4-2009

Modeling HIV Drug Resistance

Mingfu Zhu
Clemson University, mzhu@clemson.edu

Modeling HIV Drug Resistance

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematics

---

by
Mingfu Zhu
Apr 22 2009

Accepted by:
Dr. Shuhong Gao, Co-chair
Dr. Chin-Fu Chen, Co-chair
Dr. Hiren Maharaj
Dr. Elena Dimitrova

Abstract

Despite the development of antiviral drugs and the optimization of therapies, the emergence of drug resistance remains one of the most challenging issues for successful treatments of HIV-infected patients. The availability of massive HIV drug resistance data provides us not only exciting opportunities for HIV research, but also the curse of high dimensionality.

We provide several statistical learning methods in this thesis to analyze sequence data from different perspectives. We propose a hierarchical random graph approach to identify possible covariation among residue-specific mutations. Viral progression pathways were inferred using an EM-like algorithm in literature, and we present a normalization method to improve the accuracy of parameter estimations. To predict the drug resistance from genotypic data, we also build a novel regression model utilizing the information from progression pathways. Finally, we introduce a computational approach to determine viral fitness, for which our initial computational results closely agree with experimental results.

Work on two other topics are presented in the Appendices. Latent class models find applications in several areas including social and biological sciences. Finding explicit maximum likelihood estimation has been elusive. We present a positive solution to a conjecture on a special latent class model proposed by Bernd Sturmfels from UC Berkeley.

Monomial ideals provide ubiquitous links between combinatorics and commutative algebra. Irreducible decomposition of monomial ideals is a basic computational problem and it finds applications in several areas. We present two algorithms for finding irreducible decomposition of monomial ideals.

Acknowledgments

Many people have helped me reach this point in my education. I am sincerely grateful to all the faculty in the Department of Mathematical Sciences at Clemson University who helped me in ways large and small. Of these, my greatest debt is to my advisor, Dr. Shuhong Gao. Because of his confidence in me, I was able to explore research directions with great freedom, while getting guidance from him at critical times with valuable suggestions, insights, and feedback.

I am fortunate to have had Dr. Chin-Fu Chen from the Department of Genetics and Biochemistry as my co-advisor. He advised me on research topics in biological sciences, and frequent discussion with him helped me to develop my interdisciplinary research ability. I am also grateful to other members of my committee, Dr. Hiren Maharaj and Dr. Elena Dimitrova. Dr. Maharaj taught me various courses and I have learned greatly from him. Dr. Dimitrova encouraged me to work on mathematical biology despite my background in pure mathematics.

I would like to express my appreciation to Dr. Joel Brawley and Dr. Judith Cottingham for their help and encouragement.

I would like to particularly thank to Prof. Bernd Sturmfels from UC Berkeley for his encouragement and support. I benefited from discussions with Prof. Niko Beerenwinkel from ETH-Zürich, who also laid out much frontier work in the topics of this thesis.

Thanks are due to my fellow students Yinhua Guan, Raymond Heindl, Jang-Woo Park, and many other graduate students in our department.

Special thank goes to my parents for their unceasing support. Finally, I would like to thank my wife Yuan Song for her unwavering and endless love.

Table of Contents

List of Tables

List of Figures

Chapter 1

HIV Biology

Many diseases spread from animals to humans. Recent examples include avian influenza (bird flu) and severe acute respiratory syndrome (SARS), which was tracked to civet cats. The human immunodeficiency virus (HIV), which crossed from primates into humans, is the most deadly pathogen so far.

Acquired immunodeficiency syndrome (AIDS) is caused by HIV, the first cases of the current epidemic probably occurred in the 1930s, and the disease spread rapidly in the 1970s. It was first publicly reported in the United States in 1981 and has since become a major worldwide epidemic [6]. It is estimated that 1.3 million persons in the United States are currently living with HIV infection, and more than 550,000 of them have already died [52]. Globally 40 million people are infected, the vast majority in developing countries, and numbers continue to rise. Today AIDS is the major killer of young adults.

This chapter aims to provide some background information of HIV biology. Many material and facts are cited from publicly available sources, such as Wikipedia [6] and AIDSinfo [3]. Also our presentation is far from complete. For example the following topics are not covered: the transmission/prevension of HIV, viral dynamics,

therapy optimization and HIV vaccine. We focus on self-contain material relevant for drug resistance, which serves for latter chapters.

1.1 Viral replication cycle

There are two main sub-types of virus: HIV-1 and HIV-2, the former being prevailing and the latter being harder to transmit and slower-acting. In this thesis the object of our study is HIV-1.

HIV is genetic material covered with a coat of protein molecules. It has fewer than 10 genes. It has ribonucleic acid (RNA) as its genetic material that contains the instructions specifying the biological development of cellular life. HIV also belongs to the family of viruses known as lentiviruses (a member of the retrovirus family), which means slow-acting.

Viruses are parasitic, and can only replicate by entering host cells. HIV primarily infects vital cells in the human immune system such as helper T cells (specifically $CD4^+$ T cells), macrophages, and dendritic cells. Within these cells, it produces more virus particles by converting viral RNA into DNA in the cell and then making many RNA copies. HIV replication cycle consists of the following main steps [3]. A visualization of the cycle can be seen in Figure 1.1.

Binding and Fusion: HIV begins its life cycle when it binds to a $CD4^+$ receptor and one of two co-receptors on the surface of a $CD4^+$ T-lymphocyte. The virus then fuses with the host cell. After fusion, the HIV RNA and various enzymes, including reverse transcriptase, integrase, ribonuclease and protease, are injected into the cell.

Reverse Transcription: An HIV enzyme called **reverse transcriptase** (RT) converts the single-stranded HIV RNA to double-stranded HIV DNA. The con-

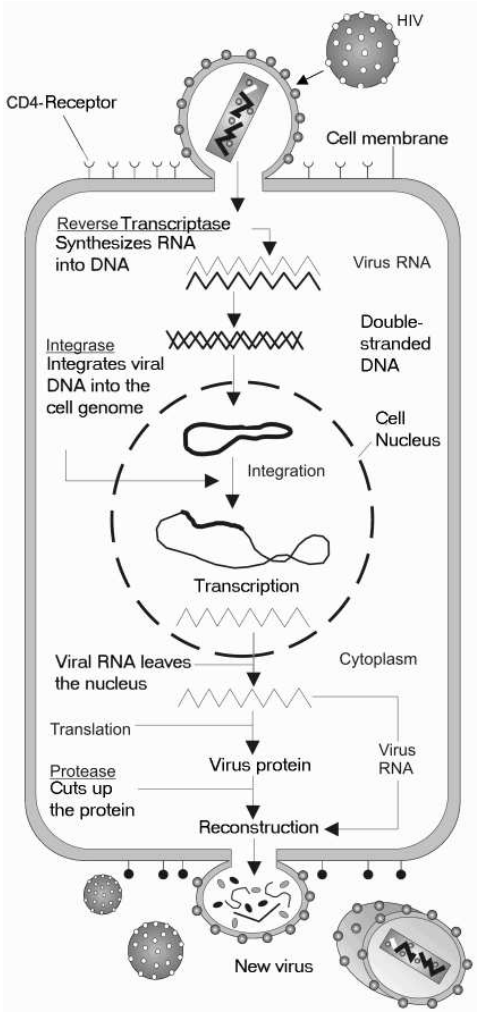


Figure 1.1: The HIV replication cycle [6].

version from RNA to DNA is done through reverse transcriptase. The switch from RNA to DNA and back to RNA is significant and makes combating HIV difficult. Each time it occurs there is a possibility of errors and the virus mutating. This is made more likely because reverse transcriptase lacks the normal proofreading that occurs with DNA replication.

Integration: The newly formed HIV DNA enters the host cell's nucleus, where an HIV enzyme called integrase "hides" the HIV DNA within the host cell's

own DNA. The integrated HIV DNA is called provirus. The provirus may remain inactive for several years, producing few or no new copies of HIV.

Transcription: When the host cell receives a signal to become active, the provirus uses a host enzyme called RNA polymerase to create copies of the HIV genomic material, as well as shorter strands of RNA called messenger RNA (mRNA). The mRNA is used as a blueprint to make long chains of HIV proteins.

Assembly: An HIV enzyme called **protease** (PR) cuts the long chains of HIV proteins into smaller individual proteins. As the smaller HIV proteins come together with copies of HIV's RNA genetic material, a new virus particle is assembled.

Budding: The newly assembled virus pushes out ("buds") from the host cell. During budding, the new virus steals part of the cell's outer envelope. This envelope, which acts as a covering, is studded with protein/sugar combinations called HIV glycoproteins. These HIV glycoproteins are necessary for the virus to bind $CD4^+$ and co-receptors. The new copies of HIV particles break out of the cell, destroying it and can now move on to infect other cells.

HIV infection leads to low levels of $CD4^+$ T cells through three main mechanisms: firstly, direct viral killing of infected cells; secondly, increased rates of apoptosis in infected cells; and thirdly, killing of infected $CD4^+$ T cells by CD8 cytotoxic lymphocytes that recognize infected cells. When $CD4^+$ T cell numbers decline below a critical level, cell-mediated immunity is lost, and the body becomes progressively more susceptible to opportunistic infections.

1.2 Anti-viral drugs and drug resistance

After nearly 30 years, HIV infection remains a significant cause of morbidity and mortality. Major advances in HIV treatment have revolutionized patient care and

Table 1.1: FDA Approved Antiretroviral drugs.

Generic Name	Brand & Other Names	FDA Approval Date
Non-nucleoside Reverse Transcriptase Inhibitors (NNRTIs)		
Delavirdine	Rescriptor, DLV	April 4, 1997
Efavirenz	Sustiva, EFV	Sept. 17, 1998
Etravirine	Intelence, Celsentri,	Jan. 18, 2008
Nevirapine	TMC125, ETR	June 21, 1996
Nucleoside Reverse Transcriptase Inhibitors (NRTIs)		
Abacavir	Ziagen, ABC	Dec. 17, 1998
Didanosine	Videx, ddI, Videx EC	Oct. 9, 1991
Emtricitabine	Emtriva, FTC, Coviracil	July 2, 2003
Lamivudine	Epivir, 3TC	Nov. 17, 1995
Stavudine	Zerit, d4T	June 24, 1994
Tenofovir	DFViread, TDF	Oct. 26, 2001
Zidovudine	Retrovir, AZT, ZDV	March 19, 1987
Protease Inhibitors (PIs)		
Amprenavir	Agenerase, APV	April 15, 1999
Atazanavir	Reyataz, ATV	June 20, 2003
Darunavir	Prezista, TMC114, DRV	June 23, 2006
Fosamprenavir	Lexiva, FPV	Oct. 20, 2003
Indinavir	Crixivan, IDV	March 13, 1996
Lopinavir, Ritonavir	Kaletra, LPV/r	Sept. 15, 2000
Nelfinavir	Viracept, NFV	March 14, 1997
Ritonavir	Norvir, RTV	March 1, 1996
Saquinavir	Invirase, SQV	Dec. 6, 1995
Tipranavir	Aptivus, TPV	June 22, 2005
Entry/Fusion Inhibitors		
Enfuvirtide	Fuzeon, T-20 Hoffmann-La	March 13, 2003
Maraviroc	Selzentry, MVC	Aug. 6, 2007
Integrase Inhibitors		
Raltegravir	Isentress	Oct. 12, 2007

prolonged survival, with the result that HIV infection can now be effectively managed as a chronic disease.

1.2.1 Antiretroviral drugs and drug resistance

Anti-viral drugs are 'smart' drugs that target at certain step of viral replication cycle to make an obstacle to its replication. The first drug, AZT (zidovudine), became available in 1987 but effective treatment was achieved in 1996/7 with the advent of highly active antiretroviral therapy (HAART), initially based on two nucleoside reverse transcriptase inhibitors (NRTIs) and a protease inhibitor (PI). The first successful regimens were relatively toxic, with a high pill burden and complexity but the impact on morbidity and mortality was dramatic.

Since then, nearly 25 antiretroviral drugs (ARV) have been licensed by the US Food and Drug Administration (FDA) for the treatment of HIV-1 [48]: nine nucleoside reverse transcriptase inhibitors (NRTI), four nonnucleoside reverse transcriptase inhibitors (NNRTI), nine protease inhibitors (PI), one fusion inhibitor, one CCR5 inhibitor, and one integrase inhibitor. The first CCR5 and integrase inhibitors were approved in 2007, increasing the number of ARV classes from four to six.

1.2.2 Emergence of drug resistance

With HIV, drug resistance is caused by changes in the virus's genetic structure [1, 4]. When HIV reproduces, it makes new copies of its genetic material. But HIV does not always make perfect copies of itself since it does not contain the proteins needed to correct the mistakes it makes at the reverse transcription step. Lots of small mistakes can be made. These mistakes are called mutations and will create a mutant strain. They occur naturally and make the new virus just a little bit different from

the old virus. Mutations are very common in HIV. This is because HIV replicates at an extremely rapid rate. In a person with HIV who is not taking ARV drugs, billions of copies of HIV are made every day. Mutations occur by chance. The more virus that is made, the more likely it is that a 'mutant' strain will appear.

Mutations occur randomly, on a daily basis, but many are harmless. In fact, most mutations actually put HIV at a disadvantage. They reduce the virus's 'fitness' and slow its ability to infect $CD4^+$ cells in the body. However, a number of mutations can give HIV a survival advantage when ARV drugs are used, because these mutations can block drugs from working against the HIV enzymes they are designed to target. These are the mutations we are concerned with when we talk about drug resistance.

The **wildtype** virus refers to the most common strain of virus that HIV-infected people have as the predominant strain when they are taking no ARV drugs. Anything different from the wildtype is considered a mutation.

Sometimes a mutant version of HIV is resistant to more than one drug. When this happens, the drugs are called **cross-resistant**. For example, most HIV that is resistant to nevirapine (Viramune) is also resistant to efavirenz (Sustiva). This means that nevirapine and efavirenz are cross-resistant.

1.3 Resistance testing

A **phenotype** is any observable characteristic or trait of an organism: such as its morphology, development, biochemical or physiological properties, or behavior. Phenotypes result from the expression of an organism's genes as well as the influence of environmental factors and possible interactions between the two. The **genotype** of an organism is the inherited instructions it carries within its genetic code. It represents an organism's exact genetic makeup, and contains an organism's full hereditary

information, even if not expressed.

There are simple blood tests that can detect HIV resistance and help clinicians identify which HIV medications will work best against the mutant virus. Two main types of resistance testing are now available: phenotype testing and genotype testing [2]. Phenotype testing looks at drug resistance in vitro, measuring the extent to which specific drugs or drug combinations inhibit viral replication in cultured cells. Genotype testing analyzes the sequence information of the virus, inferring drug resistance from mutations associated with resistance.

1.3.1 Phenotypic testing

Phenotypic analysis determines the degree to which a drug inhibits replication of the patient's virus. The concentration of the drug required to inhibit the virus replication by 50% is called the IC50. One usually measures IC50 for a patient virus sample, in comparison to IC50 required for a wildtype reference strain. Then the **fold change** is defined as

$$\text{FC} = \frac{\text{IC50 of sample}}{\text{IC50 of wildtype}} \tag{1.1}$$

The clinical cut-off refers to the FC of virus susceptibility above which the drug has less activity in vivo. Often there are two cut-offs. Virus with a fold-change in susceptibility below the lower cut-off is fully susceptible, while virus with a FC in susceptibility above the higher cut-off is very unlikely to be inhibited at all. Virus with a fold-change between the cut-offs is partially susceptible.

Advantages of phenotypic testing for resistance include relatively easy interpretation, the ability to provide quantitative information on the degree of resistance, the ability to assess interactions among resistance mutations on overall resistance and susceptibility, and the fact that it does not require an understanding of genotypic cor-

relation of resistance.

One of the limitations of the use of phenotypic resistance tests is that clinical cut-offs have not been clearly established for some of the NRTIs or PIs.

1.3.2 Genotypic testing

In genotype testing, the reverse transcriptase and protease samples are sequenced. The genotype assay returns either the RNA sequence data or a list of the differences from the wildtype sequence.

HIV genotypic assays detect known mutations associated with drug resistance. Genotypic assays are commonly used for detection of ARV drug resistance because of lesser cost as compared to phenotypic assays and rapid turnaround time. The sensitivity of most of these assays range between 100-1000 viral RNA copies/ml. These assays involve detection of mutations in the HIV-1 genome, in the regions that are targeted by the different ARV drugs, namely PR and RT.

1.4 Data representation

Throughout this article we consider genetic variation in the viral pathogen, rather than in the host. Specific mutations between PR positions 10-93 and RT positions 41-236 are associated with resistance to PIs and RTIs, respectively.

1.4.1 Stanford HIV Database (HIVdb)

The HIVdb is a public web-based interpreting system at Stanford University that stores sequences of the PR and RT genes (http://hivdb.stanford.edu/). The web-based system interprets the user-entered mutations to infer levels of resistance

to the available drug classes: NRTIs, NNRTIs and PIs.

The database is designed to store, analyze and make available the diverse forms of data underlying drug resistance knowledge to the broad community of researchers and clinicians studying HIV drug resistance and using HIV drug resistance tests. It also provides a publicly available online resource to help those performing HIV drug resistance surveillance, interpreting HIV drug resistance tests, and developing new ARV drugs.

As of March 2009, HIVdb contains 51402 RT sequences, 52046 PR sequences, 3215 Integrase sequences from more than 80,000 distinct virus isolations obtained from nearly 40,000 individuals. 98%-99% of the viruses are human HIV-1 isolates; 1%-2% of viruses are human HIV-2 isolates or other non-human primate lentiviruses (NHPL). The HIVdb also contains 1691198 mutations including 1036762 RT mutations, 590885 Protease mutations and 63551 Integrase mutations.

1.4.2 Data representation

Resistance mutations are described as a letter corresponding to wildtype virus amino acid preceding a number, which shows the position of the relevant codon in the HIV-1 genome PR or RT. The succeeding letter describes the amino acid in the mutant virus. Without cause of confusion the first letter can be omitted.

We consider HIV-1 drug resistance to the nucleoside reverse transcriptase inhibitor zidovudine as a test case. The most common RT mutations ("classical zidovudine mutations") that develop under zidovudine therapy are M41L, D67N, K70R, L210W, T215F/Y, and K219E/Q. Other mutations are less frequent and typically occur under prolonged combination therapies containing two or more nucleoside RT inhibitors.

In a typical setting, N samples are collected and for each sample the mutations of genotypes are observed. Suppose there are ℓ possible mutations, say $m_1, \ldots, m_\ell$. For each sample, its genotype can be characterized as the set of mutations that occur in this sample. We use a binary vector of length ℓ to represent this genotype, say $X_i = (x_{i1}, \ldots, x_{i\ell})$, where

$$x_{ij} = \begin{cases} 1 & \text{if mutation } j \text{ occurs in sample } X_i, \\ 0 & \text{otherwise.} \end{cases}$$

So each genotype can be either represented by a set of mutations, or by a vector of indicator variables (it is also called a **pattern** in the literature [12]). We will use both representations below. For example if the nonzero positions of X are $j_1, \ldots, j_{\ell_i}$, then X can also be represented by the set $\{m_{j_1}, \ldots, m_{j_{\ell_i}}\}$. We represent the censored wildtype genotype by a zero vector $(0, \ldots, 0)$.

Let $\mathcal{T} = (V, E, WT)$ be a tree where $V = \{m_1, \ldots, m_\ell\}$ (each vertex represents a mutation) and WT is the root representing wildtype. A subtree $\mathcal{T}' = (V', E', WT)$ is said to be *root closed* if all the vertices lie on the path from the root WT to a vertex in V' are also in V'. A genotype X is **compatible** with $\mathcal{T}$ if there is a root closed subtree of $\mathcal{T}$ whose vertices are exactly the mutations occuring in X, and this subtree is called an *induced subtree* of X. In [12] it is also said that X can be **generated** by $\mathcal{T}$. We denote by $\Omega = \{0, 1\}^\ell$ the set of all possible patterns of length ℓ.

Throughout this thesis we use Table 1.2 as our testing data, which was used in [12] from an old version of Stanford HIVDB in 2003. These data were derived from previously untreated patients under AZT mono-therapy. The sample size is $N = 364$. We consider the occurrence of the following six mutants in TAMs only: 41L, 67N, 70R, 210W, 215FY, and 219EQ. There are 35 mutation patterns (maximum possible

Table 1.2: 364 samples.

Pattern	41L	67N	70R	210W	215Y	219Q	Frequency	Percentage
1	0	0	0	0	1	1	1	0.275
2	0	1	0	0	0	0	1	0.275
3	0	1	0	0	0	1	1	0.275
4	0	1	0	0	1	0	1	0.275
5	0	1	0	0	1	1	1	0.275
6	1	0	0	1	0	0	1	0.275
7	1	0	1	0	0	0	1	0.275
8	1	0	1	0	1	1	1	0.275
9	1	1	0	0	0	0	1	0.275
10	1	1	1	0	0	0	1	0.275
11	1	1	1	0	0	1	1	0.275
12	0	0	0	0	0	1	2	0.549
13	0	1	0	1	1	0	2	0.549
14	1	0	0	0	0	0	2	0.549
15	1	1	0	1	1	1	2	0.549
16	0	0	0	1	1	0	3	0.824
17	0	0	1	1	1	1	3	0.824
18	0	1	1	0	1	0	3	0.824
19	1	1	0	0	1	0	3	0.824
20	1	1	1	0	1	0	4	1.10
21	1	1	1	1	1	0	4	1.10
22	0	1	1	0	0	0	5	1.37
23	1	1	0	1	1	0	5	1.37
24	0	0	1	0	0	1	6	1.65
25	0	0	1	0	1	0	7	1.92
26	0	0	1	0	1	1	7	1.92
27	1	0	1	0	1	0	7	1.92
28	0	1	1	0	1	1	10	2.75
29	1	1	1	0	1	1	11	3.02
30	0	0	0	0	1	0	14	3.85
31	0	1	1	0	0	1	23	6.32
32	1	0	0	1	1	0	25	6.87
33	1	0	0	0	1	0	30	8.24
34	0	0	1	0	0	0	60	16.5
35	0	0	0	0	0	0	115	31.6
Total							364	100

$2^6 = 64$) consisting of these six mutants. Two mutations are present at codon 215 (Y or F). However we don't distinguish 215F from 215Y because 215Y is prevailing. Same reason applies for 219E and 219Q.

1.5 Computational problems and outline

The availability of massive HIV drug resistance data provides us exciting opportunities for HIV research. It also challenges us with its high-dimensionality. For example the number of major mutations in RT is estimated to be 30∼40 for RTI resistance, and around 20 for PI resistance. Genotype drug resistance testing is much faster and cheaper, but sequence data provide only indirect evidence of resistance and require a more detailed interpretation [37]. Mining high-dimensional data is an urgent problem of great practical importance.

Several statistical machine learning techniques are utilized to analyze the sequence data in this thesis. We are interested in the following problems.

1. To identify possible covariation among residue-specific mutations.
2. To characterize viral progression pathways.
3. To predict drug resistance from viral genetic sequence.
4. To evaluate viral fitness from viral genetic sequence.

We begin with covariation study among mutations at different residues in Chapter 2. The hierarchical random graph has been introduced for network inference, and an optimal graph can be derived by a Markov chain Monter Carlo method. We generalize this approach to expose the structural relation of the mutations by carefully defined parameters.

The accumulation of resistance-associated mutations along multiple evolutionary pathways is analyzed in Chapter 3. The tree inference algorithms were developed in literature. We reveal the bias existing in previous algorithms, and then propose our method to improve the accuracy of parameter estimations.

Predicting an organism's phenotype from its genotype is a fundamental problem of biology. In Chapter 4 we propose a method to build a regression model from matched genotype-phenotype pairs. The novelty of our method is that we use the evolutionary pathways to determine coefficients for the terms in the regression function.

In Chapter 5 we approach the task of evaluating the relative viral fitness of certain mutation patterns via computational methods. Our initial computational results closely agree with experimental results.

Finally, several future research problems are presented in Chapter 6.

Chapter 2

Covariation Analysis of Mutations via Hierarchical Random Graphs

High covariation between mutations at different positions (residues) result from a similar selective pressure, a functional interaction, or structural dependency.

Hierarchical random graphs have been introduced by Clauset *et al.* for network inference and missing links prediction [16]. In an optimal hierarchical random graph, closely related pairs of vertices have lower common ancestors in the tree than those of more distantly related pairs. The optimal hierarchical random graph can be derived by a maximum likelihood approach combing a Markov chain Monte Carlo algorithm.

We consider the mutations as discrete random variables. By careful definition of distance metric, we use a hierarchical random graph to study covariation between position-specific mutations.

2.1 Introduction

A strong correlation between position-specific mutations is interpreted as evidence of functional interactions or structural dependency under substantial selection pressure. Covariation analysis of mutations in different regions of HIV genome has been widely used to infer functional interactions between different sites in a protein, and it has identified a number of correlated mutation pairs, many of which have known biological interactions [25, 26, 31, 35, 50, 53]. Therefore, studying covariation of amino acid mutations in HIV will improve our understanding of HIV drug resistance as well as help vaccine design [35, 53].

Mutations associated with reduced drug sensitivity do not accumulate independently from each other. For example, in RTI drug resistance mutations, there are two most prominent complexes associated with NRTI resistance, the thymidine analogue mutations (TAMs), groups 1 and 2, consisting of mutations M41L/L210W/T215Y and K70R/K219Q/D67N, respectively.

In protease, drug resistance associated mutations are often classified as either primary or secondary. Primary mutations appear early in the evolution of inhibitor resistance. No single mutation provides high-level resistance, which is generally attained only following the acquisition of one or more primary mutations in combination with one or more secondary mutations. Secondary mutations by themselves can be associated with low-level resistance and are common polymorphisms in untreated populations. These latter mutations may compensate for a reduction in fitness (and enhance resistance) or may further reduce drug sensitivity but with an additional fitness loss [31].

Covariation between amino acid positions in protease can be caused by (i) the specificity of the compensatory effect of one mutation for another; (ii) and additive

reduction in sensitivity to a particular drug by two or more mutations in the presence of drug selection; or (iii) the preexistence of a mutation creating a favorable context for the emergence of a specific resistance mutation at another position [31].

Hierarchical random graphs have been recently introduced by Clauset *et al.* for network inference and missing links prediction [16](a paper published in *Nature*). The vertices of the graphs are found to correspond to known functional units, such as ecological niches in food webs, modules in biochemical networks (protein interaction networks, metabolic networks, or genetic regulatory networks), or communities in social networks. Hierarchical structural study provides more information than simple clustering, especially on hierarchy of the vertices. In an optimal hierarchical random graph, closely related pairs of vertices have lower common ancestors in the tree than those of more distantly related pairs. The optimal hierarchical random graph can be derived by a maximum likelihood approach combing a Markov chain Monte Carlo algorithm.

The algorithm in [16] is to find the hierarchical structure for an unweighted and incomplete graph. The interaction between two vertices in a network either exists or does not. In fact, we can consider unweighted graphs as a special case of weighted complete graphs with weight 0 for disconnected edges and weight 1 for connected edges. With this idea we want to generalize the algorithm to work on weighted complete graphs.

We consider mutations as random variables, and build a weighted complete graph with each vertex being a variable. We carefully define the distance of a pair of vertices by their Jaccad similarity coefficient. A modified version of the algorithm in [16] is then used to study the hierarchy of the correlation of the position-specific mutations.

2.2 Previous approaches

Several computational methods are proposed in this topic from different perspectives, see for example [31] for the pairwise covariation in protease, [38] for spectral clustering in protease, and [44] in protease and reverse transcriptase.

2.2.1 Mutual information analysis

In [31], a total of 31 positions within HIV-1 protease was analyzed. The protease sequences were retrieved from the Stanford HIV Drug Resistance database.

Mutual information (M) was calculated for all possible 465 pairwise combinations of these 31 variable positions. M is the sum of the Shannon entropies at each of two positions minus the joint entropy of the two positions:

$$M(X,Y) = H(X) + H(Y) - H(X,Y)$$

where

$$H(X) = -\sum_{i=1}^{m} p(X_i) \log p(X_i)$$

and

$$H(X,Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(X_i, X_j) \log p(X_i, X_j)$$

Here m and n are the numbers of different amino acids represented at positions X and Y, respectively. $p(X_i)$ is the relative frequency of the amino acid i at position X, and $p(X_i, Y_j)$ is the relative frequency of the combination of amino acid pair X_i and Y_j. The intuitive idea is that if the amino acids at the two positions vary independently they will form many combinations and $H(X,Y)$ will be large, reducing the value of $M(X,Y)$. On the other hand if the positions covary, there will be fewer

combinations, and since $H(X,Y)$ is small, $M(X,Y)$ will remain relatively large.

Significance of pairwise association was assessed using permutation tests. Strong statistical support for linkage in the treated data set was seen for 32 pairs and 9 pairs of sites in the untreated data set. Most associations were positive, although negative associations were seen for five pairs of interactions. Structural proximity suggests that numerous pairs may interact within a local environment.

In addition to the mutual information analysis, Liu *et al.* [38] developed spectral partitioning methods for efficient analysis of the matrices of pairwise mutual information. The residues were partitioned into clusters, such that the similarity is high among the nodes within a cluster and low across different clusters.

2.2.2 Jaccard similarity coefficients

The Jaccard similarity coefficient (J) was employed to assess covariation among PR and RT mutations in [44]. For a given pair of mutations X and Y, the Jaccard similarity coefficient is calculated as

$$J = \frac{N_{XY}}{N_{XY} + N_{X0} + N_{0Y}}$$

where N_{XY} represents the number of sequences containing X and Y, N_{X0} represents the number of sequences containing X but not Y, and N_{0Y} represents the number of sequences containing Y but not X. This coefficient represents the probability of both mutations occurring together when either mutation occurs.

To test whether observed Jaccard similarity coefficients were statistically significant, the expected value of the Jaccard similarity coefficients (J_{RAND}) and its standard error (J_{SE}) were also calculated for each pair of mutations by assuming two mutations (X and Y) occur independently. J_{RAND} was calculated as the mean Jac-

card similarity coefficient using 2,000 random rearrangements of the X or Y vector (containing 0 or 1 for presence or absence of a mutation, respectively). J_{SE} was calculated using a jackknifed procedure, which removes one sequence at a time, repeatedly for each sequence. The standardized score Z, $Z = \frac{J-J_{RAND}}{J_{SE}}$, indicates a significant positive association (if $Z > 2.56$) or a significant negative association (if $Z < -2.56$) at an unadjusted p-value< 0.01.

A complete list of 327 statistically significant RT mutation pairs and 161 protease mutation pairs was derived, where several previously reported patterns of amino acid covariation were confirmed and many new patterns of covariation were identified. Multidimensional scaling further organized many of the correlations into clusters of co-occurring mutations.

2.3 Hierarchical structure analysis

Firstly we construct a graph G with ℓ vertices corresponding to ℓ position-specific mutations. G is a weighted complete graph. For an edge e with end points u and v, we associate a weight w_{uv}. There are many metrics to define this weight. Here we use Jaccard similarity coefficient, that is

$$J(u,v) = \frac{p(v|u)p(u|v)}{p(v|u) + p(u|v) - p(v|u)p(u|v)}, \tag{2.1}$$

where $p(v|u)$ is the conditional probability of the occurrence of v given the occurrence of u.

Table 2.1 lists all the pairwise Jaccard similarity coefficients of the six mutations in Table 1.2.

For two sets of mutations we define the Jaccard similarity coefficient to be the

Table 2.1: Pairwise Jaccard similarity coefficients.

w_{uv}	41L	67N	70R	210W	215Y	219Q
41L	1	0.219	0.135	0.346	0.609	0.098
67N	0.219	1	0.363	0.117	0.260	0.495
70R	0.135	0.363	1	0.036	0.237	0.385
210W	0.346	0.117	0.036	1	0.303	0.046
215Y	0.609	0.260	0.237	0.303	1	0.203
219Q	0.098	0.495	0.385	0.046	0.203	1

arithmetic mean of the pairwise Jaccard similarity coefficients, that is, for two sets S_1 and S_2,

$$J(S_1, S_2) = \frac{\sum_{u \in S_1, v \in S_2} J(u, v)}{|S_1| \cdot |S_2|}.$$

Next we define a hierarchical random graph from a weighted complete graph G analogous to the definition for unweighted graphs in [16]. It is defined through a *dendrogram* D, which is a binary tree with ℓ leaves corresponding to the vertices of G. Each of the $\ell - 1$ internal vertices of D corresponds to a group of vertices descended from it. For each internal vertex r, let L_r and R_r, respectively, be the sets of leaves in the left and right subtrees rooted at r. We associate r with a weight w_r defined as the Jaccard similarity coefficient between L_r and R_r,

$$w_r = J(L_r, R_r) = \frac{\sum_{u \in L_r, v \in R_r} J(u, v)}{|L_r||R_r|} \tag{2.2}$$

where $|\cdot|$ is the size of a set.

So for any two vertices u, v of G, their Jaccard similarity coefficient in D becomes w_r where r is their lowest common ancestor in D. The dendrogram D along with the set of probabilities $\{p_r : 1 \leq r \leq \ell - 1\}$ then defines a *hierarchical random*

graph.

We need to build an objective function for evaluating a hierarchical random graph for given data. Since Jaccard similarity coefficient can be interpreted as the probability of both mutations occurring together when either mutation occurs, we can follow the objective function in [16] by using likelihood function. The likelihood of the hierarchical random graph from the dendrogram D is

$$\begin{aligned} \mathcal{L}(D) &= \prod_{r \in D} w_r^{w_r |L_r||R_r|} (1 - w_r)^{|L_r||R_r| - w_r |L_r||R_r|} \\ &= \prod_{r \in D} \left[w_r^{w_r} (1 - w_r)^{1 - w_r} \right]^{|L_r||R_r|} \end{aligned} \tag{2.3}$$

with the convention that $0^0 = 1$. For a given G, the dendrogram with maximum likelihood is called an *optimal* hierarchical random graph.

Taking the logarithm of the likelihood function (2.3) yields

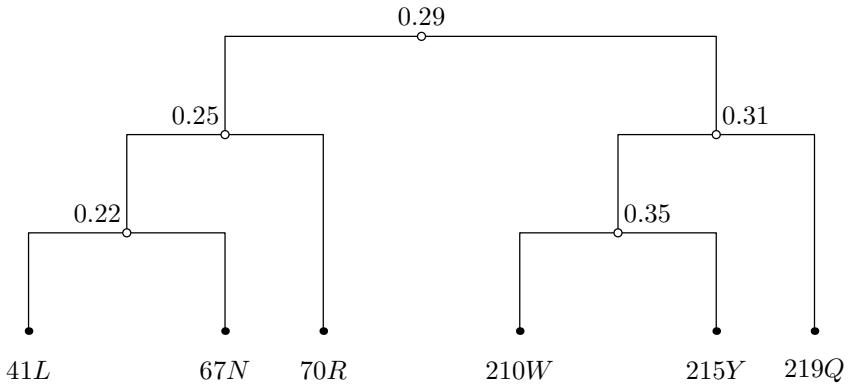
$$\log \mathcal{L}(D) = - \sum_{r \in D} |L_r||R_r| h(w_r), \tag{2.4}$$

where $h(p) = -p \log p - (1 - p) \log(1 - p)$ is the Shannon entropy function. Note that each term $-|L_r||R_r|h(w_r)$ is maximized when w_r is close to 0 or to 1, i.e., when the entropy is minimized. In other words, high-likelihood dendrograms are those that partition the vertices into groups between which connections are either very common or very rare.

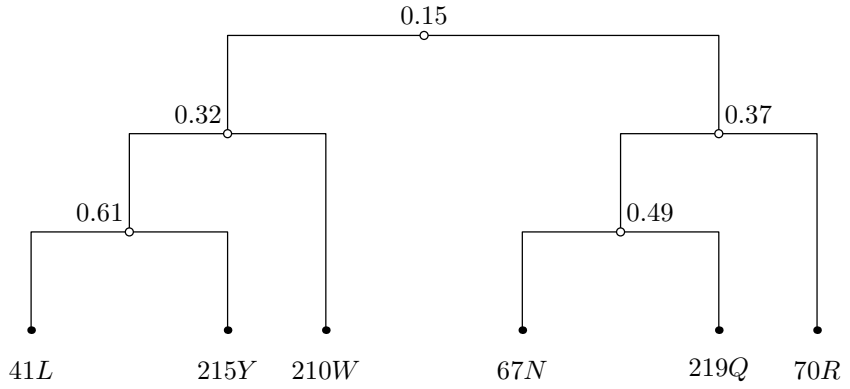
The mean squared error (MSE) can be considered as an alternative objective function. It can be calculated by

$$\mathrm{MSE}(D, \{w_r\}) = \frac{\sum_{r \in D} \sum_{u \in L_r, v \in R_r} (J(u, v) - w_r)^2}{\ell - 1}. \tag{2.5}$$

A dendrogram with minimum MSE is called optimal for a given G.



(a) $\mathcal{L} = 1.29 \times 10^{-4}$, MSE=$3.06 \times 10^{-2}$



(b) $\mathcal{L} = 4.36 \times 10^{-4}$, MSE=$6.39 \times 10^{-3}$

Figure 2.1: A small example of a hierarchical random graph.

A small example of a hierarchical random graph consisting of 6 vertices is given in Figure 2.1, where the dendrogram in Figure 2.1(b) has larger likelihood than in Figure 2.1(a). In fact (b) achieves global maximum likelihood and least MSE.

When the number of vertices ℓ becomes large, the number of the candidate trees grows dramatically large to the order of $\ell!$ and finding an optimal tree becomes difficult. In [16], a Markov chain Monte Carlo method is proposed. To create the

Markov chain they pick a set of transitions between possible dendrograms. The transitions they use consist of rearrangements of subtrees of the dendrogram. At each step of their Markov chain, choose an internal node r (other than the root) uniformly at random and switch a subtree descended from r with a subtree descended from a sibling of r. The result is a new dendrogram D'. They accept the transition from D to a new dendrogram D' if $\mathcal{L}(D') \geq \mathcal{L}(D)$; otherwise they accept the transition with probability $\frac{\mathcal{L}(D')}{\mathcal{L}(D)}$. The Markov chain appears to converge relatively quickly, with the likelihood reaching equilibrium after roughly $O(\ell^2)$ steps. Many dendrograms can be found with roughly competitive likelihoods. In this case they use some resampling technique to find the optimal candidate.

Their algorithm can be applied to our study immediately in our new settings, for both likelihood function in (2.3) or MSE in (2.5).

In summary we use the standard algorithm from [16] as followings:

1. Initialize the Markov chain by choosing a random starting dendrogram.

2. Run the Monte Carlo algorithm until equilibrium is reached.

3. Resample to find the optimal candidate if needed.

Chapter 3

Evolutionary Pathway Modeling

The fundamental observation for this modeling is that viral progression is accumulated by certain pathways. In this chapter, we first review the work of tree inference from cross-sectional data by Beerenwinkel *et al.* in [12]. We then point out the bias that may exist in the cross-sectional data. Such bias can affect the correctness of the tree structures reconstructed by their method. We then propose a method for improving the accuracy of the parameter estimation in the tree structure inference.

3.1 Introduction

Viral genotypes can be characterized as belonging to clinical meaningful discrete states. In [12] Beerenwinkel *et al.* developed a method for learning viral progression pathways from cross-sectional data using a mixture trees model and achieved considerable success. Recently, Beerenwinkel [13, 24] developed waiting time models of viral progression using Bayesian network analysis. In their model, HIV genotypes evolve on spanning trees. The accumulation on the trees generates different mutation patterns. Each pattern is regarded as a different state. In another direction, Foulkes

and De Gruttola [23] grouped sequences with similar patterns of mutation into clusters to reduce dimensionality; Each cluster is treated as a super node. They then introduce a Markov model on the super nodes that assumes an exponential hazard for each transition. They estimate the transition rates between states.

Many resistance-associated mutations have been identified by the worldwide effort. However, the order in which the resistance-associated mutations accumulate is much less known. Some mutations are known to occur preferentially in clusters, but the order of accumulation is usually unknown. A few studies based on longitudinal (time series) data have revealed directed dependencies between mutations. For example, under zidovudine therapy, K70R and T215F/Y are generally the first mutations to occur. The mutations 41L, 215F/Y and 210W tend to occur together, so do 70R and 219E/Q. Substitution M41L may also appear first though less frequently. However, in contrast to the 70R+215F/Y double mutant, the 41L+70R co-occurrence is hardly ever observed [33].

In general, the evolution of drug resistance is driven by several factors including codon usage bias, random genetic drift, and natural selection. Under therapy, the viral population is exposed to a strong selective pressure. Mutations almost always arise one at a time, and each single advantageous mutation must be fixed into the population. Consequently, although large number of possible mutational patterns can occur, there are only relatively few evolutionary pathways led from the wildtype to mutants with high resistance and good replication capability [27].

The computational challenge we are facing is the limited availability of longitudinal data. Longitudinal data are time series data based on repeated observations on units observed over time. Large longitudinal samples from patients under the same therapeutic regimen are relatively difficult to obtain. On the other hand, cross-sectional data may come from different patients at different time points, which are

much more abundant.

The method of tree structure inference based on cross-sectional data was introduced by Desper *et al.* [19]. They proposed an algorithm that reconstructs the correct tree structure from cross-sectional data. Beerenwinkel *et al.* generalized it to a mixture model of trees by employing an EM-like algorithm and applied this model to study the evolutionary pathways of HIV drug resistance [12]. This model seems to achieve considerable succusses ([12, 13, 24]).

3.2 Previous inference methods: Pathway inference from cross-sectional data

3.2.1 Mutagenetic tree and tree reconstruction

The evolution of drug resistance is viewed as the accumulation of permanent genetic changes in [12]. Their model uses a stochastic evolutionary process to identify directed dependencies between mutational events. The basic building block of the model is a directed tree. Vertices of the tree represent binary random variables, each indicating the occurrence of an event (mutation). Each edge is weighted with the conditional probability of the child given that the parent mutation has occurred. Precisely, a *mutagenetic tree* $\mathcal{T} = (V, E, WT, p)$ consists of a set of vertices $V = \{m_1, ..., m_\ell\}$ representing mutations, a set of edges E, a special vertex $WT \in V$, and a map $p : E \to [0, 1]$ such that

1. (V, E) is a spanning tree, i.e., a digraph whose underlying undirected graph is a forest, and each vertex has at most one entering edge,

2. the vertex WT represents the widetype and has no entering edge,

3. for all edges $e = (u, v) \in E$,

   (a) $p(e) = p(v|u)$ is the conditional probability of mutation v given that mutation u has occurred,

   (b) $p(e) > 0$ (if $p(e) = 0$, we can delete e from E),

   (c) $p(e) < 1$ if e leaves the root (if $p(v|r) = 1$, mutations v and r can be merged).

Figure 3.1 shows a mutagenetic tree for the development of resistance to zidovudine.

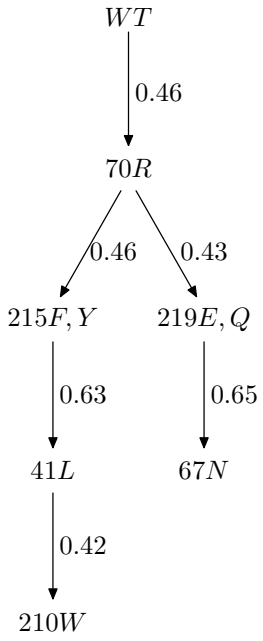


Figure 3.1: A mutagenetic tree for the development of zidovudine resistance.

A mutagenetic tree induces a probability distribution on the set Ω of all possible mutational patterns as follows. The sample space contains all the patterns that are compatible with $\mathcal{T}$. Pick each edge e independently with probability $p(e)$. Then the set of vertices that are reachable from the root is the outcome of the experiment.

Desper *et al.* [19] showed how to reconstruct the mutagenetic tree from all pairwise joint probabilities of mutations. Consider the complete digraph $G = (V, V \times V, w)$ and weight on edge (u, v) being

$$w(u, v) = \log \frac{p(u, v)}{(p(u) + p(v))p(v)}, \tag{3.1}$$

where $p(u)$ denotes the marginal probability of mutation u and $p(u, v)$ the joint probability of mutations u and v. Then the mutagenetic tree is the spanning tree in G that maximizes the sum of its edge weights. The maximum weight spanning tree can be computed in $O(|V||E|)$ time by Edmonds' branching algorithm [22].

In practice, one does not know the joint probabilities of mutations, but have to estimate them from the data. For sufficiently many samples, the above algorithm will reconstruct the correct mutagenetic tree with high probability (see [19] for proofs and a quantitative version of this statement).

3.2.2 Likelihood computation

Given a mutagenetic tree $T = (V, E, WT, p)$, the likelihood of a pattern X of mutations is the probability that T generates X: $L(X|T) = p(X|T)$. Let $S \subset V$ be the set of mutations specified by X. If there is a subset $E' \subset E$ such that S is exactly the set of all vertices reachable from r in the subtree (V, E'), then X can be generated by T, and the likelihood is given by

$$L(X|T) = \prod_{e \in E'} p(e) \prod_{e \in S \times V \setminus S} (1 - p(e)).$$

For example, for the mutagenetic tree displayed in Fig. 3.1 and the pattern

$X = \{70R, 219Q\}$, we find

$$L(X|T) = 0.46 \cdot 0.46 \cdot (1 - 0.43) \cdot (1 - 0.65) = 0.045.$$

If there is no such edge subset, the topology of T does not allow for generating X, and hence $L(X|T) = 0$.

The likelihood computation can be done efficiently by traversing the mutagenetic tree in a breadth-first search starting from r.

3.2.3 Mixture trees model

The tree models provide a detailed and interpretable description of the process of accumulating genetic changes. They represent a considerable improvement over independence or linear path models. However, a single tree model is too strict for the noisy real world data. To overcome this shortcoming, Beerenwinkel *et al.* introduced the broader class of mixture models of trees [12]. In particular, they introduced a "noise component" that includes all otherwise unexplained samples using a star tree. We call $\mathcal{T}$ a star, if all edges $e \in E$ leave the root vertex r. A star models mutations as being independent of each other.

Firstly we follow [12] to define the necessary notations. Suppose that $Z_1, \ldots, Z_K$ are multivariate discrete random variables with range Ω that are distributed according to mutagenetic trees

$$\mathcal{T}_k = (V, E_k, WT, p_k), \ k = 1, \ldots, K,$$

respectively. Let $\Delta_1, \ldots, \Delta_K \in \{0, 1\}$ be binary random variables with $p(\Delta_k = 1) =$

α_k. We call the model

$$\mathcal{M} = \sum_{k=1}^{K} \alpha_k \mathcal{T}_k \text{ with } \alpha_k \in [0,1] \text{ and } \sum_{k=1}^{K} \alpha_k = 1$$

that generates the random variable $Z = \sum_{k=1}^{K} \Delta_k Z_k$, a *K-mutagenetic trees mixture model*.

Thus, the likelihood of a pattern of mutations X in the mixture model is

$$\sum_{k=1}^{K} \alpha_k L(X_i|\mathcal{T}_k).$$

The goal is to find mutagenetic trees $\mathcal{T}_1, \ldots, \mathcal{T}_K$ and mixture parameters $\alpha_1, \ldots, \alpha_K$ that maximize the log-likelihood of the data, that is

$$\sum_{i=1}^{N} \log \sum_{k=1}^{K} \alpha_k L(X_i|\mathcal{T}_k).$$

The responsibility of model component k for sample X_i is defined as

$$\gamma_{ik} = p(\Delta_k = 1|\mathcal{M}, X_i). \tag{3.2}$$

Let $N_k = \sum_{i=1}^{N} \gamma_{ik}$ be the weighted number of samples generated by $\mathcal{T}_k$.

An EM-like algorithm developed by them is presented in Figure 3.2 for learning structure and parameters of the model from data in Table 1.2.

The stability of the EM-like algorithm is examined in [14], where the algorithm

INPUT:

- Patterns of events $X = (x_{ij})_{\substack{1 \le i \le N \\ 1 \le j \le \ell}}$
- Number of mutagenetic trees $K \geq 2$

OUTPUT:

- K-mutagenetic trees mixture model $\sum_{k=1}^{K} \alpha_k \mathcal{T}_k$

PROCEDURE:

1. Guess initial responsibilities:
   (a) Run $(K-1)$-means clustering algorithm
   (b) Set responsibilities
   $$\gamma_{ik} = \begin{cases} \frac{1}{2}, & \text{if } x_i \text{ is in cluster } k-1 \\ \frac{1}{2(K-1)}, & \text{else.} \end{cases}$$
2. *M-like step.* Update model parameters:
   Set $N_k = \sum_{i=1}^{N} \gamma_{ik}$ for all $k = 1, \ldots, K$.
   Let $\mathcal{T}_1$ be a star with edge weights $\beta = \frac{1}{\ell N_1} \sum_{j=1}^{\ell} \sum_{i=1}^{N} \gamma_{i1} x_{ij}$.
   For $k = 2, \ldots, K$:
   (a) For all pairs of events (j_1, j_2), $1 \le j_1, j_2 \le \ell$, estimate their joint probabilities $p_k(j_1, j_2) = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_{ij_1} x_{ij_2}$.
   (b) Compute the maximum weight branching $\mathcal{T}_k$ from the complete digraph with weights w derived from p_k.
   (c) Compute the mixture parameter $\alpha_k = N_k / N$.
3. *E step.* Compute responsibilities:
   $$\gamma_{ik} = \frac{\alpha_k L(x_i | T_k)}{\sum_{m=1}^{K} \alpha_m L(x_i | T_m)}.$$
4. Iterate steps 2 and 3 until convergence.

Figure 3.2: EM-like algorithm for learning a K-mutagenetic trees mixture model [12].

was improved by setting different assignments of the initial responsibilities as follows.

$$\gamma_{ik} = \begin{cases} 0.01, & \text{if } k = 0; \\ 0.99 \cdot \frac{d}{d+(K-2)}, & \text{if } x_i \text{ is in cluster } (k-1); \\ 0.99 \cdot \frac{1}{d+(K-2)}, & \text{else (the remaining } (K-2) \text{ clusters).} \end{cases} \tag{3.3}$$

In this setting the diversity parameter d controls the softness of the initial assignment. Simulations were performed to choose an optimal value for d.

3.3 Parameter normalization

We want to point out the bias in raw cross-sectional data used in previous algorithms. We then propose a method to reduce the bias by normalization of the raw data.

3.3.1 Bias on cross-sectional data

For a pattern its *observing time* is the time during which it can be observed. The frequency for each pattern is determined by two main factors: the probability induced by the true mutagenetic trees, and the length of its observing time. The latter factor was ignored in [19] and [12], where it was assumed that the frequency of a pattern is proportional to the likelihood of this pattern generated by the mutagenetic trees. The bias occurs inevitably. So the precision of the parameters estimation for the tree(s) is not clear. We illustrate how severely this bias can affect the parameters estimation by the following example.

Suppose a true mutagenetic tree is known. Let WT represent the wildtype, and m_1, m_2 represent 2 different mutations. Assume the genotype evolves in the order

of WT, m_1, m_2, as drawn in the mutagenetic tree $\mathcal{T}$ in Figure 1. By this tree structure, the only possible patterns generated by this tree are WT, X_1, X_2 where $X_1 = \{m_1\}$ and $X_2 = \{m_1, m_2\}$. We build a specific tree model by assigning $P(m_1|WT) = 1$ and $P(m_2|m_1) = 0.5$.

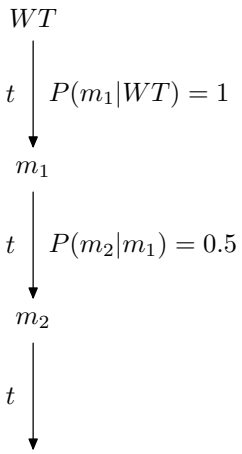


Figure 3.3: A timed mutagenetic tree $\mathcal{T}$.

Then likelihood of patterns generated by $\mathcal{T}$ are

$$L(WT|\mathcal{T}) = 0,$$

$$L(X_1|\mathcal{T}) = 1 \cdot (1 - 0.5) = 0.5 \text{ and}$$

$$L(X_2|\mathcal{T}) = 1 \cdot 0.5 = 0.5.$$

Assume the waiting time for the acquisition of m_1 from WT is t where t represents a reasonable time period, and t for the acquisition of m_1m_2 from m_1. Thus at the time from t to $2t$, the observed pattern is X_1. Suppose the study ends at $3t$. At the time from $2t$ to $3t$, the observed patterns are X_1 and X_2 with equal frequency since $P(m_2|m_1) = 0.5$. Here we assume that the clinical visits for all the patients

in the sample distribute uniformly with respect to the time line. That is, a patient visit the clinician at random, so the probability of the visit lies in each time interval of $(0,t),(t,2t),(2t,3t)$ is $\frac{1}{3}$. Let $P(WT), P(X_1)$ and $P(X_2)$ be the probabilities of observing WT, X_1 and X_2 in the sample respectively. We then have the following calculation.

$$P(WT) = \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot 0 + \frac{1}{3}\cdot 0 = \frac{1}{3},$$

$$P(X_1) = \frac{1}{3}\cdot 0 + \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot \frac{1}{2} = \frac{1}{2}, \text{ and}$$

$$P(X_2) = \frac{1}{3}\cdot 0 + \frac{1}{3}\cdot 0 + \frac{1}{3}\cdot \frac{1}{2} = \frac{1}{6}.$$

When we reconstruct the mutagenetic tree from cross-sectional data we collected, the reconstruction algorithm is still able to determine the correct structure. However, the estimated conditional probabilities differ from the original conditional probabilities.

$$\hat{P}(m_1|WT) = \frac{P(X_1)}{P(WT)+P(X_1)} = 0.6 \text{ and}$$

$$\hat{P}(m_2|m_1) = \frac{P(X_2)}{P(X_1)+P(X_2)} = 0.25.$$

This is a small example with 2 vertices and moderate settings. In this example, the tree structure (topology structure and the positions of the vertices) can be reconstructed correctly. Only some of the parameters are wrong. In more complicated examples, the algorithm may even not be able to reconstruct the tree structure correctly. The tree reconstruction algorithm is to find the tree with maximum weight spanning tree from a complete graph with weight $w(u,v)$ defined in Equation (3.1). Thus this algorithm depends heavily on the values of these probabilities, which are affected by the bias caused by the different lengths of observing times. In large examples, the reconstructed tree is likely very different from the true tree $\mathcal{T}$.

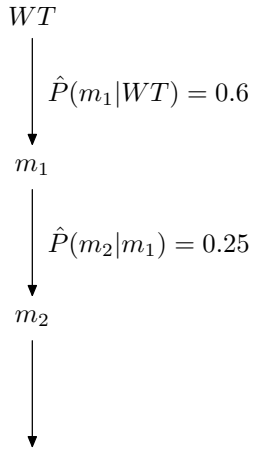


Figure 3.4: The reconstructed mutagenetic tree from data.

3.3.2 Parameter estimation

It seems more reasonable to assume that the frequency of a pattern is proportional to the product of the likelihood and the observing time for this pattern. Nevertheless, it is hard to estimate the observing time from cross-sectional data. However, observing times may be computed using longitudinal data. Below we propose a method to reduce the bias by assuming observing times are available from longitudinal data. We then normalize the frequency of a pattern by dividing a factor proportional to the expected value of the length of the observing time for this pattern.

Let $t(X)$ be the acquisition time of a pattern X from wildtype. Here our big assumption is that $t(X)$ can be estimated by certain approach from longitudinal data. Some related work was done by Healy and De Gruttola in [28], where they developed methods to incorporate interval-censored data in the setting when the exact times of transition are unknown but known only within an interval or known with error. With the advance of the technology the longitudinal data will become more available.

A pattern $X_i = (x_{i1}, \ldots, x_{i\ell})$ can be observed in two situations. The first case

is when X_i remains unchanged. Suppose the total length of the study is T. Then the length of this duration is $T - t(X_i)$. The second case is that it can be observed before the occurrence of any of its proceeding mutations. A proceeding mutation can only occur at the positions where $x_{ij} = 0$. We denote the set of these mutations to be

$$\phi(X_i) = \{m_j : x_{ij} = 0\}.$$

Thus the possible proceeding patterns after X_i are

$$X_i + m_j = (x_{i1}, \ldots, x_{i,j-1}, 1, x_{i,j+1}, \ldots, x_{i\ell}),\ m_j \in \phi(X_i).$$

For example, if $X_i = (1, 0, 1, 0, 0, 1)$, then $\phi(X_i) = \{m_2, m_4, m_5\}$, and $X_i + m_2 = (1, 1, 1, 0, 0, 1)$.

In the second case the length of the duration is $t(X_i+m_j)-t(X_i)$. Now we want to calculate the expected value of the length of the observing time for X_i, denoted as $\theta(X_i)$. In a single mutagenetic tree model, suppose we are given the normalized conditional probabilities $\bar{p}(X_i|X_i)$ and $\bar{p}(X_i + m_j|X_i)$ for $m_j \in \phi(X_i)$. Then

$$\theta(X_i) = \bar{p}(X_i|X_i)(T - t(X_i)) + \sum_{m_j \in \phi(X_i)} \bar{p}(X_i + m_j|X_i)(t(X_i + m_j) - t(X_i)).$$

If there are K mutagenetic trees $\mathcal{T}_k$ with mixture parameter α_k, $k = 1, \ldots, K$. Suppose $\bar{p}_k(X_i|X_i), \bar{p}_k(X_i + m_j|X_i)$ are the conditional probabilities associated with $\mathcal{T}_k$. Then the weighted time when X_i is observable is

$$\theta(X_i) = \sum_{k=1}^{K} \alpha_k \theta_k(X_i), \tag{3.4}$$

where

$$\theta_k(X_i) = \bar{p}_k(X_i|X_i)(T - t(X_i)) + \sum_{m_j \in \phi(X_i)} \bar{p}_k(X_i + m_j|X_i)(t(X_i + m_j) - t(X_i)).$$

Here we assume $t(X_i)$ and $t(X_i + m_j) - t(X_i)$ don't change despite the fact that X_i may evolve through different mutagenetic trees. In fact $t(X_i + m_j) - t(X_i)$ is not sensitive to different trees, and when T is relatively large comparing to $t(X_i)$, $T - t(X_i)$ is not sensitive as well.

So $\theta(X_i)$ varies for different X_i. We want to normalize the indicator vectors by dividing a bias factor proportional to $\theta(X_i)$, say $B(X_i)$, which will be defined in (3.9). Given $B(X_i)$, we normalize the vector X_i by

$$\bar{X}_i = \frac{X_i}{B(X_i)}. \tag{3.5}$$

By this normalization performed on X_i, the bias caused by the different observing time for X_i gets reduced. Now the normalized distribution of the samples is supposed to be identical to the probability distribution induced by the true mutagenetic trees. The sample size is also normalized and becomes $\bar{N} = \sum_{i=1}^{N} \frac{1}{B(X_i)}$.

Let γ_{ik} be the responsibility of model component k for sample X_i as defined in (3.2) and let $N_k = \sum_{i=1}^{N} \gamma_{ik}$ be the weighted number of samples generated by $\mathcal{T}_k$. Given estimates of γ_{ik}'s, we compute the normalized probabilities in the k-th component as follows. For a mutation j, $1 \leq j \leq \ell$, its marginal probability is

$$\bar{p}_k(j) = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \bar{x}_{ij}. \tag{3.6}$$

For a pair of mutations (j_1, j_2), $1 \leq j_1, j_2 \leq \ell$, their joint probability is

$$\bar{p}_k(j_1, j_2) = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \bar{x}_{ij_1} \bar{x}_{ij_2}. \tag{3.7}$$

Along with the weight function defined in (3.1), we can reconstruct $\mathcal{T}_k$ by finding the maximum weight spinning tree as described before. The mixture parameter for $\mathcal{T}_k$ is

$$\alpha_k = \frac{N_k}{\sum_{k=1}^{K} N_k}.$$

Given an estimate of $\mathcal{M} = \sum_{k=1}^{K} \alpha_k \mathcal{T}_k$, we can update the responsibilities by

$$\gamma_{ik} = \frac{\alpha_k L(X_i|\mathcal{T}_k)}{\sum_{m=1}^{K} \alpha_m L(X_i|\mathcal{T}_m)}. \tag{3.8}$$

Note that $\sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{ik} = \bar{N}$. Meanwhile $\sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{ik} = N$, thus $\sum_{i=1}^{N} \frac{1}{B(X_i)} = N$. Now assume $B(X_i) = c \cdot \theta(X_i)$ where $c \in \mathbb{R}$ is a constant. Then $\sum_{j=1}^{N} \frac{1}{c \cdot \theta(X_j)} = N$. We can solve for c and it follows that

$$B(X_i) = \frac{\theta(X_i) \sum_{j=1}^{N} \frac{1}{\theta(X_j)}}{N}. \tag{3.9}$$

We replace the parameters $(X, p_k(j), p_k(j_1, j_2))$ in the algorithm in Figure 3.2 by the normalized parameters $(\bar{X}, \bar{p}_k(j), \bar{p}_k(j_1, j_2))$. The flow of the algorithm remains good. The initial bias factor for each pattern is set to be 1 and will be updated iteratively in the E-step.

Algorithm for learning mixture models

INPUT:

- Patterns of mutations $X = (x_{ij}), 1 \leq i \leq N, 1 \leq j \leq \ell$.
- Number of mutagenetic trees $K \geq 2$.

OUTPUT:

- K-mutagenetic trees mixture model $\sum_{k=1}^{K} \alpha_k \mathcal{T}_k$.

PROCEDURE:

1. Initialization:
   (a) Run $(K-1)$-means clustering algorithm
   (b) Set the initial responsibilities γ_{ik} given in (3.3).
   (c) Set $B(X_i) = 1$ for $1 \leq i \leq N$.
2. *M-like step.* Update model parameters:
   Set $N_k = \sum_{i=1}^{N} \gamma_{ik}$ for all $k = 1, \ldots, K$.
   Let $\mathcal{T}_1$ be a star with edge weight $\beta = \frac{1}{\ell N_1} \sum_{j=1}^{\ell} \sum_{i=1}^{N} \gamma_{i1} \bar{x}_{ij}$.
   For $k = 2, \ldots, K$:
   (a) $\bar{X} = \frac{X}{B(X)}$.
   (b) Estimate $\bar{p}_k(j)$ for $1 \leq j \leq \ell$ and $\bar{p}_k(j_1, j_2)$ for $1 \leq j_1, j_2 \leq \ell$ as defined in (3.6) and (3.7) respectively.
   (c) Compute the maximum weight spanning tree $\mathcal{T}_k$ from the complete digraph with weights w using normalized probabilities derived in (b) of step 2.
   (d) Compute the mixture parameter $\alpha_k = \frac{N_k}{N}$.

3. *E-step.*

   (a) Compute the responsibilities as defined in (3.8).

   (b) Compute $B(X_i)$ as defined in (3.9).

4. Iterate steps 2 and 3 until convergence.

Chapter 4

Predicting Drug Resistance from Genotypes

In this chapter we develop a regression model for predicting phenotypic drug resistance from genotypes. The novelty of our method is that we use the evolutionary pathways to determine the coefficients for the terms in the regression function. The model combines several linear regression models, but the model itself is not linear any more.

4.1 Introduction

Considerable attention has been focused on predicting phenotypic drug resistance from genotypes. This research is essential to developing new antiretroviral drugs and optimizing the use of current drugs. Typical predicting approaches include decision trees [9], linear regression [51, 54], linear discriminant analysis [47], neural networks, support vector regression (SVR) [8], ridge regression, principal component analysis, and some non-parametric methods [20].

Phenotypic drug susceptibility testing provides a direct and quantitative measure of the viral behavior in vitro, however, the assays are costly and time-consuming. Several practical difficulties have obstructed its adoption. Phenotype assays are very expensive, costing $700 to $900 per sample. In addition, the testing procedure is complex and only offered by a few companies. Furthermore, processing time can take as long as 8 weeks. Unless these issues can be significantly changed, there is little chance that phenotype testing can possibly be adopted on a widespread basis.

On the contrary, genotype testing has enjoyed much wider usage because the procedure is much simpler, the costs are lower, and the processing time is shorter. Genotype assays typically cost around $400 and take 1-2 weeks for processing. A recent cost benefit analysis supported genotypic testing after treatment failure. These factors suggest that genotype testing will continue to be a more common method.

4.2 Previous linear regression models

Various genotypic interpretation algorithms have been developed for this study. We review below in details the algorithms in [51, 54] using linear regression models. Both studies follow the general settings of a multiple linear regression modeling: A dependent variable is expressed as a linear combination of independent variables and an intercept. Both models treat the natural logarithm of FC (defined in (1.1)), $\log \mathrm{FC}$, as a linear combination of position- and type-specific mutations plus a constant. In this case, the independent variables mut_j are terms representing the presence of mutations in a genetic sequence (In our language, we use x_{ij} to be the indicator variable for a pattern X_i with respect to a mutation m_j). The model assigns a coefficient a_j, or weight, to each mutation corresponding to its contribution to the $\log \mathrm{FC}$. The intercept a_0 represents the FC prediction for a wild type sequence lacking

any susceptibility altering mutations. See Equation (4.1).

$$\log \mathrm{FC} = a_0 + a_1 \mathrm{mut}_1 + a_2 \mathrm{mut}_2 + \cdots + a_n \mathrm{mut}_n \tag{4.1}$$

4.2.1 Simple linear model

Wang *et al.* [51] used a linear regression model to investigate the relationship between HIV genotype and drug resistance. They built a full regression model based on current knowledge about HIV drug resistance. In this model, the dependent variable, $\log \mathrm{FC}$, were determined by either Virologic's PhenoSense assay or Virco's Antivirogram assay. To reduce the number of independent variables in their regression model, they only used mutations in 'important' positions that are known to influence drug resistance. In a few cases where the sequence contained a mixture of two or more amino acids at the same position, they took the value of each of the corresponding indicator variables to be 1.

They used a backward stepwise regression method to optimize the parameters for each independent variable and the constant. The stepwise regression begins with the full model, where all independent variables are contained in the model. In each subsequent step, the removal statistic is computed for each independent variable eligible to be removed from the model, and the variable with the highest removal statistic is removed from the current model if it is more than a critical removal value. Then the entry statistic is computed for each independent variable that is not included in the current model, and the variable with the lowest entry statistic is added into the current model if it is lower than the entry statistic. The stepwise regression procedure stops if neither entry nor removal is possible. The remaining variables comprise the reduced model. They used the P-value as the entry and removal statistic, which

indicates the possibility of observing such data when such variable is not associated with the dependent variable. The critical P-values for removal and entry were set to 0.051 and 0.05, respectively. Regression analysis and data manipulation were done using the statistics software STATA.

They collected 5507 genotype-phenotype paired records covering 17 anti-HIV drugs from the Virologic dataset. They constructed a separate regression model for each drug and checked the validity of these models.

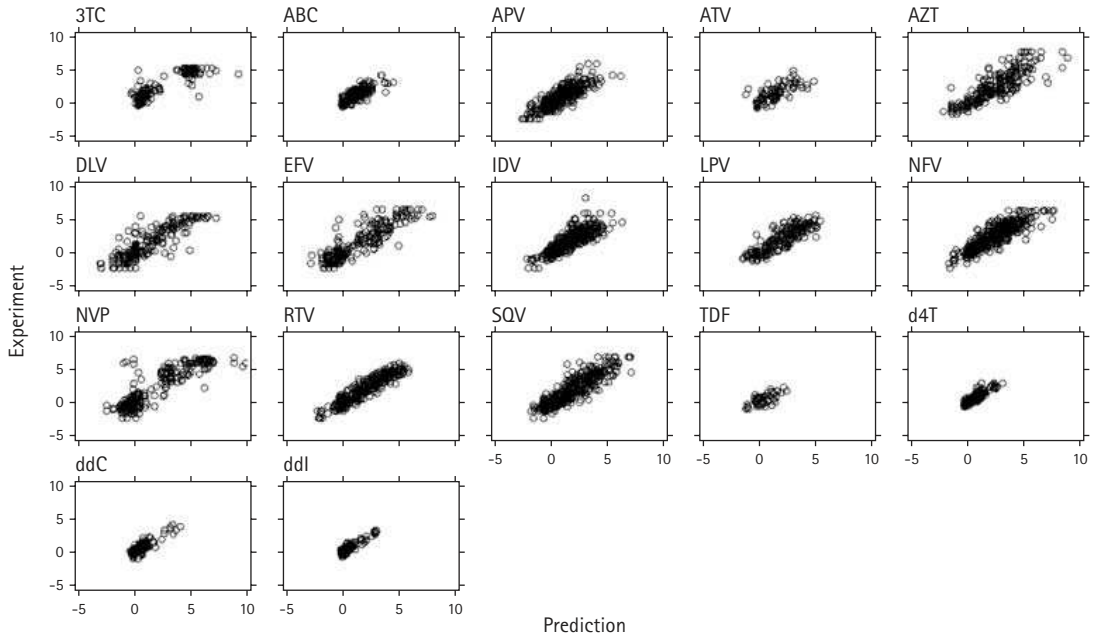


Figure 4.1: Scatter plot of experimental versus predicted resistance values

Their model performed well despite the absence of statistical terms for interaction effects. The scatter plots in Figure 4.1 also demonstrated the good correspondence between the experimental values and their quantitative predictions. Their tests on an independent dataset show that their method outperformed six other publicly available HIV genotypic interpretation algorithms for most drugs at predicting drug resistance in published datasets.

4.2.2 Linear model considering interactions

In a similar manner, Vermeiren *et al.* [54] applied linear regression modeling to predict the HIV-1 resistance phenotype from the viral genotype. In their approach, the quantitative phenotypic measurement is also estimated as the weighted sum of the effects of individual mutations, see (4.1). The difference from [51] is that the second order interaction terms (mutation pairs) were included to account for synergistic and antagonistic effects between mutations. The most significant mutations and interactions identified by the linear regression models for 17 approved antiretroviral drugs are reported, and their negative or positive impact on the FC is quantified by a resistance weight factor.

Linear regression models were calculated by selecting mutations having a significant contribution to the $\log \mathrm{FC}$ and determining the magnitude of their effect. Linear regression models per drug were calculated in two steps. A first model was calculated using single mutations only. Based on the mutations that proved to be significant in this model, mutation pairs (simultaneous presence of two mutations) were subsequently created to assess the importance of interaction terms in addition to single terms in a second order regression step. These interaction terms account for synergistic and antagonistic effects between mutations. Parameter selection for both steps was based on mutation prevalence in the data sets.

The number of terms was arranged as follows. For relatively old drugs having larger data sets, at least 20 single mutations were considered, and 10 for relatively new drugs (namely LPV, FTC, ATV, TPV and DRV) having smaller data sets. The same minimal count per drug was applied to pairs, where in addition every mutation of the pair had to be present at least 20 (or 10) times in the data set without the other.

Linear regression models were calculated using stepwise regression utilizing SAS. In this method, the independent variables are alternately added to and removed from the growing model based on statistical threshold values (P-value). In each forward step, the most predictive parameter is chosen, and in the subsequent backward step, the more stringent P-values allow elimination of the least predictive variables.

4.3 Tree-specific linear model

First we want to formulate the problem precisely. For each genotype X_i, let Y_i be the the corresponding phenotype, say $\log FC$. Suppose that we are given N sample pairs $(X_i, Y_i), i = 1, \ldots, N$. We want to find a regression function F so that $F(X_i) = Y_i + \varepsilon_i$. We hope F satisfies the following properties:

1. The number of terms of F is small, and these terms represent the major factors that have effects on F;
2. $\sum_{i=1}^{N} \varepsilon_i^2$ is small;
3. F can predict drug fold resistance, that is, for a genotype $X \neq X_i, i = 1, \ldots, N$, $F(X)$ approximates to phenotype of X.

We will use information from the evolutionary pathways, i.e. the mutagenetic trees, to determine the terms in the regression function F. In the following a set of mutations represents a pattern in which exactly these mutations occur.

In Chapter 3, tree models are introduced to identify the progression of the mutations. Due to high selective advantage, if there is any correlation or interaction between two mutations, then it is not likely that these two mutations occur independently. This means that one mutation must be a descendant of the other in a

tree representation. In other words, if two mutations occur independently, then the value of F is linear on their individual presence. We illustrate this assumption by two examples. Suppose m_1 and m_2 are the children of the root. It is reasonable to assume that the increment on F when m_1 evolves from the wildtype is independent of the presence of m_2 and vice versa. Thus

$$F(m_1) - F(WT) = F(m_1, m_2) - F(m_2). \tag{4.2}$$

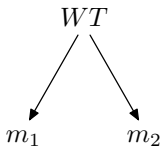


Figure 4.2: m_1 and m_2 are independent.

In another situation, suppose m_1 is a child of the root and m_2, m_3 are children of m_1. So m_2, m_3 are independent conditionally on m_1. Hence the increment on F when m_1 evolves to m_1, m_2 is independent on the presence of m_3. Thus

$$F(m_1, m_2) - F(m_1) = F(m_1, m_2, m_3) - F(m_1, m_3). \tag{4.3}$$

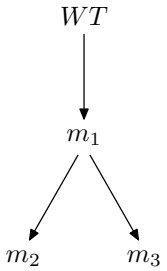


Figure 4.3: m_2 and m_3 are independent conditionally on m_1.

Equations (4.2) and (14) can be extended to more general situation by recursion. The evolution of a viral genotype is the accumulation of mutations. Given a tree $\mathcal{T}$, suppose in the next step a mutation m_{j_2} will occur after m_{j_1}. Then the increment on F from current pattern to the next pattern is an invariant. It is independent of the presence of any mutation outside the path from WT to m_{j_1}. For an edge of $\mathcal{T}$ with endpoint m_j, we associate it with an increment function, say $\delta(m_j)$. If the data are generated by a single tree, then there are only $\ell + 1$ terms in the regression function. With sufficient input, $\delta(m_j)$'s and $F(WT)$ can be learned by some classical approaches, for example, the least square method. Then for a genotype $X = (x_1, \ldots, x_\ell)$,

$$F(X) = \sum_{j=1}^{\ell} x_j \delta(m_j) + F(WT).$$

Our independent assumption is similar in spirit to the one in Sec 2.3 of [29], where they assume that the probability of acquiring a mutation at a branch does not depend on the presence of a mutation at another branch which meets the first branch at the root. Relaxation of our assumption requires either additional trees (which will be discussed below) or allowing error.

Our analysis shows that if the genotypes are generated by a single tree, then a linear regression model is reasonable. This provides a support for the good performance of the linear regression model in [51].

However, previous studies already suggested that mixture model of trees is necessary [12]. Our model can be naturally generalized to a mixture model of trees. For two patterns X_1 and X_2, define their union by

$$X_1 \cup X_2 = (\max\{x_{1j}, x_{2j}\} : j = 1, \ldots, \ell).$$

Suppose there are K mutagenetic trees and $\mathcal{T}_1$ is the star representing the noise component. For a pattern X, suppose it can be decomposed as $\bigcup_{k=1}^{K} X^k$, where each X^k is generated by $\mathcal{T}_k$. Since $\mathcal{T}_1$ can generate all possible patterns and each mutation occurs independently, we choose $\mathcal{T}_1$ with least priority. Still the decomposition is not unique. For example, suppose the mutagenetic trees are given in Figure 4.4. For a pattern $X_i = (0, 0, 1, 0, 1, 1)$, it can be decomposed as

1. $X_i = X_i^3 = (0, 0, 1, 0, 1, 1)$,
2. $X_i = X_i^2 \cup X_i^3$ where $X_i^2 = (0, 0, 1, 0, 0, 1)$ and $X_i^3 = (0, 0, 1, 0, 1, 0)$, or
3. $X_i = X_i^2 \cup X_i^3$ where $X_i^2 = (0, 0, 1, 0, 0, 1)$ and $X_i^3 = (0, 0, 0, 0, 1, 0)$.

We have to set up certain rules for selection of decompositions. It is desirable that each induced subtree by X_i^k in $\mathcal{T}_k$ is in a sense of maximality. For a mutagenetic tree $\mathcal{T}_k$, we define $\psi(T_k)$ to be the set of patterns that are compatible with $\mathcal{T}_k$. That is,

$$\psi(\mathcal{T}_k) = \{X : X \text{ is compatible with } \mathcal{T}_k\}.$$

Here we view X_i as a set of mutations. For $k \geq 2$, we define X_i^k for a pattern X_i to be the maximal $X \in \psi(\mathcal{T}_k)$ such that $X \subset X_i$.

$$X_i^k = \max\{X \in \psi(\mathcal{T}_k) : X \subset X_i\}. \tag{4.4}$$

Note that for given X_i and $\psi(\mathcal{T}_k)$, X_i^k can be uniquely determined. For $k = 1$, let

$$X_i^1 = X_i \setminus \bigcup_{k=2}^{K} X_i^k. \tag{4.5}$$

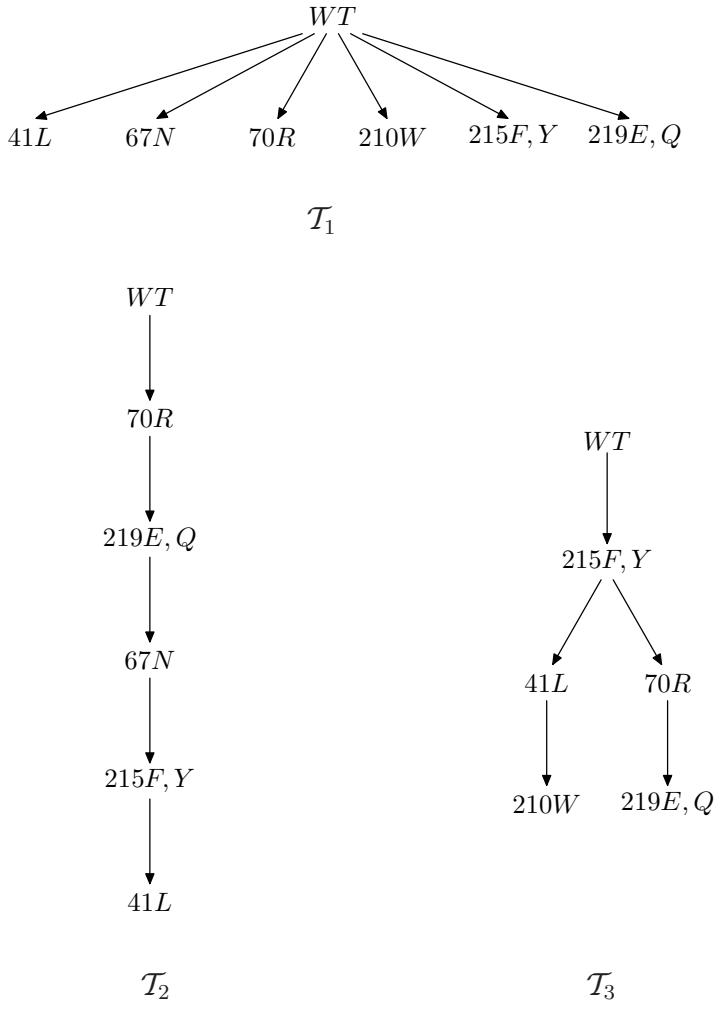


Figure 4.4: Three-mutagenetic trees mixture model.

Suppose we have $X_i = \bigcup_{k=1}^{K} X_i^k$. Now we view X_i and X_i^k's as binary vectors. For each X^k, we can have a regression function

$$F_k(X_i^k) = \sum_{j=1}^{\ell} x_{ij}^k \delta_k(m_j) + F_k(WT).$$

Then

$$F(X_i) = \sum_{k=1}^{K} F_k(X_i^k). \tag{4.6}$$

In a classical linear regression function, for a pattern X_i, there is only 2 choices for the coefficient of a mutation m_j:

$$\text{coefficient of } m_j = \begin{cases} \delta(m_j) & \text{if } x_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

In our model, for a pattern X_i, the coefficient of a mutation m_j is a combination of up to K coefficients. That is, $\sum_{k=1}^{K} \delta_k(m_j)$, where

$$\text{the } k\text{-th coefficient of } m_j = \begin{cases} \delta_k(m_j) & \text{if } x_{ij}^k = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Thus the total number of possible combinations for m_j becomes 2^K. This provides more flexibility for the regression model. However, we need only to find $K(\ell + 1)$ undetermined coefficients. The parameter learning can use classical approaches for parameter learning for a simple linear regression model, like the least square method.

In summary, for given mutagenetic trees $\mathcal{T}_k$, $k = 1, \ldots K$ and sample pairs (X_i, Y_i), $i = 1, \ldots, N$, our regression model contains the following steps:

1. For each X_i, decompose it as $X_i = \bigcup_{k=1}^{K} X_i^k$ where X_i^k's are defined in (4.5) and (4.4).

2. Build the regression function defined in (4.6), and learn the undetermined coefficients $\delta_k(m_j)$'s and $F_k(WT)$'s.

4.4 Discussion

For multiple mutagenetic trees, the linearity in our model actually is destroyed. That is, for two mutations m_1 and m_2, $F(m_1 + m_2)$ may not necessarily equal $F(m_1) + F(m_2)$. Only in a single tree model, our regression model degenerates to a simple linear regression model, like the one in [51]. This makes us believe that the performance of our method for multiple tree models is comparable to the performance of [51], which outperformed 6 other algorithms. This also provides some reason why their linear regression model can have good performance.

Another way to view the difference of our regression function from classical linear regression functions in [51] is, even the same mutation in different patterns might have different effects on F. It really depends on which tree this mutation lies in at the decomposition step.

Our regression function uses at most $K(\ell + 1)$ terms in total. In [45], second-order polynomials were proposed to model interactions between each of the input mutations using Least Angle Regression method. The number of terms of the regression function becomes ℓ^2. The number of major mutations in RT is estimated to be 30$\sim$40 for RTI resistance. This produces either a large regression function or long process of term selection.

The goodness of our regression model depends heavily on the correctness of the topological structures of the mutagenetic trees. The parameters (mixture parameters, conditional probabilities) of the trees can be ignored. Cross-validation method can be used for performance test. The accuracy of the models in predicting the phenotype can be expressed as the mean squared error (MSE), with the error being defined as the difference between prediction and measurement. The MSE is calculated both on the data set used for training the models (seen data) and on unseen data in a 10-fold

cross-validation. In the cross-validation, 10 models are generated on 90% of the data and the MSE is calculated on the remaining 10%. The mean of those 10 MSE values serves as a measure for the error on unseen data of the reported model (trained on all data).

Regression models encounter a common problem that they cannot be used for novel rare mutations that are not included in the training process. Classification methods can be considered for treating minor strains, both for data training and for the prediction. Hopefully with the increasing size of public databases, this problem will be less severe in the future.

Chapter 5

Relative Fitness of Mutation Patterns

Determining the fitness of drug-resistant HIV-1 strains is necessary for the development of population-based studies of resistance patterns. We propose a method with quantitative measurement for the viral fitness. Our initial computational results closely agree with the experimental results from [41] and [32].

5.1 Introduction

The short life cycle and high replicative error rate of human immunodeficiency virus type 1 (HIV-1) cause the virus to mutate very rapidly, resulting in a high genetic variability of HIV. Most of the mutations either are inferior to the parent virus or convey no advantage, but some of them have a natural selection superiority to their parents and can enable them to slip past defenses such as the human immune system and escape from the anti-retroviral drugs. The more active copies of the virus, the more resistant to anti-retroviral drugs. The fitness of a mutation is interpreted

as the replication capability, or by how well it can fit the environment. Determining the fitness of drug-resistant HIV-1 strains is necessary for the development of population-based studies of resistance patterns [18], and it has potential application for therapeutic strategies.

All currently recommended anti-retroviral drugs therapy protocols employ reverse transcriptase inhibitors (RTIs). However, mutations within the reverse transcriptase (RT) domain can lead to resistance to these agents and treatment failure [36, 46, 17]. Zidovudine (AZT) is the first approved nucleoside analogs inhibitor (NRTI). For zidovudine an accumulation of multiple mutations is required to confer high-level resistance [36, 34]. Resistance to NTRIs develops either through enhanced excision of chain terminators (thymidine analog mutations, TAMs), e.g., 41L, 67N, 70R, 210W, 215FY, and 219EQ, or by causing lower binding affinity, e.g., 184V, 65R, and 74V [42].

Paintsil *et al.* (2006) utilized a reproducible, systematic real time RT/PCR-based assay that allows the relative contribution of mutant viral fitness and drug resistance as selective pressures underlying resistance patterns of HIV-1 [18], to measure these forces for mutations that confer resistance to nevirapine and zidovudine, and found that viral fitness does play a role in the evolution of resistance to these inhibitors.

Conventional laboratory methods are resource and time consuming, and experiments in vitro may not reflect the natural viral response in vivo. On the other hand, advanced sequencing technologies produce rich sequence data. It is desirable to find the knowledge hidden in the data using computational approaches. In this study we propose a computational method to determine the relative fitness of different mutation patterns of HIV-1 in the presence of AZT. We first use a clustering method to determine the groups, namely TAM 1 and TAM 2. Next, we employ an

EM algorithm to find an optimal model to fit the data. Then for a certain mutant pattern, we define a quantitative measure to reveal the relative fitness of that mutant pattern. Our computational results strongly agree with the experimental results from two independent studies [41] and [32].

5.2 Results

We use the same data that was used in [12] (Table 1.2).

Our computational results show that in the presence of AZT, in TAMs pathway 1 (41L, 210W, 215Y), the triple mutants 41L + 210W + 215Y appear fitter than the double mutants 41L + 210W or 210W + 215Y. And in TAMs pathway 2 (67N, 70R, 219Q), the triple mutants 67N + 70R + 219Q is fitter than the double mutants 67N + 70R or 67N + 219Q as well. The multiple mutants in TAM 1 out-compete the multiple mutants in TAM 2.

Our computational results strongly agree with the experimental result from [41], which showed that the fitness of four selected multiple mutation patterns in the presence of AZT is in the following order: 41L + 210W + 215Y > 210W + 215Y > 67N + 70R + 219Q > 67N + 70R. The only exception is that our result is against 67N + 70R> 70R. We try to explain this in the Discussion section.

In another independent study [32], it is showed that in the fitness comparison, $41L{+}210W{+}215Y > 41L{+}215Y$ and $67N{+}70R{+}219Q > 67N{+}70R{+}210W{+}219Q$. Our results agree with these.

See table 5.1 for all the comparisons. For each comparison a yes or no in the column of consistence indicates whether or not our computational result agrees with the experimental result.

Table 5.1: Experimental results v.s. computational results.

Reference	Comparisons	Our scores	Consistence
[41]	41L+210W+215Y > 41L+215Y	1.21 > 1.09	yes
[41]	41L+215Y > 67N+70R+219Q	1.09 > 0.944	yes
[41]	67N+70R+219Q > 67N+70R	0.944 > 0.772	yes
[41]	215Y > 67N+70R	0.874 > 0.772	yes
[41]	67N+70R > 70R	0.772 < 1.06	no
[32]	41L+210W+215Y > 41L+215Y	1.21 > 1.09	yes
[32]	67N+70R+219Q+215F > 67N+70R+219Q	2.97 > .944	yes
[32]	67N+70R+219Q > 67N+70R+219Q+210W	0.944 > 0	yes
[32]	67N+70R+219Q+215F > 67N+70R+219Q+210W	2.97 > 0	yes

5.3 Method

5.3.1 Clustering algorithm to determine TAMs

In the literature, TAMs have been found to cluster into two main groups [15, 26]. These are 41L, 210W, 215Y (pathway 1) and 67N, 70R, 219Q (pathway 2). Without this information we can also derive these clusters from our data by some standard clustering approach. For convenience we assume the mutants 41L, 67N, 70R, 210W, 215Y, and 219Q are represented by $m_1, m_2, m_3, m_4, m_5, m_6$ respectively. For two mutants m_i and m_j, we construct the following contingency table 5.2 and define the distance d_{ij} to be the Chi-squared test score.

Table 5.2: Chi-squared test for independence of m_i, m_j.

percentage (neither of m_i, m_j occurs)	percentage (only m_j occurs)
percentage (only m_i occurs)	percentage (both m_i, m_j occur)

By such definition of distance, the smaller of d_{ij}, the more likely the mutants m_i and m_j are independent with each other. All the pairwise distance is in the following table.

Table 5.3: Chi-squared test and p-values.

d_{ij}	41L	67N	70R	210W	215Y
67N	0.025(0.87)				
70R	0.22(0.64)	0.15(0.70)			
210W	0.22(0.64)	0.0043(0.95)	0.041(0.84)		
215Y	0.45(0.50)	0.040(0.84)	0.0020(0.96)	0.20(0.65)	
219Q	0.0035(0.95)	0.50(0.48)	0.22(0.64)	0.0057(0.94)	0.016(0.90)

Next we construct a complete graph G whose vertices are the six single mutations and whose weight on the edge e_{ij} is the distance d_{ij}. We then remove the edges with smaller weight from G in an increasing order and determine the groups using some clustering approach, for example the one in [39]. Nevertheless, for our case, it is clear that two groups are perfect, with only one exceptional edge (edge 1-3). In figure 5.1, dotted edges have weight less than 0.05 (p-value: 0.82) thus are removed.

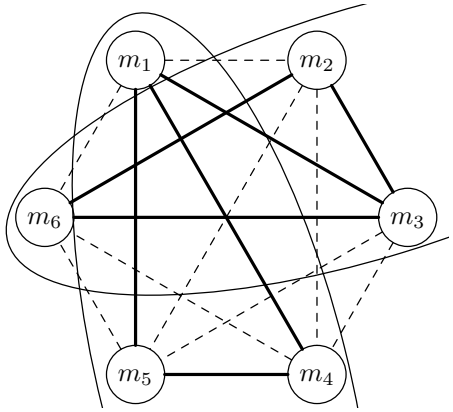


Figure 5.1: Weighted graph.

5.3.2 EM algorithm

We now know that a mutant in TAM 1 occurs relatively independent on the occurrence of another mutant in TAM 2. It is not necessary to assume they are completely independent. Instead, we will use the incompleteness on independence to measure the degree in which a pattern presents. The measurement should also reveal its relative fitness.

Table 5.4: The observed probabilities of all patterns.

o_{ij}	wt_a	41L	210W	215Y	41L/210W	41L/215Y	210W/215Y	41L/210W/215Y
wt_b	.316	.00550	0	.0385	0	.0824	.00824	.0687
67N	.00275	.00275	.00275	.00275	0	.00824	.00550	.0137
70R	.165	.00275	0	.0192	0	.0192	0	0
219Q	.00550	0	0	.00275	0	0	0	0
67N/70R	.0137	.00275	0	.00824	0	.0110	.00324	.0110
67N/219Q	.00275	0	0	.00275	0	0	0	.00550
70R/219Q	.0165	0	0	.0192	0	.00275	0	0
67N/70R/219Q	.0632	.00275	0	.00275	0	.0302	0	0

Note that we have 64 possible mutation patterns, and each of them can be expressed by the combinations of one pattern from TAM 1 and another one from TAM 2. More precisely, in TAM 1, let patterns $X_i : i = 1, \ldots, 8$ represent the patterns wildtype, m_1, m_4, m_5, m_1m_4, m_1m_5, m_4m_5, $m_1m_4m_5$ respectively. And the simulated probability for pattern X_i is denoted by a_i. For example, $m_1m_4m_5$ represents the triple mutants 41L + 210W + 215Y, and its simulated probability is a_8. In a similar manner in TAM 2, let patterns $\bar{X}_i : i = 1, \ldots, 8$ represent the patterns wildtype, m_2, m_3, m_6, m_2m_3, m_2m_6, m_3m_6, $m_2m_3m_6$ respectively, and b_i for the simulated probability of $\bar{X}_i$. Let o_{ij} be the observed probability of the mutation pattern $X_i \cup X_j$

(defined as $X_i \cup X_j = (\max\{x_{i1}, x_{j1}\}, \ldots, \max\{x_{i6}, x_{j6}\})$), where X_i is from TAM 1 and X_j is from TAM 2. We build a contingency table 5.4 for all the values of o_{ij}'s. A 0 means the corresponding pattern is not detected in the data.

We want to find the simulated probabilities of the patterns that can best explain the data. That is, we need to find all a_i's, b_i's such that

$$\sum_{i,j}(o_{ij} - a_i b_j)^2$$

is minimum. This problem can be done by employing EM algorithm.

Algorithm for parameter estimation of an optimal model

1. Set initial values $a_i = \sum_{j=1}^{8} o_{ij}$ for i from 1 to 8.
2. Update
$$b_j = \frac{\sum_{i=1}^{8} o_{ij} a_i}{\sum_{i=1}^{8} a_i^2} \text{ and } a_i = \frac{\sum_{j=1}^{8} o_{ij} b_j}{\sum_{j=1}^{8} b_j^2}.$$
Repeat until $\sum_{i,j}(o_{ij} - a_i b_j)^2$ is convergent.
3. Output a_i, b_j.

By iterations, the sum of squares converges very quickly. The result is shown in Table 5.5. This model covers around 78% of the original data. Note that if TAM 1 an TAM 2 are completely independent then the model can cover 100% of the data in theory.

Table 5.5: EM result.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
.622	.0120	.000317	.0860	0	.154	.0148	.111
b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
.512	.0112	.249	.0849	.0286	.00593	.0286	.108

5.3.3 Fitness score function

For a pattern $X_i \cup X_j$, the observed probability is o_{ij} and the simulated probability is $a_i b_j$. Take a consideration on the ratio $\frac{o_{ij}}{a_i b_j}$. The higher it is, the relatively more the pattern over present, and the higher fitness it should have. By this intuition we define the fitness score function of mutation $X_i \cup X_j$ to be

$$S(X_i \cup X_j) = \frac{o_{ij}}{a_i b_j}. \tag{5.1}$$

Table 5.6: Scores.

	W	41L	210W	215Y	41L/210W	41L/215Y	210W/215Y	41L/210W/215Y
W	0.993	0.892	0	0.874	0	1.04	1.09	1.21
67N	0.392	20.3	769	2.84	0	4.73	32.9	11.0
70R	1.06	0.917	0	0.898	0	0.500	0	0
219Q	1.04	0	0	3.77	0	0	0	0
67N/70R	0.772	7.98	0	3.35	0	2.49	19.4	3.48
67N/219Q	0.745	0	0	5.39	0	0	0	8.39
70R/219Q	0.926	0	0	7.81	0	0.621	0	0
67N/70R/219Q	0.944	2.12	0	2.97	0	1.82	0	0

For example, the pattern 67N + 70R + 219Q+215F is made up of the pattern 215F from TAM 1 and 67N + 70R + 219Q from TAM 2. In particular, we treat the

pattern 41L+210W+215Y as a combination of a pattern from TAM 1 and a wildtype pattern from TAM 2. Although wildtype indicates no mutation occurs, it is still a genotype and has biochemical properties. For the patterns with $o_{ij} = 0$, we simply put 0 as its relative fitness score. The computational result is shown in Table 5.6.

For the patterns with $0 < o_{ij} < 1\%$, our method might not be reliable because since the score function becomes extremely sensitive because of the small value of denominator. Thus we remove these scores and leave as blanks. This results in Table 10.

Table 5.7: Reliable scores.

	W	41L	210W	215Y	41L/210W	41L/215Y	210W/215Y	41L/210W/215Y
W			0	0.874	0	1.04		1.21
67N					0			11.0
70R	1.06		0	0.898	0	0.500	0	0
219Q		0	0		0	0	0	0
67N/70R	0.772		0		0	2.49		3.48
67N/219Q		0	0		0	0	0	
70R/219Q	0.926	0	0	7.81	0	0.621	0	0
67N/70R/219Q	0.944		0	2.97	0	1.82	0	0

From Table 5.7 we predict that the patterns 41L+210W+215Y+67N and 70R+219Q+215Y have very large relative fitness scores.

5.4 Discussion

In our method, higher score means the corresponding pattern has relatively better fitness and higher resistance to a drug. Thus once a patient has such mutation pattern in his/her plasma, it is not wise to apply this drug to this person.

Since we are working on cross-sectional data, our method also suffers from the bias problem. Several studies shows that 70R is generally the first mutation to occur. It is possible that this mutation is over observed, which leads to higher relative fitness score according to our formula (5.1).

We have only applied our method to the mono-therapy of AZT. Current treatment usually involves a combination of multiple drugs. Our future work is to generalize the methods to more general situations, for example for other drugs or even in multi-drug environment.

Chapter 6

Future Problems

This thesis aims at development of methods for interpretation of HIV genotype data from several perspectives. Some are improvement of previous results and some are newly developed. We describe several related problems that remain challenging.

Cross-resistance. To my knowledge there are no computational models for this study so far, though cross-resistance is prevailing under current drug therapy. For example, the area of pathway inference, or the drug resistance prediction under multi-drug environment remains untouched.

TB-HIV co-infection. HIV and Tuberculosis (TB) have variously been referred to as 'the terrible twins' and 'Bonnie and Clyde'. Among the people died of tuberculosis in 2006 worldwide, it is estimated that 200,000 were infected with HIV. TB is the leading cause of death among HIV infected people in Africa. The TB-HIV co-infection crisis is twice as big as previously thought. In 2007, there were at least 1.37 million cases of HIV-positive TB, or nearly 15 percent of the total incident cases [7]. That's double the previous WHO estimates.

When someone is infected with TB the likelihood of them becoming sick with the disease is increased many times if they are also HIV positive. Each disease speeds

up progress of the other: TB shortens the survival time of people with HIV/AIDS, killing up to half of all AIDS patients worldwide; HIV-positive people have increased likelihood of acquiring new TB infection, are more likely to develop active TB, and relapse if previously treated. Having HIV makes the diagnosis of TB more complex, and prophylactic and curative treatment for TB in HIV-positive people is more costly and problematic.

Appendices

Appendix A Maximum Likelihood Estimation for Latent Class Models: Solving 100 Swiss Francs Problem

Abstract

Sturmfels offered 100 Swiss Francs in 2005 to a conjecture, which deals with a special case of the maximum likelihood estimation for a latent class model. A positive solution for this conjecture is presented in this paper.

A.1 The conjecture and its statistics background

Sturmfels [7] proposed the following problem: Maximize the likelihood function

$$L(P) = \prod_{i=1}^{4} p_{ii}^4 \times \prod_{i \neq j} p_{ij}^2 \tag{1}$$

over the set of all 4×4-matrices $P = (p_{ij})$ whose entries are nonnegative and sum to 1 and whose rank is at most two. Based on numerical experiments by employing an EM algorithm, Sturmfels [5,7] conjectured that the matrix

$$P = \frac{1}{40} \begin{pmatrix} 3 & 3 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 \end{pmatrix}$$

is a global maximum of $L(P)$. He offered 100 Swiss francs for a rigorous proof in a postgraduate course held at ETH Zürich in 2005.

Partial results were given in the paper [2], where the general statistical background for this problem is also presented. This problem is a special case of the maximum likelihood estimation for a latent class model. More precisely, by following [2], let $(X_1, \ldots, X_d)$ be a discrete multivariate random vector where each X_j takes value from a finite state set $S_j = \{1, \ldots, s_j\}$. Let $\Omega = \otimes_{j=1}^{d} S_j$ be the sample space. For each $(x_1, \ldots, x_d) \in \Omega$, the joint probability mass function of $(X_1, \ldots, X_d)$ is denoted as

$$p(x_1, \ldots, x_d) = P\{(X_1, \ldots, X_d) = (x_1, \ldots, x_d)\}.$$

The variables $X_1, \ldots, X_d$ may not be mutually independent generally. By introducing an unobservable variable H defined on the set $[r] = \{1, \ldots r\}$, $X_1, \ldots, X_d$ become mutually independent. The joint probability mass function in the newly formed model is

$$\begin{aligned} p(x_1, \ldots, x_d, h) &= P\{(X_1, \ldots, X_d, H) = (x_1, \ldots, x_d, h)\} \\ &= p(x_1|h) \cdots p(x_d|h)\lambda_h \end{aligned}$$

where λ_h is the marginal probability of $P\{H = h\}$ and $p(x_j|h)$ is the conditional probability $P\{X_j = x_j|H = h\}$. We denote this new r-class mixture model by $\mathcal{H}$. The marginal distribution of $(X_1, \ldots, X_d)$ in $\mathcal{H}$ is given by the probability mass function (which is also called *accounting equations* [3])

$$p(x_1, \ldots, x_d) = \sum_{h \in [r]} p(x_1, \ldots, x_d, h) = \sum_{h \in [r]} p(x_1|h) \cdots p(x_d|h)\lambda_h.$$

In practice, a collection of samples from Ω are observed. For each $(x_1, \ldots, x_d)$, let $n(x_1, \ldots, x_d) \in \mathbb{N}$ be the number of observed occurrences of $(x_1, \ldots, x_d)$ in the samples. While the parameters $p(x_1|h), \cdots, p(x_d|h)$, $\lambda_h, p(x_1, \ldots, x_d)$ are unknown.

The maximum likelihood estimation problem is to find the model parameters that can best explain the observed data, that is, to determine the global maxima of the likelihood function

$$L(\mathcal{H}) = \prod_{(x_1,\ldots,x_d)\in\Omega} p(x_1,\ldots,x_d)^{n(x_1,\ldots,x_d)}.$$

Since each $p(x_1,\ldots,x_d)$ is nonnegative, it is equivalent but more convenient to use the log-likelihood function

$$l(\mathcal{H}) = \sum_{(x_1,\ldots,x_d)\in\Omega} n(x_1,\ldots,x_d)\ln p(x_1,\ldots,x_d), \tag{2}$$

where we define $\ln(0) = -\infty$. Finding the maxima of (2) is difficult and remains infeasible by current symbolic software [1,4]. We can only handle some special cases: small models or highly symmetric table. The 100 Swiss francs problem is the special case of $\mathcal{H}$ when $d = 2$, $S_1 = S_2 = \{\text{A,C,G,T}\}$, $s_1 = s_2 = 4$ and $r = 2$. It is related to a DNA sequence alignment problem as described in [5]. In that example, the contingency table for the observed counts of ordered pairs of nucleotides (i.e. AA, AC, AG, AT, CA, CC, $\cdots$) is

$$\begin{array}{c} \\ \text{A} \\ \text{C} \\ \text{G} \\ \text{T} \end{array} \begin{array}{c} \begin{array}{cccc} \text{A} & \text{C} & \text{G} & \text{T} \end{array} \\ \begin{pmatrix} 4 & 2 & 2 & 2 \\ 2 & 4 & 2 & 2 \\ 2 & 2 & 4 & 2 \\ 2 & 2 & 2 & 4 \end{pmatrix} \end{array}.$$

So the likelihood function (2) in this example is exactly (1).

Even for this simple case, the problem is highly non-trivial and remains sur-

prisingly difficult. We know that the global maxima must exist, as the region of the parameters is closed. By using EM algorithm or Newton-Raphson method and starting from suitable points, one can find some local maxima of the likelihood function. However, the global maximum property is not guaranteed.

Our paper is organized as follows. We first derive some general properties for optimal solutions in Section A.2, then we prove the conjecture in Section A.3. In Section A.4, we make some comments on why Gröbner basis technique does not work efficiently, and suggest several new conjectures in more general cases.

A.2 General Properties

We focus on general $n \times n$ matrices $P = (p_{ij})$ in this section. For convenience we scale each entry of P by n^2 so the entries sum to n^2, and take square root of the original likelihood function. So we may assume that

$$L(P) = \prod_{i=1}^{n} p_{ii}^2 \times \prod_{i \neq j} p_{ij}. \tag{3}$$

The problem is

$$\begin{aligned} &\text{Maximize:} && \text{L(P)} \\ &\text{Subject to:} && \sum_{1 \leq i,j \leq n} p_{ij} = n^2, \text{ and} \\ & && p_{ij} \geq 0,\ 1 \leq i, j \leq n. \end{aligned}$$

Suppose $P = (p_{ij})_{n \times n}$ is a global maximum of $L(P)$. Apparently P cannot be the following $n \times n$ matrix

$$J = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix},$$

as we have many other matrices (for instance, the resulting matrix by setting $s = 2, t = 1$ for the matrix in Conjecture 12 and then scaling by n^2) with larger likelihood value.

Since the function (3) is a continuous function in p_{ij}'s, if one of the entries of P approaches 0, the product has to approach 0 too, as the other entries are bounded by n^2. Hence the optimal solutions must occur in interior points and we don't need to worry about the boundary where some $p_{ij} = 0$.

In the following discussion, we assume that $P \neq J$ and all its entries are positive. We show that P must have certain symmetry properties.

Lemma 1 *For an optimal solution P, its row sums and column sums must all equal n.*

Proof. Let $\sum_{j=1}^{n} p_{ij} = s_i$. Then $\sum_{i=1}^{n} s_i = n^2$ and $\prod_i s_i \leq n^n$ with equality if and only if $s_i = n$ for all i. Let $\bar{p}_{ij} = \frac{n}{s_i} p_{ij}$ and $\bar{P} = (\bar{p}_{ij})_{n \times n}$. Then $\text{rank}(\bar{P}) = \text{rank}(P)$ and $\sum_{i,j} \bar{p}_{ij} = n^2$. However,

$$L(\bar{P}) = \left(\frac{n^n}{\prod_i s_i} \right)^{n+1} \cdot L(P) \geq L(P)$$

with equality if and only if $s_i = n$ for all i. Since P is a global maximum, $L(\bar{P}) \leq L(P)$. Therefore each row sum equals n. Similarly, each column sum equals n as well. □

We shall express P in a form that involves fewer variables and has no rank constraint. Since P has rank at most two, by singular value decomposition theorem,

there are column vectors u_1, u_2, v_1 and v_2 of length n such that

$$P = \sigma_1 u_1 v_1^t + \sigma_2 u_2 v_2^t$$

for some nonnegative numbers σ_1 and σ_2. By Proposition 1, P has equal row and column sums, so P has the vectors $(1, 1, \ldots, 1)$ and $(1, 1, \ldots, 1)^t$ as its left and right eigenvectors both with eigenvalue 1. Hence we may assume that $\sigma_1 = 1$ and $u_1 = v_1 = (1, 1, \ldots, 1)^t$. Let $v_2 = (a_1, a_2, \ldots, a_n)^t$ and $\sigma_2 u_2 = (b_1, b_2, \ldots, b_n)^t$. Then P has the form

$$P = J + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} (a_1, a_2, \ldots, a_n) = \begin{pmatrix} 1 + a_1 b_1 & \cdots & 1 + a_n b_1 \\ \vdots & 1 + a_i b_j & \vdots \\ 1 + a_1 b_n & \cdots & 1 + a_n b_n \end{pmatrix}.$$

In this form, P has rank at most two. Also, the condition $\sum_{ij} p_{ij} = n^2$ becomes

$$\sum_{i=1}^{n} a_i \cdot \sum_{i=1}^{n} b_i = 0. \tag{4}$$

We have transformed the original problem to the following optimization problem:

$$\begin{aligned} &\text{Maximize:} \quad l(P) = 2 \sum_{i=1}^{n} \ln(1 + a_i b_i) + \sum_{i \neq j} \ln(1 + a_i b_j) \\ &\text{Subject to:} \quad \text{equation (4) and } 1 + a_i b_j > 0,\ 1 \leq i, j \leq n. \end{aligned}$$

The Lagrangian function would be

$$\Lambda(P, \lambda) = l(P) + \lambda \sum_{i=1}^{n} a_i \cdot \sum_{i=1}^{n} b_i$$

where $\lambda \in \mathbb{R}$. Any local extrema must satisfy

$$\frac{\partial \Lambda(P,\lambda)}{\partial a_i} = \sum_{j=1}^{n} \frac{b_j}{1+a_i b_j} + \frac{b_i}{1+a_i b_i} + \lambda \sum_{j=1}^{n} b_j = 0, \quad 1 \le i \le n, \tag{5}$$

and

$$\frac{\partial \Lambda(P,\lambda)}{\partial b_j} = \sum_{i=1}^{n} \frac{a_i}{1+a_i b_j} + \frac{a_j}{1+a_j b_j} + \lambda \sum_{i=1}^{n} a_i = 0, \quad 1 \le j \le n. \tag{6}$$

By Lemma 1, for an optimal solution P, its row sums and column sums must be all equal to n. This means that

$$\sum_{i=1}^{n} a_i = 0, \tag{7}$$

and

$$\sum_{i=1}^{n} b_i = 0. \tag{8}$$

Plugging (7) and (8) into (5) and (6) respectively, we obtain the following lemma.

Lemma 2 *A global maximum P must satisfy*

$$\sum_{j=1}^{n} \frac{b_j}{1+a_i b_j} + \frac{b_i}{1+a_i b_i} = 0, \quad 1 \le i \le n, \tag{9}$$

and

$$\sum_{i=1}^{n} \frac{a_i}{1+a_i b_j} + \frac{a_j}{1+a_j b_j} = 0, \quad 1 \le j \le n. \tag{10}$$

Doing some simple algebra yields

Corollary 3 *An optimal solution must satisfy*

$$\sum_{j=1}^{n} \frac{1}{1+a_i b_j} + \frac{1}{1+a_i b_i} = n+1, \quad 1 \le i \le n, \tag{11}$$

and

$$\sum_{i=1}^{n} \frac{1}{1+a_i b_j} + \frac{1}{1+a_j b_j} = n+1, \quad 1 \le j \le n. \tag{12}$$

Proof. Multiply (9) by a_i and then add $\sum_{j=1}^{n} \frac{1}{1+a_i b_j} + \frac{1}{1+a_i b_i}$ at both sides, we can get (11). □

The $2n$ equations derived by clearing denominators of the equations in Lemma 2 or Corollary 3 along with equations (7) and (8) form a system $\mathcal{S}$ of $2n+2$ polynomial equations with $2n$ unknowns, whose solution contains all global maxima. From computational view point we may solve $\mathcal{S}$ by brute force, like utilizing Gröbner basis method, and then compare the local extrema to achieve the global maximum. However current symbolic computation softwares like Magma, Maple, Singular and so on are still infeasible to compute a Gröbner basis for $n = 4$ by using the computers available to us.

Our strategy below is to prove that P should have high symmetry: firstly a_i's and b_i's are in the same order: if $a_i > a_j > 0$, then $b_i > b_j > 0$ correspondingly (Lemma 4 and 5). For the case $n = 4$ once we force $a_1 = b_1$ by scaling, we can eventually prove $a_i = b_i$ for all other i's (Lemma 7 and 9). With 4 a_i's remaining, we prove that the a_i's with the same signs must be identical. Finally one can solve the system by hand. Note that in [2] they derived results similar to Lemmas 4 and 5, but our approach is fairly different.

Lemma 4 *For every i,*

1. *$a_i = 0$ if and only if $b_i = 0$, and*

2. *$a_i > 0$ if and only if $b_i > 0$.*

Proof. (1) Plugging in $a_i = 0$ to the equation (9), we have $\sum_{j=1}^{n} b_j + b_i = 0$, thus $b_i = 0$. Similarly, if $b_i = 0$ then $a_i = 0$.

(2) Note that $g(x) = \frac{1}{x}$ is concave up in $(0, \infty)$. By Jensen's Inequality,

$$\sum_{j=1}^{n} \frac{1}{n} \cdot \frac{1}{1 + a_i b_j} \geq \frac{1}{\sum_{j=1}^{n} \frac{1}{n}(1 + a_i b_j)} = 1.$$

That is,

$$\sum_{j=1}^{n} \frac{1}{1 + a_i b_j} \geq n.$$

Compare with equation (11), we get

$$\frac{1}{1 + a_i b_i} \leq 1,$$

so $a_i b_i \geq 0$. We conclude that $a_i > 0$ if and only if $b_i > 0$. □

Lemma 5 *For i and j,*

1. *$a_i = a_j$ if and only if $b_i = b_j$, and*
2. *$a_i > a_j$ if and only if $b_i > b_j$.*

Proof. (1) Suppose $b_i = b_j$. Then, by (10),

$$\sum_{k=1}^{n} \frac{a_k}{1 + a_k b_i} + \frac{a_i}{1 + a_i b_i} = 0 \text{ and } \sum_{k=1}^{n} \frac{a_k}{1 + a_k b_j} + \frac{a_j}{1 + a_j b_j} = 0.$$

Then $\frac{a_i}{1+a_i b_i} = \frac{a_j}{1+a_j b_j}$, so $a_i = a_j$. Similarly, using (9), we have that if $a_i = a_j$ then $b_i = b_j$.

(2) Switch b_i, b_j in P to form a new matrix $\bar{P}$. Then we should have $L(P) \geq L(\bar{P})$. Note that

$$\begin{aligned} L(P) - L(\bar{P}) &= C_1 \cdot ((1+a_ib_i)^2(1+a_ib_j)(1+a_jb_i)(1+a_jb_j)^2 \\ &\quad -(1+a_ib_j)^2(1+a_ib_i)(1+a_jb_j)(1+a_jb_i)^2) \\ &= C_2 \cdot ((1+a_ib_i)(1+a_jb_j) - (1+a_ib_j)(1+a_jb_i)) \\ &= C_2 \cdot (a_ib_i + a_jb_j - a_ib_j - a_jb_i) \\ &= C_2 \cdot (a_i - a_j)(b_i - b_j) \end{aligned}$$

where C_1, C_2 are products of some entries of P, so C_1, C_2 are positive. Thus $(a_i - a_j)(b_i - b_j) \geq 0$. Note that $a_i = a_j$ if and only if $b_i = b_j$ by part(1), we conclude that $a_i > a_j$ if and only if $b_i > b_j$. □

A.3 Proof for the conjecture

We complete the proof for the conjecture in this section. From now on we focus on the case when $n = 4$. By Lemma 5, we can always assume $a_1 \geq a_2 \geq a_3 \geq a_4$ and $b_1 \geq b_2 \geq b_3 \geq b_4$. We know $a_1 \neq 0$, otherwise $P = J$. So we replace (a_1, a_2, a_3, a_4) in P by $\sqrt{\frac{a_1}{b_1}}(a_1, a_2, a_3, a_4)$ and $(b_1, b_2, b_3, b_4)^t$ by $\sqrt{\frac{b_1}{a_1}}(b_1, b_2, b_3, b_4)^t$. It turns out that $1 + \sqrt{\frac{a_1}{b_1}}a_i\sqrt{\frac{b_1}{a_1}}b_i = 1 + a_ib_j$ for any i and j, so we may always assume $a_1 = b_1$. Thus P can be expressed as the form

$$\begin{pmatrix} 1+a_1^2 & 1+a_2a_1 & 1+a_3a_1 & 1+a_4a_1 \\ 1+a_1b_2 & 1+a_2b_2 & 1+a_3b_2 & 1+a_4b_2 \\ 1+a_1b_3 & 1+a_2b_3 & 1+a_3b_3 & 1+a_4b_3 \\ 1+a_1b_4 & 1+a_2b_4 & 1+a_3b_4 & 1+a_4b_4 \end{pmatrix}. \tag{13}$$

If $a_2 \leq 0$, we then replace (a_1, a_2, a_3, a_4) in P by $(-a_4, -a_3, -a_2, -a_1)$ and $(b_1, b_2, b_3, b_4)^t$ by $(-b_4, -b_3, -b_2, -b_1)^t$. The new matrix with $-a_4 \geq -a_3 \geq 0$ has the same likelihood function as P. Thus we may assume $a_1 \geq a_2 \geq 0$. Without loss of generality, we may make the following assumption.

Assumption 6 *We can always assume the following*

1. $a_1 \geq a_2 \geq a_3 \geq a_4$ *and* $b_1 \geq b_2 \geq b_3 \geq b_4$,
2. $a_1 = b_1 > 0$, *and*
3. $a_1 \geq a_2 \geq 0$.

The results in the rest of this section are all based on Assumption 6. Our first goal is to prove $a_2 = b_2$.

Lemma 7 $a_2 = b_2$.

Proof. If one of a_2, b_2 is 0, then $a_2 = b_2 = 0$ by Lemma 4. We assume that both a_2, b_2 are nonzero.

Apply Corollary 3 to the first row of matrix (13). We have

$$\frac{2}{1+a_1^2} + \frac{1}{1+a_2a_1} + \frac{1}{1+a_3a_1} + \frac{1}{1+a_4a_1} = 5.$$

Also

$$a_1^2 + a_2a_1 + a_3a_1 + a_4a_1 = 0.$$

From above two equations we get

$$a_3a_1 \cdot a_4a_1 = f_1(a_1a_1, a_1a_2) \tag{14}$$

where f_1 is a bivariate function in x, y defined as

$$f_1(x,y) = \frac{2-x-y}{5-\frac{2}{1+x}-\frac{1}{1+y}} + x + y - 1.$$

Similarly, apply Corollary 3 to the second row of matrix (13). We get

$$\frac{1}{1+a_1b_2} + \frac{2}{1+a_2b_2} + \frac{1}{1+a_3b_2} + \frac{1}{1+a_4b_2} = 5.$$

Along with

$$a_1b_2 + a_2b_2 + a_3b_2 + a_4b_2 = 0,$$

we get

$$a_3b_2 \cdot a_4b_2 = f_1(a_2b_2, a_1b_2). \tag{15}$$

Since a_1, b_2 are nonzero, we combine equations (14) and (15) to get

$$\frac{f_1(a_1^2, a_1a_2)}{a_1^2} = \frac{f_1(a_2b_2, a_1b_2)}{b_2^2}. \tag{16}$$

Normalizing (16) we can derive a trivariate polynomial equation, say

$$f_2(a_1, a_2, b_2) = 0. \tag{17}$$

Symmetrically apply Corollary 3 to the first column and the second column 13, we get

$$\frac{f_1(a_1^2, a_1b_2)}{a_1^2} = \frac{f_1(a_2b_2, a_1a_2)}{a_2^2}. \tag{18}$$

One can see that equation (18) is obtainable by switching a_2 with b_2 in equation (16). Thus we have

$$f_2(a_1, b_2, a_2) = 0. \tag{19}$$

Subtracting (19) from (17) yields

$$f_2(a_1, a_2, b_2) - f_2(a_1, b_2, a_2) = 0.$$

Since we only switched a_2 and b_2 in polynomial f_2, there must be a factor $a_2 - b_2$ for $f_2(a_1, a_2, b_2) - f_2(a_1, b_2, a_2)$, say

$$(a_2 - b_2) f_3(a_1, a_2, b_2) = 0, \tag{20}$$

where

$$\begin{aligned} f_3(a_1, a_2, b_2) = \quad & (20a_1^4 b_2^2 + 15a_1^3 b_2 + 3a_1^2 b_2^2 + 2a_1 b_2 - 4b_2^2)a_2^2 \\ & + (3a_1^4 b_2 + 15a_1^3 b_2^2 + 2a_1^3 + 10a_1^2 b_2 + 2a_1 b_2^2 - 3a_1 - b_2)a_2 \\ & - 4a_1^4 + 2a_1^3 b_2 - a_1^2 - 3a_1 b_2 - 2. \end{aligned}$$

Thus $a_2 = b_2$ if $f_3(a_1, a_2, b_2) \neq 0$. This is true because we have some bounds

for a_1^2, a_1a_2, a_1b_2 as presented in Lemma 8 below, which can be applied to get

$$\begin{aligned} f_3(a_1,a_2,b_2) = \quad & (20a_1^4b_2^2 + 15a_1^3b_2 + 3a_1^2b_2^2 + 2a_1b_2 - 4b_2^2)a_2^2 \\ & + (3a_1^4b_2 + 15a_1^3b_2^2 + 2a_1^3 + 10a_1^2b_2 + 2a_1b_2^2 - 3a_1 - b_2)a_2 \\ & - 4a_1^4 + 2a_1^3b_2 - a_1^2 - 3a_1b_2 - 2 \\ < \quad & \frac{20}{5^4} + \frac{15}{5^3} + \frac{3}{4}a_2^2b_2^2 + \frac{2}{5}a_2b_2 - 4a_2^2b_2^2 \\ & + \frac{3}{2^25^2} + \frac{15}{5^3} + \frac{2}{2^25} + \frac{10}{5^2} + \frac{2}{5}a_2b_2 - a_2b_2 \\ & + \frac{2}{2^25} - 2 \\ < \quad & -\frac{13}{4}a_2^2b_2^2 - \frac{1}{5}a_2b_2 - \frac{549}{500} \\ < \quad & 0. \end{aligned}$$

Therefore, $f_3(a_1,a_2,b_2) \neq 0$ and $a_2 = b_2$, just as needed. □

Lemma 8

1. $a_1^2 \leq \frac{1}{2}$,

2. $0 \leq a_1a_2 \leq \frac{1}{5}$, *and*

3. $0 \leq a_1b_2 \leq \frac{1}{5}$.

Proof.(1) Let $A_i = 1 + a_1a_i$ for $i = 1,\ldots,4$, then $\sum_{i=1}^{4} A_i = 4$, $A_1 \geq A_2 \geq 1$, $A_3 \geq A_4 > 0$ and

$$\frac{2}{A_1} + \frac{1}{A_2} + \frac{1}{A_3} + \frac{1}{A_4} = 5.$$

Since

$$\frac{1}{A_3} + \frac{1}{A_4} \geq \frac{4}{A_3 + A_4} = \frac{4}{4 - A_1 - A_2},$$

we have

$$5 = \frac{2}{A_1} + \frac{1}{A_2} + \frac{1}{A_3} + \frac{1}{A_4} \geq \frac{2}{A_1} + \frac{1}{A_2} + \frac{4}{4 - A_1 - A_2}. \tag{21}$$

Let

$$g(A_2) = \frac{1}{A_2} + \frac{4}{4 - A_1 - A_2},$$

where g is a function in $\mathbb{R}[x]$. Then

$$\frac{\partial g(A_2)}{\partial A_2} = -\frac{1}{A_2^2} + \frac{4}{(4 - A_1 - A_2)^2}.$$

Note that $A_1 \geq A_2 \geq 1$, thus $4 - A_1 - A_2 \leq 2$ and $\frac{\partial g(A_2)}{\partial A_2} \geq 0$. Therefore $g(A_2) \geq g(1)$ for $A_2 \geq 1$, that is,

$$\frac{1}{A_2} + \frac{4}{4 - A_1 - A_2} \geq 1 + \frac{4}{3 - A_1}.$$

Hence by inequality (21),

$$5 \geq \frac{2}{A_1} + \frac{1}{A_2} + \frac{4}{4 - A_1 - A_2} \geq \frac{2}{A_1} + 1 + \frac{4}{3 - A_1}.$$

We get $2A_1^2 - 5A_1 + 3 \leq 0$, i.e. $1 \leq A_1 \leq \frac{3}{2}$. Thus $a_1^2 \leq \frac{1}{2}$.

(2) Assume $A_2 = 1 + a_1 a_2 > \frac{6}{5}$. Then $g(A_2) > g(\frac{6}{5})$. That is,

$$5 \geq \frac{2}{A_1} + \frac{1}{A_2} + \frac{4}{4 - A_1 - A_2} > \frac{2}{A_1} + \frac{5}{6} + \frac{4}{\frac{14}{5} - A_1}.$$

The solution set of A_1 is $(-\infty, 0) \cup (\frac{28}{25}, \frac{6}{5}) \cup (\frac{14}{5}, \infty)$. Note that $A_1 > 0$ and $A_1 = 1 + a_1^2 \leq \frac{3}{2}$, we then get $\frac{28}{25} < A_1 < \frac{6}{5}$, which contradicts with $A_1 \geq A_2$. Thus $A_2 \leq \frac{6}{5}$ and $0 \leq a_1 a_2 \leq \frac{1}{5}$.

(3) This result is followed by letting $A_1 = 1 + a_1^2$ and $A_i = 1 + a_1 b_i$ for $i \geq 2$. The above proofs in part (1) and (2) remain good. □

Lemma 9 *$a_i = b_i$ for $i = 3, 4$.*

Proof. Let $A_i = 1 + a_i b_1$ for $i = 1, \ldots, 4$. Then

$$\sum_{i=1}^{4} A_i = 4$$

and

$$\frac{2}{A_1} + \frac{1}{A_2} + \frac{1}{A_3} + \frac{1}{A_4} = 5.$$

By the above two equations, since $A_3 \geq A_4$, we can derive explicit expression for A_3, A_4 in the variables A_1, A_2, say $A_3 = h_1(A_1, A_2)$ and $A_4 = h_2(A_1, A_2)$. If we let $B_i = 1 + a_1 b_i$, we can get $B_3 = h_1(B_1, B_2)$ and $B_4 = h_2(B_1, B_2)$ in a similar manner. Note that $A_1 = B_1$ and $A_2 = 1 + a_2 b_1 = 1 + b_2 a_1 = B_2$, we deduce that $A_i = B_i$ for $i = 3, 4$. Since $a_1 = b_1 > 0$, $a_i = b_i$ for $i = 3, 4$. □

By Lemmas 7 and 9, we have $a_i = b_i$ for all i. Hence P can be expressed as

$$P = \begin{pmatrix} 1 + a_1^2 & 1 + a_2 a_1 & 1 + a_3 a_1 & 1 + a_4 a_1 \\ 1 + a_1 a_2 & 1 + a_2^2 & 1 + a_3 a_2 & 1 + a_4 a_2 \\ 1 + a_1 a_3 & 1 + a_2 a_3 & 1 + a_3^2 & 1 + a_4 a_3 \\ 1 + a_1 a_4 & 1 + a_2 a_4 & 1 + a_3 a_4 & 1 + a_4^2 \end{pmatrix}$$

where

$$\sum_{i=1}^{4} a_i = 0. \tag{22}$$

By Corollary 3 we have the following system of equations

$$\begin{cases} \dfrac{2}{1+a_1^2}+\dfrac{1}{1+a_2a_1}+\dfrac{1}{1+a_3a_1}+\dfrac{1}{1+a_4a_1}=5, \\ \dfrac{1}{1+a_1a_2}+\dfrac{2}{1+a_2^2}+\dfrac{1}{1+a_3a_2}+\dfrac{1}{1+a_4a_2}=5, \\ \dfrac{1}{1+a_1a_3}+\dfrac{1}{1+a_2a_3}+\dfrac{2}{1+a_3^2}+\dfrac{1}{1+a_4a_3}=5, \\ \dfrac{1}{1+a_1a_4}+\dfrac{1}{1+a_2a_4}+\dfrac{1}{1+a_3a_4}+\dfrac{2}{1+a_4^2}=5. \end{cases} \tag{23}$$

With (22) and (23), we claim that

Lemma 10 *$a_i = a_j$ if $a_ia_j > 0$.*

Proof. Let

$$F(x)=\frac{1}{1+a_1x}+\frac{1}{1+a_2x}+\frac{1}{1+a_3x}+\frac{1}{1+a_4x}+\frac{1}{1+x^2}-5=0.$$

Normalizing $F(x)$ yields a polynomial (the numerator) of degree 6 in x whose constant is 0 and whose coefficient of the term x is $\sum_{i=1}^{4} a_i = 0$. So $a_1, a_2, a_3, a_4, 0, 0$ are all the zeros of $F(x)$. Suppose there exists consecutive i, j such that $a_i > a_j > 0$ (or $a_j < a_i < 0$ respectively). Then $F(x)$ is continuous in the interval $(-\frac{1}{a_j}, -\frac{1}{a_i})$. Note that

$$\lim_{x\to -\frac{1}{a_j}^+} F(x)=\infty \text{ and } \lim_{x\to -\frac{1}{a_i}^-} F(x)=-\infty.$$

There must be a zero lying in $(-\frac{1}{a_j}, -\frac{1}{a_i})$, say a_0. Then $a_0 < -\frac{1}{a_i}$ (or $a_0 > -\frac{1}{a_j}$ respectively), i.e. $1 + a_ia_0 < 0$ (or $1 + a_ja_0 < 0$ respectively). Since $a_0 \neq 0$, x_0 must be one of a_k, $k = 1, \ldots, 4$. Thus $1 + a_ia_0$ (or $1 + a_ja_0$, respectively) is an entry in matrix P, contradicting the fact that each entry of P is positive. Therefore if i, j are consecutive and $a_ia_j > 0$, we must have $a_i = a_j$. Hence $a_ia_j > 0$ implies $a_i = a_j$ for any i, j. □

With Lemma 10 it is handy to solve the system (23). Under Assumption (6) there are only 4 possible patterns of signs for (a_1, a_2, a_3, a_4). If the signs are $(+,+,+,-)$, then $a_1 = a_2 = a_3 = -\frac{1}{3}a_4$. Substitute this to any equation in (23) yields $a_1 = a_2 = a_3 = \frac{1}{\sqrt{15}}$ and $a_4 = -\frac{3}{\sqrt{15}}$. The matrix would be

$$P_1 = \begin{pmatrix} \frac{16}{15} & \frac{16}{15} & \frac{16}{15} & \frac{4}{5} \\ \frac{16}{15} & \frac{16}{15} & \frac{16}{15} & \frac{4}{5} \\ \frac{16}{15} & \frac{16}{15} & \frac{16}{15} & \frac{4}{5} \\ \frac{4}{5} & \frac{4}{5} & \frac{4}{5} & \frac{8}{5} \end{pmatrix}.$$

For the case when the signs are $(+,+,-,-)$, we get $a_1 = \frac{1}{\sqrt{5}}$ and the matrix would be

$$P_2 = \begin{pmatrix} \frac{6}{5} & \frac{6}{5} & \frac{4}{5} & \frac{4}{5} \\ \frac{6}{5} & \frac{6}{5} & \frac{4}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{4}{5} & \frac{6}{5} & \frac{6}{5} \\ \frac{4}{5} & \frac{4}{5} & \frac{6}{5} & \frac{6}{5} \end{pmatrix}.$$

When the signs are $(+,+,0,-)$, $a_1 = \frac{1}{2\sqrt{2}}$, and the matrix would be

$$P_3 = \begin{pmatrix} \frac{9}{8} & \frac{9}{8} & 1 & \frac{3}{4} \\ \frac{9}{8} & \frac{9}{8} & 1 & \frac{3}{4} \\ 1 & 1 & 1 & 1 \\ \frac{3}{4} & \frac{3}{4} & 1 & \frac{3}{2} \end{pmatrix}.$$

And when the signs are $(+,0,0,-)$, $a_1 = \frac{1}{\sqrt{3}}$ and the matrix would be

$$P_4 = \begin{pmatrix} \frac{4}{3} & 1 & 1 & \frac{2}{3} \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{2}{3} & 1 & 1 & \frac{4}{3} \end{pmatrix}.$$

The matrices obtaining local maximum of the likelihood function must be among the matrices above. We conclude that matrix P_2 obtains the global maximum. Finally, multiplying matrix P_2 by $\frac{1}{16}$ yields

$$P = \frac{1}{40} \begin{pmatrix} 3 & 3 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 \end{pmatrix}.$$

A.4 Some comments on general cases and discussion

A natural question arises here is that why Gröbner basis technique [6], the most powerful tool for solving systems of polynomial equations, does not work efficiently for this particular problem. We get a polynomial $(a_2 - b_2) f_3(a_1, a_2, b_2)$ in the proof for Lemma (7). Our approach is to justify the factor $f_3(a_1, a_2, b_2)$, a trivariate polynomial with 17 terms, is nonzero by applying some bounds from Lemma 8, so that we can derive the simplest equation $a_2 - b_2 = 0$. In the proof we used the fact that we are looking only for real solutions. However, a Gröbner basis encodes both real and complex solutions. It is possible that $f_3(a_1, a_2, b_2)$ is zero for some complex solutions. The locus of all solutions may be much more complicated than that of real solutions,

hence the Gröbner basis is much more time consuming to compute.

The limitation of our work is that the arguments in the last section only work for $n = 4$. For $n > 4$ we are still lack of an efficient method. However, our methods can still be applied to general likelihood functions. In the following, we illustrate for a few examples.

Conjecture 11 *For given $0 < t < s$ where t, s are two integers, among the set of all non-negative 4×4 matrices whose rank is at most 2 and whose entries sum to 1, the matrix*

$$P = \frac{1}{4s + 12t} \begin{pmatrix} \frac{s+t}{2} & \frac{s+t}{2} & t & t \\ \frac{s+t}{2} & \frac{s+t}{2} & t & t \\ t & t & \frac{s+t}{2} & \frac{s+t}{2} \\ t & t & \frac{s+t}{2} & \frac{s+t}{2} \end{pmatrix}$$

is a global maximum for the following likelihood function

$$L(P) = \prod_{i=1}^{4} p_{ii}^{s} \times \prod_{i \neq j} p_{ij}^{t}. \tag{24}$$

The results in Section A.2 remain good for this likelihood function. The equation (10) becomes

$$\frac{b_1}{1 + a_i b_1} + \frac{b_2}{1 + a_i b_2} + \frac{b_3}{1 + a_i b_3} + \frac{b_4}{1 + a_i b_4} + \frac{(\frac{s}{t} - 1)a_i}{1 + a_i b_i} = 0.$$

But the bounds in Lemma 8 involve the fraction $\frac{s}{t}$ and become complicated. A similar equation to (20) can be derived, but the nonzero factor is difficult to claim. Hopefully we may also prove $a_2 = b_2$. Then $a_3 = b_3$ and $a_4 = b_4$ can be derived in a similar manner to Lemma 9. So does Lemma 10. Finally we can find 4 local extrema and

need only compare them to obtain the global maximum. In the case when the signs of (a_1, a_2, a_3, a_4) are $(+, +, +, -)$, we have the equation

$$a_1^2((3s + 9t)a_1^2 - (s - t)) = 0.$$

Thus $a_1 = \sqrt{\frac{s-t}{3s+9t}}$, and the matrix would be

$$P_1 = \begin{pmatrix} \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{12t}{3s+9t} \\ \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{12t}{3s+9t} \\ \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{4s+8t}{3s+9t} & \frac{12t}{3s+9t} \\ \frac{12t}{3s+9t} & \frac{12t}{3s+9t} & \frac{12t}{3s+9t} & \frac{12s}{3s+9t} \end{pmatrix}.$$

In the case when the signs are $(+, +, -, -)$, we get $a_1 = \sqrt{\frac{s-t}{s+3t}}$ and the matrix would be

$$P_2 = \begin{pmatrix} \frac{2s+2t}{s+3t} & \frac{2s+2t}{s+3t} & \frac{4t}{s+3t} & \frac{4t}{s+3t} \\ \frac{2s+2t}{s+3t} & \frac{2s+2t}{s+3t} & \frac{4t}{s+3t} & \frac{4t}{s+3t} \\ \frac{4t}{s+3t} & \frac{4t}{s+3t} & \frac{2s+2t}{s+3t} & \frac{2s+2t}{s+3t} \\ \frac{4t}{s+3t} & \frac{4t}{s+3t} & \frac{2s+2t}{s+3t} & \frac{2s+2t}{s+3t} \end{pmatrix}. \tag{25}$$

One can prove that $L(P_1) < L(P_2)$ by some calculus technique, for example, taking the partial derivative of $\frac{L(P_1)}{L(P_2)}$ with respect to s. In similar approaches one can also show that $L(P_3) < L(P_2)$ and $L(P_4) < L(P_2)$ where P_3, P_4 are the corresponding matrices for the cases when signs are $(+, +, 0, -)$ and $(+, 0, 0, -)$ respectively. Thus the matrix in (25) is a global maximum.

More generally, let $(u)_{l_1 \times l_2}$ be a block matrix with every entry being u where $l_1 \times l_2 \in \mathbb{N}^2$ and $u > 0$.

Conjecture 12 *Let $n \geq 2$ and $0 < t < s$. Then the matrix*

$$P = \frac{1}{ns + (n-1)nt} \begin{pmatrix} (\frac{s-t}{\lceil \frac{n}{2} \rceil} + t)_{\lceil \frac{n}{2} \rceil \times \lceil \frac{n}{2} \rceil} & (t)_{\lceil \frac{n}{2} \rceil \times \lfloor \frac{n}{2} \rfloor} \\ (t)_{\lfloor \frac{n}{2} \rfloor \times \lceil \frac{n}{2} \rceil} & (\frac{s-t}{\lfloor \frac{n}{2} \rfloor} + t)_{\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor} \end{pmatrix}$$

is a global maximum for the likelihood function L(P) in (24).

Conjecture 13 *Let $n \geq 2$ and $0 < s \leq t$. Then the matrix*

$$P = \begin{pmatrix} \frac{2s}{n^2(s+t)} & \frac{1}{n^2} & \cdots & \frac{1}{n^2} & \frac{2t}{n^2(s+t)} \\ \frac{1}{n^2} & \frac{1}{n^2} & \cdots & \frac{1}{n^2} & \frac{1}{n^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n^2} & \frac{1}{n^2} & \cdots & \frac{1}{n^2} & \frac{1}{n^2} \\ \frac{2t}{n^2(s+t)} & \frac{1}{n^2} & \cdots & \frac{1}{n^2} & \frac{2s}{n^2(s+t)} \end{pmatrix}$$

is a global maximum for the likelihood function L(P) in (24).

Bibliography

[1] F. Catanese, S. Hoşten, A. Khetan and B. Sturmfels, The maximum likelihood degree, *American Journal of Mathematics* 128 (2006), 671-697.

[2] S. Fienberg, P. Hersh, A. Rinaldo and Y. Zhou, Maximum likelihood estimation in latent class models for contingency table data, to appear as a chapter in *Algebraic and geometric methods in statistics*, Cambridge University Press.

[3] N.W. Henry and P.F. Lazarfeld, *Latent Structure Analysis*, Houghton Mufflin Company, 1968.

[4] S. Hoşten, A. Khetan and B. Sturmfels, Solving the likelihood equations, *Foundations of Computational Mathematics* 5 (2005), 389-407.

[5] L. Pachter and B. Sturmfels, *Algebraic Statistics for Computational Biology*, Cambridge University Press, 2005.

[6] B. Sturmfels, *Solving Systems of Polynomial Equations*, CBMS Regional Conference Series in Mathematics, vol 97, Amer. Math. Society, Providence, 2002.

[7] B. Sturmfels, Open problems in Algebraic Statistics, in *Emerging Applications of Algebraic Geometry*, (editors M. Putinar and S. Sullivant), I.M.A. Volumes in Mathematics and its Applications, 149, Springer, New York, 2008, pp. 351-364.

Appendix B Computing Irreducible Decomposition of Monomial Ideals

Abstract

The paper presents two algorithms for finding irreducible decomposition of monomial ideals. The first one is recursive, derived from staircase structures of monomial ideals. This algorithm has a good performance for highly non-generic monomial ideals. The second one is an incremental algorithm, which computes decompositions of ideals by adding one generator at a time. Our analysis shows that the second algorithm is more efficient than the first one for generic monomial ideals. Furthermore, the time complexity of the second algorithm is at most $O(n^2p\ell)$ where n is the number of variables, p is the number of minimal generators and ℓ is the number of irreducible components. Another novelty of the second algorithm is that, for generic monomial ideals, the intermediate storage is always bounded by the final output size which may be exponential in the input size.

B.1 Introduction

Monomial ideals provide ubiquitous links between combinatorics and commutative algebra [24, 16]. Though simple they carry plentiful algebraic and geometric information of general ideals. Our interest in monomial ideals is motivated by a paper of [9], where they studied the connection between the structure of monomial basis and the geometric structure of the solution sets of zero-dimensional polynomial ideals. Irreducible decomposition of monomial ideals is a basic computational problem and it finds applications in several areas, ranging from pure mathematics to computational

biology, see for example [12] for computing integer programming gaps, [3] for computing tropical convex hulls, [22] for finding the joins and secant varieties of monomial ideals, [2] for partition of a simplicial complex, [19] for solving the Frobenius problem, and [13] for modeling gene networks.

We are interested in efficient algorithms for computing irreducible decomposition of monomial ideals. There are a variety of algorithms available in the literature. The so-called splitting algorithm: Algorithm 3.1.2 in [23] is not efficient on large scale monomial ideals. [17] gives two algorithms: one is based on Alexander duality [14], and the other is based on Scarf complex [4]. [18] improves the Scarf complex method by a factor of up to more than 1000. Recently, [20] proposed several slicing algorithms based on various strategies.

Our goals in this paper are to study the structure of monomial ideals and present two new algorithms for irreducible decomposition. We first observe some staircase structural properties of monomial bases in Section B.4. The recursive algorithm presented in Section B.5 is based on these properties, which allow decomposition of monomial ideals recursively from lower to higher dimensions. This algorithm was presented as posters in ISSAC 2005 and in the workshop on Algorithms in Algebraic Geometry at IMA in 2006. Our algorithm was recently generalized by [20] where several cutting strategies were developed and our algorithm corresponds to the minimum strategy there. Also, the computational experiments there shows that our algorithm has good performance for most cases, especially for highly non-generic monomial ideals.

Our second algorithm is presented in Section B.6. It can be viewed as an improved Alexander dual method ([14, 17]). It is incremental based on some distribution rules for "+" and "$\cap$" operations of monomial ideals. We maintain an output list of irreducible components, and at each step we add one generator and update the

output list. In [17], there is no specific criterion for selecting candidates that need to be updated, and the updating process is inefficient too. Our algorithm avoids these two deficiencies. Our analysis in Section B.7 shows that the second algorithm works more efficiently than the first algorithm for generic monomial ideals. We prove that, for generic monomial ideals, the intermediate storage size (ie. number of irreducible components at each stage) is always bounded by the final output size, provided that the generators are added in lex order. This enables us to show that the time complexity of the second algorithm is at most $O(n^2 p\ell)$ where n is the number of variables, p is the number of minimal generators and ℓ is the number of irreducible components.

In Section B.2, we present some notations and introductory materials on monomial ideals. In Section B.3 we discuss tree representations and operations of monomial ideals.

B.2 Monomial Ideals

We refer the reader to the books of [5] for background in algebraic geometry and commutative algebra, and to the monograph [16] for monomial ideals and their combinatorial properties.

Let $\mathbb{K}$ be a field and $\mathbb{K}[X]$, the polynomial ring over $\mathbb{K}$ in n indeterminates $X = x_1, \ldots, x_n$. For a vector $\alpha = (a_1, \ldots, a_n) \in \mathbb{N}^n$, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of nonnegative integers, we set

$$X^\alpha = x_1^{a_1} \ldots x_n^{a_n},$$

which is called a **monomial**. Thus monomials in n variables are in $1-1$ correspondence with vectors in $\mathbb{N}^n$. Suppose $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ are two

vectors in $\mathbb{N}^n$, we say

$$\alpha \leq \beta \text{ if } a_j \leq b_j \text{ for all } 1 \leq j \leq n.$$

This defines a partial order on $\mathbb{N}^n$, which corresponds to division order for monomials since $x^\alpha | x^\beta$ if and only if $\alpha \leq \beta$. We say

$$\alpha < \beta \text{ if } \alpha \leq \beta \text{ but } \alpha \neq \beta.$$

Also we define

$$\alpha \prec \beta \text{ if } a_j < b_j \text{ for all } 1 \leq j \leq n.$$

Then $\alpha \not\prec \beta$ means that $a_j \geq b_j$ for at least one j.

An ideal $I \subset \mathbb{K}[X]$ is called a **monomial ideal** if it is generated by monomials. Dickson's Lemma states that every monomial ideal in $\mathbb{K}[X]$ has a unique minimal set of monomial generators, and this set is finite. Denote this set to be $\text{Min}(I)$, that is,

$$\text{Min}(I) = \{X^\alpha \in I : \text{ there is no } X^\beta \in I \text{ such that } \beta < \alpha\}.$$

A monomial ideal I is called **Artinian** if I contains a power of each variable, or equivalently, if the quotient ring $\mathbb{K}[X]/I$ has finite dimension as vector space over $\mathbb{K}$. For convenience of notations, we define

$$x_i^\infty = 0, \quad 1 \leq i \leq n.$$

By adding infinity power of variables if necessary, a non-Artinian monomial ideal can be treated like an Artinian monomial ideal. For example, $I = \langle x^2y^3 \rangle = \langle x^\infty, x^2y^3, y^\infty \rangle$. Instead of adding infinity powers, we can also add powers $x_i^{c_i}$ where c_i is a sufficiently

large integer, say larger than the largest degree of x_i in all the monomials in $\mathrm{Min}(I)$. Then the irreducible components of the original ideal are in 1-1 correspondence to those of the modified Artinian ideal; See Exercise 5.8 in [16] or Proposition 3 in [20]. In our algorithms belows, we will use infinity powers, but in the proofs of all the results, we will use powers $x_i^{c_i}$.

An ideal $J \subset \mathbb{K}[X]$ is called **irreducible** if it can not be expressed as the intersection of two strictly larger ideals in $\mathbb{K}[X]$. That is, $J = J_1 \cap J_2$ implies that $J = J_1$ or $J = J_2$. A monomial ideal I is irreducible if and only if I is of the form

$$m^\beta = \langle x_1^{b_1}, \ldots, x_n^{b_n} \rangle$$

for some vector $\beta = (b_1, \ldots, b_n) \in \overline{\mathbb{N}}^n$ where $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\} \setminus \{0\}$. Thus irreducible monomial ideals are in 1-1 correspondence with $\beta \in \overline{\mathbb{N}}^n$.

An **irreducible decomposition** of a monomial ideal I is an expression of the form

$$I = m^{\beta_1} \cap \cdots \cap m^{\beta_r} \tag{26}$$

where $\beta_1, \ldots, \beta_r \in \overline{\mathbb{N}}^n$. Since the polynomial ring $\mathbb{K}[X]$ is Noetherian, every ideal can be written as irredundant intersection of irreducible ideals. Such an intersection is not unique for a general ideal, but unique for a monomial ideal. We say that the irreducible decomposition (26) is **irredundant** if none of the components can be dropped from the right hand side. If (26) is irredundant, then the ideals $m^{\beta_1}, \ldots, m^{\beta_r}$ are called **irreducible components** of I. We denote by $\mathrm{Irr}(I)$ the set of exponents of irreducible components of I, that is,

$$\mathrm{Irr}(I) = \{\beta_1, \ldots, \beta_r\}.$$

By this notation, we have

$$I = \bigcap_{\beta \in \mathrm{Irr}(I)} m^{\beta}.$$

Note that, for two vectors α and β,

$$X^{\alpha} \in m^{\beta} \text{ if and only if } \alpha \not\prec \beta,$$

and

$$m^{\alpha} \subset m^{\beta} \text{ if and only if } \beta \leq \alpha.$$

A monomial ideal I is called **generic** if no variable x_i appears with the same non-zero exponent in two distinct minimal generators of I. This definition comes from [4]. For example,

$$I_1 = \langle x^4, y^4, x^3y^2z, xy^3z^2, x^2yz^3 \rangle$$

is generic, but

$$I_2 = \langle x^4, y^4, x^3y^2z^2, xy^3z^2, x^2yz^3 \rangle$$

is non-generic, as z^2 appears in two generators. Loosely speaking, we can say I_2 is nearly generic, but

$$I_3 = \langle xy, yz, xz, z^2 \rangle$$

is highly non-generic. Previous algorithms [17, 18] behave very different for generic monomial ideals and highly non-generic monomial ideals. For example, the Scarf complex method works more efficient when dealing with generic monomial ideals [17].

In the following sections, we always assume that we are given the minimal

generating set of a monomial ideal. Though our algorithms work for monomial ideals given by an arbitrary set of generators, it will be more efficient if the generators are made minimal first.

B.3 Tree Representation and Operations

Note that monomials are represented by vectors in $\mathbb{N}^n$ and irreducible components are represented by vectors in $\overline{\mathbb{N}}^n$. To efficiently represent a collect of vectors, we use a tree structure. This is used in [9, 17]. This data structure is also widely used in computer science, where it is called a trie.

Tree representation. First we want to define the orderings on $\mathbb{N}^n$ or $\overline{\mathbb{N}}^n$. Suppose $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$ are two vectors in $\mathbb{N}^n$ or $\overline{\mathbb{N}}^n$, and the variable ordering is $x_1 < \cdots < x_n$ in $\mathbb{K}[X]$. We say $\alpha <_{lex} \beta$ if $a_j = b_j$ for $k+1 \leq j \leq n$, but $a_k < b_k$ for some $1 \leq k \leq n$.

Next, suppose $S \subset \mathbb{N}^n$ is a set of vectors corresponding to the generators of a monomial ideal $I \subset \mathbb{K}[X]$. We represent S as a rooted tree $\mathcal{T}$ of height n in a natural way. The tree should have $|S|$ leaves and the unique path of the tree from the root to a leaf represents a vector in S. Precisely, to represent a vector $\alpha = (a_1, \ldots, a_n)$, we label all the nodes except the root of the path simply by $a_n, \ldots, a_1$ in the order from the root to the leaf. We regard the root as being at height 0. For two vectors $\alpha = (a_1, \ldots, a_n)$ and $\beta = (b_1, \ldots, b_n)$, if $a_j = b_j$ for $k+1 \leq j \leq n$ but $a_k \neq b_k$, then α and β share their corresponding path until height $n-k$. After that their children are listed in increasing order with respect to their coordinates. Figure 1 is the tree representation for $I = \langle x^4, y^4, x^3y^2z^2, xy^3z^2, x^2yz^3 \rangle$ with variable order $x < y < z$.

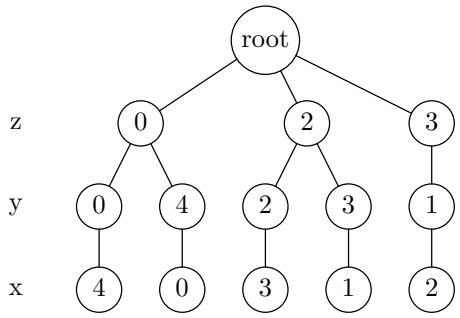


Figure 1: An example of tree representation.

The tree representation for a set of irreducible components could be constructed in a similar manner.

To perform the operations on sets of vectors, we need only perform on trees. We need three basic tree operations: Merge, MinMerge and MaxMerge.

Merge. Given q rooted trees $\mathcal{T}_1, \ldots, \mathcal{T}_q$ with the same height, merge them to form one rooted tree with the same height. Here we simply put the paths from all the trees together with repetition ignored (actually no repeated paths occur in our algorithms). We stress that no reduction work is performed under this operation.

MinMerge. We use MinMerge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$ to represent the set of minimal elements in Merge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$. For two vectors α, β in Merge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$, if $\alpha \leq \beta$, ie. $x^\alpha | x^\beta$, then the path for β should be removed in this operation. The purpose is to find the minimal generating set for the ideal $I_1 + \cdots + I_q$ where $\mathcal{T}_i$ is the tree representation for I_i.

MaxMerge. Similarly, the set of maximal elements in Merge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$ is represented by MaxMerge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$. If $\alpha \leq \beta$, ie. $m^\beta \subset m^\alpha$, then the path for α should

be removed in this operation. Hence, if $\mathcal{T}_i$ represents the set of irreducible components of I_i, $1 \leq i \leq q$, then MaxMerge$(\mathcal{T}_1, \ldots, \mathcal{T}_q)$ represents the the set of irreducible components of the ideal $I_1 \cap \cdots \cap I_q$.

B.4 Structure Properties of Monomial Bases

In the results and their proofs below, we explicitly assume that all the ideals are Artinian, adding large powers x_i^N if necessary where N is an integer, though infinity powers will be used in the Algorithms and Examples.

The monomial basis $\mathrm{B}(I)$ for a monomial ideal I is defined as

$$\mathrm{B}(I) = \{\gamma \in \mathbb{N}^n : X^\gamma \notin I\},$$

which form a linear basis for the quotient ring $\mathbb{K}[X]/I$ over $\mathbb{K}$. Thus, for $\gamma \in \mathbb{N}^n$, $\gamma \in B(I)$ if and only if $\alpha \not\leq \gamma$ for every $\alpha \in \mathrm{Min}(I)$. Note that $\mathrm{B}(I)$ is a δ-set, that is, if $\gamma \in \mathrm{B}(I)$ and $\mu \leq \gamma$, then $\mu \in \mathrm{B}(I)$. The next lemma characterizes $B(I)$ in terms of $\mathrm{Irr}(I)$.

Lemma 14 *For $\gamma \in \mathbb{N}^n$, $\gamma \in B(I)$ if and only if $\gamma \prec \beta$ for some $\beta \in Irr(I)$.*

Proof. Since $I = \cap_{\beta \in \mathrm{Irr}(I)} m^\beta$, we have $X^\gamma \in I$ if and only if $X^\gamma \in m^\beta$, ie., $\gamma \not\prec \beta$, for each $\beta \in \mathrm{Irr}(I)$. Hence $X^\gamma \notin I$ if and only if $\gamma \prec \beta$ for some $\beta \in \mathrm{Irr}(I)$, as desired. $\square$

We now want to express $\mathrm{Irr}(I)$ in terms of $\mathrm{B}(I)$. Since I is Artinian, for $\beta = (b_1, \ldots, b_n) \in \mathrm{Irr}(I)$, we have $b_i > 0$ for $1 \leq i \leq n$. Define

$$\beta \ominus 1 = (b_1 - 1, b_2 - 1, \ldots, b_n - 1).$$

Lemma 14 implies that, for each $\beta \in \mathrm{Irr}(I)$, we have $\beta \ominus 1 \in \mathrm{B}(I)$.

A vector $\gamma \in \mathbb{N}^n$ is called *maximal* in $\mathrm{B}(I)$ if

$$\gamma \in \mathrm{B}(I) \text{ and there is no } \mu \in \mathrm{B}(I) \text{ such that } \mu > \gamma.$$

Lemma 15 *For any vector $\beta \in \mathbb{N}^n$, $\beta \in Irr(I)$ if and only if $\beta \ominus 1$ is maximal in $B(I)$.*

Proof. By Lemma 14, $\beta \ominus 1 \in \mathrm{B}(I)$ if and only if there is $\alpha \in \mathrm{Irr}(I)$ such that $\beta \ominus 1 \prec \alpha$. Notice that $\alpha \ominus 1 \in \mathrm{B}(I)$ and $\beta \ominus 1 \prec \alpha$ is equivalent to say $\beta \ominus 1 \leq \alpha \ominus 1$. Hence $\beta \ominus 1$ is maximal in $B(I)$ if and only if $\beta \ominus 1 = \alpha \ominus 1$, that is, $\beta = \alpha \in \mathrm{Irr}(I)$. □

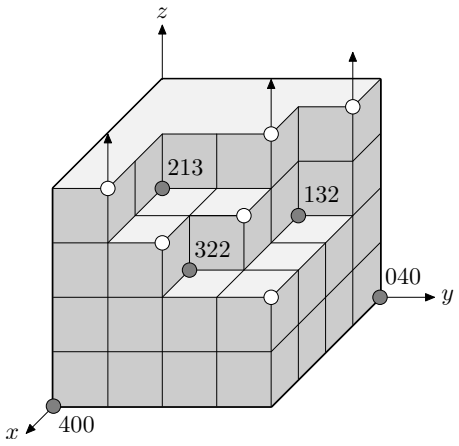


Figure 2: An example of staircase diagram.

The staircase diagram will help us visualize the structural properties of monomial ideals. For example, Figure 2 is the staircase diagram for the monomial ideal $I = \langle x^4, y^4, x^3y^2z^2, xy^3z^2, x^2yz^3 \rangle$. In this figure the gray points are in 1-1 correspondence with the minimal generators, while the white points are in 1-1 correspondence with the irreducible components of I. Geometrically, $\mathrm{B}(I)$ is exactly the set of interior

integral points of the solid.

B.5 Recursive Algorithm

For bivariate monomial ideals, irreducible decomposition is simple [15]. Suppose

$$\mathrm{Min}(I) = \left\{x^{a_1}, x^{a_2}y^{b_2}, \ldots, x^{a_{p-1}}y^{b_{p-1}}, y^{b_p}\right\}$$

where $a_1 > \cdots > a_{p-1} > 0$, $0 < b_2 < \cdots < b_p$, and a_1 or b_p can be infinity. Then the irreducible decomposition of I is

$$I = \langle x^{a_1}, y^{b_2}\rangle \cap \langle x^{a_2}, y^{b_3}\rangle \cap \cdots \cap \langle x^{a_{p-2}}, y^{b_{p-1}}\rangle \cap \langle x^{a_{p-1}}, y^{b_p}\rangle.$$

Our recursive algorithm is a generalization of the above observation to higher dimensions. Let $I \subset \mathbb{K}[x_1, \ldots, x_n]$ be a monomial ideal. Suppose all the distinct degrees of x_n in $\mathrm{Min}(I)$ are

$$0 = d_0 < d_1 < \ldots < d_s.$$

For example, in $I = \langle x^2y^3\rangle = \langle x^\infty, x^2y^3, y^\infty\rangle$, the distinct degrees in y are $d_0 = 0, d_1 = 3$ and $d_3 = \infty$. We collect the coefficients of $m \in \mathrm{Min}(I)$ as polynomials in x_n. Precisely, for $0 \le k \le s$, let

$$I_k = \langle \mathrm{Coeff}_{x_n}(m) : \ m \in \mathrm{Min}(I) \text{ and } \deg_{x_n} m \le d_k\rangle \subseteq \mathbb{K}[x_1, \ldots, x_{n-1}].$$

Then

$$I_0 \subsetneq I_1 \subsetneq \cdots \subsetneq I_s. \tag{27}$$

By (27), it follows that

$$B(I_0) \supsetneq B(I_1) \supsetneq \cdots \supsetneq B(I_s).$$

For the example with $I = \langle x^\infty, x^2y^3, y^\infty \rangle$, $I_0 = \langle x^\infty \rangle = \{0\}$, $I_1 = \langle x^\infty, x^2 \rangle = \langle x^2 \rangle$, and $I_2 = \langle x^\infty, x^2, 1 \rangle = \langle 1 \rangle = \mathbb{K}[x]$.

We show how to read off the irreducible components of I from those of I_k's, which have one less variables. For any vector $\mu = (u_1, \ldots, u_{n-1}) \in \mathbb{N}^{n-1}$ and $d \in \mathbb{N}$, define

$$(\mu, d) = (u_1, \ldots, u_{n-1}, d) \in \mathbb{N}^n.$$

Lemma 16 *For any $\mu \in \mathbb{N}^{n-1}$ and $d \in \mathbb{N}$, $(\mu, d) \in B(I)$ if and only if there exists k, where $1 \leq k \leq s$, such that $d_{k-1} \leq d < d_k$ and $\mu \in B(I_{k-1})$.*

Proof. $(\mu, d) \in B(I)$ if and only if there is no $m \in \mathrm{Min}(I)$ such that $m | X^{(\mu,d)}$. As $d_{k-1} \leq d < d_k$, we only need to see that there is no $m \in \mathrm{Min}(I)$ with $\deg_{x_n} m \leq d_{k-1}$. But this is equivalent to requiring that $\mu \in B(I_{k-1})$. □

For a set of vectors U and an integer d, define

$$U \otimes d = \{(u, d) : u \in U\}.$$

Theorem 17 *$Irr(I) = \bigcup_{k=1}^{s} \big(Irr(I_{k-1}) \setminus Irr(I_k)\big) \otimes d_k$, which is a disjoint union.*

Proof. Assume $\mu \in \mathrm{Irr}(I_{k-1}) \setminus \mathrm{Irr}(I_k)$. We first show that $(\mu, d_k) \ominus 1 \in B(I)$ and $\mu \ominus 1 \in B(I_{k-1}) \setminus B(I_k)$. Since $\mu \in \mathrm{Irr}(I_{k-1})$, we have $\mu \ominus 1 \in B(I_{k-1})$, so

$(\mu, d_k) \ominus 1 = (\mu \ominus 1, d_k - 1) \in \mathrm{B}(I)$ by Lemma 16. Also, by Lemma 15, there is no $\gamma \in B(I_{k-1})$ such that $\gamma > \mu \ominus 1$, in particular no $\gamma \in B(I_k)$ such that $\gamma > \mu \ominus 1$, as $B(I_k) \subset B(I_{k-1})$. Thus $\mu \ominus 1 \notin B(I_k)$, otherwise we would have $\mu \in \mathrm{Irr}(I_k)$ which contradicts the assumption on μ.

For $(\mu, d_k) \in \mathrm{Irr}(I)$, we need to prove that $(\mu, d_k) \ominus 1$ is maximal in $B(I)$. Assume otherwise, say $(\gamma, d) \in B(I)$ and $(\gamma, d) > (\mu, d_k) \ominus 1$. Then $d \geq d_k$ or $d = d_k - 1$. If $d \geq d_k$, then $\gamma \in B(I_j)$ where $k \leq j \leq s$ by Lemma 16. Since $\gamma \geq \mu \ominus 1$ and $B(I_k)$ is a δ-set, $\gamma \in B(I_j)$ implies $\mu \ominus 1 \in B(I_j) \subset B(I_k)$ too, a contradiction. If $d = d_k - 1$, then $\gamma > \mu \ominus 1$. Note that $(\gamma, d_k - 1) \in B(I)$ implies $\gamma \in B(I_{k-1})$ by Lemma 16. However, $\mu \in \mathrm{Irr}(I_{k-1})$ so there is no $\gamma \in B(I_{k-1})$ such that $\gamma > \mu \ominus 1$, a contradiction. Hence such (γ, d) does not exist. Consequently, $(\mu, d_k) \in \mathrm{Irr}(I)$.

Conversely, assume $(\mu, d) \in \mathrm{Irr}(I)$, we need to prove that there exist some $1 \leq k \leq s$ such that $d = d_k$ and $\mu \in \mathrm{Irr}(I_{k-1}) \setminus \mathrm{Irr}(I_k)$. By Lemma 15, $(\mu, d) \in \mathrm{Irr}(I)$ implies

$$(\mu, d) \ominus 1 \in B(I), \tag{28}$$

and there is no $(\gamma, l) \in B(I)$ such that

$$(\gamma, l) > (\mu, d) \ominus 1. \tag{29}$$

By Lemma 16, (28) implies there exists k such that $\mu \ominus 1 \in B(I_{k-1})$, and

$$d_{k-1} \leq d - 1 < d_k. \tag{30}$$

By Lemma 16 again, $(\mu \ominus 1, d_k - 1) \in \mathrm{B}(I)$. Then (29) and (30) imply that $d = d_k$. (29) and (30) also imply that there is no γ such that $\gamma \in B(I_{k-1})$ and $\gamma > \mu \ominus 1$, so $\mu \in \mathrm{Irr}(I_{k-1})$.

It remains to prove $\mu \notin \mathrm{Irr}(I_k)$. Assume $\mu \in \mathrm{Irr}(I_k)$. Then $\mu \ominus 1 \in B(I_k)$. By Lemma 16, $(\mu \ominus 1, d_k) \in \mathrm{B}(I)$ and $(\mu \ominus 1, d_k) > (\mu, d_k) \ominus 1$, contradicting to $(\mu, d_k) \in \mathrm{Irr}(I)$. Thus $\mu \in \mathrm{Irr}(I_{k-1}) \setminus \mathrm{Irr}(I_k)$. □

Theorem 17 gives us the following recursive algorithm for finding irreducible decomposition of monomial ideals. Suppose we are given $I = \langle X^{\alpha_1}, \ldots, X^{\alpha_p} \rangle$ and fixed variable order $x_1 < \cdots < x_n$. We encode the set $\{\alpha_1, \ldots, \alpha_p\}$ as a tree $\mathcal{T}$ of height n. Our algorithm $\mathrm{Irr}(\mathcal{T})$ takes $\mathcal{T}$ as input and produce $\mathrm{Irr}(I)$ as output. That is, $\mathrm{Irr}(I) = \mathrm{Irr}(\mathcal{T})$.

Recursive Algorithm: $\mathrm{Irr}(\mathcal{T})$

Input: $\mathcal{T}$, a tree encoding $\mathrm{Min}(I)$

Output: S, a set (or a tree) representing $\mathrm{Irr}(I)$

Step 1. Start at the root of $\mathcal{T}$. If the height of $\mathcal{T}$ is 1, then $\mathcal{T}$ consists of a few leaves; let d be the largest label on these leaves and let $S := \{d\}$.

Return S (and stop the algorithm).

Step 2. Now assume $\mathcal{T}$ has height at least two. Set $S := \{\ \}$.

Step 3. Suppose $d_0 < d_1 < \cdots < d_s$ are the labels of the children under the root of $\mathcal{T}$, and let $\mathcal{T}_k$ be the subtree extending from d_k, $0 \le k \le s$.

Note that the root of $\mathcal{T}_k$ is the node labeled by d_k, but now unlabeled.

Find $V_0 := \mathrm{Irr}(\mathcal{T}_0)$ by recursive call of this algorithm.

For k from 1 to s do

3.1. Find $\mathcal{T}_k := \mathrm{MinMerge}(\mathcal{T}_{k-1}, \mathcal{T}_k)$, and delete $\mathcal{T}_{k-1}$.

3.2. Find $V_k := \mathrm{Irr}(\mathcal{T}_k)$ by recursive call of this algorithm.

3.3. Find $V := V_{k-1} \setminus V_k$, delete V_{k-1}, and $S := \mathrm{Merge}(S, V \otimes d_k)$.

Step 4. Return (S).

Example 18 *We end this section by demonstrating how the algorithm is used to decompose the ideal* $I = \langle x^4, y^4, x^3y^2z^2, xy^3z^2, x^2yz^3 \rangle$. *First represent the monomials as a tree with variable order* $x < y < z$, *where* $\mathcal{T}_k$*'s are the subtrees extending from the node with label* d_k, $k = 0, 1, 2, 3$.

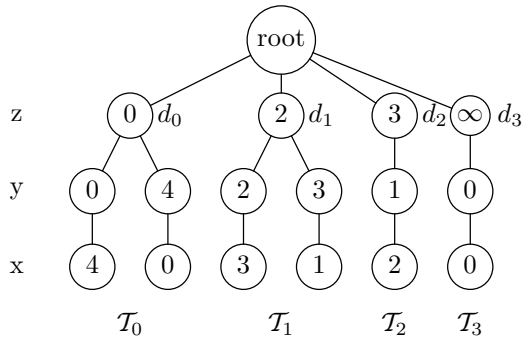


Figure 3: Tree representation.

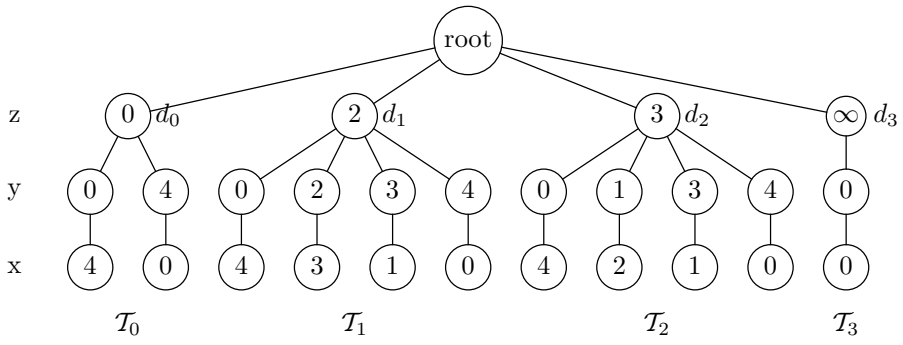


Figure 4: MinMerge step.

Figure 4-5 show the process of finding the irredundant irreducible decomposition of I. *For each* $\mathcal{T}_k$, *inductively MinMerge the subtrees from left to right, corresponding to Step 3.1 in the Recursive algorithm. See Figure 4. In Figure 5 we call the procedure*

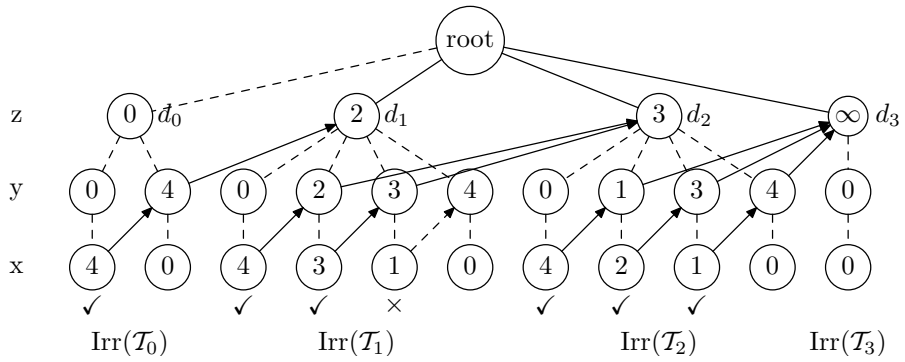


Figure 5: Shifting step.

Irr( ) for each $\mathcal{T}_k$ to compute Irr($\mathcal{T}_k$), corresponding to Step 3.2. Since the height of $\mathcal{T}_k$ is 2, we bind each leaf that is not in the most-right side of $\mathcal{T}_k$ with the node of height 2 on the next path - just do the shifting in adjacent paths, see Figure 5. Finally we find the paths in Irr($\mathcal{T}_{k-1}$) that are not in Irr($\mathcal{T}_k$). The one with a mark $\times$ in Irr($\mathcal{T}_k$) is discarded. Then bind the resulting paths with d_k. The irreducible components can be read from the last figure:

$$Irr(I) = \{(4,4,2),(4,2,3),(3,3,3),(4,1,\infty),(2,3,\infty),(1,4,\infty)\}.$$

B.6 Incremental Algorithm

In this section we shall present an incremental algorithm based on the idea of adding one generator at a time. This algorithm can be viewed as an improvement of Alexander Dual method ([14, 17]). We maintain an output list of irreducible components, and at each step we use a new generator to update the output list. In [17], it is not clear how to select good candidates that need to be updated, and the updating process there is also inefficient. Our algorithm avoids these two deficiencies. We establish some rules that help us to exclude many unnecessary comparisons.

Monomial ideal are much simpler than general ideals. The next theorem tells us that monomial ideals satisfy distribution rules for the operations "+" and "$\cap$". These rules may not be true for general ideals.

Theorem 19 (Distribution Rules) *Let $I_1, \ldots, I_t, J$ be any monomial ideals in $\mathbb{K}[X]$. Then*

(a) $(I_1 + \ldots + I_t) \cap J = I_1 \cap J + \ldots + I_t \cap J$, *and*

(b) $(I_1 \cap \ldots \cap I_t) + J = (I_1 + J) \cap \ldots \cap (I_t + J)$.

Proof. By induction, we just need to prove the case for $t = 2$. Note that (b) follows form (a), as

$$\begin{aligned}(I_1 + J) \cap (I_2 + J) &= I_1 \cap (I_2 + J) + J \cap (I_2 + J) \\ &= I_1 \cap I_2 + I_1 \cap J + J \cap I_2 + J \\ &= I_1 \cap I_2 + J.\end{aligned}$$

To prove (a) for the case $t = 2$, suppose h is a generator for $(I_1 + I_2) \cap J$. Then h must be in $(I_1 + I_2)$ and J. Since $(I_1 + I_2) \cap J$ is also a monomial ideal, h is a monomial. The fact that $h \in I_1 + I_2$ implies that h is in either I_1 or I_2. Hence h is in $I_1 \cap J$ or in $I_2 \cap J$, so $h \in I_1 \cap J + I_2 \cap J$. Going backward yields the proof for the other direction. $\square$

Theorem 19 gives us an incremental algorithm for irreducible decomposition of monomial ideals. Precisely, we have the following situation at each incremental step: Given the irreducible decomposition $\mathrm{Irr}(I)$ of an arbitrary ideal I and a new monomial X^α where $\alpha = (a_1, \ldots, a_n) \in \mathbb{N}^n$, we want to decompose $\widetilde{I} = I + \langle X^\alpha \rangle$. By

the distribution rule (b),

$$\widetilde{I} = \left(\bigcap_{\beta \in \mathrm{Irr}(I)} m^\beta \right) + \langle X^\alpha \rangle = \bigcap_{\beta \in \mathrm{Irr}(I)} \left(m^\beta + \langle X^\alpha \rangle \right). \tag{31}$$

We need to see how to decompose each ideal on the right hand side of (31) and how to get rid of redundant components. We partition $\mathrm{Irr}(I)$ into two disjoint sets:

$$T_1^\alpha = \{\beta \in \mathrm{Irr}(I) : \alpha \not\prec \beta\}, \text{ and} \tag{32}$$

$$T_2^\alpha = \{\beta \in \mathrm{Irr}(I) : \alpha \prec \beta\}. \tag{33}$$

Note that if $X^\alpha \in I$ then $T_2^\alpha = \phi$. For each $\beta \in T_1^\alpha$, we have $X^\alpha \in m^\beta$, thus

$$m^\beta + \langle X^\alpha \rangle = m^\beta. \tag{34}$$

For each $\beta \in T_2^\alpha$, we have $X^\alpha \notin m^\beta$. In this case, we split $\langle X^\alpha \rangle$ as

$$\langle X^\alpha \rangle = \bigcap_{j=1}^{n} \langle x_j^{a_j} \rangle.$$

By the distribution rule (b), we have

$$m^\beta + \langle X^\alpha \rangle = \bigcap_{j=1}^{n} \left(m^\beta + \langle x_j^{a_j} \rangle \right).$$

Define

$$\beta^{(\alpha,j)} = (b_1, \ldots, b_{j-1}, a_j, b_{j+1}, \ldots, b_n), \quad 1 \le j \le n.$$

Since $\alpha \prec \beta$, we have $a_j < b_j$ for all $1 \leq j \leq n$. Hence $m^\beta + \langle x_j^{a_j} \rangle = m^{\beta^{(\alpha,j)}}$, and

$$m^\beta + \langle X^\alpha \rangle = \bigcap_{j=1}^{n} m^{\beta^{(\alpha,j)}}. \tag{35}$$

Therefore,

$$\mathrm{Irr}(\widetilde{I}) = \mathrm{MaxMerge}\left(T_1^\alpha, \{\beta^{(\alpha,j)} : \beta \in T_2^\alpha \text{ and } 1 \leq j \leq n\}\right). \tag{36}$$

It remains to see which of the components in the right hand side of the above expression belong to $\mathrm{Irr}(\widetilde{I})$, so others are redundant.

Lemma 20 *$T_1^\alpha \subset Irr(\widetilde{I})$.*

Proof. Let $\beta_1 \in T_1^\alpha$. By equation (36) if $\beta_1 \notin \mathrm{Irr}(\widetilde{I})$, then there exists some $\beta_2 \in T_2^\alpha$ such that β_1 is maxmergeed by $\beta_2^{(\alpha,j)}$ for some j, ie. $\beta_1 \leq \beta_2^{(\alpha,j)}$. Since $\beta_2^{(\alpha,j)} < \beta_2$, $\beta_1 \leq \beta_2^{(\alpha,j)}$ implies that $\beta_1 < \beta_2$, which contradicts with the fact that $\beta_1, \beta_2 \in \mathrm{Irr}(I)$. Hence $\beta_1 \in \mathrm{Irr}(\widetilde{I})$ as claimed. □

Lemma 20 shows that the elements in T_1^α will be automatically in $\mathrm{Irr}(\widetilde{I})$. Now we turn to the components $\beta^{(\alpha,j)}$. For $\beta \in T_2^\alpha$, define

$$M_\beta = \{m \in \mathrm{Min}(I) : m | X^\beta\}. \tag{37}$$

For $m \in M_\beta$, if $\deg_{x_u} m = b_u$, then we say m *matches* β in x_u. It is possible that one monomial matches β in multiple variables. For example, with $I = \langle x^2, y^2, z^2, xy, xz, yz \rangle$ and $\beta = (1,1,2) \in \mathrm{Irr}(I)$, the monomial xy matches β in x and y. We say m matches β only in x_u if $\deg_{x_u} m = b_u$ and $\deg_{x_k} m < b_k$ for all $k \neq u$.

Lemma 21 *For each $\beta = (b_1, \ldots, b_n) \in T_2^\alpha$ and each $1 \leq u \leq n$, there exists $m \in M_\beta$ such that m matches β only in x_u.*

Proof. Note that a vector $\gamma \in \mathrm{B}(I)$ is maximal if and only if $X^\gamma \cdot x_u \in I$ for every u. Since $\beta \in \mathrm{Irr}(I)$, $\beta \ominus 1$ is maximal in $B(I)$. Thus, for each $1 \leq u \leq n$, $X^{\beta \ominus 1} \cdot x_u \in I$, so there exists a monomial say $m \in \mathrm{Min}(I)$ such that $m | X^{\beta \ominus 1} \cdot x_u$. Then $\deg_{x_k} m < b_k$ for $k \neq u$. If $\deg_{x_u} m < b_u$ as well, then $m | X^{\beta \ominus 1}$, which implies that $X^{\beta \ominus 1} \in I$, a contradiction. Therefore $\deg_{x_u} m = b_u$. Note that $X^{\beta \ominus 1} \cdot x_u | X^\beta$, so $m \in M_\beta$. □

For any set of monomials $A \subset \mathbb{K}[X]$, define $\mathbf{max}(A)$ be the exponent γ such that $X^\gamma = \mathrm{Lcm}(A)$.

Lemma 22 $\max(M_\beta) = \beta$.

Proof. By the definition of M_β, we know that $\max(M_\beta) \leq \beta$. By Lemma 21 we have $\max(M_\beta) \geq \beta$. Thus $\max(M_\beta) = \beta$. □

For $k \neq u$, let

$$d(\beta, u, k) = \min\{\deg_{x_u} m : m \in M_\beta \text{ matching } \beta \text{ only in } x_k\}. \tag{38}$$

Note that $d(\beta, u, k) < b_u$. Define

$$d(\beta, u) = \max_{1 \leq k \leq n, k \neq u} \{d(\beta, u, k)\}.$$

Lemma 23 *For each $\beta \in T_2^\alpha$ and $1 \leq u \leq n$, $\beta^{(\alpha,u)} \in Irr(\widetilde{I})$ if and only if $d(\beta, u) < a_u$.*

Proof. Suppose $d(\beta, u) < a_u$. We want to prove that $\beta^{(\alpha,u)} \in \mathrm{Irr}(\widetilde{I})$. By Lemma 15, this is equivalent to proving that $\beta^{(\alpha,u)} \ominus 1 \in \mathrm{B}(\widetilde{I})$ and is maximal. Assume $\beta^{(\alpha,u)} \ominus 1 \notin \mathrm{B}(\widetilde{I})$. Then there exists $m \in \mathrm{Min}(I) \cup \{X^\alpha\}$ such that $m | X^{\beta^{(\alpha,u)} \ominus 1}$. First note that $m \neq X^\alpha$ because X^α can not divide $X^{\beta^{(\alpha,u)} \ominus 1}$. Thus $m \in \mathrm{Min}(I)$, which implies $X^{\beta^{(\alpha,u)} \ominus 1} \in I$. Since $\beta^{(\alpha,u)} \ominus 1 < \beta \ominus 1$, we have $X^{\beta \ominus 1} \in I$, contradicting to $\beta \in \mathrm{Irr}(I)$. Hence $\beta^{(\alpha,u)} \ominus 1 \in \mathrm{B}(\widetilde{I})$. We next need to prove that $\beta^{(\alpha,u)} \ominus 1$ is maximal in $\mathrm{B}(\widetilde{I})$, that is, $X^{\beta^{(\alpha,u)} \ominus 1} \cdot x_k \in \widetilde{I}$ for every k. In the case for $k = u$, we have $X^\alpha | X^{\beta^{(\alpha,u)} \ominus 1} \cdot x_u$. For any $k \neq u$, let m be any monomial in (38) such that $\deg_{x_u} m = d(\beta, u, k)$. Then $\deg_{x_u} m = d(\beta, u, k) \leq d(\beta, u) < a_u$, hence $m | X^{\beta^{(\alpha,u)} \ominus 1} \cdot x_k$ as $\deg_{x_k} m = b_k$ and $\deg_{x_j} m \leq b_j - 1$ for $j \neq u, k$.

Conversely, suppose $\beta^{(\alpha,u)} \in \mathrm{Irr}(\widetilde{I})$. We want to prove that $d(\beta, u) < a_u$. We know that $\beta^{(\alpha,u)} \ominus 1$ is maximal in $\mathrm{B}(\widetilde{I})$. Thus $X^{\beta^{(\alpha,u)} \ominus 1} \cdot x_k \in \widetilde{I}$ for every k. For any $k \neq u$, suppose $X^{\beta^{(\alpha,u)} \ominus 1} \cdot x_k$ is divisible by $m \in \mathrm{Min}(I) \cup \{X^\alpha\}$. Then

$$\deg_{x_u} m \leq a_u - 1 < b_u, \quad \deg_{x_j} m \leq b_j - 1, \ j \neq u, k, \tag{39}$$

and $\deg_{x_k} m \leq b_k$. As $X^{\beta^{(\alpha,u)} \ominus 1} \in \mathrm{B}(\widetilde{I}) \subset \mathrm{B}(I)$, m can not divide $X^{\beta^{(\alpha,u)} \ominus 1}$. Hence $\deg_{x_k} m \leq b_k$. So m matches β only in x_k. Note that $m \neq X^\alpha$, so $m \in M$ and thus $m \in M_\beta$. It follows that $d(\beta, u, k) \leq a_u - 1$ by (39). Therefore, $d(\beta, u) < a_u$ as desired. □

By the above lemma, for each $\beta \in T_2^\alpha$, we only need to find M_β and $d(\beta, u)$, which will tell us whether $\beta^{(\alpha,u)} \in \mathrm{Irr}(\widetilde{I})$. This gives us the following incremental algorithm.

Incremental algorithm

Input: M, a set of monomials in n variables $x_1, \ldots, x_n$.

Output: $\mathrm{Irr}(I)$, the irredundant irreducible components of the ideal I generated by M.

Step 1. Compute MinMerge(M) and sort it into the form:

$$\mathrm{MinMerge}(M) = \{x_1^{c_1}, \ldots, x_n^{c_n}, X^{\alpha_1}, \ldots, X^{\alpha_p}\},$$

where c_i can be ∞ and $\{X^{\alpha_1}, \ldots, X^{\alpha_p}\}$ are sorted in lex order with variable order $x_1 < \ldots < x_n$. Set

$$T := \{(a_1, \ldots, a_n)\}.$$

Step 2. For each k from 1 to p do:

2.1. Set the temporal variables $V = \emptyset$ and $\alpha := \alpha_k$.

2.2. For every $\beta \in T$ with $\alpha \not\prec \beta$ do

$$V := V \cup \{\beta\}.$$

2.3. For every $\beta \in T$ with $\alpha \prec \beta$ do,

- find M_β as defined in (37);
- for $1 \leq u \leq n$, compute $d(\beta, u)$, and if $d(\beta, u) < a_u$ then update

$$V := V \cup \{\beta^{(\alpha,u)}\}.$$

2.4. Set $T := V$.

Step 3. Output T.

We next prove that there is a nice property of the above algorithm for generic monomial ideals, that is, the size of T is always non-decreasing at each stage when a new generator is added. This will allow us to bound the running time of the algorithm in term of input and output sizes.

Theorem 24 *Suppose I is generic and $Min(I) = \{x_1^{c_1}, \ldots, x_n^{c_n}, X^{\alpha_1}, \ldots, X^{\alpha_p}\}$ where X^{α_k}'s are sorted in lex order with variable order $x_1 < \ldots < x_n$. Let $\widehat{I} = \langle x_1^{c_1}, \ldots, x_n^{c_n}, X^{\alpha_1}, \ldots, X^{\alpha_{p-1}} \rangle$. Then $|Irr(\widehat{I})| \leq |Irr(I)|$.*

Proof. Keep notations as above. For every $\beta \in T_2^\alpha$, $b_n = c_n$. Thus $x_n^{c_n}$ is the

only monomial in M_β that has degree in x_n larger than a_n. Hence $d(\beta, n) < a_n$ and $\beta^{(\alpha,n)} \in \mathrm{Irr}(I)$. By the equation (36) and Lemma 20,

$$|\mathrm{Irr}(I)| \geq |T_1^\alpha| + |\{\beta^{(\alpha,n)} : \beta \in T_2^\alpha\}| = |T_1^\alpha| + |T_2^\alpha| = |\mathrm{Irr}(\widehat{I})|.$$

□

The reader might wonder whether a similar statement holds in non-generic case as well. The answer is negative. Let $I = \langle x^3, y^3, z^2, w^2, x^2yz, xy^2w \rangle \subset \mathbb{K}[x, y, z, w]$ with lex order and $x < y < z < w$. Then

$$\mathrm{Irr}(I) = \{(3, 3, 1, 1), (2, 3, 2, 1), (3, 2, 1, 2), (3, 1, 2, 2), (2, 2, 2, 2), (1, 3, 2, 2)\}.$$

By adding $X^\alpha = xyzw$, we can see $\beta = (2, 2, 2, 2) \in T_2^\alpha$. Note that $M_\beta = \{x^2yz, xy^2w, z^2, w^2\}$. Since $d(\beta, u) = 1 = a_u$ for $u = 1, 2, 3, 4$, no new $\beta^{(\alpha,j)}$ will be generated. Thus the number of irreducible components decreases by 1 instead.

We find the irreducible components for the monomial ideal in Example 18 again by the flow of our incremental algorithm.

Example 25 *Decompose*

$$I = \langle x^4, y^4, x^3y^2z^2, xy^3z^2, x^2yz^3 \rangle.$$

Note: "✓" means $\beta^{(\alpha,u)} \in Irr(\widetilde{I})$ for corresponding β, α and u, while "×" means not.

Step 1. $M = \{x^4, y^4, z^\infty, x^3y^2z^2, xy^3z^2, x^2yz^3\}$. *Set* $T := \{(4, 4, \infty)\}$.
Step 2. (i) For $\alpha = (3, 2, 2)$ *do:*

2.1. $V := \phi$.

2.2. Since $\alpha \prec (4, 4, \infty)$, $V := \phi$.

2.3. Let $\beta = (4, 4, \infty)$. *We find* $M_\beta = \{x^4, y^4\}$.

So we have $d\{\beta, 1\} = 0$*(*✓*),* $d\{\beta, 2\} = 0$*(*✓*) and* $d\{\beta, 3\} = 0$*(*✓*).*

Then $V := \{(3, 4, \infty), (4, 2, \infty), (4, 4, 2)\}$.

2.4. Let $T := V$.

(ii) For $\alpha = (1, 3, 2)$ *do:*

2.1. $V := \phi$.

2.2. Update V *by* $V := \{(4, 4, 2), (4, 2, \infty)\}$.

2.3. $\alpha \prec (3, 4, \infty)$.

Let $\beta = (3, 4, \infty)$. *We find* $M_\beta = \{y^4, x^3y^2z^2\}$.

So $d\{\beta, 1\} = 0$*(*✓*),* $d\{\beta, 2\} = 2$*(*✓*) and* $d\{\beta, 3\} = 2$*(*×*).*

Then $V := \{(4, 4, 2), (4, 2, \infty), (1, 4, \infty), (3, 3, \infty)\}$.

2.4. Let $T := V$.

(iii) For $\alpha = (2, 1, 3)$ *do:*

2.1. $V := \phi$.

2.2. $V := \{(4, 4, 2), (1, 4, \infty)\}$.

2.3. $\alpha \prec (4, 2, \infty)$, *and* $\alpha \prec (3, 3, \infty)$.

- *Let* $\beta = (4, 2, \infty)$. *We find* $M_\beta = \{x^4, x^3y^2z^2\}$.

  So $d\{\beta, 1\} = 3$*(*×*),* $d\{\beta, 2\} = 0$*(*✓*) and* $d\{\beta, 3\} = 2$*(*✓*).*

  Then $V := \{(4, 4, 2), (1, 4, \infty), (4, 1, \infty), (4, 2, 3)\}$.

- *Let* $\beta = (3, 3, \infty)$. *Then* $M_\beta = \{x^3y^2z^2, xy^3z^2\}$.

  $d\{\beta, 1\} = 1$*(*✓*),* $d\{\beta, 2\} = 2$*(*×*),* $d\{\beta, 3\} = 2$*(*✓*).*

  So $V := \{(4, 4, 2), (1, 4, \infty), (4, 1, \infty), (4, 2, 3), (2, 3, \infty), (3, 3, 3)\}$.

2.4. Let $T := V$.

Step 3. Output $T = \{(4,4,2),(1,4,\infty),(4,1,\infty),(4,2,3),(2,3,\infty),(3,3,3)\}$
$= \{(4,4,2),(4,2,3),(3,3,3),(4,1,\infty),(2,3,\infty),(1,4,\infty)\}.$

Some preprocess can be taken right before Step 2 to improve the efficiency of the incremental algorithm. For each $u \in \{1,\ldots,n\}$, we partition M into disjoint subsets such that the monomials in each subset have the same degree in x_u. We then store these information, which requires memory complexity $O(n \cdot p)$. For each $\beta \in T_2^\alpha$, we can find M_β by only checking the monomials in the subset with degree b_u in variable x_u for every u. Note that for generic monomial ideals each subset contains a unique monomial. In this case M_β contains n monomials, and it can be found by $O(n)$ operations, instead of $O(p)$ operations by scanning through the whole input monomial set.

B.7 Time Complexity and Conclusion

We estimate the running time of our algorithms by counting the number of monomial operations (ie. comparisons and divisibility) used. Our recursive algorithm depends heavily on the number of distinct degrees of each variable. Let s_j be the number of distinct degrees of x_j where $j = 1,\ldots,n$. Then the total number of merge of subtrees used by the algorithm is at most $\prod_{j=1}^{n} s_j$. Since each subtree has at most p leaves(ie. p generators), each merge takes $O(p^2)$ monomial operations. Hence the algorithm uses $O(p^2 \cdot \prod_{j=1}^{n} s_j)$ monomial operations. This algorithm is more efficient for highly non-generic monomial ideals. The benchmark analysis in [20] compare several algorithms based on various slicing strategies, including our recursive algorithm. It is shown there that our algorithm performs as a very close second best one.

The running time of our incremental algorithm is harder to estimate for general ideals. For generic ideals, however, we can bound the time in terms of input and output sizes. More precisely, suppose

$$I = \langle x_1^{c_1}, \ldots, x_n^{c_n}, X^{\alpha_1}, \ldots, X^{\alpha_p} \rangle$$

is a generic monomial ideal in $\mathbb{K}[X]$ where X^{α_k}'s are sorted in lex order with variable order $x_1 < \ldots < x_n$. For $0 \leq k \leq p$, let

$$I_{(k)} = \langle x_1^{c_1}, \ldots, x_n^{c_n}, X^{\alpha_1}, \ldots, X^{\alpha_k} \rangle.$$

All these ideals are generic. By Theorem 24, we have

$$1 = |\mathrm{Irr}(I_{(0)})| \leq |\mathrm{Irr}(I_{(1)})| \leq \cdots \leq |\mathrm{Irr}(I_{(p)})| = |\mathrm{Irr}(I)|.$$

In an arbitrary stage of the incremental algorithm, we try to find the irreducible components of $I_{(k)}$ from those of $I_{(k-1)}$. For each $\beta \in \mathrm{Irr}(I_{(k-1)})$, only those β in $T_2^{\alpha_k}$ (as defined in (33)) need to be updated. Note that I is generic, by the preprocess M_β can be found in $O(n)$ operations. The numbers $d(\beta, u, k)$, $1 \leq u \neq k \leq n$, can be computed by scanning through the monomials in M_β once, thus using only $O(n)$ monomial operations. Then the numbers $d(\beta, u)$, $1 \leq u \leq n$, can be computed in $O(n^2)$ operations. Hence for each $\beta \in T_2^{\alpha_k}$, Step 2.3 uses at most $O(n + n^2) = O(n^2)$ monomial operations. Since $T \supset T_2^{\alpha_k}$ has at most ℓ elements where $\ell = |\mathrm{Irr}(I)|$, Step 2.3 needs at most $O(n^2\ell)$ monomial operations. Therefore, the total number of monomial operations is at most $O(n^2 p\ell)$. In fact, $T_2^{\alpha_k}$ is usually a small subset of T, the actual running time is much better than our worst-case estimate indicates.

We also want to point out that for generic monomial ideals, the incremental

algorithm is an improved version of the recursive algorithm. Suppose we add the new monomial X^{α_k} into $I_{(k-1)}$. In Step 3.2 of the recursive algorithm, we need to compute $\mathrm{Irr}(\mathcal{T}_k)$. But in Step 2.3 of the incremental algorithm, only $\beta \in T_2^{\alpha_k}$ need to be updated. We have the observation that $T_2^{\alpha_k}$ is a small subset of $\mathrm{Irr}(\mathcal{T}_k) \otimes c_n$. By this observation we conclude the incremental algorithm is more efficient than the recursive algorithm for generic monomial ideals. In non-generic case, the comparison is not clear.

In all previous algorithms (including our recursive one) for monomial decomposition, the storage in the intermediate stages may grow exponentially larger than the output size. Our incremental algorithm seems to be the first algorithm for monomial decomposition that the intermediate storage is bounded by the final output size. Note that the output size ℓ can be exponentially large in n. In fact, it is proven in [1] that $\ell = O(p^{[\frac{n}{2}]})$ for large p. Since the output size can be exponential in n, it is impossible to have a polynomial time algorithm for monomial decomposition.

B.8 Acknowledgement

We thank Alexander Milowski and Bjarke Roune for comments and suggestions, and Ezara Miller for helpful communications (especially for providing some of the diagrams).

Bibliography

[1] Agnarsson, G., 1997. The number of outside corners of monomial ideals. J Pure Appl Algebra. 117&118, 3-22.

[2] Anwar, I., 2007. Janet's Algorithm. Eprint arXiv, 0712.0068.

[3] Block, F., Yu, J., 2006. Tropical convexity via cellular resolutions. J Algebr Comb. 24(1), 103-114. Eprint arXiv,math/0503279.

[4] Bayer,D., Peeva, I., Sturmfels, B., 1998, Monomial resolutions. Math Res Lett. 5(5),31-46.

[5] Cox, D., Little, J., O'Shea, D., 1997. Ideals, Varieties, and Algorithms, An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer-Verlag.

[6] Cox, D., Little, J., O'Shea, D., 1998. Using Algebraic Geometry. In: Graduate Texts in Mathematics, vol. 185. Springer.

[7] Eisenbud, D., 1995. Commutative algebra, with a view toward algebraic geometry. In: Graduate Texts in Mathematics, vol. 150, Springer.

[8] Far, J., Gao, S., 2006. Computing Gröbner bases for vanishing ideals of finite sets of points. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes.

In: Springer Lecture Notes in Computer Science, no. 3857, Springer-Verlag, 118-127.

[9] Gao, S., Rodrigues, V., Stroomer, J., 2003. Gröbner basis structure of finite sets of points. Preprint.

[10] Gao, S., Zhu, M., 2008. Upper bound on the number of irreducible components of monomial ideals. In preparation.

[11] Hoşten S., Smith, G., 2002. Monomial ideals. Computations in algebraic geometry with Macaulay 2, Springer-Verlag.

[12] Hoşten S., Sturmfels, B., 2007. Computing the integer programming gap. Combinatorica, 27, 367-382.

[13] Jarrah, A., Laubenbacher, R., Stigler, B., Stillman, M., 2006. Reverse-engineering of polynomial dynamical systems. Adv Appl Math, 39(4), 477-489.

[14] Miller, E., 2000. Resolutions and Duality for Monomial Ideals. PhD thesis, University of California, Berkeley, Mathematics Department.

[15] Miller, E., Sturmfels, B., 1999. Monomial ideals and planar graphs. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. In: Springer Lecture Notes in Computer Science, no. 1719, Springer-Verlag, AAECC-13 proceedings (Honolulu, Nov. 1999), pp. 19-28.

[16] Miller, E., Sturmfels, B., 2004. Combinatorial Commutative Algebra. In: Graduate Texts in Mathematics, vol. 227, Springer.

[17] Milowski, A., 2004. Computing Irredundant Irreducible Decompositions of Large Scale Monomial Ideals. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation 04, 235-242.

[18] Roune, B., 2007. The label algorithm for irreducible decomposition of monomial ideals. Eprint arXiv,0705.4483.

[19] Roune, B., 2008. Solving Thousand-Digit Frobenius Problems Using Gröbner Bases. J Symb Comput, 43(1), 1-7. Eprint arXiv,math/0702040.

[20] Roune, B., 2008. The Slice Algorithm For Irreducible Decomposition of Monomial Ideals. To appear in J Symb Comput. Eprint arXiv,0806.3680.

[21] Sturmfels, B., Gröebner Bases and Convex Polytopes. In: AMS University Lecture Series, vol. 8.

[22] Sturmfels, B., Sullivant, S., 2006. Combinatorial secant varieties. Pure and Applied Mathematics Quarterly, 2, 285-309. Eprint arXiv,math/0506223.

[23] Vasconcelos, W., 1998. Computational Methods in Commutative Algebra and Geometry. Algorithms and Computation in Mathematics, vol. 2. Springer-Verlag.

[24] Villarreal, R., 2001. Monomial algebras. Monographs and Textbooks in Pure and Applied Mathematics, vol. 238. CRC Press.

Bibliography

[1] http://www.aids.org

[2] http://aids.about.com

[3] http://www.aidsinfo.nih.gov

[4] http://www.aidsmeds.com

[5] http://hivdb.stanford.edu/

[6] http://en.wikipedia.org/wiki/HIV

[7] http://en.wikipedia.org/wiki/Tuberculosis

[8] Beerenwinkel, N., Daumer, M., Oette, M., Korn, K., Hoffmann, D., Kaiser, R., Lengauer, T., Selbig, J. and Walter, H. (2003). Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes. *Nucl Acids Res*, 31(13), 3850-3855.

[9] Beerenwinkel, N., Däumer, M., Sierra, S., Schmidt, B., Walter, H., Korn, K., Oette, M., Fätkenheuer, G., Rockstroh, J., Lengauer, T., Hoffmann, D., Kaiser, R. and Selbig, J. (2002). Geno2pheno is predictive of short-term virological response. *Antivir Ther*, 7 (Suppl. 1), 98.

[10] Beerenwinkel, N. and Drton, M. (2007). A mutagenetic tree hidden Markov model for longitudinal clonal HIV sequence data. *Biostatistics*, 8(1), 53-71.

[11] Beerenwinkel, N., Eriksson, N. and Sturmfels, B. (2006). Evolution on distributive lattices. *J Theor Biol*, 242(2): 409-420.

[12] Beerenwinkel, N., Rahnenführer, J., Däumer, M., Hoffmann, D., Kaiser, R., Selbig, J. and Lengauer, T. (2005). Learning multiple evolutionary pathways from cross-sectional data, *J Comput Biol*, 12(6): 584-598.

[13] Beerenwinkel, N. and Sullivant, S. (2007). Markov models for accumulating mutations, *Biometrika*, in press. Eprint arXiv: 0709.2646.

[14] Bogojeska, J., Lengauer and T., Rahnenführer J. (2008). Stability analysis of mixtures of mutagenetic trees, *BMC Bioinf*, 9: 165.

[15] Boucher, C. A. B., E. O'Sullivan, J. W. Mulder, C. Ramautarsing, P. Kellam, G. Darby, J. M. A. Lange, J. Goudsmit, and B. A. Larder. (1992). Ordered appearance of zidovudine resistance mutations during treatment of 18 human immunodeficiency virus-positive subjects. *J Infect Dis*, 165:105-110.

[16] Clauset, A., Moore, C. and Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nat*, 453: 98-101.

[17] Coffin, J.M. (1995). HIV population dynamics in vivo: Implications for genetic variation, pathogenesis, and therapy, *Sci*, 267: 483-489.

[18] Collins, J., Thompson, M., Paintsil, E., Ricketts, M., Gedzior, J. and Alexander, L. (2004). Competitive fitness of nevirapine-resistant human immunodeficiency virus type 1 mutants, *J Virol*, 78: 603-611.

[19] Desper, R., Jiang, F., Kallioniemi, O., Moch, H., Papadimitriou, C., and Schaffer, A. (1999). Inferring tree models for oncogenesis from comparative genome hybridization data. *J Comput Biol*, 6, 37-51.

[20] DiRienzo, A.G., DeGruttola, V., Larder, B. and Hertogs, K. (2003). Nonparametric methods to predict HIV drug susceptibility phenotype from genotype. *Stat Med*, 22(17):2785-98.

[21] Drton, M., Sturmfels, B. and Sullivant, S. (2009). *Lectures on Algebraic Statistics*. Series in Oberwolfach Seminars Vol. 39, Springer.

[22] Edmonds, J. (1967). Optimum branchings. *J Res Nat Bur Stand*, 71B, 233-240.

[23] Foulkes, A. F. and DeGruttola, V. (2003). Characterizing the progression of viral mutations over time. *J Am Stat Assoc*, 98(464), 859-867.

[24] Gerstung, M. and Beerenwinkel, N. (2008). Waiting time models of cancer progression. Eprint arXiv: 0807.3638.

[25] Gilbert, P.B., Novitsky, V. and Essex, M. (2005) Covariability of selected amino acid positions for HIV type 1 subtypes C and B. *AIDS Res Hum Retroviruses*, 21: 1016-1030.

[26] Gonzales, M.J., Wu, T.D., Taylor, J., Belitskaya, I., Kantor, R., Israelski, D., Chou, S., Zolopa, A.R., Fessel, W.J. and Shafer, R.W. (2003). Extended spectrum of HIV-1 reverse transcriptase mutations in patients receiving multiple nucleoside analog inhibitors. *AIDS*, 17:791-799.

[27] Hall, B. (2002). Predicting evolution by in vitro evolution requires determining evolutionary pathways. *Antimicrob Agents Chemother*, 46(9), 3035-3038.

[28] Healy, B. and DeGruttola, V. (2006). Hidden Markov models for settings with interval censored transition times and uncertain time origin: Application to HIV genetic analyses. *Biostatistics*, doi: 10.1093/biostatistics/kxl021.

[29] Healy, B., DeGruttola, V. and Pagano, M. (2007). Combining cross-sectional and prospective data methods to improve transition parameter estimation for characterizing the accumulation of HIV-1 drug resistance mutations. *Biometrics*, 63(3), 742-750.

[30] Healy, B., DeGruttola, V. and Hu, C. (2008). Accommodating uncertainty in a tree set for function estimation. *Stat Appl Gene Mo B*, 7(1): Article 5.

[31] Hoffmana, N., Schifferb, C. and Swanstrom, R. (2003). Covariation of amino acid positions in HIV-1 protease. *Virol*, 314, 536-48.

[32] Hu, Z., Giguel, F., Hatano, H., Reid, P., Lu, J. and Kuritzkes, D.R. (2006). Fitness comparison of thymidine analog resistance pathways in human immunodeficiency virus type 1. *J Virol*, 80:7020-70.

[33] Jeeninga, R., Keulen, W., Boucher, C., Sanders, R.W. and Berkhout, B. (2001). Evolution of AZT resistance in HIV-1: The 41-70 intermediate that is not observed in vivo has a replication defect. *Virol*, 283, 294-305.

[34] Kellam, P., Boucher, C.A. and Larder, B.A. (1992). Fifth mutation in human immunodeficiency virus type 1 reverse transcriptase contributes to the development of high-level resistance to zidovudine, *Proc Natl Acad Sci USA*, 89:1934-1938.

[35] Korber, B.T., Farber, R.M., Wolpert, D.H and Lapedes, A.S. (1993). Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis. *Proc Natl Acad Sci USA*, 90: 7176-7180.

[36] Larder, B. and Kemp S. (1989). Multiple mutations in HIV-1 reverse transcriptase confer high-level resistance to zidovudine (AZT). *Sci*, 246:1155-1158.

[37] Lengauer, T. and Sing, T. (2006). Bioinformatics-assisted anti-HIV therapy. *Nat Rev Microbio*, 4(10):790-7.

[38] Liu, Y., Eyal, E. and Bahar, I. (2008). Analysis of correlated mutations in HIV-1 protease using spectral clustering. *Bioinf*, 24(10): 1243-1250.

[39] Luo, F., Yang, Y., Chen, C.-F., Chang, R., Zhou, J. and Scheuermann, R.H. (2007). Modular organization of protein interaction network, *Bioinf*, 23(2): 207-214.

[40] Pachter, L. and Sturmfels, B. (2005). *Algebraic Statistics for Computational Biology*, Cambridge University Press.

[41] Paintsil, E., Margolis, A., Collins, J. and Alexander, L. (2006). The contribution of HIV fitness to the evolution pattern of reverse transcriptase inhibitor resistance. *J Med Virol*, 78: 425-43.

[42] Quinones-Mateu, M.E., Gao, Y.,Ball, S.C., Marozsan, A.J., Abraha, A. and Arts, E.J. (2002). In vitro intersubtype recombinants of human immunodeficiency virus type 1: Comparison to recent and circulating in vivo recombinant forms. *J Virol*, 76:9600-9613.

[43] Rhee, S., Gonzales, M., Kantor, R., Betts, B., Ravela, J. and Shafer. R. (2003). Human immunodeficiency virus reverse transcriptase and protease sequence database, *Nucl Acids Res*, 31(1): 298-303.

[44] Rhee, S., Liu, T., Holmes, S. and Shafer, R. (2007). HIV-1 subtype B protease and reverse transcriptase amino acid covariation. *PLoS Comput Biol*, 3(5): e87.

[45] Rhee, S. Y., J. Taylor, G. Wadhera, A. Ben-Hur, D. L. Brutlag, and R. W. Shafer. (2006). Genotypic predictors of human immunodeficiency virus type 1 drug resistance. *Proc Natl Acad Sci USA*, 103:17355-17360.

[46] Richman, D.D., Havlir, D., Corbeil, J., Looney, D., Ignacio, C., Spector, S.A., Sullivan, J., Cheeseman, S., Barringer, K., Pauletti, D., et al. (1994). Nevirapine resistance mutations of human immunodeficiency virus type 1 selected during therapy, *J Virol*, 68:1660-1666.

[47] Sevin, A. D., DeGruttola, V., Nijhuis, M., Schapiro, J.M., Foulkes, A.S., Para, M.F. and Boucher, C.A. (2000). Methods for investigation of the relationship between drug-susceptibility phenotype and human immunodeficiency virus type 1 genotype with applications to AIDS clinical trials group 333. *J Infect Dis*, 182:59-67.

[48] Shafer, R. and Schapiro, J. (2008). HIV-1 drug resistance mutations: an updated framework for the second decade of HAART. *AIDS Rev*, 10: 67-84

[49] Stanley, R.P. (1997). *Enumerative combinatorics*, Vol. 1, volume 49 of Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge.

[50] Svicher, V., Ceccherini-Silberstein, F., Erba, F., Santoro, M., Gori, C., *et al.* (2005). Novel human immunodeficiency virus type 1 protease mutations potentially involved in resistance to protease inhibitors. *Antimicrob Agents Chemother*, 49: 2015-2025.

[51] Wang, K., Jenwitheesuk, E., Samudrala, R. and Mittler, J.E. (2004). Simple linear model provides highly accurate genotypic predictions of HIV-1 drug resistance *Antivir Ther*, 9: 343-52.

[52] Whiteside, A. (2008). *HIV/AIDS: A Very Short Introduction*, Oxford University Press.

[53] Wu, T.D., Schiffer, C.A., Gonzales, M.J., Taylor, J., Kantor, R., *et al.* (2003) Mutation patterns and structural correlates in human immunodeficiency virus type 1 protease following different protease inhibitor treatments. *J Virol*, 77: 4836-4847.

[54] Vermeiren, H., Van Craenenbroeck, E., Alen, P., Bacheler, L., Picchio, G. and Lecocq, P. (2007). Prediction of HIV-1 drug susceptibility phenotype from the viral genotype using linear regression modeling. *J Virol Methods*, 145:47-55.