



# UCL

# Bayesian nonparametric models of genetic variation

Lloyd T. Elliott

B.Math., Pure Mathematics, University of Waterloo, Canada (2007)

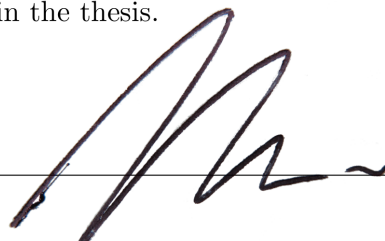
**Gatsby Computational Neuroscience Unit**  
**University College London**  
17 Queen Square, WC1N 3AR  
London, United Kingdom

THESIS

Submitted for the degree of  
**Doctor of Philosophy, University of London**

2016

I, Lloyd Thomas Elliott, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.



---

## Abstract

We will develop three new Bayesian nonparametric models for genetic variation. These models are all dynamic-clustering approximations of the ancestral recombination graph (or ARG), a structure that fully describes the genetic history of a population. Due to its complexity, efficient inference for the ARG is not possible. However, different aspects of the ARG can be captured by the approximations discussed in our work. The ARG can be described by a tree valued HMM where the trees vary along the genetic sequence. Many modern models of genetic variation proceed by approximating these trees with (often finite) clusterings. We will consider Bayesian nonparametric priors for the clustering, thereby providing nonparametric generalizations of these models and avoiding problems with model selection and label switching.

Further, we will compare the performance of these models on a wide selection of inference problems in genetics such as phasing, imputation, genome wide association and admixture or bottleneck discovery. These experiments should provide a common testing ground on which the different approximations inherent in modern genetic models can be compared. The results of these experiments should shed light on the nature of the approximations and guide future application of these models.

---

## Acknowledgments

I was extremely lucky to have Yee Whye Teh as a PhD supervisor. His love of science, intuition in machine learning, careful attention and confidence in his students has made my time at Gatsby both inspiring and productive.

I would also like to thank all of the other supervisors I have had: Jonathan Marchini, Wolfgang Lehrach, John-Dylan Haynes, Chris Eliasmith and Paul Thagard. I am particularly thankful towards Anna Goldenberg for supporting my internship at SickKids hospital in Toronto and to William and Michael Andregg for supporting my internship at Halcyon Molecular in Redwood City. I would also like to thank all of my collaborators, with whom I have been extending the work described in this thesis: Barbara Engelhardt, Derek Aguiar, Maria De Iorio and Jan-Willem van der Meet.

During my time at Gatsby, I had the pleasure of having among my friends many inspiring academics: Agnieszka, Andriy, Arthur, Balaji, Ben, Bharath, Biljana, David B, David S, Demis, Dilan, Dino, Heiko, Jan, Jeff, Joana, Kai, Kristin, Lars, Laurence, Loïc, Mani, Maria, Marius, Mijung, Nicolas, Pedro, Phillipp, Ross, Sam, Shane, Srin, Ulrik, Vincent, Zoltan and also the original members of Alexandra House, room 503B: Vinayak and Charles. The level of science at Gatsby is superb and I loved all of the time I spent there. For this, I would like to thank the exceptional Gatsby faculty: Arthur Gretton, Maneesh Sahani, Peter Latham and the director: Peter Dayan.

I would also like to thank all of the new friends I made during my time as a student: Abi, Abninder, Anna, Andy, Anne, Brad, Cameron, Chris, Dan, Dana, Dylan, Emilie, Ernst, Ewan, Gail, Guy, Iain, Irina, James, Jean-Frédéric, Jenni, Jennifer, Jeremy, Liz, Maasa, Mark, Martin, Mary, Maureen, Nando, Owen, Pierre, Ricardo, Tatiana and anyone else I'll be sorry I've forgotten. I would also like to thank friends from my hometown Kingston, Ontario: David, Fiona, James, Jamie, J.C., Jeff, Jordan, Matt K, Matt L, Pat, Paul, Randall and Steve. I would also like to make room for the loving memory of D'Arcy Bernier.

Finally, I would like to thank my family for being so wonderful: Caitlyn, Dane, the Fels Elliotts, George & Noriko, Grandmamargaret, Iris, the Jarvises, the Junior Baileys and the Peltons. And above all else, I would like to thank my parents: Bruce and Janet Elliott, for their boundless support, love and patience. I am forever grateful to them.

The work in this thesis was supported by the Gatsby Charitable Foundation and by NSERC grant 312864.

# Contents

## Front matter

|                              |    |
|------------------------------|----|
| Abstract . . . . .           | 3  |
| Acknowledgments . . . . .    | 4  |
| Contents . . . . .           | 5  |
| List of figures . . . . .    | 8  |
| List of tables . . . . .     | 10 |
| List of algorithms . . . . . | 11 |

## 1 Introduction 12

|   |    |
|---|----|
| 1.1 Data types and problems in statistical genetics . . . . .                 | 14 |
| 1.1.1 Phased data . . . . .   | 14 |
| 1.1.2 Imputation . . . . .  | 16 |
| 1.2 Review of statistical genetics and related work . . . . .                 | 17 |
| 1.2.1 The coalescent and the ancestral recombination graph . . . . .          | 19 |
| 1.2.2 Mutation models . . . . .   | 20 |
| 1.2.3 Assumptions for the coalescent with recombination . . . . .             | 21 |
| 1.2.4 Inference and approximations . . . . .                                  | 21 |
| 1.2.5 Approximating the SMC with dynamic-clustering . . . . .                 | 24 |
| 1.2.6 The product of approximate conditionals . . . . .                       | 24 |
| 1.2.7 Classification of HMMs in statistical genetics . . . . .                | 26 |
| 1.3 Contributions of this thesis . . . . .                                    | 28 |
| 1.3.1 Introduction to Bayesian nonparametrics . . . . .                       | 29 |
| 1.3.2 Bayesian nonparametric models of genetic variation: a preview . . . . . | 31 |
| 1.3.2.1 The Bayesian nonparametric version of <b>fastPHASE</b> . . . . .      | 31 |
| 1.3.2.2 The discrete fragmentation-coagulation process . . . . .              | 32 |
| 1.3.2.3 The Wright-Fisher partition valued process . . . . .                  | 33 |

## 2 Bayesian nonparametrics and dynamic-clustering 35

|   |    |
|---|----|
| 2.1 Introduction . . . . .  | 35 |
| 2.2 The Dirichlet process through measures, partitions and sequential schemes <span style="float: right;">37</span> |    |
| 2.2.1 Ewens' sampling formula and random partitions . . . . .   | 39 |
| 2.2.2 The CRP through a sequential scheme . . . . .   | 40 |
| 2.3 The hierarchical Dirichlet process . . . . .  | 40 |

|          |   |           |
|----------|---|-----------|
| 2.4      | Fragmentation and coagulation operators . . . . .   | 41        |
| 2.4.1    | Conditionals for fragmentation and coagulation operators . . . . .                                  | 43        |
| 2.4.2    | Conditionals for clustering a single item . . . . .   | 44        |
| 2.5      | Summary . . . . .   | 45        |
| <b>3</b> | <b>The Bayesian nonparametric version of fastPHASE</b>  | <b>46</b> |
| 3.1      | Introduction . . . . .  | 46        |
| 3.1.1    | Population bottlenecks and genetic sequence data . . . . .  | 48        |
| 3.1.2    | Intuition for the BNPPHASE model . . . . .  | 48        |
| 3.1.3    | Likelihood of phased data under the BNPPHASE model . . . . .  | 51        |
| 3.1.4    | Inference for the BNPPHASE model . . . . .  | 51        |
| 3.2      | Methods . . . . .   | 52        |
| 3.2.1    | Generative process for the BNPPHASE model from stick breaking . . . . .                             | 52        |
| 3.2.2    | Generative process for the BNPPHASE model from partitions . . . . .                                 | 54        |
| 3.2.3    | The hierarchical Dirichlet process through partitions . . . . .                                     | 54        |
| 3.2.4    | Marginalizing the allele emission variables $\theta$ . . . . .                                      | 59        |
| 3.2.5    | MCMC for inference and imputation . . . . .   | 59        |
| 3.2.5.1  | Gibbs update for latent cluster assignment of sequence $i$ . . . . .                                | 60        |
| 3.2.5.2  | Gibbs updates for HDP parameters $\tilde{\omega}$ and $\varphi$ . . . . .                           | 65        |
| 3.2.5.3  | Slice sampling for parameters $\alpha_0, \alpha, \gamma_\ell, \beta_\ell, b$ and $r_\ell$ . . . . . | 66        |
| 3.2.6    | Summary . . . . .   | 67        |
| 3.3      | Relationship to the FastPHASE model . . . . .   | 68        |
| 3.3.1    | Finite truncations of the BNPPHASE model . . . . .  | 68        |
| 3.3.2    | Non-reversibility of fastPHASE and related models . . . . .   | 68        |
| 3.4      | Experiments . . . . .   | 69        |
| 3.4.1    | MCMC initialization, burn-in, iteration, restarts and schedules . . . . .                           | 70        |
| 3.5      | Results . . . . .   | 71        |
| 3.5.1    | Results I: simulated data . . . . .   | 71        |
| 3.5.1.1  | Imputation of bottleneck with identity-by-descent . . . . .   | 71        |
| 3.5.1.2  | Examination of runtime . . . . .  | 72        |
| 3.5.2    | Results II: TMRCA regression on the out-of-Africa bottleneck . . . . .                              | 72        |
| 3.5.3    | Results III: imputation of male X chromosome data . . . . .   | 73        |
| 3.6      | Discussion . . . . .  | 75        |
| 3.6.1    | Intuition for TMRCA regression results . . . . .  | 78        |
| 3.7      | Conclusion . . . . .  | 78        |
| <b>4</b> | <b>The discrete fragmentation-coagulation processes</b>   | <b>81</b> |
| 4.1      | Introduction . . . . .  | 81        |
| 4.1.1    | Relation to the genetic process . . . . .   | 83        |
| 4.1.2    | Definition of the DFCP through fragmentation and coagulation . . . . .                              | 84        |
| 4.1.3    | Relation to the CFCP . . . . .  | 85        |
| 4.2      | Methods . . . . .   | 86        |

|          |   |            |
|----------|---|------------|
| 4.2.1    | Likelihood model and parameter priors . . . . .                                 | 86         |
| 4.2.2    | Joint probability distribution for DFCEP . . . . .                              | 89         |
| 4.2.3    | Gibbs update for latent block assignment of sequence $i$ . . . . .              | 89         |
| 4.2.4    | Slice sampling for parameters $\alpha$ , $d_\ell$ , and $\gamma_\ell$ . . . . . | 92         |
| 4.2.5    | Genotype imputation for unphased data . . . . .                                 | 93         |
| 4.2.6    | Phasing . . . . .   | 97         |
| 4.2.7    | The length of a haplotype . . . . .   | 97         |
| 4.3      | Experiments . . . . .   | 104        |
| 4.4      | Results . . . . .   | 106        |
| 4.5      | Discussion . . . . .  | 107        |
| 4.6      | Conclusion . . . . .  | 107        |
| <b>5</b> | <b>The Wright-Fisher partition valued processes</b>                             | <b>109</b> |
| 5.1      | Introduction . . . . .  | 109        |
| 5.1.1    | Relation to work in genetics . . . . .  | 110        |
| 5.2      | Methods . . . . .   | 111        |
| 5.2.1    | Generative process for the Wright-Fisher partition valued diffusion             | 111        |
| 5.2.2    | Likelihoods for voting data . . . . .   | 113        |
| 5.2.3    | Relation to time-varying generalized urn schemes . . . . .                      | 114        |
| 5.2.4    | Probabilistic programming and inference . . . . .                               | 115        |
| 5.2.5    | Describing $\mathcal{R}_t$ as a partition valued process . . . . .              | 117        |
| 5.3      | Experiments . . . . .   | 118        |
| 5.3.1    | Experiment I: bloc discovery . . . . .  | 118        |
| 5.3.2    | Experiment II: vote prediction . . . . .  | 118        |
| 5.4      | Results . . . . .   | 120        |
| 5.5      | Discussion . . . . .  | 121        |
| 5.6      | Conclusions . . . . .   | 122        |
| <b>6</b> | <b>Conclusions and future work</b>  | <b>124</b> |
| 6.1      | Conclusions . . . . .   | 124        |
| 6.2      | Future work . . . . .   | 125        |
|          | <b>References</b>   | <b>126</b> |

# List of figures

|      |   |     |
|------|---|-----|
| 1.1  | Representation of data from the Thousand Genomes Project Consortium   | 16  |
| 1.2  | Example of a draw from Kingman’s coalescent                           | 18  |
| 1.3  | Worked example of a sample from the sequentially Markov coalescent    | 23  |
| 1.4  | Genetic similarity is a function of chromosome location               | 29  |
| 1.5  | Haplotype structure of the CEU and YRI populations from HapMap        | 32  |
| 2.1  | Plate diagram for the LDA model                                       | 36  |
| 2.2  | Example clustering $\mathcal{R}$ of the set $R = \{1, \dots, 7\}$     | 39  |
| 3.1  | Genealogy of 6 homologous sequences with simulated ancestry           | 49  |
| 3.2  | Genealogy of 5 genetic sequences with simulated ancestry              | 49  |
| 3.3  | Plate diagram for entire BNPPHASE model                               | 53  |
| 3.4  | Plate diagram for hierarchical likelihood used by BNPPHASE model      | 54  |
| 3.5  | Stick breaking construction for Dirichlet process                     | 55  |
| 3.6  | Plate diagram for marginalized version of BNPPHASE model              | 57  |
| 3.7  | Relationship between partition structure and dynamic clustering       | 58  |
| 3.8  | Simulated ‘toy’ data using the identity-by-descent paradigm           | 69  |
| 3.9  | Imputation on simulated (‘toy’) identity-by-descent data              | 72  |
| 3.10 | Scalability of BNPPHASE   | 73  |
| 3.11 | Regression of TMRCA against number of clusters                        | 74  |
| 3.12 | Example region of male X chromosomes                                  | 76  |
| 3.13 | Imputation accuracy for X chromosomes                                 | 77  |
| 3.14 | Number of unique haplotypes and TMRCA along chromosome                | 78  |
| 4.1  | Mosaic structure found by the FCP                                     | 82  |
| 4.2  | Plate diagram for the DFCP  | 88  |
| 4.3  | Probability of extending a haplotype                                  | 101 |
| 4.4  | Approximation for haplotype lengths                                   | 102 |
| 4.5  | Allele imputation for X chromosomes from the Thousand Genomes project | 105 |
| 4.6  | Runtimes per iteration per sequence for FCP                           | 106 |
| 5.1  | A sample from the WFP prior   | 111 |
| 5.2  | Probability for partitions sampled according to the WFP               | 117 |



---

|     |  |     |
|-----|--|-----|
| 5.3 | Cases for transitions . . . . .  | 119 |
| 5.4 | Composition of the clusters found by the <b>WFP</b> model . . . . .        | 121 |
| 5.5 | Band for the bipartition found by clustering MPs into two clusters . . . . | 122 |

# List of tables

|     |  |     |
|-----|--|-----|
| 3.1 | RMSE for regression of TMRCA against # of clusters . . . . . | 73  |
| 5.1 | Percent correct for vote predictions. . . . .                | 121 |

# List of algorithms

|     |   |     |
|-----|---|-----|
| 3.1 | Retrospective stick breaking for the <b>BNPPHASE</b> model . . . . .          | 64  |
| 4.1 | Computation of expected haplotype lengths for the <b>DFCP</b> model . . . . . | 100 |

# Chapter 1

## Introduction

The cost per basepair of DNA sequencing is rapidly decreasing ([Wetterstrand, 2014](#)) allowing large volumes of genetic sequence data to be collected by academic consortiums, corporations and hospitals. Along with this increase in the availability of genetic sequence data is a need for modern machine learning methods tailored to specific problems in genetics. Such problems include disease association, inference of demographic history and inference of properties such as recombination rates and mutation rates. The scientific, economic and health benefits that could be derived from effective solutions to these problems are clear. But the effectiveness of any solution to such problems relies on the accuracy of the underlying statistical models used to describe genetic sequence data. Bayesian nonparametric methods are modern machine learning methods which, due to their flexibility, provide efficient and accurate statistical models with many properties that are well suited for describing genetic sequence data ([Teh et al., 2006](#); [Xing et al., 2006, 2007](#); [Sohn and Xing, 2007](#); [Airoldi et al., 2006](#); [Xing and Sohn, 2007b](#); [Sohn and Xing, 2007](#)).

All genetic material arises through inheritance and mutation. Random processes can be used to describe both of these phenomena: inherited material is governed by recombination and natural selection, whereas mutated material is governed by a variety of random processes such as single nucleotide variations (SNVs), copy number variation and other processes ([Hein et al., 2005](#)). However, two main concerns prevent us from fully characterizing the joint distribution of a set of genetic sequences. First, although we know the form of most of the random processes required to describe inheritance and mutation, we remain uncertain about many of the parameters involved in the processes such as their rates. Second, even if the parameters were known, the complexity of the latent objects involved (such as the taxons or the ancestral recombination graphs) often precludes efficient inference. Because of this, researchers in statistical genetics often make simplifying assumptions about the latent objects and parameters of the genetic processes and provide approximate models so that tractable inference can proceed.

Bayesian nonparametric models allow prior distributions to be specified in which the

---

complexity of the latent objects is arbitrary and learned along with the model parameters during inference (Hjort et al. 2010, Orbanz and Teh 2010, Teh 2010). These prior distributions are attractive choices for approximate models of genetic processes because efficient and accurate inference can be conducted while still providing complex latent objects. Further, many aspects of genetic processes, such as allele or species sampling formula (Ewens, 1972) naturally arise from these priors.

In this thesis, we will present three new closely related dynamic-clustering models for sequence data based on Bayesian nonparametric methods and explore their application to problems in genetics through a series of experiments. In these models, the genetic sequences are clustered separately into genetically similar clusters at each location of interest on the chromosome according to a Bayesian nonparametric joint distribution on partitions. Our approach extends other traditional methods in dependent Dirichlet processes in Bayesian nonparametrics such as those based on MacEachern (1999).

Dynamic-clustering models have many diverse applications beyond genetics. In machine learning, these models have been used as topic models for documents and also to describe social networks, geopolitical organization and the formation of political blocs in affairs of state (examples of such applications can be found for example in Blei and Lafferty 2007, Kemp et al. 2006 and Friggeri 2012). The models developed in this thesis can also be applied to these diverse domains. In a short departure from the main application of this thesis, in order to show the versatility of these methods, we will use one of the Bayesian dynamic-clustering models presented in this thesis to describe the voting behavior of Members of Parliament in the Canadian House of Commons (this is done in Chapter 5). We apply our model to detect when Members of Parliament cross the floor (*i.e.*, switch parties) and also to predict the voting behavior of Members of Parliament.

In the remainder of Chapter 1, we will summarize the contributions of this thesis and then provide a review of relevant background and related methods in statistical genetics. In section 1.1, we will provide a description of the sources of data that are relevant for Bayesian nonparametric haplotype models (haplotypes are patterns of mutations that are inherited together). In section 1.2 we will review the coalescent with recombination and the genetic basis for its assumptions and approximations. Further, we will describe other popular hidden Markov models for dynamic-clustering in genetics and their relation to the three new methods presented in this thesis and we will provide a new unified view of these models through the classification of their transition matrices. In section 1.3 we will introduce Bayesian nonparametrics and then we will provide intuition for the three new dynamic-clustering models presented in this thesis, and we will preview the experiments and baselines that we will use in later Chapters to explore these models.

In Chapter 2, we review dynamic-clustering models and Bayesian nonparametric methods and provide a unified framework for this theory using random partitions and

hierarchical Dirichlet processes. In Chapters 3 and 4, we present BNPPHASE (Elliott and Teh, 2015), the Bayesian nonparametric version of the fastPHASE (Scheet and Stephens, 2006) model, and the discrete fragmentation-coagulation process (or DFCP Elliott and Teh 2012). In Chapter 5, we present the Wright-Fisher partition valued process. These three models constitute our three new Bayesian nonparametric models of genetic sequence data (and other sorts of data, such as voting data) based on dynamic-clustering. We compare these three models with some established parametric models such as BEAGLE (Browning and Browning, 2009), fastPHASE (Scheet and Stephens, 2006), IMPUTE/IMPUTE2 (Marchini et al., 2007; Howie et al., 2009) and a method based on collaborative filtering (Salakhutdinov and Mnih, 2007). We explore the posterior distributions of these models, develop some of their asymptotic properties, and highlight their advantages through a series of experiments in which data from The 1000 Genomes Project Consortium (2010), The International HapMap Consortium (2003) and data from simulations are considered. Finally, in Chapter 6 we conclude and outline programs for future work.

## 1.1 Data types and problems in statistical genetics

In this section, we review the types and sources of data that we will use in the experiments described in later Chapters.

### 1.1.1 Phased data

Humans are diploid organisms and therefore if a biallelic marker (*i.e.* location on the chromosome at which genetic material can occur in two forms) is observed then the minor allele (*i.e.* the less common form) would occur 0, 1 or 2 times in each individual. These values correspond to genotypes consisting of a homozygous major allele, a heterozygous allele or a homozygous minor allele, respectively. An example of a biallelic marker is a single nucleotide polymorphism (SNP): a location at which a mutation occurring in the ancestry of the population has resulted in two possible DNA basepairs that can be observed at the location. When multiple SNPs are observed, it is often important to know from which of the two copies of the chromosome the minor alleles originates. This information is essential for a description of the haplotype structure of the population (Daly et al., 2001). If an individual is found to be heterozygous at two SNPs at locations A and B, then the minor alleles can be ordered in two ways:

1. One chromosome could have the minor allele at SNPs A and B and the other chromosome could have the major allele at SNPs A and B (the chromosomes are ‘11’ and ‘00’).
2. One chromosome could have the minor allele at SNP A and the major allele at SNP B and the other chromosome could have the major allele at SNP B (the

chromosomes are ‘01’ and ‘10’).

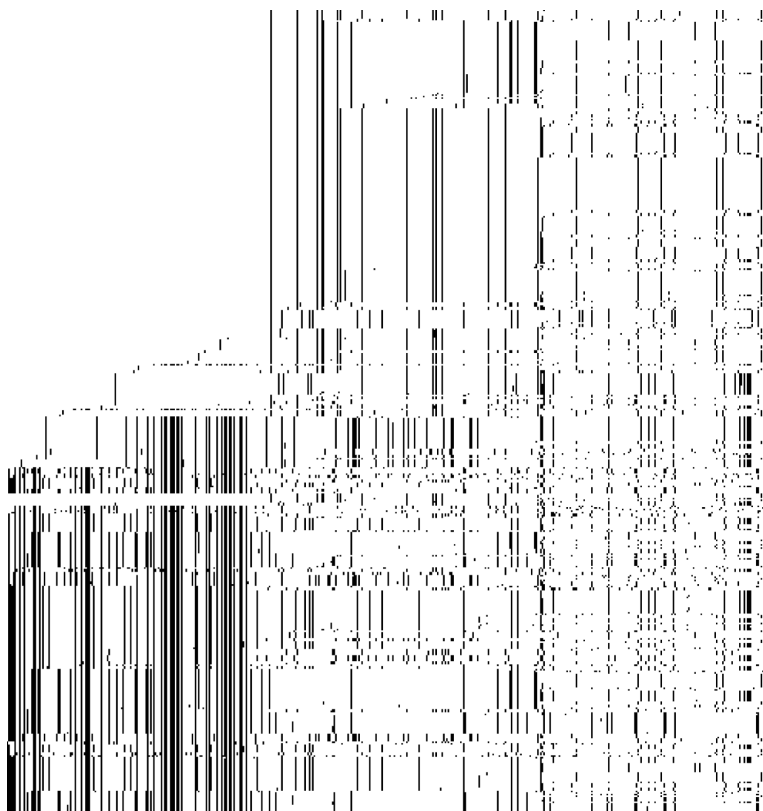
Genetic sequence data which includes the ordering of the heterozygous alleles is referred to as phased data. For the experiments conducted in this thesis, we will consider three sources of phased data:

1. *Phased trio data.* If a diploid individual is sequenced and both of the individual’s parents are also sequenced, then the chromosome from which an allele originates for that individual can be determined for every location at which the three individuals are not all heterozygous. Thus, trio data provides a source of phased data. The proportion of sites that can be phased this way for a trio depends on the expected minor allele frequency of the sample. Assuming Mendelian inheritance and Hardy-Weinberg equilibrium (we refer to [Hein et al. 2005](#) for an explanation of these conditions), this proportion is simply  $p$  where  $p$  is the minor allele frequency at that site.
2. *Male X chromosome data.* In humans, males have one copy of the X chromosome. Although some of the X chromosome is homologous to the Y chromosome (the pseudoautosomal regions), by omitting these regions phased data can be formed. Since the X chromosome undergoes meiotic recombination in females, the male X chromosome is a good model for the other 22 chromosome. An example of male X chromosome data from [The 1000 Genomes Project Consortium \(2010\)](#) is presented in [Figure 1.1](#).
3. *Simulated data.* Data simulated from the ARG provides phased information, as all aspects of the process can be recorded during simulation.

Presently most DNA sequencing methods are unable to determine the ordering of the minor alleles. These data are unphased data and are prevalent due to the currently prohibitive cost of DNA sequencing methods based on chromosome sorting ([Yang et al., 2011](#)) or imaging ([Payne et al., 2013](#)). In unphased data, the observation of the two copies of a chromosome for a diploid individual are represented by a sequence of unordered pairs of alleles. Phasing is the process of ordering the alleles within each pair so that the pattern of alleles for one chromosome is given by the first coordinate of the pairs and the pattern of alleles for the other chromosome is given by the second coordinate of the pairs.

We will often focus on phased rather than unphased data in this thesis for two reasons. Firstly, phased data is simpler to model. As Bayesian nonparametric models are already quite complicated, we will focus on their detailed description for phased data. Their extension to unphased data will often be clear. Secondly, we expect that in the future the cost of sequencing methods that provide phased data will decrease and *in silico* phasing will become obsolete.

We have found that the accuracy of imputation tasks performed on phased data is highly correlated with the accuracy of similar tasks on unphased data. Therefore, analysis of



**Figure 1.1:** Representation of data from [The 1000 Genomes Project Consortium \(2010\)](#).  $x$ -axis indicates chromosome position,  $y$ -axis indicates sequence identity. Rows (which are exchangeable) sorted lexicographically from left according to allele pattern: each row corresponds to an individual's haplotype and each column corresponds to a marker. White indicates major allele, black indicates minor allele. The ordering of the rows is chosen by a left-aligned lexicographical sorting (*i.e.*, individuals with minor alleles in the first marker are ordered first, and if the first  $\ell$  markers of two individuals match, the individual with a minor allele at marker  $\ell + 1$  is ordered first).

phased data provides a simple framework for comparison; this analysis carries over to other more complicated models.

### 1.1.2 Imputation

Assume we are given  $N$  phased chromosomes observed at  $L$  possible locations. Assume further that the  $L$  locations are biallelic markers (*i.e.*, the  $L$  locations correspond to mutations that occur in only two forms such as SNPs or SNVs). Imputation tasks involve predicting the alleles on each chromosome that are unobserved at some subset of the  $L$  locations. Imputation is required in study/reference paradigms (explained below) and situations in which the observation of genetic material is noisy. Imputation is also useful to assess the accuracy of a model. In this last case, often a hold out condition will be considered. This hold out condition can be designed to emulate a study/reference paradigm or uniform or location-biased noise. After imputation,



models can be assessed by comparing the imputation accuracy of each model on the held out data.

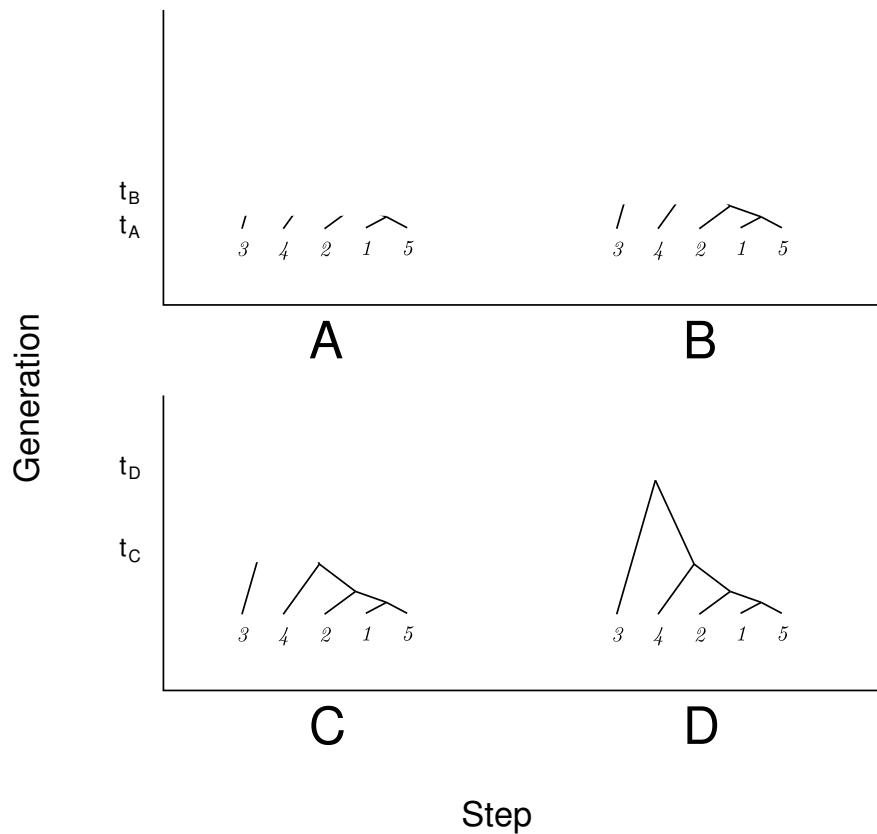
We will denote by  $x_{i\ell}$  the allele of the  $i$ -th individual at the  $\ell$ -th marker. Since the markers are biallelic,  $x$  is a 0/1 matrix (*i.e.*, each entry is either zero or one). Thus,  $x_{i\ell} = 1$  means that the  $i$ -th individual has the minor allele at location  $\ell$  whereas  $x_{i\ell} = 0$  means that the  $i$ -th individual has the major allele at location  $\ell$ . Furthermore, we will indicate the case where the  $\ell$ -th marker is not observed for individual  $i$  by the notation  $x_{i\ell} = '?'$ . We will refer to the set of unobserved entries of  $x$  by  $x_{\text{HID}}$  and the set of observed entries by  $x_{\text{OBS}}$ . So,  $x_{\text{HID}} = \{(i, \ell) : x_{i\ell} = '?'\}$  and  $x_{\text{OBS}} = \{(i, \ell) : x_{i\ell} = 0 \text{ or } 1\}$ . Thus, the goal of imputation is to describe  $\Pr(x_{\text{HID}}|x_{\text{OBS}})$ .

In study/reference paradigms, study chromosomes are typed at a small number of locations and reference chromosomes are typed at all  $L$  locations. This situation occurs frequently as a preprocessing step in genome wide association studies in which limited resources lead to sparse observations of the genetic sequences of study participants. Study power can be gained by registering these study individuals against publicly available reference panels ([The Wellcome Trust Case Control Consortium, 2007](#); [Marchini and Howie, 2010](#)). In this case, if we have  $N_S$  study individuals and  $N_R$  reference individuals, and the study individuals are observed only at  $\{\ell_1, \dots, \ell_L\} \subseteq \{1, \dots, L\}$  then  $x_{\text{HID}} = \{(i, \ell) : 1 \leq i \leq N_S, \ell \in \{\ell_1, \dots, \ell_L\}\}$ . We refer to [Browning and Browning \(2011\)](#) for a review of imputation methods and their application to association studies and study/reference paradigms.

In uniform noise conditions, inclusion of  $(i, \ell)$  in  $x_{\text{HID}}$  occurs independently with probability  $p$  for each pair  $(i, \ell)$ . Finally, in biased noise conditions, inclusion of  $(i, \ell)$  occurs independently with a probability that is a function of the minor allele frequency at location  $\ell$ . Usually, alleles with small minor allele frequency exhibit more uncertainty in observation. In this thesis, we will mainly consider imputation tasks with uniform noise conditions.

## 1.2 Review of statistical genetics and related work

Suppose that  $N$  genetic sequences from a population are observed. Most of the genetic material at a fixed chromosome location will be identical across all of  $N$  sequences. This is due to the shared ancestry of the population. Differences in the material will only be present at locations for which a mutation has occurred more recently than the most recent common ancestor of the sample. In the remainder of this Chapter, we will give an overview of the statistics governing the joint distribution of the pattern of mutations in the sample of  $N$  genetic sequences.



**Figure 1.2:** Example of a draw from Kingman's coalescent with  $N=5$  sequences. The  $x$ -axis indicates sequence identity and the  $y$ -axis indicates time in generations (with most recent lineages below and most ancient lineages above). All sequences coalesce in  $N-1=4$  events. Sequences 1 and 5 have a common ancestor at time  $t_A$ . Sequences 1, 5 and 2 have a common ancestor at time  $t_B$ , and so it continues in this fashion until all sequences coalesce. The intensity of this genealogy is found using Kingman's coalescent as follows: in step A, coalescence is found at time  $t_A$  and between times 0 and  $t_A$  the coalescent rate is  $\binom{5}{2}=10$ , and so intensity of first step is  $10 \cdot \exp(-10t_A)$ . In step B, coalescence is found at time  $t_B$  and between times  $t_A$  and  $t_B$  the coalescent rate is  $\binom{4}{2}=6$ , and so intensity of second step is  $6 \exp(-6(t_B - t_A))$ . Intensity of the entire process is the product of intensities for each step, yielding  $180 \cdot \exp(-4t_A - 3t_B - 2t_C - t_D)$ .

### 1.2.1 The coalescent and the ancestral recombination graph

Under mild genetic assumptions (which are briefly discussed in the next subsection), the ancestry of a haploid population (*i.e.*, a population of organisms in which there is only one copy of each chromosome per cell and no recombination) is given by Kingman's coalescent (Kingman, 1982). This process is a prior on genealogies formed by tracing the lineage of the  $N$  sequences backwards in time and placing coalescent events with rate  $\frac{2}{N_e} \binom{k(t)}{2}$  where  $k(t)$  is the number of distinct lineages existing at time  $t$ , and  $N_e$  is the effective population size (which is proportional to the total number of individuals in the population). At each coalescent event, two lineages chosen uniformly among all pairs of lineages are combined into one lineage. This continues until all the lineages have coalesced into the most recent common ancestor. A worked example of this process is given in Figure 1.2. The parameter  $N_e$  governs the total rate of coalescence and is defined to be twice the expected time until two given lineages coalesce. Thus, as the effective population size increases, the rate at which the lineages coalesce decreases. This can be seen intuitively because the probability that two individuals share a recent ancestor increases as the size of the total population decreases.

To account for recombination events occurring in the ancestry of a diploid population (*i.e.*, a population of organisms that undergoes meiotic recombination and has two copies of most chromosomes per cell), Kingman's coalescent can be extended to form a model known as the coalescent with recombination (Hudson, 1983). In this extended model, the ancestry can be completely described by an ancestral recombination graph (abbreviated as ARG). In an ARG, a recombination process in which recombination events occur with rate  $\rho k(t)/2$  is superimposed on Kingman's coalescent. Here,  $\rho$  is a scaled recombination rate. At each recombination event, a lineage is chosen uniformly among all lineages and that lineage is split at a random point along the sequence to form two new ancestors for the lineage. All material to the left of the splitting point is inherited from one of the ancestors and all material to the right of the splitting point is inherited from the other ancestor. In this way, the coalescent with recombination can be simulated by tracing lineages backwards in time until a most recent common ancestor for the entire sample is reached.

In addition to this view of the coalescent with recombination as a process simulated backwards in time (*i.e.*, a Markov process whose axis is time), the coalescent with recombination can also be viewed as a spatially defined non-Markovian process that takes values in an augmented space of genealogies (Wiuf and Hein, 1999). In this spatial representation (which we will refer to as the spatial construction of the coalescent with recombination), the axis of the process is the chromosome location. Given a population of chromosomes, this process defines a genealogy at the first chromosome position, and then moves from left to right along the chromosome and updates the genealogy at ancestral recombination points. By superimposing the genealogies from each chromosome location, a structure is formed that is identical in interpretation to

the ancestral recombination graph. Further, generating an ARG by simulating lineages backwards in time yields the same distribution over graphs as does the superposition of the genealogies from the spatial construction of the coalescent with recombination. We refer to (Wiuf and Hein, 1999) for the details of the non-Markovian nature of the spatial construction.

### 1.2.2 Mutation models

Conditioned on the ancestral recombination graph, mutations can be modelled by specifying the ancestral time, lineage and chromosome location at which each mutation occurs. Then, the mutated lineage can be traced forward in time to arrive at observed genetic material. All observed genetic material that coalesces with the mutated lineage more recently than the time of the mutation will inherit the mutation.

Many models have been proposed for the joint distribution over the time, lineage, location and nature of mutations. The most simple and general model is the infinite sites model (Kimura and Crow, 1964). In this model, chromosome locations are indexed by the unit interval. The time, lineage and location of the mutations are modelled by a Poisson process with intensity given by  $\theta k(t) dt d\ell$ , where  $\theta$  is a mutation rate parameter,  $t$  is the time of the mutation,  $k(t)$  is the number of lineages at time  $t$ , and  $\ell$  is the chromosome location of the mutation: a mutation occurring at time  $t$  is placed at the chromosome location  $\ell \sim \text{Uniform}(0, 1)$  on a lineage chosen uniformly at random from all lineages existing at time  $t$ . Due to the nature of the uniform distribution, with probability 1 all mutations will occur at distinct locations. This model eliminates much of the complexity that arises from recurrent mutations, polyallelic sites, structure in mutation rates and natural selection. The assumptions underlying this model are further discussed in the next subsection.

In the posterior inference for genetic sequences considered in this thesis, we will always condition on a set of observed mutations. Therefore, although much work has been done to extend the infinite sites model to capture more aspects of the genetic process, the results of our inference procedures are agnostic about many aspects of the mutation model such as the joint distribution over the location, time and precise nature of the mutation. For example, in SNP data in humans it is known that the relative rate of mutation between the purine and pyrimidine classes (*i.e.*, the mutations  $A \leftrightarrow T$ , and  $G \leftrightarrow C$  between basepairs) are larger than the relative rate of mutation within the two classes (Felsenstein, 1981). However, in a study/reference genome wide association study, the bases for the alleles for each SNP are given, and so these relative rates do not affect posterior inference.

### 1.2.3 Assumptions for the coalescent with recombination

In order to derive Kingman’s coalescent we must adopt a neutral mutation model (*i.e.*, we assume that mutations do not affect fitness), and we must also assume that the effective population size is constant. In order to extend Kingman’s coalescent to ARGs, we must further adopt the assumptions of random mating (*i.e.*, we must assume that each pair of individuals is equally likely to have offspring) and uniform recombination rates along the chromosome. We will adopt all of these assumptions throughout this thesis with some exceptions. Firstly, instead of assuming uniform recombination, we will often consider location-varying recombination rates with arbitrary functional form. Secondly, we will sometimes consider inference in situations wherein the effective population size varies throughout the ancestry of the population (this is done with population bottlenecks in Chapter 3).

The extent to which these assumptions bias studies is controversial. For example, it has been argued that most mutations are either deleterious or have no effect on fitness (Kimura, 1983) and so the neutral mutation model could be accurate for the vast majority of observed mutations (according to Kimura 1983, if a mutation can be observed in a postfoetal organism, it was not deleterious). Also, random mating seems like a reasonable assumption to adopt for studies in which the data arise from a small number of unrelated individuals sampled from a large population. However, studies adopting the random mating assumption can be confounded if they involve large numbers of unrelated individuals, or individuals sampled from a small population, or if they involve chromosome regions which experience significant selective pressure. This is due to cryptic relatedness, a phenomenon which can lead to inflated false discovery rates in association studies (Voight and Pritchard, 2005). For more discussion about these assumptions we refer to Hein et al. (2005).

### 1.2.4 Inference and approximations

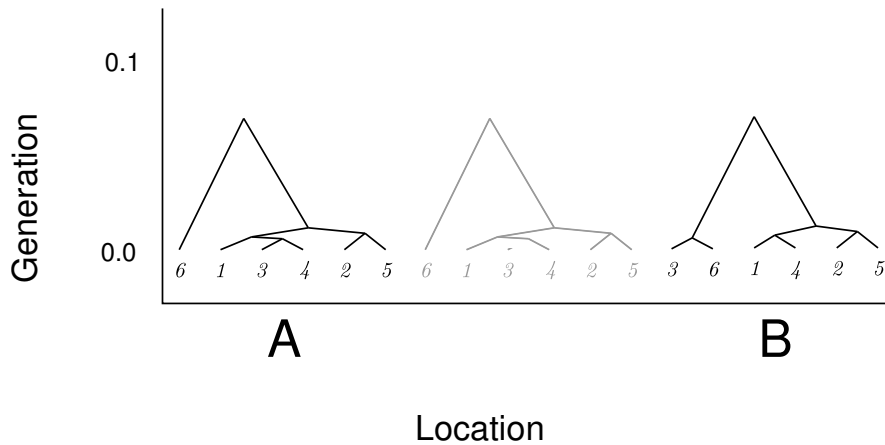
Inference based on ARGs and Kingman’s coalescent is difficult due to the combinatorial size of the latent spaces involved, the complicated dependence structures induced by recombination events and the lack of analytic forms for many of the posterior statistics involved in ARGs and coalescents (such as the recombination rates and effective population sizes). Despite these difficulties, inexact methods such as approximate Bayesian computation (Huelsenbeck and Ronquist, 2001), sequential Monte Carlo methods (Görür and Teh, 2009) and methods based on discretization of ARGs (Rasmussen et al., 2013) have been used. It is, however, unlikely that these methods could scale to large datasets consisting of thousands of genomes. For example, in Rasmussen et al. (2013) the authors apply their `argweaver` model to phase a dataset consisting of only 54 genomes provided by Complete Genomics and note that they could not scale their model to larger datasets. The computational complexity of `argweaver` and related

methods is  $\mathcal{O}(N^2L)$ , where  $N$  is the sample size and  $L$  is the number of markers.

Because of the difficulty inherent in conducting inference directly on ARGs, ARGs are often approximated by simpler processes and then inference is conducted using these simpler, approximate processes. One of the most successful approximations of the ARG is the sequentially Markov coalescent (McVean and Cardin, 2005). The sequentially Markov coalescent (abbreviated as SMC) takes as its starting point the spatial construction of the coalescent with recombination (Wiuf and Hein, 1999) described earlier in this section. The SMC relaxes the non-Markovian nature of the spatial construction by providing a Markovian version of the transition rules of the latent genealogy-valued process from (Wiuf and Hein, 1999). So, whereas the spatial statistics of the genetic process are not Markov along the chromosome, the SMC provides the ‘closest’ Markov version the genetic process. In McVean and Cardin (2005), the authors argue that not much is actually lost by the SMC approximation. In an experiment in which two sequences with 20 markers were simulated from either the SMC or the full spatial construction from Wiuf and Hein (1999), the pairwise correlation among the markers was found to be essentially the same for both cases. In other work, the estimates of the time to the most recent common ancestor of a marker found using the SMC was found to be quite accurate (Li and Durbin, 2011).

In the SMC, the genealogy of the observed genetic material is assumed to be governed by the following genealogy-valued Markov jump process (MJP). As before, we assume that we are constructing a genealogy of  $N$  genetic sequences. For the SMC, first, the latent genealogy of the  $N$  sequences at the left-most location of the chromosome is sampled from Kingman’s coalescent. Next, the MJP is simulated from left to right such that jump events occur with rate  $\rho T(\ell)/2$  where  $T(\ell)$  is the total branch length of the latent genealogy at location  $\ell$ . If a jump event occurs at location  $\ell$ , the latent genealogy is modified by drawing a point uniformly on the genealogy at  $\ell - \delta$  and then removing the edge that the point lies on. This partitions the genealogy into two sub-genealogies: a floating genealogy and a main genealogy. (The main genealogy is the sub-genealogy that coalesced more anciently.) The floating genealogy and the main genealogy are then coalesced to form the new genealogy at location  $\ell$ . This is done by extending the lineage of the floating genealogy backwards in time and coalescing it with a lineage chosen uniformly from the lineages of the main genealogy with rate  $k(t)\rho/2$  where, as for the definition of Kingman’s coalescent above,  $k(t)$  is the number of distinct lineages existing at time  $t$  in the main genealogy. (The new coalescent time of the two genealogies may be more ancient than the TMRCA of the main genealogy). A worked example of the intensity of a sample from the SMC is given in Figure 1.3. For a more detailed description of the SMC, we refer to Wiuf and Hein 1999.

Even though its definition is simple, inference based directly on the SMC, such as the genealogy-valued hidden Markov model (HMM) from Webb et al. (2009), is still unlikely to scale to large datasets. (In Webb et al. 2009 the authors applied their model to phase



**Figure 1.3:** Worked example of a sample from the sequentially Markov coalescent. Suppose that the SMC is simulated with  $N = 6$  sequences and the tree at the first location (A), the location of the first event (B), and the tree at the first event are all given as above. From section 1.2.1, the intensity of drawing the tree at A under Kingman's coalescent is  $2700 \cdot \exp(-5t_1 - 4t_2 - 3t_3 - 2t_4 - t_5)$  where  $t_1, \dots, t_5$  are the times of the coalescent events. The total size of the tree at A is  $6t_1 + 5(t_2 - t_1) + \dots + 2(t_5 - t_4) = T$ . The event rate of the MJP for the SMC after location A is  $\rho T/2$  and so the intensity of the first event is  $\rho T/2 \exp(-\rho T/2 \ell_B)$  where  $\ell_B$  is the distance from A to B. At B, a point is chosen on the edge connecting sequence 3 to its coalesce with sequence 4. Since the point is chosen uniformly, the intensity of that event is  $1/T$ . Finally, the floating genealogy consisting of the lineage of sequence 3 is coalesced with the main genealogy (this is the gray image between locations A and B in the plot). Since the floating genealogy coalesces before the first event of the main genealogy, this event occurs with intensity  $5 \exp(-5t_B)$  where  $t_B$  is the time to coalescence of sequence 3 at B. Thus, the intensity of the sample from the SMC shown in the plot is the product of these intensities:  $13500 \cdot \exp(-5t_1 - 4t_2 - 3t_3 - 2t_4 - t_5 - 5t_B)/T$ .

15 mouse genomes sequenced by Perlegen Sciences.) Instead, many recent models use the Markov assumption of the SMC and further simplify the latent space by considering partition-valued processes and HMMs instead of the genealogy-valued process. These models are discussed in the next subsection.

### 1.2.5 Approximating the SMC with dynamic-clustering

As we saw earlier in this section, the joint distribution governing genetic sequences sampled from a population can be approximated by a genealogy-valued Markov process that varies along the chromosome (this is known as the SMC approximation). Each sequence in the sample corresponds to a leaf in the genealogies (*i.e.*, a vertex at the bottom of the tree). At each location on the chromosome, the genetic similarity between each pair of sequences can be measured by taking the time until the material from the two sequences at that location coalesce, implied by the genealogy (*i.e.*, the hypermetric induced by the genealogy viewed as a tree). All sequences that are genetically similar with respect to the genealogy at a chromosome location have similar mutation patterns around that location.

Genealogies can be well approximated by partitions. A partition of a finite set  $S$  is a set of disjoint subsets (called blocks) of that set such that the blocks are nonempty and their union is all of  $S$ . For a given genealogy, we can induce a partition by choosing a time  $t$  and placing all elements that coalesce earlier than that time into the same block. In a similar way, a genealogy-valued process induces a dynamic-clustering (*i.e.*, a partition-valued process) by repeating this procedure at every location of the process. Note that the models we will discuss in this thesis do not operate by firstly inferring a genealogy and then forming the partitions induced by choosing a time  $t$  and partitioning the sequences based on their coalescence classes. Instead, this view of induced partitions serves as intuition about how approximating genealogy-valued processes by dynamic-clustering works.

### 1.2.6 The product of approximate conditionals

The SMC approximation, combined with this intuitive link between dynamic-clustering and genealogy-valued processes, has led to much research based on HMM approximations of the genetic process. The first such model proposed was the product of approximate conditionals (PAC) model (Li and Stephens, 2003). In the PAC model, each sequence is modelled as a composition of noisy copies of segments from the other sequences. The boundaries between the segments is governed by a transition rate  $c$ , which can depend on the chromosome location. In the construction of the PAC model, the sequences are indexed and each sequence is modelled in order, conditioned only on sequences with smaller indices.



The PAC model approximates the SMC by providing a simple HMM defined along the chromosome in which the genetic similarity between sequences is a function of chromosome location. Inference based on the PAC model has an  $\mathcal{O}(N^2L)$  complexity, which is relatively tractable when compared to the complexity of models based directly on the SMC or ARG. In addition, unlike the models which were developed before it (such as the composite likelihood model from [Fearnhead and Donnelly 2002](#)), the PAC model achieves this relatively tractable complexity while considering the joint distribution over all locations rather than just considering the joint distributions between pairs of locations.

Conditioned on the allele patterns of the  $N$  sequences, this generative process induces a posterior distribution on the rates  $c_\ell$  and the mutation rate  $\theta$ . Posterior inference about  $c_\ell$  and  $\theta$  can be done using MCMC and the forwards-backwards algorithm. The conditional distribution of  $c_\ell$  and  $\theta$  can thus be represented by samples, or the MAP of these parameters can be estimated ([Li and Stephens, 2003](#)).

Unfortunately, the construction of the PAC model does not lead to an exchangeable distribution—the distribution of the rates depends on the order in which the individuals are presented in the study. Furthermore, while tractable relative to inference based on the SMC or ARG, the  $\mathcal{O}(N^2L)$  complexity for inference based on the PAC model still precludes scalability to large studies.

The PAC model induces a dynamic-clustering on a collection of sequences. The PAC model requires that the order of these sequences be specified. The clustering is provided in a sequential scheme in which one individual is considered at a time ( $i = 1, 2, \dots$ ). There are  $i - 1$  possible clusters for each location of each sequence  $i > 1$ . For  $i = 2$ , each location is assigned to the first cluster. For  $i > 2$ , the first location of sequence  $i$  is assigned to cluster  $j < i$  with probability  $1/(i - 1)$ . Then, for each location  $\ell > 1$ , with probability  $c_\ell$  the cluster assignment of sequence  $i$  at location  $\ell$  is copied from the cluster assignment of sequence  $i$  at location  $\ell - 1$ , and with probability  $1 - c_\ell$  the cluster assignment of sequence  $i$  at location  $\ell$  is again assigned to cluster  $j < i$  with probability  $1/(i - 1)$ . Here,  $c_\ell$  captures the probability of breaks in the haplotype mosaic induced by ancestral recombination events.

This model, and the likelihood that relates it to the observed SNPs, is given formally by the generative process presented in the following enumeration:

1. The alleles for the biallelic locations on the first sequence are drawn uniformly from all  $2^L$  possible haplotypes.
2. For each sequence  $i$  such that  $1 < i \leq N$ :
  - (a) A Markov chain is drawn with  $i - 1$  states corresponding to the first  $i - 1$  sequences. The initial distribution of the Markov chain is uniform over the  $i - 1$  states. Then, between each consecutive pair of locations, with probability  $c_\ell/(i - 1)$  the Markov chain transitions to one of the other states

drawn uniformly from all  $i - 2$  other states. With probability  $1 - c_\ell$ , the Markov chain has a self transition (and the state stays the same).

- (b) For each location  $\ell$  such that  $1 \leq L$ :
- i. With probability  $\theta$ : the allele at location  $\ell$  for sequence  $i$  is set to be the same as the allele of the sequence corresponding to the Markov chain state at location  $\ell$ .
  - ii. Otherwise: the allele at location  $\ell$  for sequence  $i$  is set to 0 with probability  $1/2$  and to 1 with probability  $1/2$ .

This dynamic-clustering has a couple counter-intuitive properties. Firstly, because the clusters available to each sequence depends on the ordering of the sequences, the resulting dynamic-clustering is not an exchangeable distribution (this can be seen for example because sequence  $i > 1$  can only join clusters  $1, \dots, i - 1$ ). Secondly, the cluster assignment of the first sequence is undefined (instead of assigning the first sequence to clusters and then generating the alleles for the first sequence as an imperfect mosaic formed by those clusters, instead the alleles for the first sequence are drawn uniformly from all  $2^L$  possible haplotypes).

In [Li and Stephens 2003](#), the authors propose averaging over many random orderings of the sequences in order to overcome the limitations listed above. Many methods based on [Li and Stephens 2003](#) (such as the three methods we will present in this thesis) are designed to be exchangeable, mitigating the need for averaging over random orderings.

### 1.2.7 Classification of HMMs in statistical genetics

The limitations and counter intuitive properties of the PAC model have been addressed extensively by the HMM methods for genetic sequences developed over the past decade. In addition, these models have been extended to capture more advanced aspects of the genetic process such as population structure and relatedness.

We can classify all HMMs based on the PAC model broadly into three classes according to the nature of the transition matrices that their generative processes induce on the conditional state assignment of each sequence. Many of these models use a version of the transition rate  $c$  of the PAC model to regulate self-transitions (as in ‘sticky’ HMMs [Fox et al. 2011](#)) and haplotype lengths: in the prior, a sequence will transition with rate  $c$  and if a transition occurs, a new state is chosen with a probability specified by the model. If a transition does not occur, the next state of the item is a copy of the old state. The parameterization of that probability can involve latent parameters associated with the sequence identity ( $i$ ), or the chromosome location ( $\ell$ ), or both, or neither of these two indices.

1. *Location dependent models.* The first class of HMMs includes models for which the

transition matrices of the conditional state assignment of each sequence depend only on the chromosome location. This class contains **fastPHASE** (Scheet and Stephens, 2006), which builds on the PAC model by supposing that, rather than copying one of the  $i-1$  sequences that appear before it, the  $i$ -th sequence copies one of  $K$  latent, unobserved haplotypes. The prior probability in the **fastPHASE** model of copying the  $k$ -th latent haplotype given that a transition occurs is  $\pi_{\ell k}$ , where  $\pi_{\ell k}$  is the latent proportion of haplotype  $k$  at location  $\ell$ . The allele emission probabilities for each haplotype and the proportions  $\pi_{\ell k}$  are learned during inference. The **fastPHASE** model is exchangeable and the  $K$  latent haplotypes (rather than the observed sequences) provide centroids for the clusters. Other models in this class include **IMPUTE/IMPUTE2** (Marchini et al., 2007; Howie et al., 2009) and **SHAPEIT/SHAPEIT2** (Delaneau et al., 2012, 2013). In these models, the transitions at each location are parameterized by their location index within latent haplotype structures.

The prior transition and emission matrices (respectively) induced on the conditional state assignment of the  $i$ -th sequence under the **fastPHASE** model are as follows:

$$\begin{pmatrix} 1-c_\ell+c_\ell\pi_{\ell 1} & c_\ell\pi_{\ell 2} & \cdots & c_\ell\pi_{\ell K} \\ c_\ell\pi_{\ell 1} & 1-c_\ell+c_\ell\pi_{\ell 2} & \cdots & c_\ell\pi_{\ell K} \\ \vdots & \vdots & \ddots & \vdots \\ c_\ell\pi_{\ell 1} & c_\ell\pi_{\ell 2} & \cdots & 1-c_\ell+c_\ell\pi_{\ell K} \end{pmatrix}, \begin{pmatrix} \theta_{\ell 1} & \cdots & \theta_{\ell K} \\ 1-\theta_{\ell 1} & \cdots & 1-\theta_{\ell K} \end{pmatrix}. \quad (1.1)$$

From (1.1), we see that the off-diagonal elements of the transition matrix depend only on the location  $\ell$  and not the sequence index  $i$ . The **BNPPHASE** model that we will present in Chapter 3 is also contained in this first class of HMMs.

2. *Sequence dependent models.* For the second class of HMMs, the transition matrices depend only on the sequence identity (or, the sequence index). The most popular model in this class is the admixture model **STRUCTURE** (Pritchard et al., 2000; Falush et al., 2003), which, in its multilocus form has the following conditional transition and emission matrices:

$$\begin{pmatrix} 1-c_\ell+c_\ell\pi_{i1} & c_\ell\pi_{i1} & \cdots & c_\ell\pi_{i1} \\ c_\ell\pi_{i2} & 1-c_\ell+c_\ell\pi_{i2} & \cdots & c_\ell\pi_{i2} \\ \vdots & \vdots & \ddots & \vdots \\ c_\ell\pi_{iK} & c_\ell\pi_{iK} & \cdots & 1-c_\ell+c_\ell\pi_{iK} \end{pmatrix}, \begin{pmatrix} \theta_{\ell 1} & \cdots & \theta_{\ell K} \\ 1-\theta_{\ell 1} & \cdots & 1-\theta_{\ell K} \end{pmatrix}. \quad (1.2)$$

When the off-diagonal entries of the transition matrix in (1.2) are normalized (*i.e.*, when we condition on the event that a transition occurs) we see that the transition does not depend on the chromosome location  $\ell$  and instead depends on

only the sequence identity  $i$ . In admixtures, each individual inherits alleles from the admixed populations, but the proportion of alleles from each population in the admixture can vary from one individual to another due to genetic drift. This variance is captured by the **STRUCTURE** model and other models with sequence dependence. Recently, a Bayesian nonparametric version of **STRUCTURE** has been developed (De Iorio et al., 2015). In that work, the transition matrix (1.2) is extended to an HMM with infinitely many states.

3. *Models with neither location nor sequence dependence.* In this third class of HMMs, all of the structure in the genetic process is encoded directly in the state transition probabilities rather than in latent variables associated with individual sequences or chromosome locations. In the priors induced by HMMs from the first two classes, when transitions occur, the previous state of a sequence is ‘forgotten’ and a new state is chosen with either a location-specific or an individual-specific distribution.

This class includes homogeneous HMMs in which the transition matrix is a stochastic matrix (*i.e.*, there are no restrictions on the transition matrix other than that its columns sum to one). The HDP-HMM from Xing et al. (2006); Xing and Sohn (2007a) is of this form. Other models in this class include **BEAGLE** (Browning and Browning, 2009). The **DFCP** that we will present in Chapter 4 is also an example of this class. The **BEAGLE** software, like the **DFCP**, infers latent haplotype graphs that parsimoniously describe a population genetic sequence data. However, the **BEAGLE** model does this in an *ad-hoc*, non-Bayesian way. As a result, the **BEAGLE** model is not reversible or exchangeable.

4. *Location and sequence dependent models.* In the final class of HMMs, the normalized off-diagonal elements of the transition matrices depend on both the chromosome location and the sequence identity. These models can arise when the definitions of the first two classes of models are combined. For example, in Scheet and Stephens (2006), an extension to the **fastPHASE** model is considered for data collected from several subpopulations. The authors assume that each sequence is drawn from one of the subpopulation, and they allow the proportions  $\pi_\ell$  to vary among the subpopulations. In this case, (1.1) is extended by replacing  $\pi_{\ell k}$  with  $\pi_{s_i, \ell, k}$  where  $s_i$  is the subpopulation assignment of individual  $i$  (thus adding sequence dependence through  $s_i$ ).

### 1.3 Contributions of this thesis

Due to recombination events occurring in the ancestry of a population, similarity among genetic sequences in individuals is a function of chromosome location. Therefore, if at one end of a chromosome two sequences have identical patterns of mutations, at the

Sequence 1 CTACGATTA . . . TAATCGTAG

Sequence 2 CTACGATTA . . . TAATBTGTAG

rs13441248

rs1665289

**Figure 1.4:** Genetic similarity is a function of chromosome location. Sequences of material are from two different individuals. Sections from each end of Chromosome 1 are displayed (Karchin et al., 2005). Sections are neighborhoods of the first and last SNP on Chromosome 1 as reported in data from The International HapMap Consortium (2003). Red and blue indicate alleles. Grey indicates homologous material (*i.e.*, basepairs that are the same for all humans). At location rs13441248, the two sequences have the same alleles whereas at rs1665289 the two sequences have different alleles. Genetic processes leading to this sort of structure are explained in section 1.2.

other end of the chromosome the mutation patterns of the two sequences could be different from each other. See Figure 1.4 for an example involving single nucleotide polymorphism, or SNP, data (SNPs are defined in section 1.1). Hidden Markov models (HMMs) and chromosome painting models are commonly used to approximate this location dependent genetic similarity (Scheet and Stephens, 2006; Browning, 2006; Marchini et al., 2007; Delaneau et al., 2012, 2013). In such models, each genetic sequence is associated with a sequence of latent states. The states are clusters of locally-similar sequences: two sequences that share the same state at a given location have similar patterns of mutations around that location.

This work contributes two new HMMs for genetic similarity based on Bayesian non-parametric priors. In section 1.3.1 we provide the intuition behind Bayesian statistics and Bayesian nonparametrics. Then, in section 1.3.2 we preview the three models that constitute the contribution of this thesis.

In addition to the presentation and exploration of these models, we also contribute a derivation of the conditional distributions for random coagulation and fragmentation of partitions (Elliott and Teh, 2012). These conditionals are required for inference in the DFCP model.

### 1.3.1 Introduction to Bayesian nonparametrics

In Bayesian statistics, inference is performed by first placing a prior distribution on a model's parameter space. Then, after some data are observed, the posterior distribution of the parameters conditioned on the observed data is computed using Bayes rule. This can be done through Monte Carlo Markov chain simulation or in some cases through analytic calculation (*i.e.*, by solving the integral appearing in the denominator of Bayes rule). Alternatively, this can be done through sequential Monte Carlo (SMC)

or approximately through variational inference. Finally, using the posterior distribution, the model parameters can be estimated. For genetic data, this can provide insight into the processes governing the data such as recombination rates, mutation rates or time to most recent common ancestor (TMRCA). Estimation of missing data can also be provided by this framework, allowing imputation of noisy or missing genotype data in assays. Bayesian methods are standard in statistics and confer many benefits such as quantification of uncertainty and shrinkage (for a review of Bayesian methods see [Gelman and Meng 2004](#) or [Hjort et al. 2010](#)).

Bayesian nonparametric statistics were originally designed to provide priors with both large support and tractable posterior distributions ([Hjort et al., 2010](#); [Ferguson, 1973](#)). The bedrock of Bayesian nonparametric statistics is the Dirichlet process (DP). The DP can be thought of as a prior on the component weights of a mixture model with an infinite number of components. As such, the DP can be thought of as a generalization of the Dirichlet distribution to an infinite simplex wherein each simplex dimension represents the location of an atomic mass (*i.e.*, a draw from a DP is a weighted sum of countably many atomic masses, whose weights sum to one). Inference in mixture models based on the DP prior simultaneously infer both the number of mixture components and the likelihood parameters of the components (parameters such as the minor allele frequencies in our genetic applications). The DP provides a particularly useful prior for HMMs. Marginally, finite HMMs define a mixture model over the space of emissions and transitions. In this formulation, the states of the HMM correspond to mixture components, and marginally the coefficient of each mixture component is given by the stationary distribution over the HMM states ([Teh et al., 2006](#)). By using a DP priors on the transition matrix, the DP can allow the number of latent HMM states to be inferred and to be unbounded ([Beal et al., 2002](#)) (although, if the true number of latent HMM states is finite, the DP will be inconsistent [Ghosal 2010](#)).

There are three main advantages for inference conferred by DP priors for HMMs:

1. In many parametric HMMs (*i.e.*, HMMs with a fixed and finite number of states), the HMM states are labelled with parameter values or indices. These models are invariant to permutations of the labels. The symmetries arising from each of the permutations of the labels therefore create an abundance of posterior modes in the model. Because modes are attractive, these symmetries tend to make MCMC inference algorithms converge more slowly. The intuition behind this can be understood as follows: if an MCMC state is the same ‘distance’ from two of the modes then the state is ‘pulled’ towards both of the modes with an ‘equal force’ and so the state will not move as quickly towards any given mode as it would if there were fewer modes. This is known as the label switching problem ([Celeux, 1998](#); [Jasra et al., 2005](#)). DP priors can avoid this problem by integrating out the label of the underlying mixture models, which results in distributions defined directly on the space of partitions of the data items. This is illustrated in [Teh](#)

- et al.* (2011).
- Typically, the number of clusters used in HMMs for genetic variation is chosen using model selection, or reversible jump MCMC (RJCMCMC). Model selection requires either training the model separately for each proposed number of states (the model classes) and evaluating Bayes factors and information criterion, or *ad-hoc* methods for each model class. With DP priors, the number of clusters is automatically inferred along with the other model parameters. Inference using DP priors is often simpler than RJCMCMC and the prior specified by DPs are often more naturally connected with the assumptions about the data than the prior specified by RJCMCMC (this is due to the connection between the Dirichlet process and allele sampling, which is explained in Chapter 2). DP priors can therefore reduce the amount of computation time required to conduct inference on data for which the number of clusters is not known.
  - DP priors add flexibility to models by increasing their expressiveness (*i.e.*, their priors have a larger support). This can lead to higher imputation accuracy and faster inference (Hjort *et al.*, 2010).

The three models we contribute in this thesis allow these benefits to be realized: in all three models we provide DP priors for HMMs of genetic variation and we integrate out the labels and parameters of the HMM states to avoid the label switching problem.

### 1.3.2 Bayesian nonparametric models of genetic variation: a preview

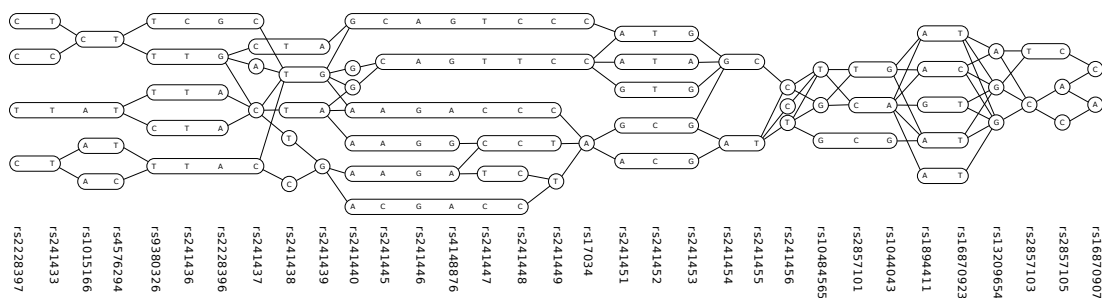
With these benefits of Bayesian nonparametrics listed above in mind, in the remainder of this section, we will preview the three new models contributed in this thesis, giving a brief overview of their natures.

#### 1.3.2.1 The Bayesian nonparametric version of fastPHASE

The first model we will present in this thesis (the BNPPHASE model) is based on a hierarchical Dirichlet process (Teh *et al.*, 2006) in which the latent states correspond to genetic founders or admixture components of a population. In previous research, hierarchical Dirichlet process HMMs (HDP-HMMs) with arbitrary transition matrices have been applied to genetic data (Xing *et al.*, 2006; Xing and Sohn, 2007a). As we saw in section 1.2, genetic data is highly structured. By ignoring this structure, arbitrary transition matrices can over-fit when trained on genetic data. Furthermore, genetic processes often have a nonhomogeneous component, and in Xing *et al.* (2006) and Xing and Sohn (2007a), the authors assumed that the transition matrices were homogeneous (*i.e.*, the same transition matrix was used at each location on the chromosome).

The BNPPHASE model extends Xing *et al.* 2006 and Xing and Sohn 2007a by forming a





**Figure 1.5:** Haplotype structure of the Utah residents with ancestry from northern and western Europe (CEU) and Yoruba in Ibadan, Nigeria (YRI) populations from HapMap ([The International HapMap Consortium, 2003](#)) found by the DFCP model. Data consists of SNPs from a region near the *TAP2* gene from the HapMap project.  $x$ -axis indicates SNP location and label.  $y$ -axis represents clusters from last sample of an MCMC chain converging to DFCP posterior. Letters inside clusters indicate base identity. Lines between haplotypes indicate transitions between contiguous haplotypes.

nonhomogeneous HMM and adding additional structure to the transition matrix such as emphasized self transitions ([Fox et al., 2011](#)). This additional structure is informed by approximations of the genetic process such as those developed in [Scheet and Stephens \(2006\)](#). The resulting structure in the transition matrix of the BNPPHASE model implies that BNPPHASE has fewer free parameters than an HDP-HMM with arbitrary transition matrices. This leads to more efficient learning in the BNPPHASE model and also less over-fitting.

As in [Xing et al. \(2006\)](#) and [Xing and Sohn \(2007a\)](#), the BNPPHASE model is nonparametric and the number of states is learned during inference simultaneously with the other model parameters. The finite truncation of the BNPPHASE model onto the first  $K$  states is similar to a version of the fastPHASE model [Scheet and Stephens \(2006\)](#) with  $K$  latent states, hence its name (this will be explored further in Chapter 3).

In Chapter 3, we derive the BNPPHASE model and show how it approximates the genetic processes that will be described in section 1.2. We develop inference for the BNPPHASE model based on MCMC and we apply it to genotype imputation and to estimation of the time to the most recent common ancestor of a sample.

### 1.3.2.2 The discrete fragmentation-coagulation process

The second model that we will present in this thesis is the discrete fragmentation and coagulation process (DFCP). The DFCP is a partition-valued HMM wherein the latent partition transitions from one location to the next by the splitting and merging of its clusters. As a model of genetic data, the DFCP provides a dynamic-clustering for the observed genetic sequences. At each location of interest on the chromosome, a latent partition of all of the genetic sequences is proposed. The transitions between partitions at adjacent locations are given by random fragmentation and coagulation



operators (Pitman, 2006). The parameterization of the operators is chosen in a way such that the resulting marginal prior distribution on the partition structure at each location of interest is induced by a DP, and has other desirable statistical properties (these properties are discussed in more detail in Chapters 2 and 4).

The DFCP is informed by the fine-scale haplotype structure of genetic variation (Daly et al., 2001). A haplotype is a pattern of mutations on a chromosome that all tend to be inherited together by virtue of their proximity to each other on the chromosome. Genetic variation of a population can often be described by piecing together a haplotype mosaic using the haplotypes that recur in the population. The end points of these blocks correspond to recombination hotspots (Jeffreys et al., 2001), or to locations of recombination in the ancestry of the population. This phenomenon was explained further in section 1.2.

The DFCP is a discrete analogue of the continuous fragmentation-coagulation process (CFCP) which was previously proposed for modelling local mosaic structure in genetic sequences (Teh et al., 2011). Inference algorithms derived for the CFCP also scale linearly in the number and length of the sequences (Teh et al., 2011). However, since the CFCP is a Markov jump process the computational overhead needed to model the arbitrary number of latent events located between two consecutive observations might preclude scalability to large datasets. The DFCP provides the advantages of the CFCP whilst being more scalable. The CFCP can also be derived as the limit of the DFCP achieved as the sampling frequency of the chromosome goes to infinity.

In Chapter 4, we will fully describe the DFCP model. We will present inference for the DFCP based on a forwards-filtering/backwards-sampling MCMC algorithm. In a series of experiments, we compared the scalability, MCMC mixing and imputation accuracy of the DFCP and the CFCP models. The experiments were done using SNP data from the Thousand Genomes project (The 1000 Genomes Project Consortium, 2010) and data simulated from the coalescent with recombination model (Hudson, 2002). An example of a draw from an MCMC chain with the DFCP posterior is given in Figure 1.5. In that figure, a dynamic-clustering of sequences of mutations around the TAP2 gene in a dataset from the HapMap project (The International HapMap Consortium, 2003) is shown.

### 1.3.2.3 The Wright-Fisher partition valued process

The third and final model that we will present in this thesis is the Wright-Fisher partition valued process (WFP). Like the DFCP and the CFCP, the WFP defines a process directly on the set of partitions of a set of data items. In the WFP, the latent partitions transition through the shrinking and growing of their clusters according to simple rates. The WFP model has much similarity to the BNPPHASE model: in both models, HMM states correspond to population proportions that vary ‘smoothly’ over the duration of

---

the process. As a result, the WFP is useful for the same sort of imputation problems that we will apply the BNPPHASE model to. However, because the WFP is not based on a hierarchy, its construction is simpler. Further, the WFP model is reversible (*i.e.*, it assigns the same probability to observed genetic data regardless of which end of the chromosome is at the first HMM location). In contrast, BNPPHASE model and the fastPHASE model from [Scheet and Stephens \(2006\)](#) are not reversible.

We provide inference for the WFP model using particle MCMC methods. This is explained in more detail in Chapter 5. Also in Chapter 5, in a short departure from the main application of this thesis, we apply the WFP model to voting data from the Canadian House of Commons.

## Chapter 2

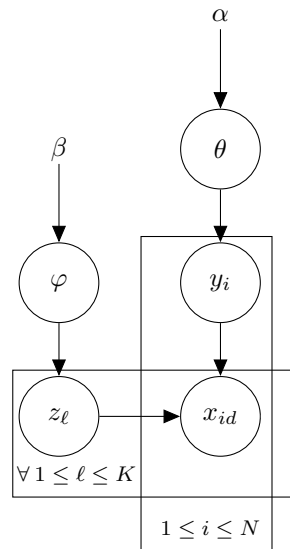
# Bayesian nonparametrics and dynamic-clustering

### 2.1 Introduction

Bayesian nonparametrics were first applied as a model for the prior distribution of nonparametric parameter spaces (Ferguson, 1973). In this classical application, the Dirichlet processes were used to associate a latent parameter with each observed data item. A draw from the DP posterior induces a clustering of the data items through the equivalence classes formed by the identity of the latent parameters: all data items with the same latent parameter are placed in the same cluster.

The DP enjoys many statistical properties that make it a versatile and tractable prior. For example, it is exchangeable: given a DP prior, the posterior distribution on the data items does not depend on the order in which the data items are observed. Exchangeability is a desirable property for distributions designed to model studies in which the inclusion of data items into the study is an independent, random procedure (examples of such studies include surveys in which respondents are polled). However, for studies in which covariates are also collected, the joint distribution of the data items conditioned on the covariates is no longer exchangeable.

We will illustrate this conditional non-exchangeability by considering latent Dirichlet allocation (LDA) for topic models of documents (Blei et al., 2002), which is a typical example application. We will think of documents as collections of words. In a topic model, each document is associated with a latent distribution over topics. Further, we will associate to each topic a latent distribution over all of the words in a vocabulary. Under the LDA model, each word in a document is assumed to be generated by the following process: first, a topic is chosen according to the document-specific distribution over topics. Second, the word is chosen according to the topic-specific distribution over words given by the topic chosen in the first step. A description of the LDA model is



**Figure 2.1:** Plate diagram for the LDA model.  $x_{id}$  is the  $d$ -th word of the  $i$ -th document.  $y_i$  is the distribution over topics for the  $i$ -th document.  $z_\ell$  is the distribution over words of the  $\ell$ -th topic.  $\theta$  and  $\varphi$  are priors on  $y$  and  $z$  respectively.

given more precisely by the graphical model in Figure 2.1.

The formulation of the LDA in the above paragraph is exchangeable. But suppose that the documents are papers published in the proceedings of some annual conference. If we also observe as a covariate the year in which each of the documents was published, then due to trends in the keywords, and the popularity of various topics discussed in the conference, we would not expect the documents to be exchangeable conditioned on the publication year. We would, however, still expect the documents to be exchangeable within a given year (*i.e.*, the joint distribution induced on the subset of documents that were all published in a given year is exchangeable).

To deal with covariates and conditional non-exchangeability, dependent Dirichlet processes have been developed that incorporate covariates into the model through a dependency structure (MacEachern, 1999). This has given rise to the field of dependent random processes (DRPs), which generally augment exchangeable random processes with covariates.

Like the Dirichlet process, DRPs can also be used to induce clusterings on data. There are two main ways in which such clusterings can be realized by DRPs. In the first way, a latent random process is parameterized by a covariate  $t$  (here,  $t$  could be the observation time or location of the item). Each data item  $i$  is associated with a single covariate value  $t_i$ , and a single clustering is produced for the items (this clustering is ‘static’ in the sense that each item is in a single block of the clustering, and the clustering is not parameterized by  $t$ ). This is the classical way in which DRPs were introduced in MacEachern (1999), and is used for example in dynamic LDA models (Rao and Teh, 2009), function estimation (Dunson, 2006) and in relational models (Miller et al., 2009; Ho et al., 2010).

In the second way, both the latent random process and the data items are parameterized by  $t$ . In this case, each data item  $i$  is associated with a series of observations. The items are clustered jointly at each value of the covariate  $t$ , producing a dynamic-clustering. This second way is more relevant for models of genetic variation: The chromosome is a linear structure and genetic sequence data typically samples genetic material from many chromosome locations. The models that we will present in this paper are examples of this second way of clustering data through DRPs. Other examples include (Palla et al., 2014), (Ahmed and Xing, 2008), (Beal et al., 2002) and (Blei and Frazier, 2011).

In this Chapter, we will give a formal development of the Dirichlet process and the hierarchical Dirichlet process (Teh et al., 2006). In sections 2.2 and 2.3 we develop the theory and notation required to derive inference for the three dynamic-clustering models presented in this thesis. In section 2.4 we describe the fragmentation and coagulation operators, which are ways of introducing dependencies between clusterings through the splitting and merging of their clusters. We will discuss the duality of the fragmentation and coagulation operators. As a novel contribution of this Chapter, we derive the conditional distributions of fragmentation and coagulation. These conditionals will be used in Chapter 4 to derive Gibbs updates for the DFCP.

## 2.2 The Dirichlet process through measures, partitions and sequential schemes

In finite mixture models, data items are assumed to be generated by a process in which first, the latent component assignment of each data item is drawn from a distribution over  $K$  mixture components and second, each data item is drawn from a distribution parameterized by its component assignment. This is illustrated in equation (2.1) below. To provide a conjugate posterior distribution, often the Dirichlet distribution is used as a prior on the distribution of the data items over the mixture components. Because the Dirichlet distribution is supported on the  $K$ -simplex, a draw from the Dirichlet distribution can be thought of as a random probability distribution function over the  $K$  mixture components. This is illustrated in the following generative process for a mixture model of some data items  $x_i$ :

$$\begin{aligned} (\omega_1, \dots, \omega_K) &\sim \text{Dirichlet}(a_1, \dots, a_K), \\ \psi_1, \dots, \psi_K &\stackrel{\text{i.i.d.}}{\sim} \mu, \\ z_1, \dots, z_n | \omega &\stackrel{\text{i.i.d.}}{\sim} \omega, \text{ (so } \Pr(z_i = k) = \omega_k), \\ x_i | z_i &\sim f(\psi_{z_i}). \end{aligned} \tag{2.1}$$

Here  $(\omega_1, \dots, \omega_K) \sim \text{Dirichlet}(a_1, \dots, a_K)$  means that the random vector  $(\omega_1, \dots, \omega_K)$  is governed by the density  $\Gamma(a_1 + \dots + a_K) / \Gamma(a_1) \cdots \Gamma(a_K) \omega_1^{a_1-1} \cdots \omega_K^{a_K-1}$  supported

on the set  $(\omega_1, \dots, \omega_K)$  such that  $\sum_{k=1}^K \omega_k = 1$  and  $\omega_k > 0$ . The vector  $a_1, \dots, a_K$  are hyperparameters ( $a_k > 0$ ) and  $z_i$  are the latent component assignments for the data items. The symbol  $f$  is a law governing the likelihood of the data for each mixture component, under the parameters of the mixture component ( $\psi$ ). The law  $f$  is parameterized by  $\psi \in X$ , and the probability measure  $\mu$  is a prior on  $\psi$ .

The Dirichlet process extends this Bayesian theory to infinite mixture models. Rather than providing a random distribution function on the  $K$ -simplex as in equation (2.1), the Dirichlet process provides a random probability measure  $G$  supported on the parameter space  $X$  (the space  $X$  must be a Polish space, more detail is given in Ghosal 2010). The Dirichlet process is defined through its joint distribution on finite collections of disjoint measurable subsets of  $X$ .

**Definition 1.** *Let  $\mu$  be a probability measure on  $X$  and let  $\alpha > 0$  be a concentration parameter. A random probability measure  $G$  on  $X$  is a Dirichlet process if for every partition of  $X$  into a collection of disjoint measurable subsets  $B_1, \dots, B_K$ :*

$$(G(B_1), \dots, G(B_K)) \sim \text{Dirichlet}(\alpha\mu(B_1), \dots, \alpha\mu(B_K)). \quad (2.2)$$

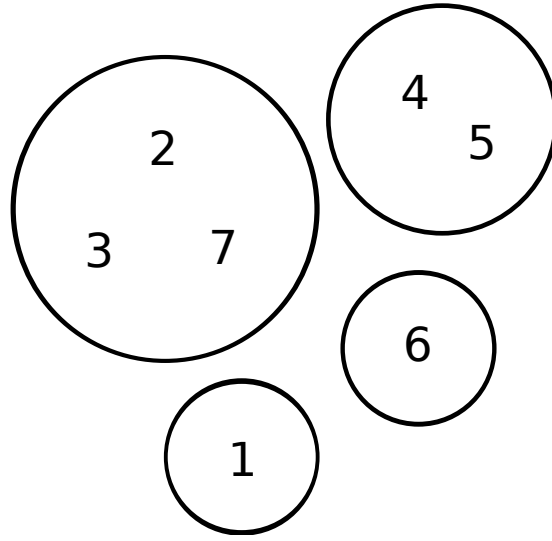
For each  $\alpha > 0$  and probability measure  $\mu$  there exists a unique random probability measure  $G$  satisfying (2.2). The existence and uniqueness of  $G$  can be proven using the normalization of Lévy processes (Ferguson, 1973). We will denote this Dirichlet process by  $G \sim \text{DP}(\alpha, \mu)$ . With probability one,  $G$  is a discrete probability measure and as long as  $\mu$  does not have finite support,  $G$  is a sum of a countably infinite number of atoms (Ghosal, 2010).

By using the countability of the support of the Dirichlet process, with probability 1 we can also construct it through the following stick breaking scheme, which makes explicit the joint distribution among the atom weights and atom locations:

$$G = \sum_{k=1}^{\infty} \omega_k \delta_{\psi_k}, \quad \nu_k \stackrel{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha), \quad \omega_k = \nu_k \prod_{k'=1}^{k-1} (1 - \nu_{k'}), \quad \psi_k \stackrel{\text{i.i.d.}}{\sim} \mu. \quad (2.3)$$

Here the  $\delta$ s in the definition of  $G$  are the atoms (they are Dirac delta functions). We can define a random variable  $\varphi$  with range  $1, 2, \dots$  induced by  $\omega$  through  $\Pr(\varphi = k) = \omega_k$ . We will denote this by  $\varphi | \omega \sim \omega$ . Through this definition,  $\omega$  can be thought of as a specification of a prior on the masses of the components of the mixture model: each  $k = 1, 2, \dots$  is a component of the mixture model, and the random variable  $\varphi$  selects a component of the mixture by sampling from the distribution given by the masses of the components. The distribution of the infinite random vector  $\omega$  is referred to as the Griffiths-Engen-McCloskey (GEM) distribution, and is denoted by  $\omega \sim \text{GEM}(\alpha)$ .

As in (2.1),  $\psi_k$  is the parameter of the  $k$ -th mixture component. The component assignments  $(\varphi_i)_{i \in R}$  induce a partition  $\mathcal{R}$  on  $R$  through the equivalence relation given



**Figure 2.2:** Example clustering  $\mathcal{R}$  of the set  $R = \{1, \dots, 7\}$  into 4 blocks.  $\#\mathcal{R} = 4$ , and  $\mathcal{R} = \{\{1\}, \{2, 3, 7\}, \{4, 5\}, \{6\}\}$ . Block assignment of item 1 is  $\varphi_1 = \{1\}$ , block assignment of item 2 is  $\varphi_2 = \{2, 3, 7\}$ , block assignment of item 3 is  $\varphi_3 = \{2, 3, 7\}$  and so on (*i.e.*, block assignment of  $n$  is the unique  $a \in \mathcal{R}$  such that  $n \in a$ ).

by  $i \equiv j$  if  $\varphi_i = \varphi_j$ . (A partition of a finite set  $R$  is a set of nonempty disjoint subsets of  $R$ , which we will refer to as blocks, whose union is all of  $R$ .) The distribution on partitions of  $R$  formed by marginalizing  $\omega$  is called the CRP (Chinese restaurant process) distribution and it is denoted by  $\mathcal{R} \sim \text{CRP}(R, \alpha)$ . The law of the CRP distribution is given by the following equation (we refer to [Aldous 1985](#) for a derivation):

$$\Pr(\mathcal{R} = A | \alpha) = \frac{\alpha^{\#A} \Gamma(\alpha)}{\Gamma(\alpha + \#R)} \prod_{a \in A} \Gamma(\#a). \quad (2.4)$$

Here,  $A$  is a partition of  $R$  and  $\#A$  is the cardinality of  $A$  as a set (*i.e.*,  $\#\mathcal{R}$  is the number of blocks in the partition  $\mathcal{R}$ ).

### 2.2.1 Ewens' sampling formula and random partitions

We will now consider the distribution on partitions (*i.e.*, clusterings) of the set  $R = \{1, \dots, n\}$  defined through Ewens' sampling formula ([Ewens, 1972](#); [Fisher et al., 1943](#)). This formula arises under mild genetic assumptions as the distribution on the pattern of alleles observed at a locus. Under Ewens' sampling formula, if we observe alleles of  $n$  individuals and if  $\zeta_i$  denotes the allele of individual  $i$ , and if  $s_j$  is the number of alleles that appear  $j$  times in the sample (*i.e.*,  $s_1 + 2s_2 + \dots + ns_n = n$ ), then:

$$\Pr(s_1, \dots, s_n | \alpha) = \frac{n! \Gamma(\alpha)}{\Gamma(\alpha + n)} \prod_{j=1}^n \frac{\alpha^{s_j}}{j^{s_j} s_j!} \quad (2.5)$$

Here  $\alpha > 0$  is a concentration parameter. As in the case of the CRP defined in the previous section, the allele assignments  $\zeta_i$  induce a partition (or clustering) on the set  $R = \{1, \dots, n\}$ . This distribution on partitions is given by  $\zeta_1, \dots, \zeta_n$  and each block of the partition is formed by taking blocks of  $R$  for each equivalence class of the relation in which  $i \equiv j$  if they have the same allele (*i.e.*, if  $\zeta_i = \zeta_j$ ). We will denote this partition by  $\mathcal{R}$ . We can use Ewens' sampling formula to assign a probability to  $\mathcal{R}$  and thereby define a random partition. In order to do this, we must multiply equation (2.5) by the number of partitions of  $R$  that yield the sequence  $(s_1, \dots, s_n)$  to correct for the multiplicity of the sequence. This number is  $n!^{-1} \prod_{j=1}^n j! s_j!$ . After this multiplication and the change of variables  $s_m = \#\{a \in \mathcal{R} : \#a = m\}$ , we find the probability of  $\mathcal{R}$  is given by the CRP probabilities in equation (2.4).

### 2.2.2 The CRP through a sequential scheme

A sample from a CRP can be realized through the following sequential scheme. In this scheme,  $R$  can be enumerated in any fixed order (Blackwell and MacQueen, 1973):

1. The first element of  $R$  joins a block by itself.
2. For  $i > 1$ , the  $i$ -th element of  $R$  joins a block by itself with probability (*w.p.*)  $\alpha/(i + \alpha - 1)$  or an existing block  $a$  *w.p.*  $\#a/(i + \alpha - 1)$ .

The probability of arriving at a given partition  $\mathcal{R}$  through this scheme is also given by equation (2.4), which shows that this scheme does not depend on the fixed order in which the items of  $R$  are enumerated. The invariance of the probability of  $\mathcal{R}$  to permutations of  $R$  means that the CRP distribution is exchangeable. As an example, consider the partition of  $\{1, \dots, 7\}$  given in Figure 2.2. In that figure, for Ewens' sampling formula,  $s_1 = 2, s_2 = 1, s_3 = 1, s_4 = 0, \dots, s_7 = 0$  and so the probability of this partition as given by equation (2.5) is  $7! \Gamma(\alpha) / \Gamma(\alpha + 7) \cdot \alpha^2 / (1^2 \cdot 2!) \cdot \alpha^1 / (2^1 \cdot 2!) \cdot \alpha^1 / (3 \cdot 1!)$  multiplied by  $1!2! \cdot 2!1! \cdot 3!1! / 7!$ , the number of partitions with the same values of  $(s_1, \dots, s_7)$ . Under the CRP, equation (2.4) defines the probability of the partition given in Figure 2.2 to be  $\alpha^4 \Gamma(\alpha) / \Gamma(\alpha + 7) \cdot 2!1!0!0!$ . Finally, under the above sequential scheme, assuming that the items are enumerated in increasing order, the probability of arriving at the given partition is  $1 \cdot \alpha / (\alpha + 1) \cdot 1 / (\alpha + 2) \cdot \alpha / (\alpha + 3) \cdot 1 / (\alpha + 4) \cdot \alpha / (\alpha + 5) \cdot 2 / (\alpha + 6)$ . These three probabilities are equal. They are equal to  $2\alpha^3 / (\alpha + 1) \cdot 1 / (\alpha + 2) \cdots 1 / (\alpha + 6)$ .

## 2.3 The hierarchical Dirichlet process

Suppose that we have sets  $R_1, \dots, R_L$  and we wish to form partitions of these sets and then link the blocks of all of the partitions together to form a dynamic-clustering. One of the simplest and most standard ways of doing this is through a hierarchical Dirichlet process (HDP), which we now describe. Let  $\alpha_0 > 0$  and  $\alpha > 0$  be concentration



parameters and let  $\mu$  be a probability measure on a space  $X$ . Let  $G_0 \sim \text{DP}(\alpha_0, \mu)$ , so  $G_0 = \sum_{k=1}^{\infty} \omega_k \delta_{\psi_k}$ , as in equation (2.3). Since  $G_0$  is a probability measure, it can be used as the mean measure of other Dirichlet processes. We will define  $G_1, \dots, G_L | G_0 \stackrel{\text{i.i.d.}}{\sim} \text{DP}(\alpha, G_0)$ . For each  $1 \leq \ell \leq L$ , by equation (2.3), we have that  $G_\ell = \sum_{k=1}^{\infty} \omega_{\ell k} \delta_{\phi_{\ell k}}$  where  $\phi_{\ell k} | G_0 \stackrel{\text{i.i.d.}}{\sim} G_0$  for  $k = 1, 2, \dots$

Since  $G_0$  is a discrete measure and  $\phi_{\ell k} | G_0 \sim G_0$ , for every fixed  $k$  there will be infinitely many indices  $k'$  such that  $\phi_{\ell k'} = \psi_k$ . If  $\xi \sim G_\ell$ , then in order to find the law of the index of  $\xi$  in  $\psi_1, \psi_2, \dots$ , we must first sum together all of the weights  $\omega_{\ell k'}$  of  $G_\ell$  such that  $\phi_{\ell k'} = \psi_k$ . We will denote this summation by  $\mathcal{R}_{\ell k} = \sum_{k': \phi_{\ell k'} = \psi_k} \omega_{\ell k'}$ . Then, if  $\xi | G_\ell \sim G_\ell$ , the unique index  $k$  of  $\xi$  in  $\psi_1, \psi_2, \dots$  has the law  $\Pr(k) = \mathcal{R}_{\ell k}$ . From equation (2.3) and from the properties of the Dirichlet distribution, we have the following conditional stick breaking construction for  $\mathcal{R}_{\ell k}$  (this construction is given in section 4.1 of Teh et al. 2006, and the construction is conditioned on  $\omega$  and the concentration parameter  $\alpha$ ):

$$\nu_{\ell k} \stackrel{\text{i.i.d.}}{\sim} \text{Beta} \left( \alpha \omega_k, \alpha \left( 1 - \sum_{k'=1}^{k-1} \omega_{k'} \right) \right), \quad \pi_{\ell k} = \nu_{\ell k} \prod_{k'=1}^{k-1} (1 - \nu_{\ell k'}). \quad (2.6)$$

We will refer to the distribution on  $\pi | \omega, \alpha$  induced by equation (2.6) as the coagulated version of the GEM distribution. To realize a dynamic-clustering of  $R_1, \dots, R_L$  we draw component assignments  $z_{i\ell}$  for each element  $i \in R_\ell$  using the distribution  $\Pr(z_{i\ell} = k) = \pi_{\ell k}$ . Then, we will suppose that items in blocks that share the same component assignment are in same cluster under the HDP (even if the indices of the  $R$ s are different). The dynamic-clustering of  $R_1, \dots, R_L$  is then formed according to the following equivalence relation: elements  $i \in R_\ell$  and  $i' \in R_{\ell'}$  are in the same cluster if  $z_{i\ell} = z_{i'\ell'}$ . Here,  $i$  and  $i'$  can be equal or unequal as can  $\ell$  and  $\ell'$ . This provides a link between the clustering of a single item at two different values of  $\ell$ , as well as a link between the clustering of two different items at a single value of  $\ell$ . In this way, the dynamic-clustering induces partitions  $\mathcal{R}_\ell$  of  $R_\ell$  (two elements  $i, i'$  are in the same cluster if and only if  $z_{i\ell} = z_{i'\ell}$ ).

## 2.4 Fragmentation and coagulation operators

Another way to realize dynamic-clustering on partitions is through the joint distributions on partitions defined by fragmentation and coagulation operators. These operators are random partition valued functions of partitions. In Chapter 4, we will describe the DFCP through a latent Markov chain of partitions such that the joint distribution of each pair of adjacent partitions is defined through the splitting and merging of their clusters according to the fragmentation and coagulation operators. We will define these operators in this section and we will examine their conditional distributions, which are required to derive effective Gibbs updates for the DFCP. We will also show that the frag-

mentation and coagulation operators are dual: if the parameters of the fragmentation and coagulation operators are chosen correctly, then their composition will leave the CRP distribution invariant (*i.e.*, if we draw a partition from a CRP, and then apply the fragmentation operator and then the coagulation operator, then the resulting distribution will be marginally CRP distributed). The conditionals for the fragmentation and coagulation operators were used to derive message passing in [Elliott and Teh \(2012\)](#), but the precise equation for the conditional distributions are presented for the first time in this thesis.

Before defining these operators, we will first extend the definition of the CRP distribution on partitions by adding a discount parameter. The two parameter version of the CRP distribution is given as follows:

$$\Pr(\text{CRP}(R, \alpha, d) = \mathcal{R}) = \frac{[\alpha + d]_d^{\#\mathcal{R}-1}}{[\alpha + 1]_1^{N-1}} \prod_{a \in \mathcal{R}} [1 - d]_1^{\#a-1}. \quad (2.7)$$

Here the number of elements  $\#\mathcal{R}$  is  $N$  and  $[x]_d^n = (x)(x + d) \dots (x + (n - 1)d)$  is Kramp's symbol and  $\alpha > -d, d \in [0, 1)$  are the concentration and discount parameters respectively ([Pitman, 2006](#)). This definition agrees with the one parameter version ( $d = 0$ ) of the CRP defined in equation (2.4). An equivalent sequential scheme is given as follows:

1. The first element of  $R$  joins a block by itself.
2. For  $i > 1$ , the  $i$ -th element of  $R$  joins a block by itself *w.p.*  $(\alpha + dK)/(i + \alpha - 1)$  or an existing block  $a$  *w.p.*  $(\#a - d)/(i + \alpha - 1)$ , where  $K$  is the number of blocks.

From the sequential scheme, we can see that the discount parameter encourages new items to join new blocks, and the extent of this encouragement increases to balance the tendency of blocks to join large blocks that already exist. This balance leads to a power-law in the number of blocks in the partition: if  $d < 0 \leq 1$  then  $\#\mathcal{R} = \mathcal{O}(n^d)$  whereas if  $d = 0$ ,  $\mathcal{R}$  follows the law (2.4) and  $\#\mathcal{R} = \mathcal{O}(\alpha \log(n))$  ([Pitman, 2002](#)).

As mentioned earlier, the fragmentation and coagulation operators are random partition valued functions of partitions. The fragmentation  $\text{FRAG}(\mathcal{R}, \alpha, d)$  of a partition  $\mathcal{R}$  is formed by independently partitioning further each cluster  $a$  of  $\mathcal{R}$  according to  $\text{CRP}(a, \alpha, d)$  and then taking the union of the resulting partitions, yielding a partition of  $R$  that is finer than  $\mathcal{R}$ . Conversely, the coagulation  $\text{COAG}(\mathcal{R}, \alpha, d)$  of  $\mathcal{R}$  is formed by partitioning the set of clusters of  $\mathcal{R}$  (*i.e.*, the set  $\mathcal{R}$  itself) according to  $\text{CRP}(\mathcal{R}, \alpha, d)$  and then replacing each cluster with the union of its elements, yielding a partition that is coarser than  $\mathcal{R}$ . (If every cluster of a partition  $A$  is contained in at least one of the clusters of a partition  $B$  then  $A$  is said to be finer than  $B$  and  $B$  is said to be coarser than  $A$ . Note that this is not a strict relationship and so a partition is always finer and coarser than itself.) The fragmentation and coagulation operators are linked through the following theorem from [Pitman \(1999\)](#).

**Theorem 1.** *Let  $R$  be a set, let  $\mathcal{A}_1, \mathcal{B}_1, \mathcal{A}_2, \mathcal{B}_2$  be random partitions of  $R$  such that:*

$$\begin{aligned} \mathcal{A}_1 &\sim \text{CRP}(R, \alpha d_2, d_1 d_2), & \mathcal{B}_1 | \mathcal{A}_1 &\sim \text{FRAG}(\mathcal{A}_1, -d_1 d_2, d_2), \\ \mathcal{B}_2 &\sim \text{CRP}(R, \alpha d_2, d_2), & \mathcal{A}_2 | \mathcal{B}_2 &\sim \text{COAG}(\mathcal{B}_2, \alpha, d_1). \end{aligned}$$

*Then, for all partitions  $\mathcal{A}$  and  $\mathcal{B}$  of the set  $R$  such that  $\mathcal{B}$  is finer than  $\mathcal{A}$ :*

$$\Pr(\mathcal{A}_1 = \mathcal{A}, \mathcal{B}_1 = \mathcal{B}) = \Pr(\mathcal{A}_2 = \mathcal{A}, \mathcal{B}_2 = \mathcal{B}). \quad (2.8)$$

This theorem is implied by [Pitman \(1999\)](#). In that work, the duality is presented in terms of a 2-parameter version of the Dirichlet process known as the Pitman-Yor process ([Pitman and Yor, 1997](#)). A purely algebraic version of this duality theorem for partitions was given in [Gasthaus and Teh \(2010\)](#).

### 2.4.1 Conditionals for fragmentation and coagulation operators

Suppose that  $\mathcal{R}$  and  $\mathcal{Q}$  are partitions of  $R = \{1, \dots, n\}$  such that  $\mathcal{Q} \sim \text{FRAG}(\mathcal{R}, 0, d)$ . Then, by the definition of the fragmentation operator, the distribution of  $\mathcal{Q}$  conditioned on  $\mathcal{R}$  is only supported on pairs of partitions such that  $\mathcal{Q}$  is finer than  $\mathcal{R}$ . Thus, for each block  $a \in \mathcal{R}$  there is a unique set of blocks in  $\mathcal{Q}$  that are contained in  $a$ . These are the blocks into which  $a$  fragments. We will denote these blocks in  $\mathcal{Q}$  by  $F_a$ : so  $F_a = \{b \in \mathcal{Q} : b \subseteq a\}$  and  $a = \cup_{b \in F_a} b$ . The conditional distribution of  $\mathcal{Q}$  given  $\mathcal{R}$  is as follows:

$$\begin{aligned} \Pr(\mathcal{Q} | \mathcal{R}, d) &= \prod_{a \in \mathcal{R}} \frac{\Gamma(\#F_a) d^{\#F_a - 1}}{\Gamma(\#a) \Gamma(1 - d)^{\#F_a}} \prod_{b \in F_a} \Gamma(\#b - d), \\ &= \frac{d^{\#\mathcal{Q} - \#\mathcal{R}}}{\Gamma(1 - d)^{\#\mathcal{Q}}} \left( \prod_{b \in \mathcal{Q}} \Gamma(\#b - d) \right) \left( \prod_{a \in \mathcal{R}} \frac{\Gamma(\#F_a)}{\Gamma(\#a)} \right). \end{aligned} \quad (2.9)$$

For coagulation, suppose that the partitions  $\mathcal{Q}$  and  $\mathcal{R}$  are such that  $\text{COAG}(\mathcal{Q}, \alpha/d, 0) = \mathcal{R}$ . As in fragmentation, the coagulation operator only gives support to the distribution of  $\mathcal{R}$  conditioned on  $\mathcal{Q}$  if  $\mathcal{Q}$  is finer than  $\mathcal{R}$  and we will denote the blocks in  $\mathcal{Q}$  that coagulate to form a block  $a \in \mathcal{R}$  by  $C_a$ . Thus, for each  $a \in \mathcal{R}$ ,  $C_a = \{b \in \mathcal{Q} : b \subseteq a\}$ . The conditional distribution of  $\mathcal{R}$  given  $\mathcal{Q}$  is as follows:

$$\Pr(\mathcal{R} | (\mathcal{Q}, \alpha, d) = \mathcal{R}) = \frac{(\alpha/d)^{\#\mathcal{R}} \Gamma(\alpha/d)}{\Gamma(\alpha/d + \#\mathcal{Q})} \prod_{a \in \mathcal{R}} \Gamma(\#C_a). \quad (2.10)$$

Equations (2.9) and (2.10) both assume that  $\mathcal{Q}$  is finer than  $\mathcal{R}$ . If  $\mathcal{Q}$  is not finer than  $\mathcal{R}$  then both the joint probabilities (2.9) and (2.10) are zero.

### 2.4.2 Conditionals for clustering a single item

To derive Gibbs updates for the location-varying cluster assignment of a single item in the DFCEP in Chapter 4, we will need the distribution of the cluster assignment of a single item in a fragmented or coagulated partition conditioned on the partition of all the other elements. In particular, if  $\mathcal{R}$  is a partition of  $R$  then by  $\mathcal{R}^{-i}$  we will refer to the projection of the partition  $\mathcal{R}$  onto  $R - \{i\}$ , here  $-$  denotes set difference. (The projection of  $\mathcal{R}$  onto  $S \subset R$  is formed by removing all elements of  $R - S$  from each block of  $\mathcal{R}$  and also removing any resulting empty sets from  $\mathcal{R}$ .) In this section, we will derive  $\mathcal{R}|\mathcal{R}^{-i}$ ,  $\mathcal{Q}^{-i}$  and  $\mathcal{Q}|\mathcal{Q}^{-i}$ ,  $\mathcal{R}^{-i}$  where  $\mathcal{R}$  and  $\mathcal{Q}$  are related as before through random fragmentation and coagulation.

Let  $a_i$  (respectively  $b_i$ ) be the cluster assignment of  $i \in R$  in  $\mathcal{R}$  (respectively  $\mathcal{Q}$ ). We will consider the distribution over  $a_i$  and  $b_i$  conditioned on  $\mathcal{R}^{-i}$  and  $\mathcal{Q}^{-i}$  respectively. If the  $i$ -th item is placed in a new cluster by itself in  $\mathcal{R}$  (*i.e.*, if it forms a singleton cluster) we will denote this event by  $a_i = \emptyset$ . For  $\mathcal{Q}^{-i}$  we will denote the respective event by  $b_i = \emptyset$ . Otherwise, the  $i$ -th item is placed in an existing cluster in  $\mathcal{R}^{-i}$  (respectively  $\mathcal{Q}^{-i}$ ) and we will denote this event by  $a_i \in \mathcal{R}^{-i}$  (respectively  $b_i \in \mathcal{Q}^{-i}$ ). Thus the support of the random objects  $a_i$  and  $b_i$  are respectively  $\mathcal{R}^{-i} \cup \{\emptyset\}$  and  $\mathcal{Q}^{-i} \cup \{\emptyset\}$ . In particular, the event  $a_i = \emptyset$  means that  $\mathcal{R} = \mathcal{R}^{-i} \cup \{\{i\}\}$  and the event  $a_i \in \mathcal{R}^{-i}$  means that  $\mathcal{R} = (\mathcal{R}^{-i} - a_i) \cup \{\{i\} \cup a_i\}$  (the same is true for  $b_i$  and  $\mathcal{Q}^{-i}$ ).

If  $\mathcal{R} \sim \text{CRP}(\alpha, 0)$ , then the distribution of  $a_i$  conditioned on  $\mathcal{R}^{-i}$  is given by the sequential scheme for the CRP distribution:

$$\Pr(a_i = a | \mathcal{R}^{-i}) = \begin{cases} \#a / (n - 1 + \alpha) & \text{if } a \in \mathcal{R}^{-i}, \\ \alpha / (n - 1 + \alpha) & \text{if } a = \emptyset. \end{cases} \quad (2.11)$$

To find the conditional distribution of  $b_i$  given  $a_i$  under the fragmentation and coagulation operators, we use their definition as combinations of independent CRP partitions of the clusters in  $\mathcal{R}$  and  $\mathcal{Q}$ . First, we will consider the fragmentation  $\text{FRAG}(\mathcal{R}, 0, d) = \mathcal{Q}$ . If  $a_i = \emptyset$ , then the  $i$ -th data item is in a cluster by itself in  $\mathcal{R}$  and so it will remain in a cluster by itself after the fragmentation operator is applied. Thus,  $b_i = \emptyset$  with probability 1. On the other hand, if  $a_i = a \in \mathcal{R}^{-i}$  then  $b_i$  must be one of the clusters in  $\mathcal{Q}$  into which  $a_i$  fragments. This can be a singleton cluster, in which case  $b_i = \emptyset$ , or it can be one of the clusters  $b \in \mathcal{Q}^{-i}$  in which case  $b \in F_a$ . Since  $a$  is fragmented according to  $\text{CRP}(a, 0, d)$ , when the  $i$ -th data item is added to this CRP it is placed in a cluster  $b \in F_a$  with probability proportional to  $(\#b - d)$  and it is placed in a singleton cluster with probability proportional to  $d\#F_a$ . Normalizing these probabilities yields

the following joint distribution:

$$\Pr(b_i = b | a_i = a, \mathcal{R}^{-i}, \mathcal{Q}^{-i}) = \begin{cases} (\#b - d)/\#a & \text{if } a \in \mathcal{R}^{-i}, b \in F_a, \\ d\#F_a/\#a & \text{if } a \in \mathcal{R}^{-i}, b = \emptyset, \\ 1 & \text{if } a = b = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (2.12)$$

Next, we will consider the coagulation  $\text{COAG}(\mathcal{Q}, \alpha/d, 0) = \mathcal{R}$ . To find the conditional distribution of  $a_i$  given  $b_i = b$ , we will use the definition of the coagulation operation. If  $b \neq \emptyset$ , then the  $i$ -th data item could not have been in a singleton cluster in  $\mathcal{Q}^{-i}$  and so it must follow the rest of the data items in  $b$  to the unique  $a \in \mathcal{R}^{-i}$  such that  $b \subseteq a$  (i.e.,  $b$  coagulates with other clusters to form  $a$ ). If  $b = \emptyset$  then the  $i$ -th data item is in a singleton cluster in  $\mathcal{Q}^{-i}$  and so we can imagine it being the last item added to the coagulating CRP( $\mathcal{Q}, \alpha/d, 0$ ) of the clusters of  $\mathcal{Q}$ . Hence the probability that  $i$ -th data item is placed in a cluster  $a \in \mathcal{R}^{-i}$  is proportional to  $\#C_a$  while the probability that it forms a cluster by itself in  $\mathcal{R}^{-i}$  is proportional to  $\alpha/d$ . After normalization, this yields the following joint probability:

$$\Pr(a_i = a | b_i = b, \mathcal{R}^{-i}, \mathcal{Q}^{-i}) = \begin{cases} 1 & \text{if } a \in \mathcal{R}^{-i}, b \in C_a, \\ d\#C_a/(\alpha + d\#\mathcal{Q}^{-i}) & \text{if } a \in \mathcal{R}^{-i}, b = \emptyset, \\ \alpha/(\alpha + d\#\mathcal{Q}^{-i}) & \text{if } a = b = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (2.13)$$

## 2.5 Summary

In this Chapter, we have outlined the history of dynamic-clustering and distance dependent random processes and their use as priors in Bayesian nonparametric statistics. We have shown a connection between the Dirichlet process and Ewen's sampling formula, a distribution on partitions that arises naturally in allele sampling. We have shown two ways to realize dynamic-clustering through the Dirichlet process, firstly through sequences of dependent random processes (the hierarchical Dirichlet process) and secondly, through the fragmentation and coagulation operators. We have derived the conditional distributions for the cluster assignment of a single item in partitions defined through fragmentation and coagulation operators. These conditional distributions appear for the first time in this thesis and they will be used in Chapter 4 to derive Gibbs updates for the conditional cluster assignment of sequences in the discrete fragmentation and coagulation process. The mathematics developed in this Chapter will be used throughout the remainder of this thesis in the application of the three new dynamic-clustering methods presented in this thesis.

## Chapter 3

# The Bayesian nonparametric version of fastPHASE

### 3.1 Introduction

We will now present a Bayesian nonparametric HMM for dynamic-clustering of genetic sequences based on the hierarchical Dirichlet process (Elliott and Teh, 2015). This model allows tractable inference and it captures properties important for the genetic process such as haplotypes and nonhomogeneous structure (these are reviewed in section 1.2.1). The popular `fastPHASE` model (Scheet and Stephens, 2006) can be seen as a finite truncation of this model. We will refer to this model as the `BNPPHASE` model (for the Bayesian nonparametric version of `fastPHASE`). The Bayesian nature of the `BNPPHASE` model allows the statistical properties of the genetic process to directly inform the structures found by `BNPPHASE` during inference. This leads to high accuracy for genotype imputation and also to interpretability of the model parameters. Further, by defining distributions directly on the space of partitions, the `BNPPHASE` model avoids the label switching problem (Jasra et al., 2005). The advantages of using Bayesian nonparametrics in this situation are reviewed in more detail in section 1.3.1.

The nonhomogeneous structure of the `BNPPHASE` model makes it particularly well suited for modelling data from population bottlenecks. Population bottlenecks are events occurring in the ancestry of a population in which the number of individuals in the population shrinks suddenly due to external factors such as environmental or ecological changes, migration or changes in human behavior. For example, in the 19th century, the northern elephant seal was hunted into near extinction, and shrunk to a population of fewer than 30 animals. After hunting ceased, the population expanded (Hoelzel et al., 1993). The genetics of populations which have experienced bottlenecks display founder effects in which all genetic material of the post-bottleneck individuals originates from a small number of founders. In such data, genetic variation of an observed sample

is predominantly explained by the original variation between the founders and also the recombination events occurring in the post-bottleneck genealogies. Such data is modelled well by HMMs in which each population founder corresponds to an HMM state.

We conducted three experiments using the **BNPPHASE** model involving sequences of biallelic markers in phased genetic data. In our first three experiments, we examined the imputation accuracy of the **BNPPHASE** model and compared it to that of **fastPHASE** and also other baselines. We found that the **BNPPHASE** model performed competitively with the state-of-the-art in the imputation of missing data. In our first experiment, we examined imputation accuracy on a ‘toy’ dataset generated from the ARG with an identity-by-descent rule wherein all mutations were assumed to have occurred more anciently than the bottleneck. This simulated a very recent bottleneck from a small number of founders. In our second experiment, we performed genotype imputation on male X chromosome data from the Thousand Genomes Project ([The 1000 Genomes Project Consortium, 2010](#)). As explained in section 1.1, the phase of male X chromosome data is known, and so a ground-truth for the imputation of held out data can be established, providing valid accuracies.

In our third experiment we examined the correlation between the time to the most recent common ancestor (TMRCA) and the number of clusters used by the **BNPPHASE** model (*i.e.*, the latent dimensionality of the nonparametric HMM). For the data, we generated sequences from a population bottleneck data designed to model the out-of-Africa population bottleneck in humans. We found a strong negative correlation between these values in both the **BNPPHASE** and **fastPHASE** models. After regressing the TMRCA against the number of clusters, residual error of the **BNPPHASE** model was smaller than that of other methods.

Markov models based on the HDP have been used previously to describe genetic variation. In [Xing et al. \(2006\)](#), an HDP-HMM was used to model genetic sequences. The HDP-HMM places an HDP prior on the full transition matrix of an infinite HMM ([Beal et al., 2002](#)), resulting in a homogeneous process. In contrast, the **BNPPHASE** model introduces nonhomogeneity into the HMM prior. This allows the **BNPPHASE** model to capture genetic structure in which the proportions for genetic founders or admixture components varies along the chromosome. We refer to section 1.2.7 for more detail about the relation of the **BNPPHASE** model to other HDP-HMMs.

In the remainder of this section, we discuss the statistical properties of genetic sequence data arising from population bottlenecks. Then, we provide some intuition for the **BNPPHASE** model and the likelihood of phased genotype data given by the **BNPPHASE** model. We also give intuition as to why the **BNPPHASE** model is a good model for population bottleneck data. In section 3.2, we provide the details for the generative process of the **BNPPHASE** model, and derive inference for the **BNPPHASE** model based on MCMC using Gibbs updates for the latent state assignments of a sequence and slice

sampling for updating the parameters. In section 3.4 we describe the experimental paradigms for the three experiments that we conducted involving the BNPPHASE model. In sections 3.5, 3.6 and 3.7 we provide the results of these experiments, some discussion and then we conclude.

Open source code implementing MCMC inference for the BNPPHASE model is provided at the website <http://www.github.com/lell/BNPPPhase>. This code is written in a combination of java and scala and is published under the BSD 2-clause license.

### 3.1.1 Population bottlenecks and genetic sequence data

In Kingman’s coalescent for genealogies, the coalescent rate of the lineages in the genealogy is twice the inverse effective population size  $\frac{2}{N_e}$  (this is explained in section 1.2.1). To simulate from a version of Kingman’s coalescent in which the population size changes during the ancestry, we can parameterize  $N_e$  by time and then sample a nonhomogeneous Poisson process with intensity  $\frac{2}{N_e(t)}$ . At the times given by the points in the Poisson process, we can then coalesce a pair of lineages chosen uniformly from all pairs extant at that time. In a similar way, the coalescent with recombination can be simulated for varying effective population size by superimposing the nonhomogeneous version of Kingman’s coalescent and a nonhomogeneous recombination process with rate  $\frac{2\rho}{N_e(t)}$  (we refer to Hein et al. 2005 for more detail).

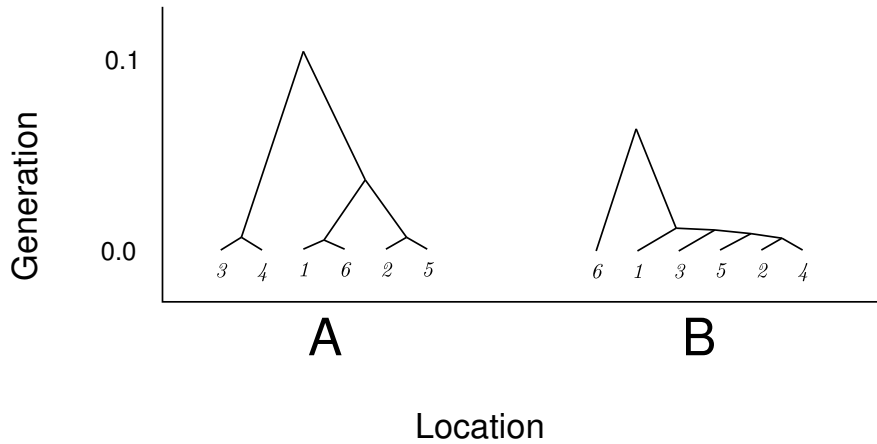
Population bottlenecks can be defined through the shape of the effective population size  $N_e(t)$  as a function of time—any sudden shrinking of  $N_e(t)$  specifies a bottleneck. Data from a population bottleneck can therefore be simulated by sampling from the time-varying version of the coalescent with recombination with such an  $N_e(t)$  (an example is given in Figure 3.1). Since  $N_e(t)$  is proportional to the inverse of the coalescence rate, we see that if a large bottleneck occurs recently in the ancestry of a population, most of the coalescence should occur during the bottleneck. However, coalescence that occurs more anciently than the bottleneck will tend to occur at a much slower rate.

This remark implies that for such a population, the TMRCA as a function of chromosome location will tend to be drawn either at a time during the bottleneck, or at a time governed by a heavy-tailed distribution centered much more anciently than the population bottleneck. This leads to nonhomogeneous structure, as the TMRCA will transition along the chromosome between intervals of coalescence during the bottleneck and coalescence much more anciently than the bottleneck.

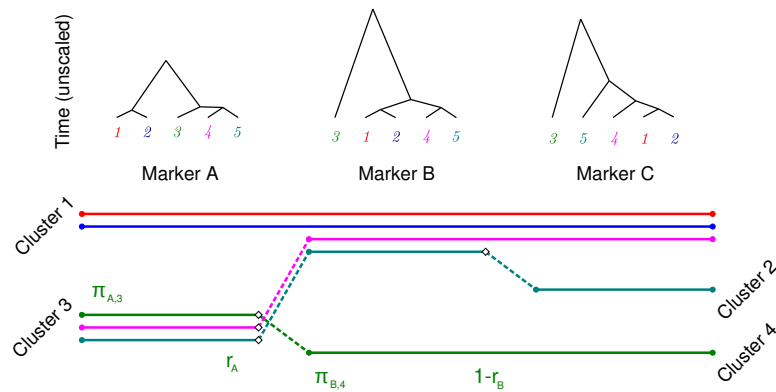
### 3.1.2 Intuition for the BNPPHASE model

The BNPPHASE model is an HMM approximation of the coalescent with recombination. Suppose that  $N$  phased genetic sequences from a population are typed at  $L$  biallelic markers. The BNPPHASE model associates a latent cluster assignment to each sequence





**Figure 3.1:** Genealogy of 6 homologous sequences with simulated ancestry.  $y$ -axis indicates time to coalescence.  $x$ -axis indicates chromosome position, with  $A$  labelling a location near one end of chromosome and  $B$  labelling a location near other end. Simulation conducted with parameters from (Li and Durbin, 2011) designed to model out-of-Africa bottleneck in humans. This illustrates location-dependent nature of genetic similarity: sequences 1 and 6 would be quite similar at location  $A$ , but quite different at location  $B$ .



**Figure 3.2:** *Top:* Genealogy of 5 genetic sequences with simulated ancestry from a population bottleneck.  $y$ -axis indicates time to coalescence.  $x$ -axis indicates chromosome location, with marker  $A$  labelling one end of chromosome, marker  $B$  labelling the middle of chromosome and marker  $C$  labelling the other end. Coalescence of lineage indicates arrival at common ancestor. Note that TMRCA is a function of chromosome location. *Bottom:* Illustration of reasonable latent sample found by BNPPHASE model. Color indicates sequence identity (with red being sequence 1, blue being sequence 2 and so on). Dotted lines indicate cluster transitions.  $y$ -axis indicates cluster assignment (sequences close to each other on the  $y$ -axis are in the same cluster). Cluster assignment ‘matches’ genealogical structure of *top*. Sequences remain in the same cluster from one marker to the next with probability  $1 - r_{\ell-1}$  or transit to cluster  $k$  with probability  $r_{\ell-1}\omega_{\ell,k}$ . Factors describing conditional probability of the green sequence (sequence 3) are shown by the terms in green.

and location. Between each pair of consecutive locations, a given sequence either remains in the same cluster, or with some probability proportional to a rate  $r_\ell$  (which depends on chromosome location  $\ell$ ) is reassigned to one of the clusters with a locus-dependent probability.

The latent rate  $r_\ell$  governs the dependence among the clusterings: if  $r_\ell = 0$  then the clusterings at each location are the same and if  $r_\ell = 1$  then the clusterings are all independent. Intuitively, we want to infer small values of  $r_\ell$  on regions of the genetic sequence for which the underlying genealogy structure from the coalescent does not change much, and larger values of  $r_\ell$  for locations where recombination events in the ancestry of the population have led to substantial changes in the latent genealogy structure. (For example, for the recombination hotspots described in [S. Myers et al. 2005](#)  $r_\ell$  should be relatively large compared to the background regions.)

We introduce latent variables  $z_{i\ell}$  denoting the cluster assignment of the  $i$ -th sequence at the  $\ell$ -th location and auxiliary variables  $y_{i\ell}$  indicating if the  $i$ -th sequence has a transition event after the  $\ell$ -th location. We also introduce cluster weights  $\omega_{\ell k}$  for the  $k$ -th cluster at locus  $\ell$  (such that  $\omega_{\ell k} \geq 0$  and  $\sum_k \omega_{\ell k} = 1$ ). If  $y_{i,\ell-1} = 1$ , then the cluster assignment of the  $i$ -th sequence at position  $\ell$  is *a priori* drawn from a discrete distribution with the probability of  $z_{i\ell} = k$  being  $\omega_{\ell k}$ . Otherwise (if  $y_{i,\ell-1} = 0$ ) the cluster assignment of the  $i$ -th sequence at position  $\ell$  is copied from  $z_{i,\ell-1}$ . For the first position ( $\ell = 1$ ) the prior distribution on the cluster assignment of the  $i$ -th individual is given by  $\omega_{1k}$ .

The number of clusters at each location, and the prior distribution over the local cluster weights  $\omega_{\ell k}$ , are given by a hierarchical Dirichlet process. In order to make this distribution well defined, we will have to identify the clusters at each location  $\ell$  with global clusters that persist across the whole process. The hierarchical Dirichlet process is the simplest method for identifying the clusters at each location such that the number of clusters is unbounded and the induced prior distribution on the cluster assignments does not depend on the order in which the individuals are observed or the size of the population from which the study individuals were selected (*i.e.*, it is exchangeable and projective).

An illustration of the BNPPHASE model is provided in [Figure 3.2](#). If we assume that the number of clusters is fixed at  $K$ , then we get a finite truncation of BNPPHASE which is described later in [section 3.3.1](#). This finite truncation is similar to the `fastPHASE` model ([Scheet and Stephens, 2006](#)), but with a Bayesian prior on the parameters.

In [Figure 3.2\(bottom\)](#), we imagine that our data consist of five phased genetic sequences typed at three biallelic markers (labelled as A, B and C). The sequences are labelled by color: red, blue, green, magenta and teal. The latent cluster assignment for BNPPHASE has represented the data using four latent clusters (corresponding to the  $y$ -level of the sequences in [Figure 3.2\(bottom\)](#)). At the first marker, BNPPHASE has clustered the

data into two clusters: with the red and blue sequences in cluster 1 (contributing  $\omega_{A,1}^2$  to the probability), and remaining sequences in cluster 3 (contributing  $\omega_{A,3}^3$  to the probability). Between marker A and marker B, the green, magenta and teal sequences transit to new clusters contributing  $(1-r_A)^2 r_A^3$  to the probability (the fact that the red and blue sequences do not transit contributes an additional  $(1-r_A)^2$ ). Between marker A and B, the green, magenta and teal sequences have transitioned to clusters 1, 1 and 4 respectively contributing  $\omega_{B,1}^2 \omega_{B,4}$  respectively to the prior. The probabilities for the transitioning and clustering between markers B and C can similarly be read off of Figure 3.2 resulting in a total probability (conditioned on  $\omega$  and  $r$ ) of the illustration in Figure 3.2:

$$(1-r_A)^2 r_A^3 (1-r_B)^4 r_B \omega_{A,1}^2 \omega_{A,3}^3 \omega_{B,1}^2 \omega_{B,4} \omega_{C,3}.$$

### 3.1.3 Likelihood of phased data under the BNPPHASE model

We will assume that the observed genetic sequences are phased and typed at biallelic markers, they can be summarized by the matrix  $x = ((x_{i\ell})_{\ell=1}^L)_{i=1}^N$  where  $x_{i\ell} = 1$  indicates that sequence  $i$  has the minor allele at location  $\ell$  and  $x_{i\ell} = 0$  indicates that sequence  $i$  has the major allele at location  $\ell$  (this is the form of phased data described in section 1.1). Given a fixed setting of the latent variables and parameters of the BNPPHASE model, the matrix  $x$  is a matrix of independent Bernoulli random variables. The distribution of each entry  $x_{i\ell}$  depends only on the cluster assignment ( $z_{i\ell}$ ) of the  $i$ -th sequence at location  $\ell$ . In particular, if the  $i$ -th sequence is in cluster  $k$  at location  $\ell$ , then the probability that  $x_{i\ell} = 1$  is  $\theta_{\ell k}$  and the probability that  $x_{i\ell} = 0$  is  $1 - \theta_{\ell k}$ . Here,  $\theta_{\ell k} \in [0, 1]$  is a parameter associated with the  $k$ -th cluster and the  $\ell$ -th location.

In the BNPPHASE model, we place a hierarchical prior on  $\theta_{\ell k}$  as follows:  $\theta_{\ell k}$  is drawn from a beta distribution with local mean and mass which both depend on  $\ell$ , so  $\theta_{\ell k} \sim \text{Beta}(\gamma_\ell \beta_\ell, \gamma_\ell (1 - \beta_\ell))$ . The local mean  $\beta_\ell$  is drawn from the beta distribution  $\text{Beta}(b, b)$  where  $b$  is a global parameter controlling the variance of the allele frequencies. We placed exponential priors with rate 1 on  $b$  and on each of the local masses  $\gamma_\ell$ .

### 3.1.4 Inference for the BNPPHASE model

We use MCMC to conduct inference on the posterior distribution of the BNPPHASE model conditioned on observed data. In the experiments described in this Chapter, we will be interested in the conditional distribution of missing alleles. We will also be interested in the posterior distribution over the number of clusters found by the BNPPHASE model. These statistics are estimated by averaging the marginal distributions of all MCMC iterations produced after a number of burn-in iterations have been completed.

For a fixed sequence  $i$ , we update the latent cluster assignments  $z_{i\ell}$  and transitions  $y_{i\ell}$

for  $\ell = 1, \dots, L$  by using the forwards filtering/backwards sampling algorithm (Früwirth-Schnatter, 1994) along with a bespoke auxiliary variable method to efficiently handle the infinite state space of the hierarchical Dirichlet process. The rates  $r_\ell$  and the parameters of the likelihood  $b, \gamma_\ell$  and  $\beta_\ell$  are updated using slice sampling (Neal, 2003). The likelihood parameters  $\theta_{\ell k}$  are integrated out. These updates are all derived in section 3.2.

## 3.2 Methods

In this section, we will formulate the full distribution of the BNPPHASE model using the stick breaking representation of hierarchical Dirichlet processes. We will then develop the MCMC inference methods required to provide tractable updates for the BNPPHASE posterior distribution. In sections 3.2.1 and 3.2.2, we will provide two equivalent generative processes for the BNPPHASE model. The first generative process will make use of the stick breaking construction of the HDP. For the second generative process, we will marginalize some aspects of the stick breaking construction for the hierarchical Dirichlet process. This will allow us to define the BNPPHASE HMM directly on the space of partitions of the sequence identities. We will describe the marginalized version of the HDP in 3.2.3. The second generative process provides a representation of the BNPPHASE for which tractable inference can be derived, which is done in section 3.2.5.

### 3.2.1 Generative process for the BNPPHASE model from stick breaking

The BNPPHASE model can be described by the following generative process. We will suppose that the concentration parameters  $\alpha_0 > 0, \alpha > 0$  as well as the likelihood parameters  $b > 0, \beta_\ell > 0$  and  $\gamma_\ell > 0$  are fixed.

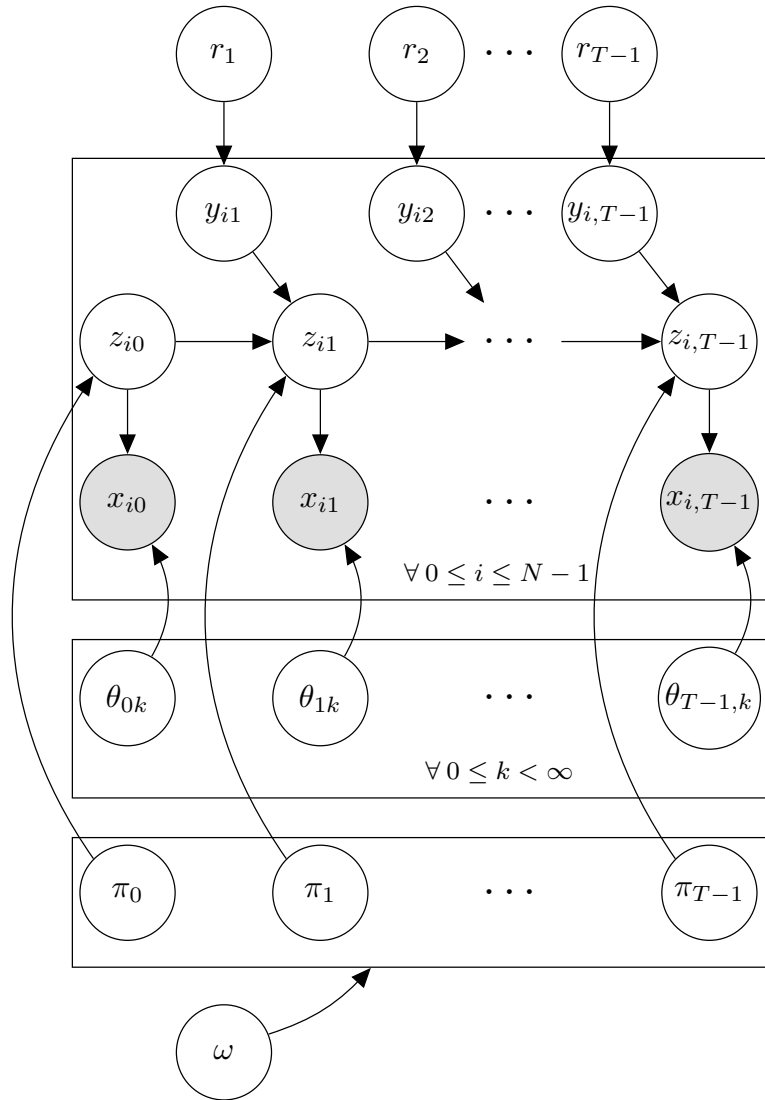
1. Draw  $\omega | \alpha_0$  according to the GEM distribution from equation (2.3):

$$\nu_k \stackrel{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha_0), \quad \omega_k = \nu_k \prod_{k'=1}^{k-1} (1 - \nu_{k'}).$$

2. For each  $1 \leq \ell \leq L$ : draw  $\pi_{\ell k} | \alpha, \omega$  according to the coagulated version of the GEM distribution from equation (2.6):

$$\eta_{\ell k} \stackrel{\text{i.i.d.}}{\sim} \text{Beta} \left( \alpha \omega_k, \alpha \left( 1 - \sum_{k'=1}^{k-1} \omega_{k'} \right) \right), \quad \pi_{\ell k} = \eta_{\ell k} \prod_{k'=1}^{k-1} (1 - \eta_{\ell k'}).$$

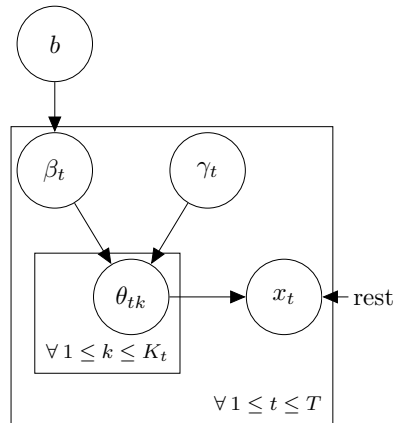
3. For each  $1 \leq \ell < L$ : draw  $r_\ell \sim \text{LogUniform}(r_{\min}, 1)$  (*i.e.*  $\log r_\ell$  is uniformly distributed).
4. For each  $1 \leq i \leq n$ :



**Figure 3.3:** Plate diagram for entire BNPPHASE model. For brevity, the hierarchical Dirichlet process parameters  $\alpha_0$  and  $\alpha$  and the hyperparameters for  $\theta$  are not shown.  $T$  denotes number of markers.

- (a) Draw  $z_{i1} | \pi_1 \sim \pi_1$ .
- (b) For each  $1 \leq \ell < L$ : draw  $y_{i\ell} | r_\ell \sim \text{Bernoulli}(r_\ell)$  and:
  - i. If  $y_{i\ell} = 1$ : draw  $z_{i,\ell+1} | \pi_{\ell+1} \sim \pi_{\ell+1}$ .
  - ii. Otherwise if  $y_{i\ell} = 0$ : set  $z_{i,\ell+1}$  to  $z_{i\ell}$ .
5. For each  $1 \leq \ell \leq L$ : draw  $\beta_\ell | b \sim \text{Beta}(b, b)$ .
6. For each  $1 \leq \ell \leq L, k = 1, 2, \dots$ : draw  $\theta_{\ell k} | \gamma_\ell, \beta_\ell \sim \text{Beta}(\gamma_\ell \beta_\ell, \gamma_\ell (1 - \beta_\ell))$ .
7. For each  $1 \leq \ell \leq L, 1 \leq i \leq n$ : draw  $x_{i\ell} \sim \text{Bernoulli}(\theta_{\ell, z_{i\ell}})$ .

A plate diagram showing the independence relations among the variables and parameters of the BNPPHASE model implied by this generative process is provided in Figures 3.3



**Figure 3.4:** Plate diagram for hierarchical likelihood used by BNPPHASE model. Node ‘rest’ indicates prior from BNPPHASE model. Variables  $\theta_{tk}$  will be integrated out in MCMC inference.

and 3.4. Here,  $r_\ell \sim \text{LogUniform}(u, v)$  means that  $r_\ell$  is a random variable such that  $\log r_\ell$  is uniformly distributed on the interval  $[\log u, \log v]$ . We chose this weakly informative heavy tailed prior on  $r_\ell$  so that haplotypes can extend over large chromosome regions (over which  $r_\ell$  has small values) while still allowing recombination hotspots (S. Myers et al., 2005) to occur (these are locations for which  $r_\ell$  is close to 1).

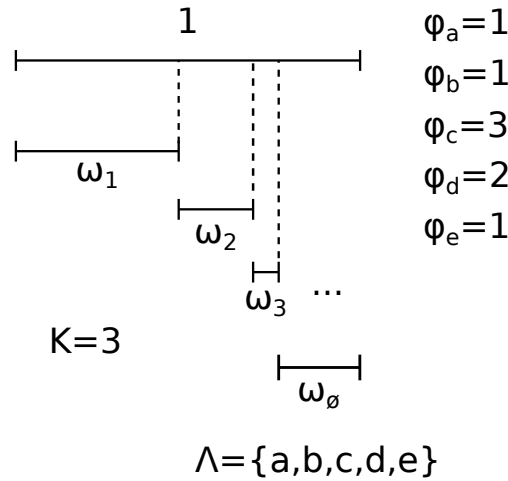
### 3.2.2 Generative process for the BNPPHASE model from partitions

We found that inference based on equations (2.3) and (2.6) was hard to specify because message passing algorithms for the generative process enumerated in the above section do not have finite support (the messages would be parameterized by the support of  $z$ , which is infinite). To overcome this problem, we derive inference for a marginalized version of the HDP. In this version of the HDP, we marginalize the GEM proportions  $\pi_\ell$  and define messages directly on the space of the partitions of the sequences. Marginalizing latent variables in Bayesian nonparametric models also tends to improve the efficiency of inference (this can be seen for example in the collapsed LDA sampler from Kurihara et al. 2007), which further recommends this approach.

### 3.2.3 The hierarchical Dirichlet process through partitions

The marginalized version of the HDP works by replacing  $\pi_\ell$  by a sequence of partitions  $\mathcal{R}_\ell$ . Each block of the partition is assigned to a component of the ‘upstairs’ Dirichlet process  $G_0$ . Then, blocks that are assigned to the same component are identified (in essence, this construction combines the CRP distribution with the stick breaking construction through marginalizing  $\pi_\ell$ ).

The marginalized version of the HDP is constructed as follows. Let  $\mathcal{R}_1, \dots, \mathcal{R}_L$  be random *i.i.d.* partitions such that  $\mathcal{R}_\ell | \alpha \sim \text{CRP}(R_\ell, \alpha)$  for  $1 \leq \ell \leq L$  (here,  $\alpha > 0$



**Figure 3.5:** Stick breaking construction for Dirichlet process with  $K = 3$ . Unit interval divided into  $\tilde{\omega}_1, \tilde{\omega}_2, \tilde{\omega}_3$  and  $\tilde{\omega}_\emptyset = 1 - \tilde{\omega}_1 - \tilde{\omega}_2 - \tilde{\omega}_3$ . Component assignments  $\varphi_a, \dots, \varphi_e$  are sampled from  $\Pr(k|\omega) = \omega_k$ .

is the concentration parameter of the ‘downstairs’ Dirichlet processes). We will now assign the blocks of  $\mathcal{R}_\ell$  to atoms (or, components) of the DP  $G_0$ . For each  $\ell$  and for each block  $a$  of the partition  $\mathcal{R}_\ell$  we will draw the component assignment  $\varphi_{\ell a}$  associated with the block  $a$  by sampling  $\varphi_{\ell a}|G_0 \sim G_0$  *i.i.d.* as in section 2.2.

As before, let  $z_{i\ell}$  be the component assignment of the  $i$ -th sequence at location  $\ell$ , and let  $y_{i\ell}$  be a binary variable indicating if the  $i$ -th sequence has a ‘jump’ event after location  $\ell$ . We will denote the set of all individuals that ‘jump’ after location  $\ell$  by  $R_\ell$ . Thus,  $R_\ell = \{i : 1 \leq i \leq n, y_{i,\ell-1} = 1\}$  for  $\ell > 1$  and we will define  $R_1$  to be  $\{1, \dots, n\}$ . Finally, if  $y_{i,\ell-1} = 1$ , then we will denote by  $\zeta_{i\ell}$  the block in  $\mathcal{R}_\ell$  containing  $i$  (*i.e.*, the block that sequence  $i$  ‘jumps’ to after  $\ell$ ). If  $y_{i,\ell-1} = 0$ , we will set  $\zeta_{i\ell}$  to 0.

Since the set of all blocks  $a \in \mathcal{R}_\ell$  is finite, there will be a finite number of distinct components among the draws  $(\varphi_{\ell a})_{a \in \mathcal{R}_\ell}$ . We will refer to this number of distinct components from the set  $\{\varphi_{\ell a}\}_{a \in \mathcal{R}_\ell, \ell=1, \dots, L}$  by  $K$  and we will assume that the masses of these  $K$  components are given by  $\tilde{\omega}_1, \dots, \tilde{\omega}_K$ . Further, we will refer to the remaining components of  $\omega$  by  $\tilde{\omega}_\emptyset = 1 - \sum_{k=1}^K \tilde{\omega}_k$ . This is the sum of the weights of the components that are not among the  $K$  unique components appearing in  $(\varphi_\ell)_{\ell=1, \dots, L}$ . An example of this construction, with  $K = 3$  is given in Figure 3.5. Note that the subscripts of  $\tilde{\omega}_1, \dots, \tilde{\omega}_K$  do not correspond to the order in which the masses  $\omega_1, \omega_2, \dots$  that are sampled through stick breaking in equation (2.3) (hence the tilde distinction). Instead, the order of the subscripts is arbitrary, and chosen for convenience.

If  $y_{i,\ell-1} = 1$ , then the distribution of  $\zeta_{i,\ell}$  is given by the conditional CRP probabilities from equation (2.11). On the other hand, if  $y_{i,\ell-1} = 0$ , then we will set  $\zeta_{i\ell} = 0$  and set  $z_{i\ell} = z_{i,\ell-1}$ . In this case, the cluster assignment of individual  $i$  at location  $\ell$  is copied from the assignment at location  $\ell - 1$  and we can ignore the block for individual  $i$  at location  $\ell$  because the cluster was not found by examining the component assignment

of a block, (recall that this is denoted by setting  $\zeta_{i\ell}$  to zero). Thus, for the rest of this methods section we have:

$$\zeta_{i\ell} = 0 \Leftrightarrow y_{i,\ell-1} = 0 \text{ for } \ell > 1. \quad (3.1)$$

Due to equation (3.1), the value of  $y_{i,\ell-1}$  can be inferred by  $\zeta_{i\ell}$  for all  $1 < \ell \leq L$ . Therefore, we will drop the variable  $y_{i\ell}$  from the rest of this methods section, and just write  $\zeta_{i\ell} = 0$  if sequence  $i$  does not ‘jump’ before  $\ell$  and  $\zeta_{i\ell} \neq 0$  if sequence  $i$  ‘jumps’ before  $\ell$  (and in this latter case  $\zeta_{i\ell}$  will be the block of  $\mathcal{R}_\ell$  individual  $i$  ‘jumps’ to after location  $\ell$ ).

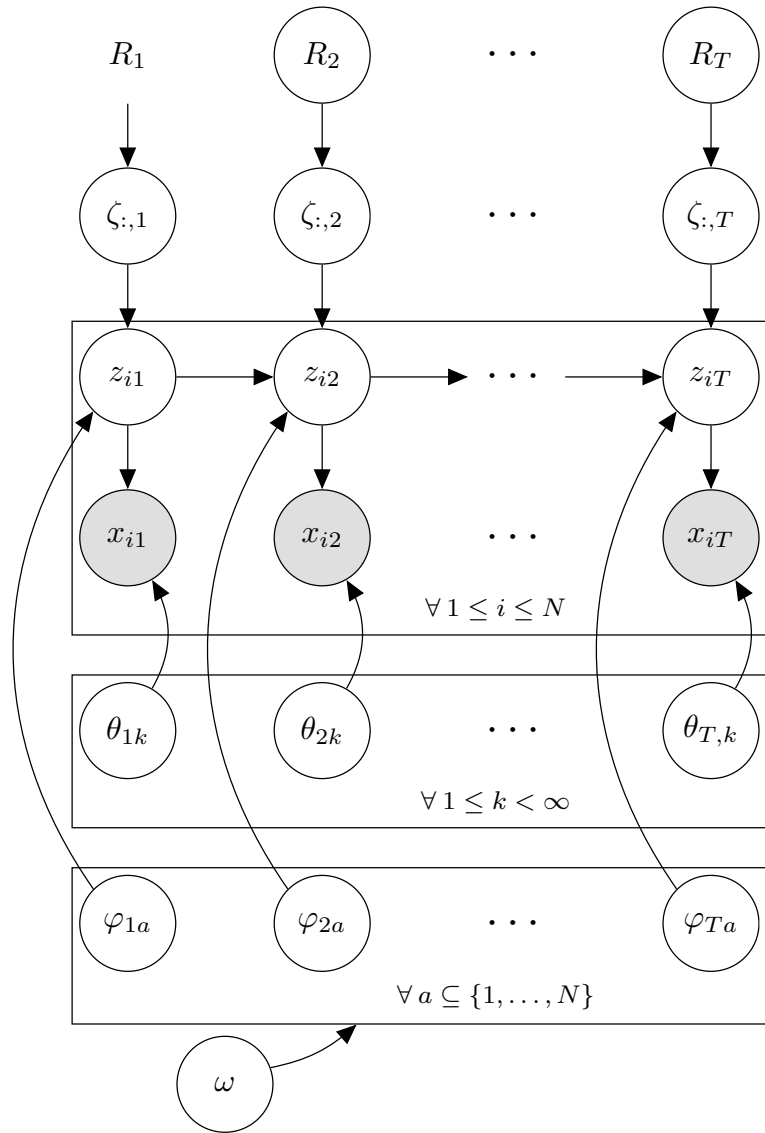
Suppose that  $\alpha_0 > 0$  and  $\alpha > 0$  are fixed concentration parameters, and the hyperparameters  $b$  and  $\gamma_\ell$  and  $\beta_\ell$  of the likelihood model are also fixed. Then, the **BNPPHASE** model is given by the following generative process:

1. Draw  $\mathcal{R}_1 \sim \text{CRP}(\{1, \dots, n\}, \alpha)$  (2.4).
2. For each  $1 < \ell \leq L$ :
  - (a) Draw  $R_\ell \subseteq \{1, \dots, n\}$  according to  $\Pr(R_\ell) = r_{\ell-1}^{\#R_\ell} (1 - r_{\ell-1})^{n - \#R_\ell}$ .
  - (b) Draw  $\mathcal{R}_\ell \sim \text{CRP}(R_\ell, \alpha)$ .
3. Draw  $\omega \sim \text{GEM}(\alpha_0)$  (2.3).
4. For each  $1 \leq \ell \leq L, a \in \mathcal{R}_\ell$ : draw  $\varphi_{\ell a}$  according to the probability density function  $\Pr(\varphi_{\ell a} = k) = \omega_k$
5. For each  $1 \leq \ell \leq L, 1 \leq k \leq K$ : draw  $\theta_{\ell k} \stackrel{\text{i.i.d.}}{\sim} \text{Beta}(\gamma_\ell \beta_\ell, \gamma_\ell (1 - \beta_\ell))$ .
6. For each  $1 \leq \ell \leq L, 1 \leq i \leq N$ :
  - (a) If  $i \in R_\ell$ : set  $\zeta_{i\ell} \leftarrow$  the unique  $a \in \mathcal{R}_\ell$  s.t.  $i \in a$  and set  $z_{i\ell} \leftarrow \varphi_{\ell a}$ .
  - (b) Otherwise: set  $\zeta_{i\ell} \leftarrow 0$  and set  $z_{i\ell} \leftarrow z_{i,\ell-1}$ .
7. For each  $1 \leq \ell \leq L, 1 \leq i \leq N$ : draw  $x_{i\ell} \stackrel{\text{i.i.d.}}{\sim} \text{Bernoulli}(\theta_{\ell, z_{i\ell}})$ .

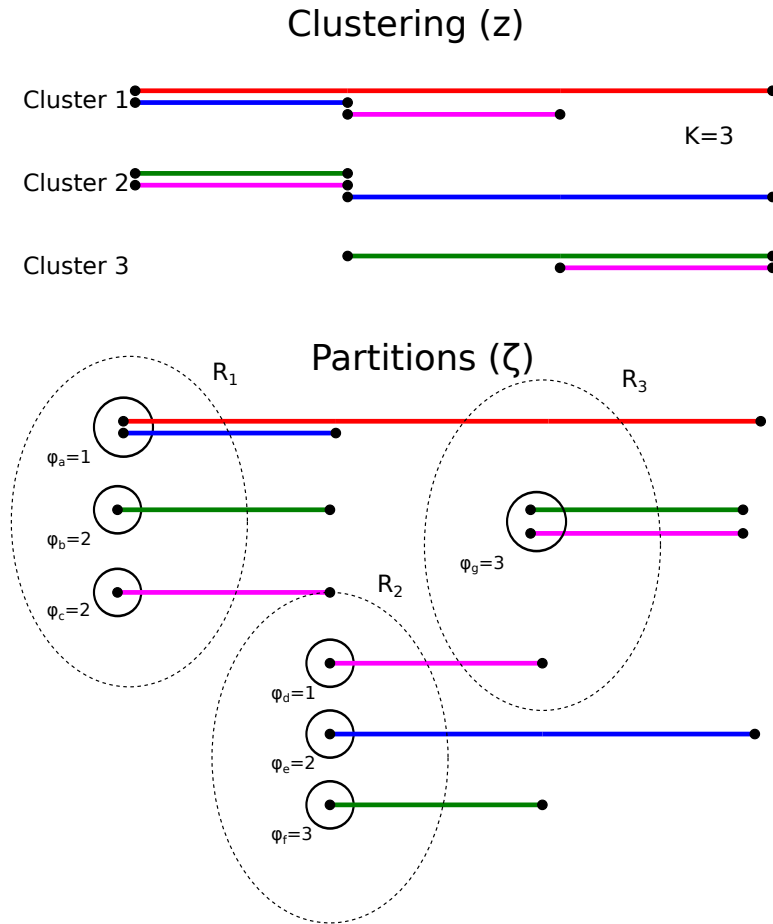
The generative process presented in this section is equivalent to the enumeration in section 3.2.1. The joint density of  $x, z, \mathcal{R}, \omega, \varphi$  and the parameters is given by the following equation:

$$\begin{aligned} & \Pr(x, z, \mathcal{R}, \omega, \varphi, \alpha_0, \alpha, b, \beta, \gamma, \theta, r) \\ &= \Pr(\omega | \alpha_0) \Pr(\varphi | \omega) \Pr(\mathcal{R}_1 | \alpha) \prod_{\ell=2}^L \Pr(\mathcal{R}_\ell | R_\ell, \alpha) r_{\ell-1}^{\#R_\ell} (1 - r_{\ell-1})^{N - \#R_\ell} \prod_{i,\ell} \Lambda(x_{i\ell} | z_{i\ell}, \theta_{\ell,:}) \\ & \cdot \Pr(\alpha) \Pr(\alpha_0) \Pr(b) \left( \prod_{\ell=1}^L \Pr(\beta_\ell | b) \Pr(\gamma_\ell) \prod_{\ell,k} \Pr(\theta_{\ell,k} | \gamma_\ell, \beta_\ell) \right) \prod_{\ell=1}^{L-1} \Pr(r_\ell). \end{aligned} \quad (3.2)$$





**Figure 3.6:** Plate diagram for marginalized version of BNPPHASE model. Support of  $\varphi_{ta}$  is blocks (subsets of  $\{1, \dots, N\}$ ) appearing in partition induced by  $\zeta_{:, \ell}$ . Parameters  $\alpha, \alpha_0, r_\ell, \gamma_{\ell k}, \beta_{\ell k}, b$ . As in 3.3 HDP parameters and hyperparameters not shown for brevity.  $T$  denotes number of markers.



**Figure 3.7:** Relationship between partition structure (*bottom*) and dynamic clustering (*top*).  $x$ -axis indicates marker position and  $y$ -axis indicates cluster or block identity. Colors indicate sequence identity. At  $\ell = 1$ ,  $\mathcal{R}_1 = \{a, b, c\}$  where  $a = \{\text{red, blue}\}$ ,  $b = \{\text{green}\}$  and  $c = \{\text{magenta}\}$ . Since  $\varphi_b = \varphi_c = 2$ , blocks  $b$  and  $c$  are in the same cluster ( $z_{\text{green},1} = z_{\text{magenta},1}$ ).

Here  $\Lambda(x_{i\ell}|z_{i\ell}, \theta_{\ell,:}) = \theta_{\ell,z_{i\ell}}^{x_{i\ell}} (1 - \theta_{\ell,z_{i\ell}})^{1-x_{i\ell}}$  is the likelihood of the observed allele from sequence  $i$  at location  $\ell$  conditioned on its cluster assignment (this likelihood is further discussed in section 3.1.3). Here, and for the remainder of the text, we adopt the MATLAB notation  $A_{b,:} = (A_{bc})_{c \in \mathcal{A}}$ , where  $\mathcal{A}$  is the support of the second index of  $A$  (and equivalently for  $A_{,c}$ ).

The variables  $\zeta$  and  $R$  are determined by  $\mathcal{R}$  (and *vice versa*) and although we have written equation (3.2) in terms of  $\mathcal{R}$ , the equivalent equations for  $\zeta$  should be clear. The dependence relationships of equation (3.2) are illustrated in the graphical model shown in Figure 3.6. A summary of all of the distributions on the parameters of the

prior is as follows:

$$\alpha_0 \sim \text{LogNormal}(\log \alpha_{0\text{mean}}, \alpha_{0\text{var}}), \quad (3.3)$$

$$\alpha \sim \text{LogNormal}(\log \alpha_{\text{mean}}, \alpha_{\text{var}}), \quad (3.4)$$

$$\text{for all } 1 \leq \ell \leq L, \gamma_\ell \sim \text{Exponential}(1), \quad (3.5)$$

$$\text{for all } 1 \leq \ell \leq L, \beta_\ell | b \sim \text{Beta}(b, b), \quad (3.6)$$

$$b \sim \text{Exponential}(1), \quad (3.7)$$

$$\text{for all } z \in \mathcal{Z}, 1 \leq \ell \leq L, \theta_{tz} | \gamma_\ell, \beta_\ell \sim \text{Beta}(\gamma_\ell \beta_\ell, \gamma_\ell (1 - \beta_\ell)), \quad (3.8)$$

$$\text{for all } 1 \leq \ell < L, r_\ell | r_{\min} \sim \text{LogUniform}(r_{\min}, 1). \quad (3.9)$$

The constants  $r_{\min} < 1$ ,  $\alpha_{0\text{mean}}$ ,  $\alpha_{0\text{var}}$ ,  $\alpha_{\text{mean}}$  and  $\alpha_{\text{var}}$  are all positive fixed real valued hyperparameters.

An example dynamic clustering demonstrating this view of the BNPPHASE model as a hierarchy with partitions at the bottom level and a Dirichlet process at the top level is given in Figure 3.7. Note that in this example, multiple different partition structures could have given rise to the dynamic clustering at the top of the figure given the right settings of  $\varphi$ . For example, if  $\mathcal{R}_1 = \{a, b'\}$  where  $a = \{\text{red}, \text{blue}\}$  as in the figure, and  $b' = \{\text{green}, \text{magenta}\}$  and  $\varphi_{b'} = 2$  then dynamic clustering would have been the same.

### 3.2.4 Marginalizing the allele emission variables $\theta$

For a fixed location  $\ell$ , we consider the conditional probability  $\Pr(x_\ell | z_\ell, \gamma_\ell, \beta_\ell)$  with the allele emission variables  $\theta_{\ell k}$  marginalized. Due to the conjugacy of the hierarchical likelihood, the conditional distribution of the observed alleles  $x$  can be expressed in terms of the allele counts of the sequences assigned to each cluster at location  $\ell$ . Let  $n_{1\ell k} = \#\{i : z_{i\ell} = k, x_{i\ell} = 1\}$  and  $n_{0\ell k} = \#\{i : z_{i\ell} = k, x_{i\ell} = 0\}$  denote the counts of the number of times each allele is observed among the sequences assigned to each cluster. Then the conditional distribution for  $x$  is given as follows:

$$\Pr(x_{:, \ell} | z_{:, \ell}, \gamma_\ell, \beta_\ell) \propto \prod_{k=1}^K \frac{\Gamma(\gamma_\ell \beta_\ell + n_{1\ell k}) \Gamma(\gamma_\ell (1 - \beta_\ell) + n_{0\ell k})}{\Gamma(\gamma_\ell + n_{1\ell k} + n_{0\ell k})}. \quad (3.10)$$

### 3.2.5 MCMC for inference and imputation

We will provide a bespoke Gibbs update for sampling the latent cluster assignment variables for a sequence (*i.e.*, the vectors  $z_{i,:}$  and  $\zeta_{i,:}$ ) conditioned on  $x$ ,  $z_{i',:}$  and  $\zeta_{i',:}$  for  $i' \neq i$  and  $\tilde{\omega}$ ,  $\varphi$  and  $\beta$  and  $\gamma$  (we will refer to these variables as ‘rest’). Following this, we will provide Gibbs updates and slice sampling updates for the HDP variables and likelihood parameters. The concatenation of all of these updates provides an MCMC kernel which leaves the posterior distribution of the BNPPHASE model invariant. Note

that in these procedures,  $\theta$  (and  $\pi_\ell$ ) will always be integrated out.

### 3.2.5.1 Gibbs update for latent cluster assignment of sequence $i$

The sequences  $z_{i,:}, \zeta_{i,:}$  will be resampled using message passing specified by a two-step scheme. In the first step,  $z_{i,:}, \zeta_{i,:}$  will be updated using a forwards-filtering/backwards-sampling (Früwirth-Schnatter, 1994) method wherein the supports for the messages for  $z_{i\ell}$  and  $\zeta_{i\ell}$  are augmented with the symbol  $\emptyset$  which represents events in which new blocks are created in the partitions  $\mathcal{R}_\ell$ . In the second step, all of the newly created blocks are assigned to components of the DP  $G_0$ . We note that this two-step scheme obviates a problem that would have arisen if we had marginalized  $\zeta_{i\ell}$  (namely, that new messages would have had to have been introduced for each partition of the set of locations  $\ell'$  such that  $z_{i\ell'} = \emptyset$ , leading to an exponentially sized support for the messages).

To find the distribution of the sequences  $z_{i,:}, \zeta_{i,:}$  conditioned on the rest of the variables, we will use the notation from section 2.4.1 to refer to the partitions induced by removing  $i$  from  $\mathcal{R}_\ell$  for  $\ell = 1, \dots, L$  and also removing any resulting empty components from  $\tilde{\omega}_1, \dots, \tilde{\omega}_K$  (recall that these are the unique elements among the assignments  $\varphi_{\ell a}$ ). By the exchangeability of the CRP, we can then assume that individual  $i$  is the last individual observed, and use the sequential scheme for the CRP and the definition of the DP to find the joint conditional distribution of the variables  $\zeta_{i,:}$  and  $z_{i,:}$ .

Suppose that we are given a fixed setting of all of the BNPPHASE latent variables and parameters including the dynamic clustering of the  $n$  individuals. This induces a dynamic clustering on the set of all of the individuals except for the  $i$ -th individual through the ‘forgetting’ of the assignments of the  $i$ -th individual. Adopting the notation from section 2.4.1, will denote the induced dynamic clustering as follows: by  $R_\ell^{-i}$  we will refer to the set consisting of  $R_\ell$  but with  $i$  removed. So, if  $\zeta_{i\ell} \neq 0$  then the  $i$ -th sequence ‘jumps’ before  $\ell$  and thus  $\ell \in R_\ell$  and in this case  $R_\ell^{-i} = R_\ell - \{i\}$ . If alternatively  $\zeta_{i\ell} = 0$  then  $i$  is not in  $R_\ell$  and  $R_\ell^{-i} = R_\ell$ .

Recall that  $\tilde{\omega}$  refers to the weights of the top-level Dirichlet process corresponding to atoms that exist among the assignments of blocks  $a \in \mathcal{R}_\ell$  to atoms (this is defined in section 2.3). By  $\tilde{\omega}^{-i}$  we will refer to the components of  $\tilde{\omega}$  that the blocks of  $\mathcal{R}_\ell^{-i}$  are assigned to (*i.e.*,  $\tilde{\omega}^{-i}$  is formed from  $\tilde{\omega}$  by removing components that appear only among the component assignments of singleton blocks  $\{i\}$  — blocks that were removed from  $\mathcal{R}_\ell$  to form  $\mathcal{R}_\ell^{-i}$ , for any  $\ell$ ).

By  $K^{-i}$  we will denote the number of distinct component assignments  $\varphi_{\ell a}$  among the blocks of the restricted partitions:  $K^{-i} = \#\{\varphi_{\ell a} : a \in \mathcal{R}_\ell^{-i}, 1 \leq \ell \leq L\}$ . So,  $K^{-i} \leq K$ , and  $K^{-i} = K$  if and only if sequence  $i$  is never in a cluster by itself among  $\mathcal{R}_1, \dots, \mathcal{R}_L$ .

Note that if a component  $\tilde{\omega}_k$  with  $k < K$  is such that the only assignments of blocks

to  $k$  involve the block  $\{i\}$ , then  $\tilde{\omega}_k$  does not appear in  $\tilde{\omega}^{-i}$ . In this case, the indices of  $\tilde{\omega}^{-i}$  are not consecutive. To avoid excessive notation, without loss of generality we will suppose that  $\tilde{\omega}$  is actually ordered such that the indices are consecutive and  $\tilde{\omega}^{-i} = (\tilde{\omega}^{-i}, \dots, \tilde{\omega}_{K-i}^{-i})$ .

We will now consider the possible events that could occur when sequences  $z_{i,:}, \zeta_{i,:}$  are sampled. We will augment the state space of  $z_{i,:}, \zeta_{i,:}$  with the symbol  $\emptyset$  and we will denote the event that sequence  $i$  joins a singleton block at location  $\ell$  by  $\zeta_{i\ell} = \emptyset$ . In that case, the component assignment for that block will be a component that already exists in  $\tilde{\omega}^{-i}$ , (the  $k$ -th component, say) which we will denote by  $z_{i\ell} = k \geq 1$ , or that sequence is in cluster by itself which we will denote by  $z_{i\ell} = \emptyset$ . Conditioning on the set ‘rest’ =  $\{x, \mathcal{R}_{\cdot}^{-i}, (\zeta_{i'})_{i' \neq i}, \tilde{\omega}^{-i}, \alpha_0, \alpha, b, \beta, \gamma, \theta, r\}$ , the distribution of  $z_{i,:}, \zeta_{i,:}$  is given in the following display. Note that since  $\mathcal{R}$ , and  $\zeta$  are completely determined by  $z$  and  $y$  (and *vice versa*), conditioning on  $\mathcal{R}_{\cdot}^{-i}, (\zeta_{i'})_{i' \neq i}$  is equivalent to conditioning on all of the variables  $\mathcal{R}_{\cdot}^{-i}$  and  $(\zeta_{i'})_{i' \neq i}$ .

$$\Pr(z_{i,:}, \zeta_{i,:} | \text{‘rest’}) = J_1(z_{i1}, \zeta_{i1}) \prod_{\ell=2}^L \begin{cases} r_{\ell-1} J_{\ell}(z_{i\ell}, \zeta_{i\ell}) & \text{if } \zeta_{i\ell} \neq \emptyset, \varphi_{\ell \zeta_{i\ell}} = z_{i\ell}, \\ 1 - r_{\ell-1} & \text{if } \zeta_{i\ell} = \emptyset, 1 \leq z_{i\ell} \leq K^{-i}, \\ 0 & \text{otherwise.} \end{cases} \cdot \prod_{\ell=1}^L \Lambda(x_{i\ell} | z_{i\ell}, x, \mathcal{R}_{\ell}^{-i}). \quad (3.11)$$

$$\text{Where } J_{\ell}(z_{i\ell}, \zeta_{i\ell}) = \frac{1}{\#\mathcal{R}_{\ell}^{-i} + \alpha} \cdot \begin{cases} \#\zeta & \zeta_{i\ell} = \zeta \in \mathcal{R}_{\ell}^{-i}, \\ \alpha \tilde{\omega}_z & \zeta_{i\ell} = \emptyset, z_{i\ell} = z \in \mathcal{Z}^{-i}, \\ \alpha \tilde{\omega}_{\emptyset} & \zeta_{i\ell} = z_{i\ell} = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \quad (3.12)$$

Here  $J_{\ell}(z, \zeta)$  it is the prior distribution over  $(z_{i\ell}, \zeta_{i\ell})$  given that sequence  $i$  ‘jumps’ immediately before location  $\ell$ . The marginalized likelihood  $\Lambda(x_{i\ell} | z_{i\ell}, x, \mathcal{R}_{\ell}^{-i})$  is found by restricting equation (3.10) to  $i$ : let  $n_{1\ell k}^{-i} = \#\{i' : z_{i'\ell} = k, i' \neq i, x_{i'\ell} = 1\}$  and let  $n_{0\ell k}^{-i} = \#\{i' : z_{i'\ell} = k, i' \neq i, x_{i'\ell} = 0\}$  denote the counts of the number of times each allele is observed for the cluster  $k$  at location  $\ell$  among sequences other than the  $i$ -th sequence. If  $x_{i\ell}$  is unobserved (*i.e.*,  $x_{i\ell} = \text{‘?’}$ ) then  $\Lambda(x_{i\ell} | z_{i\ell} = k, x, \mathcal{R}_{\ell}^{-i}) = 1$  for all  $k$ . If  $x \neq \text{‘?’}$  (*i.e.*,  $x \in \{1, 0\}$ ) and  $1 \leq k \leq K$  then:

$$\Lambda(x_{i\ell} | z_{i\ell} = k, x, \mathcal{R}_{\ell}^{-i}) = \frac{1}{\gamma_{\ell} + n_{1\ell k}^{-i} + n_{0\ell k}^{-i}} \begin{cases} \gamma_{\ell} \beta_{\ell} + n_{1\ell k}^{-i} & \text{if } x_{i\ell} = 1, \\ \gamma_{\ell} (1 - \beta_{\ell}) + n_{0\ell k}^{-i} & \text{if } x_{i\ell} = 0. \end{cases} \quad (3.13)$$

Finally, if  $z_{i\ell} = \emptyset$  and  $x \in \{1, 0\}$  then  $\Lambda(x_{i\ell} = x | z_{i\ell} = \emptyset, x, \mathcal{R}_{\ell}^{-i}) = \beta_{\ell}^x (1 - \beta_{\ell})^{1-x}$ .

We now present a Gibbs update for the  $i$ -th sequence based on the distribution in equation (3.11) for  $z_{i,:}, \zeta_{i,:}$  conditioned on the variables ‘rest’. First, in Step 1 we will conduct forwards-filtering/backwards-sampling on  $z_i, \zeta_i$  with the augmented state space described in this section. Then, in Step 2, for all  $\ell$  with  $z_{i\ell} = \emptyset$  or  $\zeta_{i\ell} = \emptyset$ , we assign

new clusters through a retrospective stick breaking construction.

### Step 1: forwards-filtering/backwards-sampling

The forwards messages will be used in the forwards-filtering/backwards-sampling algorithm and the backwards messages will be used to compute marginal probabilities of an allele for imputation of missing data. Since  $i$  is fixed, for the rest of the specification of Step 1 the subscript  $i$  will be suppressed to make the notation more compact (so for example, by  $z_\ell$  we will mean  $z_{i\ell}$ ). A glossary of symbols for the BNPPhase model is provided at the end of this Chapter.

The messages are defined as follows:

$$\begin{aligned} m_1^f(z_1, \zeta_1) &= \Pr(x_1, z_1, \zeta_1 | \text{'rest'}), \\ \text{For } 1 < \ell \leq L, m_\ell^f(z_\ell, \zeta_\ell) &= \Pr(x_1 \dots x_\ell, z_\ell, \zeta_\ell | \text{'rest'}), \\ \text{For } 1 \leq \ell < L, m_\ell^b(z_\ell, \zeta_\ell) &= \Pr(x_{\ell+1} \dots x_L | z_\ell, \zeta_\ell, \text{'rest'}), \\ m_L^b(z_L, \zeta_L) &= 1. \end{aligned} \tag{3.14}$$

Here the set of variables referred to as ‘rest’ is the same as that given in the paragraph before equation (3.11). For each of these messages, if  $\ell > 1$  the support of  $(z_\ell, \zeta_\ell)$  is given by the union of the following three sets:

$$\{(z, \zeta) : z \in \{1, \dots, K^{-i}, \emptyset\}, \zeta = 0\}, \tag{3.15}$$

$$\{(z, \zeta) : \zeta \in \mathcal{R}_\ell^{-i}, z = \varphi_{\ell\zeta}\}, \tag{3.16}$$

$$\{(z, \zeta) : \zeta = \emptyset, z \in \{1, \dots, K^{-i}, \emptyset\}\}. \tag{3.17}$$

These three sets describe the three possible types of allowable cluster assignments described in section 3.2.5.1. These three sets represent the events that (3.15): individual  $i$  does not ‘jump’, (3.16): individual  $i$  ‘jumps’ to a cluster in  $\mathcal{R}^{-i}$ , and (3.17): individual  $i$  ‘jumps’ to a new cluster by itself. The probabilities of settings of  $(z, \zeta)$  that lie outside of this support are zero. We will refer to the union of these three sets by  $\text{sup}(\ell)$ . Note that if  $\ell = 1$ ,  $\text{sup}(\ell)$  is given by the union of sets (3.16) and (3.17) only.

For the backwards messages, if we condition on  $z_\ell$  then  $\zeta_\ell$  and  $x_{\ell+1}, \dots, x_L$  are independent, and so for a fixed  $z$  the messages  $m_\ell^b(z, \zeta)$  all have the same value for each  $\zeta : (z, \zeta) \in \text{sup}(\ell)$  and therefore we will refer to this value by  $m_\ell^b(z)$ . Further, we will often find it useful to sum the forwards messages over the possible values of their

parameters and so we will introduce the following shorthand notation:

$$\mathbf{m}_\ell^f = \sum_{(z,\zeta) \in \text{sup}(\ell)} \mathbf{m}_\ell^f(z, \zeta), \quad (3.18)$$

$$\text{and for a fixed } z \text{ with } 0 \leq z \leq K^{-i}, \mathbf{m}_\ell^f(z) = \sum_{\substack{(z',\zeta) \in \text{sup}(\ell), \\ z'=z}} \mathbf{m}_\ell^f(z', \zeta). \quad (3.19)$$

The forwards messages in display (3.14) can be computed recursively as follows:

$$\begin{aligned} \mathbf{m}_1^f(z_1, \zeta_1) &= L(x_1|z_1)P_1(z_1, \zeta_1), \\ \text{and for } 1 < \ell \leq L, \mathbf{m}_\ell^f(z_\ell, \zeta_\ell) &= L(x_\ell|z_\ell) \cdot \begin{cases} (1 - r_{\ell-1}) \cdot \mathbf{m}_{\ell-1}^f(z_\ell) & \text{if } \zeta_\ell = 0, \\ r_{\ell-1} \cdot \mathbf{m}_{\ell-1}^f \cdot P_\ell(z_\ell, \zeta_\ell) & \text{otherwise.} \end{cases} \end{aligned} \quad (3.20)$$

The probability  $P_\ell(z_\ell, \zeta_\ell)$  is  $\Pr(z_\ell, \zeta_\ell | \text{'rest'}, \zeta_\ell \neq 0)$  as in equation (3.12). In a similar way, the backwards messages can also be computed recursively as follows:

$$\mathbf{m}_\ell^b(z_\ell) = (1 - r_\ell)L(x_{\ell+1}|z_\ell)\mathbf{m}_{\ell+1}^b(z_\ell) + r_\ell \sum_{(z,\zeta) \in \text{sup}(\ell+1)} L(x_{\ell+1}|z)P_{\ell+1}(z, \zeta)\mathbf{m}_{\ell+1}^b(z). \quad (3.21)$$

After computing the forwards messages, the cluster assignments for the  $i$ -th sequence can be sampled through a backwards-sampling algorithm. By Bayes rule, the Markov property of the cluster assignments and the definition of the forwards messages, the probabilities are as follows:

$$\begin{aligned} \Pr(x, z_L, \zeta_L | \text{'rest'}) &\propto \mathbf{m}_L^f(z_L, \zeta_L) \\ \Pr(x, z_\ell, \zeta_\ell | z_{\ell+1}, \zeta_{\ell+1}, \text{'rest'}) &\propto \Pr(x_1, \dots, x_\ell, z_\ell, \zeta_\ell | \text{'rest'}) \Pr(z_{\ell+1}, \zeta_{\ell+1} | z_\ell, \zeta_\ell, \text{'rest'}), \\ &\propto \mathbf{m}_\ell^f(z_\ell, \zeta_\ell) \cdot \begin{cases} \delta(z_\ell = z_{\ell+1}) & \text{if } \zeta_{\ell+1} = 0, \\ 1 & \text{otherwise.} \end{cases} \end{aligned} \quad (3.22)$$

In equation (3.22), the domain of  $z_\ell, \zeta_\ell$  is always restricted to  $\text{sup}(\ell)$ . Step 1 of the Gibbs update for  $z_i, \zeta_i$  is thus given by sampling  $z_\ell, \zeta_\ell$ , recursively in descending order ( $\ell = L, \dots, 1$ ) using the probabilities given in equation (3.22).

## Step 2: retrospective stick breaking

We now provide a retrospective stick breaking scheme to select the components for the singleton blocks which were sampled in Step 1 but whose assigned components were not in  $\tilde{\omega}^{-i}$ . That is, we will now sample the values  $z_{i\ell}$  for all of the locations  $\ell$  such that after Step 1,  $z_{i\ell} = \emptyset$ . We will refer to such  $\ell$  by the set  $S_\emptyset = \{\ell : z_{i\ell} = \emptyset\}$ . For a given setting of  $z_{i,\cdot}, \zeta_{i,\cdot}$  sampled using the backwards-sampling from Step 1,  $S_\emptyset$  is found deterministically. Applying Step 1 followed by Step 2 yields a full Gibbs update for  $z_{i,\cdot}, \zeta_{i,\cdot}$ .

By the definition of the symbol  $\emptyset$  from section 3.2.5.1, the variables  $z_{i\ell} : \ell \in S_\emptyset$  should only be assigned to components of the DP  $\omega$  that none of the other block in  $\mathcal{R}_i^{-i}$  are assigned to. It is, however, possible for more than one  $z_{i\ell} : \ell \in S_\emptyset$  to be assigned to the same component. For each  $\ell \in S_\emptyset$ ,  $z_{i\ell}$  marginally follows the law  $\Pr(z_{i\ell} | \text{'rest'}, z_{i\ell} \notin \omega^{-i})$ . Since  $i$  is fixed, for a fixed location  $\ell$  there is at most one  $z_{i\ell}$  that needs to be sampled for  $\ell$ , and so the allele counts  $n_{1tk}^{-i}, n_{0tk}^{-i}$  are conditionally independent (given 'rest') of the random variables  $z_{it'} : t' \neq \ell$ . Further, because  $\zeta_{i\ell} \neq 0$  for all  $\ell \in S_\emptyset$ , the  $z_{i\ell} : \ell \in S_\emptyset$  are independent conditioned on the weights  $\tilde{\omega}$ . Combining these two observations, it is clear that the variables  $z_{i\ell} : \ell \in S_\emptyset$  are sampled *i.i.d.* directly from the prior, but conditioned on the event that  $z_{i\ell} \notin \mathcal{R}_i^{-i}$ . This can be done by using the stick breaking construction in equation (2.3) to instantiate components of  $\omega$  that are not in  $\tilde{\omega}^{-i}$  (we will refer to these components by  $\tilde{\omega}_{\emptyset 1}, \tilde{\omega}_{\emptyset 2}, \dots$ ) and then sampling  $z_{i\ell} : \ell \in S_\emptyset$  from the GEM distribution  $\omega$  restricted to these new components. This can be done efficiently by sampling a uniform variate  $u_\ell$  *i.i.d.* for each  $\ell \in S_\emptyset$  and then setting  $z_{i\ell}$  to the smallest  $k$  such that:

$$\frac{\sum_{k'=1}^k \tilde{\omega}_{\emptyset k'}}{\tilde{\omega}_\emptyset} > u_\ell \quad (3.23)$$

With this scheme,  $\tilde{\omega}_{\emptyset 1}, \tilde{\omega}_{\emptyset 2}, \dots$  can be sampled in sequence, stopping as soon as equation (3.23) is satisfied for all  $\ell \in S_\emptyset$ . Step 2 is made explicit in the following algorithm, which should be performed immediately after sampling  $z_{i,\cdot}, \zeta_{i,\cdot}$  according to Step 1. The

---

**Algorithm 3.1** Retrospective stick breaking for the BNPPhase model

---

1. Set  $S_\emptyset \leftarrow \{\ell : z_{i\ell} = \emptyset\}$ .
  2. Set  $\omega'_{\emptyset,\cdot} \leftarrow ()$  and set  $K'_\emptyset \leftarrow 0$ .
  3. For each  $\ell \in S_\emptyset$ :
    - (a) Set  $\mathcal{R}_\ell \leftarrow \mathcal{R}_\ell^{-i} \cup \{\{i\}\}$  and set  $\zeta_{i\ell} \leftarrow \{i\}$ .
    - (b) Draw  $u \sim \text{Uniform}(0, 1)$ .
    - (c) While  $k^* = \min\{1 \leq k \leq K'_\emptyset : \sum_{k'=1}^k \tilde{\omega}_{\emptyset k'} > u\}$  does not exist:
      - i. Set  $K'_\emptyset \leftarrow K'_\emptyset + 1$ .
      - ii. Draw  $\nu \sim \text{Beta}(1, \alpha)$ .
      - iii. Set  $\omega'_{\emptyset, K'_\emptyset} \leftarrow \nu \left(1 - \sum_{k=1}^{K'_\emptyset-1} \omega'_{\emptyset k}\right)$ .
    - (d) Set  $z_{i\ell} \leftarrow k^*$ .
  4. Set  $\tilde{\omega}_{\emptyset,\cdot} \leftarrow ()$  and set  $K_\emptyset \leftarrow 0$ .
  5. For  $k' = 1$  to  $K'_\emptyset$ :
    - (a) If there exists  $\ell \in S_\emptyset$  such that  $z_{i\ell} = k'$ :
    - (b) Set  $S \leftarrow \{\ell \in S_\emptyset : z_{i\ell} = k'\}$ .
    - (c) If  $\#S > 0$ :
      - i. Set  $K_\emptyset \leftarrow K_\emptyset + 1$ .
      - ii. Set  $z_{i\ell} \leftarrow K^{-i} + K_\emptyset$  for each  $\ell \in S$ .
      - iii. Set  $\tilde{\omega}_{\emptyset, K_\emptyset} \leftarrow \tilde{\omega}_\emptyset \cdot \omega'_{\emptyset, k'}$ .
      - iv. Set  $S_\emptyset \leftarrow S_\emptyset \setminus S$ .
  6. Set  $\tilde{\omega} \leftarrow ((\tilde{\omega}_k^{-i})_{k=1}^{K^{-i}}, (\tilde{\omega}_{\emptyset, k})_{k=1}^{K_\emptyset})$  and set  $K \leftarrow K^{-i} + K_\emptyset$ .
  7. Set  $K \leftarrow K + K_\emptyset$ .
  8. Set  $\tilde{\omega}_\emptyset \leftarrow 1 - \sum_{k=1}^K \tilde{\omega}_k$ .
-



concatenation of Step 1 and Step 2 provides a full Gibbs update for the latent cluster assignment of the  $i$ -th sequence. The details of Step 2 are provided in Algorithm 3.1.

If, for a fixed  $i$ , the allele  $x_{i\ell}$  is unobserved at some fixed location  $\ell$  (*i.e.*,  $x_{i\ell} = '?'$ ), then we can use the messages defined in this section to compute the marginal probability  $\mathbb{E}[\Pr(x_{i\ell} = x, (x_{i'\ell'})_{i'\ell' \neq i\ell} | \text{'rest'})]$  where  $x = 0$  or  $1$ . Here, we will use both the forward and backward messages calculated in Step 1 to marginalize over all possible latent state assignments of sequence  $i$  at location  $\ell$ . Using the definition of the messages, the Markov property for the sequences  $z_{i\cdot}, \zeta_{i\cdot}$  and the likelihood from equation (3.13) we have:

$$\begin{aligned} & \mathbb{E}_{z_i, \zeta_i} [\Pr(x_{i\ell} = x, (x_{i'\ell'})_{i'\ell' \neq i\ell} | \text{'rest'})] \\ \propto & \sum_{(z, \zeta) \in \text{sup}(\ell)} \Pr(x_{i\ell} = x | z_{i\ell} = z, \text{'rest'}) \Pr(x_{i\ell} = x, (x_{i'\ell'})_{i'\ell' \neq i\ell, \ell' \leq \ell}, z_{i\ell} = z, \zeta_{i\ell} = \zeta | \text{'rest'}) \\ & \cdot \Pr((x_{i'\ell'})_{\ell' > \ell} | z_{i\ell} = z, \zeta_{i\ell} = \zeta, \text{'rest'}), \\ = & \sum_{(z, \zeta) \in \text{sup}(\ell)} L(x_{i\ell} = x | z_{i\ell}) m_\ell^f(z, \zeta) m_\ell^b(z, \zeta). \end{aligned} \quad (3.24)$$

This expression can be used to impute missing alleles in a set of partially observed genetic sequences.

### 3.2.5.2 Gibbs updates for HDP parameters $\tilde{\omega}$ and $\varphi$

We will now derive MCMC updates for the component weights  $\tilde{\omega}$  of the  $K$  distinct components appearing in  $(\varphi_{\ell a})_{a \in \mathcal{R}_\ell}$  and  $\tilde{\omega}_\emptyset$ . We will use a Gibbs sampling scheme based on the definition of the Dirichlet process in definition (2.2). Conditioned on  $\varphi_{\ell a}$  for all  $\ell$ ,  $\mathcal{R}_\ell$  and  $\alpha_0$  (which we will refer to as ‘rest’) the distribution of  $\tilde{\omega}_1, \dots, \tilde{\omega}_K, \tilde{\omega}_\emptyset$  is given by the Dirichlet distribution which can be readily sampled.

$$((\tilde{\omega}_k)_{k=1}^K, \tilde{\omega}_\emptyset) | \varphi, \alpha_0 \sim \text{Dirichlet}(\{(\#\{(\ell, a) : 1 \leq \ell \leq L, a \in \mathcal{R}_\ell, \varphi_{\ell a} = k\})_{k=1}^K, \alpha_0\}). \quad (3.25)$$

We update the component assignment of a block  $a \in \mathcal{R}_\ell$  using Gibbs sampling by examining equation (3.2). We find that the entries that depend on the assignment  $\varphi_{\ell a}$  for  $1 \leq \ell \leq L$  and  $a \in \mathcal{R}_\ell$  are given for each sequence  $i$  such that  $i \in a$  by examining the extent of that sequence. In particular, for each  $i$ , and for each  $\ell' > \ell$ , the component assignment  $\varphi_{\ell' a}$  only depends on  $\zeta_{i\ell}, z_{i\ell}$  and  $x_{i\ell}$  if sequence  $i$  does not jump between  $\ell$  and  $\ell'$ . We define this set to be  $E = \{(i, \ell') : \ell' > \ell, i \in a, \zeta_{i\tau} = 0 \forall \ell' \geq \tau > \ell\}$ . With this definition, conditioned on the variables  $x, \mathcal{R}_\ell, \tilde{\omega}, \gamma, \beta, \varphi_{\ell' a'}$  for  $a' \neq a$  (which we will refer to as ‘rest’) the joint distribution of  $\varphi_{\ell' a}$  is as follows:

$$\Pr(\varphi_{\ell' a} = z | \text{'rest'}) \propto \tilde{\omega}_z \Lambda(x | z_{i\ell'} = z \forall (i, \ell') \in E). \quad (3.26)$$

The likelihood term in equation (3.26) is found through (3.10). Note that equation (3.26) does not factorize over  $E$  because  $\theta$  is marginalized. This likelihood term, for a fixed  $\ell$  and  $a \in \mathcal{R}_\ell$  is as follows:

$$\Lambda(x|z_{it'}=z \forall (i, \ell') \in E) \propto \prod_{\ell'=\ell}^L \frac{\Gamma(\gamma_\ell \beta_\ell + n_{1t'}^E + n_{1t'k}^{-E}) \Gamma(\gamma_\ell (1 - \beta_\ell) + n_{0t'}^E + n_{0t'k}^{-E})}{\Gamma(\gamma_\ell \beta_\ell + n_{1t'z}^{-E}) \Gamma(\gamma_\ell (1 - \beta_\ell) + n_{0t'z}^{-E})}. \quad (3.27)$$

Here  $n_{1t'}^E = \#\{(i, \ell') \in E : x_{it'} = 1\}$  and  $n_{0t'}^E = \#\{(i, \ell') \in E : x_{i\ell} = 0\}$  are the allele counts for the sequences in  $a$  that do not jump between  $\ell$  and  $\ell'$ . Similarly,  $n_{1t'z}^{-E} = \#\{i : (i, \ell') \notin E, z_{it'} = z, x_{it'} = 1\}$  and  $n_{0t'z}^{-E} = \#\{i : (i, \ell') \notin E, z_{it'} = z, x_{it'} = 1\}$  are the allele counts for the sequences in cluster  $z \in \{1, \dots, K, \emptyset\}$  at  $\ell'$  that are not in  $a$  or that jump between  $\ell$  and  $\ell'$ . Note that if  $z = \emptyset$ ,  $n_{1t'\emptyset}^{-E} = n_{0t'\emptyset}^{-E} = 0$  for all  $\ell', E$ . With these definitions, (3.26) can be computed for each  $\ell$  and  $a \in \mathcal{R}_\ell$  providing Gibbs updates for  $\varphi_{ta}$ . In this update, if  $z = \emptyset$  is sampled then a new component is added to  $\tilde{\omega}$  and alternatively if  $\varphi_{ta}$  was the only component assignment with  $\varphi_{ta} = z$  and  $\varphi_{ta}$  is sampled such that  $z \neq \emptyset$  then a component is removed from  $\tilde{\omega}$ .

### 3.2.5.3 Slice sampling for parameters $\alpha_0, \alpha, \gamma_\ell, \beta_\ell, b$ and $r_\ell$

Slice sampling provides efficient updates for random variables with distributions known only up to a normalizing constant without requiring a choice of proposal distribution or step size. We will update the latent variables  $\alpha, \alpha_0, \gamma_\ell, \beta_\ell, b$  and  $r_\ell$  using slice sampling. The unnormalized probability density functions of these variables are given in this section. In order to specify a slice sampler, we only need to know the target conditional distribution up to a normalizing constant. In this section, we provide such unnormalized conditional distributions for these latent variables. For more detail on slice sampling, we refer to Neal (2003).

#### Conditional distributions for $\alpha_0$ and $\alpha$

These distributions follow from the priors in equations (3.4), (3.3) and the CRP marginals for  $\mathcal{R}_\ell$  and the definition of the DP (Pitman, 2006).

$$\Pr(\alpha_0 | \mathcal{R}, \varphi, K) \propto \Pr(\alpha_0) \alpha_0^K \Gamma(\alpha_0) \Big/ \Gamma \left( \alpha_0 + \sum_{\ell, k} \#\{a \in \mathcal{R}_\ell : \varphi_{ta} = k\} \right), \quad (3.28)$$

$$\Pr(\alpha | \mathcal{R}) \propto \Pr(\alpha) \prod_{\ell=1}^L \frac{\alpha^{\#\mathcal{R}_\ell} \Gamma(\alpha)}{\Gamma(\alpha + \#\mathcal{R}_\ell)}. \quad (3.29)$$

### Conditional distributions for $\gamma_\ell, \beta_\ell, b$ and $r_\ell$

By Bayes' rule the unnormalized conditional distributions for  $\gamma_\ell, \beta_\ell$  and  $b$  can be read off from the conditional likelihood in equation (3.10) and the priors (3.5), (3.6), (3.7) thus providing slice sampling updates. Finally, a slice sampling update for  $r_\ell$  is provided by the following conditional distribution:

$$\text{For } 1 \leq \ell < L, \Pr(r_\ell | R_\ell) \propto \Pr(r_\ell) r_\ell^{\#R_{\ell+1}} (1 - r_\ell)^{N - \#R_{\ell+1}}. \quad (3.30)$$

Here  $\Pr(r_\ell)$  is the prior on  $r_\ell$  from equation (3.9).

### 3.2.6 Summary

The methodology developed in this section is similar to that of the beam sampling for HDP-HMMs (Van Gael et al., 2008). Both of these procedures use forwards-filtering/backwards-sampling to provide updates for entire rows of state assignments. However, in the beam sampler, an additional auxiliary variable ( $u$ ) is introduced representing a lower bound on the cluster weights. The beam sampler then provides an auxiliary scheme in which the cluster weights (which are denoted here by  $\tilde{\omega}$ ), the cluster assignments (denoted here by  $\zeta, z$ ) and the lower bound on the cluster weights ( $u$ ) are alternately resampled. Unlike the general homogeneous situation examined by the beam sampler, the structure of the dependence in the BNPPHASE model allows exact Gibbs updates without introducing the lower bound  $u$ . This provides better mixing for the MCMC algorithm.

These inference methods also strictly improve upon the sticky-HMM methods originally proposed in Fox et al. (2011) wherein only Gibbs updates for the state assignments at a single location were considered. In other work, split/merge updates have been derived for sticky-HMMs and related models (Michael et al., 2012). It is likely that incorporation of that type of update for the BNPPHASE model would be beneficial.

To conclude this section, in summary we have provided a full MCMC algorithm for BNPPHASE through an auxiliary Gibbs update for the latent cluster assignment of a fixed sequence along with slice sampling updates for the parameters. The MCMC algorithm we have provided is a collapsed sampler that operates directly on the dynamic-clustering of the sequences. Imputation of missing data from partially observed genetic sequence data can be done by simulating the posterior distribution of the BNPPHASE model using this MCMC algorithm, and marginalizing the allele emissions at the missing entries.

### 3.3 Relationship to the FastPHASE model

#### 3.3.1 Finite truncations of the BNPPHASE model

Suppose that there are at most  $K$  clusters at each location. Then, the prior induced by BNPPHASE on clustering reduces to the following finite form:

$$\Pr(x, y, z, \omega_1, \dots, \omega_K) = \Pr(\omega_1, \dots, \omega_K) \prod_i \omega_{1, z_{i1}} \prod_{i, \ell > 1} \begin{cases} r_{\ell-1} \omega_{\ell, z_{i\ell}} y_{i, \ell-1} = 1, \\ 1 - r_{\ell-1} y_{i, \ell-1} = 0, z_{i\ell} = z_{i, \ell-1}, \\ 0 \quad \text{otherwise.} \end{cases}$$

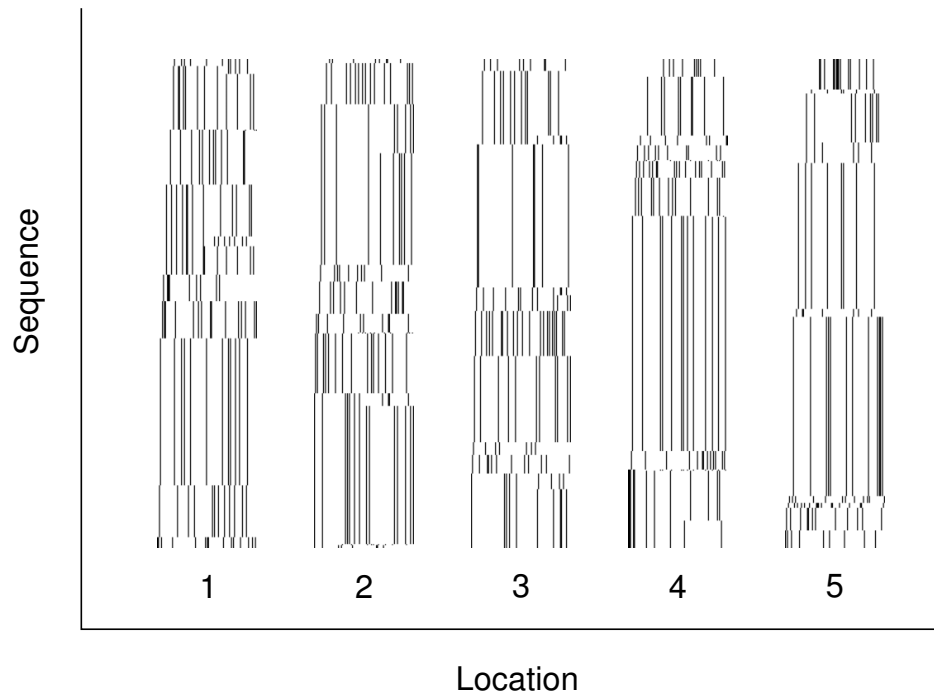
This is a Bayesian version of fastPHASE (Scheet and Stephens, 2006) with a Bayesian prior  $\Pr(\omega_1, \dots, \omega_K)$  on  $\omega_1, \dots, \omega_K$ . The BNPPHASE model is an extension of this equation in which  $K \rightarrow \infty$ .

#### 3.3.2 Non-reversibility of fastPHASE and related models

The fastPHASE and BNPPHASE models involve latent location-dependent parameters. Conditioned on these parameters, the models are not reversible. In particular, if we reverse the order of both the parameters and the data (by the transformation  $\ell \mapsto L - \ell$ ), then the posterior distribution of the data will be different. Furthermore, the marginal distribution of the partition of the items of the BNPPHASE model for  $\ell > 1$  is not given by the CRP. Both of these remarks originate from ‘sticky’ nature of the process wherein sequences only leave the cluster they are in independently with rate  $r$ .

These remarks can be illustrated through the following example. Suppose that  $L = 2$ , and that the atomic weights of the Dirichlet processes  $G_1 \neq G_2$  are given by  $\pi_{1k}$  and  $\pi_{2k}$  and the locations of the atoms for both processes given by  $\psi_1, \psi_2, \dots$ . Then, as  $r \rightarrow 0$ , the probability that a sequence is assigned to the  $k$ -th cluster at location  $\ell = 2$  (*i.e.*, that  $z_{i2} = k$ ) converges to  $\pi_{1k}$ . On the other hand, this probability converges to  $\pi_{2k}$  as  $r \rightarrow 1$ . We see from this illustration that the prior for the cluster assignment of the  $i$ -th sequence at the  $\ell$ -th location is affected by the parameters to the left of  $\ell$  (but not to the right), yielding non-reversibility.

The particular form for the marginal probability that the  $i$ -th sequence is in cluster  $k$  at location  $\ell$  is given by a mixture of the Dirichlet processes  $G_1, \dots, G_\ell$ . The weight of  $G_{\ell'}$  in this mixture for  $\ell' \leq \ell$  is given by the probability that the  $i$ -th sequence transitions at  $\ell'$  but does not transition at any of the steps between  $\ell'$  and  $\ell$ . Thus, the mixture is  $(1 - r)^\ell G_1 + \sum_{\ell'=2}^{\ell} (1 - r)^{\ell-\ell'} r G_{\ell'}$ . Since Dirichlet processes are not closed under mixtures, these mixtures are not Dirichlet processes and so the induced clustering of the sequences at  $\ell$  is not a CRP.



**Figure 3.8:** Simulated ‘toy’ data using the identity-by-descent paradigm. The first 5 datasets from the 100 simulated datasets with  $K = 9$  are shown.  $x$ -axis indicates marker position on the chromosome and  $y$ -axis indicates the sequence identity. The sequences are sorted in lexicographical order (black indicates the major allele). Since mutations are placed on the  $K = 9$  founders, ancestral recombination can be seen clearly from the patterns in the data.

### 3.4 Experiments

We conducted three experiments in which we compared the BNPPHASE model (presented in this Chapter), `fastPHASE` and various other baselines. In our first experiment, we examined the performance of the `fastPHASE` model on simulated data. We conducted imputation on held out data from simulated population bottlenecks. We simulated 700 datasets using an identity by descent paradigm. In this paradigm, we simulated the ARG backwards in time for 500 lineages until the lineages coalesced into a fixed number  $K$  lineages. We assumed that the time of coalescence at  $K$  lineages was the bottleneck time. We varied  $K$  from 4 to 10 (for a total of 7 different conditions for  $K$ ) and generated 100 datasets for each setting of  $K$ . Then, rather than placing mutations according to the infinite sites model described in section 1.2.2 (*i.e.*, by placing mutations at points chosen with intensity given by the total tree length of the genealogies) we instead placed  $L = 100$  mutations at the time of coalescence into  $K$  lineages. In this way, we constructed ‘toy’ datasets in which the founder effect was amplified through an identity-by-descent. This paradigm creates a characteristic form of structure in the data in which the dynamic-clustering is obvious. Examples of this data are given in Figure 3.8. For each dataset, we held out 50% of entries uniformly at random from all pairs of individuals and locations and then imputed the held out data using BNPPHASE and `fastPHASE`.

In another examination of simulated data, we generated data from the prior of the BNPPHASE model, and recorded the runtime of posterior simulation conditioned on this data. The parameters used for generating the data were as follows:  $\alpha_{0\text{mean}} = 10.0$ ,  $\alpha_{0\text{var}} = 1.0$ ,  $\alpha_{\text{mean}} = 1.0$ ,  $\alpha_{\text{var}} = 1.0$  and  $r_{\text{min}} = 10^{-5}$ . In this runtime experiment, we varied the number of individuals between 100 and 900 and varied the number of sites between 100 and 900. For the trials in which the number of individuals were varied, we fixed the number of sites at 200 (and *visa versa* for the trials in which the number of sites were varied). For each combination of sites and individuals, we conducted 10 trials of 200 MCMC iterations each, and recorded the runtime of each trial.

In our second experiment, we used parameters from [Li and Durbin \(2011\)](#) to simulate data designed to model the out-of-Africa bottleneck in humans. We simulated 500 phased genetic sequences on 20 independent chromosome regions. Each region was on average  $3.0 \times 10^5$  base pairs long. There were on average 2099.3 biallelic markers in each region. We recorded the time to most recent common ancestor (TMRCA) of each biallelic marker under the simulation and conducted inference of the latent clustering structure of the fully observed bottleneck data using `fastPHASE` and BNPPHASE. We then regressed the TMRCA against the number of clusters that each model used per marker. The number of clusters used by the `fastPHASE` model was computed by taking the maximum likelihood (ML) cluster assignments for each genetic sequence using the approximate posterior found by the EM algorithm for `fastPHASE` ([Scheet and Stephens, 2006](#)).

In our third and final experiment, we examined a collection of datasets consisting of 20 intervals chosen randomly from the non-pseudoautosomal region of the male X chromosome. Each dataset consisted of 500 consecutive SNPs (an average length of around  $10^5$  basepairs) from 524 male X chromosomes from the Thousand Genomes Project ([The 1000 Genomes Project Consortium, 2010](#)). Due to limitations in the `fastPHASE` software, only 524 of the 525 male X chromosomes could be used, and so we randomly removed one of the chromosomes for each interval. We held out nested sets of between 10% and 90% of the entries uniformly at random and we examined the accuracy of predicting those entries using imputation based on `fastPHASE` and BNPPHASE. In order to avoid degeneracy, in cases where all minor alleles were held out for a single marker, that marker was discarded from analysis.

### 3.4.1 MCMC initialization, burn-in, iteration, restarts and schedules

The procedure we used for simulating the posterior of the BNPPHASE model with MCMC were the same in all three experiment, except for the runtime experiment, and were as follows. First, we initialized the chain using a scheme in which one sequence of the chain was initialized at a time conditioned on previously initialized sequences. This

initialization method was similar to the product of approximate conditionals method in [Li and Stephens \(2003\)](#). Next, we performed 10 initial iterations in which only the parameters (but not the latent state assignments and jump indicators) were resampled. Subsequently, 50 MCMC iterations were performed consisting of full sweeps over the parameters and Gibbs updates for the latent state assignments and jump indicators of each sequence. In these iterations, the parameters were updated 10 times for each single update of the latent state assignments and jump indicators. The first 20 of these iterations were discarded as burn-in. This procedure was repeated 25 times, each time with an independent initialization (random restarts), yielding 750 iterations which were averaged to produce posterior predictions.

We chose the number of iterations to use by looking at trace plots of the likelihood and accuracy. These traces plateaued at 50 iterations, after which a reasonable mode was found. Other methods in genotype imputation use similar numbers of iterations. Default parameters for `IMPUTE2`, `SHAPEIT` and `BEAGLE` are 40, 35 and 10 iterations respectively (including burn-in). `SHAPEIT` has been run with this small number of iterations to produce reference haplotypes for the Thousand Genomes Project. The small number of iterations required for HMM methods in genetic imputation suggest that for haplotype models the posterior is quite peaked over its' mode.

For the second imputation experiment on male X chromosome data, in addition to conducting the MCMC procedure described above, we also did a grid-search over the latent parameters. For the grid-search, the MCMC procedure was modified by replacing the step wherein the parameter are updated with a step that only updated the parameters that were not involved in the grid-search. The grid-search was done over the parameters  $\alpha_0$ ,  $\alpha$ ,  $b$  and  $\gamma$ .

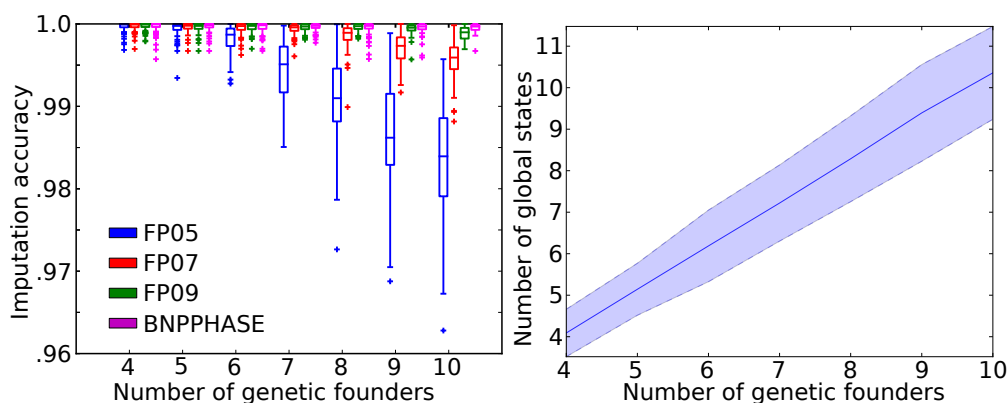
## 3.5 Results

In this section we report the results of the three experiments described above. For the first and last experiments, we show imputation results and we explore the posterior distributions of the `BNPPHASE` model for the X chromosome data. For the second experiment, we show the results of regressing the TMRCA against the number of clusters.

### 3.5.1 Results I: simulated data

#### 3.5.1.1 Imputation of bottleneck with identity-by-descent

In [Figure 3.9](#), we show the results of `BNPPHASE` on the simulated population bottleneck data from the first experiment. [Figure 3.9\(left\)](#) shows the imputation accuracy of `BNPPHASE` compared to `fastPHASE` with the number of components fixed at  $K = 5, 7$  or 9. As the number of components in the `fastPHASE` model increases, the capacity



**Figure 3.9:** Imputation on simulated identity-by-descent data. *Left:* imputation accuracy versus number of genetic founders for simulated population bottleneck data. *Right:* expected number of states under posterior of BNPPHASE model.

of the model increases. Since large capacity is not required to model a small number of genetic founders, all models with enough capacity perform roughly the same for for  $K = 4$  genetic founders. When the number of genetic founders increases beyond the number of components in the `fastPHASE` model, the accuracy decreases by an amount roughly proportional to the difference between the number of genetic founders and the number of components of `fastPHASE`. The baseline accuracy on the genetic imputation task for the simulated data (found by predicting the major allele at all held out locations) was 84.35%. While the ability of BNPPHASE to recover of the true number of components is unsurprising, this experiment gives strong evidence that our inference method is correctly specified.

Figure 3.9(*right*) shows the expected number of latent clusters found as a function of the number of genetic founders in the simulated bottleneck data. We see a direct correlation (1:1) between the number of latent clusters and the number of genetic founders. Note that for real datasets we would not necessarily expect these numbers to coincide directly because in some cases the prior induced by BNPPHASE might prefer to model contiguous haplotype blocks as a single longer haplotype.

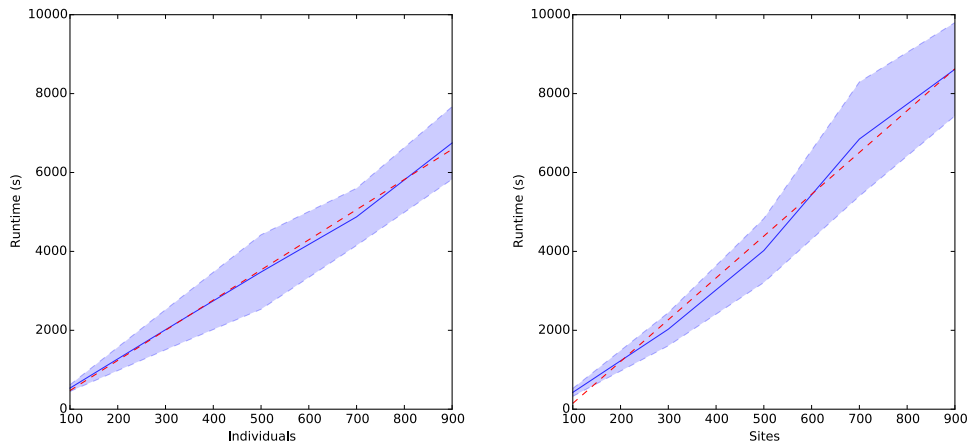
### 3.5.1.2 Examination of runtime

In Figure 3.10 we show the runtime of the BNPPHASE model on simulated data drawn from the BNPPHASE prior. The linear dependence of runtime on both the number of individuals (Figure 3.10 *left*) and the number of sites (Figure 3.10 *right*) is clear from this figure.

## 3.5.2 Results II: TMRCA regression on the out-of-Africa bottleneck

We found a strong negative correlation between the number of clusters used per marker and the TMRCA for both the BNPPHASE model and `fastPHASE`. In Figure 3.11 (*top*,





**Figure 3.10:** Scalability of BNPPHASE. Runtime required for 200 iterations of BNPPHASE as *Left*: number of individuals or *Right*: number of sites is varied. Red dotted line indicates linear fit, solid blue line indicates mean over 10 trials, blue dotted line indicates standard deviation (shaded region is within one standard deviation of mean). Linear trend in both the number of individuals and the number of sites is clear.

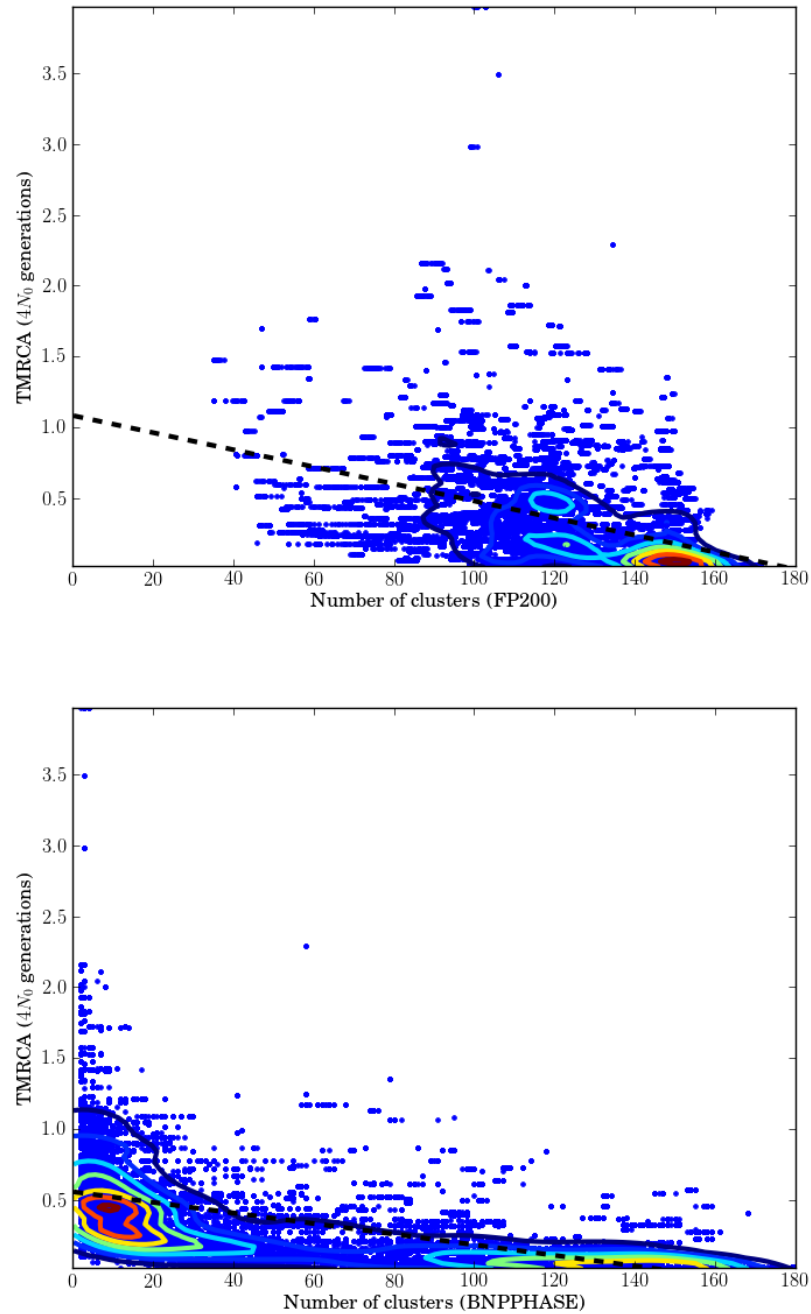
| Method | BNPPHASE      | FP     | FP200  | MAF    |
|--------|---------------|--------|--------|--------|
| RMSE   | <b>0.2724</b> | 0.2855 | 0.3063 | 0.3215 |

Table 3.1: RMSE for regression of TMRCA against # of clusters for BNPPHASE model, **fastPHASE** with default parameters (FP) or  $K = 200$  clusters (FP200), and also against the minor allele frequency (MAF).

*bottom*) we regress the TMRCA against the number of clusters used per marker. When we ran **fastPHASE** with default settings, **fastPHASE** would almost always choose to use 20 clusters in the ML cluster assignment. When we increased the number of clusters to 200 (but otherwise left the parameters of **fastPHASE** with their default settings) large numbers of clusters were still used (as can be seen in Figure 3.11). BNPPHASE often used fewer clusters than **fastPHASE**. Visual inspection of the data suggests that fewer clusters (on the order of the numbers used by BNPPHASE) are often more reasonable representations of the data. As a control, we regressed the TMRCA against the minor allele frequency and in this case we also found a negative correlation. The residual root mean squared errors of the regression were smallest in the BNPPHASE model (Table 3.1).

### 3.5.3 Results III: imputation of male X chromosome data

In Figure 3.12 we show an example region of the male X chromosome used in the imputation experiment on data from the Thousand Genomes Project. Figure 3.12 (*top left*) shows the pattern of minor alleles in this example region. In Figure 3.12 (*top right*), a single sample from an MCMC chain for the BNPPHASE posterior is displayed. By comparing this sample with Figure 3.12 (*top left*), it is clear that the clustering structure



**Figure 3.11:** Regression of TMRCA against number of clusters. Points indicate number of clusters used at marker (x-axis) *vs* TMRCA of marker (y-axis). Contours show level lines of Gaussian kernel density estimation. Dotted line shows regression. *Top:* clusters found by `fastPHASE` with 200 components (FP200). *Bottom:* clusters found by BNPPHASE model.

found by **BNPPHASE** is capturing the haplotype structure of the data. Figures 3.12 (*bottom left, right*) show the posterior distribution of the jump rate and the number of states, respectively. The spikes in the posterior of the jump rate are aligned with change points in the haplotype structure, indicating that recombination hot spots are accurately recovered by the **BNPPHASE** model.

Imputation results for the 20 regions of the male X chromosomes from the Thousand Genomes Project is shown in Figure 3.13. The **BNPPHASE** model consistently outperformed **fastPHASE** run with 10 components (the FP10 condition).<sup>1</sup> For 30%, 50% and 90% hold out conditions, the performance of **BNPPHASE** and FP20 is quite similar. The **BNPPHASE** model tended to do better than other methods in the larger hold out conditions. **BEAGLE** performed well on small hold out conditions, but poorly on large hold out conditions.

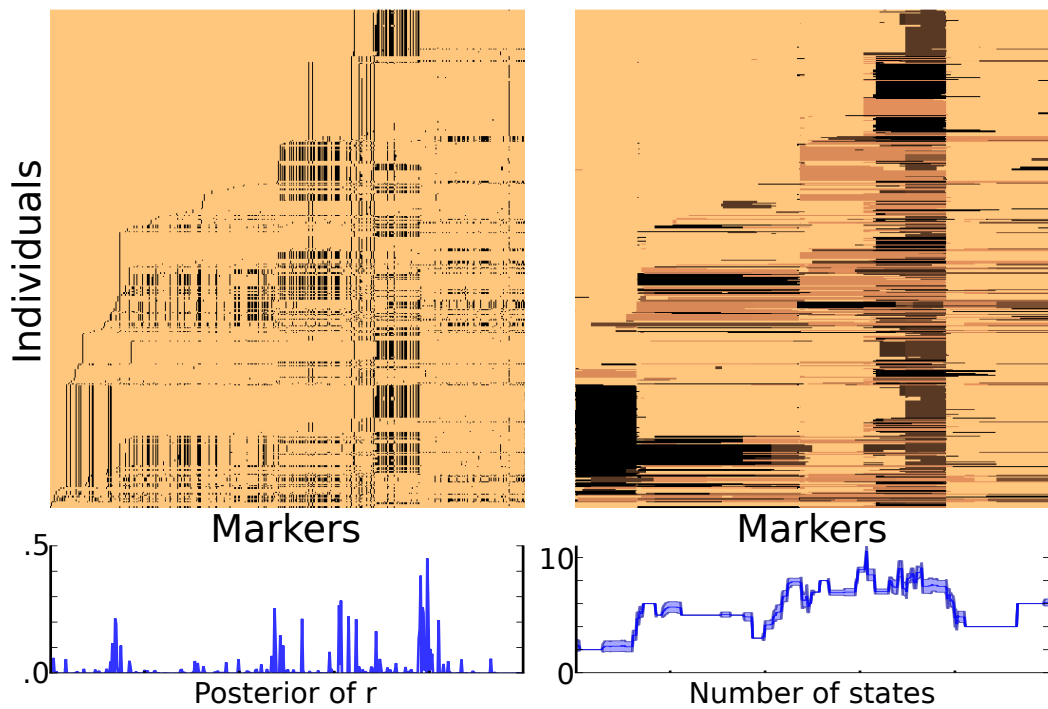
We considered two conditions for sampling the parameters of the **BNPPHASE** model: a fixed condition in which the parameters were fixed to set values, and an unfixed condition in which hyperpriors were placed on the parameters. To find the parameter values in the fixed condition, we perform a grid-search over  $\alpha, \alpha_0, \beta$  and  $\gamma$  and ran MCMC chains without updating these parameters. We chose the parameters that maximize the imputation accuracy for a fixed dataset, and used those parameters for all other datasets. In the second condition (unfixed), MCMC was done for the full model, with slice sampling for  $\alpha, \alpha_0, \beta$  and  $\gamma$ . The average accuracy of the fixed condition for the parameter values that maximized the grid-search was 0.99167 whereas the average accuracy of the unfixed condition was 0.99187. Although small, this difference was found to be significant under a sign test ( $p = 0.04$ ). Since imputation is used as a preprocessing technique in genome wide association studies, even small differences in imputation accuracy could improve the quality of GWAS results.

### 3.6 Discussion

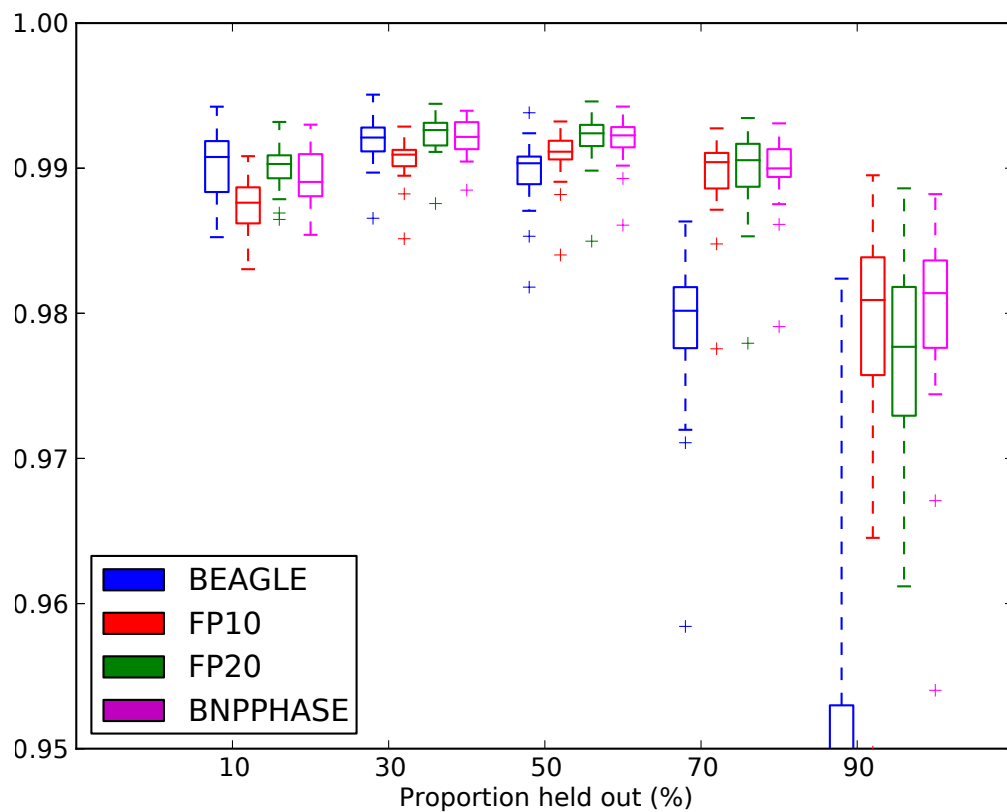
We found that the specifics of the hierarchical likelihood used in the **BNPPHASE** model were quite important. In experiments which are not shown in this paper we looked at two other likelihoods in addition to the one described in equation (3.10). The two additional likelihoods were a uniform Bernoulli likelihoods and a discrete likelihood. For the uniform Bernoulli likelihood we placed a uniform prior on  $\theta_{tk}$  and for the hierarchical deterministic likelihood we replaced the beta prior on  $\theta_{tk}$  with a Bernoulli prior with mean  $\beta_t$  (so that at a given marker, each cluster always emitted either the major or the minor allele for every genetic sequence in that cluster). When experiment II was repeated with each of these two additional priors, the **BNPPHASE** model yielded

---

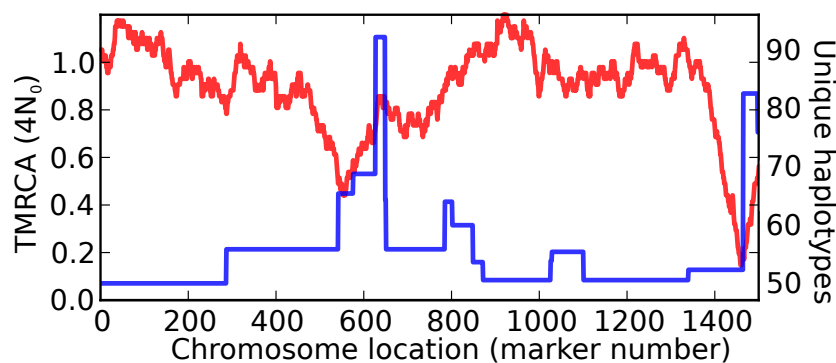
<sup>1</sup>**fastPHASE** was run with the default number of iterations and restarts, along with the ‘-.1m’ command line flag, which prevented **fastPHASE** from throwing out sites in the conditions with more than half the observations missing.



**Figure 3.12:** *Top left:* Example region of male X chromosomes from the Thousand Genomes Project.  $x$ -axis indicates chromosome position,  $y$ -axis indicates individual identity. Black pixels indicate minor alleles. Individuals are presented in sorted order to emphasize haplotype structure (all models we discuss are exchangeable and invariant to order of individuals). *Top right:* Latent cluster assignment of sample from BNPPHASE model posterior. Color indicates cluster identity. *Bottom left, right:* Posterior distributions for jump rate  $r$  and number of states averaged over 20 MCMC samples, shaded region indicates sample standard deviation.



**Figure 3.13:** Imputation accuracy for X chromosomes from the Thousand Genomes Project (*The 1000 Genomes Project Consortium, 2010*). Data from Phase I release v3, acquired on 17/5/2012. Beagle's performance for large held out conditions is low, thus y-axis is truncated to emphasize differences between methods over the whole domain.



**Figure 3.14:** Number of unique haplotypes and TMRCA along chromosome. Red plot indicates number of unique haplotypes appearing in sample in window extending 50 markers to both sides of each marker. Blue plot indicates TMRCA in units of  $4N_0$ .

worse imputation performance and sometimes failed to capture much of the haplotype structure, especially in datasets with low minor allele frequencies.

### 3.6.1 Intuition for TMRCA regression results

We were surprised to see that the correlation between the number of clusters used by `fastPHASE` or `BNPPHASE` and the TMRCA was negative. This could be explained by the nature of population bottlenecks. When mutation rate is low, genetic variation is influenced more strongly by genetic drift. In this case, as TMRCA increases the number of fixed alleles increases, leading to fewer observed haplotypes in the modern population. Bottlenecks involve exponentially expanding populations and so the total number of new mutations in the ancient population is low relative to the modern population. In Figure 3.14, we explored this hypothesis by counting the number of unique patterns of alleles in a simulated bottleneck from experiment I. We found that this empirical count was also negatively correlated with TMRCA (the Pearson correlation coefficient was  $-0.7274$ ).

## 3.7 Conclusion

We presented a new Bayesian nonparametric model for genetic sequence data (`BNPPHASE`). This model is based on a Bayesian nonparametric generalization of the `fastPHASE` model, and captures similar aspects of the genetic process such as non-homogeneous structures. These nonhomogeneous structures often occur in population bottleneck data. The `BNPPHASE` model defines distributions directly on the space of partitions and avoids the label switching problem. We showed that the `BNPPHASE` model provides imputation performance competitive with the state-of-the-art. For simulated population bottleneck data, we showed that it provides better regression against the

TMRCAs than the related `fastPHASE` model and also regression based on minor allele frequencies.

## Glossary of symbols for BNPPHASE model

|                         |   |
|-------------------------|---|
| $a$                     | Block of partition $\mathcal{R}_\ell$   |
| $b$                     | Prior mean of the mass of the allele emission probability for location $\ell$                                   |
| $\beta_\ell$            | Mean allele emission probability for location $\ell$  |
| $\gamma_\ell$           | Mass of allele emission probability for location $\ell$   |
| $\zeta_{i\ell}$         | Block assignment of $i$ -th individual at $\ell$ -th location   |
| $\theta_{\ell k}$       | Allele emission probability for component $k$ at location $\ell$  |
| $K$                     | Number of unique components among $(\varphi_{\ell a})_{\ell a}$ , $a \in \mathcal{R}_\ell$                      |
| $K^{-i}$                | Number of unique components among $(\varphi_{\ell a})_{\ell a}$ , $a \in \mathcal{R}_\ell^{-i}$                 |
| $\alpha_0$              | Concentration parameter for DP  |
| $\alpha$                | Concentration parameter for CRP   |
| $m_\ell^f(z, \zeta)$    | Forwards message  |
| $m_\ell^b(z, \zeta)$    | Backwards message   |
| $N$                     | Number of individuals   |
| $r_\ell$                | Probability of a sequence ‘jumping’ after location $\ell$   |
| $\mathcal{R}_\ell^{-i}$ | Partition of $R_\ell^{-i}$ induced by $\mathcal{R}_\ell$  |
| $\mathcal{R}_\ell$      | Partition of $R_\ell$   |
| $R_\ell^{-i}$           | Set $R_\ell$ with $i$ removed   |
| $R_\ell$                | Set of individuals that ‘jump’ after location $\ell$  |
| $S_\emptyset$           | Set of locations where an individual ‘jumps’ to a singleton cluster   |
| $T$                     | Number of markers   |
| $x_{i\ell}$             | Allele observed for individual $i$ at location $\ell$   |
| $y_{i\ell}$             | Indicates if individual $i$ ‘jumps’ after location $\ell$   |
| $z_{i\ell}$             | Cluster assignment of $i$ -th individual at $\ell$ -th location   |
| $\varphi_{\ell a}$      | Component assignment of a block $a$ of a partition $\mathcal{R}_\ell$   |
| $\omega^{-i}$           | Unique elements among $\{\omega_{z_{i'\ell}} : i' \neq i, 1 \leq \ell \leq L\}$                                 |
| $\omega_\emptyset$      | Mass of DP components other than $\omega_1, \dots, \omega_K$ ( $\omega_\emptyset = 1 - \sum_{k=1}^K \omega_k$ ) |
| $\omega$                | Mass of Dirichlet process components  |
| $\emptyset$             | Symbol representing new block or cluster  |



## Chapter 4

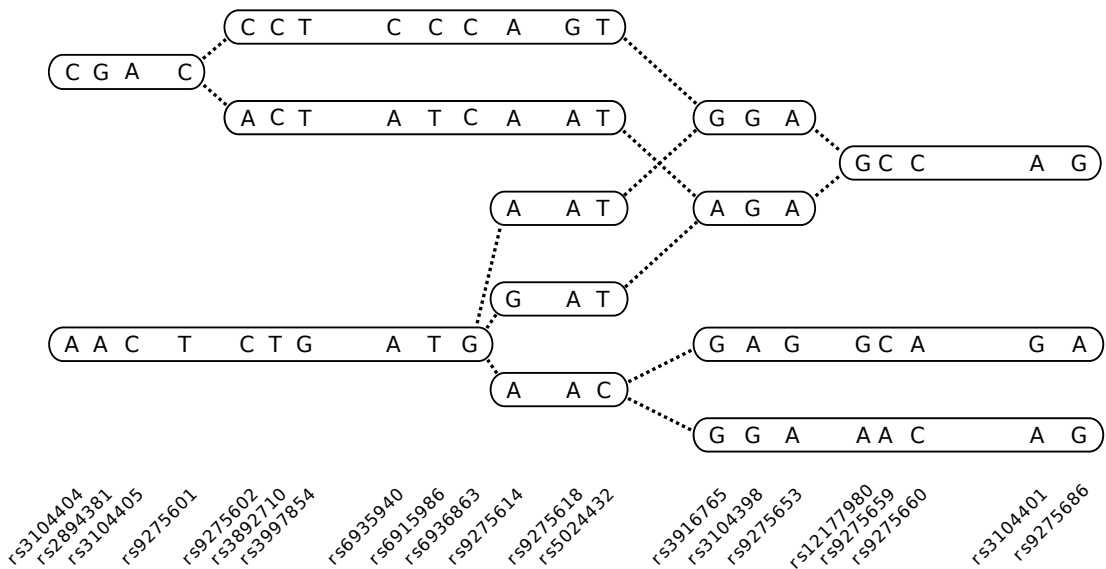
# The discrete fragmentation-coagulation processes

### 4.1 Introduction

We will now present the discrete fragmentation-coagulation process (DFCP) for genetic sequence data (Elliott and Teh, 2012). This model uses the fragmentation and coagulation operators defined in section 2.4 to form a dynamic-partition of the observed genetic sequences. The DFCP model is defined through a discrete Markov chain as follows: starting with the partition  $\mathcal{R}_\ell$  of the set of sequences at the  $\ell$ -th chromosome location, we first fragment each cluster in  $\mathcal{R}_\ell$  into smaller clusters, forming a finer partition  $\mathcal{Q}_\ell$ . Then we coagulate the clusters in  $\mathcal{Q}_\ell$  to form a coarser partition  $\mathcal{R}_{\ell+1}$  of the sequences at the  $\ell+1$ -st chromosome location. This process is repeated at every chromosome location to produce a dynamic-clustering.

Through fragmentation and coagulation events, the DFCP models the block-like, mosaic structure of haplotypes in genetic sequence data (Daly et al., 2001). This structure arises due to recombination and gene conversion occurring in the ancestry of the observed genetic sequences (we refer to section 1.2.1 for more detail). Locally, these prototypical haplotype segments are shared by a cluster of sequences: each sequence in the cluster is described well by a haplotype that is specific to the cluster's location on the chromosome. An example of such a structure found by a fragmentation-coagulation process is shown in Figure 4.1.

As mentioned in section 1.3.2.2, the DFCP is related to the continuous fragmentation-coagulation process (CFCP) which we also derived as a model for genetic sequence data (Teh et al., 2011). In the CFCP, the dynamic-clustering is defined through a latent partition valued Markov jump process in which the blocks of the partition transition



**Figure 4.1:** Mosaic structure found by the fragmentation-coagulation process. Sequences are obtained from phased trios in the CEU population in HapMap, from base pair positions 32790152 to 32795548 on Chromosome 6 (NCBI Build 36 coordinates). Each SNP sequence corresponds to a trajectory, from left to right, through the structure, passing through a number of segments. Each segment consists of a sequence of alleles, while dotted lines correspond to transitions between segments

through fragmentation events (in which one block splits into two) and coagulation events (in which two blocks merge into one). The CFCP is an infinite limit of the DFCP: as the rate of the DFCP and the distance on the chromosome between chromosome locations  $\ell$  and  $\ell+1$  both go to zero, the DFCP converges to the CFCP.

Although inference algorithms for both the DFCP and CFCP scale linearly in the number of sequences and the number of genetic markers, since the CFCP is a Markov jump process, the computational overhead needed to model the arbitrary number of latent events located between two consecutive markers might preclude scalability to large datasets. Further, because the fragmentation and coagulation events in the CFCP are binary (one block splits in two, or two blocks merge to one), the CFCP must use more events than the DFCP in order to model complex latent structures.

We conducted two experiments in which we compared the DFCP and the CFCP to other methods and demonstrated their state-of-the-art imputation accuracy. Our experiments also suggest that the DFCP is more scalable than the CFCP and that MCMC based on the DFCP mixes faster than the uniformization derived for the CFCP. In our first experiment, we compared the imputation accuracy of the CFCP and DFCP and several other methods on the same X chromosome data that we used in Chapter 3 (these data are from [The 1000 Genomes Project Consortium, 2010](#)). In our second experiment, we generated simulated data from the coalescent with recombination. To examine the scalability of the DFCP and CFCP methods, we varied the number of simulated sequences in the population. We found that the DFCP was more scalable.

In the remainder of this section, we will give some intuition about how the DFCP approximates the genetic process and then we will describe the relationship of the DFCP to other popular models in genetics such as IMPUTE and SHAPEIT (Marchini et al., 2007; Delaneau et al., 2012). Then, we will give a mathematical formulation of the DFCP through a generative process. We conclude this section with a formal construction of the CFCP as a limit of the DFCP.

In section 4.2, we derive inference for the DFCP based on forwards-filtering/backwards-sampling and slice sampling. We will also provide inference algorithms for unphased genotype imputation and suggest several approaches for phasing data using the DFCP. We will derive asymptotics related to the expected length of haplotypes in the DFCP model. In sections 4.3 and 4.4 we describe in more detail the experiments we conducted and discuss the results of those experiments. Finally, in section 4.6 we give some concluding remarks about the advantages of the DFCP model.

#### 4.1.1 Relation to the genetic process

The DFCP is an approximation of the sequentially Markov coalescent (McVean and Cardin, 2005) described in section 1.2.1. A complete description of the ancestry of a set of homologous genetic sequences can be approximated by a genealogy-valued Markov process (McVean and Cardin, 2005). The DFCP further approximates these genealogies with a dynamic-clustering in which individuals that are close together in the tree distance implied by the genealogies are in the same cluster.

As noted in section 1.2.1, many models in statistical genetics are based on this Markov dynamic-clustering approximation of the ancestry. By inducing a latent haplotype chart, the DFCP model is quite similar in style to the SHAPEIT/SHAPEIT2 algorithms (Delaneau et al., 2012, 2013). Being both efficient and accurate, the SHAPEIT2 model is currently viewed as the cutting edge of genotype phasing algorithms.

The SHAPEIT2 algorithm is a discrete HMM method in which the forwards/backwards algorithm is used to update the latent state assignments of two unphased diploid sequences (say, sequence  $i$ ). The clusters are formed by examining all of the sequences other than sequence  $i$  and forming a chart similar to the diagram in Figure 4.1. The chart in SHAPEIT2 is formed by dividing the chromosome into segments such that in every segment, there are exactly  $K$  distinct haplotypes appearing among the sequences (here,  $K$  is a user defined parameter). These segments are used as the state assignment of the  $i$ -th sequence (*i.e.*, these segments are the states of the HMM). At the interface between adjacent segments, the HMM transition rule is found by an empirical count. For each pair of segments at the interface, the probability of transiting is proportional to the number of sequences other than  $i$ -th sequence that have made the same transition between the pair of segments at that interface.

The DFCP model also forms charts wherein the segments in the charts are clusters in

the CRP, and through the nature of the ‘rich-gets-richer’ property of the CRP, the probability of a transition between segments is correlated with the number of other sequences that have made the same transition. However, instead of dividing the chromosome and forming segments between all the points in the division, the DFCP allows the boundaries of segments to overlap. Further, rather than being fixed at  $K$ , the number of segments at a given location is learned by the DFCP. These features of the DFCP are made mathematically explicit in the next sections.

#### 4.1.2 Definition of the DFCP through fragmentation and coagulation

Let  $R = \{1, \dots, N\}$  be the indices of  $N$  phased genetic sequences typed at  $L$  biallelic locations (we refer to section 1.1 for more information about this sort of data type). We will now define the DFCP as a dynamic-clustering on  $R$ . The DFCP is parameterized by a concentration  $\alpha > 0$  and rates  $(d_\ell)_{\ell=1}^{L-1}$  with  $d_\ell \in [0, 1]$ . Under the DFCP, the marginal distribution of the partition  $\mathcal{R}_\ell$  is  $\text{CRP}(R, \alpha, 0)$  and so  $\alpha$  controls the number of clusters that are found at each location (with the expected number of clusters in the prior being  $\mathcal{O}(\alpha \log N)$ ). The rate parameter  $d_\ell$  controls the strength of dependence between  $\mathcal{R}_\ell$  and  $\mathcal{R}_{\ell+1}$ , with  $d_\ell = 0$  implying that  $\mathcal{R}_\ell = \mathcal{R}_{\ell+1}$ , and  $d_\ell \rightarrow 1$  implying independence.

Given  $\alpha$  and  $(d_\ell)_{\ell=1}^{L-1}$ , the DFCP is described by the following Markov chain. First we draw a partition  $\mathcal{R}_1 \sim \text{CRP}(R, \alpha, 0)$ . This CRP describes the clustering of  $R$  at location  $\ell = 1$ . Subsequently, we draw  $\mathcal{Q}_\ell | \mathcal{R}_\ell$  from  $\text{FRAG}(\mathcal{R}_\ell, 0, d_\ell)$ , which fragments each of the clusters in  $\mathcal{R}_\ell$  into smaller clusters in  $\mathcal{Q}_\ell$ , and then  $\mathcal{R}_{\ell+1} | \mathcal{Q}_\ell$  from  $\text{COAG}(\mathcal{Q}_\ell, \alpha/d_\ell, 0)$ , which coagulates clusters in  $\mathcal{Q}_\ell$  into larger clusters in  $\mathcal{R}_{\ell+1}$ .

Each  $\mathcal{R}_\ell$  has  $\text{CRP}(R, \alpha, 0)$  as its invariant marginal distribution and each  $\mathcal{Q}_\ell$  is marginally distributed as  $\text{CRP}(R, \alpha, d_\ell)$ . This can be seen by applying Theorem 1 from Chapter 2. (The following substitution of notation must be made to see the result from Theorem 1:  $d_1 \leftarrow 0$ ,  $d_2 \leftarrow d_\ell$ ,  $\alpha \leftarrow \alpha/d_\ell$ .)

Fragmentation and coagulation operators are defined in section 2.4 in terms of CRPs which are projective and exchangeable, and so the latent Markov chain for the DFCP is projective and exchangeable in  $R$  as well. Projectivity and exchangeability are desirable properties for Bayesian nonparametric models because they imply that the marginal distribution of a given data item does not depend on the total number of other data items or on the order in which the other data items are indexed. In genetics, this captures the fact that usually only a small subset of a population is observed.

Theorem 1 also shows that conditioned on  $\mathcal{R}_{\ell+1}$ ,  $\mathcal{Q}_\ell$  has the distribution  $\text{FRAG}(\mathcal{R}_{\ell+1}, 0, d_\ell)$  while  $\mathcal{R}_\ell | \mathcal{Q}_\ell$  has the distribution  $\text{COAG}(\mathcal{Q}_\ell, \alpha/d_\ell, 0)$ . This means that the Markov chain defining the DFCP is reversible (in contrast, `fastPHASE`, `BNPPHASE` and `IMPUTE2` are all non-reversible, as is explored in section 3.3.2). Chromosome replication is directional and so statistics for genetic processes along the chromosome are not

reversible. But the strength of this effect on SNP data is not currently known and many genetic models such as the coalescent with recombination (Hudson, 2002) assume reversibility for simplicity. The non-reversibility displayed by models such as fastPHASE is an artifact of their construction rather than an attempt to capture non-reversible aspects of genetic sequences.

### 4.1.3 Relation to the CFCP

The continuous version of the fragmentation-coagulation process (Teh et al., 2011), which we refer to as the CFCP, is a partition valued Markov jump process (MJP). (The ‘time’ variable for this MJP is the chromosome location, viewed as a continuous variable.) The CFCP is a pure jump process and can be defined in terms of its rates for various jump events. There are two types of events in the CFCP: binary fragmentation events, in which a single cluster  $a$  is split into two clusters  $b$  and  $c$  at a rate of  $d\Gamma(\#c)\Gamma(\#b)/\Gamma(\#a)$ , and binary coagulation events in which two clusters  $b$  and  $c$  merge to form one cluster  $a$  at a rate of  $d/\alpha$ . (The coagulation probability is independent of the sizes of  $a, b$  and  $c$ .)

As was shown in (Teh et al., 2011) the CFCP can be realized as a continuous limit of the DFCP. Consider a DFCP with concentration  $\alpha$  and constant rate parameter  $d_\varepsilon$ . Then as  $\varepsilon \rightarrow 0$  the probability that the coagulation and fragmentation operations at a specific time step  $\ell$  induce no change in the partition structure  $\mathcal{R}_\ell$  approaches 1. Conversely, the probability that these operations are the binary events given above scales as  $\mathcal{O}(\varepsilon)$ , while all other events scale as larger powers of  $\varepsilon$ . If we rescale the time steps by  $\ell \mapsto \varepsilon\ell$ , then  $d \mapsto \varepsilon d$  and the expected number of binary events over a finite interval approaches  $\varepsilon$  times the rates given above and the expected number of all other events goes to zero, yielding the CFCP. This is shown by taking the following limits. For fragmentation, we have from equation (2.9):

$$\Pr(\text{FRAG}(\mathcal{R}, 0, \varepsilon d) = \mathcal{Q} | \mathcal{R}) \tag{4.1}$$

$$= \frac{(\varepsilon d)^{\#\mathcal{Q} - \#\mathcal{R}}}{\Gamma(1 - \varepsilon d)^{\#\mathcal{Q}}} \prod_{a \in \mathcal{R}} \frac{\Gamma(\#F_a)}{\Gamma(\#a)} \prod_{b \in \mathcal{Q}} \Gamma(\#b - \varepsilon d) \tag{4.2}$$

$$\stackrel{\varepsilon \rightarrow 0}{\cong} \begin{cases} 1 + \mathcal{O}(\varepsilon^2) & \text{if } \mathcal{Q} = \mathcal{R}, \\ \varepsilon d \frac{\Gamma(a)\Gamma(b)}{\Gamma(c)} + \mathcal{O}(\varepsilon^2) & \text{if } \#\mathcal{Q} - \#\mathcal{R} = 1 \text{ and } \mathcal{Q} = \mathcal{R} - c \cup \{a, b\}, \\ \mathcal{O}(\varepsilon^2) & \text{if } \#\mathcal{Q} - \#\mathcal{R} > 1. \end{cases} \tag{4.3}$$

In this limit, to arrive at equation (4.3) we have used that  $\Gamma(X + \varepsilon d) = \Gamma(X) + \mathcal{O}(\varepsilon)$

for  $X > 0$  as  $\varepsilon \rightarrow 0$ . For coagulation, we have from equation (2.10):

$$\Pr(\text{COAG}(\mathcal{Q}, \alpha/(\varepsilon d), 0) = \mathcal{R} | \mathcal{Q}) \quad (4.4)$$

$$= \frac{(\alpha/(\varepsilon d))^{\#\mathcal{R}} \Gamma(\alpha/(\varepsilon d))}{\Gamma(\alpha/(\varepsilon d) + \#\mathcal{Q})} \prod_{a \in \mathcal{R}} \Gamma(\#C_a) \quad (4.5)$$

$$= \frac{(\varepsilon d/\alpha)^{\#\mathcal{Q} - \#\mathcal{R}}}{1 \cdot (1 + \varepsilon d/\alpha) \cdot \dots \cdot (1 + \varepsilon d/\alpha(\#\mathcal{Q} - 1))} \prod_{a \in \mathcal{R}} \Gamma(\#C_a) \quad (4.6)$$

$$\stackrel{\varepsilon \rightarrow 0}{\approx} \begin{cases} 1 + \mathcal{O}(\varepsilon^2) & \text{if } \mathcal{Q} = \mathcal{R}, \\ \varepsilon d/\alpha + \mathcal{O}(\varepsilon^2) & \text{if } \#\mathcal{Q} - \#\mathcal{R} = 1 \text{ and } \mathcal{Q} = \mathcal{R} - c \cup \{a, b\}, \\ \mathcal{O}(\varepsilon^2) & \text{if } \#\mathcal{Q} - \#\mathcal{R} > 1. \end{cases} \quad (4.7)$$

To derive equation (4.6) we have used that  $\Gamma(X)/\Gamma(X + J) = 1/X \cdot 1/(X + 1) \cdot \dots \cdot 1/(X + J - 1)$  for  $X > 0$  and  $J \in \mathbb{N}$  and we have multiplied the top and bottom of the fraction by  $(\varepsilon d/\alpha)^{\#\mathcal{Q}}$ .

In the CFCP fragmentation and coagulation events are binary: they involve either one cluster fragmenting into two new clusters, or two clusters coagulating into one new cluster. However, for the DFCP the fragmentation and coagulation operators can describe more complicated haplotype structures without introducing more latent events. For example one cluster splitting into three clusters (as happens to the second haplotype from the top of Figure 4.1 after the 10th SNP) can be described by the DFCP using just one fragmentation operator. The order of the latent events introduced by the CFCP required does not matter, adding unnecessary local modes to its posterior.

## 4.2 Methods

We will now derive a Gibbs sampler for posterior simulation in the DFCP by making use of the exchangeability of the process. Each iteration of the sampler updates the trajectory of cluster assignments of one sequence  $i$  through the partition structure. To arrive at the updates, we will consider the conditional distribution of the  $i$ -th trajectory given all of the others, which can be shown to be a Markov chain. Coupled with the deterministic likelihood terms, we then use a backwards-filtering/forwards-sampling algorithm to obtain a new trajectory for sequence  $i$ . In this section, we derive the conditional distribution of trajectory  $i$  using the definition of fragmentation and coagulation and also the posterior distributions of the parameters  $d_\ell, \alpha$  which we will update using slice sampling (Neal, 2003).

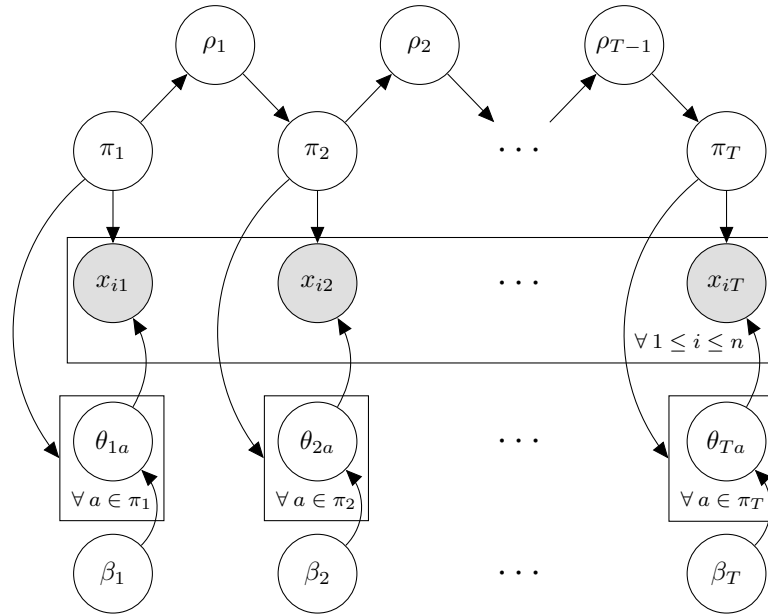
### 4.2.1 Likelihood model and parameter priors

We used a discrete likelihood in which the same observation is emitted for each sequence in a cluster. The likelihood model was specified as follows. Given the sequence

of partitions  $(\mathcal{R}_\ell)_{\ell=1}^L$ , we model the observations in each cluster at each location  $\ell$  independently. For each cluster  $a \in \mathcal{R}_\ell$  at location  $\ell$  and for each sequence  $i$ , let  $a_{i\ell} \in \mathcal{R}_\ell$  be the cluster in  $\mathcal{R}_\ell$  containing  $i$ . Let  $\theta_{\ell a}$  be the emission of cluster  $a$  at location  $\ell$ . Since SNP data has binary labels,  $\theta_{\ell a} \in \{0, 1\}$  is a Bernoulli random variable. Let the mean of  $\theta_{\ell a}$  be  $\beta_\ell$  (this is the latent allele frequency at location  $\ell$ ). We assume that conditioned on the partitions and the parameters, the observations  $x_{i\ell}$  are independent, and determined by the cluster parameter  $\theta_{\ell a}$ . Thus the probability  $\Pr(\theta_{\ell a} = 1 | \beta_\ell) = \beta_\ell$  and the probability  $\Pr(x_{i\ell} | a_{i\ell} = a, \theta_{\ell a}) = \delta(x_{i\ell} = \theta_{\ell a})$  where  $\delta$  is an indicator function (i.e., it is one if  $x_{i\ell} = \theta_{\ell a}$  and zero otherwise).

We place a beta prior on  $\beta_\ell$  with mean parameter  $1/2$  and mass parameter  $\gamma_\ell$ . The mass parameters are themselves marginally independent and we place on them an uninformative log-uniform prior over a range:  $p(\gamma_\ell) \propto \gamma_\ell^{-1}$ ,  $\gamma_\ell \geq \gamma_{\min}$ . Since this distribution is heavy tailed, the  $\beta_\ell$  variables will have more mass near 0 and 1 than they would have if  $\gamma_\ell$  were fixed, adding sparsity to the latent allele frequencies. This phenomenon is empirically observed in SNP data. The parameters  $\beta_\ell$  will be integrated out during inference.

We also place an uninformative log-uniform prior on the rates  $d_\ell$  over a range:  $p(d_\ell) \propto d_\ell^{-1}$ ,  $d_\ell \geq d_{\min}$ . Note that the prior gives more mass to values of  $d_\ell$  close to  $d_{\min}$  which we set close to zero; we expect the partitions of consecutive locations to be relatively similar so that the mosaic haplotype structure can be formed. Finally, we place a log-normal prior on  $\alpha$  with mean  $m$  and variance  $v$ :  $\log \alpha \sim \mathcal{N}(m, v)$ ,  $\alpha > 0$ . The graphical model for this generative process is shown in Figure 4.2(*Top*), and it is summarized in equation (4.8).



$$\begin{aligned}
 \mathcal{R}_1 &\sim \text{CRP}(R, \alpha, 0), \\
 \mathcal{Q}_\ell | \mathcal{R}_\ell &\sim \text{FRAG}(\mathcal{R}_\ell, 0, d_\ell), \\
 \mathcal{R}_{\ell+1} | \mathcal{Q}_\ell &\sim \text{COAG}(\mathcal{Q}_\ell, \alpha/d_\ell, 0), \\
 \log \alpha &\sim \mathcal{N}(m, v), \\
 \log d_\ell &\sim \text{Uniform}(\log R_{\min}, 0), \\
 x_{i\ell} | a_{i\ell} = \theta_{ta_{i\ell}}, \theta_{la} | \beta_\ell &\sim \text{Bernoulli}(\beta_\ell), \\
 \beta_\ell | \gamma_\ell &\sim \text{Beta}\left(\frac{\gamma_\ell}{2}, \frac{\gamma_\ell}{2}\right), \\
 \log \gamma_\ell &\sim \text{Uniform}(\log \gamma_{\min}, 0).
 \end{aligned} \tag{4.8}$$

**Figure 4.2:** *Top:* Plate diagram for the discrete fragmentation-coagulation process. For brevity hyperparameters are not shown.  $T$  denotes number of markers. *Bottom:* Generative process for genetic sequences  $(x_{i\ell})_{i=1}^N$ .



### 4.2.2 Joint probability distribution for DFCP

The full probability for the dynamic-clustering prior on the partitions  $\mathcal{R}_1, \dots, \mathcal{R}_L, \mathcal{Q}_1, \dots, \mathcal{Q}_{L-1}$  induced by the DFCP is given by the following equation:

$$\begin{aligned}
\Pr(\mathcal{R}, \mathcal{Q} | \alpha, d) &= \left( \frac{\alpha^{\#\mathcal{R}_1} \Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{a \in \mathcal{R}_1} \Gamma(\#a) \right) \\
&\cdot \left( \prod_{\ell=1}^{L-2} \frac{d_\ell^{\#\mathcal{Q}_\ell - \#\mathcal{R}_\ell}}{\Gamma(1 - d_\ell)^{\#\mathcal{Q}_\ell}} \prod_{a \in \mathcal{R}_\ell} \frac{\Gamma(\#F_a)}{\Gamma(\#a)} \prod_{b \in \mathcal{Q}_\ell} \Gamma(\#b - d_\ell) \right) \\
&\cdot \left( \prod_{\ell=1}^{L-1} \frac{(\alpha/d_\ell)^{\#\mathcal{R}_{\ell+1}} \Gamma(\alpha/d_\ell)}{\Gamma(\alpha/d_\ell + \#\mathcal{Q}_\ell)} \prod_{a \in \mathcal{R}_{\ell+1}} \Gamma(\#C_a) \right), \\
&= \frac{\alpha^{\sum_{\ell=1}^L \#\mathcal{R}_\ell} \Gamma(\alpha)}{\Gamma(\alpha + N)} \left( \prod_{\ell=2}^{L-1} \prod_{a \in \mathcal{R}_\ell} \frac{\Gamma(\#C_a) \Gamma(\#F_a)}{\Gamma(\#a)} \right) \left( \prod_{a \in \mathcal{R}_1} \Gamma(\#F_a) \right) \\
&\cdot \left( \prod_{a \in \mathcal{R}_L} \Gamma(\#C_a) \right) \prod_{\ell=1}^{L-1} \frac{\Gamma(\alpha/d_\ell) d_\ell^{\#\mathcal{Q}_\ell - \#\mathcal{R}_{\ell+1} - \#\mathcal{R}_\ell}}{\Gamma(\alpha/d_\ell + \#\mathcal{R}_\ell) \Gamma(1 - d_\ell)^{\#\mathcal{Q}_\ell}} \prod_{b \in \mathcal{Q}_\ell} \Gamma(\#b - d_\ell).
\end{aligned} \tag{4.9}$$

In the first equality listed in (4.9), the bracketed expressions correspond to the probabilities arising from the initial CRP, the  $L - 1$  fragmentation operations and the  $L$  coagulation operations respectively. The exchangeability and reversibility of the process follows from this equation. Also, from this equation, we can derive the posterior probabilities for  $\alpha$  and  $d_\ell$  conditioned on  $\mathcal{R}$  and  $\mathcal{Q}$  using Bayes' rule.

$$\begin{aligned}
\Pr(\alpha | \mathcal{R}, \mathcal{Q}, d) &\propto \frac{\alpha^{\sum_{\ell=1}^L \#\mathcal{R}_\ell} \Gamma(\alpha)}{\Gamma(\alpha + N)} \prod_{\ell=1}^{L-1} \frac{\Gamma(\alpha/d_\ell)}{\Gamma(\alpha/d_\ell + \#\mathcal{Q}_\ell)}, \\
\Pr(d_\ell | \mathcal{R}, \mathcal{Q}, \alpha) &\propto \frac{\Gamma(\alpha/d_\ell) d_\ell^{\#\mathcal{Q}_\ell - \#\mathcal{R}_{\ell+1} - \#\mathcal{R}_\ell}}{\Gamma(\alpha/d_\ell + \#\mathcal{Q}_\ell) \Gamma(1 - d_\ell)^{\#\mathcal{Q}_\ell}} \prod_{b \in \mathcal{Q}_\ell} \Gamma(\#b - d_\ell).
\end{aligned} \tag{4.10}$$

### 4.2.3 Gibbs update for latent block assignment of sequence $i$

We will use the same notation that we used in Chapter 3 to define projections of partitions and events involving the cluster assignment of an individual sequence. In particular, we will fix sequence  $i$  and for a partition  $\mathcal{R}$  of  $[N]$  we will denote by  $\mathcal{R}^{-i}$  the partition of  $[N] - \{i\}$  (i.e., the set  $1, 2, \dots, i - 1, i + 1, \dots, N$ ) induced by  $\mathcal{R}$ . That is, we remove sequence  $i$  from the block of  $\mathcal{R}$  in which it resides, and if removing sequence  $i$  from that block yields the empty set then we also remove the empty set and thereby form a partition of  $[N] - \{i\}$ . By exchangeability, we imagine that the partition structure were built sequentially and that sequence  $i$  was the last sequence to be added to it. Thus, we need notation to describe which blocks sequence  $i$  joins as it

is added to the partition structure. We will denote by  $a_\ell$  the assignment of sequence  $i$  at location  $\ell$ . If sequence  $i$  joins a cluster  $a$  that already exists in  $\mathcal{R}^{-i}$  we will denote this event by  $a_\ell = a$ . But if a new cluster is created for sequence  $i$  (i.e., if sequence  $i$  will be in the block  $\{i\}$  in  $\mathcal{R}$ ) then we denote this event by  $a_\ell = \emptyset$ . Similarly, for  $\mathcal{Q}$  we will denote the assignment of sequence  $i$  in  $\mathcal{Q}_\ell$  by  $b_\ell$  and write  $b_\ell = b$  or  $b_\ell = \emptyset$  for the cases where  $b_\ell$  joins a cluster  $b \in \mathcal{Q}_\ell^{-i}$  or a new cluster being created for sequence  $i$  in  $\mathcal{Q}_\ell$ . We now use the conditional distributions derived in section 2.4.2 to arrive at the following conditional equations:

$$\begin{aligned} \Pr(a_1 = a | \mathcal{R}_1^{-i}) &= \begin{cases} \#a / (N - 1 + \alpha) & \text{if } a \in \mathcal{R}_1^{-i}, \\ \alpha / (N - 1 + \alpha) & \text{if } a = \emptyset. \end{cases} \\ \Pr(b_\ell = b | a_\ell = a, \mathcal{R}_\ell^{-i}, \mathcal{Q}_\ell^{-i}) &= \begin{cases} (\#b - d_\ell) / \#a & \text{if } a \in \mathcal{R}_\ell^{-i}, b \in F_\ell(a), \\ d_\ell \#F_\ell(a) / \#a & \text{if } a \in \mathcal{R}_\ell^{-i}, b = \emptyset, \\ 1 & \text{if } a = b = \emptyset, \\ 0 & \text{otherwise.} \end{cases} \\ \Pr(a_{\ell+1} = a | b_\ell = b, \mathcal{R}_{\ell+1}^{-i}, \mathcal{Q}_\ell^{-i}) &= \begin{cases} d_\ell \#C_\ell(a) / (\alpha + d_\ell \# \mathcal{Q}_\ell^{-i}) & \text{if } a \in \mathcal{R}_{\ell+1}^{-i}, b = \emptyset, \\ \alpha / (\alpha + d_\ell \# \mathcal{Q}_\ell^{-i}) & \text{if } a = b = \emptyset, \\ 1 & \text{if } a \in \mathcal{R}_{\ell+1}^{-i}, b \in C_\ell(a), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (4.11)$$

Here,  $F_\ell(a)$  is the set of blocks  $b \in \mathcal{Q}_\ell$  into which the block  $a \in \mathcal{R}_\ell$  fragments, and  $C_\ell(a)$  is the set of blocks in  $\mathcal{Q}_\ell$  that coagulate to form the block  $a \in \mathcal{R}_{\ell+1}$ . As in Elliott and Teh (2012) we define the following messages for  $\ell = 1, \dots, L - 1$ :

$$\begin{aligned} m_C^\ell(a) &= \Pr(x_{i,(\ell+1):L} | a_\ell = a, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:(L-1)}^{-i}), \\ m_F^\ell(b) &= \Pr(x_{i,(\ell+1):L} | b_\ell = b, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:(L-1)}^{-i}). \end{aligned}$$

These messages can be computed recursively as follows:

$$m_{\mathcal{F}}^\ell(b) = \sum_{a \in \pi_{\ell+1}^{-i} \cup \{\emptyset\}} m_C^{\ell+1}(a) \underbrace{\Lambda(x_{i,(\ell+1)} | a)}_{\text{Likelihood}} \underbrace{\Pr(a_{\ell+1} = a | b_\ell = b, \pi_{\ell+1}^{-i}, \rho_\ell^{-i})}_{\text{Coagulation probabilities from (2.13)}}. \quad (4.12)$$

$$m_C^\ell(a) = \sum_{b \in \rho_\ell^{-i} \cup \{\emptyset\}} m_{\mathcal{F}}^\ell(b) \underbrace{\Pr(b_\ell = b | a_\ell = a, \pi_\ell^{-i}, \rho_\ell^{-i})}_{\text{Fragmentation probabilities from (2.12)}}. \quad (4.13)$$

Here  $\Lambda(x|a)$  is the likelihood term induced by the discrete likelihood defined in Section 4.2.2. In particular, in the event that  $a \neq \emptyset$ , the  $i$ -th sequence joins a nonempty cluster  $a \in \mathcal{R}_\ell$ . Since this cluster will emit the same allele for every sequence in it, the allele emitted by sequence  $i$  at  $\ell$  is determined and  $\Lambda(x|a) = \delta(x_{i\ell} = \theta_{a\ell})$ , where  $\delta$  is the Dirac delta function. On the other hand, if  $a = \emptyset$ , then  $\Lambda(x|a)$  is found by

marginalizing the  $\beta_\ell$  parameter in the beta/Bernoulli hierarchy for  $\theta$  at  $\ell$ , and:

$$\Lambda(x_{i\ell} = 1|\emptyset) = \frac{\gamma_\ell/2 + n_{1\ell}}{\gamma_\ell + n_{1\ell} + n_{0\ell}}, \quad \Lambda(x_{i\ell} = 0|\emptyset) = \frac{\gamma_\ell/2 + n_{0\ell}}{\gamma_\ell + n_{1\ell} + n_{0\ell}}. \quad (4.14)$$

Here  $n_{1\ell} = \#\{a \in \mathcal{R}_\ell^{-i} | \theta_{a\ell}\}$  is the number of clusters in  $\mathcal{R}_\ell^{-i}$  that emit the major allele and  $n_{0\ell}$  is the number of clusters in  $\mathcal{R}_\ell^{-i}$  that emit the minor allele.

As the fragmentation and coagulation conditional probabilities are only supported for clusters  $a, b$  such that  $b \subseteq a$ , these sums can be expanded so that only non-zero terms are summed over. Noting this restriction on the support and substituting (4.11) into (4.12) and (4.13) yields the following. For  $\ell = \ell, \dots, L-1$ :

$$\begin{aligned} m_F^\ell(b) &= \Pr(x_{i,(\ell+1):L} | b_\ell = b, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:(L-1)}^{-i}), \\ &= \sum_{a \in \mathcal{R}_{\ell+1}^{-i} \cup \{\emptyset\}} \Pr(x_{i,(\ell+2):L} | a_{\ell+1} = a, \mathcal{R}_{(\ell+1):L}^{-i}, \mathcal{Q}_{(\ell+1):(L-1)}^{-i}) \\ &\quad \cdot \Pr(x_{i,(\ell+1)} | a_{\ell+1} = a) \Pr(a_{\ell+1} = a | b_\ell = b, \mathcal{R}_{\ell+1}^{-i}, \mathcal{Q}_\ell^{-i}), \\ &= \sum_{a \in \mathcal{R}_{\ell+1}^{-i} \cup \{\emptyset\}} m_C^{\ell+1}(a) \Lambda(x_{i,(\ell+1)} | a_\ell = a) \Pr(a_{\ell+1} = a | b_\ell = b, \mathcal{R}_{\ell+1}^{-i}, \mathcal{Q}_\ell^{-i}), \\ &= \begin{cases} \frac{1}{\alpha + d_\ell \# \mathcal{Q}_\ell^{-i}} (m_C^{\ell+1}(\emptyset) \Lambda(x_{i,(\ell+1)} | \emptyset) \alpha + \sum_{a \in \mathcal{R}_{\ell+1}^{-i}} m_C^{\ell+1}(a) \Lambda(x_{i,(\ell+1)} | a) d_\ell \# C_\ell(a)) & \text{if } b = \emptyset, \\ m_C^{\ell+1}(a) \Lambda(x_{i,(\ell+1)} | a), \text{ where } a \in \mathcal{R}_{\ell+1}^{-i} \text{ unique, s.t. } b \in C_\ell(a) & \text{if } b \in \mathcal{Q}_\ell^{-i}. \end{cases} \end{aligned} \quad (4.15)$$

$$\begin{aligned} m_C^\ell(a) &= \Pr(x_{i,(\ell+1):L} | a_\ell = a, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:(L-1)}^{-i}), \\ &= \sum_{b \in \mathcal{Q}_\ell^{-i} \cup \{\emptyset\}} \Pr(x_{i,(\ell+1):L} | b_\ell = b, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:(L-1)}^{-i}) \Pr(b_\ell = b | a_\ell = a, \mathcal{R}_\ell^{-i}, \mathcal{Q}_\ell^{-i}), \\ &= \sum_{b \in \mathcal{Q}_\ell^{-i} \cup \{\emptyset\}} m_F^\ell(b) \Pr(b_\ell = b | a_\ell = a, \mathcal{R}_\ell^{-i}, \mathcal{Q}_\ell^{-i}), \\ &= \begin{cases} \frac{1}{\#a} \left( m_F^\ell(\emptyset) d_\ell \# F_\ell(a) + \sum_{b \in F_\ell(a)} m_F^\ell(b) (\#b - d_\ell) \right) & \text{if } a \in \mathcal{R}_\ell^{-i}, \\ m_F^\ell(\emptyset) & \text{if } a = \emptyset. \end{cases} \end{aligned} \quad (4.16)$$

To sample from the posterior distribution of the trajectory for sequence  $i$  conditioned on the other trajectories and the data, we use the Markov property for the chain

$a_1, b_1, \dots, b_{L-1}, a_L$  and the definition of the messages. Starting at location 1, we have:

$$\begin{aligned}
& \Pr(a_1 = a | x_i, \pi_{1:L}^{-i}, \rho_{1:(L-1)}^{-i}) \\
& \propto \Pr(a_1 = a | \pi_1^{-i}) \Pr(x_{i1} | a_1 = a) \Pr(x_{i,2:L} | a_1 = a, \pi_{1:L}^{-i}, \rho_{1:(L-1)}^{-i}), \\
& = \underbrace{\Pr(a_1 = a | \pi_1^{-i})}_{\text{CRP probabilities (2.11)}} \underbrace{\Lambda(x_1 | a_1)}_{\text{Likelihood from (4.14)}} m_C^1(a). \tag{4.17}
\end{aligned}$$

For subsequent  $b_\ell$  and  $a_{\ell+1}$  for locations  $\ell = 1, \dots, L-1$ ,

$$\begin{aligned}
& \Pr(b_\ell = b | a_\ell = a, x_i, \pi_{1:L}^{-i}, \rho_{1:(L-1)}^{-i}) \\
& \propto \Pr(b_\ell = b | a_\ell = a, \pi_\ell^{-i}, \rho_\ell^{-i}) \Pr(x_{i,(\ell+1):L} | b_\ell = b, \pi_{\ell:L}^{-i}, \rho_{\ell:(L-1)}^{-i}), \\
& = \underbrace{\Pr(b_\ell = b | a_\ell = a, \pi_\ell^{-i}, \rho_\ell^{-i})}_{\text{Fragmentation probabilities from (2.12)}} m_{\mathcal{F}}^\ell(b). \tag{4.18}
\end{aligned}$$

$$\begin{aligned}
& \Pr(a_\ell = a | b_{\ell-1} = b, x_i, \pi_{1:L}^{-i}, \rho_{1:(L-1)}^{-i}) \\
& \propto \Pr(a_\ell = a | b_{\ell-1} = b, \pi_\ell^{-i}, \rho_{\ell-1}^{-i}) \Pr(x_{i\ell} | a_\ell = a) \Pr(x_{i,(\ell+1):L} | a_\ell = a, \pi_{\ell:L}^{-i}, \rho_{\ell:(L-1)}^{-i}), \\
& = \underbrace{\Pr(a_\ell = a | b_{\ell-1} = b, \pi_\ell^{-i}, \rho_{\ell-1}^{-i})}_{\text{Coagulation probability from (2.13)}} \underbrace{\Lambda(x_{i\ell} | a)}_{\text{Likelihood from (4.14)}} m_C^\ell(a). \tag{4.19}
\end{aligned}$$

The complexity of this update is  $\mathcal{O}(KT)$  where  $K$  is the expected number of clusters in the posterior. This complexity class is the same as for the CFCP and other related HMM methods such as fastPHASE. But there is no exact Gibbs update for the trajectories in the CFCP. Instead the CFCP sampler relies on uniformization (Rao and Teh, 2011).

#### 4.2.4 Slice sampling for parameters $\alpha$ , $d_\ell$ , and $\gamma_\ell$

We use slice sampling (Neal, 2003) to update the  $\alpha$  and  $d_\ell$  parameters conditioned on the partition structure and also the likelihood parameters. To this end, we must derive unnormalized versions of the parameters. We use Bayes' rule, equation (4.9) and the identity  $[a]_b^N = b^N \Gamma(a/b + N) / \Gamma(a/b)$ , and then the posterior probabilities of  $\alpha$  and  $d_\ell$  given the partitions  $\mathcal{R}_{1:L}$  and  $\mathcal{Q}_{1:(L-1)}$  are as follows:

$$\begin{aligned}
\Pr(\alpha | \mathcal{R}, \mathcal{Q}, d) & \propto \Pr(\alpha) \Pr(\mathcal{R}_1 | \alpha, d_1) \Pr(\mathcal{Q}_1 | \mathcal{R}_1, \alpha, d_1) \cdots \Pr(\mathcal{R}_L | \mathcal{Q}_{L-1}, \alpha, d_{L-1}), \\
& \propto \Pr(\alpha) \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \alpha^{-L + \sum_{\ell=1}^L \#\mathcal{R}_\ell} \prod_{\ell=1}^{L-1} \frac{\Gamma(\alpha/d_\ell)}{\Gamma(\alpha/d_\ell + \#\mathcal{Q}_\ell)}. \tag{4.20}
\end{aligned}$$

$$\begin{aligned}
\Pr(d_\ell | \mathcal{R}, \mathcal{Q}, \alpha) & \propto \Pr(d_\ell) \Pr(\mathcal{Q}_\ell | \mathcal{R}_\ell, \alpha, d_\ell) \Pr(\mathcal{R}_{\ell+1} | \mathcal{Q}_\ell, \alpha, d_\ell), \\
& \propto \Pr(d_\ell) d_\ell^{\#\mathcal{Q}_\ell - \#\mathcal{R}_\ell - \#\mathcal{R}_{\ell+1} + 1} \frac{\Gamma(\alpha/d_\ell) \Gamma(1 - d_\ell)^{-\#\mathcal{Q}_\ell}}{\Gamma(\#\mathcal{Q}_\ell + \alpha/d_\ell)} \prod_{b \in \mathcal{Q}_\ell} \Gamma(\#b - d_\ell). \tag{4.21}
\end{aligned}$$

The conditional distribution for  $\gamma_\ell$  is given by the definition of the likelihood in section 4.2.2. We can derive this update for  $\gamma_\ell$  with  $\beta_\ell$  marginalized, using the following equation:

$$\Pr(\gamma_\ell|x, \mathcal{R}_\ell) = \Pr(\gamma_\ell) \frac{\Gamma(\gamma_\ell)\Gamma(\gamma/2 + n_{1\ell})\Gamma(\gamma/2 + n_{0\ell})}{\Gamma(\gamma/2)^2\Gamma(\gamma + n_{1\ell} + n_{0\ell})} \quad (4.22)$$

Here  $\Pr(\gamma_\ell)$  is the log uniform prior from equation (4.8) and  $n_{1\ell}, n_{0\ell}$  are the number of clusters at location  $\ell$  that emit the major or minor allele respectively, as in equation (4.14).

This concludes the specification of MCMC simulation of the DFCP posterior for phased genetic sequence data.

#### 4.2.5 Genotype imputation for unphased data

We will now derive an MCMC algorithm for genetic imputation for unphased data using the DFCP model. As in the case considered in the previous section, we use message passing to specify a Gibbs update for the latent clustering of the diploid pair of chromosomes for each individual conditioned on the dynamic clusterings of all the other individuals. The allele emission at a missing location can then be predicted by collecting MCMC samples and then marginalizing the cluster assignments of the pair of cluster assignments for the diploid sequence of an individual.

A genotype is a sequence of unordered alleles and so the Gibbs steps we will derive update the latent cluster assignments of both of the sequences representing the pair of haplotypes comprising a given chromosome for a given diploid individual. Consequently, we also produce an update for the relative ordering (*i.e.*, the phase) of alleles at each pair of consecutive locations for which that individual is heterozygous.

As in Chapter 2, we denote the cluster assignments of the coagulated states of a sequence by  $a_\ell \in \mathcal{R}_\ell$  and of the fragmented states by  $b_\ell \in \mathcal{Q}_\ell$ . So,  $\mathcal{R}_\ell$  is the partition induced by the clustering of the sequences at location  $\ell$  and  $\mathcal{Q}_\ell$  is the clustering at location  $\ell$  found by fragmenting  $\mathcal{R}_\ell$ . Since we are considering diploid sequences here, we will write  $a_{i\ell} = (a_{i\ell}^{(1)}, a_{i\ell}^{(2)})$  for the clustering assignment of the two sequences that comprise the  $i$ -th diploid individual. Thus  $a_{i\ell}^{(1)}$  and  $a_{i\ell}^{(2)}$  are blocks of the partition  $\mathcal{R}_\ell$  representing the cluster assignments of the first two sequences comprising the  $i$ -th diploid pair at location  $\ell$ . The notation  $(b_{i\ell}^{(1)}, b_{i\ell}^{(2)})$  is defined in an analogous way.

Because we are considering a Gibbs update for the two sequences comprising the  $i$ -th diploid individual, by  $\mathcal{R}_\ell^{-i}$  (and likewise by  $\mathcal{Q}_\ell^{-i}$ ) we mean the partition of all of the sequences except the two sequences comprising the  $i$ -th diploid individual. So, if there are  $N$  individuals, then  $\mathcal{R}_\ell$  will be a partition of  $2n$  sequences and  $\mathcal{R}_\ell^{-i}$  will be a partition of  $2n - 2$  sequences. Finally, the notation for the cases where the sequences are in clusters by themselves are handled as follows. If  $a^{(1)}$  is in a cluster by itself, we

will write  $a^{(1)} = \emptyset^{(1)}$  (likewise for  $a^{(2)} = \emptyset^{(2)}$ ,  $b^{(1)} = \emptyset^{(1)}$  and  $b^{(2)} = \emptyset^{(2)}$ ). If  $a^{(1)}$  and  $a^{(2)}$  are both in the same cluster, but no other sequence is in that cluster, we will write  $a^{(1)} = a^{(2)} = \emptyset$ . Thus, by  $(a^{(1)}, a^{(2)}) = (\emptyset^{(1)}, \emptyset^{(2)})$  we mean that  $a^{(1)}$  and  $a^{(2)}$  are in separate clusters, each of size 1 and by  $(a^{(1)}, a^{(2)}) = (\emptyset, \emptyset)$  we mean that  $a^{(1)}$  and  $a^{(2)}$  are in the same cluster, a cluster of size 2 (i.e., one that is not in  $\mathcal{R}^{-i}$ ).

The joint distributions for the cluster assignment of  $b_{i\ell}^{(1)}$  and  $b_{i\ell}^{(2)}$  under fragmentation and the joint distribution for the cluster assignment of  $a_{i\ell}^{(1)}, a_{i\ell}^{(2)}$  under coagulation are given in the following display. For brevity, we suppress the location subscripts ( $\ell$  or  $\ell + 1$ ) on the right hand side of the equations.

$$\begin{aligned}
& \Pr \left( b_{i\ell}^{(1)} = b^{(1)}, b_{i\ell}^{(2)} = b^{(2)} \mid a_{i\ell}^{(1)} = a^{(1)}, a_{i\ell}^{(2)} = a^{(2)}, \mathcal{R}_\ell^{-i}, \mathcal{Q}_\ell^{-i} \right) \\
&= \left\{ \begin{array}{ll}
\frac{\#b-d}{\#a} \cdot \frac{\#b+1-d}{\#a+1} & \text{if } a^{(1)} = a^{(2)} = a \in \mathcal{R}^{-i}, b^{(1)} = b^{(2)} = b \in F(a), \\
\frac{\#b^{(1)}-d}{\#a} \cdot \frac{\#b^{(2)}-d}{\#a+1} & \text{if } a^{(1)} = a^{(2)} = a \in \mathcal{R}^{-i}, b^{(1)} \neq b^{(2)}, \text{ and } b^{(1)}, b^{(2)} \in F(a), \\
\frac{d\#F(a)}{\#a} \cdot \frac{\#b^{(2)}-d}{\#a+1} & \text{if } a^{(1)} = a^{(2)} = a \in \mathcal{R}^{-i}, b^{(1)} = \emptyset^{(1)}, b^{(2)} \in F(a), \\
\frac{d\#F(a)}{\#a} \cdot \frac{d(\#F(a)+1)}{\#a+1} & \text{if } a^{(1)} = a^{(2)} = a \in \mathcal{R}^{-i}, b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, \\
\frac{d\#F(a)}{\#a} \cdot \frac{1-d}{\#a+1} & \text{if } a^{(1)} = a^{(2)} = a \in \mathcal{R}^{-i}, b^{(1)} = \emptyset, b^{(2)} = \emptyset, \\
\frac{\#b^{(1)}-d}{\#a^{(1)}} \cdot \frac{\#b^{(2)}-d}{\#a^{(2)}} & \text{if } a^{(1)}, a^{(2)} \in \mathcal{R}^{-i}, a^{(1)} \neq a^{(2)}, b^{(1)} \in F(a^{(1)}), b^{(2)} \in F(a^{(2)}), \\
\frac{\#b^{(1)}-d}{\#a^{(1)}} \cdot \frac{d\#F(a^{(2)})}{\#a^{(2)}} & \text{if } a^{(1)}, a^{(2)} \in \mathcal{R}^{-i}, a^{(1)} \neq a^{(2)}, b^{(1)} \in F(a^{(1)}), b^{(2)} = \emptyset^{(2)}, \\
\frac{d\#F(a^{(1)})}{\#a^{(1)}} \cdot \frac{d\#F(a^{(2)})}{\#a^{(2)}} & \text{if } a^{(1)}, a^{(2)} \in \mathcal{R}^{-i}, a^{(1)} \neq a^{(2)}, b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, \\
\frac{d\#F(a^{(1)})}{\#a^{(1)}} & \text{if } a^{(1)} \in \mathcal{R}^{-i}, a^{(2)} = \emptyset^{(2)}, b^{(1)} \in F(a^{(1)}), b^{(2)} = \emptyset^{(2)}, \\
\frac{1-d}{\#a^{(1)}} & \text{if } a^{(1)} \in \mathcal{R}^{-i}, a^{(2)} = \emptyset^{(2)}, b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, \\
1 & \text{if } a^{(1)} = \emptyset^{(1)}, a^{(2)} = \emptyset^{(2)}, b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, \\
1-d & \text{if } a^{(1)} = a^{(2)} = \emptyset, b^{(1)} = b^{(2)} = \emptyset, \\
d & \text{if } a^{(1)} = a^{(2)} = \emptyset, b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, \\
0 & \text{otherwise.}
\end{array} \right. \tag{4.23}
\end{aligned}$$

$$\begin{aligned}
& \Pr \left( a_{i,\ell+1}^{(1)} = a^{(1)}, a_{i,\ell+1}^{(2)} = a^{(2)} \mid b_{i\ell}^{(1)} = b^{(1)}, b_{i\ell}^{(2)} = b^{(2)}, \mathcal{R}_{\ell+1}^{-i}, \mathcal{Q}_\ell^{-i} \right) \\
&= \left\{ \begin{array}{ll}
\frac{d\#C(a)}{\alpha+d\#\mathcal{Q}_\ell^{-i}} \cdot \frac{d(\#C(a)+1)}{\alpha+d(\#\mathcal{Q}_\ell^{-i}+1)} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, a^{(1)} = a^{(2)} = a \in \mathcal{R}_{\ell+1}^{-i}, \\
\frac{d\#C(a^{(1)})}{\alpha+d\#\mathcal{Q}_\ell^{-i}} \cdot \frac{d\#C(a^{(2)})}{\alpha+d(\#\mathcal{Q}_\ell^{-i}+1)} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, a^{(1)}, a^{(2)} \in \mathcal{R}_{\ell+1}^{-i}, a^{(1)} \neq a^{(2)}, \\
\frac{d\#C(a)}{\alpha+d\#\mathcal{Q}_\ell^{-i}} \cdot \frac{\alpha}{\alpha+d(\#\mathcal{Q}_\ell^{-i}+1)} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, a^{(1)} \in \mathcal{R}_{\ell+1}^{-i}, a^{(2)} = \emptyset^{(2)}, \\
\frac{\alpha}{\alpha+d\#\mathcal{Q}_\ell^{-i}} \cdot \frac{R}{\alpha+d(\#\mathcal{Q}_\ell^{-i}+1)} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, a^{(1)} = a^{(2)} = \emptyset, \\
\frac{\alpha}{\alpha+d\#\mathcal{Q}_\ell^{-i}} \cdot \frac{\alpha}{\alpha+d(\#\mathcal{Q}_\ell^{-i}+1)} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} = \emptyset^{(2)}, a^{(1)} = \emptyset^{(1)}, a^{(2)} = \emptyset^{(2)}, \\
\frac{\alpha}{\alpha+d\#\mathcal{Q}_\ell^{-i}} & \text{if } b^{(1)} = b^{(2)} = \emptyset, a^{(1)} = a^{(2)} = \emptyset, \\
\frac{d\#C(a)}{\alpha+d\#\mathcal{Q}_\ell^{-i}} & \text{if } b^{(1)} = b^{(2)} = \emptyset, a^{(1)} = a^{(2)} = a \in \mathcal{R}_{\ell+1}^{-i}, \\
\frac{\alpha}{\alpha+d\#\mathcal{Q}_\ell^{-i}} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} \in C(a^{(2)}), \text{ where } a^{(1)} = \emptyset^{(1)}, a^{(2)} \in \mathcal{R}_{\ell+1}^{-i}, \\
\frac{d\#C(a^{(1)})}{\alpha+d\#\mathcal{Q}_\ell^{-i}} & \text{if } b^{(1)} = \emptyset^{(1)}, b^{(2)} \in C(a^{(2)}), \text{ where } a^{(1)} = \emptyset^{(1)}, a^{(2)} \in \mathcal{R}_{\ell+1}^{-i}, \\
1 & \text{if } b^{(1)} \in C(a^{(1)}), b^{(2)} \in C(a^{(2)}), \text{ where } a^{(1)}, a^{(2)} \in \mathcal{R}_{\ell+1}^{-i}, \\
0 & \text{otherwise.}
\end{array} \right. \tag{4.24}
\end{aligned}$$

In the above piecewise functions, for the cases that the conditions are symmetric in  $a^{(1)}$  and  $a^{(2)}$  or  $b^{(1)}$  and  $b^{(2)}$ , only one of the possible identical conditions are listed for brevity. For example, the condition  $b^{(1)} = \emptyset^{(1)}, b^{(2)} \in C(a^{(2)})$ , where  $a^{(1)} = \emptyset^{(1)}, a^{(2)} \in \mathcal{R}_{\ell+1}^{-i}$  is identical to the condition  $b^{(1)} \in C(a^{(1)}), b^{(2)} = \emptyset^{(2)}$ , where  $a^{(1)} \in \mathcal{R}_{\ell+1}^{-i}, a^{(2)} = \emptyset^{(2)}$ , except with  $a^{(1)}$  and  $a^{(2)}$  reversed and  $b^{(1)}$  and  $b^{(2)}$  reversed. This second condition

does not appear in the piecewise function as it can be inferred by the probability listed for the first condition. (Here, as before  $C(a)$  is the set of blocks that coagulate to form  $a$  and  $F(a)$  is the set of blocks that  $a$  fragments into).

In order to link the genotype of an individual to a likelihood, we must define an ordering of the alleles at heterozygous locations. We do this by introducing a latent variable  $\eta_{is}$ . The value  $\eta_{is}$  is defined for each heterozygous location  $s$  for individual  $i$  and it indicates whether the minor allele is emitted from the cluster  $a_s^{(1)}$  or from the cluster  $a_s^{(2)}$ . The prior on  $\eta_{is}$  by symmetry is  $\Pr(\eta_{is} = 1) = 1/2$ .

The likelihood is based on a Bernoulli model with deterministic output. So:

$$\Lambda(x_{i\ell}|\theta_\ell, a_{i\ell}^{(1)}, a_{i\ell}^{(2)}, \eta) = \begin{cases} 1 & \text{if } \eta = 1 \text{ and } \theta_{\ell, a_{i\ell}^{(1)}} = x_{i\ell}^{(1)}, \theta_{\ell, a_{i\ell}^{(2)}} = x_{i\ell}^{(2)}, \\ 1 & \text{if } \eta = 0 \text{ and } \theta_{\ell, a_{i\ell}^{(1)}} = x_{i\ell}^{(2)}, \theta_{\ell, a_{i\ell}^{(2)}} = x_{i\ell}^{(1)}, \\ 0 & \text{otherwise.} \end{cases} \quad (4.25)$$

Thus, the messages for genotype imputation with unphased data are defined as follows:

$$m_C^\ell(a^{(1)}, a^{(2)}) = \Pr(x_{i, \ell+1:L} | a_{i\ell}^{(1)} = a^{(1)}, a_{i\ell}^{(2)} = a^{(2)}, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:L}^{-i}) \quad (4.26)$$

$$m_{\mathcal{F}}^\ell(b^{(1)}, b^{(2)}, \eta) = \Pr(x_{i, \ell+1:L} | b_{i\ell}^{(1)} = b^{(1)}, b_{i\ell}^{(2)} = b^{(2)}, \eta_{i\ell} = \eta, \mathcal{R}_{\ell:L}^{-i}, \mathcal{Q}_{\ell:L}^{-i}) \quad (4.27)$$

The domain of  $(a^{(1)}, a^{(2)})$  is  $(\mathcal{R}_\ell^{-i} \cup \{\emptyset^{(1)}\}) \times (\mathcal{R}_\ell^{-i} \cup \{\emptyset^{(2)}\}) \cup \{(\emptyset, \emptyset)\}$ . There are two possibilities for the domain of  $(b^{(1)}, b^{(2)}, \eta)$ . First, if  $x_{i\ell}$  is heterozygous, then the domain of  $(b^{(1)}, b^{(2)}, \eta)$  is  $((\mathcal{Q}_\ell^{-i} \cup \{\emptyset^{(1)}\}) \times (\mathcal{Q}_\ell^{-i} \cup \{\emptyset^{(2)}\})) \cup \{(\emptyset, \emptyset)\} \times \{0, 1\}$  if  $x_{i\ell}$  is heterozygous. Otherwise, if  $x_{i\ell}$  is homozygous then the domain of  $(b^{(1)}, b^{(2)}, \eta)$  is  $((\mathcal{Q}_\ell^{-i} \cup \{\emptyset^{(1)}\}) \times (\mathcal{Q}_\ell^{-i} \cup \{\emptyset^{(2)}\})) \cup \{\emptyset, \emptyset\}$ . Note that  $\eta$  only appears in the messages for  $m_{\mathcal{F}}^\ell$  and not  $m_C^\ell$  because the phase only affects the probability of the data through the clustering  $\mathcal{R}_\ell$  (and not through  $\mathcal{C}_\ell$ ). By their definition, these messages can be computed recursively as follows:

$$m_{\mathcal{F}}^\ell(b^{(1)}, b^{(2)}, \eta) = \frac{1}{2} \sum_{(a^{(1)}, a^{(2)})} m_C^\ell(a^{(1)}, a^{(2)}) \overbrace{\Lambda(x_{i, \ell+1} | \theta_{\ell+1}, a^{(1)}, a^{(2)}, \eta)}^{\text{Likelihood from (4.25)}}, \quad (4.28)$$

$$\cdot \underbrace{\Pr(a_{i\ell+1}^{(1)} = a^{(1)}, a_{i\ell+1}^{(2)} = a^{(2)} | b_{i\ell}^{(1)} = b^{(1)}, b_{i\ell}^{(2)} = b^{(2)}, \mathcal{R}_{\ell+1}^{-i}, \mathcal{Q}_\ell^{-i})}_{\text{Coagulation probabilities from (4.24)}}$$

$$m_C^\ell(a^{(1)}, a^{(2)}) = \sum_{(b^{(1)}, b^{(2)}, \eta)} m_{\mathcal{F}}^\ell(b^{(1)}, b^{(2)}, \eta) \underbrace{\Pr(b_{i\ell}^{(1)} = b^{(1)}, b_{i\ell}^{(2)} = b^{(2)} | a_{i\ell}^{(1)} = a^{(1)}, a_{i\ell}^{(2)} = a^{(2)}, \mathcal{R}_\ell^{-i}, \mathcal{Q}_\ell^{-i})}_{\text{Fragmentation probabilities from (4.23)}}. \quad (4.29)$$

These messages can be further expanded over their support using equations (4.23) and (4.24). By using the fact that the fragmentation and coagulation probabilities in equations (4.23) and (4.24) are zero over much of the support, the summations in (4.29) can be restricted to a subset of the support, adding efficiency to the message



computations. Due to their complexity, the expanded forms are not provided here.

#### 4.2.6 Phasing

Phasing algorithms are often designed to minimize the switch error of the proposed phasing of the sequence of genotypes (Scheet and Stephens, 2006). The switch error of a proposed phasing is defined to be the minimum number of crossovers required to map the proposed phasing onto the true pair of sequences. Each state in the chain of MCMC states in a posterior simulation of the DFCP induces a proposed phasing. In Scheet and Stephens (2006), the authors propose a phasing by taking each pair of consecutive heterozygous sites and choosing the phase between those two sites based on the most frequent phase occurring in the chain of MCMC states. We can adopt this method by using the messages derived in section 4.2.5.

We note that this method of choosing the most frequent phase from the MCMC corresponds to choosing the estimate for the phase that minimizes the Bayes risk. Suppose that  $\eta$  is a random object with density  $p(\eta)$ , and  $L(\eta, \eta')$  is a loss function. The Bayes risk (Lehmann and Casella, 1998) of the estimate  $\eta'$  is the expected loss  $\mathbb{E}_p(L(\eta, \eta'))$ . In our case,  $L(\eta, \eta')$  is the switch error between the true phasing ( $\eta$ ) and the proposed phasing ( $\eta'$ ). The switch error,  $L(\eta, \eta')$  is defined as the sum of Kronecker delta functions, one for each pair of consecutive heterozygous sites, which measures whether or not the minor alleles for consecutive heterozygous sites are on the same chromosomes in the phasings  $\eta$  and  $\eta'$ . Since expected value is linear,  $\mathbb{E}_p(L(\eta, \eta'))$  splits over each of the Kronecker delta functions. Minimizing  $\mathbb{E}_p(L(\eta, \eta'))$  thus reduces to minimizing the Kronecker delta functions at each pair, which is equivalent to setting the phase of  $\eta'$  to the empirical median estimate of  $\eta$  from the samples produced by the MCMC. This means that to phase genetic sequence data using equations (4.28) and (4.29), we can run MCMC using those messages, and then after discarding burn-in, we set  $\eta_\ell = \operatorname{argmax}_{\eta'} \#\{t : \eta_\ell^{(t)} = \eta'\}$ , where  $\eta_\ell^{(t)}$  are the values of the phase over MCMC samples indexed by  $t$ .

#### 4.2.7 The length of a haplotype

In this section, we study the expected length of haplotypes in the DFCP model. Under the genetic assumptions from section 1.2.1, we expect the recombination rate and the mutation rate to both affect the length of haplotypes. As we increase the number of individuals observed we would also expect the length of the haplotypes in the sample to decrease. This is because of the following phenomenon: as we observe more individuals we will tend to observe more mutations that occur with low frequency; these are known as rare variants. For the mutation models discussed in section 1.2.2, the amount of variation is proportional to the total tree size of the genealogies (this is discussed in section 1.2.2).

In order to understand the distribution of the length of haplotypes in the DFCP model, we will study the probability that a haplotype extends for just one more site (call this probability  $p$ ). We will find that this probability  $p$  can be computed exactly. If the extension of a haplotype for just one more site were independent of the length of the haplotype, then haplotype lengths would be distributed as a negative binomial with rate  $p$  and 1 failure. However, due to statistical dependence between the partition structure and the haplotype length, the haplotype length is not simply the number of independent successful extensions before the first failure to extend (*i.e.*, the mean of a negative binomial). Despite this, we will provide empirical estimates of this quantity and compare them to an approximation of the haplotype length in which we assume that the haplotype length is distributed as a negative binomial.

A haplotype in the DFCP is defined as latent block  $a$  of the partitions  $\mathcal{R}_\ell, \mathcal{Q}_\ell$  that does not fragment into other blocks and does not coagulates with other blocks for some number of steps. The length of the haplotype is simply the number of steps in which no fragmentation nor coagulation events involving the block  $a$  occur. We will now compute the probability  $p(a)$  that a the block  $a \in \mathcal{R}_\ell$  does not experience fragmentation or coagulation in the transitions  $\mathcal{R}_\ell \rightarrow \mathcal{Q}_\ell \rightarrow \mathcal{R}_{\ell+1}$ . This quantity is a function of the DFCP parameters  $\alpha$  and  $d$  (we will assume  $d_\ell = d$  is constant). We will also find that we must marginalize the partition  $\mathcal{Q}_\ell$  in order to arrive at  $p(a)$ .

Let  $a$  be a block in a partition  $\mathcal{R}_\ell$  for a DFCP on the index set  $R = \{1, \dots, N\}$ . Suppose that the size of  $a$  is  $m$  (so,  $\#a = m$ ). The value of  $p(a)$  is found as follows:

1. We will first consider fragmentation. Since  $\mathcal{Q}|\mathcal{R}, d \sim \text{FRAG}(\mathcal{R}, 0, d)$ , the probability that  $a$  does not fragment is given by:

$$\frac{\Gamma(m-d)}{\Gamma(1-d)\Gamma(m)}. \quad (4.30)$$

This is found by considering the CRP( $a, 0, d$ ) for the fragmentation applied to block  $a$  of  $\mathcal{R}$ . For no fragmentation to occur, each item of  $a$  must be added to the same block of that CRP. The first item must create a new block (an event that occurs with probability 1). The second item has two choices: start a new block with probability  $d$ , or join the same block as the first item with probability  $1-d$ . Suppose that all previous items joined the same block as the first item. In this case, a subsequent item  $i > 2$  would have the same two choices: start a new block with probability  $d$ , or join the same block as the first item with probability  $i-d$ . Thus, the probability that the  $i$ -th item joins the same block as the first item is  $(i-1-d)/(i-1)$ . The product of these probabilities yields the equation (4.30) which is listed above.

2. For coagulation, suppose that  $R|\mathcal{Q}, \alpha, d \sim \text{COAG}(\mathcal{Q}, \alpha/d, 0)$ . By exchangeability of the CRP we suppose that the block  $a$  is the last item to be added to the process CRP( $\mathcal{Q}, \alpha/d, 0$ ) that describes the coagulation. By the sequential construction

of the CRP, the probability that block  $a$  is placed in a cluster by itself is given by:

$$\frac{\alpha/d}{\alpha/d + \#\mathcal{Q} - 1}. \quad (4.31)$$

The value of  $p(a)$  conditioned on  $\mathcal{Q}$  is given by the product of equations (4.30) and (4.31) (this is therefore the probability that no fragmentation occurs and that no coagulation occurs):

$$p(a, \mathcal{Q}) = \frac{\alpha\Gamma(m-d)}{(\alpha+d(\#\mathcal{Q}-1))\Gamma(1-d)\Gamma(m)} \quad (4.32)$$

We will now marginalize over  $\#\mathcal{Q}$  to find the expected value of  $p(a)$ . By the definition of the DFCP, in the prior  $\mathcal{Q}$  is distributed as  $\text{CRP}(R, \alpha, d)$ . But since the block  $a$  does not fragment in the fragmentation step  $\mathcal{R} \rightarrow \mathcal{Q}$ , we must condition on the event that  $\mathcal{Q}$  has a block of size  $m$  (*i.e.*, we need the distribution of  $\mathcal{Q}|a \in \mathcal{Q}$ ). According to the sequential scheme for the two parameter version of the CRP given after equation (2.7), the induced distribution of the random partition  $\mathcal{Q} \setminus \{a\}$  on the set  $R \setminus a$  has law  $\text{CRP}(R \setminus a, \alpha + d, d)$ . (Here ‘ $\setminus$ ’ denotes the set minus operation.) The addition of  $d$  to the concentration parameter can be seen by normalizing the events of the sequential scheme conditioned on the event that no subsequent items join the first cluster of the CRP (this is similar to the concept of exponential tilting for Dirichlet processes).

The distribution of the number of blocks in the two parameter version of the CRP is given in (Pitman, 2006) as follows. If  $\mathcal{A} \sim \text{CRP}(\{1, \dots, N\}, \alpha, d)$ , then:

$$\Pr(\#\mathcal{A} = k) = \frac{[\alpha + d]_d^{k-1} S_{N,k}^{-1,-d}}{[\alpha + 1]_1^{N-1}}. \quad (4.33)$$

Here  $[x]_1^m = x(x+1)\dots(x+m-1)$  is Kramp’s symbol (for  $m \in \mathbb{N}$ ) and  $S_{N,k}^{-1,-d}$  is a generalized Stirling number of the first kind (Toscano, 1939). In particular:

$$S_{N,k}^{-1,-d} = \text{the coefficient of } \xi^N \text{ in } \frac{N!}{k!} \left( \sum_{j=1}^{\infty} [1-d]_{-d}^{j-1} \frac{\xi^j}{j!} \right)^k. \quad (4.34)$$

Thus, after making the substitution into equation (4.33) for the concentration parameter  $\alpha + d$ , discount parameter  $d$  and size  $\#(R - a) = N - m$ , the value of  $p(a)$  is given by the following equation:

$$p(a) = \mathbb{E}_{\mathcal{Q}}[p(a, \mathcal{Q})] = \sum_{k=1}^{N-m} \frac{\alpha\Gamma(m-d)d^{k-1}\Gamma(\alpha+d+1)\Gamma(\alpha/d+k+1)S_{N-m,k}^{-1,-d}}{(\alpha+d(k-1))\Gamma(1-d)\Gamma(m)\Gamma(\alpha/d+2)\Gamma(\alpha+d+N)}. \quad (4.35)$$

Here we have used the following identity for Kramp’s symbol:  $[x]_d^N = d^N\Gamma(x/d +$

$N)/\Gamma(x/d)$ , when  $d > 0$ , and  $N \in \mathbb{Z}_+$ . We compute the numerical value of (4.35) for various settings of  $\alpha, d$  and  $N$  and also simulate from the DFCP prior with the same settings, in order to provide an empirical estimate of this distribution. The computation of the numerical values was done using `MATHEMATICA 10` and the code used is provided in Algorithm 4.1. To find the empirical estimate, we simulate from the DFCP prior for one step and fix a cluster with a given size and record whether or not it experiences a fragmentation or coagulation event for that step. The results of these computations and simulations are given in Figure 4.3.

---

**Algorithm 4.1** Computation of expected haplotype lengths for the DFCP model, `MATHEMATICA 10` code. Input: Concentration parameter  $a$ , rate parameter  $d$ , size  $n$ . Output: Enumeration of expected lengths of haplotypes of sizes  $m = 1, \dots, n$  for DFCP prior with concentration  $\alpha$  and discount  $d$  on  $n$  individuals.

---

```

GeneralizedStirlingS1[A_, B_, N_, K_] :=
  Sum[
    StirlingS1[N, j] *
    StirlingS2[j, K] *
    A^(N - j) *
    B^(j - K),
    {j, K, N}];

PADK[A_, D_, K_, N_] :=
  D^(K - 1)*Gamma[A + 1]/Gamma[A/D + 1]*
  Gamma[A/D + K]/Gamma[A + N]*
  GeneralizedStirlingS1[-1, -D, N, K];

NOEVENT[A_, D_, K_, N_, M_] :=
  A*Gamma[M - D]/(
    (A + D*(K - 1))*
    Gamma[1 - D]*
    Gamma[M]);

PMQ[A_, D_, K_, N_, M_] :=
  NOEVENT[A, D, K, N, M]*PADK[A + D, D, K, N - M];

PM[A_, D_, N_, M_] :=
  Sum[PMQ[A, D, k, N, M], {k, 1, N - M}];

LM[A_, D_, N_, M_] := PM[A, D, N, M]/(1 - PM[A, D, N, M]);

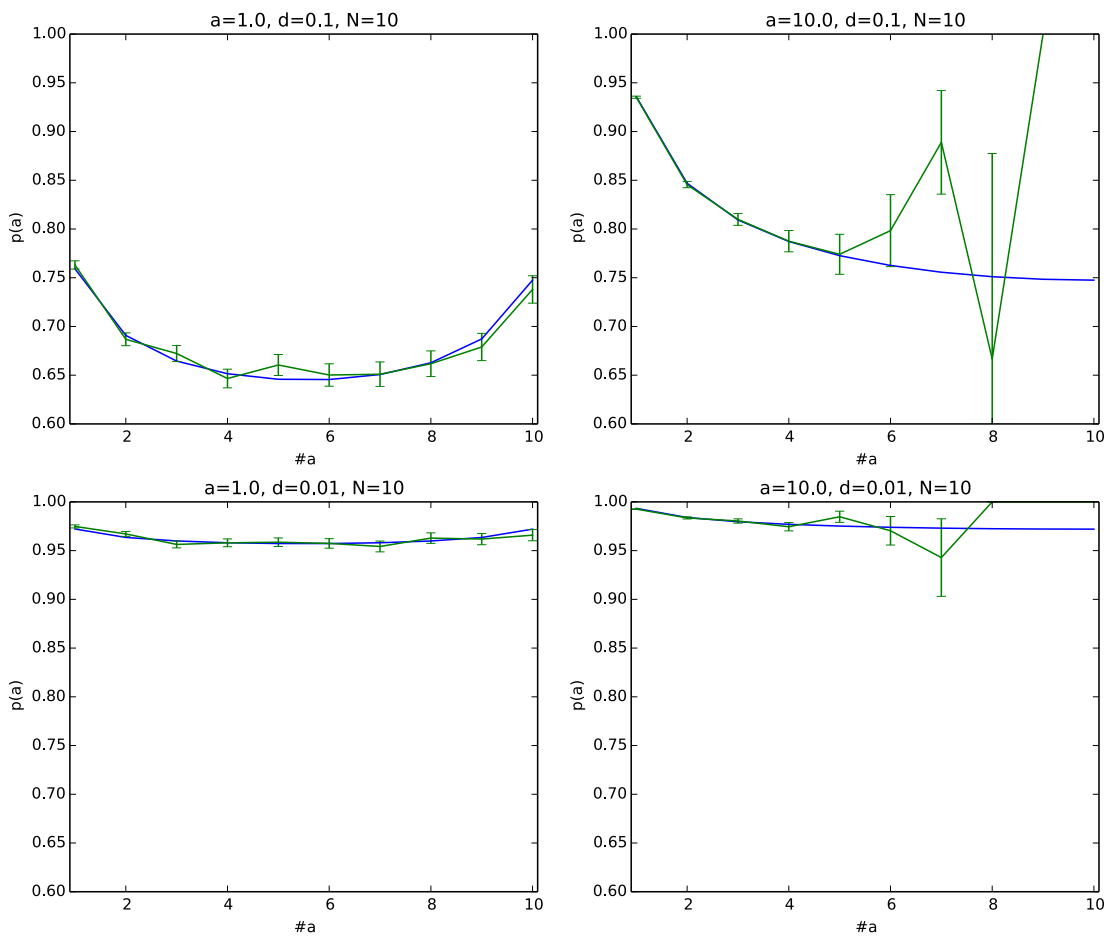
# Modify these lines to specify input
a := 1.0;
d := 0.1;
n := 10;

# Output stored in 'Result'
Result := Table[{m, LM[a, d, n, m]}, {m, 1, n - 1}];

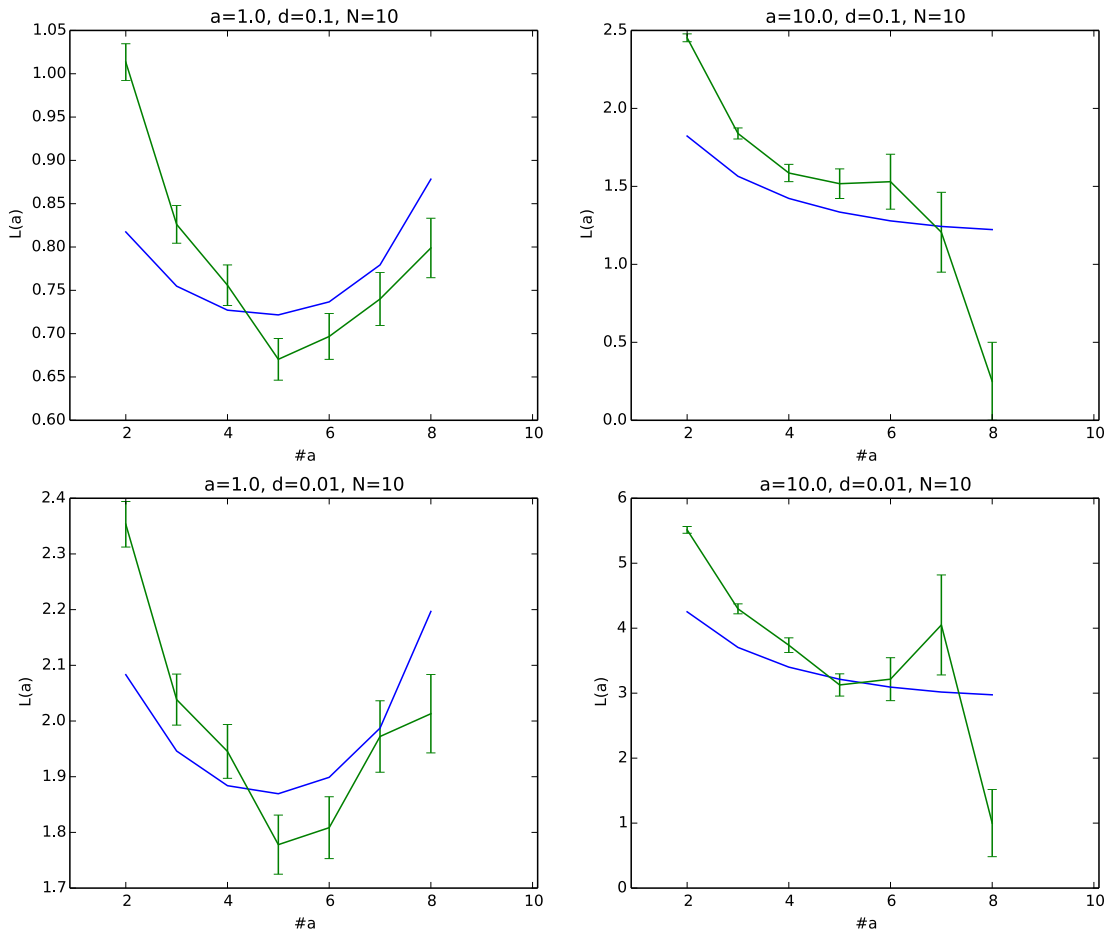
```

---

If we condition on the event that a haplotype of size  $m$  extends one step to the right



**Figure 4.3:** Probability of extending a haplotype.  $y$ -axis indicates probability of haplotypes extending and  $x$ -axis indicates number of individuals sharing the haplotype. Blue line indicates the actual probability. Green line indicates mean empirical estimate from DFCP prior simulation. Error bars indicate the standard error of the mean. Conditions are listed in plot titles.



**Figure 4.4:** Approximation for haplotype lengths.  $y$ -axis indicates length of haplotypes and  $x$ -axis indicates number of individuals sharing the haplotype. Blue line indicates approximation of haplotype length. Green line indicates mean empirical estimate from DFCP prior simulation. Error bars indicate the standard error of the mean. Conditions are listed in the title.  $y$ -axis indicates number of sites a haplotype extends beyond the first site, so an expected value of 0.65 indicates a haplotype length of 1.65.

of a location  $\ell$  without fragmenting or coagulating, then through equation (4.31), we get information about  $\mathcal{Q}$ . To compute the probability that the haplotype will extend one more step to the right (*i.e.*, a total of two steps from the first position in which the haplotype is observed), we must form a summation over all possible values of  $m$ . Rather than carrying through with that analysis here, we will instead make the assumption that the extension events are independent. Under this assumption, the number of extensions to the right of  $\ell$  is given by a negative binomial random variable with rate  $p(a)$  (and one failure) and so the expected haplotype length is approximated by  $\mathcal{L}^1(a) = p(a)/(1 - p(a))$ . In Figure 4.4, we compute this quantity and plot it against empirical estimates for the haplotype lengths found through simulating the DFCP posterior. From Figure 4.3 and Figure 4.4, we see that for low values of the concentration parameter  $\alpha$ , the expected haplotype length for haplotypes shared by relatively large and relatively small numbers of individuals tend to be longer. The expected haplotype lengths for

haplotypes shared by roughly half the population tend to be shorter. We also see that the qualitative behavior is mostly captured: under the assumption of independence for the haplotype extension probabilities and approximating these probabilities with  $\mathcal{L}^1(a)$ , the haplotype lengths are correctly estimated within an order of a magnitude (*i.e.*, roughly within 1/10).

In Figure 4.3 and Figure 4.4, we simulated 1,000 draws from the DFCP, and computed the empirical probability of a haplotype extending just one more site, conditioned on the size of the haplotype. In these figures, the green lines correspond to the empirical probabilities of each haplotype extending just one more site. We computed these empirical probabilities by conditioning on each of the 1,000 draws. Therefore, since large haplotypes with many individuals are rarer, in our simulations we observed fewer samples with large haplotypes. This lead to a larger standard error for the right hand sides of Figure 4.3(*upper right, lower right*) and also Figure 4.4(*upper right, lower right*): fewer samples were observed for large haplotype sizes in these conditions, and therefore the variance in the simulation is larger. In fact, for Figure 4.3(*upper right*), only one sample was observed for  $N = 10$  among the 1,000 simulated samples. We note that the effect of simulation variance for these estimates could be made uniform over  $N$  by running more simulations, and discarding simulations until 100 samples remain for each setting of  $N$ . We leave such extensive exploration of these simulations for future work.

We note that for many conditions in these simulations, the probability of a haplotype extending to a particular length exhibits a ‘u’-type shape (this is observed for Figure 4.3 *upper right* and Figure 4.4 *upper left, lower left*). This is related to two competing pressures on the haplotype length. On one hand, haplotypes lengths are encouraged to be short because many events cause them to end. On the other hand, (since long haplotypes include more sites), haplotype lengths are more likely to be observed among the haplotypes that we simulate. Therefore, many haplotypes are exhibited with either short or long lengths, explaining the ‘u’-type shape. The increased probability of large haplotypes is related to the waiting paradox: because more individuals are involved in large haplotypes, they are more likely to be observed.

In the above analyses, we note that the way in which we define haplotypes for the DFCP model is perhaps too conservative. Suppose a partition  $\mathcal{R}_\ell$  experiences a nontrivial fragmentation into the finer partition  $\mathcal{Q}_\ell$ , and then  $\mathcal{Q}_\ell$  coagulates into a partition  $\mathcal{R}_{\ell+1}$ , with the same configuration as the partition  $\mathcal{R}_\ell$  (*i.e.*,  $\mathcal{R}_\ell = \mathcal{R}_{\ell+1}$ ). Under the above analysis, the position  $\ell$  would be considered as the right-most endpoint of a haplotype containing fragmented blocks of  $\mathcal{R}_\ell$ . However, as can be seen by the plate diagram (4.8), the likelihood of observed data is not affected by fragmentations that are immediately reversed by coagulations, and so it is not necessarily correct to include them in the computations of haplotype lengths. In future work, we will extend this analysis to a setting in which  $\mathcal{Q}_\ell$  is marginalized, and so haplotype endpoints are only

reported for  $\mathcal{R}_\ell \neq \mathcal{R}_{\ell+1}$ .

### 4.3 Experiments

To examine the accuracy and scalability of the DFCP we conducted an allele imputation experiment on SNP data from the Thousand Genomes project<sup>1</sup> (The 1000 Genomes Project Consortium, 2010). We also compared the runtime of the samplers for the DFCP and CFDP on data simulated from the coalescent with recombination model (Hudson, 2002). In this section, we describe the setup of these experiments and in section 4.4 we present the results.

For the allele imputation experiment, we considered SNPs from 524 male X chromosomes. We chose 20 intervals uniformly at random, each containing 500 consecutive SNPs. In five conditions we held out nested sets of between 10% and 90% of the alleles uniformly over all pairs of sites and individuals, and used `fastPHASE` (Scheet and Stephens, 2006), `BEAGLE` (Browning and Browning, 2009), `CFDP` (Teh et al., 2011) and the DFCP to predict the held out alleles. For these datasets the mean at-chance accuracy which would be found by always predicting the major alleles was 93.44%. We note that this missing-at-random is not a realistic assumption for genetic data, which often has a structured missingness induced by a study/reference paradigm. Missing-at-random is however a good measure for model fit.

We used the most recent versions of `BEAGLE` and `fastPHASE` software available to us. We implemented the DFCP with many of the same libraries and programming techniques as the CFDP and both versions were optimized. In each missing data condition, the CFDP and DFCP were run with five random restarts and 46 MCMC iterations per restart (26 of which were discarded for burn-in and thinning). We computed accuracies for the DFCP and CFDP by thresholding the empirical marginal probabilities of the held out alleles at 0.5. We matched the priors on the hyper parameters and the likelihood specification of the two models and we initialized the samplers using a sequential Monte Carlo method in which one sequence was added to the model at a time, conditioned on all other previously added sequences.

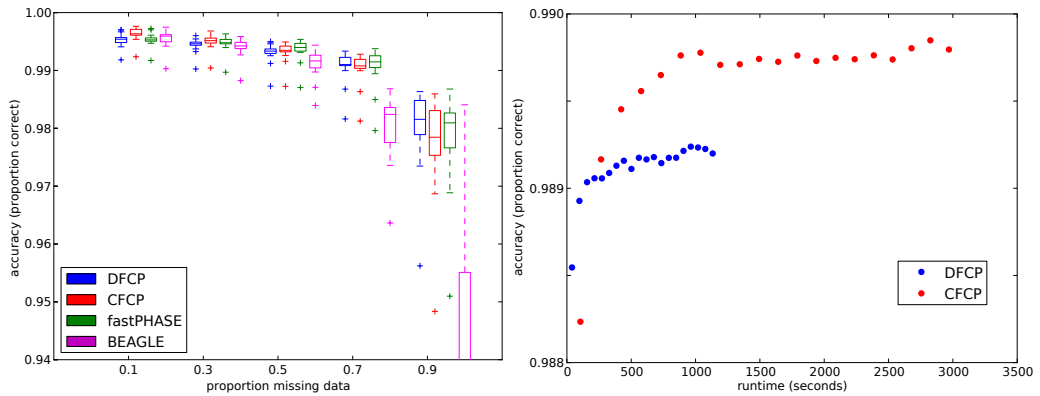
The posterior distributions of the concentration parameter  $\mu$  for the two methods are different. In order to match the expected number of clusters in the posterior, we also conducted allele imputation in the 50% missing data condition with  $\mu$  fixed at 10.0 for both models. We simulated 500 MCMC iterations with no random restarts. We then computed the accuracy of the samples by predicting held out alleles based on the cluster assignments of the sample.

In our second experiment we simulated datasets from the coalescent with recombination model consisting of between 10,000 and 50,000 sequences using the software

---

<sup>1</sup>March 2012 v3 release of the Thousand Genomes Project.

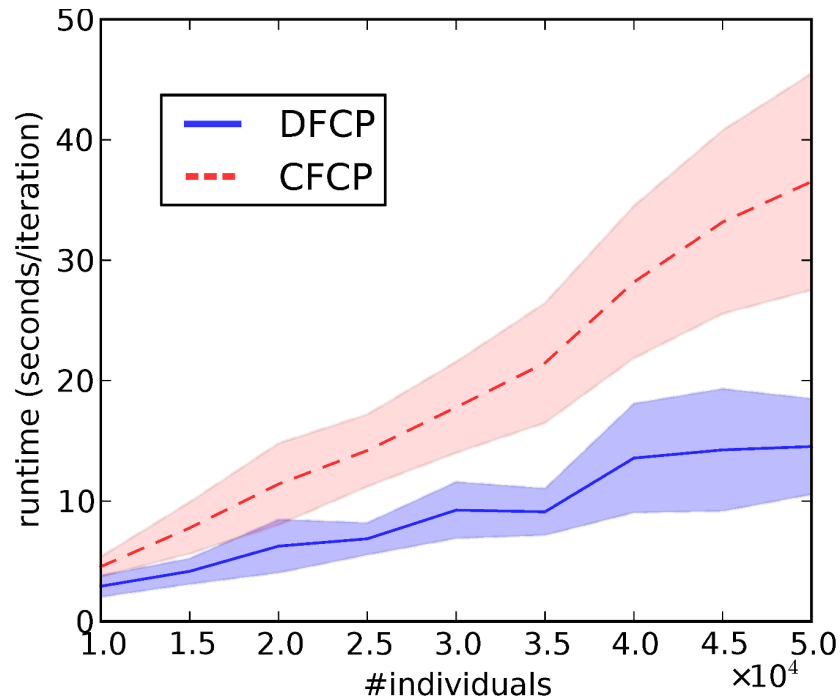




**Figure 4.5:** Allele imputation for X chromosomes from the Thousand Genomes project. *Left:* Accuracy for prediction of held out alleles for continuous (CFCP) and discrete (DFCP) versions of fragmentation-coagulation process and for popular methods BEAGLE and fastPHASE. 90% missing data condition truncates BEAGLE accuracies to emphasize other conditions. *Right:* Runtime versus accuracy for 500 MCMC iterations for DFCP and CFCP in 50% missing data condition. Points are averaged over 20 datasets and 25 consecutive samples.

ms (Hudson, 2002). We conducted posterior MCMC simulation in both models and compared the computation time required per iteration. We performed all MCMC simulations using the same computer system and without computing unnecessary marginal statistics.

In our third experiment, we explored the accuracy of the CFCP model in a study/reference paradigm using two sources of data. The first source of data was unphased data from the SeattleSNPs Project (National Heart, Lung, and Blood Institute Program for Genomic Applications, 2011). This project provides unphased SNP sequences for 320 genes from 47 individuals. The genes had between 13 and 416 SNPs. There were 47 individuals in the study. The second source of data was the phased male X chromosomes from the Thousand Genomes Project in a study/reference paradigm. We examined the same 20 intervals that were used in the first experiment. For both of the data sources, we chose  $q\%$  of the sequences chosen to be in the study panel, and  $p\%$  of the sites chosen to be typed only in the reference panel. We held out  $p\%$  of the sites in the  $q\%$  study sequences. This setup mimics the common situation in which experimenters have access to a densely typed reference panel. More detail about study/reference paradigms is given in section 1.1. We varied  $p\%$  in the range 10%, ..., 50% and we also varied  $q\%$  in the same range, leading to 25 conditions. The inference we used for the CFCP is based on uniformization for MJPs (Rao and Teh, 2011). Details of the inference and the parameter settings we used for the MCMC in these experiments are explained further in Teh et al. (2011).



**Figure 4.6:** Runtimes per iteration per sequence of DFCP and CFCP on simulated datasets consisting of large numbers of sequences. Lines indicate mean. Shaded region indicates standard deviation.

## 4.4 Results

The accuracy of the DFCP in the first allele imputation experiment was comparable to that of the CFCP and `fastPHASE` in all missing data conditions Figure 4.5(left). For the 70% and 90% missing data conditions, `BEAGLE` performed poorly (its median accuracy for this condition was 93.90% and mean at chance accuracy for all conditions was 93.44%). In Figure 4.5(right) we compare the accuracy and runtime for the 50% missing data condition. This figure shows that the runtime required for each iteration is lower for the DFCP than for the CFCP, and the sequential Monte Carlo initialization is better (i.e., closer to a posterior mode) for the DFCP. No difference in mixing time is suggested by the figure. As an aside, we estimated the Shannon entropy in these samples and found that the DFCP had slightly more entropy per sample than the CFCP. (The difference was small but statistically significant under a sign test.) This could indicate that the DFCP has better mixing. Improved mixing in the DFCP is also suggested by the observation that the accuracy for the DFCP plateaus after fewer iterations.

For the second experiment, we plotted the runtime per iteration of both models against the number of sequences in the simulated dataset (Figure 4.6). The DFCP was approximately 2.5 times faster than the CFCP for the condition with 50,000 sequences. In both models, most of the computation time was spent calculating the messages in the backwards-filtering step. The CFCP has an arbitrary number of latent events between consecutive observations and it is likely that the runtime improvement shown by the

DFCP is due to the reduction in the number of required message calculations in the DFCP.

## 4.5 Discussion

The DFCP and CFCP induce different joint distributions on the partitions at adjacent locations. The CFCP is a Markov jump process with an arbitrary number of latent binary events wherein a single cluster is split into two clusters, or two clusters are merged into one. The DFCP however can model any partition structure with one pair of fragmentation and coagulation operations. Exact Gibbs updates for the partitions are possible in the DFCP whereas sampling in the CFCP uses uniformization (Rao and Teh, 2011).

In future work we will explore better calling and calibration methods to improve imputation accuracies. Another avenue of future research is to understand how other genetic processes can be incorporated into the fragmentation-coagulation framework, including population admixture and gene conversion. Although haplotype structure is a local property, the Markov assumption does not hold in real genetic data. This could be reflected through hierarchical FCP models or adaptation of other dependent nonparametric models such as the spatially normalized Gamma process (Rao and Teh, 2009).

## 4.6 Conclusion

In this Chapter we have presented a discrete fragmentation-coagulation process. The DFCP is a partition-valued Markov chain, where partitions change along the chromosome by a fragmentation operation followed by a coagulation operation. The DFCP is designed to model the mosaic haplotype structure observed in genetic sequences.

We derived message passing for the DFCP based on the conditional distributions for the fragmentation and coagulation operators defined in Chapter 2. Through message passing, efficient forwards-filtering/backwards-sampling updates can be derived for the block assignment of each sequence in the DFCP. We also extended the message passing to handle unphased genotypes and we showed that the method of minimizing switch error in phasing from Scheet and Stephens (2006) is equivalent to minimizing Bayes risk.

We applied the DFCP to an allele prediction task on data from the Thousand Genomes Project yielding accuracies comparable to state-of-the-art methods and runtime requirements that were shorter than the runtime requirements of the continuous fragmentation-coagulation process (Teh et al., 2011). Although the asymptotic computation cost of

inference in the DFCP is the same as for the CFCP, we have found that the runtime requirements were shorter for the DFCP than for the CFCP.

## Chapter 5

# The Wright-Fisher partition valued processes

### 5.1 Introduction

In this Chapter, we present a new Bayesian nonparametric model for dynamic partitions in which the clusters of the partitions shrink and grow according to balanced rates. The model is Markov, exchangeable, and reversible and its marginals are given by the CRP distribution on partitions. Our model is based on a continuous version of the Wright-Fisher diffusion for a countable set of species (Donnelly and Kurtz, 1996). We use our model as a prior on dynamic partitions and we conduct posterior inference using the particle Gibbs (PG) variant of particle MCMC (Andrieu et al., 2010). The PG is implemented through a probabilistic program (Wood et al., 2014; Paige and Wood, 2014). Particle Gibbs is applicable to our model even though the conditional distributions of the cluster assignments in our model are not Markov (this is shown in section 5.2). In previous Chapters, we have applied models of dynamic-clustering to genetic data. To demonstrate the versatility of these models and illustrate their application in a problem domain other than genetics, in this Chapter we will apply our model to voting data from the Canadian House of Commons.

Our model, which we refer to as the WFP (for *Wright-Fisher partition valued diffusion process*) is described by a Markov jump process (MJP) that takes values in the set of partitions of  $N$  items. MJPs are characterized by their initial distribution and their transition rates. The initial partition of the WFP is drawn from the CRP (Pitman, 2006) distribution with concentration parameter  $\alpha$ . With constant rate  $rN(N - 1 + \alpha)$ , the process transitions by choosing an element at random, removing it from the partition, and then adding it back again according to the CRP marginal probabilities. A sample from the WFP prior is shown in Figure 5.1.

To model data with an WFP prior, we assume that covariates and observations associated

with the  $N$  data items are available at points along a one dimensional axis. The WFP specifies a latent clustering structure  $\mathcal{R}_t$  at each point  $t$ . For each observation, if the observation occurs at the point  $t'$ , then the clustering structure  $\mathcal{R}_{t'}$  parameterizes the likelihood function of that observation. In particular, we assume that there is a latent parameter  $\theta_{t'a}$  associated with each cluster  $a \in \mathcal{R}_{t'}$ . Further, we assume that the observation for the  $i$ -th data item at the point  $t'$  is drawn from  $f_{\theta_{t'a}}$  where  $a$  is the cluster that  $i$  belongs to at  $t'$ , and  $f_\theta$  is a distribution function parameterized by  $\theta$ , representing the likelihood of the data.

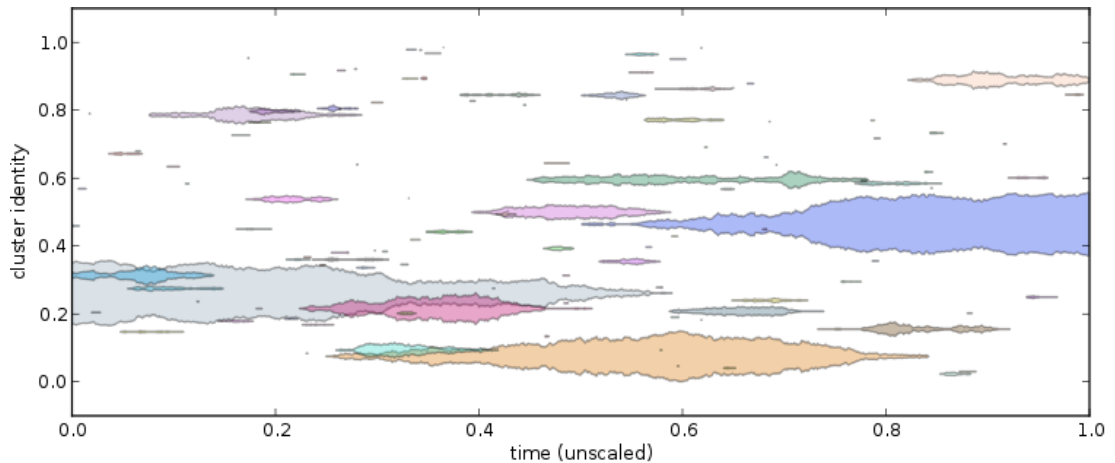
In our experiments, we will be interested in modelling votes. Suppose that  $N$  members of parliament (MPs) vote on motions. The clusters of  $\mathcal{R}_t$  represent similarity in voting patterns (*i.e.*, political parties or political blocs within parties). We apply this model to predicting the votes of MPs and also to discovering political blocs.

In the remainder of this section we discuss the relation of the WFP to the Wright-Fisher model and other models in genetics. We describe the WFP and the likelihood models we will use in our experiments through a generative process. We describe the relation of the WFP to other recent work in Bayesian nonparametrics and survey the relationship between dynamic partitions and distant dependent processes. In section 5.2 we describe the construction of the WFP through a sequential process and we prove its statistical properties. In section 5.3 we apply this inference method to model voting data from the Canadian House of Commons. We show that the WFP, using only data from voting behavior, can be used to detect changes in the party allegiances of members of parliament. We also show that it can be used to predict voting behavior.

### 5.1.1 Relation to work in genetics

The Wright-Fisher model is usually thought of as a discrete coalescent model for a constant population of  $N$  individuals (Fisher, 1930; Wright, 1931). In the Wright-Fisher model, each successive generation chooses one individual (or two individual sequences, in the case of a diploid model) from the previous generation and inherits all material from that individual. The Wright-Fisher model has been extended to continuous diffusion models with mutations (Dawson and Hochberg, 1982), and we use this extension as a basis for the inference defined directly on the space of partitions.

In genetics the Wright-Fisher model is used as a model for  $K$  species (for example, to model the proportions of the population sizes of  $K$  species in an ecosystem). The WFP can be viewed as a version of the Wright-Fisher model defined directly on the space of partitions of a set. The resulting dynamic-clustering can be used as a clustering of genetic sequences along the chromosome. Like the `fastPHASE` and `BNPPHASE` models, the WFP provides a location varying clustering in which the proportions of the clusters, and the tendency of individuals to join each of the clusters, is a function of the chromosome location. Unlike the `fastPHASE` and `BNPPHASE` models, the WFP is a reversible process.



**Figure 5.1:** A sample from the WFP prior with  $N = 100$  items and concentration  $\alpha = 2.0$ .  $x$ -axis represents time. Clusters are identified by color and random (uniform  $[0,1]$ )  $y$ -axis position. Size of clusters (*i.e.*, number of elements in the cluster) indicated by extent of cluster on  $y$ -axis. (Extent of the cluster along the  $y$ -axis is proportional to the number of elements in the cluster.) Reversibility and stationarity can be seen through the balanced nature of this plot.

Diffusion models based on the Wright-Fisher model have been extended to the case of infinite species ( $K = \infty$ ). This has been done through the Fleming-Viot process (Fleming and Viot, 1979), and the Moran model (Moran, 1962) which are diffusions defined on the infinite simplex. A construction of the Fleming-Viot process based on finite Wright-Fisher models has also been developed (Donnelly and Kurtz, 1996) and the relation of the WFP to this work is a subject for future research.

## 5.2 Methods

In this section, we will provide a generative process for the WFP model and describe its properties (including exchangeability and reversibility). We will then explain how to model voting data using the WFP. Finally, we will describe the particle Gibbs and probabilistic programming methods we used to do posterior inference on the WFP.

### 5.2.1 Generative process for the Wright-Fisher partition valued diffusion

The CRP (see Chapter 2) with concentration parameter  $\alpha$  can be described by the following sequential scheme, in which the items  $R = \{1, \dots, N\}$  are enumerated in any fixed order:

1. The first item joins a cluster by itself.
2. For each  $i > 1$ , the  $i$ -th item joins a cluster by itself with probability  $\alpha/(i-1+\alpha)$  or for each  $1 \leq j < i$ , joins the cluster containing  $j$  with probability  $1/(i-1+\alpha)$ .

The formulation of the CRP given above is equivalent to the sequential scheme from Chapter 2: the probability of arriving at a partition  $\mathcal{R}$  through the sequential scheme is the same as the probability of drawing  $\mathcal{R}$  from the set of all partitions of  $R$  according to the law given in equation (2.4). Consequently, the CRP is exchangeable: its distribution is invariant to the order in which the items are added to the partition.

We will now present a hierarchical generative process for the WFP based on the CRP distribution on partitions from Chapter 2 and its marginals. We assume that a scaling parameter  $r > 0$  and a concentration parameter  $\alpha$  are fixed.

1. Draw event times  $t_1, t_2, \dots$  from a Poisson process with rate  $rN(N - 1 + \alpha)$  on the positive real line.

2. For each event  $k = 1, 2, \dots$ :

(a) Draw  $E_k$  *i.i.d.* from the set  $\{(i, j) : 1 \leq i \leq j \leq N\}$  with probability:

$$\Pr(E_k = (i, j)) = \begin{cases} 1/(N(N - 1 + \alpha)) & \text{if } i \neq j, \\ \alpha/(N(N - 1 + \alpha)) & \text{if } i = j. \end{cases}$$

3. Draw  $\mathcal{R}_0 \sim \text{CRP}(\{1, \dots, N\}, \alpha)$ .

4. Let  $\mathcal{R}_t$  be constant on the interval  $[0, t_1)$

5. For each event  $k = 1, 2, \dots$ :

(a) Let  $(i_k, j_k) = E_k$ .

(b) Form the induced distribution  $\mathcal{R}_{t_k}^{-i_k}$ .

i. If  $i_k = j_k$  then form  $\mathcal{R}_{t_k}$  by adding  $i_k$  to its own cluster in  $\mathcal{R}_{t_k}^{-i_k}$ .

ii. Otherwise, form  $\mathcal{R}_{t_k}$  by adding  $i_k$  to the cluster in  $\mathcal{R}_{t_k}^{-i_k}$  containing  $j_k$ .

(c) Let  $\mathcal{R}_t$  be constant on the interval  $[t_k, t_{k+1})$ .

Here  $\mathcal{R}_{t-}$  denotes the value  $\lim_{t' \rightarrow t-} \mathcal{R}_{t'}$ . Because the transitions of  $\mathcal{R}_t$  occur on a discrete set with probability 1,  $\mathcal{R}_{t-}$  exists for all  $t > 0$ .

Intuitively, this process transitions by choosing an element  $i \in \mathcal{R}_{t-}$  at a constant rate and removing it to form the induced partition  $\mathcal{R}_t^{-i}$  on  $\{1, \dots, N\} \setminus \{i\}$  and then adding  $i$  back into  $\mathcal{R}_t^{-i}$  according to the probabilities of the sequential CRP scheme in equation (2.4), forming  $\mathcal{R}_t$ . We note that in order to model situations in which more than one item changes clusters between observations occurring at times  $t_1$  and  $t_2$ , multiple events (one for each item) must occur between times  $t_1$  and  $t_2$ .

**Theorem 2.** *The partition valued process  $\mathcal{R}_t$  is a) a Markov jump process, b) exchangeable, c) stationary with CRP marginals, and d) reversible.*



- Proof.* a) The transitions of  $\mathcal{R}_t$  can only occur at the event times  $t_1, t_2, \dots$ . Since these are the points of a Poisson process with bounded rate, with probability 1 the set of event times intersecting any bounded set is finite. Therefore,  $\mathcal{R}_t$  is a Markov jump process.
- b)  $\mathcal{R}_0$  is exchangeable and  $\Pr(E_k = (i, j)) = \Pr(E_k = (\sigma i, \sigma j))$  for all permutations  $\sigma$  and all  $k \geq 1$ . Therefore,  $\mathcal{R}_t$  is exchangeable.
- c) We prove stationarity by using induction on  $k$ . Suppose that  $\mathcal{R}_{t_k}$  is marginally CRP distributed for all  $t \in [0, t_k)$ . By the induction hypothesis,  $\mathcal{R}_{t_k-}$  is marginally CRP distributed. Suppose that the  $t_k - th$  event is given by  $E_k = (i, j)$ . By the projectivity of the CRP, and because  $\mathcal{R}_{t_k-}^{-i} = \mathcal{R}_{t_k}^{-i}$ ,  $\mathcal{R}_{t_k-}^{-i}$  is also marginally CRP distributed on  $\{1, \dots, N\} \setminus \{i\}$ . According to step 2 and step 5 of the generative process above, item  $i$  joins its own cluster in  $\mathcal{R}_{t_k-}^{-i}$  with probability  $\alpha/(N-1+\alpha)$  and joins the cluster containing  $j$  with probability  $1/(N-1+\alpha)$ . This is the conditional probability of the CRP, and so  $\mathcal{R}_{t_k}$  is CRP distributed.
- d) Let  $\mathcal{R}_t$  be restricted to  $t \in [0, T]$ . Define  $\mathcal{R}_t^{\leftarrow} = \mathcal{R}_{T-t}$  and  $E_k^{\leftarrow} = (j, i)$  for each  $E_k = (i, j)$ . By c),  $\mathcal{R}_T$  is marginally CRP distributed. Further,  $E_k$  and  $E_k^{\leftarrow}$  have the same law. Therefore, the law of  $\mathcal{R}_t^{\leftarrow}$  is given by the above enumeration, and this proves reversibility. □

Not all of the events produced by this generative process lead to transitions in the partition valued process  $\mathcal{R}_t$ . If item  $i$  is in its own cluster in  $\mathcal{R}_{t_k-}$  and if  $E_k = (i, i)$ , then  $\mathcal{R}_{t_k-} = \mathcal{R}_t$ . Similarly, if item  $i$  and  $j$  are in cluster  $a \in \mathcal{R}_{t_k-}$  and  $E_k = (i, j)$  for  $i \neq j$ , then  $\mathcal{R}_{t_k-} = \mathcal{R}_t$ . Further, more than one event can lead to the same transition in  $\mathcal{R}_t$ : if  $i$  and  $j$  are both in their own cluster in  $\mathcal{R}_{t_k-}$  then both events  $E_k = (i, j)$  and  $E_k = (j, i)$  would lead to the same partition  $\mathcal{R}_{t_k}$ . (Namely, the partition  $(\mathcal{R}_{t_k-} \setminus \{\{i, j\}\}) \cup \{\{i\}, \{j\}\}$ .) We will denote the transition kernel of the MJP  $\mathcal{R}_t$  by  $\tau(\cdot, \cdot)$ . The values of this kernel are provided in Figure 5.3. This description of  $\tau$  will marginalize these redundant events.

### 5.2.2 Likelihoods for voting data

In the above subsection, we described the WFP as a prior for dynamic partitions. We will now present a model for voting data wherein the WFP is used as a prior on the political similarity for  $N$  members of parliament voting on motions occurring during a session of parliament. Each member of parliament is identified with an integer in  $\{1, \dots, N\}$ . An WFP  $\mathcal{R}_t$  is assumed to be drawn for the duration of the parliament  $[0, T]$  where  $T$  is the time of dissolution of the parliament. For a motion occurring at time  $v \in [0, T]$ , for each cluster  $a \in \mathcal{R}_v$ , all of the members of parliament identified with the elements of  $a$  vote in a similar way. The WFP  $\mathcal{R}_t$  thus describes the changes in political similarity

among the  $N$  members of parliament. Let  $y_{ig} \in \{0, 1\}$  be the vote of the  $i$ -th member of parliament on the  $g$ -th motion (0 means ‘nay’ and 1 means ‘yea’). We will place a beta/Bernoulli prior on the votes, and so the model (which we will refer to as the **WFP for voting** or **WFPV**) is described by the following hierarchical generative process:

1. Draw  $\mathcal{R}_t$  from a **WFP** on  $N$  members of parliament for  $t \in [0, T]$ .
2. Draw the motion times  $v_1, \dots, v_G$  from a Poisson process on  $[0, T]$  with rate  $\beta > 0$ .
3. For each motion  $g = 1, \dots, G$ :
  - (a) For each cluster  $a \in \mathcal{R}_{v_g}$ , draw  $\theta_{v_g a} | m_g \sim \text{Beta}(m_g, m_g)$ .
    - i. For each member of parliament  $i \in a$ , draw her or his vote  $y_{i\ell} | \theta_{\ell} \sim \text{Bernoulli}(\theta_{\ell a})$ .

Here  $m_g$  is a mass parameter describing how polarizing the  $g$ -th motion is (*i.e.*, if  $m_g$  is large members of a bloc will tend to vote together). In our experiments, we will fix  $m_g$  at a value close to the empirical estimate (*i.e.*, the value that gives the empirical voting frequencies the highest probability). This is done to simplify the MCMC inference. We note that in order to take a more Bayesian approach, we could instead place a prior on  $m_g$  as is done in Chapter 3.

### 5.2.3 Relation to time-varying generalized urn schemes

The **WFP** is related to generalized Polya urn schemes for time varying Dirichlet process mixtures (Caron et al., 2007). In Caron et al. (2007), a discrete sequence of partitions of a collection of data items are considered. As in the **WFP**, the partitions are modified by removing some of the items at random at each step of the sequence, and then adding new items according to a CRP. However, unlike the **WFP**, the items in Caron et al. (2007) are not identified between partitions. In Caron et al. (2007), the items removed from the partitions at a given are not added again to the process, and instead new items that have not yet been considered are added to the partition at each step. This difference allows the **WFP** to describe a truly dynamic partitioning: at each point  $t$ , the same  $N$  items are clustered. For distinct points  $t_1, t_2$ , the resulting partitions could differ (the dependence between the partitions at  $t_1$  and  $t_2$  decreases with the scaling parameter  $r$ , with  $r > 0$  implying that the two partitions are equal). The **WFP** formulation is useful for describing situations in which clustering changes in time, for example with changes in the political allegiances of members of parliament.

The **WFP** is also similar to the continuous fragmentation-coagulation process (CFCP) from Teh et al. (2011). Both the **WFP** and the **CFCP** define partition valued Markov processes. The partitions of the **CFCP** transition through the splitting and merging of clusters (according to the fragmentation and coagulation operators defined in Chapter 2), whereas the partitions of the **WFP** transition through the shrinking and growing

of clusters. Generally, the CFCP provides stronger constraints on the transitions of the clusters and so the WFP provides more efficient inference for noisy data or gradually changing data.

#### 5.2.4 Probabilistic programming and inference

It is hard to specify the conditional distribution of the trajectory of a single item  $i$  through the dynamic clustering defined by the WFP. This is because an event  $E_k$  that involves an item  $j$  ‘jumping’ to the cluster containing  $i$  can result in complicated changes to the partition structure at times  $t' > t$ : the changes to the clustering caused by an event occurring at  $t' > t$  for which an item  $j' \neq j$  ‘jumps’ to  $i$  are determined in part by  $E_k(j, i)$ . Consequently, inference based on a conditional forwards-filtering/backwards-sampling (Frühwirth-Schnatter, 1994) algorithm (such as those we derived in Chapters 3 and 4) cannot be derived without including the large space of all possible partitions in the support of the messages. We will therefore use a particle Gibbs (Andrieu et al., 2010) based method for inference. We will implement this through an efficient probabilistic program.

Probabilistic programming languages provide inference for Bayesian models based on their generative processes (Mansinghka et al., 2014; Wood et al., 2014; Goodman et al., 2012; Wingate et al., 2011). By providing general methods for MCMC inference, probabilistic programming languages are similar to frameworks such as BUGS (Thomas et al., 1992) and Infer.NET (Minka et al., 2014). But unlike probabilistic programming languages, BUGS and Infer.NET have strong parametric requirements on the form of the generative process (for example, they cannot provide Dirichlet process priors). On the other hand, by operating on the stack-trace of a program (*i.e.*, the list of machine instructions that specify the output of the program), probabilistic programming languages can provide inference for any model for which a generative process can be implemented in code, regardless of the parametric form.

We will use the **Anglican** probabilistic programming language (Wood et al., 2014), which implements particle MCMC (PMCMC) inference through particle Gibbs (PG). In PG, a particle filter is run with an inexact proposal, targeting the desired posterior. The lineages of the particles and the retained particle sets are then treated as random variables. A Gibbs sampler is run, targeting the distribution induced by the lineages of the particles and the retained particle sets. Viewed as an auxiliary Gibbs method, the restriction of this chain to the particles arriving at the last step of the filter form an MCMC chain targeting the desired posterior.

More formally, imagine we have a target distribution  $p(x_{0:L})$  and a factorization  $p(x_{0:L}) = p(x_{0:0}) \prod_{\ell=1}^L p_{\ell}(x_{\ell:\ell}|x_{0:\ell-1})$  and proposal distributions  $q_0(x_{0:0}), \dots, q_L(x_{L:L})$ . In particle Gibbs,  $S$  particles  $x_{0:0}^{0,0}, \dots, x_{0:0}^{0,S}$  are drawn *i.i.d.* according to the distribution  $q_0(\cdot)$ . Then, the weights  $w_0^s = p(x_{0:0}^{0,s})/q(x_{0:0}^{0,s})$  are computed and normalized

$$(w_0^s \leftarrow w_0^s / \sum_{s'=0}^S w_0^{s'}).$$

Subsequently, for each  $0 < \ell \leq L$ , the indices of the parents of the next generation of particles  $A_{\ell-1}^s$  are drawn from the distribution  $A_{\ell-1}^s \sim \sum_{s'=0}^S w_{\ell-1}^{s'} \delta_t(\cdot)$ . (Here  $\delta_t(\cdot)$  is the Dirac delta function centered at  $s$ .) Given these indices, the next generation of particles is described as follows:  $X_{\ell:\ell}^{\ell,s} \sim q_\ell(\cdot)$ ,  $X_{0:\ell-1}^{\ell,s} \leftarrow X_{0:\ell-1}^{\ell-1, A_{\ell-1}^s}$  and the weights are computed and normalized as follows:  $w_\ell^s = p(X_{0:\ell}^{\ell,s} | X_{0:\ell-1}^{\ell,s}) / q(X_{\ell:\ell}^{\ell,s})$ ,  $w_\ell^s \leftarrow w_\ell^s / \sum_{s'=0}^S w_\ell^{s'}$ .

After arriving at the particles  $X_{0:L}^s$  and the weights  $w_L^s$ , the distribution  $\sum_0^S w_L^s \delta_{X_L}(\cdot)$  is a sequential Monte Carlo (SMC) approximation for  $X_{0:L}$ . In PG, the weights and ancestor indicators ( $w_\ell^s$  and  $A_{\ell-1}^s$ ) are now treated as latent variables and resampled using Gibbs updates. PG confers benefits over SMC such as faster mixing and reduced estimator variance (Andrieu et al., 2010).

In the `Anglican` probabilistic programming language, the distributions  $p_\ell$  and  $q_\ell$ , and the Gibbs updates for  $A_\ell$  are automatically formed given a generative process such as the generative process for the WFP in section 5.2.1. The proposals  $q_\ell$  are given by the prior distribution. While this update is not efficient on big data, it is reasonable for the sizes of data used in this Chapter. We will assume that the concentration parameter  $\alpha$  is fixed.

In our application,  $X_{0:0} = \mathcal{R}_0$  is the partition  $\mathcal{R}_0$  at time zero (the third step of the generative process in section 5.2.1) and  $X_{\ell:\ell} = (E_\ell(i_\ell, j_\ell), t_\ell)$  describes the draws  $t_k$  and  $E_k$  ( $\ell = k$ ) from the first and second step, respectively, of the generative process in section 5.2.1. The partition  $\mathcal{R}_t$  is thus induced by the particle  $X_{0:\ell}$  for all values of  $t$  such that  $0 \leq t \leq T' = \sum_{\ell'=1}^\ell t_{\ell'}$ . In particular, for  $t = 0$ ,  $\mathcal{R}_0 = X_{0:0}$ . For  $0 < t \leq T'$ , let  $\ell = \min_{\ell'} \{t_1 + \dots + t_{\ell'} \geq t\}$ . Then, form  $\mathcal{R}_t$  by taking  $\mathcal{R}_0$  and performing the ‘copying’ operations  $E_1(i_1, j_1), \dots, E_{\ell'}(i_{\ell'}, j_{\ell'})$ .

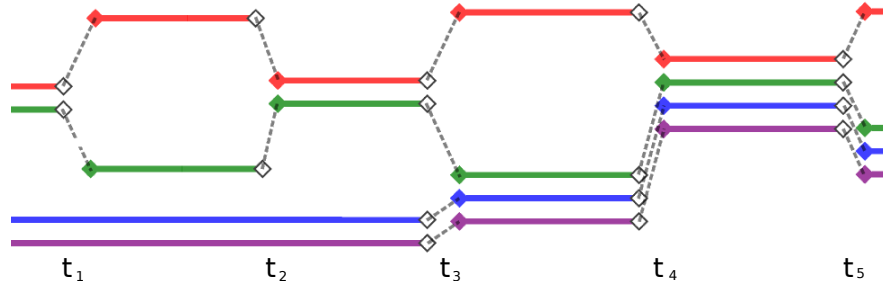
The proposals  $q_\ell(\cdot)$  are formed from the prior distribution (*i.e.*, proposals from the prior):  $q_0(X_{0:0})$  is the density of the CRP partition with concentration  $\alpha$ :  $q_0(X_{0:0}) = \text{CRP}(X_{0:0} | \alpha)$ . The proposals  $q_\ell(X_{\ell:\ell})$  for  $\ell > 0$  are such that the probability density of  $(E_\ell(i_\ell, j_\ell), t_\ell)$  is:

$$q_\ell(X_{\ell:\ell}) = \text{Exp}(t_\ell | rN(N-1+\alpha)) \cdot \begin{cases} 1/(N(N-1+\alpha)) & \text{if } i_\ell \neq j_\ell, \\ \alpha/(N(N-1+\alpha)) & \text{if } i_\ell = j_\ell. \end{cases}$$

Here  $\text{Exp}(\cdot | \lambda)$  is the density of the exponential distribution with rate  $\lambda$ . The probabilities for the weight computation incorporate the joint distribution of the dynamic partition and the observed voting data, and are given as follows:

$$p_0(X_{0:0}) = q(X_{0:0}), \tag{5.1}$$

$$p_\ell(X_{0:\ell} | X_{0:\ell-1}) = q(X_{\ell:\ell}) \cdot \prod_{g: \sum_{\ell'=0}^{\ell-1} t_{\ell'} < v_g \leq \sum_{\ell'=0}^\ell t_{\ell'}} \Lambda(y:g | \mathcal{R}_{v_g}, m_g) \tag{5.2}$$



**Figure 5.2:** Probability for partitions sampled according to the partition valued process  $\mathcal{R}_t$ . Events occur at times  $t_1, \dots, t_5$ . The probabilities for the events and holding times are given by the cases shown in Figure 5.3

Here, as in the generative process for the voting data in section 5.2.2,  $v_g$  is the time of the  $g$ -th motion and  $y_{:g}$  is an  $N$ -dimensional 0/1-vector describing the votes of the  $N$  MPs for the  $g$ -th motion. The partition  $\mathcal{R}_{v_g}$  is the partition induced by the particle  $X_{0:\ell}$  at the time of the  $g$ -th vote ( $v_g$ ). The product in (5.2) includes all votes that occur between the time of the  $\ell$ -th and  $\ell + 1$ -th change in the partition structure of the MPs. The likelihood  $\Lambda(y_{:g}|\mathcal{R}_{v_g}, m_g)$  describes the probability of observing the votes of the MPs for the  $g$ -th motion, given the partition structure at the time of that motion:

$$\Lambda(y_{:g}|\mathcal{R}_{v_g}, m_g) = \prod_{a \in \mathcal{R}_{v_g}} \Gamma(m_g + n_{1ga})\Gamma(m_g + n_{0ga})/\Gamma(2m_g + n_{1ga} + n_{0ga}) \quad (5.3)$$

Here  $n_{1ga}$  and  $n_{0ga}$  are the numbers of MPs in block  $a$  of the partition  $\mathcal{R}_t$  that vote ‘yea’ or respectively ‘nay’ for the  $g$ -th motion. This likelihood integrates out the vote-emission probabilities  $\theta_{g\cdot}$ . For more detail on probabilistic programming in the **Anglican** language, and inference in PG, we refer to [Wood et al. \(2014\)](#) and [Andrieu et al. \(2010\)](#).

### 5.2.5 Describing $\mathcal{R}_t$ as a partition valued process

The hierarchical generative construction in section 5.2.1 describes  $\mathcal{R}_t$  through a two step process. First, transition times and the events  $E_k$  are drawn. Second,  $\mathcal{R}_0$  is drawn and conditioned on  $E_k$  and  $\mathcal{R}_0$ , the partitions  $\mathcal{R}_t$  are determined for  $t > 0$ . As noted in section 5.2.1, the events  $E_k$  could be redundant. For every pair of distinct partitions  $\mathcal{R}_-$ , and  $\mathcal{R}$ , we will now consider all events  $E$  that could lead to a transition from  $\mathcal{R}_{t-} = \mathcal{R}_-$  to  $\mathcal{R}_t = \mathcal{R}$ . We sum the rates of those events to find the transition rate  $\tau(\mathcal{R}_-, \mathcal{R})$ . In this way,  $\mathcal{R}_t$  can be described as an MJP with transition rate matrix  $\tau(\cdot, \cdot)$ . (The columns and rows of this matrix correspond to each possible partition of  $\{1, \dots, N\}$ .) There are 5 possible cases for the partitions  $\mathcal{R}_-, \mathcal{R}$ . The function  $\tau$  assigns zero rate to all pairs of distinct partitions  $\mathcal{R}_-, \mathcal{R}$  that are not covered by these cases (*i.e.*, for such pairs there is no single event  $E$  from the generative process that can realize that transition). The cases are listed in Figure 5.3, and a diagram showing an

example of the WFP model viewed as partition valued transitions is given in Figure 5.2.

The cases and rates in Figure 5.3 are found by considering the rates of the events  $E_1, E_2, \dots$  from the hierarchical generative process for  $\mathcal{R}_t$  from section 5.2.1. We will verify case 1 of this derivation. Suppose that  $\{i, j\} \in \mathcal{R}_t$  ( $i \neq j$ ). There are two possibilities for  $E_k$  that could both lead to the transition described in case 1:  $E_k = (i, i)$  and  $E_k = (j, j)$ . Therefore, the rate  $\tau(\mathcal{R}_{-t}, \mathcal{R}_t)$  for the pairs of partitions described by case 1 is  $2r\alpha$ . The 4 other cases can be verified in a similar way.

If  $\mathcal{R}_t = \mathcal{R}$ , then the total rate of transition from  $\mathcal{R}$  is found by summing the rates for all the cases:

$$\tau(\mathcal{R}_{-}, \cdot) = r\alpha(N - \#\{i : \{i\} \in \mathcal{R}_{-}\}) + 2r \sum_{\{a,b\} \subseteq \mathcal{R}_{-}} \#a\#b. \quad (5.4)$$

Here,  $N - \#\{i : \{i\} \in \mathcal{R}_{-}\}$  is the number of singleton clusters in  $\mathcal{R}_{-}$  and the sum  $\sum_{\{a,b\} \subseteq \mathcal{R}_{-}}$  is over all distinct (unordered) pairs of clusters  $a, b$  in  $\mathcal{R}_{-}$ .

## 5.3 Experiments

### 5.3.1 Experiment I: bloc discovery

We conducted an experiment on voting data from the 38th parliament of the Canadian House of Commons<sup>1</sup>. This parliament lasted from October 2004 until November 2005 and involved 307 members of parliament. A total of 190 motions were voted on by the members of parliament. In May 2005 (around the 34th week of the parliament) Belinda Stronach, the member of parliament from the Newmarket–Aurora riding, left the Conservative party and joined the Liberal party. We simulated the posterior distribution of the WFPV process conditioned on the voting data. We examine the cluster assignment of Belinda Stronach over the duration of the process.

### 5.3.2 Experiment II: vote prediction

We considered votes for which there was more than 20% disagreement among members of parliament and split that voting data evenly into a testing set and a training set (a missing-at-random condition). We filtered votes with less than 20% disagreement as these votes were often on procedural motions which did not contain much information about party affiliation. We simulated the WFPV posterior conditioned on the training set and looked at the accuracy of the WFPV's predictive likelihood on testing set. We simulated the WFPV process conditioned on the votes in the training set, and predicted the held out votes in the testing set using the WFPV likelihood.

<sup>1</sup>Retrieved from <http://www.parl.gc.ca/HouseChamberBusiness/> on June 1st, 2014.

Case 1: Let  $a = \{i, j\} \in \mathcal{R}-$ . If  $\mathcal{R}$  is formed from  $\mathcal{R}-$  by removing  $a$  and from  $\mathcal{R}-$  and adding the singleton clusters  $\{i\}$  and  $\{j\}$  then  $\tau_n(\mathcal{R}-, \mathcal{R}) = 2r\alpha$ .

Case 2: Let  $a \neq b \in \mathcal{R}-$  be distinct singleton clusters  $a = \{i\}, b = \{j\}$ . If  $\mathcal{R}$  is formed from  $\mathcal{R}-$  by removing  $a, b$  from  $\mathcal{R}-$  and adding  $a \cup b = \{i, j\}$  then  $\tau_n(\mathcal{R}-, \mathcal{R}) = 2r$ .

Case 3: Let  $a, b \in \mathcal{R}-$  be such that  $\#a > 1$ . For each  $i \in a$ , if  $\mathcal{R}$  is formed from  $\mathcal{R}-$  by removing  $a$  and  $b$  from  $\mathcal{R}-$  and adding  $a \setminus \{i\}$  and  $b \cup \{i\}$  then  $\tau_n(\mathcal{R}-, \mathcal{R}) = r\#b$ .

Case 4: Let  $a, b \in \mathcal{R}-$  be such that  $a$  is the singleton cluster  $a = \{i\}$ , and  $\#b > 1$ . If  $\mathcal{R}$  is formed from  $\mathcal{R}-$  by removing  $a, b$  from  $\mathcal{R}-$  and adding  $a \cup b$  then  $\tau_n(\mathcal{R}-, \mathcal{R}) = r\#b$ .

Case 5: Let  $a \in \mathcal{R}-$  be such that  $\#a > 2$ . For each  $i \in a$ , if  $\mathcal{R}$  is formed from  $\mathcal{R}-$  by removing  $a$  from  $\mathcal{R}-$  and adding clusters  $a - \{i\}$  and  $\{i\}$  then  $\tau_n(\mathcal{R}-, \mathcal{R}) = r\alpha$ .

**Figure 5.3:** Cases for transitions arising from the description of  $\mathcal{R}_t$  as a partition valued-MJP with transition kernel  $\tau(\cdot, \cdot)$ .

We compared the accuracy of the predictions of the WFPV with a baseline given by probabilistic matrix factorization (Salakhutdinov and Mnih, 2007), a popular model in collaborate filtering. The probabilistic matrix factorization model is as follows: each member of parliament  $i$ , is associated with a  $D \times 1$  dimensional latent random vector  $u_i$ . Each motion is also associated with a  $D \times 1$  dimensional latent random vector  $v_j$ . The probability that the  $i$ -th member of parliament votes ‘yea’ for the  $j$ -th motion is given

by the inner product  $u_i^T v_j$ , passed through a link function. A Bayesian prior is placed on  $u_i$  and  $v_j$ : in the prior, each of the  $u_i$  vectors are drawn *iid* from a Gaussian with mean 0 and variance  $\lambda_u$  and each of the  $v_j$  vectors are drawn *iid* from a Gaussian with mean 0 and variance  $\lambda_v$ . We conducted MAP inference for this probabilistic matrix factorization using alternating least squares (Zhou et al., 2008). The implementation we used was provided by the GraphLab software package (Wu et al., 2011).

In both experiments, we performed inference using particle Gibbs. We used 200 particles and 100 sweeps per iteration. The parameter settings we used for the WFPV likelihood were  $r = 20.0$  (in units of weeks<sup>-1</sup>),  $\alpha = 1.5$  and  $m_\ell = 0.005$ . These settings were chosen to match our intuition about how large the political caucuses found by the WFPV should be, and the empirical frequency of agreement found within the votes of a political caucus.

## 5.4 Results

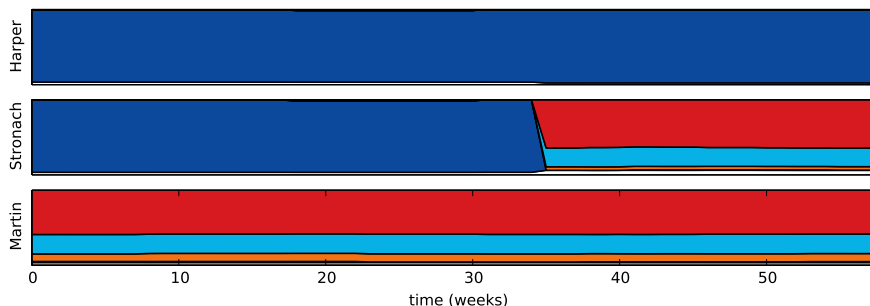
In Figure 5.4, we show the results of the first experiment. The bands indicate the clusters found by the WFPV for Belinda Stronach, Stephen Harper (the leader of the Conservative party) and Paul Martin (the leader of the Liberal party and the Prime Minister of Canada at the time of the 38th parliament). From this figure, we can see that the WFPV discovers the changed party allegiance of Belinda Stronach around the 34th week of parliament. In contrast, the cluster containing Stephen Harper remains exclusively Conservative throughout the course of the process.

To better visualize the voting data and understand the WFPV, in Figure 5.5 we show the proportion of each party among the members of parliament that voted in the same way as Belinda Stronach. In this visualization, for each motion, we examine all MPs that voted in the same way as Belinda Stronach. The  $y$ -axis indicates which week the motion occurred in. The  $x$ -axis indicates the ratio of MPs from each party that voted the same way as Belinda Stronach. For example, in the first motions of week 0, all of the MPs that voted the same way as Belinda Stronach were Liberal. In contrast, in the motions in the last week, almost all of the MPs that voted the same way as Belinda Stronach were Conservative. In this visualization (which depends on knowing the party memberships of all of the members), we can clearly see Belinda Stronach changing party allegiance. The white strip indicates a period where Belinda Stronach did not vote on any motions. We note that the WFPV was not confounded by this effect: the time at which most MPs voting the same way as Belinda Stronach switches from Liberal to Conservative roughly corresponds to the time at which Belinda Stronach changes clusters in 5.4. Further, to the right of this figure we see that the Bloc Québécois, the NDP and the Liberals all voted together. This explains the mixed party allegiances of the members of Paul Martin’s cluster in Figure 5.4.



| Method       | CF5  | PARTY | CF1  | WFPV | baseline |
|--------------|------|-------|------|------|----------|
| Accuracy (%) | 98.0 | 96.7  | 90.6 | 81.9 | 62.5     |

Table 5.1: Percent correct for vote predictions. CF5 and CF1 indicate collaborative filtering with 5 and 1 dimensions, respectively. PARTY indicates predicting votes based on party allegiances of training set data.



**Figure 5.4:** Composition of the clusters found by WFP model. Each band gives proportion of each party among the allegiances of all members of the cluster containing the member of interest (indicated by the  $y$ -label) over the course of the parliament. From top to bottom, members of interest are Stephen Harper, Belinda Stronach and Paul Martin. Colors indicate the four main political parties (blue for Conservatives, red for Liberals, orange for NDP/NPD and light blue for Bloc Québécois).

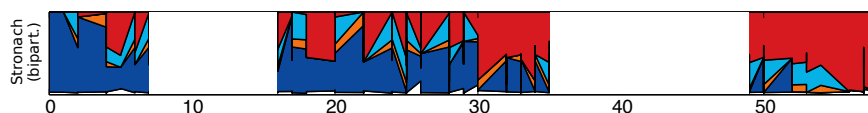
We show the accuracies for vote prediction in the second experiment in Table 5.1. The accuracy of the WFPV model was 81.9%, which was below the accuracies of collaborative filtering (CF) with 5 latent components. The baseline accuracy found by predicting the most common vote in the training set for each motion was 62.5%.

We found that a collaborative filtering (Salakhutdinov and Mnih, 2007) provided the best accuracy, which is unsurprising consider the success of spatial models in predicting role call votes (Poole and Rosenthal, 1985). By examining Figure 5.4 we see that the WFPV finds large-scale blocks which cross party lines. (The Liberals, NDP/NPD and Bloc Québécois were not a coalition, but they did vote similarly on many motions.) Based on the data in Figure 5.5 it is clear that the accuracies of the WFPV could be improved if it were to find exact party lines. (The clusters found by the WFPV cannot model the jagged structures in the real data presented in Figure 5.5.)

It is possible that hyperpriors on the concentrations, rates and emission model would improve the accuracy. To that end, particle Gibbs ancestral sampling would be appropriate and would be an interesting direction of future work (Lindsten et al., 2012).

## 5.5 Discussion

The accuracy of the WFP model in the vote imputation task was much lower than that of collaborative filtering (CF) based methods. Further, we found that the clustering found by the WFP model was not sensitive to the hyperparameters of the model. In many



**Figure 5.5:** Band for the bipartition found by clustering members of parliament into two clusters: those who vote in the same way as Belinda Stronach and those who vote in the opposite way. White indicates periods for which Belinda Stronach did not cast votes. Columns and colors are the same as those defined in previous figure.

cases, the PG produced degenerate particles (*i.e.*, the retained set often contained only one particle). We could possibly solve these problems by using ancestral resampling for particle Gibbs. Another possible solution could be the use of Metropolis Hastings with a proposal induced by a Markov assumption for the conditional cluster assignment of a sequence.

The possible non-mixing of the WFP notwithstanding, it is unlikely that any HMM model will do better in vote prediction than CF. But CF does not depend on temporal ordering and combines information from past and future times to predict a vote at a specific time. On the other hand, the only information from past and future times that an HMM uses is conveyed through the HMM state, which is relatively low-dimensional. So, while WFPV did not display significant imputation accuracy, that does not mean that it is a bad model for these time series data. If a clustering were defined based on the CF, it would only capture global effects.

For example, if an MP votes against the party line consistently 10% of the time, the WFPV might still place it in the party’s cluster. But, if an MP votes against the party line only for the last 10% of the duration of the process, the WFPV would be more inclined to reflect this in the clustering (*i.e.*, the MP would switch clusters). By ignoring temporal ordering, the CF makes no distinction at all between these two cases. Thus, the WFPV confers additional insight by capturing this temporal structure. In future work, we will consider combining the WFPV with a CF model, in a similar way to how the CRP is combined with a CF in [Sutskever et al. \(2009\)](#).

Inference based on Markov approximations of the conditional sequence are also possible. Leaving probabilistic programming, we could instead derive an MH update for the state assignment of a sequence in which the proposal distribution is defined using a Markov approximation (this MCMCM kernel would target the true posterior distribution).

## 5.6 Conclusions

We have presented inference for a new partition valued Markov jump process (the WFP) based on a countable version of the Wright-Fisher diffusion model. It is exchangeable, reversible and its marginals are given by the CRP. The WFP does not have Markovian marginals, and therefore cannot be approached by inference based on dynamic program-

ming. Instead, we used particle Gibbs to simulate the **WFP** posterior. We implemented the particle Gibbs using the Anglican probabilistic programming language.

We attached a likelihood to the **WFP** model to describe voting behavior of members of parliament (yielding the **WFPV**). We applied the **WFPV** to data from the 38th Canadian parliament and found that the posterior clusters found could be used to detect the change in party allegiances of one of the members of parliament. This was done without using any covariates associated with the members of parliament (such as their parties) or the votes (such as the texts of bills).

## Chapter 6

# Conclusions and future work

### 6.1 Conclusions

Bayesian nonparametric statistics were first developed in the late 70s to provide prior distributions which have both arbitrarily large support and also tractable posteriors. Recently, the development of the nonparametric hierarchical Dirichlet process (Teh et al., 2006) has allowed a wide variety of classical statistical tools (such as HMMs) to make use of Bayesian nonparametric priors. This has led to a resurgence of interest in Bayesian nonparametric models, and much insight into the latent structure of the data to which these models have been applied. Methodologically, the models presented in this thesis are some of the most sophisticated applications of Bayesian nonparametrics to genetics that has been derived to date. Further, we have made available the code for the BNPPHASE model, and have provided a detailed description of these methods which are of interest to the broader bioinformatics and population genetics community.

We have presented three new Bayesian nonparametric clustering models (BNPPHASE, DFCP and WFP). The BNPPHASE and DFCP models are motivated by the genetic process and have similarities to many popular models currently used in statistical genetics. We explored these models through applications to various sources of data such as simulated bottlenecks, X chromosomes from The Thousand Genomes Project, SNP data from the HapMap Project and also SNP data from the SeattleSNPs project. We showed that genotype imputation accuracy for our nonparametric models was often better than that of the related parametric models, and we were able to interpret the latent variables of the BNPPHASE model as founders in population bottlenecks or as rescaled versions of the time to most recent common ancestor. To illustrate the versatility of Bayesian nonparametric models, we also applied the WFP model to predict votes and to uncover political blocs in data from the Canadian House of Parliament.

We also discussed theoretical properties of these models: we derived expected values for the lengths of haplotypes under the DFCP model and we computed the conditional

distributions of fragmentation and coagulation operators.

## 6.2 Future work

The experiments and analysis discussed in this thesis present several avenues for future work. We derived equations for phasing using the DFCEP, but these equations can be simplified using a study/reference paradigm. The reference panels could be used as a source of phased data to build a ‘scaffold’ of haplotypes. This could be done by running the DFCEP model on the reference paradigm, and choosing a representative sample from the MCMC. Unphased data could then be registered to this scaffold by assuming that the unphased diploid sequences are independent conditioned on the scaffold, and also supposing that the diploid sequences never form new haplotypes, and instead must always join the haplotypes that already exist in the scaffold. The resulting messages would be quite simple, and the phasing of all the diploid sequences could be done in parallel.

For the WFP model, we found that the imputation accuracy was lower than that of linear methods such as collaborate filtering. However, collaborative filtering cannot capture changes to the block structure of sequences over time (or over chromosome location). To that end, we plan to examine a mixture between a collaborative filter and the WFP model in order to model changes in block structure and also produce high-accuracy predictions (as was done in [Sutskever et al. 2009](#) for static clustering).

# Bibliography

- A. Ahmed and E. P. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process. In *Proceedings of the Society for Industrial and Applied Mathematics International Conference on Data Mining*, volume 8, 2008. (page 37)
- E. M. Airoldi, D. M. Blei, E. P. Xing, and S. E. Fienberg. Mixed membership stochastic block models for relational data with application to protein-protein interactions. In *Proceedings of the International Biometrics Society Annual Meeting*, 2006. (page 12)
- D. Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII-1983*. Springer, Berlin, 1985. (page 39)
- C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010. (pages 109, 115, 116, and 117)
- M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. In *Advances in Neural Information Processing Systems*, volume 14, 2002. (pages 30, 37, and 47)
- D. Blackwell and J. B. MacQueen. Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1(2):353–5, 1973. (page 40)
- D. Blei and J. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007. (page 13)
- D. M. Blei and P. I. Frazier. Distance dependent Chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–88, 2011. (page 37)
- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 14, 2002. (page 35)
- B. L. Browning and S. R. Browning. A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American Journal of Human Genetics*, 84(2):210–23, 2009. (pages 14, 28, and 104)
- S. R. Browning. Multilocus association mapping using variable-length markov chains. *American Journal of Human Genetics*, 78(6):903–13, 2006. (page 29)

- S. R. Browning and B. R. Browning. Haplotype phasing: existing methods and new developments. *Nature Reviews Genetics*, 12(10):702–14, 2011. (page 17)
- F. Caron, M. Davy, and A. Doucet. Generalized Polya urn for time-varying Dirichlet process mixtures. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 23, 2007. (page 114)
- G. Celeux. Bayesian inference for mixture: the label switching problem. *Compstat*, 998:227–32, 1998. (page 30)
- M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and R. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–32, 2001. (pages 14, 33, and 81)
- D. A. Dawson and K. J. Hochberg. Wandering random measures in the Fleming-Viot model. *Annals of Probability*, 10(3):554–80, 1982. (page 110)
- M. De Iorio, L. T. Elliott, S. Favaro, K. Adhikari, and Y. W. Teh. Modeling population structure under heirarchical Dirichlet processes. arXiv preprint arXiv:1503.08278v1, 2015. (page 28)
- O. Delaneau, J. Marchini, and J. F. Zagury. A linear complexity phasing method for thousands of genomes. *Nature Methods*, 9(2):179–81, 2012. (pages 27, 29, and 83)
- O. Delaneau, J. F. Zagury, and J. Marchini. Improved whole chromosome phasing for disease and population genetic studies. *Nature Methods*, 10(1):5–6, 2013. (pages 27, 29, and 83)
- P. Donnelly and T. G. Kurtz. A countable representation of the Fleming-Viot measure-valued diffusion. *The Annals of Probability*, 24(2):698–742, 1996. (pages 109 and 111)
- D. Dunson. Bayesian dynamic modeling of latent trait distributions. *Biostatistics*, 7(4):551–68, 2006. (page 36)
- L. T. Elliott and Y. W. Teh. Scalable imputation of genetic data with a discrete fragmentation-coagulation process. In *Advances in Neural Information Processing Systems*, volume 24, 2012. (pages 14, 29, 42, 81, and 90)
- L. T. Elliott and Y. W. Teh. A sticky HDP-HMM for genetic imputation and inferring time to common ancestors. Paper ID 1234, 2015. Under review in *NIPS 27*. (pages 14 and 46)
- W. J. Ewens. The sampling theory of selectively neutral alleles. *Theoretical Population Biology*, 3(1):87–112, 1972. (pages 13 and 39)
- D. Falush, M. Stephens, and J. K. Pritchard. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–87, 2003. (page 27)

- P. Fearnhead and P. Donnelly. Approximate likelihood methods for estimating local recombination rates. *Journal of the Royal Statistical Society B*, 64(4):657–80, 2002. (page 25)
- J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood. *Journal of Molecular Evolution*, 17(6):368–76, 1981. (page 20)
- T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–30, 1973. (pages 30, 35, and 38)
- R. A. Fisher. *The Genetical Theory of Natural Selection*. Clarendon Press, 1930. (page 110)
- R. A. Fisher, A. Steven Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *The Journal of Animal Ecology*, 12(1):42–58, 1943. (page 39)
- W. H. Fleming and M. Viot. Some measure-valued Markov processes in population genetics theory. *Indiana University Mathematics Journal*, 28(5):817–43, 1979. (page 111)
- E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, 5(2):1020–56, 2011. (pages 26, 32, and 67)
- A. Friggeri. Agreement groups in the U.S. Senate. <http://friggeri.net/research/senate>, 2012. Accessed: Summer 2014. (page 13)
- S. Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202, 1994. (pages 52, 60, and 115)
- J. Gasthaus and Y. W. Teh. Improvements to the sequence memoizer. In *Advances in Neural Information Processing Systems*, volume 22, 2010. (page 43)
- A. Gelman and X.-L. Meng. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*. John Wiley & Sons, 2004. (page 30)
- S. Ghosal. Dirichlet process, related priors and posterior asymptotics. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *Bayesian Nonparametrics*. Cambridge University Press, 2010. (pages 30 and 38)
- N. Goodman, V. Mansinghka, and D. Roy. Church: a language for generative models. arXiv preprint arXiv:1206.3255, 2012. (page 115)
- D. Görür and Y. W. Teh. An efficient sequential Monte-Carlo algorithm for coalescent clustering. In *Advances in Neural Information Processing Systems*, volume 21, 2009. (page 21)



- J. Hein, M. H. Schierup, and C. Wiuf. *Gene Genealogies, Variation and Evolution*. Oxford University Press, 2005. (pages 12, 15, 21, and 48)
- N. L. Hjort, C. C. Holmes, P. Müller, and S. G. Walker. *Bayesian Nonparametrics*. Cambridge University Press, 2010. (pages 13, 30, and 31)
- Q. Ho, A. P. Parikh, L. Song, and E. P. Xing. Infinite hierarchical MMSB model for nested communities/groups in social networks. *arXiv preprint arXiv:1010.1868*, 2010. (page 36)
- A. R. Hoelzel, J. Halley, S. J. O'Brien, C. Campagna, T. Arnborn, B. Le Boeuf, K. Ralls, and G. A. Dover. Elephant seal genetic variation and the use of simulation models to investigate historical population bottlenecks. *Journal of Heredity*, 84(6):443–9, 1993. (page 46)
- B. N. Howie, P. Donnelly, and J. Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLOS Genetics*, 6(6), 2009. (pages 14 and 27)
- R. R. Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23(2):183 – 201, 1983. (page 19)
- R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–8, 2002. (pages 33, 85, 104, and 105)
- J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–5, 2001. (page 21)
- A. Jasra, C. C. Holmes, and D. A. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005. (pages 30 and 46)
- A. J. Jeffreys, L. Kauppi, and R. Neumann. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nature Genetics*, 29(2):217–22, 2001. (page 33)
- R. Karchin, M. Diekhans, L. Kelly, D. J. Thomas, U. Pieper, N. Eswar, D. Haussler, and A. Sali. LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources. *Bioinformatics*, 21(12):2814–20, 2005. (page 29)
- C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, 2006. (page 13)
- M. Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, 1983. (page 21)

- M. Kimura and J. Crow. The number of alleles that can be maintained in a finite population. *Genetics*, 49(4):725–38, 1964. (page 20)
- J. F. C. Kingman. The coalescent. *Stochastic Processes and their Applications*, 13(3):235–48, 1982. (page 19)
- K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *Proceedings of the International Conference on Artificial Intelligence*, volume 20, 2007. (page 54)
- E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer, 1998. (page 97)
- H. Li and R. Durbin. Inference of human population history from individual whole-genome sequences. *Nature*, 475(7357):493–6, 2011. (pages 22, 49, and 70)
- N. Li and M. Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–33, 2003. (pages 24, 25, 26, and 71)
- F. Lindsten, M. I. Jordan, and T. B. Schön. Ancestor sampling for particle Gibbs. In *Advances in Neural Information Processing Systems*, volume 25, 2012. (page 121)
- S. MacEachern. Dependent nonparametric processes. In *Proceedings of the Section on Bayesian Statistical Science*. American Statistical Association, 1999. (pages 13 and 36)
- V. Mansinghka, D. Selsam, and Y. Perov. Venture: a higher-order probabilistic programming platform with programmable inference. arXiv preprint arXiv:1404.0099, 2014. (page 115)
- J. Marchini and B. Howie. Genotype imputation for genome-wide association studies. *Nature Reviews Genetics*, 11(7):499–511, 2010. (page 17)
- J. Marchini, B. Howie, S. Myers, G. A. T. McVean, and P. Donnelly. A new multipoint method for genome-wide association studies by imputation of genotypes. *Nature Genetics*, 39(7):906–13, 2007. (pages 14, 27, 29, and 83)
- G. A. T. McVean and N. J. Cardin. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1459):1387–93, 2005. (pages 22 and 83)
- H. Michael, E. Fox, and E. Sudderth. Effective split-merge Monte Carlo methods for nonparametric models of sequential data. In *Neural Information Processing Systems*, volume 25, 2012. (page 67)
- K. Miller, T. Griffiths, and M. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, volume 22, 2009. (page 36)

- T. Minka, J. M. Winn, J. Guiver, S. Webster, Y. Zaykov, B. Yangel, A. Spengler, and J. Bronskill. Infer.NET 2.6, 2014. Microsoft Research Cambridge. <http://research.microsoft.com/infernet>. (page 115)
- P. A. P. Moran. *The Statistical Processes of Evolutionary Theory*. Oxford: Clarendon Press, 1962. (page 111)
- National Heart, Lung, and Blood Institute Program for Genomic Applications. SeattleSNPs. <http://pga.gs.washington.edu>, 2011. Accessed: Summer 2011. (page 105)
- R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3):705–67, 2003. (pages 52, 66, 86, and 92)
- P. Orbanz and Y. W. Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*. Springer, 2010. (page 13)
- B. Paige and F. Wood. A compilation target for probabilistic programming languages. In *Proceedings of the International Conference on Machine Learning*, volume 31, 2014. (page 109)
- K. Palla, D. A. Knowles, and Z. Ghahramani. A reversible infinite hidden Markov model using normalised random measures. In *Proceedings of the International Conference on Machine Learning*, volume 31, 2014. (page 37)
- A. C. Payne, M. Andregg, K. Kemmish, M. Hamalainen, C. Bowell, A. Bleloch, N. Klejwa, W. Lehrach, K. Schatz, H. Stark, A. Marblestone, G. Church, C. S. Own, and W. Andregg. Molecular threading: Mechanical extraction, stretching and placement of DNA molecules from a liquid-air interface. *PLOS ONE*, 8(7), 2013. (page 15)
- J. Pitman. Coalescents with multiple collisions. *Annals of Probability*, 27(4):1870–902, 1999. (pages 42 and 43)
- J. Pitman. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, University of California at Berkeley, 2002. Lecture notes for St. Flour Summer School. (page 42)
- J. Pitman. *Combinatorial stochastic processes*. Springer-Verlag, 2006. (pages 33, 42, 66, 99, and 109)
- J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997. (page 43)
- K. T. Poole and H. Rosenthal. A spatial model for legislative roll call analysis. *American Journal of Political Science*, 29(2):357–84, 1985. (page 121)
- J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–59, 2000. (page 27)

- V. Rao and Y. W. Teh. Spatial normalized gamma processes. In *Advances in Neural Information Processing Systems*, volume 22, 2009. (pages 36 and 107)
- V. Rao and Y. W. Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2011. (pages 92, 105, and 107)
- M. D. Rasmussen, M. J. Hubsisz, H. Gronau, and A. Siepel. Genome-wide inference of ancestral recombination graphs. *arXiv preprint arXiv:1306.5110*, 2013. (page 21)
- L. B. S. Myers, C. Freeman, G. A. T. McVean, and P. Donnelly. A fine-scale map of recombination rates and hotspots across the human genome. *Science*, 310(5746):321–4, 2005. (pages 50 and 54)
- R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2007. (pages 14, 119, and 121)
- P. Scheet and M. Stephens. A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *The American Journal of Human Genetics*, 78(4):629–644, 2006. (pages 14, 27, 28, 29, 32, 34, 46, 50, 68, 70, 97, 104, and 107)
- K.-A. Sohn and E. P. Xing. Hidden Markov Dirichlet process: modeling genetic inference in open ancestral space. *Bayesian Analysis*, 2(3), 2007. (page 12)
- I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems*, volume 22, 2009. (pages 122 and 125)
- Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010. (page 13)
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–81, 2006. (pages 12, 30, 31, 37, 41, and 124)
- Y. W. Teh, C. Blundell, and L. T. Elliott. Modelling genetic variations using fragmentation-coagulation processes. In *Advances in Neural Information Processing Systems*, volume 23, 2011. (pages 30, 33, 81, 85, 104, 105, 107, and 114)
- The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–73, 2010. (pages 14, 15, 16, 33, 47, 70, 77, 82, and 104)
- The International HapMap Consortium. The international HapMap project. *Nature*, 426(6968):789–96, 2003. (pages 14, 29, 32, and 33)

- The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145): 661–78, 2007. (page 17)
- A. Thomas, D. J. Spiegelhalter, and W. R. Gilks. A program to perform Bayesian analysis using Gibbs sampling. *Bayesian Statistics*, 4(9):837–42, 1992. (page 115)
- L. Toscano. Numeri di Stirling generalizzati operatori differenziali e polinomi ipergeometrici. *Commentationes Pontificia Academica Scientarum*, 3:721–57, 1939. (page 99)
- J. Van Gael, Y. Saatchi, Y. W. Teh, and Z. Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the International Conference on Machine Learning*, volume 25, 2008. (page 67)
- B. F. Voight and J. K. Pritchard. Confounding from cryptic relatedness in case-control association studies. *PLOS Genetics*, 1(3), 2005. (page 21)
- A. Webb, J. M. Hancock, and C. C. Holmes. Phylogenetic inference under recombination using bayesian stochastic topology selection. *Bioinformatics*, 25(2):197–203, 2009. (page 22)
- K. A. Wetterstrand. DNA sequencing costs: Data from the National Human Genome Research Institute genome sequencing program. <http://www.genome.gov/sequencingcosts>, 2014. Accessed: Spring 2014. (page 12)
- D. Wingate, A. Stuhlmüller, and N. D. Goodman. Lightweight implementations of probabilistic programming languages via transformational compilation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 14, 2011. (page 115)
- C. Wiuf and J. Hein. Recombination as a point process along sequences. *Theoretical Population Biology*, 55(3):248–59, 1999. (pages 19, 20, and 22)
- F. Wood, J. W. van de Meent, and V. Mansinghka. A new approach to probabilistic programming inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 17, 2014. (pages 109, 115, and 117)
- S. Wright. Evolution in Mendelian populations. *Genetics*, 16(2):97–159, 1931. (page 110)
- Y. Wu, Q. Yan, D. Bickson, Y. Low, and Q. Yang. Efficient multicore collaborative filtering. In *ACM KDD Cup Workshop*, 2011. (page 120)
- E. P. Xing and K.-A. Sohn. Hidden Markov Dirichlet process: Modeling genetic recombination in open ancestral space. *Bayesian Analysis*, 2(2):501–27, 2007a. (pages 28, 31, and 32)

- E. P. Xing and K.-A. Sohn. A nonparametric Bayesian approach for haplotype reconstruction from single and multi-population data. Technical Report CMU-MLD 07-107, Carnegie Mellow University, 2007b. (page 12)
- E. P. Xing, K.-A. Sohn, M. I. Jordan, and Y. W. Teh. Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture. In *Proceedings of the International Conference on Machine Learning*, volume 23, 2006. (pages 12, 28, 31, 32, and 47)
- E. P. Xing, M. I. Jordan, and R. Sharan. Bayesian haplotype inference via the Dirichlet process. *Journal of Computational Biology*, 14(3):267–84, 2007. (page 12)
- H. Yang, X. Chen, and W. H. Wong. Completely phased genome sequencing through chromosome sorting. *Proceedings of the National Academy of Sciences*, 108(1):12–7, 2011. (page 15)
- Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *Proceedings of the International Conference on Algorithmic Aspects in Information and Management*, volume 4, 2008. (page 120)