} \tag{4.4}$$

These computations are carried out iteratively for a fixed number of steps.

4.2.2 Modified "Shape-Only" EM algorithm

We also propose a shape-only version of the EM algorithm for normalization purposes. For all K classes, the average count frequency is set to 1. In other words, we impose:

$$E(C_j) = 1 \Longleftrightarrow \sum_{v=1}^{L} c_{jv} = L \tag{4.5}$$

Equation 4.1 is modified as follows:

$$P(S_i|C_j) = \prod_{v=1}^{L} \text{Poisson}(s_{iv}, \lambda_j = c_{jv}(1/L)\sum_{g=1}^{L} s_{ig}) \tag{4.6}$$

The purpose is to adjust the average count frequency of class j to the average count value of sample i.

$$E(\lambda_j) = E(S_i) \tag{4.7}$$

We further have to make sure that the average count frequency of the re-estimated class j equals 1. To this end, equation 4.3 is modified as follows:

$$c_{jl} = \frac{L\sum_{a=1}^{N} P(C_j|S_a)s_{al}}{\sum_{v=1}^{L}\sum_{a=1}^{N} P(C_j|S_a)s_{av}} \tag{4.8}$$

4.2.3 Variations - with shift and flip

We propose some variations of the basic method. In the following we show how flipping and shifting can be implemented. Note that these two options could be implemented separately. Here (for the sake of generality) we show the version which supports both. Shifting and flipping is modeled with two hidden variables, the shift index m and the flip state inv.

Let m be the shift index and M be the maximum number of shifts allowed; and let inv be equal 1 when there is no flip and equal to 2 when there is one. Note that with shifting, the patterns C_j are shorter than the samples S_i by $M-1$. The notation $s_{il}(m, \text{inv})$ will be used to represent the data for a particular shift and flip state: for inv = 1, $s_{il}(m, \text{inv}) = s_{i,l+m-1}$; for inv = 2, $s_{il}(m, \text{inv}) = s_{i,L-M+m-l+1}$. Now, the probability of sample S_i given class C_j and further conditioned on shift index m and flip state inv, is computed as follows:

$$P(S_i|C_j; m, \text{inv}) = \prod_{v=1}^{L} \text{Poisson}(s_{iv}(m, \text{inv}), \lambda = c_{jv}) \tag{4.9}$$

Now, the probability of class C_j given sample S_i is given by:

$$P(C_j, m, \text{inv}|S_i) = \frac{p_j(m, \text{inv})P(S_i|C_j; m, \text{inv})}{\sum_{b=1}^{K}\sum_{d=1}^{M}\sum_{e=1}^{2} p_b(d, e)P(S_i|C_b; d, e)} \tag{4.10}$$

Using this probability, we update the classes as follows:

$$c_{jl} = \frac{\sum_{a=1}^{N}\sum_{d=1}^{M}\sum_{e=1}^{2} P(C_j, d, e|S_a)s_{al}(d, e)}{\sum_{a=1}^{N}\sum_{d=1}^{M}\sum_{e=1}^{2} P(C_j, d, e|S_a)} \tag{4.11}$$

$$p_j^*(m,\text{inv}) = \frac{\sum_{a=1}^{N} P(C_j, m, \text{inv}|S_a)}{N} \tag{4.12}$$

Here we assume that the shift states follow a centered Gaussian distribution with equal width for all classes. Therefore, we infer only the standard deviation of the distribution from the data. Practically, this is achieved by applying the following regularization step to the re-estimated probabilities $p_j^*(m,\text{inv})$.

$$\mu = \frac{\sum_{b=1}^{K}\sum_{d=1}^{M}\sum_{e=1}^{2} p_b^*(d,e)\ d}{\sum_{b=1}^{K}\sum_{d=1}^{M}\sum_{e=1}^{2} p_b^*(d,e)} \tag{4.13}$$

$$\sigma = \sqrt{\frac{\sum_{b=1}^{K}\sum_{d=1}^{M}\sum_{e=1}^{2} p_b^*(d,e)(d-\mu)^2}{\sum_{b=1}^{K}\sum_{d=1}^{M}\sum_{e=1}^{2} p_b^*(d,e)}} \tag{4.14}$$

Let $\text{Normal}(m|(M+1)/2,\sigma)$ be the probability of shift m which has a Gaussian distribution of mean $(M+1)/2$ and standard deviation σ.

$$p_j(m,\text{inv}) = \frac{\text{Normal}(m|(M+1)/2,\sigma)}{\sum_{d=1}^{M}\text{Normal}(d|(M+1)/2,\sigma)} \sum_{h=1}^{M} p_j^*(h,\text{inv}) \tag{4.15}$$

As before, these computations are iterated for a fixed number of steps.

Since we are able to estimate the probability of each shift for every sample and class, we can use these probabilities to estimate the internal position of a given pattern in a particular sample. Under sub-heading 4.3.2, we present a biological example where we make use of this possibility.

4.2.4 Seeding and initialization strategies

Various possible seeding and initialization strategies are possible for the proposed probabilistic partitioning algorithms. Here are two such possibilities.

- Start with one class ($K = 1$). Set $P(C_1|S_i) = 1$ (for partitioning without shifts or flips) and $p_1 = 1$. The initial distribution of class one (c_{1l}) can be defined in either of these two ways: (a) we can take the mean of the entire data across all the samples; (b) choose a random distribution by either picking a random subset of the data or by choosing a random probability for each sample, and then taking the weighted sum over all the samples according to their probability value. Then increase the number of classes iteratively ($K = K+1$) till the maximum number of classes is reached. With each iteration, the new class is initialized to a uniform distribution ($c_{jl} = 1$) $\forall l$ and j is the new class. The new class will have a prior probability ($p_{\text{new class}}$) $= 1/K$ where K is the total number of classes so far. The remaining classes have a total probability ($\sum_j p_j$) of $(1-1/K)$, where each class is $p_j = (1-1/K)\ p_j^{old}$

(the earlier value of p_j is p_j^{old}). After the initialization (for each increase in the number of classes), the EM method is applied.

- Start with K classes ($K \geq 1$). Like done before, we could take K different subsets of the original data and compute their mean, and use that to compute the initial distributions for the different classes. Alternatively one could also choose K random probability vectors (each vector containing probabilities for all samples) and use that to compute K weighted sums for finding the initial distributions of the K classes. After this initialization, the EM method is applied.

Determining the optimum number of classes or clusters (choosing K) in a dataset has been a problem which has been addressed in the literature for many decades now. The number of classes should strike a balance between assigning all samples into one class and assigning each sample into a separate class. Methods which look at percentage of variance as a function of number of classes [62] or by using methods based on information criteria like Akaike information criterion or Bayesian information criterion are often used (among many others). However most of these methods have their drawbacks [128]. Since probabilistic partitioning method is to be used as an exploratory tool, we leave it to the user to manually see what is the best number of interesting classes for the dataset being used.

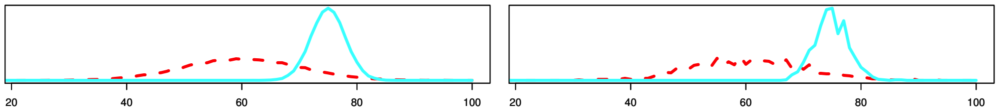
4.3 Results and Discussion

4.3.1 On simulated data

We first run the computational experiments on simulated data. The data is composed of a mixture of two classes characterized by bin count frequency profiles of different shapes. The samples were integer vectors of length 100. Counts were generated by randomly sampling from a Poisson distribution with λ varying in a class and position specific manner along the bin count frequency profiles. Since we were particularly interested in the algorithm's capability of recovering patterns from sparse count data, we varied the total count coverage f over a wide range of relevant values (f is defined as the total expected bin counts per sample). The simulated data was generated using statistical software R. The R code and additional details of the computational protocols are given in the Appendix A.

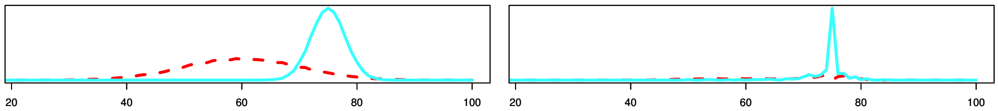
Data without shifts or flips.

We first generated random samples belonging to two classes, 1000 samples for each class. The classes were defined by bin count profiles of Gaussian shape, each one with a different mean and variance. The experiments were repeated several times with coverage f ranging from 50 to 0.5. The shape-based version of probabilistic partitioning (Partition) was compared to K-means and the recently introduced Clustered AGgretation Tool (CAGT). The latter was used with two different distance metrics, Euclidean and correlation (henceforth denoted as CAGT (Euclidean) and CAGT (correlation)). CAGT differs from the other two methods in that it tries



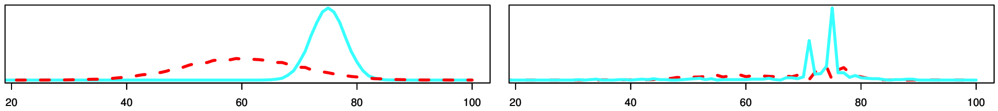
(a1) data: individual classes

(a2) data: individual classes



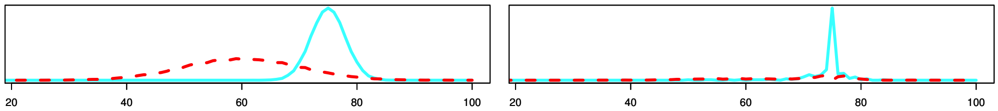
(b1) CAGT (Euclidean)

(b2) CAGT (Euclidean)



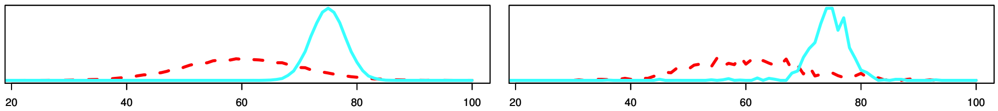
(c1) CAGT (correlation)

(c2) CAGT (correlation)



(d1) K–means

(d2) K–means



(e1) partition

(e2) partition

Figure 4.1 – ***Simulated data without shifts or flips.*** *Shows the data and the patterns found using the the K-Means clustering method, CAGT methods, and the probabilistic partition method (shape-based without shift or flips). Sub-figures* $a1$*,* $b1$*,* $c1$*,* $d1$*, and* $e1$ *are for* $f = 50$ *and* $a2$*,* $b2$*,* $c2$*,* $d2$*, and* $e2$ *are for* $f = 1$*. Red is class 1 (class* $c1$ *in Table 4.1) and cyan is class 2 (class* $c2$ *in Table 4.1).*

to infer the number of classes from the data, a behaviour that can be partly controlled by the command line parameter “K-means/median”. For the sake of fair comparison, we changed the value of this parameter, so as to force the program to always return exactly two classes. For CAGT (Euclidean), the parameter k (the number of clusters for K-means/medians) was always set to 2, while for CAGT (correlation) it was set 2 when $f < 5$ and to the default value of 40 when $f \geq 5$. For the same reason, we disabled the flipping option with CAGT. During the test we observed that CAGT (correlation) returned an error when trying to process samples consisting of zeros only. We therefore eliminated these samples from the input data sets fed to CAGT (correlation). The number of EM iterations in the probabilistic partitioning method was set to 30 for any value of f. Here and in all subsequent experiments, we used the iterative version of EM, starting with an initial class consisting of the mean bin count vector taken over all samples.

The performance of the different methods was assessed in several ways: (i) by visual inspection of aggregation plots for the true and rediscovered classes (Figure 4.1) - in the case of probabilistic partitioning method the aggregation plot represents a probability-weighted average; (ii) by measuring the similarity between the true and rediscovered patterns as a Pearson correlation coefficient of the corresponding bin count profiles (Table 4.1); (iii) by comparing

Table 4.1 – *Results for simulated data without shifts or flips. Model accuracy is expressed as Pearson correlation coefficient between original and rediscovered patterns/classes. The percentage of samples attributed to a class is shown in parentheses. The classes c1 and c2 correspond to the red and cyan curves in Figure 4.1, respectively.*

f	50	10	5	2	1	0.5
K-means c1	1 (50%)	0.9986 (53.00%)	0.9905 (58.45%)	0.5588 (88.6%)	0.5732 (92.55%)	0.5576 (96.45%)
K-means c2	1 (50%)	0.9999 (47.00%)	0.9993 (41.55%)	0.7443 (11.4%)	0.6459 (7.45%)	0.4590 (3.55%)
CAGT (Euclidean) c1	1 (50%)	1.0000 (49.9%)	0.9990 (50.2%)	0.9742 (59.15%)	0.5730 (92.55%)	0.5802 (96.35%)
CAGT (Euclidean) c2	1 (50%)	1.0000 (50.1%)	0.9998 (49.8%)	0.9950 (40.85%)	0.6459 (7.45%)	0.4965 (3.65%)
CAGT (correlation) c1	1 (50%)	0.9994 (47.9%)	0.9956 (44.53%)	0.9829 (57.06%)	0.5498 (80.62%)	0.5874 (88.30%)
CAGT (correlation) c2	1 (50%)	0.9998 (52.1%)	0.9993 (55.47%)	0.9987 (42.94%)	0.6748 (19.38%)	0.4391 (11.70%)
Partition c1	1 (50%)	1.0000 (50.03%)	1.0000 (49.99%)	0.9989 (49.23%)	0.9929 (48.59%)	0.9407 (48.44%)
Partition c2	1 (50%)	1.0000 (49.97%)	1.0000 (50.01%)	0.9998 (50.77%)	0.9985 (51.41%)	0.9862 (51.56%)

Table 4.2 – *Classification error (in percentage) between the discovered patterns and their data classes.*

f	50	10	5	2	1	0.5
K-means	0	3.00	8.85	40.20	43.85	47.75
CAGT (Euclidean)	0	0.30	3.60	32.15	43.85	48.00
CAGT (correlation)	0	1.75	5.44	17.41	39.63	43.96
Partition	0	0.00	1.05	11.20	23.55	33.95

the re-estimated class frequencies to the true class frequencies of 50% (Table 4.1); (iv) by computing the classification error defined as the percentage of misclassified samples (Table 4.2). Classification error is calculated as $\left(\frac{N-cr1-cr2}{N}\right)100$, where $cr1$ and $cr2$ are number of samples from class 1 and class 2 respectively which were correctly classified as belonging to their respective classes, and N is the total number of samples in the data. In order to compute the classification error, we need to label the classes inferred by the various algorithms. Since the setup of the simulations involves only two classes, we could easily do this by hand. In addition for the probabilistic partitioning method, we need to give a deterministic class assignment for each sample and we give it to the most probable class.

As a general trend, we can see that all methods work well when the count coverage is high ($f \geq 10$). When there is a lower coverage, probabilistic partitioning clearly outperforms all other methods. In fact it recovers the underlying patterns of the two classes surprisingly well ($r > 0.94$) even at very low coverage ($f = 0.5$) and this in spite of a high classification error of about 33% (Table 4.2). The high classification error is probably due to the expected large number of samples consisting of zeros only (60%) all of which will be attributed to class $c2$ which has the higher estimated frequency (Table 4.1). K-means and CAGT (correlation) still recover the count frequency profiles of the two classes with reasonable accuracy at a coverage as low as $f = 2$. Note further that probabilistic partitioning is the only method capable of accurately estimating the frequencies of the two classes at low coverage. This is clearly related to the probabilistic rather than deterministic assignment of class membership.

Table 4.3 – *Results for simulated data with flips. Model accuracy is expressed as Pearson correlation coefficient between the original and rediscovered patterns/classes. The percentage of samples attributed to a class is shown in parentheses. The classes c1 and c2 correspond to the red and cyan curves in Figure 4.2, respectively.*

f	50	10	5	2	1
CAGT (correlation) c1	0.9999 (50%)	0.9996 (49.8%)	0.9990 (48.99%)	0.9946 (23.08%)	0.9918 (22.97%)
CAGT (correlation) c2	1.0000 (50%)	0.9999 (50.2%)	0.9998 (51.01%)	0.9791 (76.92%)	0.9598 (77.03%)
Partition c1	0.9999 (50%)	0.9996 (49.99%)	0.9991 (50.05%)	0.9986 (50.06%)	0.9965 (50.13%)
Partition c2	1.0000 (50%)	0.9999 (50.01%)	0.9998 (49.95%)	0.9997 (49.93%)	0.9986 (49.87%)

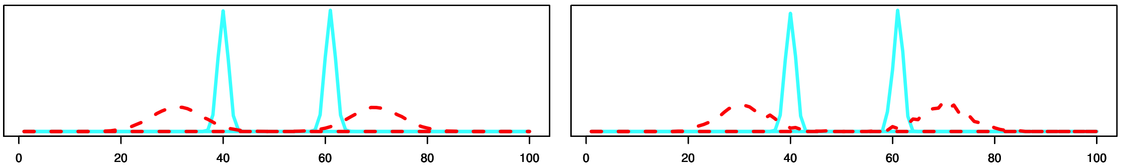
Data with flips.

The next thing we wanted to see was how well the method works when there are flips in the data. We used two classes as before. The simulated data now contains 2000 samples per class, 1000 presented in one orientation and 1000 in the reversed orientation. We compared probabilistic partitioning in shape-based mode to CAGT (correlation) with flipping enabled. Since CAGT (correlation) in default mode returned variable numbers of patterns for $f < 5$, we reduced the parameter k to 5 for $f = 2$ and to 4 for $f = 1$, in order to force the program to output only 2 classes. Overall, the results (Figure 4.2 and Table 4.3) were similar. The methods were able to recover the underlying patterns with high accuracy if coverage was not too low. At lower coverage, probabilistic partitioning worked better. Note, however, that in this test we had to increase the number of iterations from 30 to 70 to reach good performance with low coverage (for $f \leq 2$). In general it was seen that for very low values of f, we may need to increase the maximum number of EM iterations for this experiment. The probabilistic partitioning method is however seen to be robust over a wide range of EM iterations.

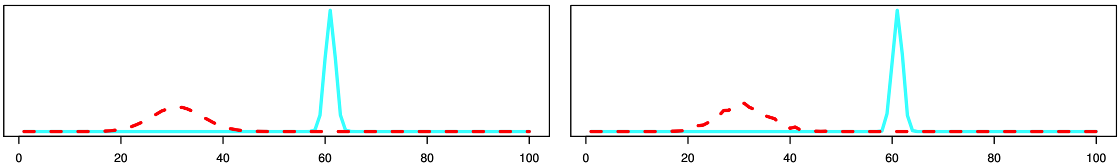
Additional tests with simulated data

We performed similar tests with mixtures of more than two classes and show that the probabilistic partitioning approach works well. As an example, we show the performance of the probabilistic partitioning algorithm on a simulated data set containing a mixture of four classes, each one represented by 1000 samples. The average total count coverage f was set to 5 for all classes. The results are presented in Figure 4.3. We first note that the algorithm is capable of discovering the four classes. Moreover, the virtually identical curves obtained for the original and rediscovered classes indicate that virtually all samples were correctly classified.

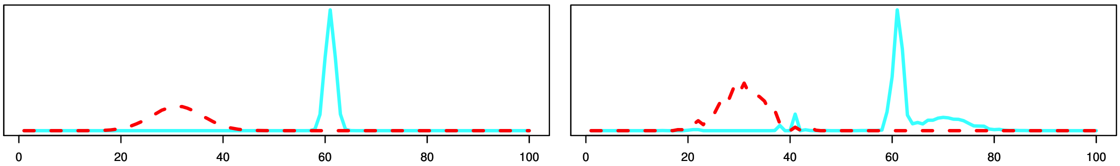
We were wondering whether probabilistic partitioning is able to discover two classes with similar count frequency distributions. To answer this question, we generated simulated data composed of two classes characterized by Gaussian distributions with identical center positions but different widths. As in previous tests, we generated 1000 samples per class. The total count coverage per sample was set to 50. We then tested four different methods: K-means, CAGT (Euclidean), CAGT (correlation), and shape-based probabilistic partitioning. The results are presented in Figure 4.4. We can see that all but one method perform well in this test. The ex-



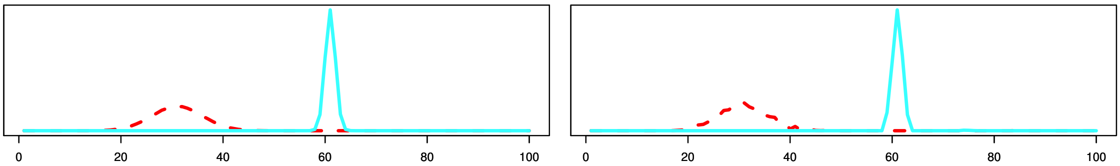
(a1) data: individual classes (a2) data: individual classes



(b1) data: flip and average (b2) data: flip and average



(c1) CAGT (correlation) (c2) CAGT (correlation)



(d1) partition (shape–based) (d2) partition (shape–based)

Figure 4.2 – ***Simulated data with flips.*** *Data (4000 samples) consist of two classes characterized by Gaussian-shaped patterns. Each class is represented by two subsets of 1000 samples, one showing the underlying pattern in native, the other one in reversed (flipped) orientation. Sub-figures* $a1$*,* $b1$*,* $c1$*, and* $d1$ *are for* $f = 50$ *and* $a2$*,* $b2$*,* $c2$*, and* $d2$ *are for* $f = 1$*.* $b1$ *and* $b2$ *are aggregation plots of the same data but with all samples presented in their native orientation. It can be seen that the probabilistic partitioning method (shape-based) using flips captures the actual data patterns at high (*$f = 50$*) and low (*$f = 1$*) coverage. The CAGT (correlation) method works well for* $f = 50$ *only. Red and cyan colors correspond to classes c1 and c2 in Table 4.3, respectively.*

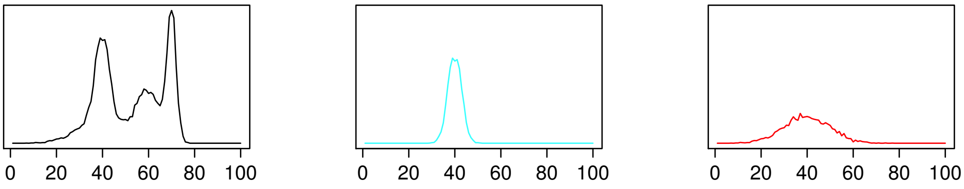
ception is CAGT (correlation) which reported only one class (Note that CAGT cannot be forced to report exactly 2 classes). Overall, this experiment confirms that probabilistic partitioning robustly performs well on a variety of data sets posing different kinds of difficulties.

4.3.2 On real ChIP-Seq data

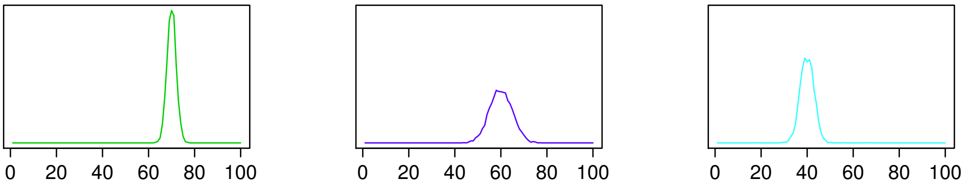
We now check the usefulness on the method on real data from ChIP-Seq experiments.

H3K4me1 and H3K4me3 promoter signatures

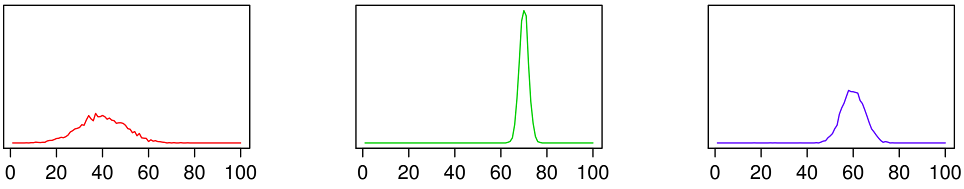
These two histone marks exhibit characteristic and distinct chromatin signatures around promoters. In the following experiment we mix H3K4me1 and H3K4me3 bin count profiles representing promoter regions to test whether automatic classification methods can correctly



(a) data (complete) (b) data: class 1 (c) data: class 2



(d) data: class 3 (e) data: class 4 (f) partition: class 1



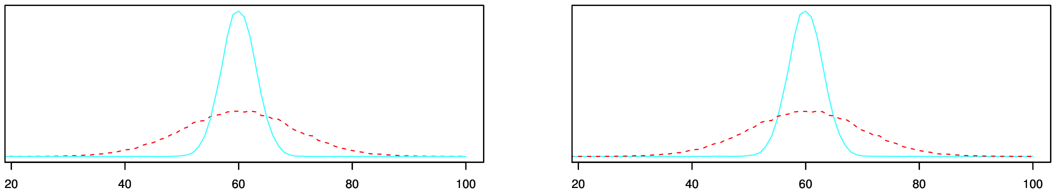
(g) partition: class 2 (h) partition: class 3 (i) partition: class 4

Figure 4.3 – ***Results from experiment with four classes.*** *(a)-(e) Aggregation plots (APs) for simulated data. (f)-(i) Probability weighted APs of samples as classified by the probabilistic partitioning algorithm.*

Table 4.4 – *Model accuracy (represented by Pearson correlation) and classification error between the discovered patterns and their data classes for the various methods. The time (in seconds) taken for each of the methods is also shown. Real data for H3K4me1 and H3K4me3 around TSS regions are mixed (34741 samples in each dataset with each sample containing 99 bins). The percentage of each class is shown in brackets. H3K4me1 and H3K4me3 stand for the 2 datasets. (Values are rounded to the fourth decimal place for model accuracy and two decimal places for classification error.)*

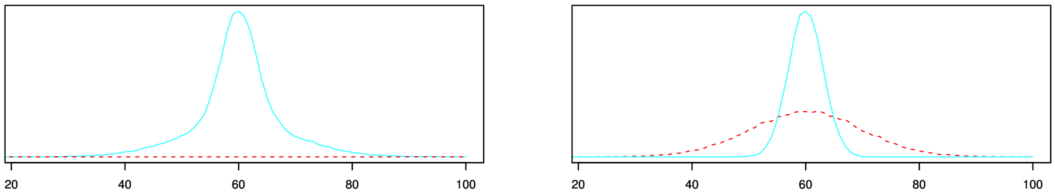
	Model accuracy H3K4me1	Model accuracy H3K4me3	Classification error	Time (sec)
K-means	0.0244 (83.65%)	0.9980 (16.35%)	33.72	1.16
CAGT (Euclidean)	0.9270 (69.03%)	0.9987 (30.97%)	23.85	106.31
CAGT (correlation)	0.9463 (42.86%)	0.9994 (57.14%)	26.98	108.35
Partition (non shape-based)	0.8959 (75.76%)	0.9997 (24.24%)	27.26	97.91
Partition (shape-based)	0.9713 (62.53%)	0.9996 (37.47%)	20.64	149.57

identify the two classes of samples and accurately reconstruct the corresponding chromatin signatures (i.e. bin count frequency profiles). As promoter collection, we used 34741 annotated transcription start sites (TSS) from ENSEMBL. We then extracted H3K4me1 and H3K4me3 tag counts from public ChIP-Seq data for mouse embryonic stem (ES) cells ([23], GEO entries



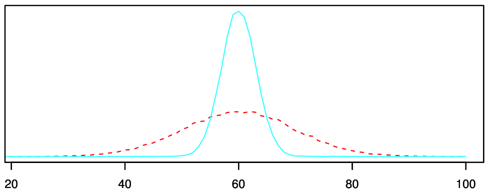
(a) data: individual classes

(b) CAGT (Euclidean)



(c) CAGT (correlation)

(d) K–means

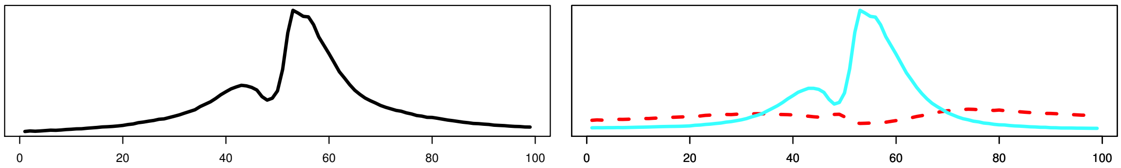


(e) partition

Figure 4.4 – ***Experiment with two classes characterized by co-localizing Gaussian peaks.*** *Shown are aggregation plots (APs) for the two classes in the data, and for the rediscovered classes obtained with different clustering algorithms. Note that CAGT (correlation) reports only one class. The absence of a second class is reflected by the horizontal red dashed line at height zero in subfigure (c).*

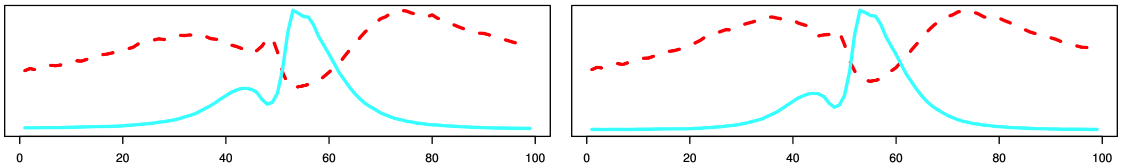
GSM594577 and GSM594581). For each sample, tags for H3K4me1 and H3K4me3 were counted in bins of 50 bp over a region of -2500 to +2500 relative the TSS. The two datasets were then combined into one. The advantage of having such a combined dataset (by mixing two real datasets) is that we know the underlying truth and we can do the quantitative comparisons similar to what we have done using simulated experiments by trying to separate the two datasets from the combined dataset.

This test data set potentially poses several new difficulties as compared to the previous synthetic data sets. (i) The two classes are likely to be inhomogeneous themselves since not all promoters are active in ES cells and this is known to be reflected by the respective histone modification signatures. (ii) The two classes are highly unequal in terms of coverage ($f = 11$ for H3K4me1, $f = 90$ for H3K4me3). This explains why an aggregation plot of the mixed data set looks quasi-identical to an aggregation plot for H3K4me3 only (Figure 4.5(a), 4.5(b)). Since unequal coverage may help to distinguish between the two classes, we tested probabilistic partitioning in both basic and shape-based mode. (iii) This data set is much larger than the previously tested synthetic data sets and thus may represent a challenge in terms of CPU



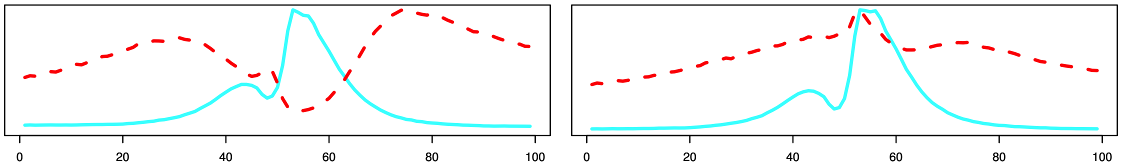
(a) data (complete): H3K4me1 and H3K4me3

(b) data: H3K4me1 (red), H3K4me3 (cyan) – globally normalized



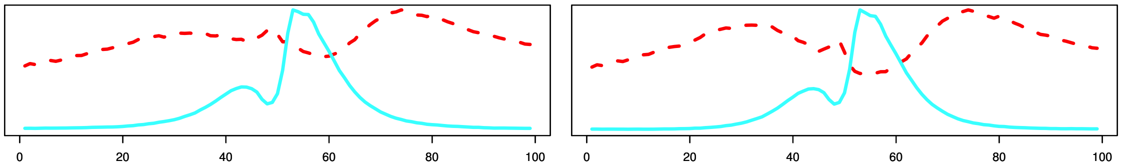
(c) data: H3K4me1 (red), H3K4me3 (cyan) – separately normalized

(d) CAGT (Euclidean)



(e) CAGT (correlation)

(f) K–means



(g) partition (non shape–based)

(h) partition (shape–based)

Figure 4.5 – ***H3K4me1 and H3K4me3 histone modification data.*** *H3K4me1 and H3K4me3 data mixed together and separated using the K-means, CAGT (correlation), CAGT (Euclidean), and probabilistic partitioning approach (non-shape and shape-based). Red is for the class which represents H3K4me1 and cyan is for H3K4me3 for figures (b) to (h). In the figures, each class is normalized so that the maximum value is 1 for the sake of clarity for each class. Only for sub-figure (b) we normalize using a global maximum of H3K4me1 and H3K4me3.*

requirements. We exploited this fact to carry out a speed comparison of the different programs.

In total, we tested 5 methods on this data set, K-means, CAGT (Euclidean), CAGT (correlation), partitioning (basic - non shape-based) and partitioning (shape-based). The results are shown in Figure 4.5 and Table 4.4. Not surprisingly, all methods perform well in reconstructing the H3K4me3 signature around promoters, which dominates the data set in terms of tag coverage. For the H3K4me1 signature, probabilistic partitioning (shape-based) performs best, followed by CAGT (correlation) and partitioning (basic). A possible explanation for this fact as that coverage is highly inhomogeneous within the H3K4me3 class, causing misclassification of low coverage H3K4me3 samples as H3K4me1 by the basic but not the shape-based version of probabilistic partitioning. It is noteworthy that CAGT (correlation) outperforms probabilistic partitioning in estimating the relative frequencies of the two classes. This may be due to the fact that CAGT (correlation) was tested on a reduced data set lacking samples with zeros only.

Regarding speed, we note that probabilistic partitioning (shape-based) is a little slower than

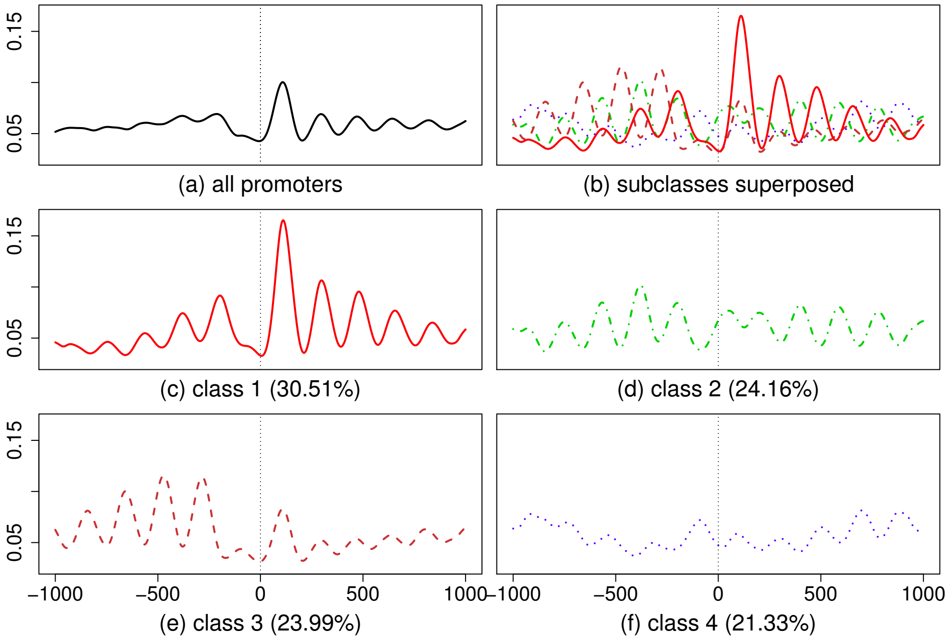


Figure 4.6 – ***Partitioning of nucleosome positioning patterns in human promoters.*** *All curves are drawn to the same scale. Probabilistic partitioning reveals strong oscillatory patterns for subclasses of promoters which partially cancel each other out when mixed together.*

CAGT but is still capable of processing the data sets in a few minutes. The speed figures should be interpreted with caution as they depend on the number of iterations carried out by the probabilistic partitioning algorithm. We further note that K-means is very fast but basically incapable of recovering the two histone modification signatures.

Application to nucleosome positioning in promoters

In the previous example, we have shown that our method can separate H3K4me1 and H3K4me3 signals that are artificially pooled together. Such a test is useful for method validation but obviously not representative of an interesting biological application. In the following, we apply probabilistic partitioning to a potentially inhomogeneous data set where the subclasses are not known in advance. Specifically, we analyze the positioning of nucleosomes in human promoters. As anchor points we use 9714 precisely mapped TSSs from EPDnew version 1 [32]. Nucleosome mapping data produced by MNase digestion were taken from [106]. Prior to partitioning, the mapped MNase tags were shifted by 70 bp towards the center of the nucleosome and then counted in bins of 20 bp. Thus the input data vectors reflect the frequency at which a nucleosome center occurs at a given distance from a TSS.

The AP plot for the complete promoter set (Figure 4.6) shows a well positioned +1 nucleosome flanked downstream by a damped oscillatory pattern with the expected period of about 200 bp. The region immediately upstream of the TSS appears to be nucleosome-free. No clear

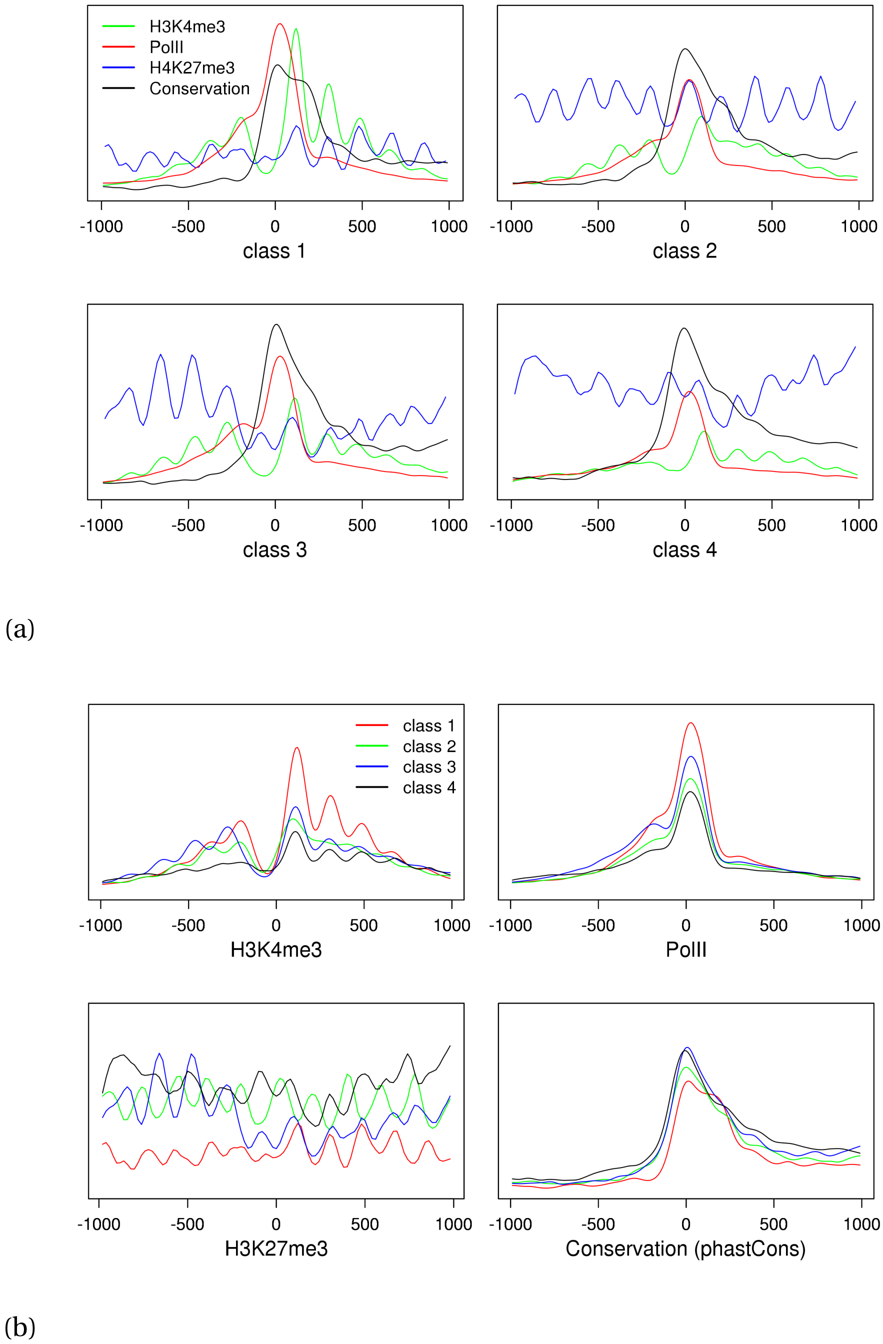


Figure 4.7 – ***Aggregation plots (APs) for various genomic features in different promoter classes.*** *(a) APs for different features superposed for the same class. (b) Same feature for different classes in one plot. Position zero corresponds to the transcription start site.*

oscillatory pattern is seen in the promoter upstream region. The absence of an oscillatory pattern could mean that nucleosomes are randomly positioned or that different promoters have regularly positioned nucleosomes with different phase shifts relative to the TSS. We used shape-based probabilistic partitioning with limited shifting (± 1 bins/20 bp) to discriminate between these two alternatives. The results obtained with $K = 4$ are shown in Figure 4.6(b-f). With one exception (class 4), the class-specific AP plots show higher nucleosome peaks and stronger oscillatory patterns than the AP plot for the complete set. Therefore, we conclude that the absence of a periodic signal in the upstream region in Fig 4.6(a) promoters results from interference of periodic patterns with different phase shifts that almost entirely cancel out each other. We were wondering whether the four promoter classes with distinct nucleosome architectures may differ in terms of regulatory properties. To this end, we analyzed the distribution of an active and a repressive histone mark (H3K4me3 and H3K27me3) as well as Pol II in the same cell type. We analyzed four genomic features in four promoter classes with distinct nucleosome architectures (see Figure 4.6). Specifically, we plotted the ChIP-Seq signal for H3K4me3, H3K27me3 and Pol II [7] and average phastCons conservation scores [110] relative to the transcription start site (see Figure 4.7). Note that the nucleosome mapping data used for defining the promoter classes and the ChIP-Seq data used here were generated with the same cell type (resting CD4+ T cells). Several interesting observations can be made. For instance, class 1 is highly enriched in PolII and the active promoter mark H3K4me3, and thus appears to be the transcriptionally most active class. Classes 2 and 3 have elevated levels of the repressive H3K27me3 modification indicative of a role in development. Interestingly, sequence conservation is inversely correlated with active chromatin marks, suggesting that repressed or weakly expressed genes are subject to more complex regulation than strongly expressed genes. Among these differences, perhaps most interestingly, classes 2 and 3 show regularly positioned H3K27me3-labeled nucleosomes indicative of a repressed state.

Shape-based peak evaluation with shifting

In this example, we apply probabilistic partitioning to improve a publicly available peak list originating from a ChIP-Seq experiment against a sequence-specific DNA-binding protein. Note that this application is different from the previous ones in that we are not trying to discover distinct classes. We are merely trying to separate typical examples (belonging to the majority class) from atypical examples, assuming that atypical examples are contaminants. The second goal is to refocus the peak center positions. To reach these objectives, we use shape-based probabilistic partitioning with two classes, one corresponding to the majority class and trained during EM, the other one with a flat count distribution representing background and not modified during EM. As output, we obtain for each peak region in the input list a probability of being a true binding site plus an optimal shifting distance under the true peak model.

To test this approach we used ChIP-Seq data for CTCF in HUVEC from Broad/ENCODE downloaded from GEO [6]. As anchor points we used the midpoints of the CTCF binding regions given in the peak file included in the GEO sample entry (GSM733716). For each binding region, we counted sequence tags in bins of 10 bp within a 1 kb region around the anchor point. Probabilistic shifting was done by evaluating the ChIP-Seq signal in 31 overlapping windows

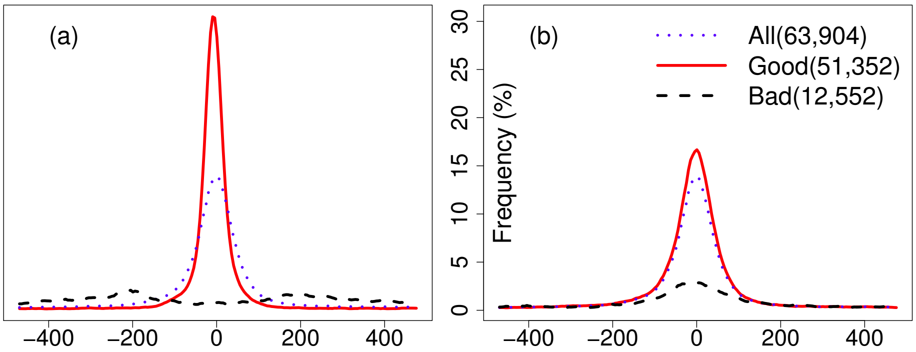


Figure 4.8 – ***Shape-based peak evaluation with shifting.*** *The figure illustrates the effects of probabilistic partitioning on a CTCF peak list provided by ENCODE in terms of motif enrichment. (a) Probabilistic partitioning with shifting. (b) Partitioning based on original p-values. Method details: CTCF binding motifs where identified by scanning the DNA sequence around peak centers with the JASPAR matrix MA0139.1 at a P-value threshold of* 10^{-5}*. The percentage of sequences containing a CTCF motif is plotted in a sliding window of 50 bp. The numbers in parentheses indicate the sizes of the peak lists. For fair comparison, the threshold for partitioning with the original P-values was chosen such as to match the numbers of good and bad peak obtained with probabilistic partitioning. The motif enrichment profile for the complete peak list (dotted line) is included in both graphs.*

of 700 bp (70 bins). After partitioning, we split the input peak list into a “good” and a “bad” peak class, applying a threshold probability of 0.5. We also shifted the center positions of the good peaks based on the posterior probability distribution over the 31 shift classes. We then evaluated the peak lists obtained in this way by motif enrichment using the CTCF position weight matrix from the JASPAR database [96]. The details of how the JASPAR CTCF is scanned against the genome is given in http://ccg.vital-it.ch/pwmscan/ which is the website we used for this purpose. Figure 4.8(a) shows the frequency of CTCF binding motifs around the peak center positions. We note an essentially flat ChIP-Seq signal distribution for the bad peaks and a drastically enhanced Gaussian-like distribution with increased height and narrower width for the shifted good peaks. Given the relatively small size of the bad peak set (12552 out of 63904), the increase in peak height primarily results from shifting and only to a lesser extent from false binding sites elimination.

As a control, we split the same peak list into good and bad examples using the p-values contained in the file downloaded from GEO. (The probability threshold was chosen such as to match the numbers of the subsets obtained with probabilistic partitioning.) With this filtering criterion, the aggregation plot for the bad peak set still shows a low Gaussian-shaped signal distribution suggesting the retention of a few true binding sites, whereas the good peaks exhibit only a modest increase in signal height (Figure 4.8(b)). The latter was expected since these peaks were not subjected to optimal shifting.

We also evaluated probabilistic peak ranking in terms of reproducibility, using a GEO sample which provides separate peak lists for replicates. This analysis is based on CTCF ChIP-Seq data for the K562 cell line downloaded from replicate GEO entries GSM749690 and GSM749733.

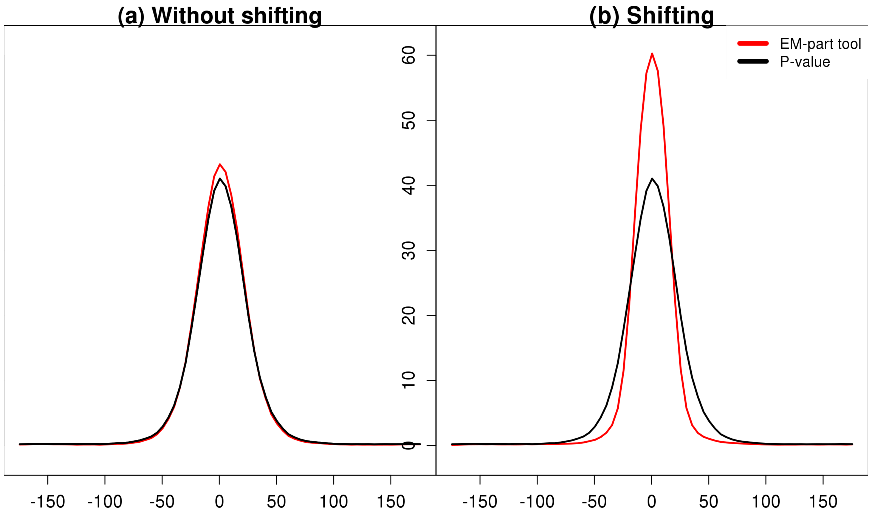


Figure 4.9 – ***Motif enrichment profiles for IDR-selected ChIP-Seq peak lists.*** *Peak sets were selected at an IDR of 1%.*

For peak shape evaluation, we used the shape-based version of probabilistic partitioning with and without shifting. As anchor points for extracting CTCF tag count profiles, we used the midpoints of the peak regions given in the "narrowPeak" file provided by the data submitters. IDR (irreproducible discovery rate) analysis was carried out with a *R* script downloaded from: https://sites.google.com/site/anshulkundaje/projects/idr. P-values were used as ranking measure for the original peak lists. In the case of probabilistic partitioning, we replaced the P-values in the original peak files with the posterior probabilities of the samples for the "good" class, prior to IDR analysis. With shifting enabled, we also shifted the peak regions by the most probable shift distance prior to IDR analysis. At an irreproducible discovery rate (IDR) of 1%, we obtain the following numbers of peaks with the different peak ranking methods: original P-values 22069, probabilistic partitioning with (without) shifting 19061 (18993). We then generated CTCF motif enrichment profiles for the three peak collections, in order to estimate their relative enrichment in biologically relevant sites. We noticed that our probabilistic evaluation of peak shapes results in a slightly higher enrichment in CTCF motifs as compared to the original P-value ranking (Figure 4.9(a)) . Furthermore, probabilistic shifting leads to a more focused motif distribution around estimated peak centers (Figure 4.9(b)). Therefore to summarize, at an equivalent IDR of 1%, our method finds slightly fewer peaks than the peak-finder used by the data submitters though our peak list was more enriched in CTCF motifs. A possible interpretation of these findings is that our method, which attempts to eliminate peaks of atypical shape, removes artifacts that are reproducibly called by other peak-finders.

Taken together, our results show that probabilistic partitioning is an effective post-processing method for filtering and focusing a publicly available ChIP-Seq peak list obtained with a state-of-the-art peak finder.

4.4 Conclusion

We presented a probabilistic partitioning method to find significant patterns in ChIP-Seq data (which works for both histone modification and transcription factor data). The corresponding algorithm runs in $O(n)$ time given a fixed number of classes and EM iterations. It is capable of processing large datasets (tens of thousands of samples) in minutes. The method is conceptually simple yet very flexible, and has been implemented in a few lines of *R* code. The basic partitioning algorithm is readily adjusted to handling flips and shifts following standard principles of EM. With low data coverage, the probabilistic partitioning method gives excellent model accuracy, superior to K-means or CAGT when tested on the same data examples. We have further shown that probabilistic partitioning can serve other purposes than pattern discovery and classification, like partitioning of nucleosome positioning patterns in human promoters, and shape-based evaluation and re-focusing of ChIP-Seq peaks from published peak lists.

5 Cell-type trees

5.1 Introduction

In the previous two chapters (chapters 3 and 4) we saw how to process ChIP-seq data and get some interesting regions from it using some statistical or computational techniques. In this chapter, we do a higher level analysis, once we have the process ChIP-Seq histone modification data (we use peak data in this chapter)

As we have seen before, in developmental biology, the process by which a less specialized cell becomes a more specialized cell type is called cell differentiation. Since all cells in one individual organism have the same genome, epigenetic factors and transcriptional factors play an important role in cell differentiation [71, 75, 76]. Thus a study of epigenetic changes among different cell types is necessary to understand cell development.

Histone modifications form one important class of epigenetic marks; such modifications have been found to vary across various cell types and to play a role in gene regulation [10]. A study of how histone marks change across various cell types could play an important role in our understanding of developmental biology and how cell differentiation occurs, particularly as the epigenetic state of chromatin is inheritable across cell generations [81].

In this chapter, we provide a definition for a *cell-type tree*. Cell-type trees are trees which represent the relationships between various cell-types. The nodes of this tree represent cell-types while the edges between two nodes tell us that one cell-type is differentiated from some cells of the other cell-type. It is not necessary that these various cell-types come from one individual, and therefore cell-type trees are different from cell-lineage trees. Cell-lineage trees, reconstructed from genomic variability caused by somatic mutations, represent the history of cell division in one individual organism from the very first cell, the zygote [44]. However we know that almost the entire genome (within one individual) is the same across cell-types; and it is also highly similar between individuals of the same species. However we know that epigenomic states are different across various cell types. So it is possible that in the cell differentiation process, a complex interplay between histone modifications, DNA methylation, transcription factors etc. plays an important role in how cells of various cell-types in one organism behave differently although the genome is almost same. Therefore in this study we attempt to build cell-type trees by looking at histone modification data. Currently we look at only some histone modifications for the sake of simplicity. We do this to see if there is a link between histone modifications and cell differentiation. Although it is possible (though not always) that we have some knowledge of the topology of a cell-type tree from known biology, inferring such trees from just histone modification data is important to see if there is correlation between histone modifications and cell differentiation. We note that in literature certain clustering techniques like hierarchical clustering have been used to cluster cell types using various kinds of data. For example, in [51] unsupervised hierarchical clustering of whole genome expression data was done for some cell-types.

Since cell differentiation transforms less specialized cell types into more specialized ones and since most specialized cells of one organ cannot be converted into specialized cells of some other organ, the paths of differentiation together form a tree, in many ways similar to the phylogenetic trees used to represent evolutionary histories. In evolution, present-day species have evolved from some ancestral species, while in cell development the more specialized cells have evolved from less specialized cells. Moreover, observed changes in the epigenetic state are inheritable, again much as mutations in the genome are (although, of course, through very different mechanisms and at very different scales); and in further similarity, epigenetic traits are subject to stochastic changes, much as in genetic mutations. (It should be noted that we are interested here in populations of cells of a certain type, not all coming from the same individual, rather than in developmental lineages of cells within one individual.) Finally, one may object that derived and more basic cell types coexist within the body, while phylogenetic analysis places all modern data at the leaves of the tree and typically qualifies internal nodes as “ancestral". However, species in a phylogenetic tree correspond to paths, not to nodes (since a species exists for a certain duration of time and the actual species may correspond to parts of branches). In particular, a species that has survived millions of years until today and yet has given rise to daughter species, much like a basic cell type that is observed within the organism, but from which derived cell types have also been produced and observed, is simply a path to a leaf in the tree, a path along which changes are slight enough not to cause a change in identification. (The time scale makes such occurrences unlikely in the case of species phylogenies, but the framework is general enough to include them.)

Therefore it may be possible to use or adapt some of the techniques used in building phylogenetic trees for building *cell-type trees* [89, 90]. The major difference between phylogenetic trees and cell-type trees is that functional changes in cell differentiation are primarily driven by programmed mutational events rather than by selection. An immediate consequence is that the design of an “evolutionary” model has hardly begun in sharp contrast to sequence evolution. However, note that the program of mutational events is itself the result of evolution, so that, as observed by Arendt [3], the cell differentiation tree often recapitulates the phylogeny of cell types. Thus we felt justified to apply phylogenetic methods to the analysis of cell types.

In this chapter, we provide evidence that such a scenario is possible. We do this by proposing new data representation techniques, distance measures, then by applying standard phylogenetic methods to produce biologically meaningful results. We used data on a few histone modifications (but mostly on H3K4me3) for many cell types, including replicate data, to construct cell-type trees—to our knowledge, these are the first such trees produced by computational methods. We show that preprocessing the data is very important: not only are ChIP-Seq data fairly noisy, but the ENCODE data are based on several individuals and this adds an independent source of noise. We propose a new peak-finding method to address this problem. We show how various patterns of histone modification change during the cell-differentiation process and the biological significance of it. We also outline some of the computational challenges in the analysis of cell differentiation, opening new perspectives that may prove of interest to computer scientists, biologists, and bioinformaticians. We also discuss how these cell-type trees can be used to study the evolution of cell types.

5.2 Methods

5.2.1 Model of differentiation for histone marks

We assume that histone marks can be independently gained or lost in regions of the genome as cells differentiate from a less specialized type to a more specialized one. Histones marks are known to disappear from less specialized cell types or to appear in more specialized ones and are often correlated with gene expression, so our assumption is reasonable. The independence assumption simply reflects our lack of knowledge, but it also enormously simplifies computations.

5.2.2 Data representation techniques

The analysis of ChIP-Seq data typically starts with a peak-finding step that defines a set of chromosomal regions enriched in the target molecule. We therefore use peak lists as the raw data for our study. We use both publically available peak lists (give in ENCODE database) and also define our own ‘peak-finder’ which basically identifies regions of the genome which have significant amounts of histone modification signal (see sub-section “Peak-finding" described later). We can decide on the presence or absence of peaks at any given position and treat this as a binary character, matching our model of gain or loss of histone marks. Since all of the cell types have the same genome (subject only to individual SNPs or varying copy numbers), we can compare specific regions across cell types. Therefore we code the data into a matrix in which each row is associated with a different ChIP-Seq library (a different cell type or replicate), while each column is associated with a specific genomic region.

We use two different data representations for the peak data for each cell type. Our first method is a simple windowing (or binning) method. We divide the genome into bins of certain sizes; if the bin contains at least one peak, we code it 1, otherwise we code it 0. The coding of each library is thus independent of that of any other library.

Our second method uses overlap and takes into account all libraries at once. We first find interesting regions in the genome, based on peaks. Denote the ith peak in library n as $P^n_i = [P^n_{iL}, P^n_{iR}]$, where P^n_{iL} and P^n_{iR} are the left and right endpoints (as basepair indices). Consider each peak as an interval on the genome (or on the real line) and build the *interval graph* defined by all peaks in all libraries. An interval graph has one vertex for each interval and an edge between two vertices whenever the two corresponding intervals overlap [41]. We simply want the connected components of the interval graph.

Definition 1 An interval in the genome is an *interesting region* iff it corresponds to a connected component of the interval graph.

A straight forward algorithm to identify these interesting regions in linear time is shown below:

We assume that we have a set of sorted peaks given to us with respect to their positions in each chromosome, otherwise we first sort the peaks.
Choose a chromosome, let PS be its set of peaks, set $AS = \{\varnothing\}$ and $z = 0$, and enter the following loop:

1. $P^*_{i^*} = \arg\min_{P^n_i \in PS} P^n_{iL}$. Set $a = P^*_{i^*L}$ and $AS = AS \cup \{P^*_{i^*}\}$
2. Set $S = \{P \mid P \cap P^*_{i^*} \neq \varnothing \text{ and } P \in PS\}$ and $AS = AS \cup S$.
3. If S is not empty, then find $P^*_{i^*} = \arg\max_{P^n_i \in S} P^n_{iR}$ and go to step 2.
4. Let $b = P^*_{i^*R}$ and set $PS = PS - AS$.
5. The interesting region lies between a and b, $IR[a,b]$. Let $D^n_{IR}[z]$ be the data representation for $IR[a,b]$ in library n. Set $z = z+1$. Set $D^n_{IR}[z] = 1$ if there is a peak in library n that lies in $IR[a,b]$; otherwise set $D^n_{IR}[z] = 0$ $(1 \leq n \leq N)$.

Repeat this procedure for all chromosomes in the genome. The above algorithm can implemented by sweeping from left to right (two ends of each chromosome) and we visit each peak only once. Therefore the algorithm takes time linear in the size of the number of all peaks in order to identify all the interesting regions. Figure 5.1 shows an example of interesting region as defined by overlap representation.

For a given collection of libraries, these interesting regions have a unique representation. We assume that it is in these interesting regions that histone marks are lost or gained and we consider that the size of the histone mark (which depends at least in part on the experimental procedures and is typically noisy) does not matter. Our major reason for this choice of representation is noise elimination: since the positioning of peaks and the signal strength both vary from cell to cell as well as from test to test, we gain significant robustness (at the expense of detail) by merging all overlapping peaks into one signal, which we use to decide on the value of a single bit. The loss of information may be illusory (because of the noise), but in any case we do not need a lot of information to build a phylogeny on a few dozen cell types.

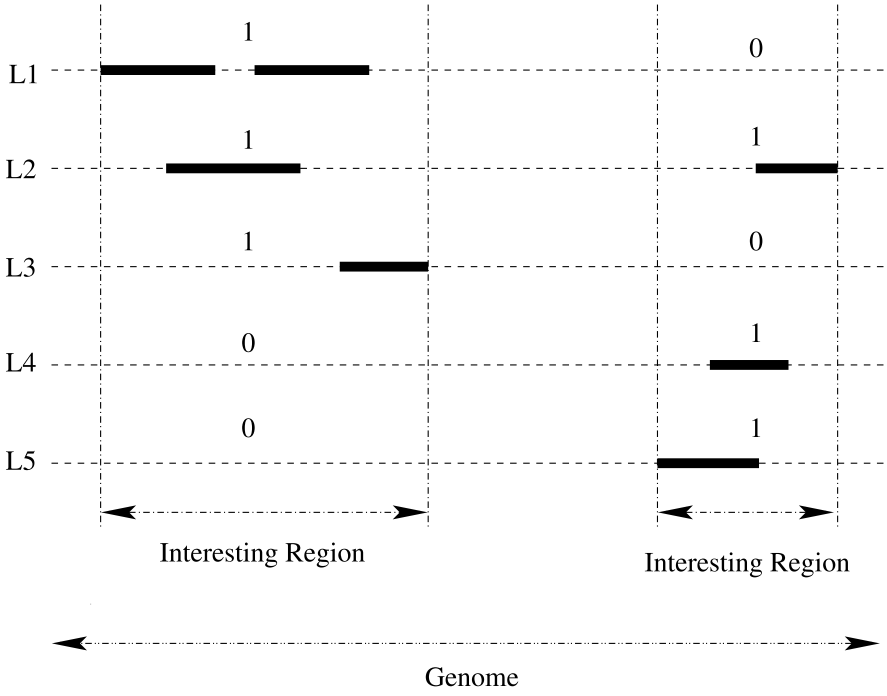


Figure 5.1 – *An example of an interesting region as defined by overlap representation is shown. The dotted horizontal lines represents a portion of the genome. Each row stands for a separate ChIP-Seq library (L1-L5). The dark lines represent peak regions. 1 and 0 is the data representation for each library.*

5.2.3 Phylogenetic analysis

Phylogenetic analysis attempts to infer the evolutionary relationships of modern species or *taxa*—they could also be proteins, binding sites, regulatory networks, etc. The best tools for phylogenetic inference, based on maximum parsimony (MP) or maximum likelihood (ML), use established models of sequence evolution, something for which we have no equivalent in the context of cell differentiation. However, one class of phylogenetic inference methods, so-called distance-based methods, are founded on hierarchical clustering under some suitable measure of pairwise distance for similarity. This type of method is directly applicable to our problem, provided we can define a reasonable measure of distance, or similarity between cell types in terms of our data representations. (We are not implying that models of differentiation do not exist nor that they could not be derived, but simply stating that none exist at present that could plausibly be used for maximum-likelihood phylogenetic inference.) Finally note that, with 0/1 data, we can also use an MP method, in effect assuming that all changes are equally likely.

In a cell type tree, most cell types coexist in the present; thus at least some of them can be found both at leaves and at internal nodes. (We may not have data for all internal nodes, as we cannot claim to have observed all cell types.) Fortunately, phylogenetic inference still works in such cases: as mentioned earlier, when the same taxon should be associated with both a leaf and an internal node, we should simply observe that each edge on the path from that internal node to that leaf is extremely short, since that distance between the two nodes should be zero (within noise limits). The tree inferred will have the correct shape; however, should we desire to reconstruct the basic cell types, then we would have to *lift* some of the leaf data by copying them to some internal nodes.

Of the many distance-based methods, we chose the most commonly used one, Neighbor-Joining (NJ) [104]. While faster and possibly better distance-based methods exist, such as FastME [29], it was not clear that their advantages would still obtain in this new domain; and, while very simple, the NJ method has the advantage of not assuming a constant rate of evolution across lineages. In each of the two data representation approaches, we compute pairwise distance between two libraries as the Hamming distance of their representations. (The Hamming distance between two strings of equal length is the number of positions at which corresponding symbols differ.) We thus obtain a distance matrix between all pairs of histone modification libraries; running NJ on this matrix yields an unrooted tree. For MP, we used the TNT software [47].

5.2.4 On the inference of ancestral nodes

We mentioned that lifting some of the leaf data into internal nodes is the natural next step after tree inference. However, in general, not all internal nodes can be labelled in this way, due mostly to sampling issues: we may not have observed the type that should be associated with a particular internal node, or we may be missing enough fully differentiated types that some internal tree nodes do not correspond to any real cell type. Thus we are faced with a problem of ancestral reconstruction and, more specifically, with three distinct questions:

- For a given internal node, is there a natural lifting from a leaf?
- If there is no suitable lifting, is the node nevertheless a natural ancestor—i.e., does it correspond to a valid (real) cell type?
- If the node has no suitable lifting and does correspond to a valid cell type, can we infer its data representation?

These are hard questions, in terms of both modelling and computational complexity; they are further complicated by the noisy nature of the data. Such questions remain poorly solved in standard phylogenetic analysis; in the case of cell-type trees, we judged it best not to address these problems until the tree inference part is better understood and more data are analyzed.

5.2.5 Peak-finding

Since our algorithms work on peak data, one needs to use some peak finder to convert the ChIP-Seq histone modification libraries into peaks. One can use any peak finder. We used the publically available peaks given by the ENCODE project for our analysis.

Since we found the peaks to be noisy, we also tried to develop our own 'peak-finder'. In chapter 3, we had proposed a two stage statistical method called ChIPnorm for identifying differential regions between two ChIP-Seq histone modification libraries. It normalizes the data, identifies significant and enriched regions, and finally identifies differential regions. This method was shown to be significantly better than the previously proposed methods. Now in this chapter, we used one of the pre-filtering steps of the ChIPnorm method to identify significant regions of individual ChIP-Seq libraries and we use them as 'peak' regions. We call this adapted method ChIPnormSig. Regions of the genome which are significantly enriched compared to a theoretical random distribution (which is estimated) are declared as significant (Figure 5.2). We used an amplified binomial distribution to represent this theoretical random distribution. Details on the theoretical distribution and how to infer its parameters from the data can be found in chapter 3. A false discovery rate threshold of 5% was used to identify significant regions as that is the value commonly used in literature (though this threshold value can be changed).

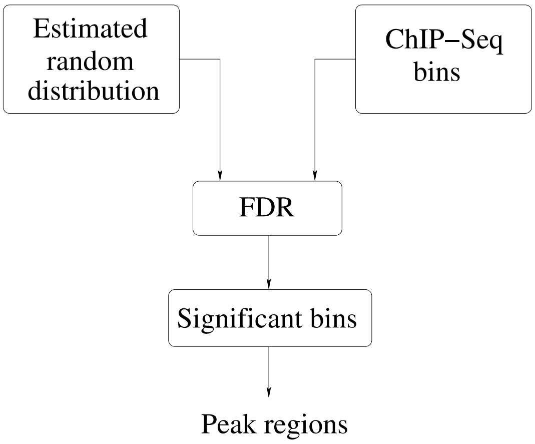


Figure 5.2 – *ChIPnormSig method to identify significant regions (which are used as peak regions) from histone modification ChIP-Seq library.*

5.3 Results and Discussion

Experimental Design

The histone modification ChIP-Seq data were taken from the ENCODE project database (UW ENCODE group) for human (hg19) data [82, 21]. We carried out experiments on H3K4me3 and H3K27me3 histone mark data from University of Washington (UW) ENCODE group and on H3K4me1, H3K9me3, H3K27ac histone mark data from Broad ENCODE group [82]. H3K4me3 is a well studied histone mark usually associated with gene activation, while the less well studied H3K27me3 is usually associated with gene repression [85]. We used data for cell types classified as "normal" and for embryonic stem cells—we did not retain cancerous or EBV cells as their differentiation processes might be completely distinct from those of normal cells. The ENCODE project provides peaks of ChIP-Seq data for each replicate of each cell type. We therefore used their peaks as the raw input data for our work. For the windowing representation, we used bins of 200 bp: this is a good size for histone marks, because 147 bp of

DNA wrap around the histone and linker DNA of about 80 bp connect two histones, so that each bin represents approximately the absence or presence of just one histone modification. We programmed our procedures in *R* and used the NJ implementation from the *ape* library in *R*.

Table 5.1 shows the list of the 37 cell types (72 libraries including all replicates) used for H3K4me3 data and 13 cell types (23 libraries including all replicates) for H3K27me3 data, giving for each an abbreviation and a short description. The table also shows the 10, 11, and 12 cell types used for H3K4me1, H3K9me3, H3K27ac respectively. In addition, the cells are classified into various groups whose names are based on their cell type. Keratinocytes (NHEK) is included in the Epithelial group. We have two replicates for most cell types, but only one replicate for types HCFaa, HFF, and CD14, and three replicates for CD20. (CD20(1) is a B-cell from an African-American individual while CD20(2) and CD20(3) are from a Caucasian individual). The replicates are biological replicates, i.e., the data come from two independent samples. For human Embryonic Stem Cells (hESC) we have data for different days of the cell culture (day 0, 2, 5, 9, 14)) for H3K4me3 and H3K27me3 data, so we shall use hESC D2 (or hESC T2) to mean data for hESC cells on day 2. For each cell type, we shall mention the replicate number in brackets, unless the cell type has only one replicate. All our experiments are done using the neighbor-joining distance based approach unless otherwise mentioned. More information about where we collected ENCODE peak data from is given in appendix B.

H3K4me3 data on individual replicates

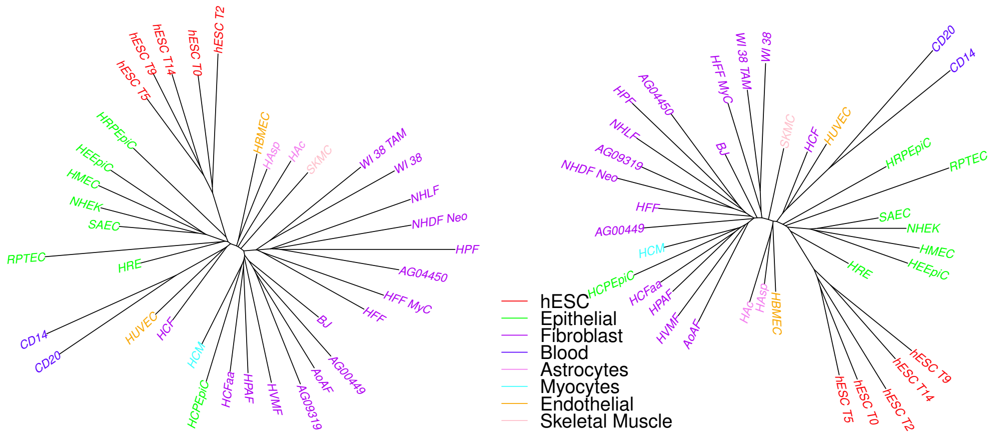
We report on our analyses using peak data from the ENCODE database for H3K4me3 histone modifications. We carried out the same analyses using H3K27me3 data, but results were very similar and so are not detailed here—we simply give one tree for comparison purposes. The similarity of results between the two datasets reinforces our contention that phylogenetic analyses yield biologically meaningful results on such data. We color-code trees to reflect the major groupings listed in Table 5.1.

Figure 5.3 shows the trees constructed using only one replicate for each cell type using both windowing and overlap representations. The color-coding shows that embryonic stem cells and blood cells are in well separated clades of their own, while fibroblasts and epithelial cells fall in just two clades each. Even within the hESC group we see that day 0 is far off from day 14 compared to its distance from day 2. Thus epigenetic data such as histone marks do contain a lot of information about cell differentiation history.

In order to quantify the quality of the groupings, we compute the total number of cells in a subtree that belong to one group. Since our groups are based on cell type only, there could be many subdivisions possible within each group. Therefore we choose the two largest such subtrees available for each group such that each subtree contains only the leaf nodes of that group. The results are shown in Table 5.2: most of the cell types in each group do cluster together in the tree.

Table 5.1 – *Cell types, short description, and general group for H3K4me3, H3K27me3, H3K4me1, H3K9me3, H3K27ac data. For details see the ENCODE website [101]. The mark ✓ shows the usage of that cell type for that particular histone mark.*

Cell Name	Short Description	Group	H3K4me3	H3K27me3	H3K4me1	H3K9me3	H3K27ac
AG04449	fetal buttock/thigh fibroblast	Fibroblast	✓				
AG04450	fetal lung fibroblast	Fibroblast	✓	✓			
AG09319	gum tissue fibroblasts	Fibroblast	✓				
AoAF	aortic adventitial fibroblast cells	Fibroblast	✓				
BJ	skin fibroblast	Fibroblast	✓	✓			
CD14	Monocytes-CD14+ from human leukapheresis production	Blood	✓	✓	✓	✓	✓
CD20(1)	B cells replicate, African American	Blood	✓				✓
CD20(2) and CD20(3)	B cells replicates, Caucasian	Blood	✓				
hESC	undifferentiated embryonic stem cells	hESC	✓	✓	✓	✓	✓
HAc	astrocytes-cerebellar	Astrocytes	✓				
HAsp	astrocytes spinal cord	Astrocytes	✓				
HBMEC	brain microvascular endothelial cells	Endothelial	✓				
HCFaa	cardiac fibroblasts- adult atrial	Fibroblast	✓				
HCF	cardiac fibroblasts	Fibroblast	✓				
HCM	cardiac myocytes	Myocytes	✓				
HCPEpiC	choroid plexus epithelial cells	Epithelial	✓				
HEEpiC	esophageal epithelial cells	Epithelial	✓				
HFF	foreskin fibroblast	Fibroblast	✓				
HFF MyC	foreskin fibroblast cells expressing canine cMyc	Fibroblast	✓				
HMEC	mammary epithelial cells	Epithelial	✓	✓	✓	✓	✓
HPAF	pulmonary artery fibroblasts	Fibroblast	✓				
HPF	pulmonary fibroblasts isolated from lung tissue	Fibroblast	✓				
HRE	renal epithelial cells	Epithelial	✓	✓			
HRPEpiC	retinal pigment epithelial cells	Epithelial	✓				
HSMM	skeletal muscle myoblasts	Skeletal Muscle			✓	✓	✓
HSMMtube	skeletal muscle myotubes differentiated from the HSMM cell line	Skeletal Muscle			✓	✓	✓
HUVEC	umbilical vein endothelial cells	Endothelial	✓	✓	✓	✓	✓
HVMF	villous mesenchymal fibroblast cells	Fibroblast	✓				
NHA	astrocytes (also called Astrocy)	Astrocytes			✓	✓	✓
NHDFAD	adult dermal fibroblasts	Fibroblast				✓	✓
NHDF Neo	neonatal dermal fibroblasts	Fibroblast	✓				
NHEK	epidermal keratinocytes	Epithelial	✓	✓	✓	✓	✓
NHLF	lung fibroblasts	Fibroblast	✓		✓	✓	✓
Osteobl	osteoblasts (NHOst)	Osteoblasts			✓	✓	✓
RPTEC	renal proximal tubule epithelial cells	Epithelial	✓				
SAEC	small airway epithelial cells	Epithelial	✓	✓			
SKMC	skeletal muscle cells	Skeletal Muscle	✓				
WI 38	embryonic lung fibroblast cells	Fibroblast	✓				



(a) *(b)*

Figure 5.3 – *Cell-type tree on H3K4me3 data (ENCODE peaks) using only one replicate: (a) windowing representation, (b) overlap representation.*

Figure 5.3 shows long edges between (most) leaf nodes and their parents—a disquieting feature, as it casts doubt as to the robustness of the tree, parts of which could be assimilated to

Table 5.2 – **Statistics for cell-type trees on H3K4me3 data.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. Rows correspond to various methods (WM: windowing; OM: overlap; TP: top peaks with threshold of 10). The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the NJ tree from the Hamming distance between the leaves. ENCODE means peaks from ENCODE data is used while ChIPnormSig means peaks from ChIPnormSig program is used. (one replicate) means only one replicate for each cell type is used, (all replicates) means all available replicates (1, 2, or 3) for each cell type is used, (profile) means a profile representation created using all replicates for each cell type is used. MP - maximum parsimony using TNT software.*

	hESC (5)	Epithelial (8)	Fibroblast (16)	Blood (2)	Astrocytes (2)	Myocytes (1)	Endothelial (2)	Skeletal Muscle (1)	SR	PD (%)
WM (one replicate)-ENCODE	5,0	6,1	8,4	2,0	1,1	1,0	1,1	1,0	0.93	3.20
OM (one replicate)-ENCODE	5,0	4,1	6,3	2,0	2,0	1,0	1,1	1,0	0.92	3.94
OM (one replicate)-ENCODE-MP	5,0	4,2	6,4	2,0	1,1	1,0	1,1	1,0	0.63	-
WM (one replicate)-ChIPnormSig	5,0	4,2	10,5	2,0	1,1	1,0	1,1	1,0	0.86	6.50
OM (one replicate)-ChIPnormSig	5,0	4,2	10,5	2,0	1,1	1,0	1,1	1,0	0.87	5.02
WM (all replicates)-ENCODE	5,0	6,1	11,2	2,0	1,1	1,0	1,1	1,0	0.84	3.30
OM (all replicates)-ENCODE	5,0	4,2	9,4	2,0	2,0	1,0	1,1	1,0	0.78	3.88
WM (all replicates)-ChIPnormSig	5,0	4,2	15,1	2,0	2,0	1,0	1,1	1,0	0.64	6.40
OM (all replicates)-ChIPnormSig	5,0	4,2	12,3	2,0	1,1	1,0	1,1	1,0	0.63	5.81
WM (all replicates)-TP-ENCODE	5,0	6,1	7,4	2,0	1,1	1,0	1,1	1,0	0.81	3.73
OM (all replicates)-TP-ENCODE	5,0	4,3	8,5	2,0	2,0	1,0	1,1	1,0	0.74	3.98
OM (profile)-ENCODE	5,0	4,3	12,2	2,0	2,0	1,0	1,1	1,0	0.90	4.05
OM (profile)-ChIPnormSig	5,0	4,2	12,3	2,0	1,1	1,0	1,1	1,0	0.85	4.84

star-shaped trees (a tree with only one internal node and the remaining nodes being leaves). To quantify this observation, we measured the SR ratio, defined as $SR = \frac{\sum_{e \in I} l(e)}{\sum_{e \in E} l(e)}$, where I is the set of all edges connecting leaf nodes to their parents, E is the set of all edges in the tree, and $l(e)$ is the length of edge e. If this ratio SR is close to 1, then the tree looks star-shaped with long branches to the leaves. This ratio was 0.93 using the windowing representation; using the overlap representation reduced it very slightly to 0.92. These long branches are due in part to the very high level of noise in the data, explaining why the overlap representation provided a slight improvement.

As a final entry in the table, we added another measure on the tree and the data. The NJ algorithm is known to return the “correct" tree when the distance matrix is ultrametric; the technical definition does not matter so much here as the consequence: if the matrix is ultrametric, then the sum of the length of the edges on the path between two leaves always equals the pairwise distance between those two leaves in the matrix. Thus one way to estimate how far the distance matrix deviates from this ideal is to compare its distances to the length of the leaf-to-leaf paths in the tree:

$$PD = \frac{\sum_{i,j} |NJ(i,j) - M(i,j)|}{\sum_{i,j} NJ(i,j)}$$

where i and j are leaf nodes, $NJ(i,j)$ is the tree distance between i and j, and $M(i,j)$ is the matrix distance between i and j. A high value of PD indicates that the data representations and measures do not fit well to any tree. We get very low values (of less than 4% for both windowing and overlap representations), suggesting that the distances we compute are in fact representative of a tree and thus offering confirmation of the validity of the inference.

Finally, the trees obtained using TNT software (MP based method) are very similar but we got a better SR ratio as shown in Table 5.2. The results using TNT software for overlap representation when using only one replicate of H3K4me3 data is shown in Figure 5.4.

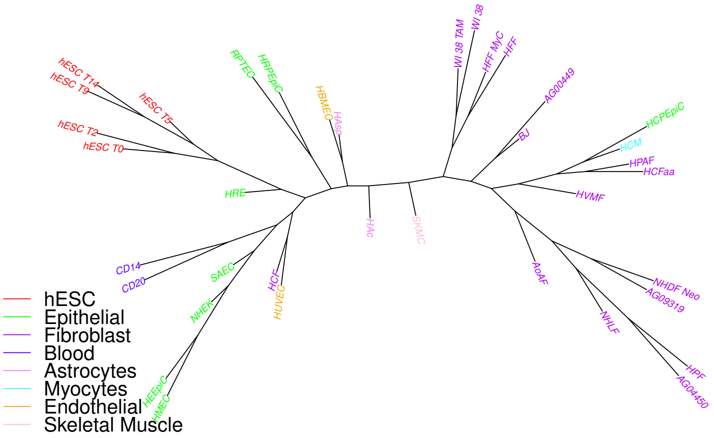
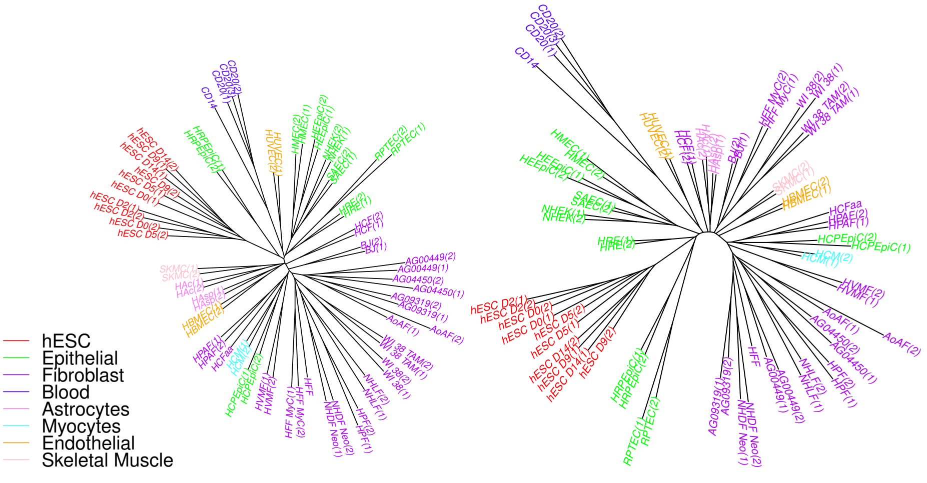


Figure 5.4 – *Using maximum parsimony (TNT software) on H3K4me3 data (ENCODE peaks) using only one replicate (overlap representation).*

H3K4me3 data with all replicates

By bringing replicates into the analysis, we can expect to see a stronger phylogenetic signal as each replicate adds to the characterization of its cell type. In particular, wherever we have two or more replicates, they should form a tight subtree of their own. We thus used our replicate data (two replicates for 33 of the 37 cell types, and three for one type, for a total of 72 libraries) in the same analysis pipeline. Figure 5.5 shows the differentiation trees obtained using windowing and overlap representations. We also include the same study (in overlap representation only) on H3K27me3 data in Figure 5.6. As expected, almost all replicates are grouped; since we usually have two replicates, we get a collection of “cherries" (pairs of leaves) where we had a single leaf before. In most cases, it is now the distance from each leaf in a cherry to their common parent that is large, indicating that the distance between the two replicates is quite large—as we can also verify from the distance matrix. This suggests much noise in the data. This noise could be at the level of raw ChIP-Seq data, but also due to the bias of peak-finding methods used—one expects a general-purpose peak finder to be biased against false negatives and more tolerant of false positives, but for our application we would be better served by the inverse bias. Another reason for the large distance is the nature of the data: these are biological replicates, grown in separate cultures, so that many random losses or gains of histone marks could happen once the cell is differentiated. Thus it may be that only a few of the variations in the data are correlated with cell differentiation. Identifying these few variations would be of high interest, but with just two replicates we are unlikely to pinpoint them with any accuracy.



(a) *(b)*

Figure 5.5 – *Cell-type tree on H3K4me3 data (ENCODE peaks, using all replicates): (a) windowing representation, (b) overlap representation.*

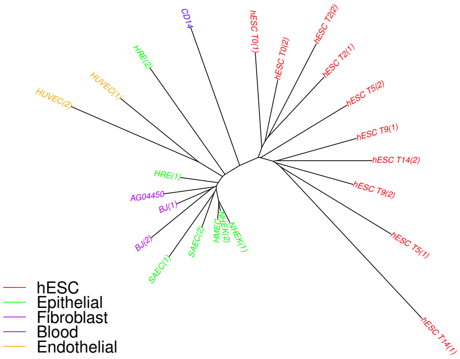


Figure 5.6 – *Cell-type tree on H3K27me3 data (ENCODE peaks), using all replicates and overlap representation.*

Looking again at Table 5.2, we see that, using the windowing representation, the value of SR for the full set of replicates is 0.84 and that here the overlap representation, which is more effective at noise filtering, yields an SR value of 0.78. This is a substantial reduction and indicates that the long edges are indeed due to noise. The PD percentage values remain very low for both representations, so the trees we obtained do represent the data well. Note that the groupings appear (in the color-coding in the figure) somewhat better than when we used only one replicate, and the values in columns 2 through 9 of Table 5.2 confirm this impression.

We also include results using windowing representation on H3K4me1 data, H3K9me3 data, and H3K27ac data in Figs. 5.7, 5.8, 5.9 respectively. We got good results on these datasets as seen from these figures as well.

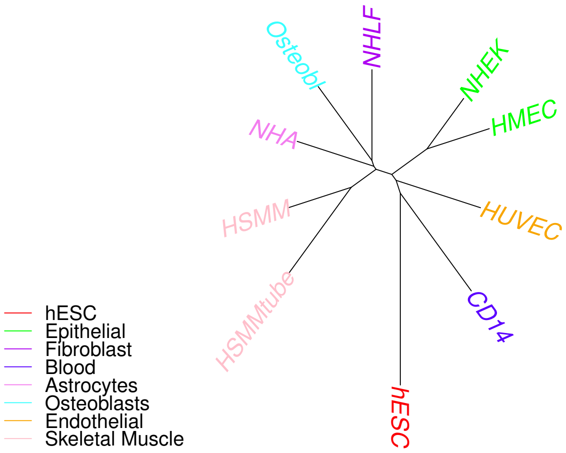


Figure 5.7 – *Cell-type tree using windowing representation on H3K4me1 data.*

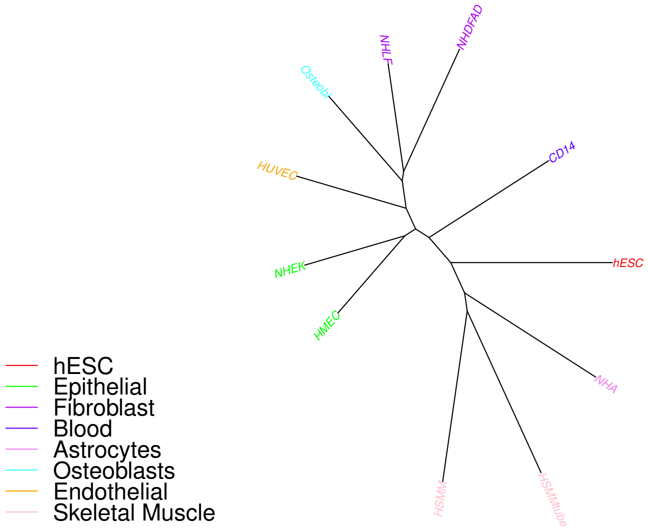


Figure 5.8 – *Cell-type tree using windowing representation on H3K9me3 data.*

Using top peaks, masking regions, IDR analysis

In order to study the nature of the noise, we removed some of the less robust peaks. The ENCODE dataset gives a p-value for each peak listed; we kept only peaks with (negative) log

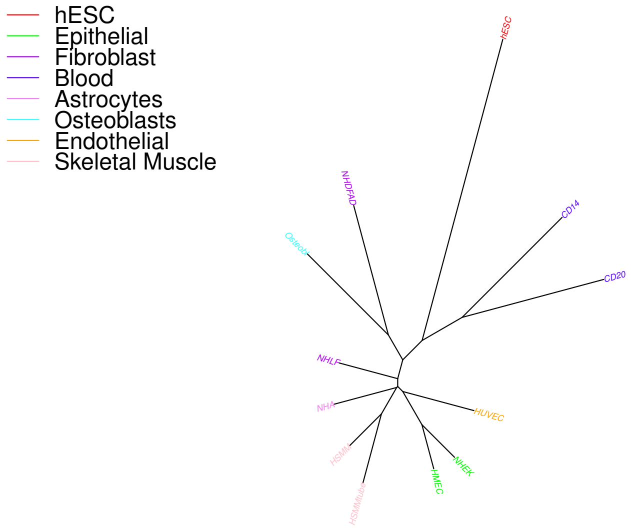
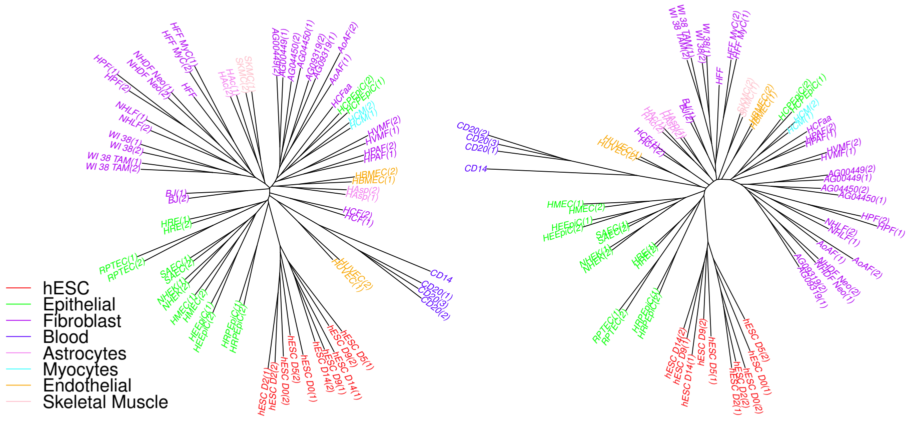


Figure 5.9 – *Cell-type tree using windowing representation on H3K27ac data.*

p-values greater than or equal to a threshold of 10. We kept all replicates and ran the analysis again, with the results depicted in Figure 5.10. The PD percentage values are again very low, so the trees once again fit the data well. The improvement looks superficially minor, but we obtained some more biologically meaningful clusters with this approach. For example, in the fibroblast group when we used only top peaks in the overlap representation, one cell type HFF moved to sub-tree containing HFF-Myc (which makes more sense as both are foreskin fibroblast cells). Such a change could be due to particularly noisy data for the HFF cells having obscured the relationship before we removed noisy peaks. Overall, removing noisy peaks further reduced the SR ratio from 0.78 to 0.74 for the overlap representation and from 0.84 to 0.81 for the windowing representation. To test for robustness of the method, we also ran the overlap representation on ENCODE peak data with (negative) log p-values greater than or equal to various thresholds. The results are shown in Table 5.3. The table shows the method works quite well in most of these thresholds.

Another typical noise-reduction procedure, much used in sequence analysis, is to remove regions that appear to carry little information or to produce confounding indications—a procedure known as masking. We devised a very simplified version of masking for our problem, for use only with replicate data, by removing any region within which at most one library gave a different result (1 instead of 0 or vice versa) from the others. In such regions, the presence of absence of peaks is perfectly conserved across all but one replicate. It is possible that replicate data differs from each other because of the noisy nature of the data or because the differences



(a) *(b)*

Figure 5.10 – *Cell-type tree on H3K4me3 data (ENCODE peaks, using all replicates) on peaks with negative log p-value ≥ 10: (a) windowing representation, (b) overlap representation.*

Table 5.3 – **Statistics for cell-type trees on H3K4me3 data using top peaks.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. Rows correspond to various methods (OM: overlap; TP: top peaks). threshold x means that the peaks which have a (negative) log p-value ≥ x is used. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the neighbor-joining (NJ) tree from the Hamming distance between the leaves. ENCODE peak data is used. (all replicates) means all available replicates (1, 2, or 3) for each cell type is used. We can see that the method is robust to various kinds of threshold used.*

	hESC (5)	Epithelial (8)	Fibroblast (16)	Blood (2)	Astrocytes (2)	Myocytes (1)	Endothelial (2)	Skeletal Muscle (1)	SR	PD (%)
OM (all replicates)	5,0	4,2	9,4	2,0	2,0	1,0	1,1	1,0	0.78	3.88
OM (all replicates)-TP (threshold 5)	5,0	5,1	9,4	2,0	2,0	1,0	1,1	1,0	0.76	4.00
OM (all replicates)-TP (threshold 8)	5,0	6,1	9,4	2,0	2,0	1,0	1,1	1,0	0.74	3.77
OM (all replicates)-TP (threshold 10)	5,0	4,3	8,5	2,0	2,0	1,0	1,1	1,0	0.74	3.98
OM (all replicates)-TP (threshold 12)	5,0	4,2	8,5	2,0	2,0	1,0	1,1	1,0	0.73	3.85

are actually present in the cells due to biological reasons. In the latter case, the differences between the two replicates are not cell type specific (as they differ among replicates), hence they are not important for our analysis. After removing such regions, we have somewhat shorter representations, but follow the same procedure. The trees returned have exactly the same topology and so are not shown; the length of edges changed very slightly, as the SR value decreased from 0.74 down to 0.70 using top peaks in the overlap representation.

IDR (irreproducible discovery rate) analysis [73] was carried out with a *R* script downloaded from: https://sites.google.com/site/anshulkundaje/projects/idr. We used data containing exactly 2 replicates on H3K4me3 ENCODE peak data. That is we removed CD14, CD20(1), HFF, HCFaa since they have only one replicate from the earlier used dataset. Therefore we have 34 cell types and 68 libraries (2 replicates per cell type). The IDR analysis was carried out for overlap representation at various IDR thresholds of 0.01, 0.1, 0.25 for the overlapping peaks between the two replicates for each cell type. The results are shown in Table 5.4 and Figure 5.11. As shown in the table, we see a slight improvement of the clustering in epithelial cell types when using an IDR analysis. Since the IDR analysis was done on overlapping peaks, we

Table 5.4 – **Statistics for cell-type trees on H3K4me3 data using IDR analysis.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades; the total number of cells of that type is in the top row. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the NJ tree from the Hamming distance between the leaves. Rows correspond to various methods OM (two replicates): overlap representation on all cell-types which have exactly two replicates (on all available peaks). OM-IDR (two replicates), (threshold x): Overlap representation used on overlapping peaks (between replicates) which have an IDR value ≤ x (output of IDR program). Since these are overlapping peaks the SR ratio is always 0 given the nature of overlap representation. ENCODE peak data is used. (two replicates) means all two replicates for each cell type is used (34 cell-types, 68 libraries). We considered data containing only two replicates for this work.*

	hESC (5)	Epithelial (8)	Fibroblast (14)	Blood (1)	Astrocytes (2)	Myocytes (1)	Endothelial (2)	Skeletal Muscle (1)	SR	PD (%)
OM (two replicates)	5,0	4,3	8,4	1,0	2,0	1,0	1,1	1,0	0.77	3.85
OM-IDR (two replicates), (threshold 0.01)	3,1	4,2	4,4	1,0	2,0	1,0	1,1	1,0	0	4.26
OM-IDR (two replicates), (threshold 0.1)	5,0	6,1	7,4	1,0	1,1	1,0	1,1	1,0	0	3.47
OM-IDR (two replicates), (threshold 0.25)	5,0	6,1	8,4	1,0	1,1	1,0	1,1	1,0	0	3.00

got an SR ratio of 0 between two replicates due to the nature of the overlap representation.

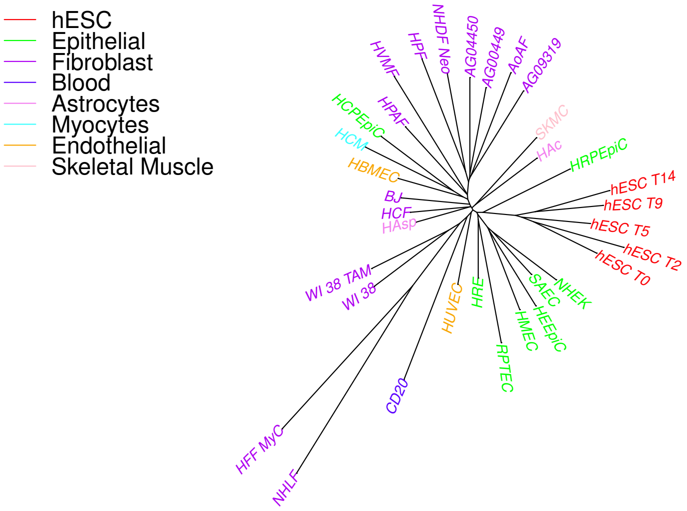


Figure 5.11 – *Overlap representation using overlapping peaks (between replicates) which have an IDR value ≤ 0.025 (output of IDR program). Since the distance between replicates is 0, the two replicates of each cell type are represented in one label.*

A better looking tree

Barring the addition of many replicates, the SR ratio of 0.70 appears difficult to reduce and yet remains high. However, the cherries of replicate pairs by themselves give an indication of the amount of “noise" (variation among individual cells as well as real noise) present in the data.

We can take that noise out directly by replacing each cherry in the tree with its parent, which is a better representative of the population of this particular cell type than either of the two leaves. We carried out this removal on the tree of Figure 5.10(b) and obtained the tree shown in Figure 5.12. Since hESC cells do not form clear pairs, we replaced the entire clade of hESC cells by their last common ancestor. The leaves with remaining long edges are those for which we did not have a replicate (CD14, HCFaa, and HFF).

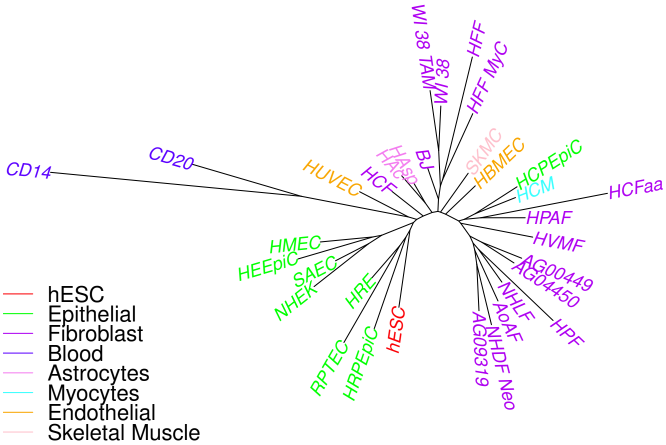


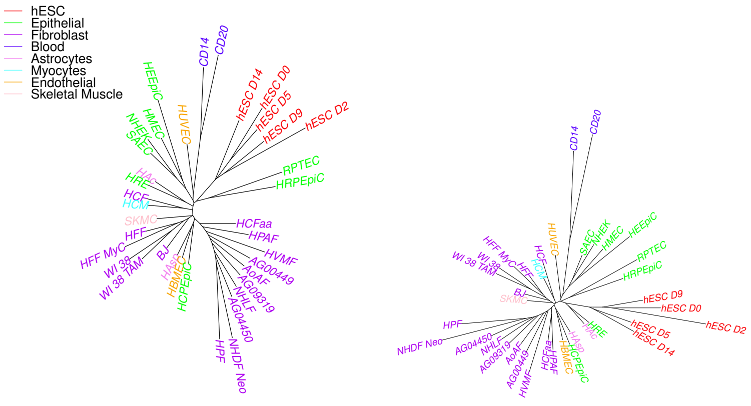
Figure 5.12 – *H3K4me3 data (ENCODE peaks), overlap representation on peaks with negative log p-value ≥ 10. Replicate leaves are removed and replaced by their parent.*

Bootstrap analysis

We did some bootstrap analysis on the cell-type tree obtained using overlap method on single replicates using ENCODE H3K4me3 peak data and neighbor-joining tree approach (Figure 5.3(b)). We found that 62.86% of internal edges have bootstrap scores of above 80%. When we used all replicates on the ENCODE H3K4me3 peak data using overlap method and neighbor-joining tree approach (Figure 5.5(b)), we found that 84.29% of internal edges have bootstrap scores of above 80%. (We used the function boot.phylo from the ape package [95] of *R* software for our analysis. 1000 bootstrap replicates were used for the analysis.)

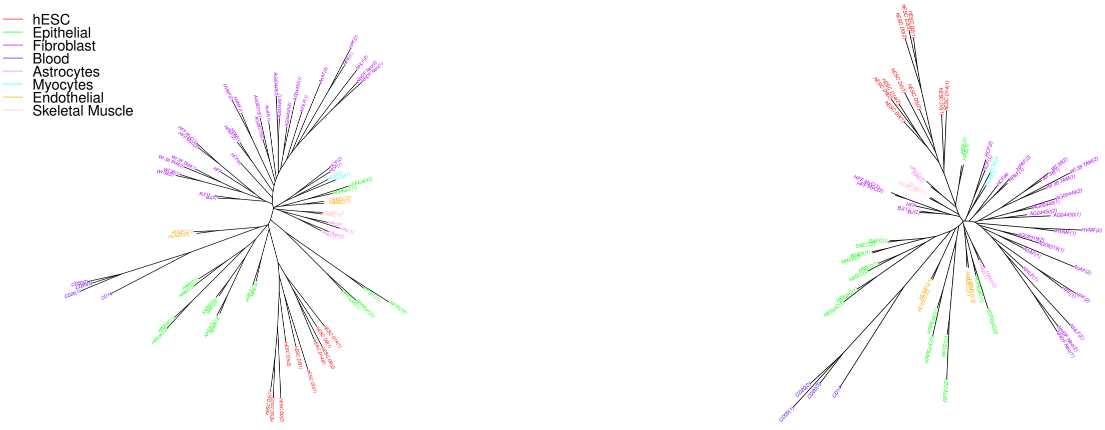
Using ChIPnormSig peaks

To reduce the amount of noise in the ENCODE peaks, we look at the cell-type trees obtained using ChIPnormSig peaks. A false discovery rate (FDR) of 5% was used. The results are shown in Figure 5.13 (using only one replicate for each cell type), Figure 5.14 (using all replicates), and Table 5.2. We see from the figure that the results are considerably better with less noisy edge lengths between replicates (lower *SR* ratio) and also better grouping of cell types. This also indicates the importance of data preprocessing.



(a) (b)

Figure 5.13 – *Cell-type tree on H3K4me3 data (using one replicate) using (a) windowing representation (b) overlap representation. Peaks generated by ChIPnormSig method (FDR 5%).*

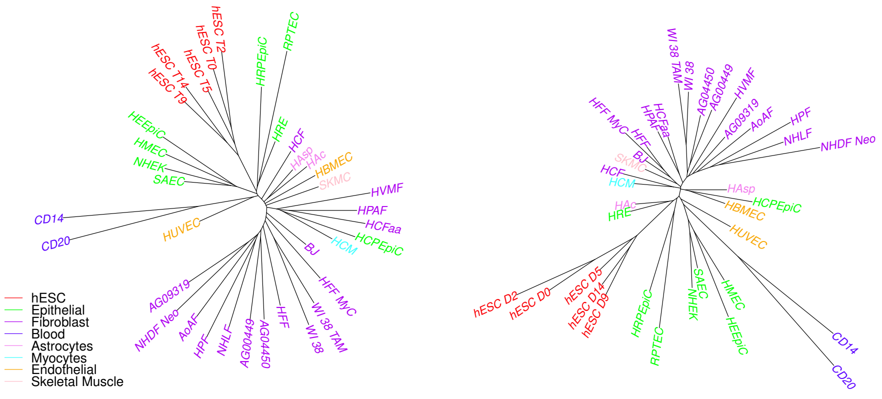


(a) (b)

Figure 5.14 – *Cell-type tree on H3K4me3 data (using all replicates) using (a) windowing representation (b) overlap representation. Peaks generated by ChIPnormSig method (FDR 5%).*

Creating a profile using replicate data

We also show a method of creating profile of a cell type using the data representation of individual replicates. For each cell type, the profile in each bin or interesting region is represented as sum of all 1/0 (data representation value) of each replicate of that cell type in that bin or interesting region divided by the number of replicates. For example, if there are 2 replicates for one cell type, the profile at interesting region i would be 1 if both replicates have 1, 0 if both replicates are 0, 0.5 if one replicate is 1 and the other is 0. Using this new data representation using the profile data representation, we build trees using the neighbor-joining method. The distance between two profile representation (one for each cell type) is sum of all the absolute value of the difference between the profile values at each bin/interesting region. The results are shown in Figure 5.15 and Table 5.2. We see an improvement of results using the profile representation when compared to using all replicates or one replicate data.



(a) (b)

Figure 5.15 – *Cell-type tree on H3K4me3 data (using profile data representation) using Overlap representation: (a) ENCODE peaks, (b) ChIPnormSig peaks.*

Looking at changes along specific branches of the tree

Phylogenetic analysis allows to reconstruct ancestral nodes and thus to study important branches of a tree. We are interested in the changes that happen early in development when ES cells start to differentiate into lineage-specific cell types. To this end, we selected genomic regions which are all 1s in the ES samples and 0s elsewhere allowing for one error in each group. We also selected genomic regions showing the opposite behavior. The results we show are all based on ENCODE peak lists (including replicates) using the overlap representation. We then looked at the enrichment of gene ontology (GO) and other gene annotation terms for genes adjacent to the identified genomic regions, using the GREAT website [83]. This type of analysis was carried for both H3K4me3 data and H3K27me3. The detailed results are shown in appendix B.

We found 322 and 126 regions that were specifically marked by H3K4me3 in ES or non-ES cells, respectively. In the ES positive group, we found significant associations with expression in neural tissues (Figure B.1). This could be explained by the fact that both brain and ES cells have unusually broad expression patterns compared to other tissues. H3K4me3-depleted regions are often flanked by transcription factor genes with zing-finger domains (Figure B.2). The majority of these genes are probably repressed in undifferentiated ES cells.

Carrying out the same type of analysis with H3K27me3 data, we found 4036 regions that were specifically marked in ES cells, but only seven regions showing the opposite histone modification pattern. We find the ES-specific regions to be enriched near genes involved in morphogenesis, consistent with the assumption that such genes have to be repressed in undifferentiated ES cells (Figure B.3). By looking at the numbers of the individual classes, it appears that loss of a histone mark is a more frequent event during development than a gain of a histone mark. The imbalance is stronger for the repressive mark H3K27me3 than for the activating mark H3K4me3.

We explored the distribution of H3K4me3 and H3K27me3 modifications along various pathways of cell differentiation. As before, the analysis was done on ENCODE peak lists (including replicates) using the overlap representation. We considered only regions which do not show ambiguity between replicates. Table 5.5 shows the distribution of histone modification patterns over days 0, 2, 5, 9 and 14 of the ES cell differentiation time course. (Note that the all-zero pattern is not included since the overlap representation requires that a peak be found in at least one sample.) We see from this table that the “all one” pattern (‘11111’) is the most dominant. We also see that patterns with one change over time such as ‘00001’, ‘00011’, ‘00111’, ‘01111’, ‘11110’, ‘11100’, ‘11000’, ‘10000’ are relatively frequent, whereas patterns involving multiple losses or gains such as ‘10101’, ‘01010’, ‘11011’ are rarely found. Patterns with a single gain followed by a loss immediately thereafter (like ‘00100’) are not so rare. However, the opposite class of patterns (like ‘11011’) is very rare. We did gene enrichment analysis on regions showing pattern ‘01000’. While analyzing H3K4me3 data, we found a great diversity of gene annotation terms, with a preponderance of terms related to proliferation and development (Figure B.4). While analyzing H3K27me3 data, many gene annotation terms associated with development were found — like heart development, palate development, nerve development etc. (Figure B.5). The gene annotation terms associated with specific histone modifications appearing on day two are compatible with a sudden response to an external stimulus activating a developmental pathway.

Table 5.6 shows results from a similar kind of analysis along another developmental pathway comprising ES (day 0), HUVEC, and HBMEC. (These three cell types should occur one after the other during development.) Table 5.7 shows results for yet another such developmental pathway consisting of ES (day 0), WI38, AG04550, and HPF. Again we see that the “all one” pattern is quite frequent for H3K4me3 data compared to other patterns. However such was not the case for H3K27me3. The contrasting behavior may be due to the fact that H3K4me3 is often associated with constitutive (house-keeping) genes whereas H3K27me3 primarily regulates developmental genes. From this perspective, it would be unlikely to find invariantly H3K27me3 marked regions along a complete differentiation pathway starting from ES.

Discussion on the evolutionary interpretation of cell-type trees

In this chapter, we have used cell-type trees for studying cell differentiation. We used phylogenetic methods such as neighbor-joining for our work because of the similarities between cell-differentiation process and evolution (as we outlined earlier). Now we discuss how cell-type trees can be used to study the evolution of cell-types among different species.

Arendt [3] outlines the interrelationship between the evolution of cell types and the cell development process, mentioning that, in some cases, cell type development seems to recapitulate cell type evolution. Cell-type trees can be used to study the evolution of cell types. These trees are somewhat similar to phylogenetic trees based on gene duplication-loss models or trees build on morphology based characters. We explain the concept through an example. Figure

Table 5.5 – **Analysis on paths.** *Table shows the number of different types of changes across various days of ES cells. Dx means day x of ES cell type. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column.*

Row no.	D0	D2	D5	D9	D14	H3K4me3 (total)	H3K27me3 (total)
1	0	0	0	0	1	1075	3496
2	0	0	0	1	0	342	743
3	0	0	0	1	1	387	599
4	0	0	1	0	0	331	1459
5	0	0	1	0	1	15	97
6	0	0	1	1	0	40	112
7	0	0	1	1	1	247	461
8	0	1	0	0	0	1278	1919
9	0	1	0	0	1	14	22
10	0	1	0	1	0	9	57
11	0	1	0	1	1	30	82
12	0	1	1	0	0	60	74
13	0	1	1	0	1	5	11
14	0	1	1	1	0	9	34
15	0	1	1	1	1	147	253
16	1	0	0	0	0	450	641
17	1	0	0	0	1	11	9
18	1	0	0	1	0	11	5
19	1	0	0	1	1	14	3
20	1	0	1	0	0	24	40
21	1	0	1	0	1	6	5
22	1	0	1	1	0	10	5
23	1	0	1	1	1	101	81
24	1	1	0	0	0	630	1140
25	1	1	0	0	1	17	21
26	1	1	0	1	0	11	26
27	1	1	0	1	1	47	71
28	1	1	1	0	0	309	548
29	1	1	1	0	1	54	52
30	1	1	1	1	0	263	335
31	1	1	1	1	1	25112	10926

5.16 shows an constructed example of a particular current species $S2$ (bottom cell-type tree $T2$) to a particular ancestral species $S1$ (top cell-type tree $T1$). The leaf nodes in tree $T2$: $C1-1$, $C1-2$, $C2$, $C3$, $C4$ represent blood cells of the current species $S2$. The leaf nodes of tree $T1$: $C1$, $C2$, $C3$ represent blood cells of ancestral species $S1$. The internal nodes of each tree represent cell types of some ancestral species. We can see that leaf node $C1$ of tree $T1$ is the parent of leaf nodes $C1-1$ and $C1-2$ of tree $T2$. Similarly some other nodes are from one tree to another are marked by red arrows. The leaf nodes of each tree represent various blood cell types present in that species. The figure shows how the ancestral nodes in $S2$ could be leaf nodes in ancestor $S1$. Other possibilities are also shown. One possibility is that current species have more blood cell types than an ancestral species and this is captured by a cell-type tree. Thus the cell-type trees we generate using histone modification data could also be used to study the evolution of cell types.

Table 5.6 – **Analysis on paths.** *Table shows the number of different types of changes for ES, HUVEC, and HBMEC cell types. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column. NA - not applicable (because data for the cell-type is not available).*

H3K4me3 (row no.)	ES (D0)	HUVEC	HBMEC	Total
1	0	0	1	3407
2	0	1	0	1769
3	0	1	1	1805
4	1	0	0	5224
5	1	0	1	1415
6	1	1	0	417
7	1	1	1	23824
H3K27me3 (row no.)	ES (D0)	HUVEC	HBMEC	Total
1	0	1	NA	12468
2	1	0	NA	14684
3	1	1	NA	8403

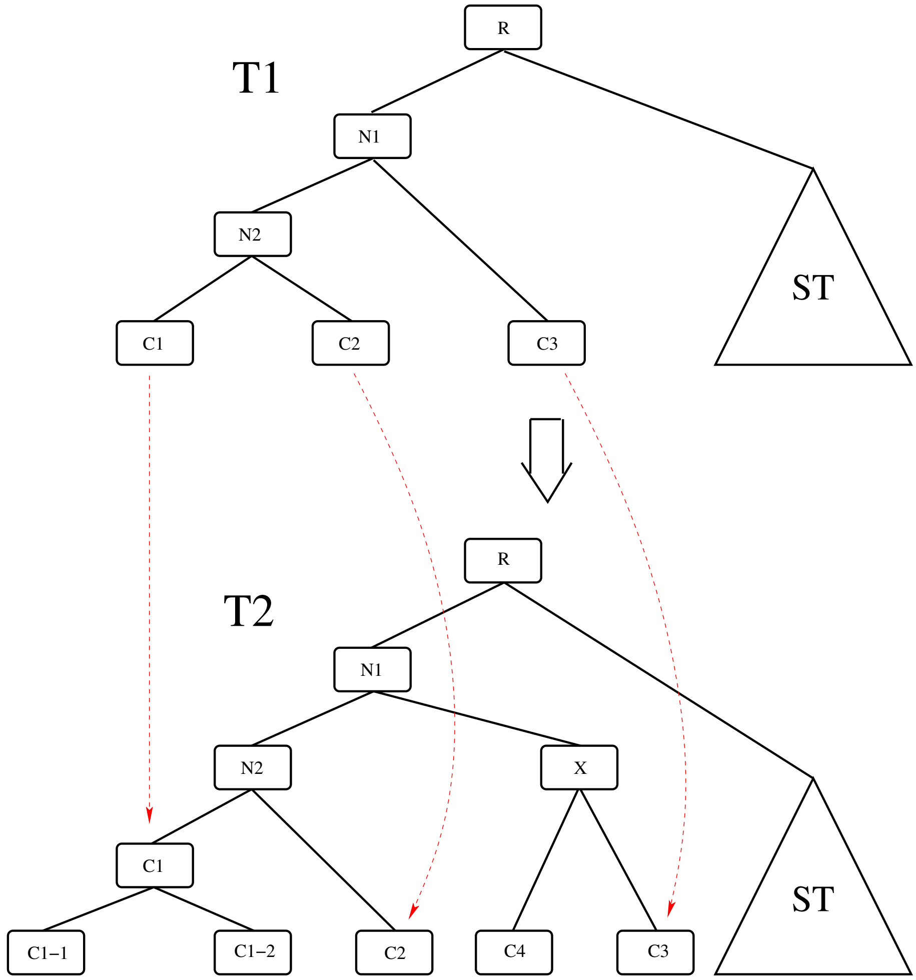


Figure 5.16 – *Cell-type trees to study evolution of cell types - a constructed example is shown in this figure. Tree $T2$ - current species $S2$. Tree $T1$ - ancestral species $S1$. ST - sub-tree.*

Table 5.7 – **Analysis on paths.** *Table shows the number of different types of changes for ES, WI38, AG04550, HPF cell types. 1 and 0 represents the presence or absence of a peak as defined by the overlap representation in one region of the genome. The number of such 1-0 patterns are counted and presented in the last column. NA - not applicable (because data for the cell-type is not available).*

H3K4me3 (row no.)	ES (D0)	WI38	AG04550	HPF	Total
1	0	0	0	1	2946
2	0	0	1	0	1050
3	0	0	1	1	1106
4	0	1	0	0	1670
5	0	1	0	1	382
6	0	1	1	0	879
7	0	1	1	1	4644
8	1	0	0	0	5465
9	1	0	0	1	354
10	1	0	1	0	353
11	1	0	1	1	989
12	1	1	0	0	62
13	1	1	0	1	35
14	1	1	1	0	506
15	1	1	1	1	21806
H3K27me3 (row no.)	ES (D0)	WI38	AG04550	HPF	Total
1	0	NA	1	NA	14734
2	1	NA	0	NA	20108
3	1	NA	1	NA	6342

5.3.1 Code

The code for finding cell-type trees (in *R* code) and also ChIPnormSig significant regions (peaks) (in MATLAB code) is made available in http://lcbb.epfl.ch/software.html.

5.4 Conclusions

We studied the novel problem of inferring cell-type trees from histone modification data. We defined methods for representing the peaks as 0/1 vectors and used these vectors to infer trees. We obtained meaningful trees, conforming closely to expectations and biologically plausible, in spite of the high level of noise in the data and the very limited number of samples per cell type. Our results confirm that histone modification data contain much information about the history of cell differentiation. We carried out a number of experiments to understand the source of the noise, using replicate data where available, but also devising various noise filters. Our results show that larger replicate populations are needed to infer ancestral nodes, an important step in understanding the process of differentiation. We also discussed how cell-type trees can be used to study the evolution of cell types.

Much work remains to be done on methods for building good cell-type trees. In particular, the noisy nature of the data remains an issue. We are exploring various other data preprocessing and representation techniques which can be used for this purpose. Refining the model of gain or loss of marks may enable the use of maximum likelihood methods, which deal better with large ranges of pairwise dissimilarities and also yield more accurate inferences for internal nodes.

Since many histone marks appear independent of cell differentiation, identifying which marks are most strongly correlated with the differentiation process is of significant interest. Once such marks have been identified, reconstructing their state in ancestral nodes will enable us to identify which regions of the genome play an active role in which steps of cell differentiation.

6 Further study using cell-type trees

6.1 Introduction

This chapter is continuation of the work done on cell-type trees that we have discussed in chapter 5. Here we discuss the work in progress that we have been doing. Therefore in this chapter we just show some preliminary work and results.

6.2 Lifting: inferring ancestral nodes

In chapter 5 we discussed cell-type trees. However we did not discuss how ancestral nodes could be inferred from the given data. One possible way to do that is through using a cell-type tree and then lifting leaf nodes to internal nodes in such a way as it reflects the cell-differentiation process. This lifting procedure is important because in the cell-differentiation process, more specialized cell types arise from less specialized cell types. Therefore for example ES (at day 0) could be the root node given set of cell-types.

To do the lifting procedure - we relook at some of the questions we had mentioned in chapter 5.
1. For a given internal node, is there a natural lifting from a leaf?
2. If there is no suitable lifting, is the node nevertheless a natural ancestor—i.e., does it correspond to a valid (real) cell type?
3. If the node has no suitable lifting and does correspond to a valid cell type, can we infer its data representation?

We do not currently address all the above questions but simply give a method for lifting and see how it works.

6.2.1 Algorithm for lifting

Algorithm 1 Algorithm for lifting

```
INPUT: a maximum parsimony cell-type tree $T$
OUTPUT: a new cell-type tree that internal nodes are labeled by leaf nodes
INITIAL: Set $Lift(i) = i$ for all the nodes in the input tree $T$, where $i$ is the node name.
Root this tree by some method.
Infer the ancestral states of the internal nodes and parsimony score for the tree using Fitch's algorithm [42] on $T$.
while when there are more than two nodes left in the tree do
    Compute the hamming distance from each leaf $i$ to their respective ancestor node $Ancestor(i)$.
    Choose the edge which has the minimum length between leaf $i$ and $Ancestor(i)$. The leaf $i$ is considered only if the sibling of leaf $i$ is also a leaf and not an internal node.
    Lift leaf $i$ to its ancestor $Ancestor(i)$ by setting $Lift(Ancestor(i)) = Lift(i)$.
    Set $Ancestor(i)$ to be a new leaf (or mask all other nodes in the subtree rooted at $Ancestor(i)$) and derive a new tree $T^{'}$.
    Set $T = T^{'}$.
    Recompute the ancestral states of the internal nodes using Fitch's algorithm [42] on $T$.
end while
EXIT: for each internal node $i$ in the input tree, $Lift(i)$ represents its new label after lifting.
```

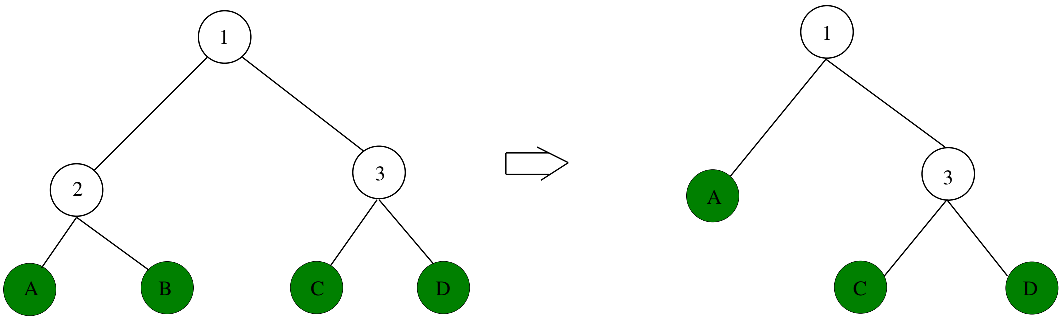


Figure 6.1 – ***Part of lifting procedure.*** *An example showing a small portion of the lifting algorithm is shown. The edge between node A and node 2 is the smallest among the considered edges. Therefore node A is lifted upwards to its ancestor and it is made the new leaf.*

A small example showing a portion of the lifting procedure is shown in Figure 6.1.

6.2.2 Results and Discussion

We select a small set of cell types for our analysis using H3K4me3 peak data from ENCODE project (University of Washington group) [82]. The cell types used are CD14, CD20, HUVEC, HBMEC, WI 38, WI 38 TAM, HPF, AG04550, ES day 0, 2, 5, 9, 14. Only one replicate of each cell type was used. ES at day 0 is marked as hESC T0 or hESC D0 in the figures. We chose these cell types because most of them were present in Tables 5.5, 5.6, 5.7. These tables showed a possible time sequence in how some cell-types occurred. (We had discussed this in some detail in chapter 5.)

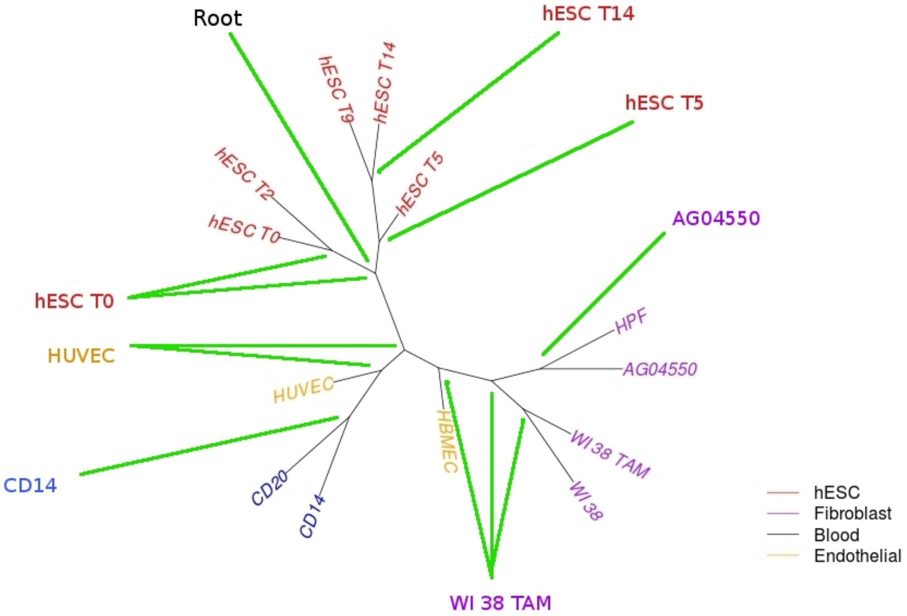


Figure 6.2 – ***Lifting procedure for ancestral node reconstruction***. *Shows cell-type tree for H3K4me3 data using overlap representation. The internal nodes are labelled using green edges to one of the cell-types in the leaf node (by lifting procedure).*

Figures 6.2 and 6.3 shows the cell-type tree (using overlap representation) along with the lifted cell-types at the internal nodes. The initial tree was found using maximum parsimony analysis using TNT software [47] on the data representation given by overlap representation. The tree obtained was then rooted at a fixed root node which was the ancestor node of all the ES cell types for various days (shown in figure) before lifting. The above mentioned lifting procedure was applied. We can get Figures 6.2 from this procedure. The internal nodes are labelled using green edges. Each internal node is labelled from one of the leaf nodes which was lifted up. We get Figure 6.3 from by Figure 6.2 by collapsing nodes which have the same label (nodes with the same label occur next to each other). The lifting shows that the tree more or less agrees with the biological pathways discussed in chapter 5 and Tables 5.5, 5.6, 5.7. We find from Figures 6.2 and 6.3 that ES at day 0 got lifted to the root (which is the ancestor of all the ES cell types for different days), which is as expected. From the figure we also find that ES day 2 occurs after ES day 0. Similarly ES cells at day 5 got lifted and is closer to the root before day 9 and day 14. However ES at day 14 is shown as closer to the root than ES at day 9, which is contrary to expectation. From the figure we see that HUVEC is closer to the root node (ES day 0) than HBMEC as shown in Table 5.6. We also see that WI 38 (or WI 38 TAM) is closer to root node (ES day 0) than AG04550, while AG04550 is an ancestor to HPF as shown in Table 5.7. This seems to suggest the lifting procedure works quite well for a small data set.

We also compare our results with a minimum spanning tree (MST) [97, 65]. The MST approach

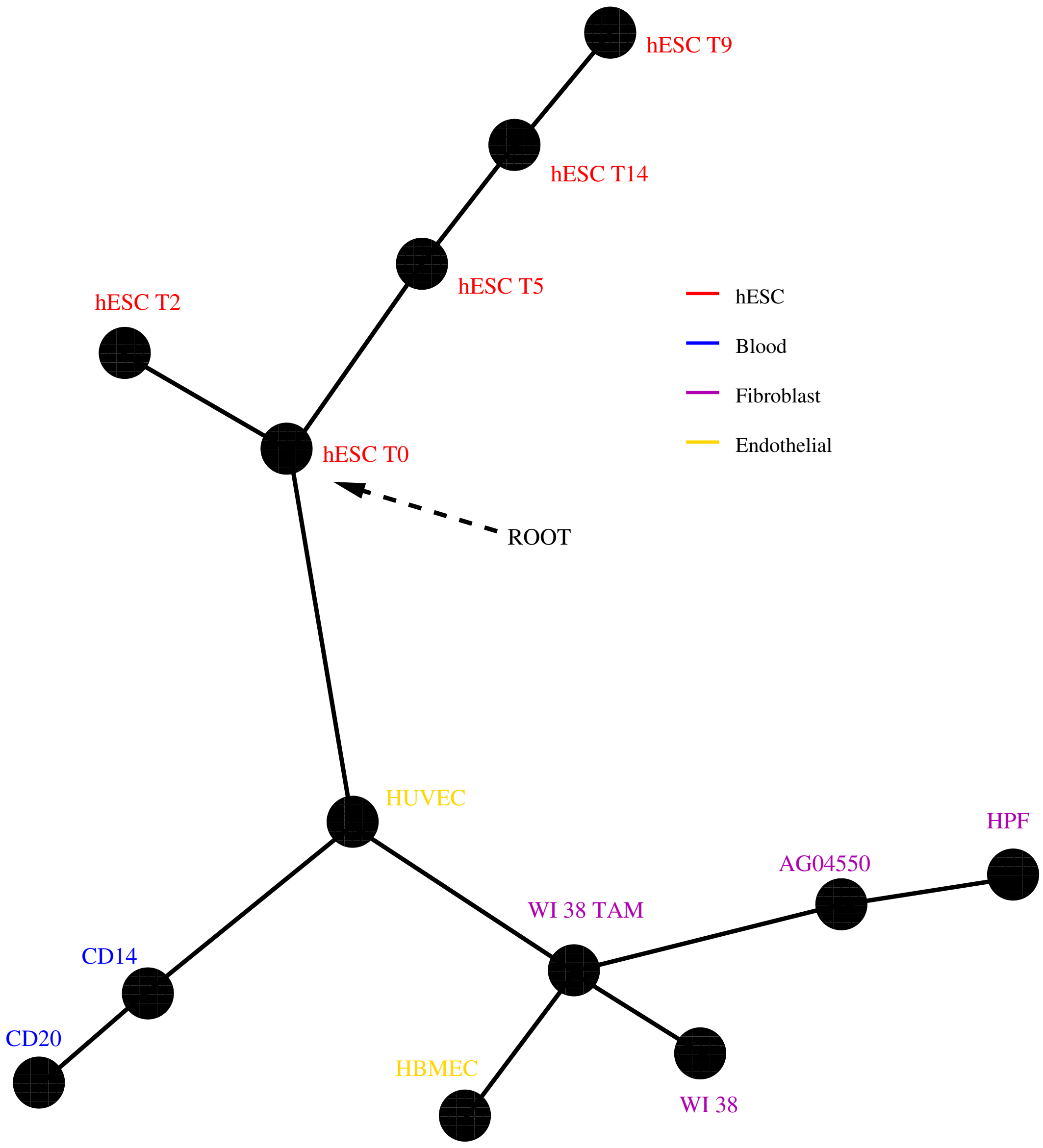


Figure 6.3 – ***Lifting procedure for ancestral node reconstruction***. *Shows cell-type tree for H3K4me3 data using overlap representation after lifting process. (Figure is not drawn to scale.)*

was used on the data obtained by finding hamming distances on the data obtained from overlap representation. The results using a minimum spanning tree are shown in Figure 6.4. We see that even using MST approach we get somewhat similar results while compared to cell-type trees on this dataset. However we get one additional error - because WI 38, WI 38 TAM and AG04550, HPF (all fibroblasts) seem to be separated by HBMEC (endothelial cell) - contrary to expectation. We don't see this error while using the lifting procedure.

More experiments need to be done on various real and simulated datasets for us to see how good the lifting techniques are for ancestral inference. Also given a tree we need to stop the lifting procedure at some point. A statistical measure about when to stop lifting is also required.

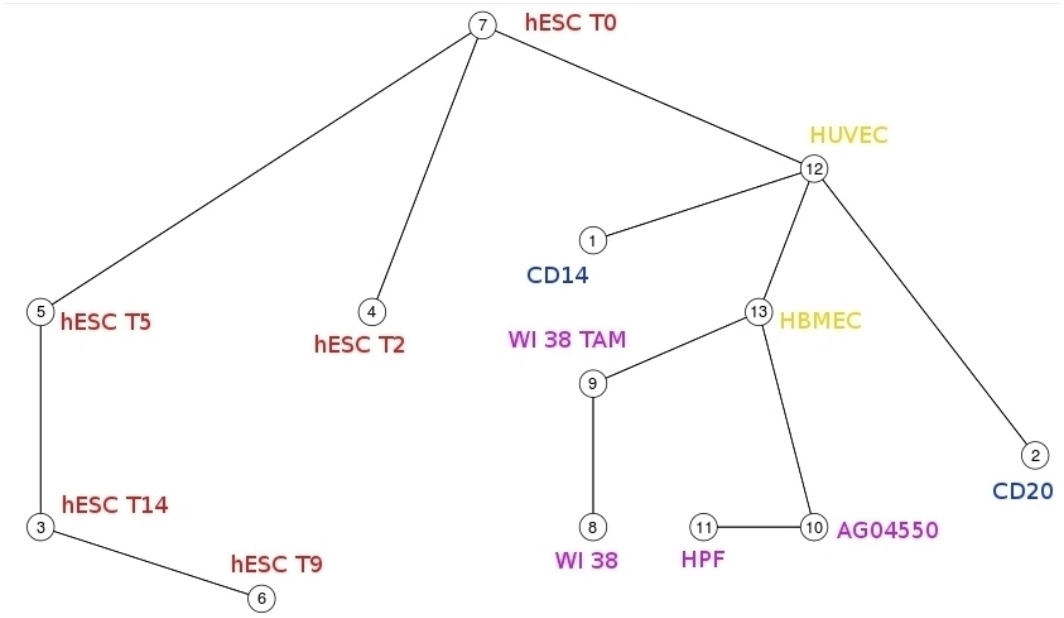


Figure 6.4 – ***Minimum spanning tree***. *Minimum spanning tree for H3K4me3 data using overlap representation (the edges are not drawn to scale).*

One possibility to study this problem is to combine lifting procedures with minimum spanning tree methods and build a new algorithm based on that.

6.3 Using normalized raw data

As we have seen in chapter 5, cell-type trees use histone modification peak data for data representation purposes. Since the peak-finding step from the raw data could lead to a loss of information and/or be itself noisy, we look at building cell type trees by using raw or normalized ChIP-Seq data for data representation purposes.

6.3.1 Evaluation of different data representations

As done in chapter 3, we get the data has by dividing the genome into bins and collecting, for each bin, a count of the mapped sequence tags that fall within the bin. The result is a “library", which is simply a list of positive integers, each successive integer associated with the next bin. Since ChIP-Seq histone modification fragments are approximately 200 bp, like what was done in Xu *et al.* 2008 [125], we approximate the center of each fragment by shifting the tag end position by 100 bp downstream or upstream, according to its orientation on the chromosome. Then we try the following ratio:

$$R = \frac{\frac{1}{2}\ \left[\sum_i |A1(i) - A2(i)| + \sum_j |B1(j) - B2(j)|\right]}{\frac{1}{4}\ \left[\sum_a |A1(a) - B1(a)| + \sum_b |A1(b) - B2(b)| + \sum_c |A2(c) - B1(c)| + \sum_d |A2(d) - B2(d)|\right]}$$

Table 6.1 – R **ratio.** *The R ratios for different kinds of data representation are shown for 37 cell-types of H3K4me3 data, each cell-type compared with every other. The frequency of occurrence with a small range of R ratio is shown for all the data representations.*

	Peak data	Raw data	Mean normalization	Mean normalization per chromosome	Standard score normalization
>1	0	0	0	0	1
[0.9 1)	3	9	4	4	1
[0.8 0.9)	6	46	5	7	4
[0.7 0.8)	80	138	31	41	3
[0.6 0.7)	229	195	117	148	7
[0.5 0.6)	221	136	184	213	31
[0.4 0.5)	22	36	155	114	52
[0.3 0.4)	0	1	61	33	118
[0.2 0.3)	0	0	4	1	174
[0.1 0.2)	0	0	0	0	170

where $A1(i)$ and $A2(i)$ are bincount values for replicates 1 (library $A1$) and 2 (library $A2$) respectively of cell type A at bin i; and $B1(i)$ and $B2(i)$ are bincount values for replicates 1 (library $B1$) and 2 (library $B2$) respectively of cell type B at bin i. $|x|$ means the absolute value of x. We can also calculate the value for R for peak data too - in such a case value of $A1(i)$, $A2(i)$, $B1(i)$ or $B2(i)$ will be either 1 (peak) or 0 (no peak) for each bin i (data representation as in window representation in chapter 5).

A smaller value of R (much below 1) shows that the distance between replicates of same cell type is much smaller than the distance between different cell types. We find the ratio of R for various types of input - whether we use peak data or different forms of normalized raw data.

We tried the following approaches for our data representation. Let $L(i)$ be the value at bin i.
1. Peak data (as done before): $L(i)$ is 1 (peak) or 0 (no peak).
2. Raw data without any normalization: consider bincount values in each bin. $L(i) = BC(i)$ where $BC(i)$ is bincount at bin i.
3. Mean normalization of raw data: $L(i) = \frac{BC(i)}{\mu_{BC}}$, where μ_{BC} is the mean bincount value of all bins in the genome.
4. Mean normalization of raw data (for each chromosome): $L(i) = \frac{BC(i)}{\mu^C_{BC}}$, where μ^C_{BC} is the mean bincount value of all bins in chromosome C (to which BC belongs).
5. Standard score normalization of raw data (for each chromosome): $L(i) = \frac{BC(i)-\mu^C_{BC}}{\sigma^C_{BC}}$, where σ^C_{BC} is the standard deviation of the bincount values of all the bins in the genome.

We first test the R ratio for each of these five data representations for 37 cell-types of H3K4me3 data, each cell-type compared with every other. The frequency of occurrence with a small range of R ratio is shown for all the data representations in Table 6.1. We see from the table that standard score normalization works best.

Now for each of the above five mentioned data representations we using windowing representation to build cell-type trees. The pairwise distance measure used (instead of the simple Hamming distance used earlier) between the two data representations of the two libraries is the sum of the absolute values of the differences between two data representations. Neighbor-joining method was the tree building technique used. We have shown the results in Table 6.2

Table 6.2 – **Statistics for cell-type trees on H3K4me3 data for different data representation.** *2nd to 9th columns show the number of cells (of the same type) belonging to the largest and second-largest clades (fraction values indicate only one of the two replicates fall in that clade); the total number of cells of that type is in the top row. Fractions are used when only one of the two replicates fall into the clades. Rows correspond to various data representation techniques. Windowing representation is used for all. The second last column shows the SR ratio. The last column contains the percent deviation (PD) of the distances between the leaves found using the neighbor-joining tree from the Hamming distance between the leaves. Only peaks from the ENCODE data are used.*

	hESC (5)	Epithelial (8)	Fibroblast (16)	Blood (2)	Astrocytes (2)	Myocytes (1)	Endothelial (2)	Skeletal Muscle (1)	SR	PD (%)
Peak data	5,0	6,1	11,2	2,0	1,1	1,0	1,1	1,0	0.84	3.30
Raw data	5,0	2,2	8,2	2,0	1,1	1,0	1,0.5	1,0	0.86	6.03
Mean normalization	5,0	4,2	8,4	2,0	1,1	1,0	1,1	1,0	0.77	6.43
Mean normalization per chromosome	5,0	4,2	10,4	2,0	1,1	1,0	1,1	1,0	0.79	6.17
Standard score normalization per chromosome	5,0	4,2	8,5	2,0	1,1	1,0	1,1	1,0	0.54	8.70

and Figure 6.5. We see that some of the normalization techniques work better than using raw data, however they did not so far better than using peak data. Probably more sophisticated normalization techniques need to be used to get better trees.

6.4 Comparing human and mouse cell types

In this section, we study the phylogenetic relationships on the evolution of cell types across species. In the previous chapter, we had discussed about how the evolution of cell types can be studied using cell-type trees. In this chapter, we look at cell types between human and mouse.

Orthologous regions between two species are those regions which are inferred to be descended from the same ancestral sequence separated by a speciation event. For all practical purposes, all pairwise aligned regions are considered orthologous regions. The orthologous regions (pairwise aligned regions) between human and mouse genomes were selected. Mouse genome was aligned to the human genome. The data was taken from UCSC Genome Browser [60] and the link for this data is http://hgdownload.soe.ucsc.edu/goldenPath/hg19/vsMm9/. We look at the peak regions of H3K4me3 ChIP-Seq data only on these orthologous regions. The peaks which fell within the orthologous regions in mouse where mapped to corresponding positions in the human genome. We used both overlap representation with neighbor-joining approach to build cell-type trees. We used some cell types from the ENCODE project (UW data) for the human data. For the mouse data (mm9), we took it from the LICR ENCODE project [21]. Cell type mE14 is mouse ES cell type (day 0), strain 129/Ola, and cell type mBMDM is mouse bone marrow derived macrophage data.

Figure 6.6 with shows the results considering overlap representation for peaks in the orthologous regions of human and mouse. We had expected the mBMDM to be close to blood cells of human, and mE14 close to human ES cells in the phylogenetic tree. However we find that the mouse cell types seem to cluster together. We see that the number of mouse peaks in the data far exceeds the number of human peaks. Therefore to reduce the bias due the varying number of peaks, we try to reduce the variations in the following ways. (a) Delete peaks in mouse data whose negative log p-value is smaller than 50. (b) Divide the genome into windows of 1 Mbp.

Then remove those windows if the length of the peak region in either human or mouse data was 50% more than the other (comparing hESC T0 and mE14). The results are shown in Figure 6.7. However we still find that mouse data cells cluster together. This could be because of various reasons. One could be biological, i.e. maybe H3K4me3 is not that strongly associated with cell differentiation as some other factors and may not be the best histone modification to use. So we would need to explore other histone modification data. We may also need to look at only some regions of the genome which are tissue specific (and not all orthologous regions). The other reason could be due to noise in the data. We get the mouse and human data from different labs and there could be lab specific bias in the data. The number of peaks for mouse is very different from that of human, so even that could be an issue. So preprocessing of data maybe very important. We are currently exploring many of these points.

6.5 Conclusion

In this chapter, we outlined some details of three pieces of ongoing work. First we provide an algorithm for lifting for reconstructing ancestral nodes in a cell-type tree. Secondly, we explore the use of normalized raw data instead of peak data for building cell-type trees. Thirdly, we study the evolution of cell types in human and mouse using histone modifications. In each of these problems we outline some of the issues we faced and the challenges ahead.

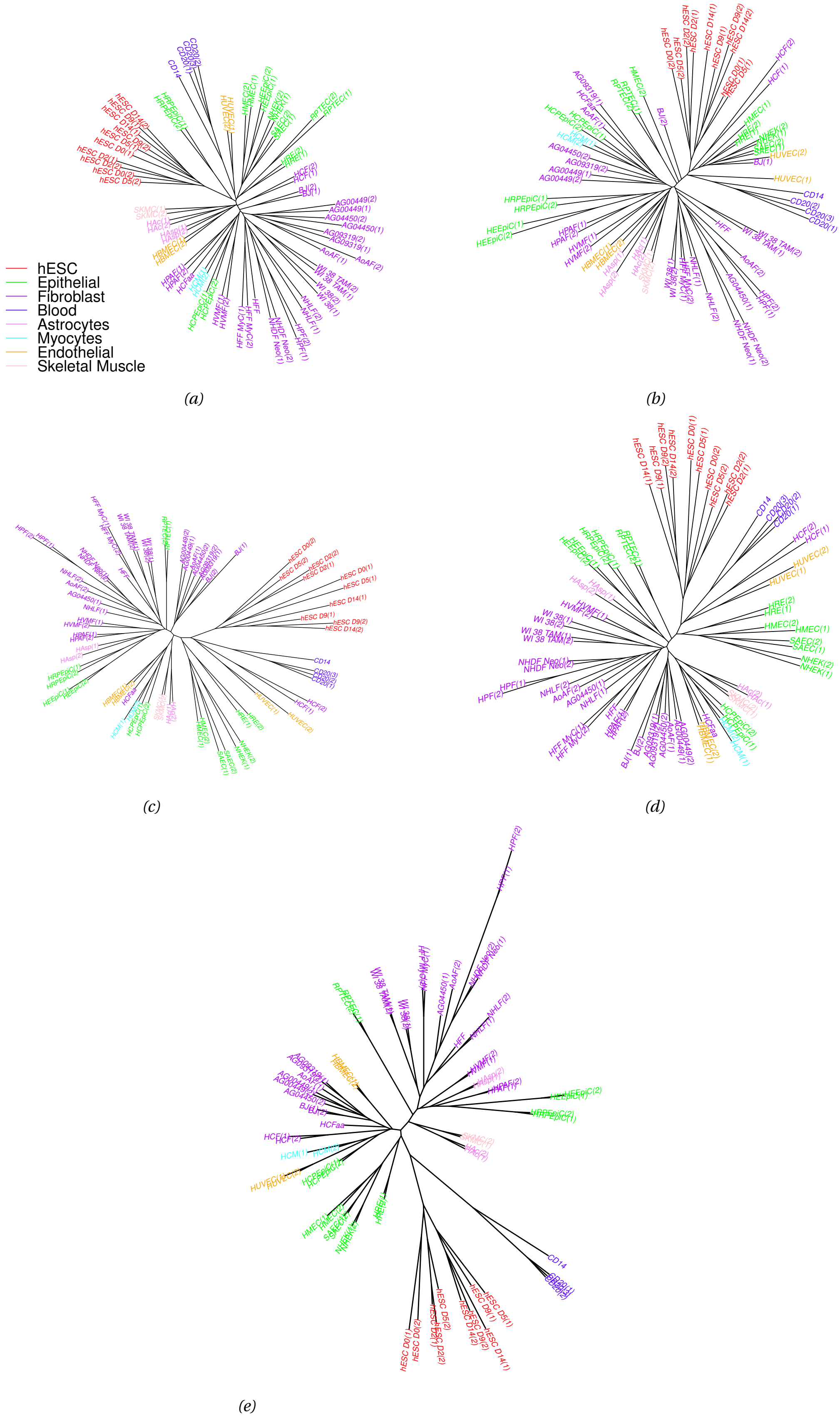


Figure 6.5 – ***Cell-type tree using windowing representation.*** *Data representation using: (a) Peak data, (b) Raw data, (c) Mean normalization of raw data, (d) Mean normalization of raw data (for each chromosome), (e) Standard score normalization of raw data (for each chromosome).*

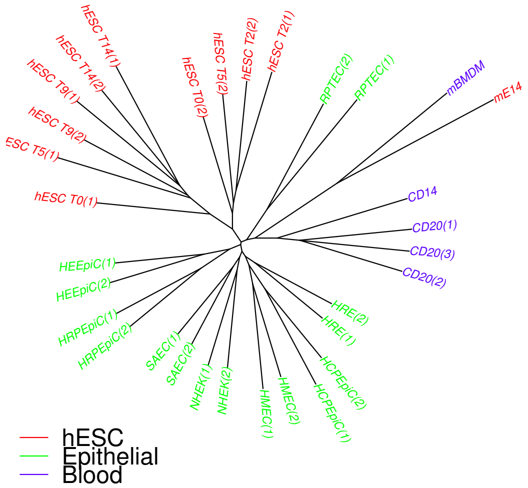
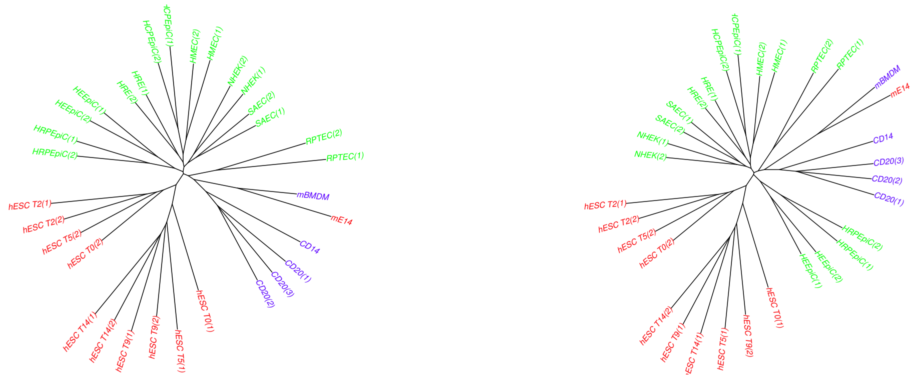


Figure 6.6 – ***Human and mouse cell types****. Cell-type tree using overlap representation on human and mouse cell types. Peak data considered only on orthologous regions of human and mouse.*



(a)

(b)

Figure 6.7 – ***Human and mouse cell types by removing some peaks****. The number of peaks available for mouse where much more than for human. Therefore the number of peaks used in mouse data were reduced so as to reduce the bias in the results. (Peak data considered only on orthologous regions of human and mouse.) (a) Deleting peaks in mouse data whose negative log p-value is smaller than 50. (b) Divide the genome into windows of 1 Mbp. Then remove those windows if the length of the peak region in either human or mouse data was 50% more than the other (comparing hESC T0 and mE14).*

7 Conclusions

In this thesis, we addressed problems in computationally analyzing and studying problems in histone modification ChIP-Seq data.

We first proposed a two-stage statistical technique called ChIPnorm to normalize ChIP-Seq data, and to find differential regions in the genome, given two libraries of histone modifications of different cell types. We showed that the ChIPnorm program removes most of the bias in the data and provides a normalization that enables direct comparison of values between the two cell types. We showed that our approach improved upon the state of the art, and we were able to highlight some aspects of H3K27me3 modifications in mouse embryonic stem cells and neural progenitor cells, including a so far unnoticed transition of bivalent mark of H3K4me3 and H3K27me3 in embryonic stem cell to a H3K27me3-only state in differentiated cells, possibly reflecting permanent repression of developmental genes.

The next thing that we proposed was a probabilistic partitioning method to find significant patterns in ChIP-Seq data. The probabilistic partitioning method works on the principles of expectation and maximization. The method is simple, very flexible and implementable in a few lines of *R* code and is capable of running large datasets in a few minutes. We showed that the when there is low data coverage, the probabilistic partitioning method gives excellent model accuracy, superior to some of the existing techniques. We showed that other than pattern discover and classification, the probabilistic partitioning approach could also be used in partitioning of nucleosome positioning patterns in human promoters, and shape-based evaluation and re-focusing of ChIP-Seq peaks from published peak lists.

In the third part of this thesis, we addressed the novel problem of inferring cell-type trees from histone modification data. We defined new data representation techniques, a new peak finder, and phylogenetic methods to infer these trees. We obtained biologically meaningful trees in spite of the high level of noise in the data and the very limited number of samples per cell type — thus confirming that histone modification data contain much information about the history of cell differentiation. We also discussed how cell-type trees can be used to study the evolution of cell types. Thus we tried to combine the fields of evolution and development.

In the final part of the thesis, we present some on-going work trying to build on the work done on cell-type trees. We suggest an algorithm for lifting for reconstructing ancestral nodes in a cell-type tree. We explore the use of normalized raw data instead of peak data for building cell-type trees. We also study the evolution of cell types in human and mouse using histone modifications. We outline some of the issues we faced and the challenges ahead, in each of these problems.

A Appendix for Chapter 4

A.1 *R* code for probabilistic partitioning

This section present prototype *R* code for the various flavors of probabilistic partitioning presented and exemplified in this paper. The functions performing a single EM iteration, and the envelope programs implement different seeding strategies, are given in separate tables. The code for the EM functions can be pasted and executed in an R command line without modification. The code for the envelope programs may require adjustments. For instance, the user may want to change the number classes to be discovered, or the number of EM iterations to be carried out.

A.1.1 Expectation-Maximization step for basic ChIP-partitioning algorithm

```
1  em_basic = function(c,q,data) {
2    K=dim(c)[1]; N=dim(data)[1]
3    l=matrix(nrow=N, ncol=K); p=matrix(nrow=N, ncol=K)
4    for(i in 1:N) { for (j in 1:K) {
5       l[i,j] = sum(dpois(data[i,], c[j,], log=T)) }}
6    for(i in 1:N) {
7       p[i,] = q*exp(l[i,]-max(l[i,])); p[i,] = p[i,]/sum(p[i,])}
8    q = colMeans(p)
9    c = (t(p) %*% data)/colSums(p)
10
11   c <<- c; q <<- q; p <<- p;
12 }
```

Notes:

Line 1: The arguments of the function have the following structure and contents:

c: a matrix containing the classes to be optimized. $c[i,j]$ is the expected bincount value of class i at position j.

q: a vector defining the prior probabilities of each class.

data: a matrix containing the samples. data$[i,j]$ is the bincount of sample i at position j.

Line 9: The expected bincounts for each classes are computed at once by means of a matrix multiplication.
Line 11: *c*, *q*, *p* are exported for use outside this function. The output variable *p* is a matrix containing the samples' posterior probabilities of belonging to particular classes (rows correspond to samples, columns to classes).

A.1.2 Complete algorithm for partitioning with random seeds

```
1  K=2; N=dim(data)[1]
2  p=matrix(nrow=N, ncol=K)
3  for(i in 1:K) {p[,i] = rbeta(N,N**-0.5,1)}
4  c = (t(p) %*% data)/colSums(p)
5  q=rep(1/K,K)
6
7  for(i in 1:20) {em_basic(c,q,data)}
```

Notes:
Line 1: *K* is the number of classes to be found.
Line 3: Generation of random seeds. Samples are randomly assigned probabilities (weights) for each class. (The choice of a Beta function is not crucial). On the next line, these probabilities will be used to generate expected bincount vectors for each class. The probabilistic class assignment makes sure that classes will be free of zero values. (Initial zero values are undesirable as they cannot be changed during EM.).
Line 5: *q* is a vector containing the prior probabilities for each class.
Line 7: For simplicity, the code proposes a fixed number of EM iterations. Alternatively, the loop could be terminated after some convergence criterion is reached.

A.1.3 Iterative partitioning - standard version for two classes

```
1  K=2; N=dim(data)[1]; L=dim(data)[2]
2  c    = matrix(data=colMeans(data), nrow=1, ncol=L)
3  flat = matrix(data=    mean(data), nrow=1, ncol=L)
4  q = 1
5
6  for (m in 1:(K-1)) {
7     c = rbind(flat,c)
8     q = c(1/m,q); q = q/sum(q)
9     for(i in 1:20) {em_basic(c,q,data)}
10    }
```

Notes:
Line 1: *K* is the number of classes to be found.
Line 2: The process is initialized with a single class corresponding to the average bincount profile from all samples as visualized by an aggregation plot.
Line 3: Defines a "flat" class, which serves to absorb atypical samples not corresponding to

any of the already defined classes during iterative partitioning.
Line 6: Outer loop: m is the current number of classes.
Line 7: The flat class is added to the current set of classes. (The number of classes is now m+1.)
Line 8: Redefines the probabilities of the current classes (variable q). The new class enters the EM algorithm with a prior probability of $1/(m+1)$. The probabilities of the already existing classes are diminished proportionally.

A.1.4 Iterative EM partitioning with untrained flat class

```
 1  K=2; N=dim(data)[1]; L=dim(data)[2]
 2  c    = matrix(data=colMeans(data), nrow=1, ncol=L)
 3  flat = matrix(data=    mean(data), nrow=1, ncol=L)
 4  q = 1
 5
 6  for (m in 1:(K-1)) {
 7     c = rbind(flat,c)
 8     q = c(1/m,q); q = q/sum(q)
 9     for(i in 1:20) {c[1,]=flat; em_basic(c,q,data)}
10     }
```

Notes:
Differences with respect to the standard version shown before are marked in red.

A.1.5 Variations of the EM algorithm - shape-based partitioning

```
 1  em_shape = function(c,q,data) {
 2     K=dim(c)[1]; N=dim(data)[1]
 3     l=matrix(nrow=N, ncol=K); p=matrix(nrow=N, ncol=K)
 4     for(i in 1:K) {c[i,]=c[i,]/mean(c[i,])}
 5     rm=rowMeans(data)
 6     for(i in 1:N) { for (j in 1:K) {
 7        l[i,j] = sum(dpois(data[i,], c[j,]*rm[i], log=T)) }}
 8     for(i in 1:N) {
 9        p[i,] = q*exp(l[i,]-max(l[i,])); p[i,] = p[i,]/sum(p[i,])}
10     q = colMeans(p)
11     c = (t(p) %*% data)/colSums(p)
12
13     c <<- c; q <<- q; p <<- p;
14     }
```

Notes:
Differences with respect to the basic version shown before are marked in red.
Line 4: Sets mean of class vector elements to 1.
Line 5: rm is a vector containing the mean of the bin counts for each sample.
Line 7: The expected bin counts are adjusted to the sample mean.

A.1.6 Shape-based partitioning with flips

```
1  em_shape_flip = function(c,q,data) {
2     K=dim(c)[1]; N=dim(data)[1]
3     l=array(dim=c(N,K,2)); p=array(dim=c(N,K,2))
4     for(i in 1:K) {c[i,]=c[i,]/mean(c[i,])}
5     rm=rowMeans(data)
6     for(i in 1:N) {
7        for(j in 1:K) {l[i,j,1] = sum(dpois(data[i,],     c[j,] *rm[i],log=T))}
8        for(j in 1:K) {l[i,j,2] = sum(dpois(data[i,],rev(c[j,])*rm[i],log=T))}
9        }
10    for(i in 1:N) {
11       p[i,,] = q*exp(l[i,,]-max(l[i,,])); p[i,,] = p[i,,]/sum(p[i,,])}
12    q = apply(p, c(2,3), mean)
13    c = (t(p[,,1]) %*% data) + (t(p[,,2]) %*% t(apply(data,1,rev)))
14    c = c / apply(p,2,sum)
15
16    c <<- c; q <<- q; p <<- p;
17    }
```

Notes:

Differences with respect to the shape-based version without flips are marked in red.

Line 1: The argument *q* is now a matrix with rows corresponding to classes and columns to flip states.

Line 3: *l* and *p* are now 3-dimensional arrays.

Line 11: The mean is also taken over the two flip states. This reflects the assumption and ensures that the two flip states are equally probable for equal class. Different choices are conceivable.

A.1.7 Shape-based partitioning with shifts

```
em_shape_shift = function(c,q,data) {
   K=dim(c)[1]; L=dim(c)[2]; N=dim(data)[1]; S=dim(q)[2]
   l=array(dim=c(N,K,S)); p=array(dim=c(N,K,S))
   for(i in 1:K) {c[i,]=c[i,]/mean(c[i,])}
   rm=matrix(nrow=N, ncol=S)
   for(k in 1:S) {rm[,k] = rowMeans(data[,k:(k+L-1)])}
   for(i in 1:N) { for (j in 1:K) { for (k in 1:S) {
      l[i,j,k]=sum(dpois(data[i,k:(k+L-1)], c[j,] *rm[i,k],log=T)) }}}
   for(i in 1:N) {
      p[i,,] = q*exp(l[i,,]-max(l[i,,])); p[i,,] = p[i,,]/sum(p[i,,])}
   q = apply(p, c(2,3), mean)
   c = 0; for(k in 1:S) {
   c = c + (t(p[,,k]) %*% data[,k:(k+L-1)])/colSums(p[,,k])}
   c = c / apply(p,2,sum)

   m=sum((1:S)*colSums(q)); s=sum(((1:S)-m)**2*colSums(q))**0.5
   for (i in 1:K) {
      q[i,] = sum(q[i,]) * dnorm(1:S,floor(S/2)+1,s) /
              sum(dnorm(1:S,floor(S/2)+1,s))
      }

   c <<- c; q <<- q; p <<- p;
23    }
```

Notes:

Differences with respect to the shape-based version without flips are marked in red.

Line 1: The argument q is now a matrix with rows corresponding to classes and columns to shift indices.

Line 2: L is the length of the pattern/class, which now is shorter than the length of the samples.

Line 3: l and p are now 3-dimensional arrays. The 3rd dimension relates to the shift index.

Line 5: rm is now a matrix because the mean bincount value varies as a function of the shift index.

Line 12: c is computed as a probability-weighted sum over samples and shift indices. The sum over samples is achieved via a matrix multiplication, the sum over shift states via a loop.

Line 16: This and the following four lines implemented is a regularization step, which prevents the shift state probability distribution from moving out of the center. In fact, a centered Gaussian distribution is imposed. Only the variance parameter is estimated from the data in a non-class specific manner. Other regularization recipes are conceivable.

A.1.8 Shape-based partitioning with flips and shifts

```
em_shape_shift_flip = function(c,q,data) {
   K=dim(c)[1]; L=dim(c)[2]; N=dim(data)[1]; S=dim(q)[2]
   l=array(dim=c(N,K,S,2)); p=array(dim=c(N,K,S,2))
   for(i in 1:K) {c[i,]=c[i,]/mean(c[i,])}
   rm=matrix(nrow=N, ncol=S)
   for(k in 1:S) {rm[,k] = rowMeans(data[,k:(k+L-1)])}
   for(i in 1:N) { for (j in 1:K) { for (k in 1:S) {
      l[i,j,k,1]=sum(dpois(
             data[i,k   :(k+L-1)],c[j,]*rm[i,k    ],log=T))
      l[i,j,k,2]=sum(dpois(
         rev(data[i,])[k:(k+L-1)],c[j,]*rm[i,S-k+1],log=T))
      }}}

   for(i in 1:N) {
      p[i,,,] = q*exp(l[i,,,]-max(l[i,,,]))
      p[i,,,] = p[i,,,]/sum(p[i,,,])
      }
   q = apply(p, c(2,3,4), mean)
   c = 0; for(k in 1:S) {c = c
      + (t(p[,,k,1]) %*%         data[,k      :(k+L-1)])
      + (t(p[,,k,2]) %*% t(apply(data[,(S-k+1):(S+L-k)],1,rev)))
      }
   c = c / apply(p,2,sum)

   m= sum((1:S)*      apply(q,2,sum));
   s=sum(((1:S)-m)**2*apply(q,2,sum))**0.5
   for (i in 1:K) {
         q[i,,1] = q[i,,2] = sum(q[i,,]) * dnorm(1:S,floor(S/2)+1,s) /
                          sum(dnorm(1:S,floor(S/2)+1,s)) / 2
      }

   c <<- c; q <<- q; p <<- p;
33     }
```

Notes:

Differences with respect to the shape-based version with shifts are marked in red.

Line 1: The argument q is now a 3-dimensional array with dimensions 1, 2 and 3 corresponding classes, shift indices and flip states, respectively.

Line 3: l and p are now 4-dimensional arrays.

Line 8: (.. and following lines) The two flip states are treated separately.

A.2 *R* code for generating simulated data

A.2.1 Two classes without shifts and flips

The code below was used to generate the data for the experiments summarized by Figure 4.1 and Tables 4.1, 4.2. Note that the coverage parameter f (here set to 50) was varied from 0.5 to 50 to generate data sets of different coverage.

```
n_samples = 1000
class1_m  = 60
class1_s  = 10
class2_m  = 75
class2_s  = 3
f         = 50

data1 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class1_m, class1_s)
   lambda = f*lambda/sum(lambda)
   data1[sample,] = rpois(100,lambda)
   }
data2 = matrix(NA,nrow=n_samples, ncol=100)
   for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class2_m, class2_s)
   lambda = f*lambda/sum(lambda)
   data2[sample,] = rpois(100,lambda)
   }
data = rbind( data1, data2)
```

A.2.2 Two classes with flips

The code below was used to generate the data for the experiments summarized by Figure 4.2 and Table 4.3.

```
n_samples = 1000
class1_m  = 40
class1_s  = 1
class2_m  = 70
```

```
class2_s  = 5
f         = 50

data1 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class1_m, class1_s)
   lambda = f*lambda/sum(lambda)
   data1[sample,] = rpois(100,lambda)
   }
data2 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class1_m, class1_s)
   lambda = f*lambda/sum(lambda)
   data2[sample,] = rev(rpois(100,lambda))
   }
data3 = matrix(NA,nrow=n_samples, ncol=100)
   for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class2_m, class2_s)
   lambda = f*lambda/sum(lambda)
   data3[sample,] = rpois(100,lambda)
   }
data4 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class2_m, class2_s)
   lambda = f*lambda/sum(lambda)
   data4[sample,] = rev(rpois(100,lambda))
}

data = rbind( data1, data2, data3, data4)
```

A.2.3 Four classes

The code below was used to generate the data for the experiments summarized by Figure 4.3:

```
n_samples = 1000

class1_m = 40
class1_s =  3
class2_m = 40
class2_s = 10
class3_m = 70
class3_s =  2
class4_m = 60
class4_s = 5

f        = 5

data1 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class1_m, class1_s)
   lambda = f*lambda/sum(lambda)
   data1[sample,] = rpois(100,lambda)
   }
data2 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
  lambda = dnorm(1:100, class2_m, class2_s)
  lambda = f*lambda/sum(lambda)
  data2[sample,] = rpois(100,lambda)
}
data3 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
  lambda = dnorm(1:100, class3_m, class3_s)
  lambda = f*lambda/sum(lambda)
  data3[sample,] = rpois(100,lambda)
}
data4 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
  lambda = dnorm(1:100, class4_m, class4_s)
  lambda = f*lambda/sum(lambda)
  data4[sample,] = rpois(100,lambda)
}

data  = rbind( data1, data2, data3, data4)
```

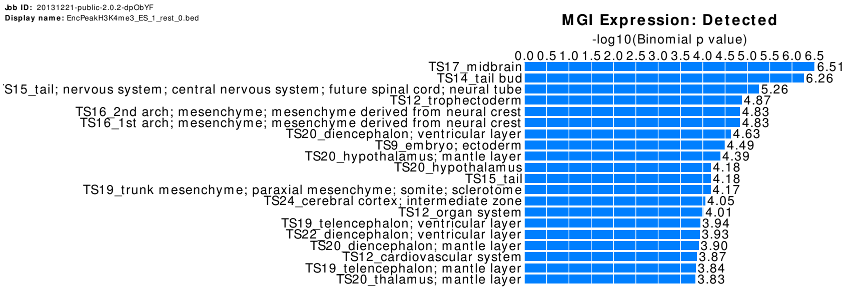
A.2.4 Two classes characterized by co-localizing peaks of different width

The code below was used to generate the data for the experiments summarized by Figure 4.4:

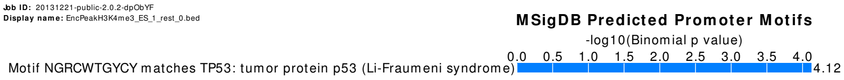
```
n_samples = 1000
class1_m = 60
class1_s = 10
class2_m = 60
class2_s = 3
f = 50

data1 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class1_m, class1_s)
   lambda = f*lambda/sum(lambda)
   data1[sample,] = rpois(100,lambda)
   }
data2 = matrix(NA,nrow=n_samples, ncol=100)
for(sample in 1:n_samples) {
   lambda = dnorm(1:100, class2_m, class2_s)
   lambda = f*lambda/sum(lambda)
   data2[sample,] = rpois(100,lambda)
   }

data = rbind( data1, data2)
```

B Appendix for Chapter 5

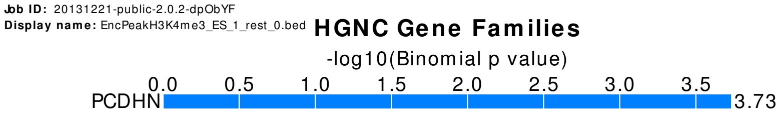
For the H3K4me3, H3K27me3 ENCODE peak data we used the University of Washington (UW) ENCODE data. Both replicates used and the file type used is “narrowPeak".
For H3K4me3 data the GEO accession numbers are: GSM945187, GSM945177, GSM945166, GSM945170, GSM945178, GSM945225, GSM945198, GSM945229, GSM945185, GSM945191, GSM945322, GSM945190, GSM945321, GSM945242, GSM945249, GSM945163, GSM945310, GSM945312, GSM945308, GSM945306, GSM945286, GSM945218, GSM945239, GSM945159, GSM945292, GSM945284, GSM945276, GSM945271, GSM945181, GSM945273, GSM945251, GSM945175, GSM945262, GSM945216, GSM945199, GSM945214, GSM945265, GSM945215.
For H3K27me3 data the GEO accession numbers are: GSM1010913, GSM945204, GSM945301, GSM945183, GSM945325, GSM945326, GSM945323, GSM945320, GSM945277, GSM945160, GSM945180, GSM945300, GSM945200.
For the H3K4me1, H3K9me3, H3K27ac ENCODE peak data we used the Broad ENCODE data. The file type used is “broadPeak".
For H3K4me1 data the GEO accession numbers are: GSM1003535, GSM733782, GSM733705, GSM733761, GSM733661, GSM733690, GSM733710, GSM733698, GSM733649, GSM733704.
For H3K9me3 data the GEO accession numbers are: GSM1003538, GSM1003585, GSM1003485, GSM733730, GSM1003482, GSM1003517, GSM1003491, GSM1003553, GSM1003528, GSM1003531, GSM733681.
For H3K27ac data the GEO accession numbers are: GSM1003559, GSM1003459, GSM733718, GSM733660, GSM733755, GSM733666, GSM733691, GSM733763, GSM733662, GSM733674, GSM733646, GSM733739.



(a) MGI Expression: Detected

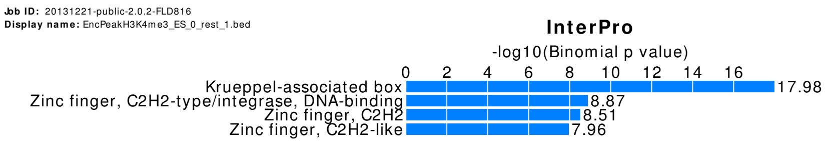


(b) MSigDB Predicted Promoter Motifs

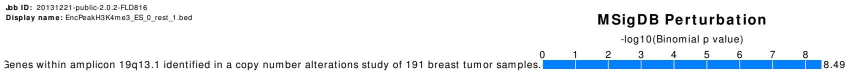


(c) HGNC Gene Families

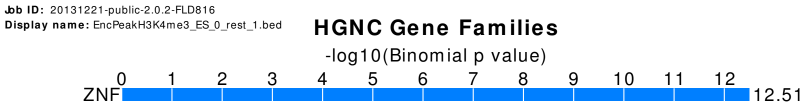
Figure B.1 – *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 1 and rest of the cell types (62 replicates) have all 0 (one error allowed at most on both sides) using all replicate ENCODE peak data.*



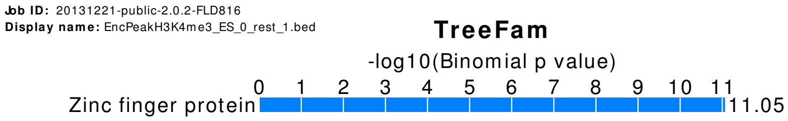
(a) InterPro



(b) MSigDB Perturbation

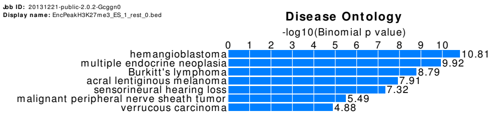


(c) HGNC Gene Families

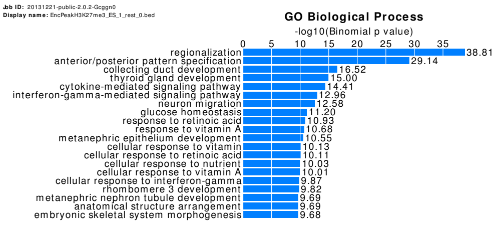


(d) TreeFam

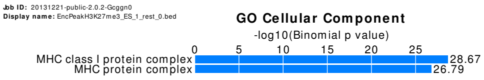
Figure B.2 – *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 0 and rest of the cell types (62 replicates) have all 1 (one error allowed at most on both sides) using all replicate ENCODE peak data.*



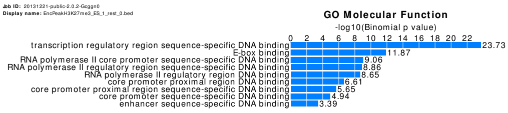
(a) Disease Ontology



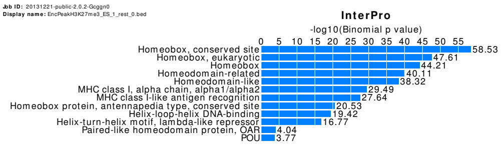
(b) GO Biological Process



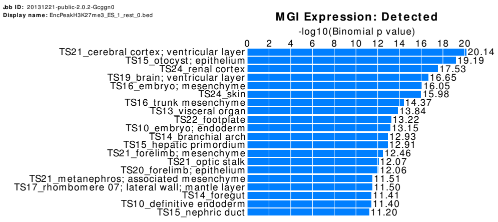
(c) GO Cellular Component



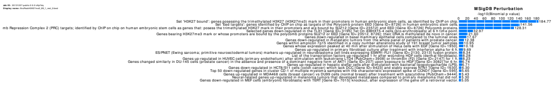
(d) GO Molecular Function



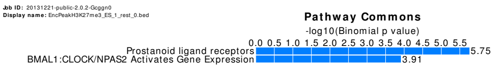
(e) InterPro



(f) MGI Expression

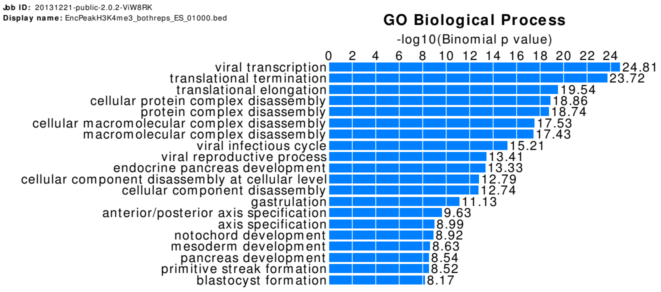


(g) MSigDB Perturbation

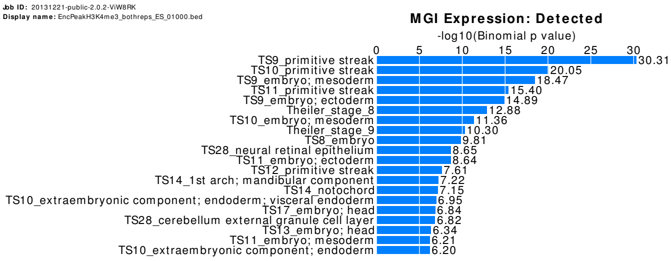


(h) Pathway Commons

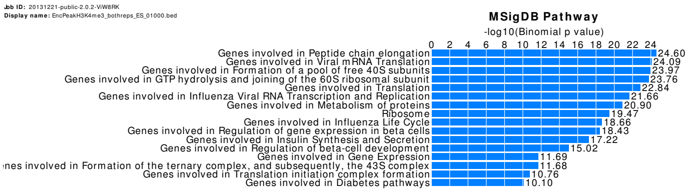
Figure B.3 – *For H3K27me3 data: gene enrichment analysis containing regions of the genome where ES cells (10 replicates) are all 1 and rest of the cell types (13 replicates) have all 0 (one error allowed at most on both sides) using all replicate ENCODE peak data.*



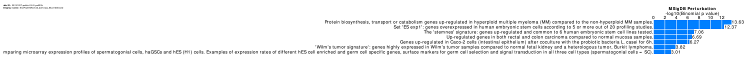
(a) GO Biological Process



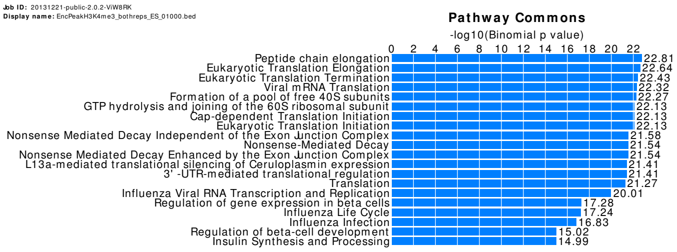
(b) MGI Expression



(c) MSigDB Pathway

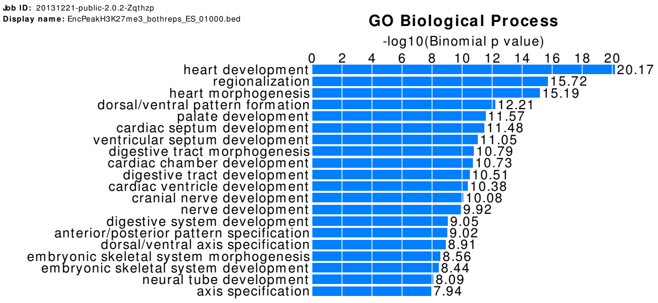


(d) MSigDB Perturbation

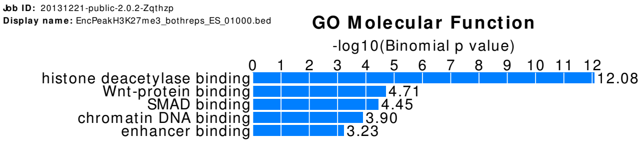


(e) Pathway Commons

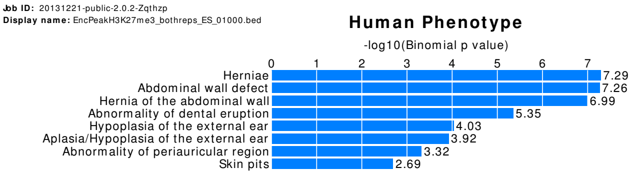
Figure B.4 – *For H3K4me3 data: gene enrichment analysis containing regions of the genome where ES cells for day 0, 2, 5, 9, 14 have a pattern 01000. ENCODE peaks are used and only regions which have identical values of all the replicates for each cell type are considered.*



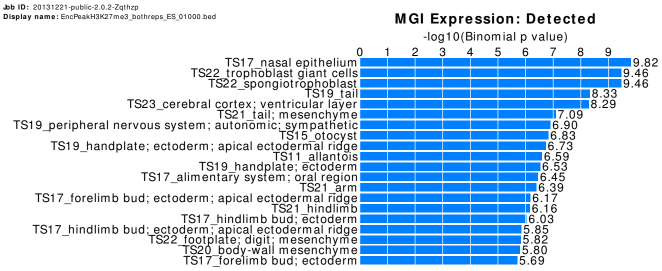
(a) GO Biological Process



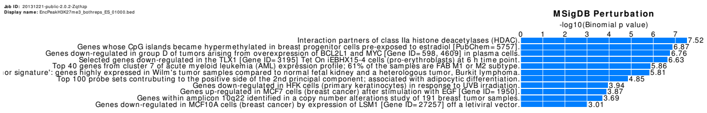
(b) GO Molecular Function



(c) Human Phenotype



(d) MGI Expression



(e) MSigDB Perturbation

Figure B.5 – *For H3K27me3 data: gene enrichment analysis containing regions of the genome where ES cells for day 0, 2, 5, 9, 14 have a pattern 01000. ENCODE peaks are used and only regions which have identical values of all the replicates for each cell type are considered.*

Bibliography

[1] Dhammika Amaratunga and Javier Cabrera. Analysis of data from viral dna microchips. *Journal of the American Statistical Association*, 96(456):1161–1170, 2001.

[2] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome biol*, 11(10):R106, 2010.

[3] Detlev Arendt. The evolution of cell types in animals: emerging principles from molecular studies. *Nature Reviews Genetics*, 9(11):868–882, 2008.

[4] Kevin Atteson. The performance of neighbor-joining algorithms of phylogeny reconstruction. In *Computing and Combinatorics*, pages 101–110. Springer, 1997.

[5] Timothy Bailey, Pawel Krajewski, Istvan Ladunga, Celine Lefebvre, Qunhua Li, Tao Liu, Pedro Madrigal, Cenny Taslim, and Jie Zhang. Practical guidelines for the comprehensive analysis of chip-seq data. *PLoS computational biology*, 9(11):e1003326, 2013.

[6] Tanya Barrett et al. NCBI GEO: archive for functional genomics data sets-update. *Nucleic acids research*, 41(D1):D991–D995, 2013.

[7] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, 2007.

[8] Elizaveta V Benevolenskaya. Histone h3k4 demethylases are essential in development and differentiation this paper is one of a selection of papers published in this special issue, entitled 28th international west coast chromatin and chromosome conference, and has undergone the journal's usual peer review process. *Biochemistry and cell biology*, 85(4):435–443, 2007.

[9] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

[10] Shelley L Berger. Histone modifications in transcriptional regulation. *Current Opinion in Genetics & Development*, 12(2):142–148, 2002.

[11] Bradley E Bernstein, Tarjei S Mikkelsen, Xiaohui Xie, Michael Kamal, Dana J Huebert, James Cuff, Ben Fry, Alex Meissner, Marius Wernig, Kathrin Plath, et al. A bivalent chromatin structure marks key developmental genes in embryonic stem cells. *Cell*, 125(2):315–326, 2006.

[12] Manoj Bhasin, Ellis L Reinherz, and Pedro A Reche. Recognition and classification of histones using support vector machine. *Journal of Computational Biology*, 13(1):102–112, 2006.

[13] Ewan Birney, John A Stamatoyannopoulos, Anindya Dutta, Roderic Guigó, Thomas R Gingeras, Elliott H Margulies, Zhiping Weng, Michael Snyder, Emmanouil T Dermitzakis, Robert E Thurman, et al. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799–816, 2007.

[14] Christoph Bock and Thomas Lengauer. Computational epigenetics. *Bioinformatics*, 24(1):1–10, 2008.

[15] Hamid Bolouri. Modeling genomic regulatory networks with big data. *Trends in Genetics*, 30(5):182–191, 2014.

[16] Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[17] Judy A Brusslan, Ana M Rus Alvarez-Canterbury, Nishanth Ulhas Nair, Judd C Rice, Michael J Hitchler, and Matteo Pellegrini. Genome-wide evaluation of histone methylation changes associated with leaf senescence in Arabidopsis. *PLoS ONE*, 7(3):e33151, 2012.

[18] Orion J Buske, Michael M Hoffman, Nadia Ponts, Karine G Le Roch, and William S Noble. Exploratory analysis of genomic segmentations with Segtools. *BMC bioinformatics*, 12(1):415, 2011.

[19] Ming-Sin Cheung, Thomas A Down, Isabel Latorre, and Julie Ahringer. Systematic bias in high-throughput sequencing data and its correction by BEADS. *Nucleic acids research*, 39(15):e103–e103, 2011.

[20] William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

[21] ENCODE Project Consortium et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[22] Geoffrey M Cooper and Robert E Hausman. *The cell.* Sinauer Associates Sunderland, 2000.

[23] M.P. Creyghton et al. Histone H3K27ac separates active from poised enhancers and predicts developmental state. *Proceedings of the National Academy of Sciences*, 107(50):21931–21936, 2010.

[24] Kairong Cui, Chongzhi Zang, Tae-Young Roh, Dustin E Schones, Richard W Childs, Weiqun Peng, and Keji Zhao. Chromatin signatures in multipotent human hematopoietic stem cells indicate the fate of bivalent genes during differentiation. *Cell stem cell*, 4(1):80–93, 2009.

[25] Gerard V Dallal. *The little handbook of statistical practice.* Gerard V. Dallal, 1999.

[26] William HE Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of mathematical biology*, 49(4):461–467, 1987.

[27] Alvaro R De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Transactions on Medical Imaging*, 14(1):132, 1995.

[28] Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

[29] Richard Desper and Olivier Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of computational biology*, 9(5):687–705, 2002.

[30] Patrik D'haeseleer et al. How does gene expression clustering work? *Nature biotechnology*, 23(12):1499–1502, 2005.

[31] Chuong B Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897–900, 2008.

[32] René Dreos et al. Epd and epdnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic acids research*, 41(D1):D157–D164, 2013.

[33] Sandrine Dudoit, Yee Hwa Yang, Matthew J Callow, and Terence P Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica sinica*, 12(1):111–140, 2002.

[34] Sean R Eddy and Richard Durbin. RNA sequence analysis using covariance models. *Nucleic acids research*, 22(11):2079–2088, 1994.

[35] Anthony WF Edwards and Cavalli LL Sforza. The reconstruction of evolution. *Heredity*, 18, 1963.

[36] Jason Ernst and Manolis Kellis. Discovery and characterization of chromatin states for systematic annotation of the human genome. *Nature biotechnology*, 28(8):817–825, 2010.

[37] Laurent Excoffier and Montgomery Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular biology and evolution*, 12(5):921–927, 1995.

[38] Daniel H Farkas. *DNA simplified: the hitchhiker's guide to DNA*. American Association for Clinical Chemistry, Incorporated, 1996.

[39] Anthony P Fejes, Gordon Robertson, Mikhail Bilenky, Richard Varhol, Matthew Bainbridge, and Steven JM Jones. FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology. *Bioinformatics*, 24(15):1729–1730, 2008.

[40] Joseph Felsenstein. *Inferring phylogenies*. Sinauer Associates, 2004.

[41] Peter C Fishburn. *Interval orders and interval graphs: A study of partially ordered sets*. Wiley, New York, 1985.

[42] Walter M Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

[43] Walter M Fitch, Emanuel Margoliash, et al. Construction of phylogenetic trees. *Science*, 155(760):279–284, 1967.

[44] Dan Frumkin, Adam Wasserstrom, Shai Kaplan, Uriel Feige, and Ehud Shapiro. Genomic variability within an organism exposes its cell lineage tree. *PLoS computational biology*, 1(5):e50, 2005.

[45] Sudhindra R Gadagkar and Sudhir Kumar. Maximum likelihood outperforms maximum parsimony even when evolutionary rates are heterotachous. *Molecular biology and evolution*, 22(11):2139–2141, 2005.

[46] Eugenia Giannopoulou and Olivier Elemento. An integrated ChIP-seq analysis platform with customizable workflows. *BMC bioinformatics*, 12(1):277, 2011.

[47] P.A. Goloboff. Analyzing large data sets in reasonable times: solutions for composite optima. *Cladistics*, 15(4):415–428, 1999.

[48] Steven N Goodman. Toward evidence-based medical statistics. 1: The P value fallacy. *Annals of internal medicine*, 130(12):995–1004, 1999.

[49] Steven N Goodman. Toward evidence-based medical statistics. 2: The Bayes factor. *Annals of internal medicine*, 130(12):1005–1013, 1999.

[50] Daogang Guan, Jiaofang Shao, Youping Deng, Panwen Wang, Zhongying Zhao, Yan Liang, Junwen Wang, and Bin Yan. CMGRN: a web server for constructing multi-level gene regulatory networks using ChIP-seq and gene expression data. *Bioinformatics*, page btt761, 2014.

[51] Matthew G Guenther, Garrett M Frampton, Frank Soldner, Dirk Hockemeyer, Maya Mitalipova, Rudolf Jaenisch, and Richard A Young. Chromatin structure and gene expression programs of human embryonic and induced pluripotent stem cells. *Cell stem cell*, 7(2):249–257, 2010.

[52] Neil Hall. Advanced sequencing technologies and their wider impact in microbiology. *Journal of Experimental Biology*, 210(9):1518–1525, 2007.

[53] Jennifer Harrow, France Denoeud, Adam Frankish, Alexandre Reymond, Chao-Kung Chen, Jacqueline Chrast, Julien Lagarde, James GR Gilbert, Roy Storey, David Swarbreck, et al. GENCODE: producing a reference annotation for ENCODE. *Genome Biol*, 7(Suppl 1):S4, 2006.

[54] Michael M Hoffman, Orion J Buske, Jie Wang, Zhiping Weng, Jeff A Bilmes, and William Stafford Noble. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature methods*, 9(5):473–476, 2012.

[55] G. Hon et al. ChromaSig: a probabilistic approach to finding common chromatin signatures in the human genome. *PLoS computational biology*, 4(10):e1000201, 2008.

[56] Gary C Hon et al. Predictive chromatin signatures in the mammalian genome. *Human molecular genetics*, 18(R2):R195–R201, 2009.

[57] Soren Impey, Sean R McCorkle, Hyunjoo Cha-Molstad, Jami M Dwyer, Gregory S Yochum, Jeremy M Boss, Shannon McWeeney, John J Dunn, Gail Mandel, and Richard H Goodman. Defining the CREB regulon: a genome-wide analysis of transcription factor regulatory regions. *Cell*, 119(7):1041–1054, 2004.

[58] Justin Jee et al. ACT: aggregation and correlation toolbox for analyses of genome tracks. *Bioinformatics*, 27(8):1152–1154, 2011.

[59] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502, 2007.

[60] Donna Karolchik, Galt P Barber, Jonathan Casper, Hiram Clawson, Melissa S Cline, Mark Diekhans, Timothy R Dreszer, Pauline A Fujita, Luvina Guruvadoo, Maximilian Haeussler, et al. The UCSC Genome Browser database: 2014 update. *Nucleic acids research*, 42(D1):D764–D770, 2014.

[61] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at UCSC. *Genome research*, 12(6):996–1006, 2002.

[62] David J Ketchen and Christopher L Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6):441–458, 1996.

[63] Tae Hoon Kim and Bing Ren. Genome-wide analysis of protein-DNA interactions. *Annu. Rev. Genomics Hum. Genet.*, 7:81–102, 2006.

[64] Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander, and David Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, 1994.

[65] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[66] Chee Seng Ku et al. Studying the epigenome using next generation sequencing. *Journal of medical genetics*, 48(11):721–730, 2011.

[67] Anshul Kundaje et al. Ubiquitous heterogeneity and asymmetry of the chromatin environment at regulatory elements. *Genome research*, 22(9):1735–1747, 2012.

[68] W. Lai et al. ArchAlign: coordinate-free chromatin alignment reveals novel architectures. *Genome Biol*, 11:R126, 2010.

[69] Stephen G Landt et al. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome research*, 22(9):1813–1831, 2012.

[70] Charles E Lawrence and Andrew A Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function, and Bioinformatics*, 7(1):41–51, 1990.

[71] Jeong-Heon Lee, Suzanne RL Hart, and David G Skalnik. Histone deacetylase activity is required for embryonic stem cell differentiation. *Genesis*, 38(1):32–38, 2004.

[72] Albert Lehninger, D Lee Nelson, and Michael M Cox. Lehninger's Principles of Biochemistry, 2005.

[73] Qunhua Li, James B Brown, Haiyan Huang, Peter J Bickel, et al. Measuring reproducibility of high-throughput experiments. *The annals of applied statistics*, 5(3):1752–1779, 2011.

[74] Kun Liang and Sündüz Keleş. Detecting differential binding of transcription factors with ChIP-seq. *Bioinformatics*, 28(1):121–122, 2012.

[75] Ryan Lister et al. Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells. *Nature*, 471(7336):68–73, 2011.

[76] Corrinne G Lobe. Transcription factors and mammalian development. *Current topics in developmental biology*, 27:351–351, 1992.

[77] Karolin Luger, Armin W Mäder, Robin K Richmond, David F Sargent, and Timothy J Richmond. Crystal structure of the nucleosome core particle at 2.8 å resolution. *Nature*, 389(6648):251–260, 1997.

[78] Philip Machanick and Timothy L Bailey. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics*, 27(12):1696–1697, 2011.

[79] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[80] Elaine R Mardis. ChIP-seq: welcome to the new frontier. *Nature methods*, 4(8):613–613, 2007.

[81] C. Martin and Y. Zhang. Mechanisms of epigenetic inheritance. *Current Opinions Cell Biology*, 3(19):266–272, 2007.

[82] Project Consortium ENCODE. A user's guide to the encyclopedia of DNA elements (ENCODE). *PLoS Biol*, 9(4):e1001046, 2011.

[83] Cory Y McLean, Dave Bristor, Michael Hiller, Shoa L Clarke, Bruce T Schaar, Craig B Lowe, Aaron M Wenger, and Gill Bejerano. GREAT improves functional interpretation of cis-regulatory regions. *Nature biotechnology*, 28(5):495–501, 2010.

[84] Radu Mihaescu, Dan Levy, and Lior Pachter. Why neighbor-joining works. *Algorithmica*, 54(1):1–24, 2009.

[85] Tarjei S Mikkelsen, Manching Ku, David B Jaffe, Biju Issac, Erez Lieberman, Georgia Giannoukos, Pablo Alvarez, William Brockman, Tae-Kyung Kim, Richard P Koche, et al. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–560, 2007.

[86] David W Mount. Maximum parsimony method for phylogenetic prediction. *Cold Spring Harbor Protocols*, 2008(4):pdb–top32, 2008.

[87] Nishanth Ulhas Nair, Sunil Kumar, Bernard ME Moret, and Philipp Bucher. Probabilistic partitioning methods to find significant patterns in ChIP-Seq data. *Bioinformatics*, page btu318, 2014.

[88] Nishanth Ulhas Nair, Avinash Das Sahu, Philipp Bucher, and Bernard ME Moret. ChIPnorm: a statistical method for normalizing and identifying differential regions in histone modification ChIP-seq libraries. *PLoS ONE*, 7(8):e39573, 2012.

[89] N.U. Nair, Y. Lin, P. Bucher, and B.M.E. Moret. Phylogenetic Analysis of Cell Types using Histone Modifications. *Proc. 13th Workshop on Algorithms in Bioinformatics (WABI), in Lecture Notes in Computer Science 8126*, pages 326–337, 2013.

[90] N.U. Nair, Y. Lin, A. Manasovska, J. Antic, P. Grnarova, A.D. Sahu, P. Bucher, and B.M.E. Moret. Study of cell differentiation by phylogenetic analysis using histone modification data. *BMC Bioinformatics*, 15:269, 2014.

[91] David L Nelson and Michael M Cox. *Lehninger principles of biochemistry*. W. H. Freeman, 5th edition, 2008.

[92] Alexey I Nesvizhskii, Andrew Keller, Eugene Kolker, and Ruedi Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical chemistry*, 75(17):4646–4658, 2003.

[93] F.G.G. Nielsen et al. CATCHprofiles: Clustering and alignment tool for ChIP profiles. *PloS one*, 7(1):e28272, 2012.

[94] William S Noble. How does multiple testing correction work? *Nature biotechnology*, 27(12):1135–1137, 2009.

[95] Emmanuel Paradis, Julien Claude, and Korbinian Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20(2):289–290, 2004.

[96] Elodie Portales-Casamar et al. JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. *Nucleic acids research*, 38(suppl 1):D105–D110, 2010.

[97] Robert Clay Prim. Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401, 1957.

[98] Eric J Richards. Inherited epigenetic variation-revisiting soft inheritance. *Nature Reviews Genetics*, 7(5):395–401, 2006.

[99] Gordon Robertson, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas Zeng, Ghia Euskirchen, Bridget Bernier, Richard Varhol, Allen Delaney, et al. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature methods*, 4(8):651–657, 2007.

[100] Kate R Rosenbloom, Timothy R Dreszer, Michael Pheasant, Galt P Barber, Laurence R Meyer, Andy Pohl, Brian J Raney, Ting Wang, Angie S Hinrichs, Ann S Zweig, et al. ENCODE whole-genome data in the UCSC Genome Browser. *Nucleic acids research*, 38(suppl 1):D620–D625, 2010.

[101] Kate R Rosenbloom, Cricket A Sloan, Venkat S Malladi, Timothy R Dreszer, Katrina Learned, Vanessa M Kirkup, Matthew C Wong, Morgan Maddren, Ruihua Fang, Steven G Heitner, et al. ENCODE Data in the UCSC Genome Browser: year 5 update. *Nucleic acids research*, 41(D1):D56–D63, 2013.

[102] Joel Rozowsky, Ghia Euskirchen, Raymond K Auerbach, Zhengdong D Zhang, Theodore Gibson, Robert Bjornson, Nicholas Carriero, Michael Snyder, and Mark B Gerstein. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature biotechnology*, 27(1):66–75, 2009.

[103] Vincenzo EA Russo, Robert A Martienssen, Arthur D Riggs, et al. *Epigenetic mechanisms of gene regulation.* Cold Spring Harbor Laboratory Press, 1996.

[104] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

[105] C.D. Schmid and P. Bucher. ChIP-Seq data reveal nucleosome architecture of human promoters. *Cell*, 131(5):831 – 832; author reply 832–833, 2007.

[106] Dustin E Schones et al. Dynamic regulation of nucleosome positioning in the human genome. *Cell*, 132(5):887–898, 2008.

[107] Gabriele Schweikert, Botond Cseke, Thomas Clouaire, Adrian Bird, and Guido Sanguinetti. MMDiff: quantitative testing for shape changes in ChIP-Seq data sets. *BMC Genomics*, 14(1):826, 2013.

[108] Eran Segal, R Yelensky, and Daphne Koller. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, 19(suppl 1):i273–i282, 2003.

[109] Robin Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[110] Adam Siepel, Gill Bejerano, Jakob S Pedersen, Angie S Hinrichs, Minmei Hou, Kate Rosenbloom, Hiram Clawson, John Spieth, LaDeana W Hillier, Stephen Richards, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome research*, 15(8):1034–1050, 2005.

[111] Deepankar Pratap Singh et al. Genome-defence small RNAs exapted for epigenetic mating-type inheritance. *Nature*, 2014.

[112] Montgomery Slatkin and Laurent Excoffier. Testing for linkage disequilibrium in genotypic data using the Expectation-Maximization algorithm. *Heredity*, 76(4), 1996.

[113] Peter HA Sneath and Robert R Sokal. *Numerical taxonomy.* 1963.

[114] R Sokal and C Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[115] Qiang Song and Andrew D Smith. Identifying dispersed epigenomic domains from ChIP-Seq data. *Bioinformatics*, 27(6):870–871, 2011.

[116] Alexandros Stamatakis. Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[117] Stephen Stigler. Fisher and the 5% level. *Chance*, 21(4):12–12, 2008.

[118] John D Storey. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of statistics*, pages 2013–2035, 2003.

[119] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

[120] Brian D Strahl and C David Allis. The language of covalent histone modifications. *Nature*, 403(6765):41–45, 2000.

[121] Jack Sullivan. Maximum-likelihood methods for phylogeny estimation. *Methods in enzymology*, 395:757–779, 2005.

[122] Cenny Taslim, Jiejun Wu, Pearlly Yan, Greg Singer, Jeffrey Parvin, Tim Huang, Shili Lin, and Kun Huang. Comparative study on ChIP-seq data: normalization and binding pattern characterization. *Bioinformatics*, 25(18):2334–2340, 2009.

[123] Chia-Lin Wei, Qiang Wu, Vinsensius B Vega, Kuo Ping Chiu, Patrick Ng, Tao Zhang, Atif Shahab, How Choong Yong, YuTao Fu, Zhiping Weng, et al. A global map of p53 transcription-factor binding sites in the human genome. *Cell*, 124(1):207–219, 2006.

[124] Elizabeth G Wilbanks and Marc T Facciotti. Evaluation of algorithm performance in ChIP-seq peak detection. *PloS one*, 5(7):e11471, 2010.

[125] Han Xu, Chia-Lin Wei, Feng Lin, and Wing-Kin Sung. An HMM approach to genome-wide identification of differential histone modification sites from ChIP-seq data. *Bioinformatics*, 24(20):2344–2349, 2008.

[126] Jun Yang, Abhishek Mitra, Norbert Dojer, Shuhua Fu, Maga Rowicka, and Allan R Brasier. A probabilistic approach to learn chromatin architecture and accurate inference of the NF-κB/RelA regulatory network using ChIP-Seq. *Nucleic acids research*, 41(15):7240–7259, 2013.

[127] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, David M Lin, Vivian Peng, John Ngai, and Terence P Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic acids research*, 30(4):e15–e15, 2002.

[128] Yuhong Yang. Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950, 2005.

[129] Tao Ye et al. seqMINER: an integrated ChIP-seq data interpretation platform. *Nucleic acids research*, 39(6):e35–e35, 2011.

[130] Matthew D Young, Tracy A Willson, Matthew J Wakefield, Evelyn Trounson, Douglas J Hilton, Marnie E Blewitt, Alicia Oshlack, and Ian J Majewski. ChIP-seq analysis reveals distinct H3K27me3 profiles that correlate with transcriptional activity. *Nucleic acids research*, 39(17):7415–7427, 2011.

[131] Yong Zhang, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, et al. Model-based analysis of ChIP-Seq (MACS). *Genome Biol*, 9(9):R137, 2008.

[132] Bing Zhu and Danny Reinberg. Epigenetic inheritance: uncontested? *Cell research*, 21(3):435–441, 2011.

Nishanth Ulhas Nair

CONTACT INFORMATION

Avenue de Sévelin 4A,
1007 Lausanne
Switzerland

Cell: +41 78 9419 220
*Email:*nishanth.u.nair@gmail.com

CITIZENSHIP

India

RESEARCH INTERESTS

I am in general interested in broad area of Computational and Systems Biology. Currently I am studying problems in epigenomics, primarily computational methods for the analysis of histone marks. I am also interested in understanding the role of epigenetic markers for cell differentiation, evolution, gene regulation, and cancer.

EDUCATION

École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Ph.D. student (Computer and Communication Sciences)
Laboratory for Computational Biology and Bioinformatics, School of Computer and Communication Sciences (Sept. 2009 — present)

- Thesis: *Computational studies in epigenomics using histone modification data*
  Advisors: Prof. Bernard M.E. Moret and Dr. Philipp Bucher
  Area of Study: Computational Biology

Indian Institute of Science (IISc), Bangalore, India.

Master of Science (Engineering)
Department of Electrical Communication Engineering (ECE), (Jan. 2006 — Jan. 2009)

- Thesis: *Joint Evaluation of Multiple Speech Patterns for Speech Recognition and Training*
  Advisor: Prof. T.V. Sreenivas
  Areas of Study: Automatic Speech Recognition and Signal Processing
  Awarded best thesis medal

B.M.S. College of Engineering, Bangalore. Visveswaraiah Technological University, Belgaum, India.

Bachelor of Engineering (Electronics and Communication)
Department of Electronics and Communication Engineering, (Aug. 2001 — June 2005)
First Class with Distinction

EXPERIENCE

PROFESSIONAL TRAINING AND WORK EXPERIENCE

- Research Assistant, Laboratory of Computational Biology and Bioinformatics, EPFL (Sept. 2009 — present).
- "Mathematical and Computational Approaches in High Throughput Genomics," a semester long program by Institute of Pure and Applied Mathematics, University of California Los Angeles (Sept. 2011 — Dec. 2011).
- Consultant, Microsoft Research India (MSRI) — officially with Spectrum Consultants India Pvt. Ltd. (Aug. 2008 — Aug. 2009)

  Joint project with the Bioinformatics Group in IISc, Bangalore.

Research Area: Systems Biology
Advisors: Dr. Navin Goyal and Prof. Nagasuma Chandra

- Taught one month graduate course "Short Course on Automatic Speech Recognition" at IISc. Teaching capabilities was rated at 3.4 / 5.0 by the students.
- Software engineer, Tata Consultancy India Private Limited (2005 — 2006).
- Intern, Indian Institute of Astrophysics, Bangalore, in 2004 — 2005 for 6 months.

  Helped in designing a radioastronomy telescope to capture the outer radiations of the sun.

PUBLICATIONS **Journal Papers**

1. **N.U. Nair**, Y. Lin, A. Manasovska, J. Antic, P. Grnarova, A.D. Sahu, P. Bucher, and B.M.E. Moret, "Study of cell differentiation by phylogenetic analysis using histone modification data," BMC Bioinformatics, 15:269, 2014.
2. **N.U. Nair**, S. Kumar, B.M.E. Moret, and P. Bucher, "Probabilistic partitioning methods to find significant patterns in ChIP-Seq data," Bioinformatics, btu318, 2014.
3. **N.U. Nair***, A.D. Sahu*, P. Bucher, and B.M.E. Moret, "ChIPnorm: a statistical method for normalizing and identifying differential regions in histone modification ChIP-seq libraries," PLoS ONE 7(8):e39573, 2012 (* contributed equally).
4. J.A. Brusslan, A.M.R. Alvarez-Canterbury, **N.U. Nair**, J.C. Rice, M.J. Hitchler, M. Pellegrini, "Genome-wide evaluation of histone methylation changes associated with leaf senescence in Arabidopsis," PLoS ONE 7(3):e33151, 2012.
5. **N.U. Nair** and T.V. Sreenivas, "Multi-Pattern Viterbi Algorithm for Joint Decoding of Multiple Speech Patterns," Signal Processing, Dec. 2010 (invited paper based on the EUSIPCO 2008 paper).
6. **N.U. Nair** and T.V. Sreenivas, "Joint Evaluation of Multiple Speech Patterns for Speech Recognition and Training," Computer Speech and Language, 24, 307340, 2010.

Conference Proceedings Papers

1. **N.U. Nair**, Y. Lin, P. Bucher, and B.M.E. Moret, "Phylogenetic analysis of cell types using histone modifications," Proc. 13th Workshop on Algorithms in Bioinformatics WABI'13, in Lecture Notes in Computer Science 8126, 326337, Springer Verlag, 2013.
2. **N.U. Nair**, N. Goyal, N.R. Chandra, "Enhanced flux balance analysis to model metabolic networks," ACM International Conference On Bioinformatics and Computational Biology (ACM-BCB), 2010.
3. **N.U. Nair** and T.V. Sreenivas, "Viterbi Algorithm for Multi-Pattern Joint Decoding," IEEE TENCON 2009.
4. D. Bansal, **N. Nair**, R. Singh, B. Raj, "A Joint Decoding Algorithm for Multiple-Example-Based Addition of Words to a Pronounciation Lexicon," IEEE International Conference of Acoustics, Speech, and Signal Processing (ICASSP), 2009.
5. **N.U. Nair** and T.V. Sreenivas, "Multi Pattern Dynamic Time Warping for Automatic Speech Recognition," IEEE TENCON, 2008.

6. **N.U. Nair** and T.V. Sreenivas, “Forward/Backward Algorithms for Joint Multi Pattern Speech Recognition,” European Signal Processing Conference (EUSIPCO), 2008. (rated as one of the best papers in the conference and we got an invite to write a journal paper in EURASIP Journal Signal Processing.)
7. **N.U. Nair** and T.V. Sreenivas, “Joint Decoding of Multiple Speech Patterns for Robust Speech Recognition,” IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE ASRU), December, 2007.

Book Chapters

1. **N.U. Nair** and T.V. Sreenivas, Algorithms for Joint Evaluation of Multiple Speech Patterns for Automatic Speech Recognition, invited chapter (invited based on our IEEE ASRU 2007 paper) in the book Speech Recognition: Technologies and Applications, ISBN 9789537619299, InTeh, Nov. 2008.

Patents

1. **N.U. Nair** and T.V. Sreenivas, System and Method of Using Multi Pattern Viterbi Algorithm for Joint Decoding of Multiple Patterns, Indian patent (No. 02870/CHE/2009), EP patent 2,502,226, WO patent 2,011,061,636, and US patent (No. 12/652,608).

Technical Articles

1. **N.U. Nair**, Manjunath K., M.M. Alam, “Microcontroller-Based Infrared Tracking Robot,” Electronics For You magazine, April 2007.

ACADEMIC HONOURS

- Prof. F.M. Mowdawalla Medal for best Masters thesis award in the ECE department at Indian Institute of Science for the academic year 2008 — 2009.
- Certificate of Appreciation for exemplary work as teaching assistant, 2012 (courses as TA: Statistical Signal Processing and Applications, Spring 2012; Advanced Algorithms, Fall 2012)
- Received Swiss National Science Foundation funding for PhD work for 3 years (2009 — 2012) .
- Awarded second prize in project work at the IEEE Signal Processing Society organized Summer School in Automatic Speech Recognition.
- EUSIPCO 2008 conference paper rated as one of the best in the EUSIPCO conference. Invited to write a journal paper in EURASIP Journal Signal Processing.
- Invited to write book chapters in Speech Recognition.

RELEVANT COURSE WORK

PhD (EPFL)

Advanced Algorithms, Cell Biology and Biochemistry for Engineers, Machine Learning, Computational Molecular Biology, Applied Stochastic Processes, Cancer Biology-1, Advanced Analysis-1, Genomics and network architecture, Principles and applications of systems biology.

MSc (Engg.) (IISc)

Random Processes, Automatic Speech recognition and Algorithms, Adaptive Signal Processing, Linear and Nonlinear Time Series Analysis, Speech Information Processing, Multimedia Systems.

Undergraduate

Neural Networks, Digital Signal Processing, Mathematics - 1, 2, 3, 4, Digital Communication, Control Systems, Signals and Systems, C programming language, Object Oriented Programming (C++), Data Structures, Microprocessors and Microcontrollers, Embedded Systems.

PROGRAMMING SKILLS

C, R, Python, Perl, MATLAB; also had some exposure to Assembly Language, C++, BASIC, and PASCAL. Platforms used: Linux, Windows

TEACHING & REVIEWING EXPERIENCES

Teaching experiences

Teaching assistant for Advanced Algorithms, Computational Molecular Biology, Statistical Signal Processing and Applications. Taught one month graduate course “Short Course on Automatic Speech Recognition” at IISc.

Reviewing experiences

Reviewer for RECOMB 2012, IEEE Journal of Biomedical and Health Informatics.

Students supervised

- Ana Manasovska (intern at EPFL), Bachelor’s student at Faculty of Computer Science and Engineering (FCSE), Saints Cyril and Methodius University of Skopje, Skopje, Macedonia.
- Paulina Grnarova, Master’s student, School of Computer and Communication Sciences, EPFL.
- Jelena Antic, Master’s student, School of Computer and Communication Sciences, EPFL.
- Laura Hunter, Master’s student, Stanford University, USA.

OTHER ACTIVITIES

Vice President and core team member of YUVA-Indians@EPFL/UNIL, the official Indian student association of EPFL (January 2013 — February 2014).