Department of Mathematics and Statistics

Bayesian methods in bacterial population genomics

Lu Cheng

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII, University Main Building, on October 4th, 2013, at 12 o'clock noon.

University of Helsinki
Finland

Supervisor
Professor Jukka Corander, University of Helsinki, Finland

Pre-examiners
Professor Daniel Thorburn, Stockholm University, Sweden
Professor Tanel Tenson, University of Tartu, Estonia

Opponent
Associate Professor Zhaohui Steve Qin, Emory University, USA

Custos
Professor Jukka Corander, University of Helsinki, Finland

Contact information

Department of Mathematics and Statistics
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: mathstat-info@helsinki.fi
URL: http://www.mathstat.helsinki.fi/
Telephone: +358 9 191 51501, Fax: +358 9 191 51400

ISBN 978-952-10-9204-6 (paperback)
ISBN 978-952-10-9205-3 (PDF)
Helsinki 2013
Unigrafia

Bayesian methods in bacterial population genomics

Lu Cheng

Department of Mathematics and Statistics
P.O. Box 68, FI-00014 University of Helsinki, Finland
lu.cheng@helsinki.fi
https://wiki.helsinki.fi/display/mathstatHenkilokunta/Cheng,+Lu

PhD Thesis
Helsinki, October 2013, 52+50 pages
ISBN 978-952-10-9204-6 (paperback)
ISBN 978-952-10-9205-3 (PDF)

Abstract

Vast amounts of molecular data are being generated every day. However, how to properly harness the data remains often a challenge for many biologists. Firstly, due to the typical large dimension of the molecular data, analyses can either require exhaustive amounts of computer memory or be very time-consuming, or both. Secondly, biological problems often have their own special features, which put demand on specially designed software to obtain meaningful results from statistical analyses without imposing too much requirements on the available computing resources. Finally, the general complexity of many biological research questions necessitates joint use of many different methods, which requires a considerable expertise in properly understanding the possibilities and limitations of the analysis tools.

In the first part of this thesis, we discuss three general Bayesian classification/clustering frameworks, which in the considered applications are targeted towards clustering of DNA sequence data, in particular in the context of bacterial population genomics and evolutionary epidemiology. Based on more generic Bayesian concepts, we have developed several statistical tools for analyzing DNA sequence data in bacterial metagenomics and population genomics.

In the second part of this thesis, we focus on discussing how to reconstruct bacterial evolutionary history from a combination of whole genome sequences and a number of core genes for which a large set of samples are available. A major problem is that for many bacterial species horizontal

gene transfer of DNA, which is often termed as recombination, is relatively frequent and the recombined fragments within genome sequences have a tendency to severely distort the phylogenetic inferences. To obtain computationally viable solutions in practice for a majority of currently emerging genome data sets, it is necessary to divide the problem into parts and use different approaches in combination to perform the whole analysis. We demonstrate this strategy by application to two challenging data sets in the context of evolutionary epidemiology and show that biologically significant conclusions can be drawn by shedding light into the complex patterns of relatedness among strains of bacteria. Both studied organisms (*Escherichia coli* and *Campylobacter jejuni*) are major pathogens of humans and understanding the mechanisms behind the evolution of their populations is of vital importance for human health.

General Terms:
Bacteria, Metagenomics, Genomics, Population genetics

Additional Key Words and Phrases:
Classification, Clustering, BAPS, BratNextGen, BEBaC, DNA, Sequence

Acknowledgements

I want to give the deepest thanks to my supervisor Jukka Corander. He is more like a brother rather than a supervisor, not just because he likes to wear bizarre T-shirts. He guides me going through all kinds of difficulties in my PhD life with his eternal optimism, especially in the early stages. When I start my PhD, I feel like I am the little Jukka in the "Jukka Bros" MTV advertisement. Luckily the big Jukka is very patient and willing to help me hand by hand to catch up with the current fashion — Bayesian Statistics and Microbiology. Later, we communicate by email most of the times. The wonderful thing is that he can always reply in half an hour, which largely heals my procrastination. However, the side effect is that I feel exhausted very quickly since the process is somehow like playing ping-pong with an auto-serving machine.

All my colleagues are indispensable to this work, with whom I have lots of interesting discussions about everything in PhD life. Just to name a few, Jing Tang and Jukka Siren are the role models on my way to PhD. Jing teaches me a lot about how to be a scientist, while Jukka explains me lots of basic concepts in Bayesian statistics. Together with Jie Xiong, Hongyu Su and Alberto Pessia, we discuss about our research projects, propose weird research ideas and complain about being a PhD student. Elina Numminen, Väinö Jääskinen, Paul Blomstedt and my previous colleague Niko Välimäki help me to understand the Finnish society in many different aspects, from the locals' perspective.

I would like to thank all my Chinese friends in Helsinki for your consistent supports. Thanks for inviting me to the parties and excursions, without which the winter will be more gloomy and the summer will be more chilly. Thanks for your timely help whenever I need it. Special thanks are deserved to Huibin Shen and Mengyan Zhang, who have been bothered too many times to take care of my baby. Seven years have passed, but I still remember the scenery talking about dream and love with a guy called Yiming Zhao. It must be one of the best and important times in my life. Hongyu Su helped me many times with moving in and out. Hui Tang and

Tao Xu invites me to many excellent parties and excursions. You Zhou and Eunjee Cho tell me lots of experience of being parents. Zheng Fan is always the first person I seek help for translating Finnish documents. I want to list all your names here, but it will just not end. I remember your happy faces, sad faces, you are there, in my heart.

I thank my pre-examiners Tanel Tenson and Daniel Thorburn for their knowledgeable comments, which greatly improves the quality of the thesis. I thank the Finnish population genetics graduate school and Sigrid Juselius foundation for providing me the financial support.

I wish to thank my parents for bringing me up and encouraging me all the way. The warmest thanks go to my wife Danmei Huang, who venturesomely joins me in the long journey. The last thank goes to my baby — Zhima, who starts my new life.

Contents

List of publications and the author's contributions

This thesis consists of this summary part and the following five original publications, which are referred as article **I-V** and reprinted at the end of the thesis.

Article I Lu Cheng, Alan W. Walker and Jukka Corander (2012)
Bayesian estimation of bacterial community composition from 454 sequencing data. *Nucleic Acids Research*, 40:5240-5249.

L.C. had primary responsibility in method development, design of experiments and writing of the article. L.C. implemented the method.

Article II Lu Cheng, Thomas R. Connor, David M. Aanensen, Brian G. Spratt and Jukka Corander (2011)
Bayesian semi-supervised classification of bacterial samples using MLST databases. *BMC Bioinformatics*, 12:302.

L.C. and J.C. jointly developed the method and designed the experiments, L.C. implemented the method and all authors jointly wrote the article.

Article III Lu Cheng, Thomas R Connor, Jukka Siren, David M Aanensen and Jukka Corander (2013)
Hierarchical and spatially explicit clustering of DNA sequences with BAPS software. *Molecular Biology and Evolution*, 30:1224-8.

L.C. and J.C. jointly developed the methods, L.C. implemented the methods and all authors jointly wrote the article.

Chapter 1

Introduction

When James D. Watson and Francis Crick first discovered the double helix structure of DNA, the secret of life could be written in a mathematical form for the first time. However, until recently, it has remained a challenge to measure the bases in DNA sequences in an efficient, robust and relatively cheap manner from large collections of samples. With the emergence of novel sequencing technologies, especially those called the next generation sequencing technologies, scientists are able to read the DNA sequences in a much greater detail than ever before. Still, the secret of life is far from being totally deciphered yet, and vast amounts of biological data wait to be analyzed on the road towards increasingly detailed insights about how living organisms do function and evolve.

With the ever accumulating masses of sequence data, bioinformatics has established its position as the branch of computational and statistical sciences which acts as a powerful and necessary propeller of biological research. Modern bioinformatics related research can be divided into two broad categories. One category leans toward "informatics", which aims to develop new methods to solve specific problems in biology, i.e. formulate the problems in mathematical terms. For example, software packages such as BEAST [1], FastTree [2], MEGA [3] and RAxML [4] all provide useful tools for studying molecular evolution by a phylogenetics based approach, where the underlying methods are based on a multitude of important algorithmic, mathematical and statistical formalisms and insights that together make daunting analysis tasks possible to complete without access to nearly unlimited computing power. The other category leans more towards "biology", where one often combines different existing bioinformatics tools to provide an answer to a biological question, or make new discoveries.

This thesis, focusing on applications in the area of bacterial population genomics, will summarize my research work from the above two per-

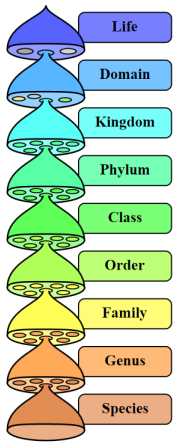

Figure 1.1: The hierarchy of biological classification's eight major taxonomic ranks [5]. Intermediate minor rankings are not shown.

spectives. Chapter 1 gives a simple introduction to bacterial population genomics; Chapter 2 focuses on clustering/classification methods in different scenarios; Chapter 3 introduces an application to retrieve bacterial evolutionary history despite recombination.

1.1 Brief introduction to bacterial domain

Organisms are categorized into different hierarchical taxonomic ranks by biologists, as shown in Figure 1.1. A common understanding [6] categorizes all living organisms into three domains: Bacteria, Archaea, and Eucaryota. It is suggested in a recent study [7] that Eukaryotes originate from a fusion of an achaebacterium and a eubacterum.

Bacteria are closely related with human health. There are trillions of bacteria within the human body cavities, such as nose, skin, gut and so on. Hence bacteria and human are actually cohabiting with each other [8]. Changes in the microbial environment of the human body may lead to diseases. Gill et al. [9] find that the bacterial composition of the gut of newborn babies represents the key factor to stimulate the development of human immune systems. Given the wide range of threats to human health caused by infectious diseases, it is important to understand how bacterial populations are evolving and how disease-causing agents are related to each other, in particular in terms of horizontally transferred genetic material.

The first step to exploring the mysterious world of bacteria was to classify or categorize them using physical appearances. Later on, taxonomists

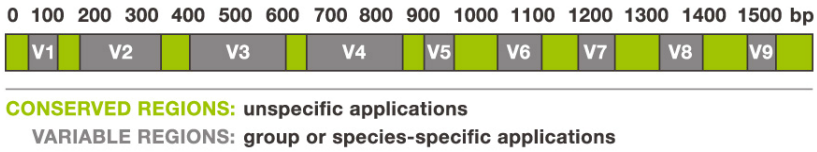

Figure 1.2: Schematic structure of 16S rRNA gene [11]. 16S rRNA gene consists of 9 variable regions (grey) and the rest (green) are universally conserved regions. The conserved regions are used as binding sites for PCR primers and the variable regions are used as fingerprints for bacterial species and genera.

started to use biochemical tests to classify bacteria at different taxonomic ranks. However, due to limitations of biochemical tests and morphological characteristics, these methods cannot usually separate different strains of bacteria representing the same species, while they may have enormous differences in terms of virulence and resistance to antibiotics. Also, in many cases the collected samples represent a mixture of many different bacteria, which can not be easily separated and grown for biochemical testing purposes.

Given that bacterial taxonomy based on physical appearance and simple biochemical properties is riddled with problems, it is not surprising that DNA sequencing represents the most promising approach to understanding and characterization of variation in the bacterial domain. The most widely used approach to DNA based classification is to sequence the 16S rRNA gene [10], which appears almost universally in all bacteria. Figure 1.2 provides a schematic description of the 16S rRNA gene. The 16S rRNA gene has an approximate length of 1500 bp, which contains 9 variable regions flanked by universally conserved regions. The conserved regions of 16S rRNA gene (green parts in Figure 1.2) are so well conserved that they are virtually identical for all bacteria, while the other variable regions (grey parts in Figure 1.2) display variation across the bacterial species. Thus the green fragments are used as binding sites for PCR primers and the grey fragments are used as fingerprints of bacterial species and genera. Therefore, this structure makes the 16S rRNA gene an attractive target for classification purposes.

Although the 16S rRNA gene provides a fairly good resolution to distinguish different bacterial species, outside metagenomics applications it is most often necessary to separate different evolutionary lineages of bacteria at the species level (Figure 1.1). To identify evolutionary relationships among bacterial strains of a single species, multi locus sequence typing

(MLST) [12] was introduced to provide a novel tool for infectious disease epidemiology. MLST genotypes refer to concatenated allelic profile of a bacterial strain at several housekeeping genes found universally within a genus. By definition, housekeeping genes are necessary for maintenance of the basic cellular functions. Hence, nearly all DNA variation occurring within the MLST loci represent neutral, i.e. synonymous mutations, which can be used to trace back the ancestry of strains in a spatio-temporal setting on a fairly large geographical scale. The choice of housekeeping genes depends on the bacteria under investigation, although it has been observed that many different bacterial species do harbour partly the same housekeeping genes such that some of the MLST loci in use are not species-specific.

Despite of the fact that MLST provides a powerful tool for infectious disease epidemiology, there are a lot of settings where MLST sequences do not harbour enough variation to be useful for discriminating between lineages that have important phenotypic differences or to reveal multiple separate transmissions of strains into a host population. When one wishes to study the evolution and transmission of bacteria at a highly detailed level, it is necessary to use whole genome sequence data since very closely related strains often have identical alleles at MLST loci, even if they can differ substantially elsewhere in the genome, e.g. due to frequent horizontal gene transfer. With the help of the whole genome sequence data, scientists are able to detect gene flow and recombination events for instance in the pathogen transmission processes. In a typical epidemiological study using the whole genome sequence data, scientists will carefully select a set of isolates of the same bacterial species, and then sequence their whole genomes. Since bacterial genomes do evolve very rapidly for most species due to phages and also by transformation for many species, even in the whole-genome setting it is typically necessary to restrict the evolutionary analyses to core genes present across all sequenced samples. Genes outside this set, often termed as accessory genes, can also be analyzed, however, it is much more challenging to put forward statistical models that trace their evolutionary dynamics compared to the core genes.

It is far from trivial to assign strains to evolutionary lineages and to estimate the levels of their relatedness using standard phylogenetic methods due to the traces of horizontally transferred genetic material. In standard phylogenetic models neutral evolution is described in terms of independent substitutions occurring in DNA at a specific rate [13]. Since horizontal transfer of DNA breaks the assumptions behind such models, resulting estimates of phylogenies can be severely distorted. Recombination events often tend to introduce many single-nucleotide polymorphism (SNP) sites,

which can obscure the true clonal relationships and distort attempts to dating evolutionary events using statistical methods such as BEAST [1]. Croucher et al. [14] show an excellent example about this phenomenon in pneumococcal evolution.

1.2 Concepts about bacterial population genomics

The term "population genomics" is a fairly new term in the field of population genetics. Below we will discuss the two terms "population" and "genomics" separately.

It is relevant to ask what a "population" actually represents in mathematical and biological terms. Waples and Gaggiotti [15] list many different definitions of a population. Here we use this definition: "A group of individuals of the same species living in close enough proximity that any member of the group can potentially mate with any other member". However, "population" is sometimes used to mean "Operational taxonomic unit (OTU)" in bioinformatics, which actually refers to a cluster given by some clustering software.

"Genomics" generally means the study of genomes, including sequencing of the genomes, study of the genome structure and analysis of the function of genome. It can also refer to the study of recombinations in genomes, i.e. traces of the past, interactions between ancestral organisms that gave rise to the observed genomes.

In a nutshell, bacterial population genomics focuses on how bacteria populations interact with each other, how to discover the interactions by statistical analysis of genomic data and what are the evolutionary routes underlying the data. The genomic analyses are typically involving also rich meta data on samples, representing both phenotypic characteristics of the bacteria (e.g. virulence and antibiotic resistance) and ecological conditions under which samples were acquired, in addition to spatio-temporal information about them.

Chapter 2

Bayesian clustering and classification methods

Bayesian methods are very popular in many different research areas due to their capability to quantify uncertainty in complex systems. In the traditional frequentist statistical approach one usually assumes that there exist a true population value of every parameter in a model, which is estimated from observational or experimental data (or from both data types). In Bayesian statistics inferences are made from a distribution over the parameters, which is learned from data by updating the prior probability distribution of the parameters.

In modern research it is common to use parameter-rich complex models, while the observed data can be very limited compared to the level of model complexity. Under such circumstances it is usually challenging to derive accurate point estimates of the target parameters if uncertainty about latent variables and auxiliary parameters in a model is not appropriately accounted for. In Bayesian statistics inferences about target parameters are sought from the marginal posterior distribution, obtained by integrating out auxiliary parameters and latent variables from a joint model for the data and all unknowns. However, this operation is in general very difficult to do, and various types of approximations are often necessary for practical applicability.

A general Bayesian model includes three key parts: **prior**, **likelihood** and **posterior**, which are abbreviations for the prior probability distribution of the unknowns included in a statistical model, the likelihood function of the model and the posterior probability distribution of the unknowns, respectively.

The **prior** refers to the distribution $p(\theta)$ of the parameters θ before gaining access to data x related to the chosen statistical model. Hence, the

prior distribution reflects the background information about the modelled phenomenon. If only very sparse background information is available, then a non-informative, reference type prior [16] is often used. Such a prior distribution can be interpreted like a default setting, say, in a image-processing software which has a number of parameters determining color saturation, contrast, brightness, etc. Even for complex models it is often possible to use reference conjugate priors for many of the auxiliary parameters conditional on other latent variables or target parameters included in the model. Since analytical integration can be used for conjugate priors, they offer huge computational advantages for fitting the models to data.

The **likelihood** $p(x|\theta)$ specifies how the data are generated under the considered statistical model, which is usually the most central part of the modelling process. In many applications a statistical model can be considered as an approximation to the mechanism causing stochastic variation among the observables, often due to both natural variation and measurement errors. The likelihood function is then usually the most relevant part of the model which captures characteristics of the underlying mechanism. The maximum likelihood (ML) method estimates the model parameters by maximizing the likelihood function. When extensive data are available, ML estimates usually agree with Bayesian estimates.

The **posterior** $p(\theta|x)$ is the conditional distribution of the parameter given the data, which combines information from the prior and the likelihood. According to Bayes' rule, the posterior is given by

$$p(\theta|x) = \frac{p(\theta)p(x|\theta)}{p(x)}. \tag{2.1}$$

The estimate θ_{MAP}, which maximizes the posterior distribution $p(\theta|x)$, is known as the maximum á posterior probability (MAP) estimate. For many inference problems it is more appropriate to use instead posterior mean as an estimate, arising from a squared error loss function in contrast to the absolute error loss which leads to the MAP estimate. Calculations of both estimates require typically efficient methods to explore the parameter space, where Markov Chain Monte Carlo methods (MCMC) and other stochastic simulation methods are often used.

For most modern applications of statistics there are **nuisance** parameters in the model, which are auxiliary parameters not of primary interests. The common approach for handling these parameters is to integrate them out, which usually leads to a more realistic quantification of the uncertainty about other parameters compared for instance to maximization of the joint posterior. This approach can be viewed as model averaging, where models with different configurations of the nuisance parameters are averaged.

This sections below provide an introduction to the general classification and clustering problems, as well as further details for the practical applications in population genomics. In general, classification problems can be divided into the following three categories: unsupervised classification (clustering), supervised classification and semi-supervised classification. Unsupervised classification means we assign the data items to different clusters solely based on the data, without any access to training data. Supervised classification refers to the situation where we have the training data assigned into K classes and we want to assign each of the test data items to one of the K classes. Semi-supervised classification lies between these two extremes and assumes that some test items can have their origins outside of the K classes or clusters for which training data are available. The presentation of Bayesian solutions to these three problems follow the scenarios presented in articles **I-III**.

Each of the following sections describing the classification problems is organized according to the three key parts mentioned above: prior, likelihood, posterior. The basic notation is introduced in the unsupervised classification subsection.

2.1 Unsupervised classification (clustering)

We start by assuming that there are n data items, each of which is denoted by $\mathbf{x}_i$, where $i = 1, 2...n$. Each data item $\mathbf{x}_i$ is a d-dimensional vector, which can be written as $\mathbf{x}_i = (x_{i1}, x_{i1}, ...x_{id})$. Here we denote samples in the whole dataset by a set $N = \{1, 2...n\}$ of integers. A subset of data items $s \subseteq N$ is represented by $\mathbf{x}^{(s)} = \{\mathbf{x}_i : i \in s\}$. Hence the whole dataset is represented by the matrix $\mathbf{x}^{(N)} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)^T$, shown as follows:

$$\mathbf{x}^{(N)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}. \tag{2.2}$$

Equation (2.2) shows that each data item $\mathbf{x}_i$ has d features, i.e. each column corresponds to the observed values of a variable. For example, if a data item represents an individual, then the features could be age, sex, height, weight and so on. The observed value x_{ij} for the jth feature of data item i can be either continuous or discrete. In the population genomics applications considered in this thesis, we assume all the features are discrete and there are r_j discrete values for feature j. This restriction

arises from the fact that DNA sequence data are discrete and only such data are considered throughout the thesis.

The aim of unsupervised classification (clustering) is to find a partition $S = (s_1, s_2, ..., s_k)$ of the whole set of samples N such that $\cup_{c=1}^{k} s_c = N$ and $s_c \cap s_{c'} = \emptyset$, for all pairs of c, c' ranging between 1 and k. In other words, the partition S assigns n data items to k mutually exclusive clusters. Here we denote the number of data items in a cluster s_c by its cardinality $|s_c|$, from which we can easily deduce that $\sum_{c=1}^{k} |s_c| = n$. All eligible partitions constitute the partition space $\mathcal{S}$. In the application of Bayesian clustering we want to find the optimal partition $\hat{S}$ in the space $\mathcal{S}$ which maximizes the posterior probability $p(S|\mathbf{x}^{(N)})$.

Clustering of data items is generally based on the similarities between data items, either in a deterministic fashion or in probabilistic terms as in the case of Bayesian clustering. The basic intuition is that data items within a cluster are assumed to be more similar to each other than to data items outside the cluster. Many different clustering methods have been introduced in the statistical and computer science literature, such as K-means [17], Expectation Maximization (EM) [18], hierarchical clustering [19] and so on. These methods usually require the number of clusters or a cutoff to define a cluster, which are most often unknown in advance. The methods introduced in this thesis, however, only necessitate the specification of the maximum number of clusters, which is denoted by K_{max}.

Both K-means and EM algorithms require the number of clusters K_C as an input and use a similar optimization process. K-means algorithm assumes that each data item is generated by its own cluster, while EM algorithm assigns probabilities of belonging to different clusters for each data item. In some sense, K-means algorithm is a simplified version of EM algorithm. Both algorithms change the labels of the data items according to their own optimization processes, such that the final partition is optimal under pre-defined loss functions. However, partitions with unequal number of clusters are not comparable, which makes it difficult to choose an appropriate K_C.

Hierarchical clustering algorithm requires a cutoff as an input. It first calculates a distance matrix for all pairs of data items. Then the data items are agglomerated sequentially from the pair with the shortest distance, during which a tree is constructed. It then uses the input cutoff to split the tree into clusters such that distances between data items within a cluster are less than the cutoff. The difficulty lying here is how to set a proper cutoff.

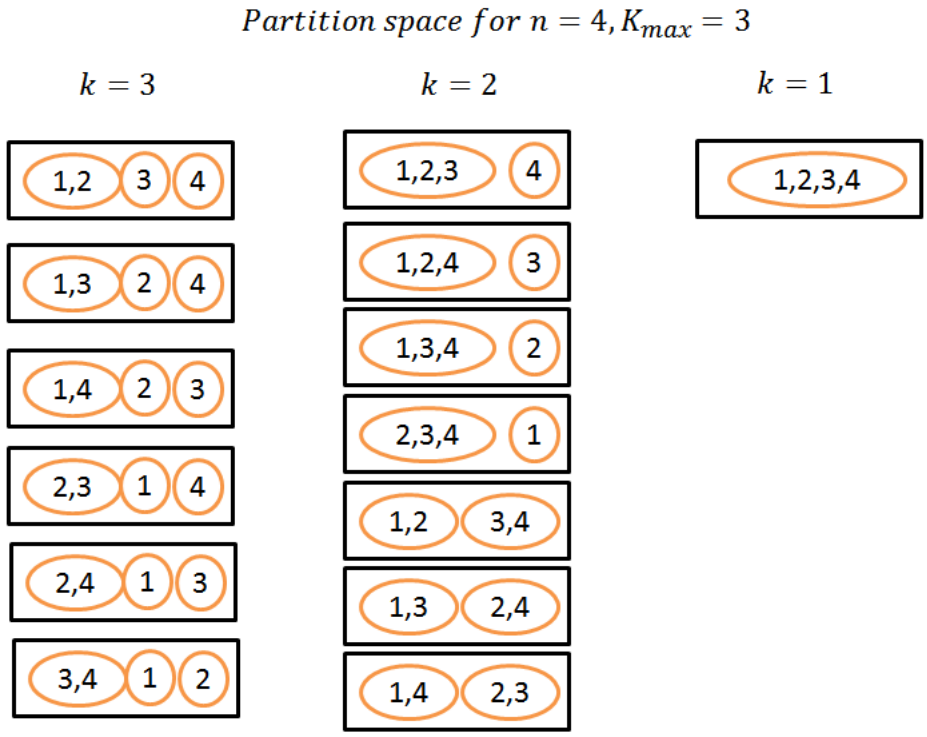

Figure 2.1: The partition space for $n = 4$ and $K_{max} = 3$

2.1.1 Prior

We now specify the prior $p(S)$ for a partition S, with the maximum number of clusters denoted by K_{max}. To provide an intuitive description about the partition space, we illustrate it for $n = 4$ data items and $K_{max} = 3$, as shown in Figure 2.1. The total number of eligible partitions is a sum of the number of partitions with $k = 1 \cdots K_{max}$, where k denotes the number of clusters in S.

Perhaps the simplest possible prior for clustering purposes is to assign equal probability to each partition in the partition space, which leads to the uniform prior shown in equation (2.3).

$$p(S) = 1/\sum_{k=1}^{K_{max}} S(n,k), \tag{2.3}$$

where $S(n,k)$ is the Stirling number of second kind [20], which is the number

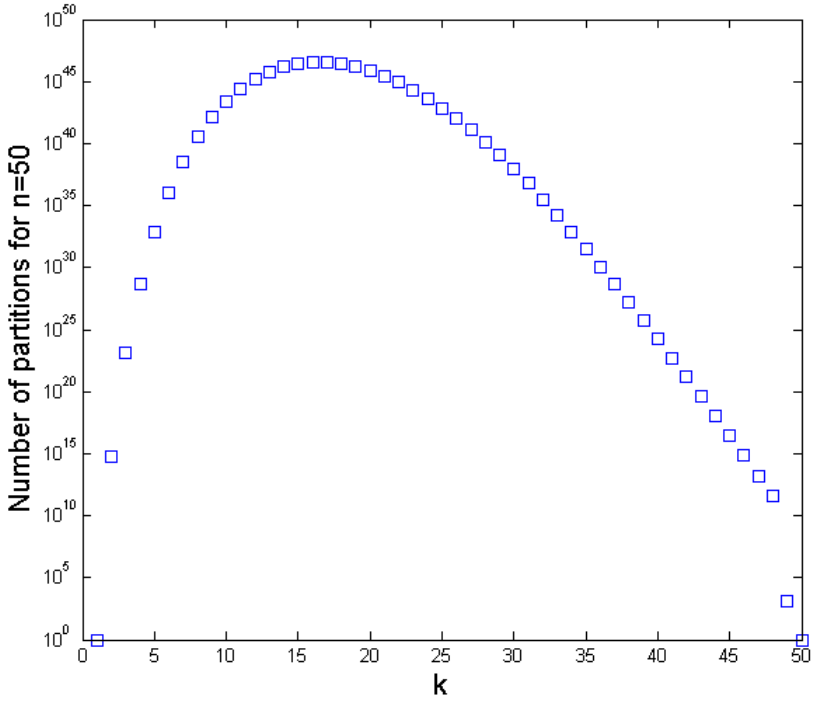

Figure 2.2: The number of partitions for $n = 50$ and $k = 1, 2, \cdots, 50$. The maximum value is attained at $k = 16$. Note that the Y-axis is in $\log_{10}$ scale.

of ways to partition a set of n objects into k non-empty subsets.

The uniform prior considers each candidate partition equally, which can be considered plausible when comparing any two partitions and there is no *a priori* information to favor one partition over the other. However, the prior does not lead to an uniform prior on the number of clusters of the partitions, which is simple consequence of the behaviour of the Stirling number of the second kind. To again provide some intuitive characterization of this behaviour, the number of partitions for $k = 1, 2, \cdots, 50$ and $n = 50$ are shown in Figure 2.2. This distribution is unimodal and the mode is located at $k = 16$. It tells us that the prior distribution prefers partitions with around 16 clusters.

Although the distribution over k implied by a uniform prior on S is non-uniform, it may still lead to reasonable inferences when the data are sufficiently high-dimensional to prevent unwanted effects from the accumu-

lation of prior probability mass to higher values of k.

As an alternative prior, we consider a uniform prior over k, which results in higher prior probabilities for partitions with smaller k, as shown in equation (2.4).

$$p(S) = \frac{1}{K_{max} \times S(n, |S|)}, \tag{2.4}$$

where $|S|$ denotes the number of clusters in partition S. However, for many of the population genomic applications considered in this thesis, the DNA sequence data are informative enough to lead to identical partitions as MAP estimates.

2.1.2 Likelihood

Next we consider calculation of the marginal likelihood $p(\mathbf{x}^{(N)}|S)$ given a partition S with k clusters. An assumption in our model is that the data items of each cluster are generated by an independent process. This means the probability of generating a data item is only related with the parameters of the cluster it belongs to. Another general assumption is that the features are conditionally independent of each other, although we also consider Markovian type of dependence among the features in some cases. Both assumptions enable us to calculate the likelihood of a cluster by multiplication of the likelihoods over the sequence of all observed features.

Under the above assumptions, we introduce a set of nuisance parameters $\theta = \{\theta_{cij} | 1 \leq c \leq k, 1 \leq i \leq d, 1 \leq j \leq r_i\}$ to derive an explicit expression of the likelihood function $p(\mathbf{x}^{(N)}|\theta, S)$, where θ_{cij} is the probability of observing the jth value of the ith feature (sequence position) in cluster c. In the specific models we assume that the values correspond to the DNA bases, which are usually written in the order of {'A','C','G','T'} and indexed by 1,2,3,4 (thus $r_i = 4$ here), respectively. Therefore, generating the data for column i of cluster c corresponds to drawing $|s_c|$ balls with replacement from an urn containing balls labelled by 'ACGT' with the probabilities specified by $\theta_{ci\cdot} = (\theta_{ci1}, \theta_{ci2}, \theta_{ci3}, \theta_{ci4})$. The likelihood is then given as follows:

$$p(\mathbf{x}^{(N)}|\theta, S) = \prod_{c=1}^{k} p(\mathbf{x}^{(s_c)}|\theta, S) = \prod_{c=1}^{k} \prod_{i=1}^{d} \prod_{j=1}^{r_j} \theta_{cij}^{n_{cij}}, \tag{2.5}$$

where n_{cij} is the observed count of the jth base in the ith feature (sequence position) of the cluster c. However, since the nuisance parameter θ is not of interest, in the current application, it should be integrated out when making

inferences about the partition S. This leads to the marginal likelihood as follows:

$$p(\mathbf{x}^{(N)}|S) = \int_{\Theta} p(\mathbf{x}^{(N)}|\theta, S)p(\theta|S)d\theta. \tag{2.6}$$

A computationally convenient standard choice [16] as a prior for the parameters θ is Dirichlet distribution, which enables analytical integration for calculation of the marginal likelihood. The probability density function for Dirichlet distribution $Dir(\alpha)$, where $\alpha = (\alpha_1, ..., \alpha_K)$, can be written as follows:

$$p(x|\alpha) = p(x_1, ..., x_K|\alpha_1, ..., \alpha_K) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}, \tag{2.7}$$

where $x_1, ..., x_K > 0$ and $x_1 + ... + x_K = 1$. In the DNA sequence case, we assume $\theta_{ci\cdot} \sim Dir(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, where $\alpha_i = 0.25, i = 1, 2, 3, 4$. Note that $\theta_{ci\cdot}$ satisfies the requirements that the sum of random variables is 1, i.e. $\sum_{j=1}^{4} \theta_{cij} = 1$, as well as that each random variable is greater than 0. Thus, the prior for feature i of cluster c is explicitly written as:

$$\begin{aligned} p(\theta_{ci\cdot}|\alpha) &= p(\theta_{ci1}, ..., \theta_{ci4}|\alpha_1, ..., \alpha_4) \\ &= \frac{\Gamma(\sum_{j=1}^{4} \alpha_j)}{\prod_{j=1}^{4} \Gamma(\alpha_j)} \prod_{j=1}^{4} \theta_{cij}^{\alpha_j - 1} \\ &= \frac{1}{\prod_{j=1}^{4} \Gamma(\alpha_j)} \prod_{j=1}^{4} \theta_{cij}^{\alpha_j - 1}. \end{aligned} \tag{2.8}$$

Note that $\sum_{j=1}^{4} \alpha_j = 1$ and $\Gamma(1) = 1$. The above prior leads to the following

analytical form of the marginal likelihood:

$$\begin{aligned}
p(\mathbf{x}^{(N)}|S) &= \int_{\Theta} p(\mathbf{x}^{(N)}|\theta, S)p(\theta)d\theta \\
&= \int_{\Theta} \prod_{c=1}^{k}\prod_{i=1}^{d}\{\prod_{j=1}^{4}\theta_{cij}^{n_{cij}} \cdot \frac{1}{\prod_{j=1}^{4}\Gamma(\alpha_j)}\prod_{j=1}^{4}\theta_{cij}^{\alpha_j-1}\}d\theta \\
&= \prod_{c=1}^{k}\prod_{i=1}^{d}\int_{\Theta_{ci\cdot}}\{\frac{1}{\prod_{j=1}^{4}\Gamma(\alpha_j)} \cdot \prod_{j=1}^{4}\theta_{cij}^{n_{cij}+\alpha_j-1}\}d\theta_{ci\cdot} \\
&= \prod_{c=1}^{k}\prod_{i=1}^{d}\frac{1}{\prod_{j=1}^{4}\Gamma(\alpha_j)} \cdot \frac{\prod_{j=1}^{4}\Gamma(n_{cij}+\alpha_j)}{\Gamma(\sum_{j=1}^{4}(n_{cij}+\alpha_j))} \\
&\cdot \int_{\Theta_{ci\cdot}}\{\frac{\Gamma(\sum_{j=1}^{4}(n_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(n_{cij}+\alpha_j)}\prod_{j=1}^{4}\theta_{cij}^{(n_{cij}+\alpha_j)-1}\}d\theta_{ci\cdot} \\
&= \prod_{c=1}^{k}\prod_{i=1}^{d}\frac{1}{\prod_{j=1}^{4}\Gamma(\alpha_j)} \cdot \frac{\prod_{j=1}^{4}\Gamma(n_{cij}+\alpha_j)}{\Gamma(\sum_{j=1}^{4}n_{cij}+1)}. \qquad (2.9)
\end{aligned}$$

After marginalization over the nuisance parameters, the marginal likelihood only depends on the hyperparameters α and the data $\mathbf{x}^{(N)}$.

2.1.3 Posterior

According to Bayes' theorem, the posterior probability of a partition is given as follows:

$$p(S|\mathbf{x}^{(N)}) = \frac{p(\mathbf{x}^{(N)}|S)p(S)}{p(\mathbf{x}^{(N)})} \propto p(\mathbf{x}^{(N)}|S)p(S) \qquad (2.10)$$

where $p(\mathbf{x}^{(N)})$ is a constant which does not depend on S. When using a uniform prior (equation (2.3)), the posterior probability $p(S|\mathbf{x}^{(N)})$ is further simplified since it is directly proportional to the marginal likelihood $p(\mathbf{x}^{(N)}|S)$. This means it is not necessary to calculate the posterior probability directly to compare any two given partitions S and S'. Instead, the comparison can be carried out through comparing the marginal likelihood (equation (2.9)).

2.1.4 Inference algorithm using a stochastic optimization process

Given the ability of comparing any two partitions in an analytic form, we need to design an algorithm to identify the partition $\hat{S}$ which maximizes the

posterior probability (equation (2.10)). However, the partition space $\mathcal{S}$ is so large that it is in practice impossible to enumerate all possible partitions. Thus, we need to resort to MCMC methods or other stochastic process based methods to explore the partition space. Efficient MCMC methods are very challenging to design for large-scale clustering applications and the resulting algorithms could be very slow if the proposal operators are not chosen appropriately (see [21]). We will focus on using a stochastic optimization process to do the inference, which can be interpreted as a "greedified" version of the non-reversible MCMC algorithm introduced in [22]. The greedy stochastic algorithm is defined as follows:

Input : the input data $\mathbf{x}^{(N)}$ and the maximum number of clusters K_{max} defined by the user.

Initialization : calculate the pairwise Hamming distance between the data items, cluster N into K_{max} clusters using complete linkage algorithm [19], set the resulting partition S as the initial partition.

Stochastic search : apply each of the four search operators described below to the the current partition S in a random order. Then, if the resulting partition leads to a higher marginal likelihood (equation (2.9)), update the current partition S, otherwise keep the current partition. If all operators fail to update the current partition, then stop and set the best partition $\hat{S}$ as the current partition S.

i In a random order relocate a data item $\mathbf{x}_i$ to another cluster that leads to the maximal increase in the marginal likelihood (equation (2.9)). The option of moving the data item into an empty cluster is also considered, unless the total number of clusters exceeds K_{max}.

ii In a random order, merge the two clusters which leads to the maximum increase in the marginal likelihood (equation (2.9)). This operator considers also merging of singleton clusters (only one data item in the cluster) that might be generated by the other operators.

iii In a random order, split each cluster into two subclusters using complete linkage clustering algorithm, where the Hamming distance is used. Then try reassigning each subcluster to another

cluster including empty clusters. Choose the split and reassignment that lead to the maximal increase in the marginal likelihood (equation (2.9)).

iv In a random order, split each cluster into m subclusters using complete linkage clustering algorithm as described in operator (**iii**), where $m = min(20, \lceil |s_c|/5 \rceil)$ and $|s_c|$ is the total number of data items in the cluster. Then try to reassign each subcluster to another cluster; choose the split and reassignment that leads to the maximal increase in the marginal likelihood (equation (2.9)).

Output : an estimate of the best partition $\hat{S}$, leading to the highest marginal likelihood $p(\mathbf{x}^{(N)}|\hat{S})$

The above greedy stochastic algorithm uses heuristics to visit the high probability areas in the partition space $\mathcal{S}$. Operator **i** exchanges data items between clusters to optimize the current partition; operator **ii** merges similar clusters together to reduce the number of clusters; operator **iii** and **iv** split out heterogeneous data items to optimize the current partition. Although the algorithm does not guarantee global optimality of the solution, it searches the high probability areas very efficiently according to our intensive experiments. In practice, we have a wealth of empirical evidence that the estimated partition tends to be biologically meaningful and more sensible than alternative estimates based on standard methods for Bayesian computation, such as the Gibbs sampler or Metropolis-Hastings algorithm using completely random proposals.

Note that other distances between samples rather than Hamming distance could also be utilized here, depending on specific scenarios. In practice, the marginal likelihood needs to be calculated on a log scale to avoid numerical overflow, since the values are in general extremely small.

2.2 Supervised classification

Supervised classification differs from unsupervised classification (clustering) in that it requires training data, which contain K classes (or groups). The primary aim is often to assign the unlabeled data items into any of the K classes, however, also purposes are frequently considered, where one typically calculates some functionals of the data assigned into each class. An example of this is the relative contribution of each known source (class) to the population of unlabeled samples.

In the context of bacterial population genomics, a popular application of supervised classification is to classify a new sample to a selected taxonomic rank (usually species) based on its sequence information. The training data is usually a sequence database, which is a large collection of sequences and their labels from various biological projects. The test data usually contain sequences produced in a new biological project, where assigning labels to the sequences is very important in understanding the data.

In this subsection, the training data are denoted by $\mathbf{z}^{(M)}$, where $M = \{1, 2, \cdots, m\}$. The training data are assumed to be divided into K classes based on some auxiliary knowledge or an earlier unsupervised analysis, which are denoted by $T = \{T_1, T_2, \cdots, T_m\}$, where $T_i \in \{1, 2, \cdots, K\}$. The whole training dataset $\mathbf{z}^{(M)} = (\mathbf{z}_1, \cdots, \mathbf{z}_m)^T$ is organized as follows:

$$\mathbf{z}^{(M)} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_m \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1d} \\ z_{21} & z_{22} & \cdots & z_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{md} \end{pmatrix} \tag{2.11}$$

We now assume that there are n data items in the test data $\mathbf{x}^{(N)}$, where $N = \{1, 2, \cdots, n\}$. Each test data item is denoted by $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$. Note that the test data items have the same features as the training data items. The aim is to assign a label S_i to each test data item $\mathbf{x}_i$ based on its resemblance to the observations within each group of the training data. The joint labeling of the test data is denoted by $S = \{S_1, S_2, \cdots, S_n\}$, where $S_i \in \{1, 2, \cdots, K\}$.

A typical assumption is that a test data item is generated from one of the underlying distributions of the K classes in the training data, where the parameters of the underlying distributions are learned from the training data. Thus the test data items are independent given the known parameters, such that the labeling of one data item will not affect the others. The labeling of one data item is solely based on the information of the training data, which does not borrow any information from other test data items. When the training data are very sparse, there is a high risk of wrong labeling of the test data.

The strategy adopted in this thesis, however, does not assume independence of the test data items. Instead, we label all test data items simultaneously such that the labeling of one test data item also borrows statistical strength from other test data items. Corander et al. [23] provide a detailed discussion about two classifiers based on the above two classification principles and another marginalized classifier.

The posterior probability of the joint labeling S is given by

$$p(S|\mathbf{x}^{(N)}, \mathbf{z}^{(M)}, T) = \frac{p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S)p(S|\mathbf{z}^{(M)}, T)p(\mathbf{z}^{(M)}, T)}{p(\mathbf{x}^{(N)}, \mathbf{z}^{(M)}, T)}$$
$$\propto p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S)p(S|\mathbf{z}^{(M)}, T), \tag{2.12}$$

where $p(\mathbf{z}^{(M)}, T)$ and $p(\mathbf{x}^{(N)}, \mathbf{z}^{(M)}, T)$ are constants with respect to S. The aim is to seek a joint labeling $\hat{S}$ of the test data items which maximizes the posterior probability, i.e.

$$\hat{S} = \arg\max_{S} p(S|\mathbf{x}^{(N)}, \mathbf{z}^{(M)}, T). \tag{2.13}$$

2.2.1 Prior

To place a prior distribution for S, we only need to know the number of classes K in the training data. Hence it is reasonable to assume that the prior distribution of S is independent of the training data $\mathbf{z}^{(M)}$. Since each test data item could be placed in any of the K classes, the prior of S equals

$$p(S|\mathbf{z}^{(M)}, T) = p(S|T) = \frac{1}{K^n}. \tag{2.14}$$

2.2.2 Likelihood

The marginal likelihood in equation (2.12) does not have an explicit form, thus we need to introduce nuisance parameter θ to calculate it. The nuisance parameter here is the same as that defined in the previous section (equation (2.5)). With the help of the θ, the likelihood in equation (2.12) can be written as follows:

$$p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S) = \int_{\Theta} p(\mathbf{x}^{(N)}|\theta, \mathbf{z}^{(M)}, T, S)p(\theta|\mathbf{z}^{(M)}, T, S)d\theta$$
$$= \int_{\Theta} p(\mathbf{x}^{(N)}|\theta, S)p(\theta|\mathbf{z}^{(M)}, T)d\theta \tag{2.15}$$

where we implicitly assume that the test data depend on the training data only through the nuisance parameters θ.

The first term in the integral of equation (2.15) is the likelihood of generating the test data $\mathbf{x}^{(N)}$ given the nuisance parameter θ and partition S, which has an identical expression as equation (2.5). The second term is the posterior probability of θ given the training data $\mathbf{z}^{(M)}$ and partition T.

Again we assume the same Dirichlet prior as that in equation (2.8) for θ, which leads to a posterior as follows:

$$\begin{aligned} p(\theta|\mathbf{z}^{(M)}, T) &\propto p(\mathbf{z}^{(M)}, T|\theta)p(\theta) \\ &\propto \prod_{c=1}^{K}\prod_{i=1}^{d}\{\prod_{j=1}^{4}\theta_{cij}^{m_{cij}} \cdot \frac{1}{\prod_{j=1}^{4}\Gamma(\alpha_j)}\prod_{j=1}^{4}\theta_{cij}^{\alpha_j-1}\} \\ &\propto \prod_{c=1}^{K}\prod_{i=1}^{d}\prod_{j=1}^{4}\theta_{cij}^{m_{cij}+\alpha_j-1}, \end{aligned} \tag{2.16}$$

where m_{cij} is the number of the jth base in the ith column of class c in the training data $\mathbf{z}^{(M)}$. We can easily observe that the posterior of $\theta_{ci\cdot}$ is a Dirichlet distribution $Dir(m_{ci1}+\alpha_1, \cdots, m_{ci4}+\alpha_4)$.

By plugging equation (2.5) and (2.16) into (2.15), we get

$$\begin{aligned} & p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S) \\ &= \int_{\Theta}\prod_{c=1}^{K}\prod_{i=1}^{d}\prod_{j=1}^{4}\theta_{cij}^{n_{cij}} \cdot \frac{\Gamma(\sum_{j=1}^{4}(m_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(m_{cij}+\alpha_j)}\theta_{cij}^{m_{cij}+\alpha_j-1}d\theta \\ &= \prod_{c=1}^{K}\prod_{i=1}^{d}\frac{\Gamma(\sum_{j=1}^{4}(m_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(m_{cij}+\alpha_j)}\int_{\Theta_{ci\cdot}}\prod_{j=1}^{4}\theta_{cij}^{n_{cij}+m_{cij}+\alpha_j-1}d\theta_{ci\cdot} \\ &= \prod_{c=1}^{K}\prod_{i=1}^{d}\frac{\Gamma(\sum_{j=1}^{4}(m_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(m_{cij}+\alpha_j)} \cdot \frac{\prod_{j=1}^{4}\Gamma(n_{cij}+m_{cij}+\alpha_j)}{\Gamma(\sum_{j=1}^{4}(n_{cij}+m_{cij}+\alpha_j))} \end{aligned} \tag{2.17}$$

where the last integration follows from the properties of the product Dirichlet distribution.

2.2.3 Posterior & Inference

Equation (2.14) and (2.17) provide explicit expressions of the prior and the marginal likelihood in equation (2.12). Thus we are able to compare the posterior probability given any two labelings S and S' of the test data.

The inference is almost the same as that in the unsupervised classification, except that we set $K_{max} = K$. Therefore, the test data items can be assigned to maximally K classes and at least 1 class. The algorithm chooses the partition $\hat{S}$ which maximizes the posterior (equation (2.12)).

2.3 Semi-supervised classification

Semi-supervised classification is a hybrid of unsupervised classification and supervised classification. Like supervised classification, it also requires

training data which are grouped *a priori* into K classes. However, the test data items are not assumed to strictly represent only the K pre-specified sources, but can either be merged with the existing K classes in the training data, or form new classes/clusters, similar to unsupervised classification. The original biological motivation of semi-supervised classification comes from [22] and the predictive semi-supervised classification approach is formally introduced in [23].

Here we use the same notations as the supervised classification case. The training data $\mathbf{z}^{(M)}$, where $M = \{1, 2, \cdots, m\}$, are assumed to be divided into K classes by the labeling $T = \{T_1, T_2, \cdots, T_m\}$ and $T_i \in \{1, 2, \cdots, K\}$. The test data $\mathbf{x}^{(N)}$, where $N = \{1, 2, \cdots, n\}$, are labeled by $S = \{S_1, S_2, \cdots, S_n\}$ and $S_i \in \{1, 2, \cdots, K\}$. Besides the training data and test data, the inputs to the semi-supervised classification also include the maximum number of classes K_{max} in the test data, which is specified by the user. To allow the discovery of novel clusters formed by test data items, K_{max} should be larger than K, i.e. $K_{max} > K$.

The posterior probability of the joint labeling S is the same as the supervised classification case (equation (2.12)). The aim is also the same – try to find a labeling $\hat{S}$ that maximize the posterior probability, shown by equation (2.13).

2.3.1 Prior

Let us first define the prior $p(S|\mathbf{z}^{(M)}, T)$ for the simultaneous labeling of the test data items, conditional on the training data and its labeling. Like the supervised classification, we assume the prior distribution of S depends on the training data only through the number of classes K in the training data. We choose an uniform prior for S like in the unsupervised classification scenario

$$p(S|\mathbf{z}^{(M)}, T) = p(S|T) = \frac{1}{|\mathcal{S}|}, \tag{2.18}$$

where $\mathcal{S}$ denotes the space of S and $|\mathcal{S}|$ is the number of all possible simultaneous labelings of the data.

Calculation of $|\mathcal{S}|$ has been given in [23] and we use its result directly. Before giving the formula, we need to introduce several notations in [23]. It is assumed that there are k_1 classes (labeled $\{1, 2, \cdots, k_1\}$) in the training data and k_2 novel classes (labeled $\{k_1 + 1, k_1 + 2, \cdots, k_1 + k_2\}$) are formed in the test data. It is obvious to see that $k_1 = K$ and $k_1 + k_2 \leq K_{max}$. We assume r out of n test data items are assigned to the k_1 classes. The r test data items can be chosen in $\binom{n}{r}$ ways and assigned to k_1 classes in k_1^r ways. Then the remaining $n - r$ test data items are randomly assigned to

a stochastic number of urns and form k_2 novel classes. $|\mathcal{S}|$ is obtained by summing over all possible values of r

$$|\mathcal{S}| = \sum_{r=0}^{n} \binom{n}{k} k_1^r B_{n-r} = \sum_{r=0}^{n} \binom{n}{k} k_1^r \sum_{k_2=1}^{\infty} \frac{k_2^{n-r}}{k_2!}, \tag{2.19}$$

where B_{n-r} is the Bell number for $n-r$ test data items. The Bell number B_n is the number of all possible partitions of a set with n items.

Although we show how to calculate $|\mathcal{S}|$, it is not necessary to calculate explicitly in practice since the uniform prior gives equal weight to each labeling, which will be canceled out when comparing the posterior probabilities of any two labelings.

2.3.2 Likelihood

We now provide an explicit form for the marginal likelihood $p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S)$, which is slightly different from the supervised classification case due to the k_2 novel classes.

Similar to the supervised classification case, we assume that the test data depend on the training data only through the nuisance parameters θ, which are defined the sames as equation (2.5). For an existing class $c \in \{1, 2, \cdots, k_1\}$, $\theta_{ci\cdot}$ is governed by the posterior $Dir(\alpha_{i1}+m_{ci1}, \cdots, \alpha_{ir_i}+m_{ci4})$, where $i \in \{1, \cdots, d\}$ is an index of a feature and m_{cij} is the count of the jth base in the ith feature of class c of the training data (see equation (2.16)). For a novel class $c \in \{k_1+1, k_1+2, \cdots, k_1+k_2\}$, $\theta_{ci\cdot}$ is only governed by the prior $Dir(\alpha_{i1}, \cdots, \alpha_{i4})$. Therefore, the marginal likelihood

equals

$$
\begin{aligned}
& p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S) \\
&= \int_{\Theta} p(\mathbf{x}^{(N)}|\theta, \mathbf{z}^{(M)}, T, S) p(\theta|\mathbf{z}^{(M)}, T, S) d\theta \\
&= \prod_{c=1}^{k_1} \prod_{i=1}^{d} \int_{\Theta_{ci\cdot}} \prod_{j=1}^{4} \theta_{cij}^{n_{cij}} \cdot \frac{\Gamma(\sum_{j=1}^{4}(m_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(m_{cij}+\alpha_j)} \theta_{cij}^{m_{cij}+\alpha_j-1} d\theta_{ci\cdot} \\
&\times \prod_{c=k_1+1}^{k_1+k_2} \prod_{i=1}^{d} \int_{\Theta_{ci\cdot}} \prod_{j=1}^{4} \theta_{cij}^{n_{cij}} \cdot \frac{\Gamma(\sum_{j=1}^{4}(\alpha_j))}{\prod_{j=1}^{4}\Gamma(\alpha_j)} \theta_{cij}^{\alpha_j-1} d\theta_{ci\cdot} \\
&= \prod_{c=1}^{k_1} \prod_{i=1}^{d} \frac{\Gamma(\sum_{j=1}^{4}(m_{cij}+\alpha_j))}{\prod_{j=1}^{4}\Gamma(m_{cij}+\alpha_j)} \cdot \frac{\prod_{j=1}^{4}\Gamma(n_{cij}+m_{cij}+\alpha_j)}{\Gamma(\sum_{j=1}^{4}(n_{cij}+m_{cij}+\alpha_j))} \\
&\times \prod_{c=k_1+1}^{k_1+k_2} \prod_{i=1}^{d} \frac{\Gamma(1)}{\prod_{j=1}^{4}\Gamma(\alpha_j)} \cdot \frac{\prod_{j=1}^{4}\Gamma(n_{cij}+\alpha_j)}{\Gamma(\sum_{j=1}^{4}(n_{cij}+\alpha_j))},
\end{aligned} \tag{2.20}
$$

where n_{cij} is the count of the jth base of feature i of classes c in the test data $\mathbf{x}^{(N)}$. The integrations are derived in the same way as equation (2.17).

2.3.3 Posterior & Inference

Similar to the supervised classification case (equation (2.12)), we have the same derivation for the posterior probability of a labeling S

$$
p(S|\mathbf{x}^{(N)}, \mathbf{z}^{(M)}, T) \propto p(\mathbf{x}^{(N)}|\mathbf{z}^{(M)}, T, S) p(S|\mathbf{z}^{(M)}, T), \tag{2.21}
$$

where equation (2.20) and (2.18) provide explicit forms to the marginal likelihood and the prior of S, respectively.

Given equation (2.21), we are able to compare the posterior probabilities of any two labelings of the test data items. Thus, as in the unsupervised scenario, we can use the stochastic optimization approach to search the labeling space, with the following modifications to the proposed operators.

i Only test data items are moved here.

ii Never merge two existing clusters in the training data.

iii-iv Never split an existing cluster in the training data.

2.4 Clustering and classification in practice

The previous sections provide three general frameworks for classification discrete data. Articles **I-III** solve real biological problems based on these frameworks. This section will provide a discussion of the practical issues encountered in the real biological applications.

Perhaps the most immediate and central arising in these applications is how to sensibly set the maximum number of clusters K_{max}. Theoretically we could set it as large as possible. In practice we usually set it to a sufficiently large number such that the number of clusters $K_{\hat{S}}$ in the derived partition $\hat{S}$ is smaller than K_{max}. Of course sometimes $K_{\hat{S}}$ equals K_{max} and this indicates that one should try to explore the posterior also for a larger K_{max}. Compared with other classification algorithms such as Expectation Maximization and K-means, setting K_{max} is much easier than choosing the correct or optimal number of clusters K_C. In the latter case, it can be necessary to consider a very large range of values of K_C and decide which is the most reasonable choice based on the clustering results. This process is in general very tedious and also computationally more burdensome than using an algorithm in which the number of clusters is not fixed.

When collecting bacterial samples, especially in an epidemiology study, scientists will often also have access to meta information, such as age, gender, symptoms, date, the location and so on. Here we consider utilizing the location data. The location data are usually stored as Global Positioning System (GPS) coordinates, which are the latitude and longitude of the locations. The locations provide in general prior information regarding the relationships of the samples. In certain applications it is reasonable to assume that two samples are more likely to be similar to each other if they are close in the geographic sense. Therefore, it is possible to use a spatially explicitly prior distribution for the clustering solutions, instead of a uniform prior on the partition of samples. The spatial prior has been considered in many applications to population genetics, for details see [24].

Sometimes, the assumption that different sites of the sequence are independent may lead to unreasonable approximation of the data likelihood under a given clustering. It is known that coding DNA sequences usually show dependence between neighboring sites. For instance, some codons coding the same amino acids are used more frequently than others in a certain gene. Higher frequencies of these codons could be approximately described by a second-order Markov property, leading to a model for the sequence as a second-order Markov chain instead of assuming independent sites. Figure 2.3 shows a example of a six letter sequence $s_1s_2s_3s_4s_5s_6$.

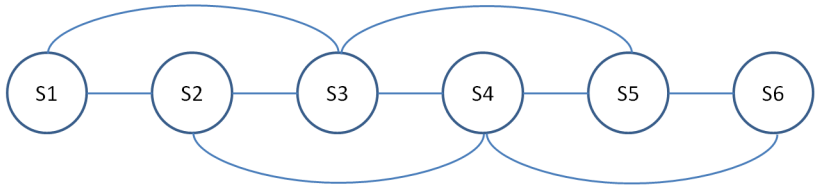

Figure 2.3: 2nd order Markov chain. This is a 2nd order Markov chain for a six letter sequence.

To calculate the likelihood of this sequence, we need to decompose the sequence into **cliques** and **separators** [25]. The cliques in Figure 2.3 are $\{s_1s_2s_3, s_2s_3s_4, s_3s_4s_5, s_4s_5s_6\}$ and the corresponding separators are $\{s_2s_3, s_3s_4, s_4s_5\}$. The likelihood of the sequence is given as follows.

$$p(s_1s_2s_3s_4s_5s_6) = \frac{\prod_{i=1}^{4} p(s_is_{i+1}s_{i+2})}{\prod_{i=2}^{4} p(s_is_{i+1})} \tag{2.22}$$

Equation (2.22) is factorized into probabilities of the cliques and separators, which enables us to use a simple modification of the marginal likelihood formula to obtain an analytic expression for the marginal likelihood of the data. Based on this idea, article **II** implements a semi-supervised classification method for classification MLST data under the 2nd order Markov model.

A common characteristic of bacterial population data is that there may exist substructures in the derived clusters. As an example, assume we collected a dataset of a pathogen from wide geographical range. We then use the introduced clustering approach to assign samples into genetically distinct groups, which could agree with the geographical locations, say at the level of a single country, due to spatio-temporal restrictions in the underlying transmission process. If we take out samples from a cluster which correspond to a specific country, we may find interesting substructure in this cluster when analyzing the samples separately from others. The reason for increased power to detect the substructure is that many nuisance parameters are typically excluded from the model when focusing on a subset of samples that are genetically distinct from the remaining data. Article **III** proposed hierarchical clustering strategy to derive biologically meaningful results in such a setting.

It is common that DNA sequences are aligned before classification. However, this is not practically possible in some cases. For example, when aligning large amounts (>100,000) of 16S rRNA sequences, it may takes

several weeks even using the fast alignment tools such as MUSCLE [26] and MAFFT [27]. In addition, the derived alignments can be very poor due to the diversity of sequences, which results in a large number of indels in the alignment. Article **I** proposed a 2-phase classification strategy to cluster the 16S rRNA sequences of different lengths. If two sequences are similar, then their 3-mer count vectors are also similar. Therefore if the 3-mer count vectors are very different, the two sequences are definitely different. Based on this idea, we could first cluster the 3-mer count vectors, which are automatically aligned. Then, we align the sequences within each cluster and continue with further clustering analysis based on the alignments.

Besides the unequal length problem of the DNA sequences, there are often missing data in the sequences. As we know, when a sequencing machine "reads" a base, it actually detects the intensities of the four bases. Sometimes the signal is so fuzzy that the sequence machine could not decide which base it is. Thus this base is marked as missing data, which is usually written as "-" or "N". Handling the missing data depends on specific applications. When handling MLST datasets, the housekeeping genes are usually of the same length and there are very few missing data, thus we can randomly replace the missing data with a random existing base at that site. When handling 16S rRNA datasets, the sequences are of different length and there are lots of missing data, therefore it is also possible to treat the missing data as a new base, i.e. now the alphabet changes to {'A','C','G','T','N'}, instead of considering them as purely missing information. Note that the marginal likelihood calculations, as shown in the previous sections, take missing data explicitly into account when comparing partitions, thus reducing the assignment certainty if a particular sample has large amounts of missing data.

Chapter 3

Reconstructing bacterial evolutionary history

This section will mainly discuss how to reconstruct the evolutionary history of closely related bacteria from whole genome data.

The foundation of molecular evolution is the molecular clock hypothesis, which assumes that DNA sequences mutate approximately at a constant rate. Under this assumption, the difference between DNA sequences of two sampled organisms is roughly proportional to the time of their divergence from the most recent common ancestor. Based on this idea, if we can calculate the distances between all considered samples, then we are able to draw conclusions about their evolutionary history. In some cases it is more appropriate to consider relaxed molecular clock models, where the substitution/mutation rate can vary across different evolutionary lineages [1].

However, mutation is not the only mechanism that bacteria use to evolve. Recombination, or horizontal gene transfer, also plays an important role in bacterial evolution. There are three generic mechanisms of recombination in bacteria: conjugation, transformation and transduction. In conjugation, the donor bacterium transmits DNA fragments to a recipient cell. In transformation, the bacteria reuse DNA fragments from their environment and incorporate them into their genomes. In transduction, a virus infecting bacteria called phage, fuses alien DNA fragments into the host genome. If the DNA sequences between the donor and recipient are very similar, then the recombination is called homologous recombination, otherwise it is called non-homologous recombination. The higher similarity between the DNA sequences is, the more efficient the recombination is. Thus homologous recombination occurs much more often than non-homologous recombination.

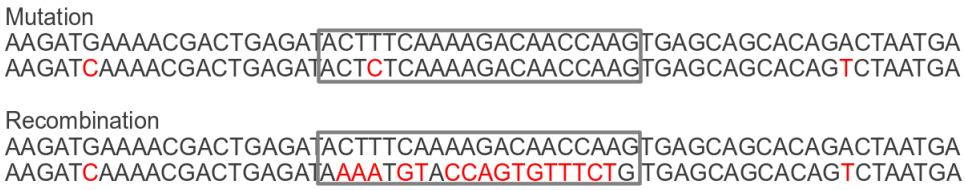

Figure 3.1: Non-homologous recombination. Short sequences are shown for two samples where mutations and non-homologous recombination are involved. The top case only involves mutations, which are indicated by the red letters. The lower case involves an additional non-homologous recombination event present in the second sample. The recombination regions are highlighted by the rectangles. As can be seen from this figure, non-homologugs recombination is capable of producing many SNPs within a short region due to the dissimilarity of the alien DNA fragment.

Since the DNA sequences between the donor and recipient are very similar, homologous recombination usually results in several single-nucleotide polymorphisms (SNPs) in the recipient DNA sequences, which can not be discriminated from mutations by looking at the DNA sequences. Non-homologous recombination, however, introduce considerable changes in the DNA sequences of the recipient, which might change the function of the cells dramatically, especially under evolutionary pressure such as antibiotic treatment or vaccination. In this thesis, we only focus on non-homologous recombination detection.

For simplicity, we refer recombination to non-homologous recombination in the following thesis. Recombination can substantially distort the estimated genetic distances between different samples in terms of DNA variation that is assumed to be clonally inherited. The genetic distance between two bacterial samples is usually estimated by the number of SNPs between sequences obtained from the two samples. A single recombination event from a distantly related bacterium can introduce many SNPs as illustrated in Figure 3.1, which makes the molecular clock hypothesis invalid. In order to retrieve the genetic distance related to point mutations, we have to eliminate such recombination fragments from bacterial genomes. Note, however, that the amount and locations of detectable recombinations within bacterial genomes are often of considerable interest themselves, since these quantities can be strongly correlated with phenotypic characteristics of the bacteria.

Housekeeping genes have been used to track the evolution of tens of

different human and animal pathogen species over the last 15 years. The advantage of these genes is that almost all mutations are synonymous due to high selection pressure, such that the resulting variation is neutral, although in some cases they can be linked to variant in nearby resistance loci. In addition, the housekeeping genes selected for these typing schemes are generally chosen to be far from each other in the chromosome, such that any single recombination events is very unlikely to change the allele at more than a single locus. However, given that a small number of housekeeping genes can only provide limited insight into the evolution of the bacterial populations, whole genome data has become an irreplaceable tool for such analyses.

This chapter is divided into three sections: sequence alignment, recombination detection and estimation of phylogenetic trees, which are the consecutive steps to derive the phylogeny from whole genome sequence data.

3.1 Sequence alignment

This section will first discuss different sequence alignment methods, then we explain how to apply these methods to different types of sequence data.

DNA sequences produced by any technology are usually of different lengths. If we want to compare DNA sequences, we have to in general align them so that the lengths are the same (although alignment-free methods of sequence comparison also do exist). The alignment makes the sequences (or parts of the sequences) as similar as possible so that we can quantify the difference between them.

Two generic types of alignments are the pairwise sequence alignment (PSA), which only aligns two sequences, and the other is multiple sequence alignment (MSA), which aligns more than two sequences. Compared with MSA, PSA is much easier and faster. However, in an MSA one considers all sequences as a whole, thus it is much more accurate when comparing many sequences. Figure 3.2 illustrates the difference between PSA and MSA.

PSA includes two general alignment methods: local alignment and global alignment. Local alignment, also known as Smith-Waterman algorithm, tries to find out the common similar subsequences of two input sequences. Global alignment, or Needleman-Wunsch algorithm, tries to match every base of the two input sequences such that they are as similar as possible. Figure 3.3 provides an example which shows the difference between a local alignment and a global alignment.

Different alignment methods have their own advantages and limitations. Therefore choosing the appropriate alignment methods, or combinations

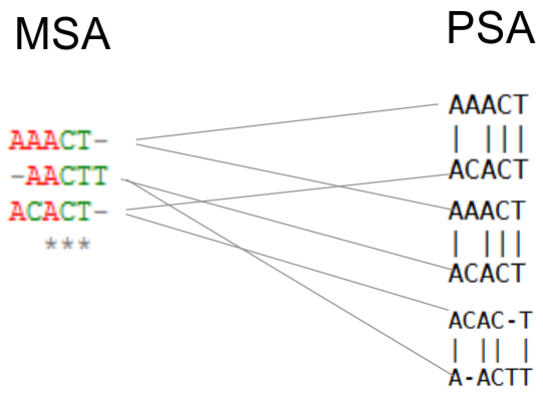

Figure 3.2: Pairwise sequence alignment and multiple sequence alignment. Three sequences on the left are aligned by MSA, each pair of which are aligned separately by PSA (global alignment) on the right. The '⋆' and '|' in the figure mean all sequences share the same base at this site. PSA does not provide a consistent alignment between all pairs, as can be seen from the inconsistent locations of the vertical bars '|'.

of alignment methods, is vital in deriving the alignments. Depending on the input data, which can vary a lot between different research projects, different alignment strategies are adopted. Here we focus on discussing how to handle the input data for article **IV** and **V**.

The whole genome data in article **IV** includes 62 *Escherichia coli* genomes, each of which represents a different strain. Since bacteria usually have only one circular chromosome, here the genome means one DNA sequence implicitly. When deriving the whole genome sequence, the genome is broken into short fragments. Then, each short fragment is sequenced using suitable technology, such as Illumina, Solid or 454 high-throughput platforms. After that the short sequences are assembled into the whole genome sequence. However, sometimes there are not enough short sequences due to various technical reasons, which makes the whole genome sequence not available. Therefore, the genome is replaced by **contig**s, which are non-overlapping sub-sequences of the whole genome sequence. In other words, the contigs are the longest sequences that can be assembled from the short sequences.

The *E. coli* genomes are relatively long (around 4.6 million bases) in terms of bacterial genome sizes. The sequences are not suitable to be aligned by general software such as MUSCLE [26] or MAFFT [27], which are designed to align large amounts of short sequences. Thus we use a

Figure 3.3: Local alignment and global alignment. The input sequences are 70 bp. The last 40 bp of the first sequence is identical to the first 40 bp of the second sequence, which are indicated by the rectangles. The vertical bars '|' in the middle of the alignments mean matches, i.e. two bases at this site of the sequences are identical. Local alignment perfectly captures the identical fragment shared by the input sequences, while global alignment provides a poor solution since it tries to match every base.

software Mugsy [28], which is specially designed to align relatively closely related genomes. The software also accepts contigs as input data if the whole genome sequence is not available.

Mugsy first performs local alignment between all pairs of genomes. Then, it constructs an alignment graph [29] using the result of previous step. After that it searches the so called "locally colinear blocks" (LCB) from the alignment graph. In the end, it calculates the multiple sequence alignment for each LCB. In simple terms, Mugsy first finds out similar and large segments between all the genomes. Then it selects those segments that keeps the same spatial order in the genomes. After that it derives the multiple sequence alignment for these parts. As a result, large re-arrangement of the genomes and large non-shared fragments will be eliminated by this method.

The whole genome data in article **V** include 480 gene sequences of 128 *Campylobacter jejuni* isolates. Also we have one reference genome sequence for *C. jejuni*, where the start and end positions of the aforementioned 480 genes are known. Note that the lengths of actual gene sequences may be different from that on the reference genome.

The strategy is straightforward, as shown in Figure 3.4. First we align the sequences of each gene, which is done by the *multialign* function in

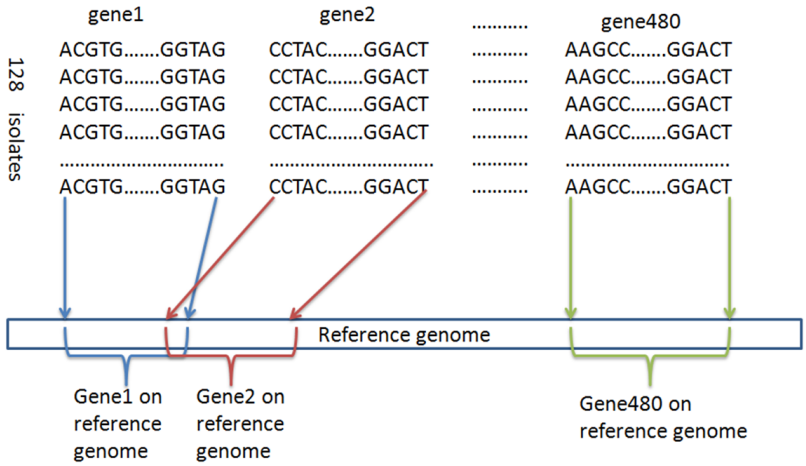

Figure 3.4: Multiple sequence alignment of 480 genes, where each row represents a *C. jejuni* isolate. Each aligned gene is then mapped to the reference genome.

MATLAB software. Then, we derive the consensus sequence for each gene, where we simply choose the majority base in each site of the alignment as the base of the consensus sequence. After that, we map the consensus sequences to the reference genome. In the end, we store the SNP sites information and their positions, which is the input for recombination detection in the next phase.

However, the mapping step is a bit tricky. Since the consensus sequences are usually slightly longer than the genes on the reference genome, we anchor each consensus sequence on the reference genome by the start position of the corresponding gene. A new problem arising from this operation is that two consecutive genes may overlap, i.e. the end of the previous gene falls into the latter gene. There also exist overlapping genes on the reference genome, but the overlapping regions are very short compared with the gene length. Therefore we cut off the tail of the previous gene that falls into the latter gene.

3.2 Recombination detection

In the previous section, we discussed how to derive reliable alignments under different scenarios. This section will discuss how to detect recombination events from the alignment. There exist various methods for re-

combination detection. Here we will not discuss the pros and cons of these methods. Instead, we focus on using the software BratNextGen [30], which is newly developed method for recombination detection from large population samples.

As can be seen from Figure 3.1, a recombination event can introduce many SNPs in a short region if the donor sequence is not highly similar to the recipient. BratNextGen models this problem using a hidden Markov model (HMM). For each sequence, BratNextGen tries to label each site into two hidden states: non-recombination state and recombination state. Based on different states, the sequence bases are emitted according to different probability distributions. Therefore we detect the recombination fragments in each sequence by picking out the sites which are marked as recombination.

BratNextGen assumes that all input sequences are closely related, i.e. all samples belong to an evolutionary lineage of a single species. However, using large-scale experimentation on several data sets, we have concluded that the approach works highly accurately also on multi-lineage data, provided that the sequences are not too dissimilar across lineages. Also it assumes that there are sufficiently many non-recombinant regions in the genomes to find the clonally evolving core genome.

BratNextGen uses a HMM to infer recombination events, but the state space of the hidden variables is not assumed to be known in advance. First, the method splits the input whole genome alignment into 5 Kb segments, from which it extracts the SNP sites information and their locations in the alignment. Then, it clusters each 5 Kb segment of the alignment into maximum K clusters using the software BAPS [31]. The biggest cluster in each segment is initially labeled as non-recombination state, while the others are all labelled as recombination states. The classification results facilitate the construction of transition matrix, which is a $K \times K$ matrix. The transition probabilities between two SNP sites are then specified by multiplying the transition matrix d times, where d is the sequence distance between the two SNP sites. BratNextGen then performs a MCMC-like iterative approach to optimize the initial labeling of hidden states and the parameters underlying the HMM.

The necessary information for BratNextGen to detect recombination events is provided by the SNP variants and their locations in the alignment. The SNP variants (alleles) are easy to extract from the alignment. Their locations, however, require careful thinking. Sometimes, the input alignment is not the alignment of whole genome sequences. For example, in the case of article **IV**, the alignment is derived from contigs, which means

Figure 3.5: Locations of 480 genes in article **V**, where white color indicates the gene regions and black means the missing parts.

many parts of the genomes are missing in the alignment. Should we take these missing parts into account when deciding the locations of the SNPs? The answer depends on the specific scenario.

In the case of article **IV**, the missing parts are very long compared with the alignment. If we take the missing parts into account, then it means we have not observed any mutations in these parts. It violates the hidden assumption by BratNextGen that the mutations are randomly scattered around the genomes, i.e. there should be at least some mutations in the missing parts instead of none. Since this could influence negatively the parameter estimation procedure in BratNextGen, we discard the missing parts, which means we assume implicitly that the mutation rate in the missing parts is the same as that in the alignment.

In the case of article **V**, each of the missing parts is relatively short compared with the whole alignment, as shown in Figure 3.5, which means the expected number of mutations is small. It also can be seen that the 480 genes are randomly scattered around the genome, as indicated by the white color. Hence here it is more reasonable to use the actual locations of the SNPs, i.e. keep the missing parts. Of course we implicitly assume no observed mutations in the missing parts, which might slightly affect the results. On the contrary, if we discard these missing parts, one side effect is that non-existing recombination fragments might be detected due to shortened distances between the SNPs.

3.3 Estimation of phylogenetic trees

After eliminating the recombinant fragments from the alignment, a main target of the analysis is to reconstruct the phylogenetic tree from the remaining alignment. The phylogenetic tree is the most universally accepted way to represent levels of relatedness among the studied samples.

There are various methods available for constructing a phylogenetic tree, among which the maximum likelihood approach [32] is the most popular. The standard maximum likelihood approach assumes independence between all sites in the alignment, i.e. each site mutates independently. It

also excludes recombination when modelling molecular evolution. It works by proposing a tree topology first, then calculating the likelihood of the data given the proposed topology. The process is repeated many times and the topology with the highest likelihood is then selected. After that, the branch lengths are optimized until the tree is fully specified.

It is obvious that the number of tree topologies grows exponentially as the number of samples increases. Thus it can take a long time to construct the maximum likelihood tree. We use the software FastTree [2] to calculate an approximately-maximum-likelihood tree from the alignment. FastTree runs much faster than other popular software such as PhyML [33] and RAxML [4], while it has been shown to produce accurate results for large and challenging data sets.

When running the FastTree software, we used the options "-gtr" and "-gamma". The "-gtr" option uses the general time reversible substitution parameters, which allows the most flexible modelling of the substitution matrix. The "-gamma" option allows site heterogeneous mutation rates, which means the different mutation rates over the sites are assumed to be distributed according to a gamma distribution.

After estimation of the tree, it is usually necessary to visualize and annotate the tree using available metadata about the samples. Popular software for this purpose is MEGA [3] and FigTree [34].

Chapter 4

Conclusions

Modern biology is almost entirely dependent on bioinformatics, since vast amounts of different biological data are waiting to be "digested" every day. In this thesis, we discussed how to analyze DNA sequence data, especially in the field of bacterial genomics.

We introduced three general Bayesian frameworks for analysis of DNA sequences: unsupervised classification, supervised classification and semi-supervised classification. One of the most significant advantages provided by these frameworks is that the user does not need to specify the exact number of clusters in the data. These approaches are also generic enough to be applied in many different contexts.

Based on the Bayesian unsupervised classification (clustering) framework, we proposed a novel method for classification large amounts of 16S rRNA sequences, which helps to estimate the composition of a sampled bacterial community. An important aspect of this method is that we avoid the huge burden of aligning all the sequences by first classification the 3-mer count vectors and only then continue clustering each derived cluster using an alignment. A minimum description length (MDL) criterion is also adopted to determine the final number of clusters, which helps to reduce sequencing errors and accurately discern closely related bacteria.

We also developed a method for simultaneously assigning several novel sample sequences into either existing or novel bacterial lineages, based on the semi-supervised classification framework. The most important contribution is to allow the sequences to form new clusters of their own. Also modelling the sequences as a second-order Markov chain increases the sensitivity of the classification the sequences.

Some special considerations are included article **III**, where we combine the spatial prior and second-order Markov chain ideas to provide a statistical tool for various application areas such as spatial infectious disease

epidemiology. Also we implemented a hierarchical clustering approach to allow for automated discovery of substructures in the data.

We discussed an biological application in which we try to reconstruct the bacterial evolutionary history in the presence of recombination events. It is necessary to remove the recombination fragments to appropriately estimate the degree of clonal relatedness among the samples. In articles **IV** and **V**, it was shown that important biological insights to the evolution of pathogen populations can be obtained by a combined application of some of the methods developed in this thesis. However, since the size of a typical bacterial genome data set is rapidly increasing, there is a considerable room to continuously develop further the methods to allow for less extensive computation times and to maintain the accuracy of the inferences.

References

[1] Drummond AJ, Suchard MA, Xie D and Rambaut A. (2012) Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology Evolution*, 29:1969-73.

[2] Price, M.N., Dehal, P.S., and Arkin, A.P. (2010) FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5:e9490.

[3] Tamura K, Peterson D, Peterson N, Stecher G, Nei M, and Kumar S (2011) MEGA5: Molecular Evolutionary Genetics Analysis using Maximum Likelihood, Evolutionary Distance, and Maximum Parsimony Methods. *Molecular Biology and Evolution*, 28:2731-2739.

[4] Stamatakis A., Ludwig T. and Meier H. (2005) RAxML-III: A Fast Program for Maximum Likelihood-based Inference of Large Phylogenetic Trees. *Bioinformatics* 21:456-463.

[5] `http://en.wikipedia.org/wiki/File:Biological_classification_L_Pengo_vflip.svg`, 17.06.2013, Wikipedia.

[6] Woese CR, Kandler O and Wheelis ML. (1990) Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences of the United States of America*, 87:4576-9.

[7] Alvarez-Ponce D, Lopez P, Bapteste E and McInerney JO. (2013) Gene similarity networks provide tools for understanding eukaryote origins and evolution. *Proceedings of the National Academy of Sciences of the United States of America*, 110:E1594-603.

[8] Pop M. (2012) We are what we eat: how the diet of infants affects their gut microbiome. *Genome Biology*, 13:152.

[9] Schwartz S, Friedberg I, Ivanov IV, Davidson LA, Goldsby JS, Dahl DB, Herman D, Wang M, Donovan SM, Chapkin RS (2012) A metagenomic study of diet-dependent interaction between gut microbiota and host in infants reveals differences in immune response. *Genome Biology*, 13:r32.

[10] Woese CR and Fox GE (1977) Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences of the United States of America*, 74:5088-90.

[11] `http://www.alimetrics.net/en/images/stories/content/M_00-06%20DNA%20based%20kuva%203.jpg`, 17.06.2013.

[12] Maiden MC, Bygraves JA, Feil E, Morelli G, Russell JE, Urwin R, Zhang Q, Zhou J, Zurth K, Caugant DA, Feavers IM, Achtman M, Spratt BG (1998) Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. *Proceedings of the National Academy of Sciences of the United States of America*, 95:3140-5.

[13] Kimura, Motoo (1983) *The neutral theory of molecular evolution*. New York, USA: Cambridge University Press.

[14] Croucher NJ, Harris SR, Fraser C, Quail MA, et al. (2011) Rapid pneumococcal evolution in response to clinical interventions. *Science*, 331:430-4.

[15] Waples RS, Gaggiotti O. (2006) What is a population? An empirical evaluation of some genetic methods for identifying the number of gene pools and their degree of connectivity. *Molecular Ecology* 15:1419-39.

[16] Bernardo JS, Smith AFM (1994) *Bayesian Theory*. Chichester, UK: Wiley.

[17] MacQueen, J. B. (1967) Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281-297.

[18] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1-38.

[19] D. Defays (1977) An efficient algorithm for a complete link method. *The computer Journal (British Computer Society)*, 20:364-366.

[20] Ronald L. Graham, Donald E. Knuth, Oren Patashnik (1988) *Concrete Mathematics*. Reading MA., USA: Addison Wesley, p. 244.

[21] Corander, J., Gyllenberg, M. and Koski, T. (2009) Bayesian unsupervised classification framework based on stochastic partitions of data and a parallel search strategy. *Advances in Data Analysis and Classification*, 3:3-24.

[22] Corander J and Marttinen P (2006) Bayesian identification of admixture events using multi-locus molecular markers. *Molecular Ecology*, 15:2833-2843.

[23] Corander J., Cui Y., Koski T. and Siren J. (2013) Have I Seen You Before ? Principles of Bayesian Predictive Classification Revisited. *Statistics and Computing*, 23:59-73.

[24] Corander, J., Siren, J. and Arjas, E. (2008) Bayesian spatial modeling of genetic population structure. *Computational Statistics*, 23:111-129.

[25] Lauritzen S. (1996) *Graphical models*. Oxford, UK: Oxford University Press.

[26] Edgar, R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32:1792-1797.

[27] Katoh, K., Misawa, K., Kuma, K. and Miyata, T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, 30:3059-3066.

[28] Angiuoli, SV. and Salzberg, SL. (2011) Mugsy: Fast multiple alignment of closely related whole genomes. *Bioinformatics* 27:334-4.

[29] Rausch, T. et al. (2008) Segment-based multiple sequence alignment. *Bioinformatics* 24:i187-i192.

[30] Marttinen, P., Hanage, W. P., Croucher, N. J., Connor, T. R., Harris, S. R., Bentley, S. D. and Corander, J. (2011) Detection of recombination events in bacterial genomes from large population samples. *Nucleic Acids Research*, 40:e6-e6.

[31] Cheng L, Connor TR, Siren J, Aanensen DM, Corander J. (2013) Hierarchical and Spatially Explicit Clustering of DNA Sequences with BAPS Software. *Molecular Biology and Evolution*, doi: 10.1093/molbev/mst028.

[32] Felsenstein J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368-76.

[33] Guindon S., Dufayard J.F., Lefort V., Anisimova M., Hordijk W., Gascuel O. (2010) New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology*, 59:307-21.

[34] `http://tree.bio.ed.ac.uk/software/figtree/`, 20.06.2013.