A COMBINATORIAL APPROACH TO BIOLOGICAL STRUCTURES AND NETWORKS IN PREDICTIVE MEDICINE

A Dissertation
Presented to
The Academic Faculty

By

Anna Kirkpatrick

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Math
College of Sciences

Georgia Institute of Technology

July 2021

A COMBINATORIAL APPROACH TO BIOLOGICAL STRUCTURES AND NETWORKS IN PREDICTIVE MEDICINE

Thesis committee:

Dr. Joshua Cooper
Department of Mathematics
University of South Carolina

Dr. Cassie Mitchell
Biomedical Engineering
Georgia Institute of Technology

Dr. Dana Randall
School of Computer Science
Georgia Institute of Technology

Dr. Lauren Steimle
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Francesca Storici
School of Biological Sciences
Georgia Institute of Technology

Dr. Prasad Tetali
School of Mathematics and School of Computer Science
Georgia Institute of Technology

Date approved: July 9, 2021

ACKNOWLEDGMENTS

I must first and foremost thank my advisors, Cassie Mitchell and Prasad Tetali, without whom I would not have been able to complete this work. In addition to providing much helpful feedback and suggestions on the research, both Prasad and Cassie have been a source of significant personal support as I navigated the completion of this thesis. I especially want to thank Cassie for her openness about her own disability and willingness to serve as a mentor to me as I have learned to navigate the intersection of my own disability and my academic work.

I would also like to thank the other members of my committee: Joshua Cooper, Dana Randall, Lauren Steimle, and Francesca Storici. All have provided valuable feedback on my work and aided me in refining my thesis. I owe an additional thank you to Dana for introducing me to the tools for analyzing Markov chain mixing time. Though we did not end up collaborating directly, the knowledge I gained from conversations with Dana was invaluable as I worked on the material of chapter 4 with my collaborator Kalen Patton. I also want to thank Josh for serving as the official reader for my thesis, for mentoring me on multiple research projects when I was an undergraduate student at the University of South Carolina, and for encouraging me to apply to the ACO program at Georgia Tech.

I also want to thank my cohort of peers in the ACO program: Digvijay Boob, Matthew Fahrbach, Kevin Lai, Samantha Petti, Samira Samadi, and Saurabh Sawlani. Their willingness to form study groups and collaborate on problem sets made the first few years of my graduate program considerably smoother.

I must also thank my many collaborators. My undergraduate mentees Kalen Patton and Chidozie Onyeze contributed to multiple projects, and the experience of working with them has helped me develop my own skills as a teacher and a mentor. I also want to thank Chidozie for helping me correct several formatting issues in this document. I owe a big thank you to everyone in the Pathology Dynamics lab for welcoming me into their commu-

nity. I especially want to thank Stephen Allegri, Evie Davalb-hakta, David Kartchner, Davi Nakajima-An, and Chidozie Onyeze, who all collaborated on the knowledge graph project from chapter 5.

For supporting me and my well-being along this journey, I want to thank my yoga teachers Maya Nenova, Hannah Onians, and Vladimir Tchakarov, as well as my good friend and “yoga buddy” Julia Mayfield.

I am extremely grateful for the support and encouragement of my parents Rick and Penny Kirkpatrick throughout my educational journey.

Finally, I wish to express my gratitude to the funders who made this work possible. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650044 to A.K. Research was also funded by the Georgia Institute of Technology President’s Undergraduate Research Award to K.P., the Georgia Institute of Technology President’s Undergraduate Research Award to C.O., the National Science Foundation CAREER award 1944247 to C.M., Alzheimer’s Association research grant AARG-2018-59104 to C.M., Emory Alzheimer’s Disease Research Center pilot grant to C.M. (P50 AG025688), National Institute of Health Grant R21CA232229 to C.M., and National Science Foundation grant DMS-1811935 to P.T.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AD Alzheimer’s disease

ADAS11 Alzheimer Disease Assessment Scale 11

ADAS13 Alzheimer Disease Assessment Scale 13

ADNI Alzheimer’s disease neuroimaging initiative

APOE4 Apolipoprotein E4

CDR Clinical Dementia Rating

CDR-SB Clinical Dementia Rating Scale Sum of Boxes

CN cognitive normal

HMM hidden Markov model

MCI mild cognitive impairment

MRI magnetic resonance imaging

nearest neighbor database NNDB

NNTM Nearest Neighbor Thermodynamic Model

PET Positron emission tomography

RNA ribonucleic acid

RNA STRAND RNA Secondary Structure and Statistical Analysis Database

SUMMARY

This work concerns the study of combinatorial models for biological structures and networks as motivated by questions in predictive medicine. Through multiple examples, the power of combinatorial models to simplify problems and facilitate computation is explored. First, continuous time Markov models are used as a model to study the progression of Alzheimer's disease and identify which variables best predict progression at each stage. Next, RNA secondary structures are modeled by a thermodynamic Gibbs distribution on plane trees. The limiting distribution (as the number of edges in the tree goes to infinity) is studied to gain insight into the limits of the model. Additionally, a Markov chain is developed to sample from the distribution in the finite case, creating a tool for understanding what tree properties emerge from the thermodynamics. Finally, knowledge graphs are used to encode relationships extracted from the biomedical literature, and algorithms for efficient computation on these graphs are explored.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Combinatorics is a branch of mathematics concerned with counting, selecting, and arranging various types of mathematical objects [1]. The mathematical objects studied in combinatorics can vary widely; some examples include integers, trees, graphs, and matchings. Generally speaking, combinatorics deals with questions of enumeration, existence, and optimization of objects [2, 3]. Enumeration problems ask the question "how many objects of this type are there?" Existence questions ask "is there an object with these properties?" Optimization problems ask "which object in the set is best", according to a criterion of (often practical) interest.

Predictive medicine is a field of study that uses mathematics, statistics, data science and machine learning to develop and implement algorithms that can be used to improve healthcare [4, 5]. More specifically, predictive medicine can provide tools to help identify the cause or mechanisms of a disease, to identify possible treatments for disease, and to directly improve patient care.

A combinatorial approach can bring many benefits to predictive medicine. Many problems in predictive medicine are so complex that direct computational approaches are impractical and direct biological or mathematical inquiry is extremely difficult or impossible. Combinatorics can provide a crucial framework in the form of a combinatorial model. Using such a model, a complex biomedical problem can be simplified to the point where meaningful computation and/or direct mathematical analysis is possible. Because all models involve simplification and assumptions, the results of such computation or analysis should then be verified against the real data, experimentally tested, or simply used to generate hypothesies and point out possible areas for future research.

Combinatorial models also provide clear benefits with respect to computation. In order for the algorithms generated by predictive medicine to be useful, they have to be efficient enough that researchers and clinicians can actually use them on the computing hardware available. The structure of a combinatorial model often allows for careful algorithm design and optimization. Additionally, some models can be used as the basis for approximation algorithms, which may be much faster than exact computation while still delivering useful results.

Broadly, this thesis approaches combinatorial models in predictive medicine in a few distinct ways. First, the models are analyzed with the goal of drawing conclusions, either about the modeled phenomenon or about the model itself. This style of analysis is dominant in chapter 2 and chapter 3. Second, the models are used as a starting point for computation, and the main results concern how efficiently quantities can be computed. These computation-inspired questions are central to chapter 4 and chapter 5.

Another persistent theme in both the models and the computation is stochasticity. Many of the models are themselves stochastic. For example, the RNA secondary structure models from chapter 3 and chapter 4 are stochastic due to the stochastic nature of physics on a molecular scale. In contrast, the continuous time Markov models used to study disease progression in chapter 2 use stochasticity as a stand-in for the uncertainty in our current state of knowledge. Stochasticity also proves useful when examining the question of how to efficiently compute quantities of interest. Randomized algorithms feature prominently in chapter 4 and chapter 5.

1.1 Overview of chapters

We begin with chapter 2, which concerns the use of continuous time Markov models to study the progression of Alzheimer's disease. Alzheimer's disease is a serious neurodegenerative disease with significant personal and societal costs. The development of Alzheimer's disease is gradual, and diagnosis in early stages is often missed [6, 7]. It is also currently quite difficult to predict who will develop Alzheimer's disease or how quickly an individual with Alzheimer's disease will progress to the more severe stages [8]. The goal of this work is to provide models and identify key characteristics that can help to predict the progression of the disease. Ultimately, the objective is to enable clinicians to provide more insight to patients and families about the likely course of the disease. This additional information has the potential to improve healthcare by reducing the stress caused by uncertainty and enabling patients and their families the plan for the future.

The key combinatorial model from this project is the continuous time Markov model, and much of the work can be viewed as an optimization problem: of all possible parameters for this Markov model, which combination best fits the observed data? The continuous time Markov models also incorporate various covariates from the dataset, allowing for the study of which covariates are the best predictors of disease progression.

In chapter 3 and chapter 4, we study a thermodynamic model for ribonucleic acid (RNA) secondary structures. In the context of predictive medicine, understanding RNA secondary structures is a crucial step to understanding the mechanisms behind disease. In biology, form and function are nearly always closely linked. Through better understanding of the structure of RNA, its specific function may be revealed. Known functions of RNA include information transfer, gene regulation, and catalysis of chemical reactions [9, 10, 11]. RNA also comprises many viral genomes [9]. A better understanding of RNA structure has the potential to shed light on the mechanisms of many diseases.

The specific combinatorial model of RNA secondary structure studied here is focused on the branching behavior of these structures. Structures are modeled by plane trees, and a thermodnyamic model defines a Gibbs probability distribution on the plane trees. Plane trees belong to a well-studied class of mathematical objects known as Catalan objects. The rich combinatorial literature on plane trees makes them especially attractive as a model for RNA structure. Both chapter 3 and chapter 4 build on this literature.

In chapter 3 we study the above-mentioned Gibbs distribution on plane tress in the limit as the number of edges in the plane trees approaches infinity. Various aspects of the work can be understood as enumeration problems. For example, "how many plane trees have a given energy value?" The results ultimately shed light on the boundaries of what structures this model can predict, regardless of the thermodynamic parameters selected. Since the plane tree model is a special case of a more general model (the Nearest Neighbor Thermodynamic Model) widely used in the study of RNA secondary structure [12], understanding these limitations may enable future researchers to develop even better models. In turn, better models will lead to better structural predictions, which will facilitate the study of the mechanisms behind numerous diseases.

In chapter 4, we study the same thermodynamic model for RNA secondary structure from a different perspective. Instead of looking at the limiting behavior of the probability distributions, we devise an algorithm for sampling from the probability distribution on plane trees of some fixed size. The sampling algorithm is based on the Metropolis algorithm, and the key result is a mixing time bound on the resulting Markov chain. Improving the mixing time bound, we encounter many combinatorial enumeration problems. As a simple example, we must answer questions like "how many plane trees with n edges have exactly m vertices with exactly one child?" Choices made in the algorithm design and proof strategy also have elements of combinatorial optimization, as we are seeking the smallest achievable mixing time bound.

This mixing time result has potential utility for performing computational experiments that can help researchers understand what types of branching properties are typical for the given thermodynamic model. This information, in turn, is important for interpreting the output of secondary structure prediction software. As with the previous chapter, better understanding of RNA secondary structure models is a stepping stone to better models, better structure predictions, and, ultimately, better understanding of disease mechanisms.

Finally, chapter 5 concerns the question of navigating a knowledge graph built from biomedical paper abstracts. Consider, as a motivating example, the problem of identifying, from the set of all known drugs, potential drug treatments for a disease. From the vast set of known drugs, it is reasonably likely that a few might provide benefit for a given disease, but it would be impractical and likely unethical to simply test all of the drugs. (Even when testing in animal models, ethical considerations demand that animal tests only be performed when no alternative methods are available, that the minimum necessary number of animals be used, and that care is taken not to unnecessarily duplicate experiments already documented in literature [13].) Instead, a researcher would like to be able to use their knowledge together with the vast amount of information available in the biomedical literature to identify promising drugs for further testing. While this researcher is likely quite familiar with the disease they are studying and the known or hypothesized mechanisms for this disease, in order to identify promising drug candidates, they would also want to be familiar with all candidate drugs, the diseases they are known to treat, and the mechanisms behind all of those diseases and treatments. No single human has the capacity to read and remember all of the relevant literature.

This is where a combinatorial model is useful, specifically a knowledge graph. A knowledge graph is a directed graph that encodes relationships between biomedical concepts [14]. This graph is built using natural language processing techniques on biomedical paper abstracts [15].

This chapter focuses on computing quantities of interest on this graph. In particular, the goal is to efficiently compute HeteSim [14], a similarity score that gives an overall measure of the relatedness of different concepts in the knowledge graph. Developing, refining, and implementing algorithms for HeteSim involves many combinatorial questions around existence, enumeration, and optimization. In fact, all of these questions are raised about paths in the knowledge graph. We ask "is there a path between s and t?" We also need to know "how many paths are there between s and t?" When optimizing our algorithms, we need to know "of the set of all paths between s and t, how many do we need to examine?" in order to accurately estimate certain quantities.

Key improvements to the HeteSim computations include changes to the underlying data structure representing the knowledge graph and the introduction of relevant randomized approximation algorithms. The more efficient algorithms and data structures developed in chapter 5 will allow researchers to compute similarity scores for concepts of interest much more quickly, facilitating the identification of possible drug therapies for disease.

Taken together, these examples will demonstrate the utility of applying combinatorial models in predictive medicine. The following chapters will show how combinatorics and predictive medicine together have the potential to reveal disease mechanisms, identify potential treatments, and improve patient care.

1.2 Technical Background

An overview of the technical background and relevant tools is provided in this section.

1.2.1 Continuous time Markov models and hidden Markov models

In chapter 2, continuous time Markov models are a tool used to model the progression of Alzheimer's disease. A continuous time Markov model is defined by a state space Ω, a transition intensity matrix Q, and an initial state x_0. The model defines a random variable $X(t)$ as a function of time t. By assumption, $X(0) = x_0$ with probability 1. The probability that X changes state is governed by the transition intensity matrix Q. More specifically, the entry q_{rs} is given by

$$q_{rs} = \lim_{\delta \to 0^+} \frac{P\left(X(t+\delta) = s | X(t) = r\right)}{\delta}. \tag{1.1}$$

Note that this definition does satisfy the Markov property. The probability of moving to any given state depends only on the current state, not on any history of prior states.

Continuous time hidden Markov models are an extension of continuous time Markov models where the true state of the Markov chain is hidden. Instead, the observed state has a probability distribution conditional on the true state. More formally, a continuous time hidden Markov model is defined by a state space Ω, a transition intensity matrix Q, an initial state x_0, and an emission probability matrix E. The first 3 quantities are defined as for a continuous time Markov model. The emission probability matrix E has entries e_{rs} given by

$$e_{rs} = P\left(O(t) = s | S(t) = r\right), \tag{1.2}$$

where $O(t)$ is the observed state at time t and $S(t)$ is the true state at time t.

Considerable study has been devoted to algorithms for fitting both discrete and continuous time Markov models and hidden Markov models to data (e.g. [16, 17, 18], also see [19, 20] for an overview) and to applying these models to biomedical problems (e.g. [21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]).

The act of fitting a continuous time Markov model or hidden Markov model to disease progression data has the potential to yield insights on its own, such as a better understanding of the actual rate of disease progression with respect to different disease states. Further insight may be gained by incorporating covariates into the model. As an example, a proportional hazards model as described by Marshall and Jones [27] replaces the original (without covariates) elements of matrix Q by

$$q_{rs}(z(t)) = q_{rs}^{(0)} e^{\beta_{rs}^T z(t)}, \tag{1.3}$$

where $q_{rs}^{(0)}$ are the original entries of Q, β_{rs} is a vector of parameters which must be fit to the data, and $z(t)$ is a vector of (time varying) covariates. The values of β can then be used to draw conclusions about the relation between covariates and disease progression.

1.2.2 Generating functions and analytic combinatorics

In chapter 3, generating functions are used as a tool to obtain asymptotic approximations of sequences. Using Carleman's condition [33], asymptotic results on sequences of moments are extended to results about limiting probability distributions. For a sequence $(a(n))$ for

$n \in \mathbb{Z}_{\geq 0}$, we define the *generating function* corresponding to the sequence to be the formal power series,

$$F(x) = \sum_{n \geq 0} a(n) x^n.$$

Fix $(a(n))$, for $n \in \mathbb{Z}_{\geq 0}$, and let $F(z)$ be the associated generating function. We can treat $F(z)$ as a function over the complex plane. We say $F(z)$ is *analytic* at a point $z_0 \in \mathbb{C}$ if there exists a region around z_0 such that $F(z)$ is differentiable. We say $F(z)$ is analytic on a domain if it is analytic at all points in the domain.

A point, $z_0 \in \mathbb{C}$, is a *singularity* of $F(z)$ if $F(z)$ is not analytic at z_0. Furthermore, that singularity is *isolated* if there exist $\epsilon > 0$ such that $F(z)$ is analytic on the domain $\{z \in \mathbb{C} : 0 < |z - z_0| < \epsilon\}$. Let $F(z)$ be such that all its singularities are isolated. We define a *dominant singularity* of $F(z)$ to be an isolated singularity with minimal distance from the origin.

A key tool comes from the work of Flajolet and Sedgewick [34]: the Transfer Theorem. The Transfer Theorem allows us to deduce asymptotic information about $(a(n))$ using $F(z)$ and its singularities. In turn, this asymptotic information is used to characterize limiting probability distributions.

1.2.3 The Metropolis algorithm and Markov chain convergence

In chapter 4, the key algorithm is the Metropolis algorithm. The Metropolis algorithm, used in the context of Markov chain Monte Carlo, allows one to design a Markov chain that converges to a given probability distribution. While Markov chain Monte Carlo has been used widely in computational biology and other applied sciences (e.g. [35, 36, 37, 38]), convergence of chains is often judged by heuristics, rather than the mathematically-rigorous analysis (see, e.g. [39, 40, 41]. The Metropolis algorithm only guarantees the convergence of a chain to a target distribution; it does not automatically suggest any generally useful bounds on how long that convergence may take [42]. This chapter concerns the question of actually bounding that mixing time for our chain of interest.

The actual mixing time bound is achieved by bounding the spectral gap of the transition probability matrix. The key tools used in bounding the spectral gap include decomposition theorems due to Martin and Randall [43] and Hermon and Salez [44], the latter of which builds on work by Jerrum, Son, Tetali, and Vigoda [45]. Other tools include coupling and comparison of Dirichlet forms, both of which are discussed by Randall [42].

1.2.4 Randomized approximation algorithms

In chapter 5, multiple randomized approximation algorithms are analyzed. An approximation algorithm is an algorithm which returns a value within a specified error (additive or multiplicative) of the true answer, with some known or bounded probability. The power of approximation algorithms lies in their ability, for some problems, to provide a fast approximation to a solution even when computing the exact solution requires exponential time (assuming $\mathbf{P} \neq \mathbf{NP}$). Though approximation algorithms have existed in the literature for some time, Garey, Graham, and Ullman [46] and Johnson [47] both introduced the idea formally in 1973 and 1974, respectively. Since then, the computer science and combinatorics literature has featured many advancements in the field of randomized approximation algorithms. For an overview of basic techniques for designing and analyzing approximation algorithms and more recent results, see [48, 49, 50].

CHAPTER 2
MARKOV MODELS FOR PROGRESSION OF ALZHEIMER'S DISEASE

This chapter represents partial results of an ongoing collaborative project, with collaborators Sri Vivek Vanga, Raghav Tandon, Albert Lee, Lauren Steimle, and Cassie Mitchell. The full results will be submitted to a journal specializing in Alzheimer's disease.

2.1 Introduction

Alzheimer's disease (AD) and related dementia [51] are characterized by decline in memory leading to loss of independence and reliance on caregivers. AD is the sixth leading cause of death in the United States population and the fifth leading cause of death among adults aged $\geq$ 65 years. The impact of Alzheimer's is expected to increase in the US as 15.0 million will have clinical AD or mild cognitive impairment by 2060 [52]. Diagnosis of AD is often missed or delayed [6]. Diagnosis of AD is hard especially at early onset. Several lines of research such as Positron emission tomography (PET) and structural images suggest that AD begins years before its clinical manifestations are obvious [6]. Early and accurate diagnosis is an important problem as early diagnosis can be used to slow down the rate of decline by apt treatment and behavioral therapy [6]. AD broadly progresses through three disease stages: cognitive normal (CN), mild cognitive impairment (MCI) and AD. Patients transition to different stages at varying rates. The ability to identify early features that predict the clinical progression of AD is imperative for clinical trial patient selection and personalized predictive medicine. The objective of the present study is to utilize a form of statistical model called a Markov model to predict the probability of transitioning to each stage using a select number of commonly measured clinical features, including cognitive function assessments, the Apolipoprotein E4 (APOE4) genotype, and standard brain volumes obtained from magnetic resonance imaging (MRI).

Alzheimer's disease neuroimaging initiative (ADNI) [53] has released a dataset tracking patients' Alzheimer's disease progression, including time varying and time independent attributes. The ADNI longitudinal dataset consistently collects measures of disease progression on study volunteers throughout their disease course. A small yet diverse subset of features from the ADNI dataset was used to build an interpretable model with an intent to analyze how different features affect Alzheimer's disease progression. The ADNI clinical dataset includes recruitment, demographics, physical examination and cognitive assessment data. For the present work, three cognitive assessments were included, the ADAS11, ADAS13 and CDRSB.

The Alzheimer Disease Assessment Scale 11 (ADAS11) [54] consists of 11 modalities that evaluate memory, praxis, and language deficiencies. The tests take 30-35 minutes to take and the items score range from 1 to 5. The total ADAS11 score ranges from 0–70 with higher scores suggesting greater impairment. The ADAS11 test has been shown successful in not only identifying Alzheimer's patients from healthy elderly controls, but it has also shown to be effective in rating severity between moderate and late-stage dementia based on decreasing performance on the test items[54]. Critics of ADAS11 [55] state that the test is less effective in rating severity in MCI and mild dementia cases. The Alzheimer Disease Assessment Scale 13 (ADAS13) [56] includes all ADAS11 modalities as well as a test of delayed word recall and a number cancellation or maze task. The inclusion of additional tests helps ADAS13 identify more mild forms of dementia. The ADAS13 scores range from 0 to 85. Just like ADAS11, higher score indicates more severe impairment. The Clinical Dementia Rating (CDR) was introduced by C P Huges et. al. [57] and provides data on inter-rater reliability and comparison with other dementia rating scales. Clinical Dementia Rating Scale Sum of Boxes (CDR-SB) [58, 59] is a modified scoring of the CDR scale. [58] shows that the CDR-SB scores allow for a more granular assessment than the CDR score and shows that the utility of the CDR-SB (over CDR) for diagnosing mild dementia. Results presented by O'Bryant SE et. al. [59] validates the CDR-SB scores by comparison with the (global) CDR score.

Apolipoprotein E4 (APOE4) is the most prevalent genetic risk factor of AD [60]. The APOE4 genetic risk factor is expressed in more than half of AD patients, making it a sought after therapeutic target [61]. The more copies of APOE4 a patient has, the higher their risk for developing dementia. In the ADNI study, the APOE4 column takes values 0 to 2. The APOE4 allele, present in approximately 10-15% of people, increases the risk for Alzheimer's and lowers the age of onset. Having one copy of E4 (E3/E4) can increase your risk by 2 to 3 times while two copies (E4/E4) can increase the risk by 12 times[62]. MRI [63] is a powerful technique for non-invasive imaging of the human brain. MRI of the brain can support the quantitative characterization of neurological conditions such as Alzheimer's disease (AD). MRI can provide informative biomarkers even before clinical symptoms are apparent or irreversible neuronal damage has occurred [64]. ADNI patients have MRI image scans during their baseline and follow-up clinic visits. The collection of images is central to meeting ADNI's objective of developing biomarkers to track both the progression of Alzheimer's disease and changes in the underlying pathology. In the present study, hippocampus and whole brain volume attributes extracted from these MRI images are exported via ADNImerge [65] tabular dataset. The whole brain volume and hippocampus volume have been used in prior work [66] to identify fast AD disease progressors. The present study utilizes the ADNI data set and Markov modeling to answer two important research questions: 1) Which standard clinic variables are most important for predicting which patients will transition to each AD stage and when do these transitions take place? 2) What standard clinical metrics or combination of attributes is most likely to result in a misdiagnosis? For example, what combination of attributes is likely to result in a patient with mild cognitive impairment but not pathological AD to be incorrectly diagnosed as having AD or vice versa? Most prior machine learning studies [67, 68, 69] have focused only on imaging or only on informatics. The present study utilizes the most ubiquitous metrics from the AD clinic and combine them into an interpretable model to better understand the transitions and how well these clinical features predict such transitions. Specifically, Markov models are used to study the longitudinal data and decision trees [70], a form of supervised machine learning, used to study the IID dataset.

2.2 Materials and Methods

This study utilizes the ADNI data set to predict disease stage progression and probability of mis-diagnosis using Markov modeling and supervised machine learning.

2.2.1 Data Sets

This work is based on the ADNImerge [65] dataset, a longitudinal dataset containing clinical and biomarker data from the Alzheimer's disease neuroimaging initiative (ADNI). Each row in ADNImerge corresponds to a patient's visit and the patient's attributes that were collected in that visit. A subset of columns from the ADNImerge dataset were used: participant roster ID (RID), which uniquely identifies a patient; column M in ADNImerge, which captures the relative time of the visit (in months) compared to the baseline visit of the patient; APOE4 genetic risk score; cognitive assessment scores for each visit, including the 11-question and 13-question Alzheimer's Disease Assessment Survey (ADAS11, ADAS13); the clinical dementia rating (CDR) at each visit; and the MRI-captured hippocampus and whole brain volumes measured during each visit. All ADNImerge rows that contain a null (i.e. missing) value for any of the aforementioned attributes were dropped.

There are two dataset variants for the present project, which vary on the number of attributes utilized. Dataset variant 1 includes 4 attributes: APOE4, CDRSB, ADAS11, ADAS13. Dataset variant 2 includes 6 attributes APOE4, CDRSB, ADAS11, ADAS13, hippocampus brain volume, whole brain volume. The "diagnosis" (the label for patient diagnostic stage or class) labelled as DX is included in all variants. Summary statistics for the dataset variants are shown in Table 2.1. Please note that variant 2 has smaller number of data points as variant 2 has more columns which also results in more null values (and more rows necessitated to be dropped). Dataset variants 1 and 2 were each used to construct separate Markov models for analysis. Finally, a separate independent and identically distributed (IID) dataset was constructed to build interpretable supervised learning models. The variant 2 IID dataset is the same as variant 2 with the exception that it only considers the baseline visit for each patient and their eventual diagnosis approximately 24 months after their baseline. Table 2.1 contains a description of the dataset variants. Please note that dataset variant 1 and 2, as mentioned in the table, are derived from ADNImerge data set [65].

Table 2.1: Description and sample sizes for Dataset Variants. Both dataset variants are derived from the ADNImerge data set [65].

Dataset variant	Attributes Contained	Visits Included	Number of data points	Number of patients
Variant 1	APOE4, CDRSB, ADAS11, ADAS13	all visits	9972	2016
Variant 2	APOE4, CDRSB, ADAS11, ADAS13, hippocampus brain volume, whole brain volume	all visits	5595	1602
Variant 2 IID	APOE4, CDRSB, ADAS11, ADAS13, hippocampus brain volume, whole brain volume	Baseline(attributes) and 24-months(DX)	1106	1106

2.2.2 Modeling to predict AD transitions

Broadly speaking, two distinct approaches are used to analyze the importance of each variable in predicting AD transitions from normal, MCI, and AD. Supervised machine learning with a binary classifier was used to identify features of highest importance in predicting patient diagnosis using baseline data. A standard Markov model was utilized to predict transition probabilities for stages of disease progression and to compute corresponding hazard ratios. A hidden Markov model was utilized to assess the probability of mis-diagnosis or an incorrect prediction of a stage transition during disease progression based on temporal attributes collected at each clinic visit.

2.2.3 Binary classifier

The binary classifier, a decision tree model, was designed to provide a simple, interpretable machine learning framework to assess which baseline feature are most important in predicting a patient's diagnosis 24 months later. As noted under Datasets, the binary classifer utilized the variant 2 IID data. Two subsets of data were created containing only two eventual diagnosis (DX) labels: MCI and CN subset; MCI and AD subset. Each subset was randomly partitioned to utilize 70% of the data for training the model and 30% of the data for independent testing of the model. For each subset, a binary classifier model was built with just one feature, and the model performance assessed using the test partition. The corresponding predictive power of each feature is reported and plotted for discriminating the two classes. The framework used to implement and train the decision tree was Python scikit-learn [71].

2.2.4 Standard Markov Model

The R package MSM [29] was used to fit a continuous time Markov model to the data. A continuous time Markov model is defined by state space S and a transition intensity matrix Q. The state space for all Markov models discussed here is

$$S = \{CN, MCI, AD\}. \tag{2.1}$$

Equation 2.1 describes the three possible disease states: control (CN), mild cognitive impairment (MCI) and Alzheimer's Disease (AD). The entries of the transition intensity matrix may be understood informally as the instantaneous transition rate from one state to another. By the Markov assumption, for any $t > 0$, $\delta > 0$ the state at time $t + \delta$ is independent of all states prior to time t. The MSM R package computes a transition intensity matrix Q which maximizes likelihood. The effect of covariates on transition intensity can also be modeled, as described below. Under a proportional hazards model, the entries of the transition intensity matrix, given covariate vector $z(t)$, are given by

$$q_{rs}(z(t)) = q_{rs}^{(0)} exp(\beta_{rs}^{T} z(t)). \tag{2.2}$$

Using this new definition of the transition intensity matrix Q, the MSM package simultaneously finds the values of $q_{rs}^{(0)}$ and β_{rs} (for all r, s) which maximize likelihood. Note that the transition intensity matrix can be written as

$$q_{rs}(z(t)) = q_{rs}^{(0)} \prod_{i=1}^{n} exp((\beta_{rs})_i))^{z_i(t)}, \tag{2.3}$$

where $(\beta_{rs})_i$ is the ith entry of the covariate vector. The quantity $exp(\beta_r s)_i$ is referred to as the hazard ratio for covariate i.

When working with multiple (possibly correlated) covariates, it is not generally possible to interpret the hazard ratio of an individual covariate. To overcome this limitation, a separate single-covariate model was fit for each covariate and the resulting hazard ratios reported. For a given covariate i and transition rs (which implies transition from state r to state s) if the corresponding hazard ratio is greater than 1, then an increase in the value of the covariate corresponds to an increased transition intensity. If the hazard ratio is less than 1, then an increase in the covariate value corresponds to a decrease in transition intensity. If the hazard ratio is exactly 1, then the covariate value does not change the transition intensity. Further, for a hazard ratio greater than 1, a larger hazard ratio shows a larger effect of the covariate on the transition probability. For a hazard ratio less than 1, a smaller hazard ratio shows a stronger effect.

2.2.5 Hidden Markov Model

To better understand what factors may affect misdiagnosis of Alzheimer's disease, the MSM R package was used to fit a hidden Markov model (HMM). In a hidden Markov model, the true state of the Markov chain is not observed. Instead, for each state in the state space, there is a distribution of emission probabilities. The observed value, then, depends on both the state and the emission distribution. Let $S_{i(t)}$ be the true state of participant i at time t. Similarly, let $O_i(t)$ be the observed state of participant i at time t. The misclassification matrix E is defined by $e_{rs} = Pr(O_i(t) = s|S_i(t) = i)$. To investigate which variables may explain misclassification, a multinomial logistic regression model is used satisfying $\log \frac{e_{rs}(t)}{e_{rs_0}(t)} = \gamma_{rs}^T w(t))$, where s_0 is a baseline state and $w(t)$ is a vector of explanatory variables. As with the analysis of hazard ratios, the individual entries of γ_{rs} do not allow the effect of correlated explanatory variables to be understood. As with hazard ratios, a separate model is built for each variable. The MSM package computes E by likelihood maximization using a continuous version of the Baum-Welch algorithm due to Bureau et al [17].

2.3 Results

2.3.1 Binary classifier results

In this section, results are examined from the binary classifier trained on two subsets of the variant 2 dataset. Two subsets of data contain only two eventual diagnosis (DX) labels: MCI and NC subset; MCI and AD subset. The predictive power of a variable for discriminating among two subsequent disease states is interpreted as feature importance of that variable in influencing disease progression to the subsequent state. The orange bars in Figure 2.1 represent the feature importance of a variable for discriminating the MCI and CN subset; hence it is labelled as feature_importance_1_2. The blue bars represent the feature importance of a variable for discriminating the MCI and AD dataset; hence, it is labelled as feature_importance_2_3. In the feature importance plot, all the variables except CDRSB are better at discriminating MCI and AD (feature_importance_2_3) as compared to the discriminating CN and MCI (feature_importance_1_2). CDRSB, on the other hand, is a better at discriminating CN and MCI as compared to discriminating MCI and AD.

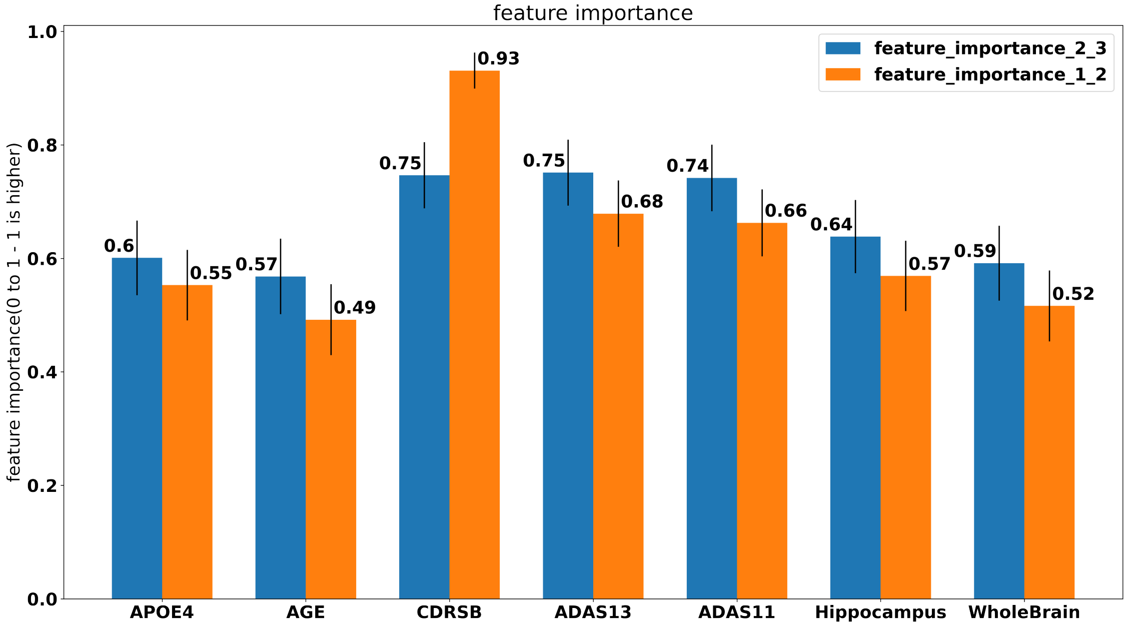


Figure 2.1: Feature importance predicted by binary classifier using the decision tree method and the variant 2 IDD dataset from Table 2.1. The blue and orange bars represent the feature importance of a variable for predicting - future transition to MCI from CN and future transition to AD from MCI. respectively. The error bars represent the 95% CI.

2.3.2 Assessing disease stage transition probability using Markov modeling

Two Markov models on two dataset variants were fit as described in the Data Structures sub-section of the Methods. All attributes present in a specific dataset variant are used as covariates in the Markov model. For dataset 1, the model did converge and appears to have a qualitatively good fit when compared to the binary classification results. No convergence was obtained in the model for dataset 2, likely due to the model having more parameters but fewer data points. Ongoing work on the project beyond the present thesis will examine the quantitative fit of the Markov model using a goodness of fit test appropriate for Markov models. Goodness of fit will also be assessed for the single-covariate models described in this chapter.

The hazard ratios for the two variants of datasets are reported in tabular form in Table 2.2 and Table 2.3, and visualized in the forest plot of Figure 2.2. The hazard ratios indicate the relative importance of the feature in predicting a disease transition. Greater hazard ratios indicate greater association between the feature and the indicated corresponding disease transition. The 95% confidence interval (CI) is illustrated in parentheses following the hazard ratio. Note that the overlapping confidence intervals are a function of data variance (patient heterogeneity) and sample size. Nonetheless, general trends are discernable based on the presented results.

The different orders of magnitude seen in the hazard ratios for different covariates reflects the fact that, in the model, different covariates can have effects on transition intensities of quite different magnitude. As an example, APOE4 has a hazard ratio between 2 and 3 for both transitions. APOE4 is a genetic risk factor which will therefore remain constant for a given patient. In contrast, ADAS11 has a hazard ratio of order 10^4 and 10^5 for the two transitions. ADAS11 is also a cognitive test which would be re-administered at frequent intervals. An elevated ADAS11 score, then, should be a much stronger predictor of a patient transitioning within a narrow time interval, compared with the constant APOE4 test result.

Table 2.2: Hazard ratios of Markov model built from dataset variant 1

Dataset variant 1 (hazard ratios)	CN → MCI Estimated (CI 95%)	MCI → AD Estimated (CI 95%)
APOE4	2.6189e+00 (1.51401,4.5301)	2.9862e+00 (2.26930,3.9296)
ADAS11	1.387e+05 (4.188e+03,4.592e+06)	1.943e+05 (4.961e+04,7.614e+05)
ADAS13	3.131e+04 (2.428e+03,4.038e+05)	4.750e+04 (1.444e+04,1.562e+05)
CDRSB	3.734e+08 (2.659e+06,5.242e+10)	4.687e+05 (8.781e+04,2.501e+06)

Results indicate that APOE4, ADAS11, ADAS13, hippocampus volume, and whole brain volume are better predictors of transition state 2 (MCI) to state 3 (AD) compared to state 1 (CN) to state 2 (MCI). CDRSB, on the other hand, is better predictor of transition state 1 to state 2 compared to state 2 to state 3. However, the aforementioned inference is

Table 2.3: Hazard ratios of Markov model built from dataset variant 2

Data variant 2 (hazard ratios)	CN → MCI Estimated (CI 95%)	MCI → AD Estimated (CI 95%)
APOE4	2.0280e+00 (0.8643,4.759)	3.5307e+00 (2.4838,5.019)
ADAS11	2.551e+04 (1.999e+02,3.257e+06)	2.096e+05 (4.241e+04,1.036e+06)
ADAS13	9.925e+04 (1.634e+03,6.028e+06)	1.218e+05 (2.665e+04,5.571e+05)
CDRSB	1.886e+06 (5.260e+01,6.761e+10)	3.311e+05 (1.148e+03,9.551e+07)
Hippocampus	1.135e-01 (7.064e-03,1.824e+00)	7.419e-04 (2.013e-04,2.734e-03)
Whole Brain	4.121e+00 (0.151043,1.125e+02)	9.474e-03 (0.001545,5.810e-02)

only conclusive for the features of hippocampus and Whole brain; for other features the confidence intervals overlap which indicate a need for more data points (patient sample observations). Results inferences are summarized in Table 2.5.

Table 2.4: Interpretation of presented quantitative study results.

Attribute	Interpretation based on point estimate of hazard ratio	Interpretation based on confidence interval of hazard ratio
APOE4	Better predicator of transition MCI - AD compared to CN - MCI	Need more data to conclude as hazard ratio of CN - MCI transition overlaps with hazard ratio of MCI - AD.
ADAS11	Better predictor of transition MCI - AD compared to CN - MCI	Need more data to conclude as hazard ratio of CN - MCI transition overlaps with hazard ratio of MCI - AD.
ADAS13	Better predictor of transition MCI – AD compared to CN – MCI.	Need more data to conclude as hazard ratio of CN - MCI transition overlaps with hazard ratio of MCI - AD.
CDRSB	Better predictor of transition CN – MCI compared to MCI – AD.	Need more data to conclude as hazard ratio of CN - MCI transition overlaps with hazard ratio of MCI - AD.
Hippocampus	Better predictor of transition MCI - AD compared to CN - MCI	Conclusive.
Whole Brain	MCI - AD compared to CN - MCI.	Conclusive.

2.3.3 What metrics are most likely to lead to a misdiagnosis?

Values in Table 2.5 indicate whether a larger value of the coefficient increases or decreases the probability of misdiagnosis. A value greater than 1 means that an increase in the value of the covariate corresponds to an increased probability of misdiagnosis. A value less than 1 means that an increase in the value of the covariate decreases the probability of misdiagnosis. APOE4 and ADAS11 are more likely to cause misclassification of MCI as CN as compared to misclassifying of AD as MCI. A reversal of trend is noticed for CDRSB

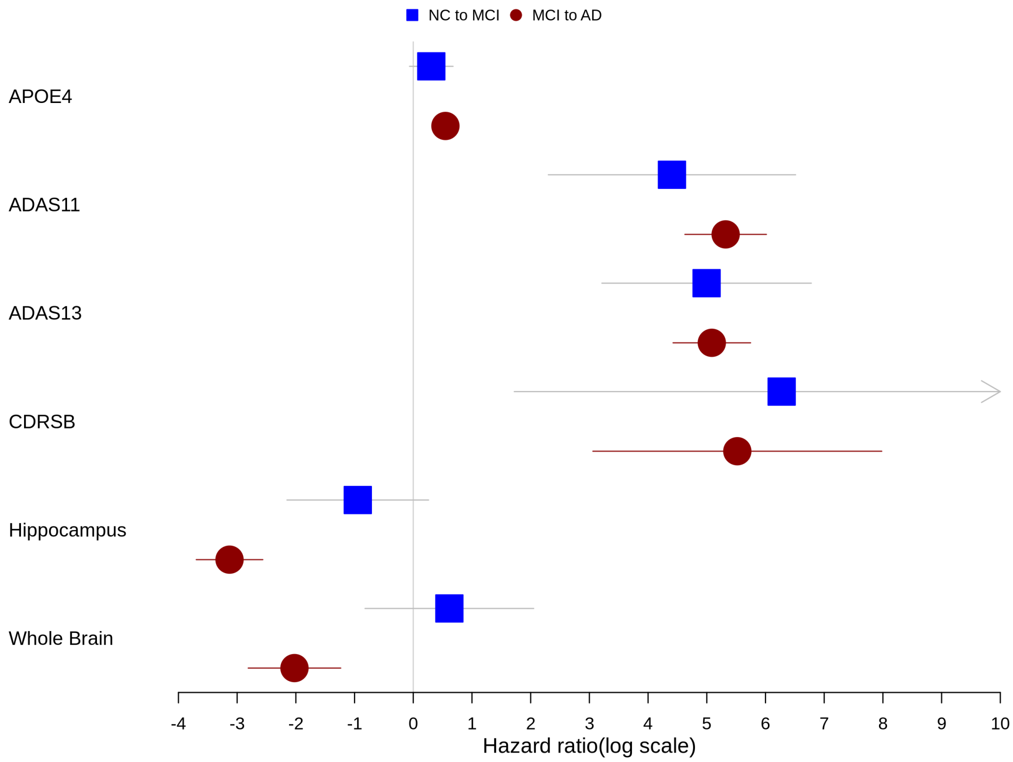


Figure 2.2: Hazard ratios for attributes assess in the Markov model (dataset variant 2). Blue represents transition from cognitively normal to mild cognitive impairment whereas burgundy represents transition from mild cognitive impairment to Alzheimer's Disease. The error bars represent 95% CI.

– it is more likely to cause misclassification of AD as MCI as compared to misclassifying MCI as CN.

2.4 Discussion

The results from the three techniques to assess cognitive disease progression in AD – binary classifiers, Markov models and Hidden Markov model – are discussed in context of the prior literature studies. Binary classifiers, specifically decision trees, predict well the future disease state based on the current disease state and one feature. The performance of the interpretable decision tree is explained by the fact most clinicians primarily utilize basic clinical attributes, namely neuropsychological survey performance, to diagnose AD.

Table 2.5: Diagnostic misclassification coefficients for hidden Markov model.

Dataset variant 1	APOE4 coefficient	ADAS11 coefficient	CDRSB coefficient
Obs MCI | CN	3.4982 (1.1922433,10.264)	8.856e+18 (1.016e+15,7.719e+22)	1.941e-01 (1.589e-01,2.348e-01)
Obs AD | CN	0.4524 (0.0029691,68.932)	1.801e+02 (4.273e-18,7.590e+21)	4.964e-01 (3.34e-01,6.602e-01)
Obs CN | MCI	0.3508 (0.11619,1.059)	3.753e-09(1.843e-12,7.643e-06)	8.321e-07 (1.051e-07,6.585e-06)
Obs AD | MCI	1.6783 (0.7617180,3.698)	3.753e-09 (1.843e-12,7.643e-06)	1.47e-02 (9.700e-03,2.303e-02)
Obs CN | AD	0.4427 (0.00011,1700.936)	2.402e-01 (2.162e-71,2.669e+69)	6.726e-03 (3.314e-03,1.360e-02)

Based on Figure 2.2, APOE4, AGE, ADAS13, ADAS11, hippocampus volume, and whole brain volume are better at discriminating later stages of Alzheimer's disease - MCI and AD. CDRSB is a better at discriminating earlier stages of Alzheimer's disease progression - CN and MCI. Binary classifiers were included in this study primarily to confirm the results of the continuous time Markov models. Since the continuous time Markov models require significant assumptions (in particular, the Markov assumption itself), the consistency between the decision trees and hazard ratios from the Markov models give additional confidence that Markov models are appropriate and hazard ratios are meaningful in this context.

Presented results indicate that underlying disease progression is approximately Markovian and that the covariates selected assist in predicting disease progression transitions. Clinicians look at basic cognitive and neuroanatomical MRI features to build a timeline of disease progression and recommend treatment options for the patient. Based on Figure 2.2, the present study concludes APOE4, ADAS11, ADAS13, hippocampus volume, and whole brain volume are better at predicting transition in later stages of disease progression (MCI to AD) than earlier stages of disease progression (CN to MCI). CDRSB, on the other hand, is better at predicting transition in earlier stages of disease progression (CN to MCI) compared to predicting transition in later stages of disease progression (MCI to AD). Based on Hidden Markov model results in Table 2.5, APOE4 and ADAS11 are better at predicting misclassification in earlier stages of disease progression (MCI as CN) as compared to later stages of misclassification (AD as MCI). However, the sparsity of the data limits the number of cases that can be examined using hidden Markov models. Nonetheless, the trends identified are expected to generalize across potentially larger sample populations or mixed cohorts.

The results from the three techniques - binary classifiers, Markov models and hidden Markov model can be together interpreted to state that for APOE4, ADAS11, ADAS13, hippocampus and whole brain volumes are better for diagnosing later stages of disease progression in Alzheimer's disease. CDRSB, on the other hand, is better for diagnosing earlier stages of disease progression in Alzheimer's disease. The quantitative findings, as shown in Figure 2.1 and Figure 2.2, indicate CDRSB are better at diagnosis for earlier stages of Alzheimer's disease. The present study's CDSRB finding is explained by prior literature [58, 59], which argues that because CDRSB is more granular, it is more helpful in diagnosing mild dementia.

The quantitative findings, as shown in Figure 2.1 and Figure 2.2, indicate ADAS11 is better at diagnosis for later stages of Alzheimer's disease. The present study's ADAS11 are also explained by prior literature. Critics of the ADAS11 [55] state that the test is less effective in rating severity in MCI and mild dementia cases. In fact, this limitation was central to later development of the ADAS13. Based on quantitative results in Figure 2.1 and Figure 2.2, ADAS13 is better at diagnosing mild dementia cases as compared to ADAS11. ADAS13 [56] includes all ADAS11 modalities as well as a test of delayed word recall and a number cancellation or maze task. The inclusion of additional tests helps ADAS13 identify more mild forms of dementia. Please note from Figure 2.1 and Figure 2.2 that ADAS13 is still an inferior neuropsychological metric for diagnosing earlier stages of Alzheimer's disease as compared to CDR-SB.

2.4.1 Future directions

As noted in the Results, work to quantitatively prove the goodness of fit of the Markov models is in progress at the time of this thesis writing. The lack of additional standard or uniformly utilized clinic predictors that assess the transition from CN to MCI exacerbate the difficulty in predicting an early Alzheimer's diagnosis in the absence of large genetic or proteomic testing, the latter of which is not typically available to the general population. The current study quantitatively highlights the need to invest in ubiquitous and easily accessible clinical metrics that better assist in early diagnosis of Alzheimer's disease.

More broadly, this chapter adds to a substantial body of work using Markov models and hidden Markov models to better understand disease progression. For example, such models have been applied to glaucoma [21], complications after lung transplantation [22], coronary occlusive disease after heart transplant [23], is that you Chido various cancers [24, 30, 31, 32], progression of HIV infection [25, 26], and development of diabetes complications [27, 28]. In principle, Markov models could be applied to any disease with multiple stages where a Markov assumption is appropriate. In other words, the probability of transitioning to a given stage of disease with in a given (small) time interval must depend only on the current disease state, not on any additional history. This assumption might be questioned in many progressive disease contexts, where intuition could lead one to believe that the probability of transitioning to a more advanced disease state would increase with the amount of time spent in the current state. However, as seen by the breadth of literature, the Markov assumption appears to be reasonable in a wide variety of disease contexts. Additionally, as seen in this chapter, large quantities of data are generally necessary for fitting meaningful Markov models, so applicability of these techniques will be limited to settings where such data are available. Further, hidden Markov models have proven useful in a variety of contexts where the true disease state cannot be known with certainty (but where a Markov assumption can still be reasonably applied) (e.g. [22, 26, 32]). Markov models and hidden Markov models should be considered as valuable tools for the analysis of longitudinal patient data when attempting to study disease progression.

2.4.2 Limitations

The study results quantitatively show the sparsity of the ADNI dataset through overlapping confidence intervals of parameter estimates for all three techniques - binary classifiers, Markov models and Hidden Markov model. Because of the sparsity of data, advanced techniques like Hidden Markov models were used to analyze only three covariates – APOE4, ADAS11, CDRSB. In order to further analyze the ADNI dataset and apply more advanced techniques, there is a need for increased data density by possibly onboarding more patients or aggregating data across other longitudinal cohorts.

CHAPTER 3
ON THE ASYMPTOTIC DISTRIBUTIONS OF CLASSES OF SUBTREE ADDITIVE PROPERTIES OF PLANE TREES UNDER THE NEAREST NEIGHBOR THERMODYNAMIC MODEL

The content of this chapter has been submitted to the Online Journal of Analytic Combinatorics, with co-author Chidozie Onyeze.

3.1 Background and Introduction

This chapter uses techniques from analytic combinatorics to explore probability distributions arising from questions in molecular biology. Specifically, the questions explored are inspired by the problem of RNA secondary structure prediction.

3.1.1 Overview of results

This chapter examines a model of RNA secondary structure in which secondary structures are modeled by plane trees. As defined more rigorously in section 3.2, we consider the set of all plane trees with n edges under a Gibbs distribution, where the energy of each tree depends on its degree sequence and root degree. The energy function is also determined by 3 thermodynamic parameters, which we treat as fixed: (α, β, γ). In this chapter, we treat these parameters as arbitrary real numbers. However, specific values of interest do arise from the thermodynamic models used in molecular biology; the biological motivation is discussed further in subsection 3.1.2. For a wide class of properties of plane trees, we show a relationship in their asymptotic distributions under different values of (α, β, γ).

One such property is the path length. The path length of a plane tree is the sum of the distances from each vertex of the tree to the root. It is known that the path length of all plane trees with n edges is Airy distributed asymptotically as $n \to \infty$ (see Theorem 5).

This case, with the plane trees being uniformly distributed, corresponds to the thermodynamic parameters $(0, 0, 0)$. Applying Theorem 42 to the path length property allows us to relate the asymptotic distribution of path length under arbitrary thermodynamic parameters (α, β, γ) to this known result. As shown in Corollary 43, we can explicitly state the asymptotic distribution of path length under arbitrary thermodynamic parameters.

For some properties, such as the path length, the asymptotic distribution in the uniform case is known. For many other properties, such as the sum of the distances from each leaf to the root, this distribution is not known. Therefore in addition to our main result relating asymptotic distributions of plane tree properties under different thermodynamic parameters, we also develop some tools which are useful for determining the asymptotic distributions of certain properties in the uniform case. Combining these tools with the main result discussed above allows us to obtain explicit forms for asymptotic distributions for a wide variety of plane tree properties under arbitrary thermodynamic parameters. More specifically, in Theorem 17, we relate the asymptotic distribution of a class of plane tree properties, which we call simple subtree additive properties, to the asymptotic distribution of path length, both in the uniform case. As shown in Corollary 31, this theorem allows us to deduce that the total leaf to root distance is also distributed asymptotically as an Airy random variable.

3.1.2 Biological background

We now proceed with some exposition on RNA secondary structure and prior work in this area. The reader interested only in the mathematics may skip this section.

RNA secondary structure

RNA is an important biological polymer with roles including information transfer, regulation of gene expression, and catalysis of chemical reactions. The *primary structure* of an RNA molecule is the sequence of nucleotides in the polymer. RNA nucleotides are adenine, cytosine, guanine, and urasil, which we frequently abbreviate as A, C, G, and U, respectively. The primary structure, therefore, may simply be understood as a string of A's, C's, G's, and U's. Because RNA is single-stranded, it has the capacity to form nucleotide-nucleotide bonds with itself. The set of such bonds is the *secondary structure* of an RNA molecule. The bonds A-U, C-G, and G-U are considered canonical, and our the only bonds considered by the model presented here. The *tertiary structure* of an RNA molecule is its three-dimensional shape. Though tertiary structure ultimately is most relevant to the determination of function, it is also very difficult to deduce with current laboratory techniques. Therefore, secondary structure is often used as a first step in the process of predicting tertiary structure [72, 73]. In fact, secondary structure is often an input to tertiary structure prediction algorithms [74, 75, 76, 77].

One of the main computational tools for predicting RNA secondary structures is thermodynamic free energy minimization using Nearest Neighbor Thermodynamics Modeling (NNTM) [78, 79, 80]. Under the NNTM, the free energy of a structure is computed as the sum of the free energy of its various substructures. This free energy is in turn used in algorithms to predict secondary structure given an RNA sequence, see, e.g., [81, 82, 83]. Though such algorithms perform reasonably well on short RNA sequences, performance rapidly degrades once sequence length exceeds a few hundred nucleotides.

Multiloops and branching

This chapter investigates an aspect of RNA secondary structure that becomes more significant as sequence length grows: multiloops. A *multiloop* is a place where 3 or more helices meet in an RNA secondary structure. Multiloops are not predicted well by the current NNTM energy assignments [84]. The number and type of multiloops determines the branching behavior of an RNA secondary structure.

We study a model for RNA secondary structure first presented by Hower and Heitsch [12]. This model isolates the multiloops and branching properties of secondary structure, allowing their study without the necessity to consider the identity of individual base pairs. Under the model, secondary structures are placed in bijection with plane trees. The minimum energy structures under the model were characterized by Hower and Heitsch in the original paper, but this leaves open the question of the full Gibbs distribution of possible structures, as well as the question of characterizing asymptotic behavior of the distribution. Kirkpatrick et al. [85] have shown the existence of a polynomial-time Markov chain-based algorithm for sampling from the Gibbs distribution on structures of a fixed size. Bakhtin and Heitsch [86] analyzed a simplification of the model and determined degree sequence properties of the distribution of plane trees asymptotically.

Several properties are used to describe the overall branching behavior of an RNA secondary structure. In particular, ladder distance and contact distance are used to characterize various aspects of a molecule's shape, size, and branching structure (see, e.g. [87, 88]). As discussed in subsection 3.2.2, ladder distance and contact distance correspond to Wiener index and path length, respectively, of plane trees.

We will examine the distribution of several plane tree properties asymptotically under the full version of the Hower and Heitsch model. Under an assumption that the root degree is bounded, the theorems developed will allow us to characterize the asymptotic distribution of many properties of RNA secondary structures under the Nearest Neighbor Thermodynamic Model (NNTM). We will further show that altering the parameters (α, β, γ) only changes the property distribution by a constant multiple. When the assumption that the root degree is bounded is removed, we will still obtain analogous results for parameter sets of the form $(\alpha, \beta, 0)$.

3.1.3 Structure of this chapter

In section 3.2, we give an overview the necessary mathematical preliminaries (including relevant known results). In subsection 3.3.1, we construct generating functions counting plane trees. In subsection 3.3.2, we relate the moments (and hence the distribution) of a class of subtree additive properties we call simple. In subsection 3.3.3, we apply the generating functions in subsection 3.3.1 to relate the moments of a class of subtree additive properties under the uniformly weighted distribution on the plane trees to the same properties under the non-uniformly weighted distribution on the plane trees arising from the Nearest Neighbor Thermodynamic Model. We, hence, show that there exists a constant that translates the asymptotic random variable in the uniformly weighted case to the asymptotic random variable in the non-uniformly weighted case. In subsection 3.4.1, we provide some miscellaneous enumeration results on plane trees.

3.2 Mathematical Preliminaries

3.2.1 Plane Trees and their Properties

A *plane tree* is a rooted ordered tree. Let $\mathfrak{T}_n$ denote the set of plane trees on n edges. Let $\mathfrak{T}_{\leq k} = \cup_{n\leq k}\mathfrak{T}_n$. It is well-known that $|\mathfrak{T}_n|$ is given by the nth Catalan number, $C_n = \frac{1}{n+1}\binom{2n}{n}$. We define the *down degree* of a vertex to be the degree of the vertex when considering the root and one less than the degree of the vertex for all other vertices. We define a *leaf* to be a non-root vertex with down degree 0 and an *internal node* to be a non-root vertex with down degree 1. For a plane tree T, let $v(T)$, $n(T)$, $d_0(T)$, $d_1(T)$ and $r(T)$ be the number of vertices, edges, leaves, internal nodes and root degree of T, respectively. Let $\mathcal{V}(T)$ be the vertex set of T and let $\overline{\mathcal{V}}(T)$ be the vertex set excluding the root vertex. For $v \in \mathcal{V}(T)$, let T_v be the subtree of T that contains all descendant of v (including v).

For plane trees $T_1 \in \mathfrak{T}_n$ and $T_2 \in \mathfrak{T}_m$ for some $m, n \geq 0$, we define the join of T_1 and T_2, $T_1 \ltimes T_2$, to be the tree formed by adding a new edges to leftmost side of the root of T_1 and attaching T_2 to this new edge. Note that $T_1 \ltimes T_2 \in \mathfrak{T}_{n+m+1}$. Notice that for a tree $T \in \mathfrak{T}_{>0}$, there is unique $T_1, T_2 \in \mathfrak{T}_{\geq 0}$ such that $T = T_1 \ltimes T_2$.

We define a *property* of a plane tree to be a function $\mathcal{P} : \mathfrak{T}_{\geq 0} \to \mathbb{R}$. We define the 2 types of properties we will consider from this point forward: *additive properties* and *subtree additive properties*. A property $\mathcal{P}$ is *additive* if, for $T \in \mathfrak{T}_{\geq 1}$ such that $T = T_1 \ltimes T_2$ for some T_1, T_2,

$$\mathcal{P}(T) = \mathcal{P}(T_1) + \mathcal{P}(T_2) + f(T_2),$$

where $f(T) : \mathfrak{T}_{\geq 0} \to \mathbb{R}_{\geq 0}$. For a tree T, let $v_1, \cdots, v_d$ be the child vertices of the root vertex of T. By repeated use of the above definition, we see that

$$\mathcal{P}(T) = \mathcal{P}(T^*) + \sum_{i=1}^{d} f(T_{v_i}) + \sum_{i=1}^{d} \mathcal{P}(T_{v_i}), \tag{3.1}$$

where d is the degree of the root of T and T^* is the tree on 1 vertex.

This is similar to the notion of an additive functional as described by Janson [89]. An additive functional is a function $F : \mathfrak{T}_{\geq 0} \to \mathbb{R}$ such that

$$F(T) = f(T) + \sum_{i=1}^{d} F(T_{v_i}),$$

where $f : \mathfrak{T}_{\geq 0} \to \mathbb{R}$ is known as the toll function. Due to $(Equation\ 3.1)$, we note that an additive property (as we have defined it) is an additive functional with toll function f where there exist a function $f^* : \mathfrak{T}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that $f(T) = c + \sum_{i=1}^{d} f^*(T_{v_i})$ where $c = \mathcal{P}(T^*)$. We will borrow the terminology of additive functionals and call the tuple (f, c) the toll of the additive property. It should be clear that a subtree additive property is uniquely determined by its toll. Thus, we will denote the additive property with a given toll by $\mathcal{P}^{(f,c)}$. We will call f, in the toll, the toll function of the subtree additive property. We will call an additive property non-negative integer valued if the co-domain of the toll function is a subset of $\mathbb{Z}_{\geq 0}$ (and $c \in \mathbb{Z}_{\geq 0}$).

It can also be shown inductively that, for $c, c_1, c_2 \in \mathbb{R}$, $f_1, f_2 : \mathfrak{T}_{\geq 0} \to \mathbb{R}$ and $T \in \mathfrak{T}_n$,

$$\mathcal{P}^{(c_1 \cdot f_1 + c_2 \cdot f_2, c)}(T) = c_1 \cdot \mathcal{P}^{(f_1,0)}(T) + c_2 \cdot \mathcal{P}^{(f_2,0)}(T) + c \cdot \mathcal{P}^{(0,1)}(T). \tag{3.2}$$

A property $\mathcal{P}$ is *subtree additive* if, for $T \in \mathfrak{T}_{\geq 1}$,

$$\mathcal{P}(T) = \sum_{v \in \overline{\mathcal{V}}(T)} f(T_v, T).$$

where $f : \mathfrak{T}_{\geq 0} \times \mathfrak{T}_{\geq 0} \to \mathbb{R}$, and $\mathcal{P}(T^*) = 0$. We will call such a property simple if

$$\mathcal{P}(T) = \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}'(T_v)$$

where $\mathcal{P}'$ is a non-negative integer valued additive property. In this case, we call $\mathcal{P}$ the subtree additive property induced by $\mathcal{P}'$.

For a plane tree T, the *energy* of the tree is given by

$$E(T) = \alpha d_0(T) + \beta d_1(T) + \gamma r(T) \tag{3.3}$$

where $\alpha, \beta, \gamma \in \mathbb{R}$ are parameters of the energy function.

For fixed $n \in \mathbb{N}$ and parameters $\alpha, \beta, \gamma \in \mathbb{R}$ and property $\mathcal{P}$, we define the random variable $\mathcal{P}_{(\alpha,\beta,\gamma)}(\mathfrak{T}_n)$ to be $\mathcal{P}(T)$ for a plane tree $T \in \mathfrak{T}_n$ selected at random with probability $\frac{e^{-E(T)}}{\mathcal{Z}_{(n,\alpha,\beta,\gamma)}}$ where $\mathcal{Z}_{(n,\alpha,\beta,\gamma)}$ is a normalizing constant given by $\mathcal{Z}_{(n,\alpha,\beta,\gamma)} = \sum_{T \in \mathfrak{T}_n} e^{-E(T)}$. For convenience, we will denote $\mathcal{P}_{(0,0,0)}(\mathfrak{T}_n)$ simply as $\mathcal{P}(\mathfrak{T}_n)$.

Let $\mathcal{P}$ and $\overline{\mathcal{P}}$ be properties and let $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2 \in \mathbb{R}$ be parameters. We use

$$\mathcal{P}_{(\alpha_1,\beta_1,\gamma_1)}(\mathfrak{T}_n) \overset{d}{\longleftrightarrow} \overline{\mathcal{P}}_{(\alpha_2,\beta_2,\gamma_2)}(\mathfrak{T}_n)$$

to imply that there exist a random variable W such that as $n \to \infty$,

$$\mathcal{P}_{(\alpha_1,\beta_1,\gamma_1)}(\mathfrak{T}_n) \xrightarrow{d} W \qquad \text{and} \qquad \overline{\mathcal{P}}_{(\alpha_2,\beta_2,\gamma_2)}(\mathfrak{T}_n) \xrightarrow{d} W,$$

where $\xrightarrow{d}$ denotes convergence in distribution. In this case, we will say that $\mathcal{P}_{(\alpha_1,\beta_1,\gamma_1)}(\mathfrak{T}_n)$

and $\mathcal{P}_{(\alpha_2,\beta_2,\gamma_2)}(\mathfrak{T}_n)$ are equivalent in distribution.

We now state the well-known Carleman's condition [33] which tells us that, under certain conditions (which the random variables we will consider satisfy), to show equivalence in distribution, it is sufficient to show equality is asymptotic moments.

Theorem 1 (Carleman's condition). *Let X be a random real-valued variable and let $m_k = \mathbb{E}[|X|^k] < \infty$ for all $k \geq 0$. If*

$$\sum_{k=1}^{\infty} |m_{2k}|^{-\frac{1}{2k}} = \infty,$$

then there exists a unique distribution with moments m_k.

3.2.2 Examples

Example 2. It can be shown inductively that the number of edges in a tree is given by the additive property with toll function $(f, 0)$ where $f(T) = 1$ for all $T \in \mathfrak{T}_{\geq 0}$. We will denote this by $\mathcal{P}^e$. Similarly, the number of vertices in a tree is given by the additive property with toll function $(0, 1)$ where 0 is the zero function. We will denote this by $\mathcal{P}^v$.

Example 3. For $f(T) = 1$ when $T = T^*$ and $f(T) = 0$ otherwise, we observe that $\mathcal{P}^{(f,0)}(T)$ represents the number of leaves in T. We will denote this by $\mathcal{P}^{d_0}$. Similarly, for $f(T) = 1$ when T has root degree 1, and $f(T) = 0$ otherwise, $\mathcal{P}^{(f,0)}(T)$ represents the number of internal nodes in T. We will denote this by $\mathcal{P}^{d_1}$. Note that for $T \in \mathfrak{T}_n$, $\mathcal{P}^{d_0}(T) \leq n$ and $\mathcal{P}^{d_1}(T) < n$.

Example 4. We define the *path length* of T, $\mathcal{P}^{PL}(T)$, to be the sum of the edge distances from each vertex of T to the root. In the biological context, this quantity is also known as the *total contact distance*. We can observe that $\mathcal{P}^{PL}(T)$ is a simple subtree additive property since

$$\mathcal{P}^{PL}(T) = \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^v(T_v) = \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^e(T_v) + n$$

where $n = \left|\overline{\mathcal{V}}\right|$ is the number of edges in T. This holds because the number of paths from vertices to the root that utilize an edge e is the number of vertices in the subtree directly below e.

It can be shown from the work of Takács [90] or more directly from the work of Janson [91] that the following result about the distribution of the path length holds.

Theorem 5. *As $n \to \infty$,*

$$\frac{\mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}} \xrightarrow{d} \int_0^1 e(t)\,dt$$

where $e(t)$ is a normalized Brownian excursion on $[0,1]$. Thus, it is Airy Distributed. Furthermore,

$$\lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}}\right)^k\right] \sim \frac{6k}{\sqrt{2}}\left(\frac{k}{12e}\right)^{\frac{k}{2}}$$

as $k \to \infty$.

Example 6. We define the *Wiener index* of T, $\mathcal{P}^{WI}(T)$, to be the sum of the edge distances between any 2 vertices. In the biological context, this quantity is also known as the *total ladder distance*. We can observe that $\mathcal{P}^{WI}(T)$ is a subtree additive property since

$$\mathcal{P}^{WI}(T) = \sum_{v\in\overline{\mathcal{V}}(T)} \mathcal{P}^v(T_v)(\mathcal{P}^v(T) - \mathcal{P}^v(T_v)).$$

This holds because the number of paths between vertices that utilize an edge e is the number of unordered pairs of vertices, one from the subtree below e and the other not from that subtree.

From the work of Janson [91], the following result about the distribution of the Wiener index holds.

Theorem 7. *As $n \to \infty$,*

$$\frac{\mathcal{P}^{WI}(\mathfrak{T}_n)}{\sqrt{2n^5}} \xrightarrow{d} \int\int_{0<s<t<1} (e(s) + e(t) - 2\min_{s\le u\le t} e(u))\,ds\,dt$$

where $e(t)$ is a normalized Brownian excursion on $[0,1]$.

Example 8. We define the *total leaf to root distance* of T, $\mathcal{P}^{LR}(T)$, and the *total internal node to root distance* of T, $\mathcal{P}^{IR}(T)$, to be the sum of the edge distances from every leaf vertex to the root and the sum of the edge distances from every internal node to the root, respectively. We can observe that $\mathcal{P}^{LR}(T)$ and $\mathcal{P}^{IR}(T)$ can be described in terms of a simple subtree additive properties as follows.

$$\mathcal{P}^{LR}(T) = \mathcal{P}^{d_0}(T) + \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^{d_0}(T_v) = \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^{d_0}(T_v) + O(n)$$

and

$$\mathcal{P}^{IR}(T) = \mathcal{P}^{d_1}(T) + \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^{d_1}(T_v) = \sum_{v \in \overline{\mathcal{V}}(T)} \mathcal{P}^{d_1}(T_v) + O(n).$$

We see this as follows. The total leaf to root distance is the sum over all edges of the number of path that use that edge which is the number of leaves in the subtree below the edge, which is T_v, where v is the vertex below the edge. We, however, notice that by our definition of a leaf, if v is a leaf in T, it will not be counted as a leaf in T_v. Thus, overall, we under count by the number of leaves in T. A similar argument holds for the total internal node to root distance.

In subsection 3.3.2, we will show that $\frac{\mathcal{P}^{LR}(\mathfrak{T}_n)}{\sqrt{n^3}}$ and $\frac{\mathcal{P}^{IR}(\mathfrak{T}_n)}{\sqrt{n^3}}$ both converge weakly to an Airy random variable.

3.2.3 Generating Functions and Analytic Combinatorics

For a sequence $(a(n))$ for $n \in \mathbb{Z}_{\geq 0}$, we define the *generating function* corresponding to the sequence to be the formal power series,

$$F(x) = \sum_{n \geq 0} a(n) x^n.$$

Similarly, for a k-dimensional sequence, $(a(n_1, \cdots, n_k))$ for $n_1, \cdots, n_k \in \mathbb{Z}_{\geq 0}$, we define the *multivariate generating function* corresponding to the sequence to be

$$F(x_1, \cdots, x_k) = \sum_{n_1 \geq 0} \cdots \sum_{n_k \geq 0} a(n_1, \cdots, n_k) x_1^{n_1} \cdots x_k^{n_k}.$$

We will use $[x_1^{n_1} \cdots x_k^{n_k}]\, F(x_1, \cdots, x_k)$ to denote the coefficient of $x_1^{n_1} \cdots x_k^{n_k}$ in the generating function $F(x_1, \cdots, x_k)$, namely $a(n_1, \cdots, n_k)$.

Fix $(a(n))$ for $n \in \mathbb{Z}_{\geq 0}$ and let $F(z)$ be the associated generating function. We can treat $F(z)$ as a function over the complex plane. We say $F(z)$ is *analytic* at a point $z_0 \in \mathbb{C}$ if there exist a region around z_0 such that $F(z)$ is differentiable. We say $F(z)$ is analytic on a domain if it is analytic at all points in the domain.

A point, $z_0 \in \mathbb{C}$, is a *singularity* of $F(z)$ if $F(z)$ is not analytic at z_0. Furthermore, that singularity is *isolated* if there exist $\epsilon > 0$ such that $F(z)$ is analytic on the domain $\{z \in \mathbb{C} : 0 < |z - z_0| < \epsilon\}$. Let $F(z)$ be such that all its singularities are isolated. We define a *dominant singularity* of $F(z)$ to be an isolated singularity with minimal distance from the origin.

From the work of Flajolet and Sedgewick [34], we now state a slightly simplified form of a so called Transfer Theorem that will allow us to deduce asymptotic information about $(a(n))$ using $F(z)$. For some $R > 1$ and $0 < \phi < \frac{\pi}{2}$, we define a Δ-domain at 1 to be the domain

$$\Delta(\phi, R) = \{z \in \mathbb{C} : |z| < R, z \neq 1, |\arg(z - 1)| > \phi\}.$$

Theorem 9 (Transfer Theorem). *Let $(a(n))$ be a sequence with associated generating functions $F(z)$. Let $F(z)$ be a function analytic at 0 with a unique dominant singularity at 1 and let $F(z)$ be analytic in a Δ-domain at 1, Δ_0. Assume there exist σ, τ such that σ is a finite linear combination of terms of the form $(1 - z)^{-\alpha}$ for $\alpha \in \mathbb{C}$ and τ is a term of the* *form* $(1-z)^{-\beta}$ *for* $\beta \in \mathbb{C}$ *such that, for* $z \in \Delta_0$*,*

$$F(z) = \sigma(z) + O(\tau(z)) \qquad \textit{as} \qquad z \to 1.$$

Then, the following asymptotic estimation holds.

$$a(n) = [z^n]F(z) = [z^n]\sigma(z) + O(n^{\beta-1}).$$

A basic application of the above result that we will be useful in section 3.3 is as follows.

Corollary 10. *Let* $(a(n))$ *be a sequence with associated generating functions* $F(z)$*. Let* $\frac{1}{\zeta} \in \mathbb{C}$ *be the unique dominant singularity of* $F(z)$ *and assume* $F(z) = (1-\zeta z)^{-\alpha} g(z)$ *where* $\alpha \in \mathbb{C}$ *and* $g(z)$ *is a complex-valued function that is analytic in the region* $R = \left\{ z \in \mathbb{C} : |z| \leq \left|\frac{1}{\zeta}\right| \right\}$*. Then, the following asymptotic estimation holds.*

$$a(n) = \frac{\zeta^n g\left(\frac{1}{\zeta}\right)}{\Gamma(\alpha)} n^{\alpha-1} + O\left(\zeta^n n^{\alpha-2}\right).$$

Proof. We expand $g(z)$ about $z = \frac{1}{\zeta}$ using Taylor's Theorem to get $g(z) = g\left(\frac{1}{\zeta}\right) + O(1-\zeta z)$. Thus

$$F\left(\frac{z}{\zeta}\right) = g\left(\frac{1}{\zeta}\right)(1-z)^{-\alpha} + O\left((1-z)^{1-\alpha}\right) \qquad \text{as} \qquad z \to 1.$$

We then apply Theorem 9 and notice that $[z^n]F\left(\frac{z}{\zeta}\right) = \frac{1}{\zeta^n}[z^n]F(z)$ to achieve the desired result.

□

3.3 Results

3.3.1 Generating functions for counting trees by leaves, internal nodes and root degree

Let $\mathcal{G}_n(d_0, d_1, r)$ be the set of plane trees on n edges with d_0 leaves, d_1 internal nodes and root degree r. Let

$$G(x,a,b) = \sum_{n=0}^{\infty}\sum_{d_0=0}^{\infty}\sum_{d_1=0}^{\infty}\sum_{r=0}^{\infty} |\mathcal{G}_n(d_0,d_1,r)| x^n a^{d_0} b^{d_1},$$

$$G_r(x,a,b) = \sum_{n=0}^{\infty}\sum_{d_0=0}^{\infty}\sum_{d_1=0}^{\infty} |\mathcal{G}_n(d_0,d_1,r)| x^n a^{d_0} b^{d_1}$$

and

$$G(x,a,b,c) = \sum_{n=0}^{\infty}\sum_{d_0=0}^{\infty}\sum_{d_1=0}^{\infty}\sum_{r=0}^{\infty} |\mathcal{G}_n(d_0,d_1,r)| x^n a^{d_0} b^{d_1} c^r.$$

As we will use it often from this point on, let $G^*(x,a,b) = G_1(x,a,b)$.

Theorem 11. *The following recurrences hold.*

$$G(x,a,b) = 1 + xG(x,a,b)^2 + (a-1)xG(x,a,b) + (b-1)xG(x,a,b)G^*(x,a,b), \quad (3.4)$$

$$G(x,a,b,c) = \frac{1}{1 - cG^*(x,a,b)}, \quad (3.5)$$

and

$$G^*(x,a,b) = 1 - \frac{1}{G(x,a,b)}. \quad (3.6)$$

Proof. To achieve Equation 3.4, we apply the decomposition of a plane tree into 2 subtrees given by $T = T_1 \ltimes T_2$. This adds a new edge. The only tree not accounted for by this process is T^*. Note that this process of adjoining trees never creates any new leaves or internal nodes in T_1. Notice that when $T_2 = T^*$, T_2 has an extra leaf which is not accounted for (since it would be at the root of T_2). Thus we re-weight these trees. (Hence $(a -$

$1)xG(x,a,b)$.) Also notice that when T_2 has root degree 1, T_2 has an extra internal node which is not accounted for (since it would be at the root of T_2). Thus we re-weight these trees. (Hence $(b-1)xG(x,a,b)G^*(x,a,b)$.)

To achieve Equation 3.5, we notice that a tree with root degree r is equivalent to a sequence of r trees with root degree 1 where we identify all the root vertices. This identification does not add or remove any edges, leaves or internal nodes. Thus the generating function for trees with root degree r, where we weight root degree, is $c^r G^*(x,a,b)^r$. Summing over all possible values for r, we get the desired expression.

To achieve Equation 3.6, we set c to 1 in (Equation 3.5) to ignore root degree, and rearrange the expression.

□

Corollary 12. *The following generating functions hold.*

$$G(x,a,b) = \frac{1+(2-a-b)x-\sqrt{(1+(2-a-b)x)^2-4x(1-(b-1)x)}}{2x} \tag{3.7}$$

and

$$G^*(x,a,b) = \frac{1+(a-b)x-\sqrt{(1+(2-a-b)x)^2-4x(1-(b-1)x)}}{2(1-(b-1)x)}. \tag{3.8}$$

Proof. The result follows immediately by solving Equation 3.4 and Equation 3.6 simultaneously.

□

Lemma 13. *For all $a,b \in \mathbb{R}_{>0}$, the dominant singularity of $G(x,a,b)$ and $G^*(x,a,b)$ occurs at $\rho = \rho(a,b) = a+b+2\sqrt{a}$.*

Proof. Fix $a,b > 0$. Let $\Psi = (1+(2-a-b)x)^2 - 4x(1-(b-1)x)$ and let $\overline{\rho} = \overline{\rho}(a,b) = a+b-2\sqrt{a}$. The roots of Ψ are $\frac{1}{\rho}$ and $\frac{1}{\overline{\rho}}$. Clearly, $0 < \frac{1}{\rho} < \frac{1}{\overline{\rho}}$. Notice that the only singularity caused by the numerator of (Equation 3.8) and (Equation 3.7) occurs when

$\Psi = 0$. Thus the most significant of such singularities is $\frac{1}{\rho}$.

Notice that if there exists another singularity of $G(x, a, b)$, it must occur at $x = 0$ (caused by the denominator). We notice that when $x = 0$, the numerator of Equation 3.7 goes to 0. Thus, taking a Laurent expansion of $G(z, a, b)$ at $z = 0$, we get

$$G(z, a, b) = \frac{1}{2z} \sum_{n \geq 1} d_n z^n = \sum_{n \geq 0} \frac{d_{n+1}}{2} z^n,$$

where d_i are constants. Thus $G(z, a, b)$ is analytic at $z = 0$. Thus, the dominant singularity of $G(x, a, b)$ is $\frac{1}{\rho}$.

For $b = 1$, the denominator of Equation 3.8 is a constant, thus cannot cause another singularity. Assume $b \neq 1$. Notice that if there exists another singularity of $G^*(x, a, b)$, it must occur at $x = \frac{1}{b-1}$ (caused by the denominator). We notice that when $x = \frac{1}{b-1}$, the numerator of Equation 3.8 goes to $d'_0 = \frac{a-1}{b-1} - \left|\frac{a-1}{b-1}\right|$. Notice that when the sign of $a - 1$ and $b - 1$ are the same (or $a = 1$), $d'_0 = 0$. Thus, taking a Laurent expansion of $G^*(z, a, b)$ at $z = \frac{1}{b-1}$, we get

$$G^*(z, a, b) = \frac{1}{2(1 - (b-1)z)} \sum_{n \geq 0} d'_n \left(z - \frac{1}{b-1}\right)^n = \sum_{n \geq -1} \frac{d'_{n+1}}{2(1-b)} \left(z - \frac{1}{b-1}\right)^n,$$

where d'_i are constants. Notice that when $d'_0 = 0$, $G^*(z, a, b)$ is analytic at $z = \frac{1}{b-1}$. Thus, the dominant singularity of $G^*(x, a, b)$ is $\frac{1}{\rho}$. When $d'_0 \neq 0$, $G^*(z, a, b)$ has a singularity at $z = \frac{1}{b-1}$. For this to be the case, we must have that $a - 1$ and $b - 1$ have the different signs. Notice that for $z = \frac{1}{b-1}$ to be the dominant singularity, $\rho^2 < (b-1)^2$, which implies that $a + 2b + 2\sqrt{a} < 1$. Since $a - 1$ and $b - 1$ have different signs, one of a and b is at least 1, thus the inequality cannot hold. We thus conclude that the dominant singularity of $G^*(x, a, b)$ in this case is also $\frac{1}{\rho}$.

□

Corollary 14. *Fix $\alpha, \beta \in \mathbb{R}$. Let $\rho = e^{-\alpha} + e^{-\beta} + 2e^{-\frac{\alpha}{2}}$. The following estimate holds.*

$$\mathcal{Z}_{(n,\alpha,\beta,0)} = \frac{\sqrt{e^{-\frac{\alpha}{2}}\rho}}{2\sqrt{\pi}} \cdot \rho^n \cdot n^{-\frac{3}{2}} + O\left(n^{-\frac{5}{2}}\right)$$

Proof. Let $\overline{\rho} = e^{-\alpha} + e^{-\beta} - 2e^{-\frac{\alpha}{2}}$. By definition, it should be clear that $\mathcal{Z}_{(n,\alpha,\beta,0)} = [x^n]G(x, e^{-\alpha}, e^{-\beta})$. Thus, from Corollary 12, for $n \geq 2$,

$$\mathcal{Z}_{(n-1,\alpha,\beta,0)} = [x^n]\left(-\frac{\sqrt{1-\overline{\rho}x}}{2} \cdot \sqrt{1-\rho x}\right).$$

By Lemma 13, for all $\alpha, \beta \in \mathbb{R}$, $\frac{1}{\rho} < \frac{1}{\overline{\rho}}$, thus $\sqrt{1-\overline{\rho}x}$ is analytic on the disk $R = \left\{z \in \mathbb{C} : |z| \leq \frac{1}{\rho}\right\}$. We thus apply Corollary 10, to see that

$$\begin{aligned}
\mathcal{Z}_{(n-1,\alpha,\beta,0)} &= -\frac{\rho^n}{2\Gamma\left(-\frac{1}{2}\right)} \cdot \left(\sqrt{\frac{4e^{-\frac{\alpha}{2}}}{\rho}}\right) n^{-\frac{3}{2}} + O\left(n^{-\frac{5}{2}}\right) \\
&= \frac{\rho^n}{2\sqrt{\pi}} \cdot \sqrt{\frac{e^{-\frac{\alpha}{2}}}{\rho}} \cdot n^{-\frac{3}{2}} + O\left(n^{-\frac{5}{2}}\right). \qquad (3.9)
\end{aligned}$$

We now set n to $n+1$ to get the desired result.

□

We will now extract the generating function $G_n(a, b)$ defined by

$$G_n(a,b) = \sum_{d_0=0}^{\infty}\sum_{d_1=0}^{\infty}\sum_{r=0}^{\infty} |\mathcal{G}_n(d_0, d_1, r)| a^{d_0} b^{d_1}$$

using the following technical lemma. We will defer the proof of the lemma.

For a continuously differentiable function $F(x_1, \cdots, x_k)$ and V, a finite multiset with elements $v_1, \cdots, v_m \in \{x_1, \cdots, x_k\}$, define

$$\frac{\partial F(x_1, \cdots, x_k)}{\partial V} = \frac{\partial^m F(x_1, \cdots, x_k)}{\partial v_1 \cdots \partial v_m}.$$

For a set V, let $\text{Part}(V)$ be the set of unordered partitions on V, (V_i). For a multiset, V, we define $\text{Part}(V)$ by distinguishing all elements of V, taking the partitions of the induced set, then removing the distinction from each of the repeated elements. Note that $\text{Part}(V)$ is itself a multiset. For example, $\text{Part}(\{x_1, x_1\}) = \{\{\{x_1\}, \{x_1\}\}, \{\{x_1, x_1\}\}\}$.

Lemma 15. *Let F be a continuously differentiable function in x, Δ be a continuously differentiable function in $x_1, \cdots, x_k$ and V be a non-empty finite set or multiset of the elements $x_1, \cdots, x_k$ with elements $v_1, \cdots, v_m$.*

$$\frac{\partial F(\Delta)}{\partial V} = \sum_{(V_i) \in Part(V)} \frac{\partial \Delta}{\partial V_1} \cdots \frac{\partial \Delta}{\partial V_p} \cdot \left. \frac{\partial^p F(x)}{\partial x^p} \right|_{x=\Delta} \tag{3.10}$$

We now achieve the following expression for $G_n(a, b)$. We also compute the coefficients of the above expression explicitly via a combinatorial argument in Corollary 54.

Corollary 16. *For $n \geq 2$,*

$$G_n(a, b) = \frac{1}{2^{n+1}} \sum_{0 \leq k \leq \frac{n+1}{2}} C_{n-k} \binom{n-k+1}{k} \cdot (a+b)^{n-2k+1} \cdot \left(4a - (a+b)^2\right)^k.$$

Proof. We consider $xG(x, a, b)$. Notice that for $n \geq 2$,

$$\frac{\partial^n xG(x, a, b)}{\partial x^n} = -\frac{1}{2} \cdot \frac{\partial \sqrt{\Delta}}{\partial V}$$

where V is the multiset containing n copies of x and $\Delta = (1 + (2 - a - b)x)^2 - 4x(1 - (b-1)x)$. Thus we apply Lemma 15. We notice that if $|V_i| > 2$, $\frac{\partial \Delta}{\partial V_i} = 0$. The number of elements in $\text{Part}(V)$ in which each part has size is at most 2 and there are l parts of size 1 and k parts of size 2 is $\frac{n!}{2^k \cdot k! \cdot l!}$. Thus, for $n \geq 2$,

$$\left.\frac{\partial^n xG(x,a,b)}{\partial x^n}\right|_{x=0} = \frac{1}{2}\sum_{2k+l=n} \frac{n!}{2^k \cdot k! \cdot l!} \cdot \left(\frac{\partial \Delta}{\partial x}\right)^l \cdot \left(\frac{\partial^2 \Delta}{\partial x^2}\right)^k \tag{3.11}$$

$$\cdot \left.\frac{\left(-\frac{1}{2}\right)^{k+l} \cdot (2(k+l)-3)!!}{\Delta^{\frac{2(k+l)-1}{2}}}\right|_{x=0} \tag{3.12}$$

$$= \frac{n!}{2^n} \sum_{0\le k\le \frac{n}{2}} C_{n-k-1}\binom{n-k}{k} \cdot (a+b)^{n-2k} \cdot \left(4a-(a+b)^2\right)^k. \tag{3.13}$$

Finally, notice that

$$\left.\frac{\partial^n xG(x,a,b)}{\partial x^n}\right|_{x=0} \quad = \quad n! \sum_{d_0=0}^{\infty}\sum_{d_1=0}^{\infty}\sum_{r=0}^{\infty} |\mathcal{G}_{n-1}(d_0,d_1,r)| a^{d_0} b^{d_1}. \tag{3.14}$$

□

3.3.2 The Distributions of Simple Subtree Additive Properties

In this section, we will consider various additive properties. We will assume all the tolls in this section are of the form $(f,0)$ since, from Example 2 and Equation 3.2, we see that for any $c \in \mathbb{Z}_{\geq 0}$ and $T \in \mathfrak{T}_n$,

$$\mathcal{P}^{(f,c)}(T) = \mathcal{P}^{(f,0)}(T) + c \cdot (\mathcal{P}^e(T)+1) = \mathcal{P}^{(f+c,0)}(T) + c, \tag{3.15}$$

where $(f+c)(T) = f(T)+c$.

Let $\mathcal{P}$ be a non-negative integer valued additive property of plane trees. Let $\mathcal{P}^*$ be the subtree additive property induced by $\mathcal{P}$. Such a subtree additive property is simple as we have defined. The main result of this section is that if the toll function of $\mathcal{P}$ is bounded, the limiting distribution of $\mathcal{P}^*$ is determined by the limiting distribution of $\mathcal{P}$. Our primary result is stated as follows.

Theorem 17. *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be non-negative integer valued additive properties of plane trees with toll functions f_1 and f_2, respectively. Let the subtree additive properties induced by $\mathcal{P}_1$ and $\mathcal{P}_2$ be $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$, respectively. Further assume that there exists $\zeta \in \mathbb{N}$ such that for all $T \in \mathfrak{T}_{\geq 0}$, $f_1(T) \leq \zeta$ and $f_2(T) \leq \zeta$. If, for all $m, n \in \mathbb{Z}$,*

$$\sum_{T \in \mathfrak{T}_n} \mathcal{P}_1(T)^m = \mu^m \cdot \sum_{T \in \mathfrak{T}_n} \mathcal{P}_2(T)^m + O\left(n^{\frac{2m-4}{2}} 4^n\right) \tag{3.16}$$

where $\mu \in \mathbb{R}$ is a constant, then as $n \to \infty$,

$$\frac{\mathcal{P}_1^*(\mathfrak{T}_n)}{\sqrt{n^3}} \overset{d}{\longleftrightarrow} \mu \cdot \frac{\mathcal{P}_2^*(\mathfrak{T}_n)}{\sqrt{n^3}}.$$

Let $\mathcal{P}$ be a non-negative integer valued property of plane trees that is additive with toll function f. Let $\mathcal{F}(n, m)$ be the set of trees, T, on n edges such that $\mathcal{P}(T) = m$. Let

$$F(x, p) = \sum_{n \geq 0} \sum_{m \geq 0} |\mathcal{F}(n, m)| x^n p^m.$$

We may also refer to $F(x, p)$ by $F_{\mathcal{P}}(x, p)$ where the property we are referring to is unclear. We now let $\mathcal{H}^v(n, m)$ be the set of trees, T, on n edges such that $\mathcal{P}(T) = m$ and $f(T) = v$. Let

$$H^v(x, p) = \sum_{n \geq 0} \sum_{m \geq 0} |\mathcal{H}^v(n, m)| x^n p^m.$$

Lemma 18. *For any fixed non-negative integer valued additive property of plane trees, the following recurrence holds.*

$$F(x, p) = 1 + xF(x, p) \sum_{v \geq 0} p^v H^v(x, p) \tag{3.17}$$

Proof. We apply the decomposition of a plane tree into 2 subtrees given by $T = T_1 \ltimes T_2$. This adds a new edge. Recall that $\mathcal{P}(T) = \mathcal{P}(T_1) + \mathcal{P}(T_2) + f(T_2)$. Thus, the tree T gains an extra $p^{f(T_2)}$ in the weighting. When $T_2 \in \mathcal{H}^v(n, m)$, T gains an extra p^v in the weighting. The only tree not accounted for by this process is T^*.

□

Lemma 19. *Let $\mathcal{P}$ be a non-zero non-negative integer valued additive property with toll function f such that $f(T) \leq \zeta$ for all $T \in \mathfrak{T}_{\geq 0}$ where f achieve ζ. For all $n, m \geq 0$,*

$$\sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)^m = \Theta\left(n^{\frac{2m-3}{2}} 4^n\right).$$

Proof. Fix $m \geq 0$. Assume $\zeta > 0$ (otherwise $\mathcal{P}(T) = 0$ for all $T \in \mathfrak{T}_{\geq 0}$). We first note that

$$\begin{aligned} \frac{\partial F_{\mathcal{P}}(x,1)}{\partial p^m} &= \sum_{n \geq 0} x^n \sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)(\mathcal{P}(T)-1)\cdots(\mathcal{P}(T)-m+1) \\ &= \sum_{n \geq 0} x^n \sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)^m + O\left(\mathcal{P}(T)^{m-1}\right) \end{aligned} \tag{3.18}$$

Consider $\mathcal{P}_1$, the additive property with toll function $f_1(T) = \zeta$ for all $T \in \mathfrak{T}_{\geq 0}$. Let $T' \in \mathfrak{T}_N$ be a tree such that $f(T') = \zeta$. Consider $\mathcal{P}_2$, the additive property with toll function $f_2(T') = \zeta$ and $f_2(T) = 0$ for all other $T \in \mathfrak{T}_{\geq 0}$. It should be clear that for any tree $T \in \mathfrak{T}_{\geq 0}$,

$$0 \leq \mathcal{P}_2(T) \leq \mathcal{P}(T) \leq \mathcal{P}_1(T). \tag{3.19}$$

We now see that $F_{\mathcal{P}_1}(x,p) = 1 + xp^\zeta F_{\mathcal{P}_1}(x,p)^2$ and $F_{\mathcal{P}_2}(x,p) = 1 + xF_{\mathcal{P}_2}(x,p)^2 + (p^\zeta - 1)x^N$, thus

$$F_{\mathcal{P}_1}(x,p) = \frac{1 - \sqrt{1 - 4xp^\zeta}}{2xp^\zeta} \qquad \text{and} \qquad F_{\mathcal{P}_2}(x,p) = \frac{1 - \sqrt{1 - 4x(1 + (p^\zeta - 1)x^N)}}{2x}.$$

Thus, for $m \geq 1$,

$$\frac{\partial F_{\mathcal{P}_1}(x,1)}{\partial p^m} = c_1 \cdot (1-4x)^{\frac{1-2m}{2}} + O\left((1-4x)^{\frac{3-2m}{2}}\right)$$

and

$$\frac{\partial F_{\mathcal{P}_2}(x,1)}{\partial p^m} = c_2 \cdot (1-4x)^{\frac{1-2m}{2}} + O\left((1-4x)^{\frac{3-2m}{2}}\right), \tag{3.20}$$

where c_1, c_2 are constants (that depend on m). We now apply Corollary 10 to see that

$$[x^n]\frac{\partial F_{\mathcal{P}_1}(x,1)}{\partial p^m} = c_1' \cdot n^{\frac{2m-3}{2}} 4^n (1+o(1)) \quad \text{and} \quad [x^n]\frac{\partial F_{\mathcal{P}_2}(x,1)}{\partial p^m} = c_2' \cdot n^{\frac{2m-3}{2}} 4^n (1+o(1)) \tag{3.21}$$

where c_1', c_2' are constants. From Equation 3.18 and Equation 3.19, we thus see that for all $n \geq 0$,

$$\sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)^m \leq \sum_{T \in \mathfrak{T}_n} \mathcal{P}_1(T)^m = c_1' \cdot n^{\frac{2m-3}{2}} 4^n (1+o(1)) \tag{3.22}$$

and

$$\sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)^m \geq \sum_{T \in \mathfrak{T}_n} \mathcal{P}_2(T)^m = c_2' \cdot n^{\frac{2m-3}{2}} 4^n (1+o(1)). \tag{3.23}$$

□

Lemma 20. *Let $\mathcal{P}$ be a non-zero non-negative integer valued additive property with toll function f such that $f(T) \leq \zeta$ for all $T \in \mathfrak{T}_{\geq 0}$ where f achieve ζ. As $n \to \infty$, the limiting distribution of $\frac{\mathcal{P}(\mathfrak{T}_n)}{n}$ is uniquely determined by its moments.*

Proof. Consider $\mathcal{P}_1$, the additive property with toll function f such that $f(T) = \zeta$ for all $T \in \mathfrak{T}_{\geq 0}$. We also see that for all $T \in \mathfrak{T}_{\geq 0}$, $\mathcal{P}(T) \leq \mathcal{P}_1(T)$. From Example 2 and (Equation 3.2), for $T \in \mathfrak{T}_n$, $\mathcal{P}_1(T) = \zeta n$. Thus $\mathbb{E}\left[\left(\frac{\mathcal{P}(\mathfrak{T}_n)}{n}\right)^k\right] \leq \mathbb{E}\left[\left(\frac{\mathcal{P}_1(\mathfrak{T}_n)}{n}\right)^k\right] = \zeta^k$. The result follows immediately by the Carleman's condition (Theorem 1).

□

The above result tells us that the condition in Equation 3.16 implies that $\frac{\mathcal{P}_1(\mathfrak{T}_n)}{n} \overset{d}{\longleftrightarrow} \mu \cdot \frac{\mathcal{P}_2(\mathfrak{T}_n)}{n}$. Thus Theorem 17 is equivalent to the following corollary.

Corollary 21. *For non-negative integer valued additive properties, $\mathcal{P}_1$ and $\mathcal{P}_2$, with toll functions f_1 and f_2, respectively, such that there exists $\zeta \in \mathbb{N}$ such that for any $T \in \mathfrak{T}_{\geq 0}$,*

$f_1(T) \leq \zeta$ and $f_2(T) \leq \zeta$,

$$\frac{\mathcal{P}_1(\mathfrak{T}_n)}{n} \overset{d}{\longleftrightarrow} \mu \cdot \frac{\mathcal{P}_2(\mathfrak{T}_n)}{n} \Rightarrow \frac{\mathcal{P}_1^*(\mathfrak{T}_n)}{\sqrt{n^3}} \overset{d}{\longleftrightarrow} \mu \cdot \frac{\mathcal{P}_2^*(\mathfrak{T}_n)}{\sqrt{n^3}}$$

where $\mu \in \mathbb{R}$ is constant and $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$ are the induced subtree additive properties.

Let $\mathcal{P}^*$ be the subtree additive property induced by $\mathcal{P}$. Let $\mathfrak{T}_{n,m}$ be the set of plane trees, T, on n edges such that $\mathcal{P}(T) = m$. We define

$$\begin{aligned} M_{k,n,m} &= \sum_{T\in\mathfrak{T}_{n,m}} \mathcal{P}^*(T)^k = \sum_{T\in\mathfrak{T}_{n,m}} \left(\sum_{v\in\overline{\mathcal{V}}(T)} \mathcal{P}(T_v) \right)^k \\ &= \sum_{T\in\mathfrak{T}_{n,m}} \sum_{(v_1,\cdots,v_k)\in\overline{\mathcal{V}}(T)^k} \mathcal{P}(T_{v_1})\cdots\mathcal{P}(T_{v_k}) \end{aligned} \tag{3.24}$$

and let

$$M_k(x,p) = \sum_{n,m\geq 0} M_{k,n,m} x^n p^m \tag{3.25}$$

and $M_k(x) = M_k(x,1)$. Note that $[x^n]M_k(x) = \sum_{T\in\mathfrak{T}_n} \mathcal{P}^*(T)^k = \mathbb{E}\left[\mathcal{P}^*(\mathfrak{T}_n)^k\right] \cdot C_n$, where C_n is the nth Catalan number.

Fix $t_1, \cdots, t_k \in \mathbb{Z}_{\geq 0}$ and $n, m \geq 0$. We consider tuples of $(v_1, \cdots, v_k) \in \overline{\mathcal{V}}(T)$ for some $T \in \mathfrak{T}_{n,m}$ where $\mathcal{P}(T_{v_i}) = t_i$. Notice that the contribution to $M_{k,n,m}$ of any such tuples is $\prod_{i=1}^k t_i$. Also notice that the number of such tuples is

$$\sum_{T\in\mathfrak{T}_{n,m}} \prod_{i=1}^k W(T, t_i)$$

where $W(T,t) = \left|\left\{v \in \overline{\mathcal{V}}(T) : \mathcal{P}(T_v) = t\right\}\right|$. Let $\mathcal{F}_k(n, m, \vec{t}, \vec{s})$ be the set of trees, T, on n edges such that $\mathcal{P}(T) = m$ and $W(T, t_i) = s_i$ where $\vec{s} = (s_1, \cdots, s_k) \in \mathbb{Z}_{\geq 0}^k$ and

$\vec{t} = (t_1, \cdots, t_k) \in \mathbb{Z}^k_{\geq 0}$. We now let

$$F_k(\vec{t}|x, \vec{y}|m) = \sum_{n\geq 0}\sum_{s_1\geq 0}\cdots\sum_{s_k\geq 0} |\mathcal{F}_k(n, m, \vec{t}, \vec{s})| x^n \prod_{i=1}^{k} y_i^{s_i}$$

and

$$F_k(\vec{t}|x, p, \vec{y}) = \sum_{m\geq 0} F_k(\vec{t}|x, \vec{y}|m) p^m.$$

Note that

$$\frac{\partial F_k(\vec{t}|x, p, \vec{1})}{\partial y_1 \cdots \partial y_k} = \sum_{n\geq 0}\sum_{m\geq 0}\sum_{s_1\geq 0}\cdots\sum_{s_k\geq 0} |\mathcal{F}_k(n, m, \vec{t}, \vec{s})| x^n p^m \prod_{i=1}^{k} s_i$$

where $\vec{1}$ the vector of appropriate size for the context consisting of all 1s.

For $V = \{v_1, \cdots, v_l\} \subset \mathbb{N}$ and F, a function, we define

$$\frac{\partial F}{\partial y(V)} = \frac{\partial F}{\partial y_{v_1} \cdots \partial y_{v_l}}.$$

We define $\frac{\partial F}{\partial z(V)}$ similarly. We also denote the set of integers from 1 to k by $[k]$. We now show the following lemma that will be integral to the rest of our analysis.

Lemma 22. *For $k \in \mathbb{N}$ and $V = \{v_1, \cdots, v_l\} \subset [k]$ such that $|V| = l \leq k$, the following holds.*

$$M_l(x, p) = \frac{\partial}{\partial z(V)} \left(\sum_{t_{v_1}\geq 0} \cdots \sum_{t_{v_l}\geq 0} \frac{\partial F_k(\vec{t}|x, p, \vec{1})}{\partial y(V)} \prod_{i=1}^{l} z_{v_i}^{t_{v_i}} \right)\Bigg|_{\vec{z}=\vec{1}}. \tag{3.26}$$

Proof. Let $W = [k] - V = \{w_1, \cdots, w_{k-l}\}$. We first see that

$$\begin{aligned}\frac{\partial F_k(\vec{t}|x,p,\vec{1})}{\partial y(V)} &= \sum_{n\ge0}\sum_{m\ge0}\sum_{s_1\ge0}\cdots\sum_{s_k\ge0} |\mathcal{F}_k(n,m,\vec{t},\vec{s})| x^n p^m \prod_{i=1}^{l} s_{v_i} \\ &= \sum_{n\ge0}\sum_{m\ge0}\sum_{s_{v_1}\ge0}\cdots\sum_{s_{v_l}\ge0} x^n p^m \prod_{i=1}^{l} s_{v_i} \sum_{s_{w_1}\ge0}\cdots\sum_{s_{w_{k-l}}\ge0} |\mathcal{F}_k(n,m,\vec{t},\vec{s})| \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.27) \\ &= \sum_{n\ge0}\sum_{m\ge0}\sum_{s_{v_1}\ge0}\cdots\sum_{s_{v_l}\ge0} |\mathcal{F}_l(n,m,\vec{t},\vec{s})| x^n p^m \prod_{i=1}^{l} s_{v_i} && (3.28) \\ &= \frac{\partial F_l(\vec{t}|x,p,\vec{1})}{\partial y(V)} && (3.29)\end{aligned}$$

We go from Equation 3.27 to Equation 3.28 by noting that when we sum $|\mathcal{F}_k(n,m,\vec{t},\vec{s})|$ over $s_{w_1}\cdots s_{w_{k-l}}$, we lose our dependency on $t_{w_1},\cdots,t_{w_{k-l}}$. We go from Equation 3.28 to Equation 3.29 by relabeling the indices from $v_1,\cdots,v_l$ to $1,\cdots,l$. Thus to prove the lemma, we need only consider the case when $V = [k]$. We now see that

$$\begin{aligned}&\frac{\partial}{\partial z([k])}\left(\sum_{t_{v_1}\ge0}\cdots\sum_{t_{v_l}\ge]0}\frac{F_k(\vec{t}|x,p,\vec{1})}{\partial y([k])}\prod_{i=1}^{l} z_{v_i}^{t_{v_i}}\right)\Bigg|_{\vec{z}=\vec{1}} \\ &\qquad = \sum_{n\ge0}\sum_{m\ge0}\sum_{t_1\ge0}\sum_{s_1\ge0}\cdots\sum_{t_k\ge0}\sum_{s_k\ge0} |\mathcal{F}_k(n,m,\vec{t},\vec{s})| x^n p^m \prod_{i=1}^{k} s_i t_i. \qquad (3.30)\end{aligned}$$

For fixed $\vec{t},\vec{s}$, we notice that trees in $\mathcal{F}_k(n,m,\vec{t},\vec{s})$ are precisely the trees from which we can get $(v_1,\cdots,v_k) \in \overline{\mathcal{V}}(T)$ for some $T \in \mathfrak{T}_{n,m}$ where $\mathcal{P}(T_{v_i}) = t_i$. The contribution of each tuple to $M_{k,n,m}$ is $\prod_{i=1}^{k} t_i$. The number of tuples $(v_1,\cdots,v_k)$ that can be achieved from each tree $T \in \mathcal{F}_k(n,m,\vec{t},\vec{s})$ is $\prod_{i=1}^{k} s_i$. Thus the total (weighted) contribution from tuples of the above form is

$$|\mathcal{F}_k(n,m,\vec{t},\vec{s})| x^n p^m \prod_{i=1}^{k} s_i t_i.$$

Hence, summing over $\vec{t}, \vec{s} \in \mathbb{Z}^k_{\geq 0}$ and $n, m \in \mathbb{Z}_{\geq 0}$, we get $M_k(x,p)$, proving the result.

□

Let $\mathcal{H}^v_k(n,m,\vec{t},\vec{s}) = \mathcal{H}^v(n,m) \cap \mathcal{F}_k(n,m,\vec{t},\vec{s})$,

$$H^v_k(\vec{t}|x,\vec{y}|m) = \sum_{n\geq 0}\sum_{s_1\geq 0}\cdots\sum_{s_k\geq 0} |\mathcal{H}^v_k(n,m,\vec{t},\vec{s})| x^n \prod_{i=1}^k y_i^{s_i}, \tag{3.31}$$

$$H^v_k(\vec{t}|x,p,\vec{y}) = \sum_{m\geq 0} H^v_k(\vec{t}|x,\vec{y}|m)p^m \tag{3.32}$$

and

$$J^v_k(x,p) = \frac{\partial}{\partial z([k])}\left(\sum_{t_1\geq 0}\cdots\sum_{t_k\geq 0}\frac{H^v_k(\vec{t}|x,p,\vec{1})}{\partial y([k])}\prod_{i=1}^k z_i^{t_i}\right)\Bigg|_{\vec{z}=\vec{1}}. \tag{3.33}$$

Using a similar argument to Lemma 22, we get that for $V = \{v_1, \cdots, v_l\} \subset [k]$ such that $|V| = l \leq k$,

$$J^v_l(x,p) = \frac{\partial}{\partial z(V)}\left(\sum_{t_{v_1}\geq 0}\cdots\sum_{t_{v_l}\geq 0}\frac{H^v_k(\vec{t}|x,p,\vec{1})}{\partial y(V)}\prod_{i=1}^l z_{v_i}^{t_{v_i}}\right)\Bigg|_{\vec{z}=\vec{1}}. \tag{3.34}$$

Note that

$$\sum_{v\geq 0} H^v_k(\vec{t}|x,p,\vec{y}) = F_k(\vec{t}|x,p,\vec{y}) \qquad \text{and} \qquad \sum_{v\geq 0} J^v_k(x,p) = M_k(x,p). \tag{3.35}$$

We define a partition of a set to be a set of disjoint subsets of the original set whose union is the original set. We call these subsets parts. We denote a partition of S into λ parts by $(S_i)_\lambda$, where $S_1, \cdots, S_\lambda$ are the parts of the partition. We say a partition of S, $(S_i)_\lambda$, refines another partition of S, $(S'_i)_\mu$ if for any S_i, there is a S'_j such that $S_i \subset S'_j$. We denote this by $(S_i)_\lambda \subset (S'_i)_\mu$.

Let $\vec{t} \in \mathbb{Z}^k_{\geq 0}$. Notice that $\vec{t}$ induces a partition of $S = [k]$ as follows. Let $(S_i)^{\vec{t}}_\lambda$ be such that the numbers i, j are in the same part if and only if $t_i = t_j$. For a fixed $(S_i)^{\vec{t}}_\lambda$, let $t^*_i = t_j$

such that $j \in S_i$.

Lemma 23. *The following recurrence holds.*

$$F_k(\vec{t}|x,p,\vec{y}) = 1 + x\sum_{v\geq 0} p^v F_k(\vec{t}|x,p,\vec{y}) H_k^v(\vec{t}|x,p,\vec{y}) \\ + xF_k(\vec{t}|x,p,\vec{y}) \sum_{v\geq 0}\sum_{i=1}^{\lambda}\left(\prod_{j\in S_i} y_j - 1\right)\cdot p^{t_i^*}\cdot p^v \cdot H_k^v(\vec{t}|x,\vec{y}|t_i^*), \tag{3.36}$$

where the S_i are the parts in $(S_i)_\lambda^{\vec{t}}$.

Proof. We apply the decomposition of a plane tree into 2 subtrees given by $T = T_1 \ltimes T_2$. By similar argument to Lemma 18, we properly weight with respect to n and m. To properly weight with respect to $\vec{y}$, we notice that the number of non-root vertices v with $\mathcal{P}(T_v) = t$, for some t, in T is the sum of the number of such vertices in T_1 and T_2. Additionally, when $\mathcal{P}(T_2) = t$, we get an extra such vertex (the root of T_2). When $\mathcal{P}(T_2) = t_i^*$, we get an extra vertex, v, where $\mathcal{P}(T_v) = t_j = t_i^*$, where $j \in S_i$. Notice that these are the trees counted by $\sum_{v\geq 0} H_k^v(\vec{t}|x,\vec{y}|t_i^*)$. These trees should get an extra $\prod_{j\in S_i} y_j$ in the weighting. We however notice that the weight with respect to p of such trees is $p^{t_i^*}$. Trees counted by $H_k^v(\vec{t}|x,\vec{y}|t_i^*)$ also get an extra p^v (to properly weight the entire tree with respect to p).

□

Differentiating both sides of (Equation 3.36), we see that

$$\frac{\partial F_k(\vec{t}|x,p,\vec{1})}{\partial y([k])} = x\sum_{V\subset[k]} \frac{\partial F_k(\vec{t}|x,p,\vec{1})}{\partial y([k]-V)} \sum_{v\geq 0} p^v \cdot \frac{\partial H_k^v(\vec{t}|x,p,\vec{1})}{\partial y(V)} + x\sum_{V\subset[k]} \frac{\partial F_k(\vec{t}|x,p,\vec{1})}{\partial y([k]-V)} \\ \times \frac{\partial}{\partial y(V)}\left(\sum_{v\geq 0}\sum_{i=1}^{\lambda}\left(\prod_{j\in S_i} y_j - 1\right)\cdot p^{t_i^*}\cdot p^v\cdot H_k^v(\vec{t}|x,\vec{y}|t_i^*)\right)\Bigg|_{\vec{y}=\vec{1}}. \tag{3.37}$$

For $V = \{v_1, \cdots, v_l\} \subset [k]$, we let $\vec{t}(V) = (t_{v_1}, \cdots, t_{v_l})$. Using Lemma 22, we simplify

$$\Phi_k^{(1)} = \frac{\partial}{\partial z([k])} \sum_{\vec{t} \in \mathbb{Z}_{\geq 0}^k} \left(\sum_{V \subset [k]} \frac{\partial F_k(\vec{t}|x, p, \vec{1})}{\partial y([k] - V)} \sum_{v \geq 0} p^v \cdot \frac{\partial H_k^v(\vec{t}|x, p, \vec{1})}{\partial y(V)} \right) \prod_{i \in [k]} z_i^{t_i}$$

as follows.

$$\begin{aligned} \Phi_k^{(1)} &= \sum_{V \subset [k]} \frac{\partial}{\partial z(V)} \left(\sum_{\vec{t}(V) \in \mathbb{Z}_{\geq 0}^k} \frac{\partial F_k(\vec{t}|x, p, \vec{1})}{\partial y(V)} \prod_{i \in V} z_i^{t_i} \right) \times \\ &\qquad \sum_{v \geq 0} p^v \cdot \frac{\partial}{\partial z([k] - V)} \left(\sum_{\vec{t}([k]-V) \in \mathbb{Z}_{\geq 0}^k} \frac{\partial H_k^v(\vec{t}|x, p, \vec{1})}{\partial y([k] - V)} \prod_{i \in [k] - V} z_i^{t_i} \right) && (3.38) \\ &= \sum_{V \subset [k]} M_{|V|}(x, p) \cdot \sum_{v \geq 0} p^v \cdot J_{k-|V|}^v(x, p) && (3.39) \end{aligned}$$

where we set z_i to 1. To simplify

$$\begin{aligned} \Phi_k^{(2)} = \frac{\partial}{\partial z([k])} \sum_{\vec{t} \in \mathbb{Z}_{\geq 0}^k} & \left(\sum_{V \subset [k]} \frac{\partial F_k(\vec{t}|x, p, \vec{y})}{\partial y([k] - V)} \times \right. \\ & \left. \frac{\partial}{\partial y(V)} \left(\sum_{v \geq 0} \sum_{i=1}^{\lambda} \left(\prod_{j \in S_i} y_j - 1 \right) \cdot p^{t_i^*} \cdot p^v \cdot H_k^v(\vec{t}|x, \vec{y}|t_i^*) \right) \right) \prod_{i \in V} z_i^{t_i}, \end{aligned}$$

we will use Lemma 22 and the following lemma (whose proof is deferred).

Lemma 24. *Let $\emptyset \neq W = \{w_1, \cdots, w_m\}$ for some $m \in \mathbb{N}$. For any $x_{w_1}, \cdots, x_{w_m}$,*

$$\prod_{i=1}^{m} x_{w_i} - 1 = \sum_{V = \{v_1, \cdots, v_l\} \subset W} (x_{v_1} - 1) \cdots (x_{v_l} - 1) \tag{3.40}$$

For $V \subset [k]$ and $\vec{t} \in \mathbb{Z}^k_{\geq 0}$, we let

$$\Phi(\vec{t}|V) = \frac{\partial}{\partial y(V)} \left(\sum_{v \geq 0} \sum_{i=1}^{\lambda} \left(\prod_{j \in S_i} y_j - 1 \right) \cdot p^{t_i^*} \cdot p^v \cdot H_k^v(\vec{t}|x, \vec{y}|t_i^*) \right).$$

We apply Lemma 24 to get

$$\Phi(\vec{t}|V) = \sum_{v \geq 0} p^v \cdot \frac{\partial}{\partial y(V)} \left(\sum_{i=1}^{\lambda} \sum_{U=\{u_1,\cdots,u_q\} \subset S_i} (y_{u_1} - 1) \cdots (y_{u_q} - 1) \cdot p^{t_i^*} \cdot H_k^v(\vec{t}|x, \vec{y}|t_i^*) \right). \tag{3.41}$$

Our goal is to take the sum

$$\sum_{\vec{t}(V) \in \mathbb{Z}^{|V|}_{\geq 0}} \Phi(\vec{t}|V) \prod_{i \in V} z_i^{t_i}.$$

For some $U = \{u_1, \cdots, u_q\} \subset [k]$, consider (S_i'), the partition of $[k]$ where the elements $u_1, \cdots, u_q$ are in the same part and all other elements are in their own parts. Notice that

$$(y_{u_1} - 1) \cdots (y_{u_q} - 1) \cdot p^{t_i^*} \cdot H_k^v(\vec{t}|x, \vec{y}|t_i^*) \tag{3.42}$$

is a term in the bracket of (Equation 3.41) if and only if the partition induced by $\vec{t}$ is such that (S_i') is a refinement of $(S_i)^{\vec{t}}_{\lambda}$, ie. $(S_i) \subset (S_i)^{\vec{t}}_{\lambda}$.

Assume V is such that $U \subset V$ (otherwise the term we are considering will vanish when we set y_i to 1). Let $V - U = \{j_1, \cdots, j_l\}$. We now consider the sum

$$\begin{aligned} \Phi(V, U) &= \sum_{v \geq 0} p^v \cdot \frac{\partial}{\partial y(V)} \sum_{t_{j_1} \geq 0} \cdots \sum_{t_{j_l} \geq 0} (y_{u_1} - 1) \cdots (y_{u_q} - 1) \times \\ &\qquad \sum_{t_i^* \geq 0} H_k^v(\vec{t}|x, \vec{y}|t_i^*) \cdot p^{t_i^*} (z_{u_1} \cdots z_{u_q})^{t_i^*} \prod_{i=1}^{l} z_{j_i}^{t_{j_i}} \\ &= \sum_{v \geq 0} p^v \cdot \sum_{t_{j_1} \geq 0} \cdots \sum_{t_{j_l} \geq 0} \frac{\partial H_k^v(\vec{t}|x, p z_{u_1} \cdots z_{u_q}, \vec{y})}{\partial y(V - U)} \prod_{i=1}^{l} z_{j_i}^{t_{j_i}}. \end{aligned} \tag{3.43}$$

We see that as we sum up $\Phi(\vec{t}|V)\prod_{i\in V} z_i^{t_i}$ over $\vec{t}$, we actually take the sum above for all $U \subset V$. Thus

$$\begin{aligned}
\frac{\partial}{\partial z(V)} \sum_{\vec{t}(V)\in\mathbb{Z}_{\geq 0}^{|V|}} \Phi(\vec{t}|V)\prod_{i\in V} z_i^{t_i}\Bigg|_{\vec{z}=\vec{1}} &= \sum_{\emptyset\neq U\subset V} \frac{\partial \Phi(V,U)}{\partial z(V)}\Bigg|_{\vec{z}=\vec{1}} && (3.44)\\
&= \sum_{\emptyset\neq U\subset V}\sum_{v\geq 0} p^v \cdot \frac{\partial J^v_{|V|-|U|}(x, pz_{u_1}\cdots z_{u_q})}{\partial z(U)}\Bigg|_{\vec{z}=\vec{1}} && (3.45)\\
&= \sum_{\emptyset\neq U\subset V}\sum_{v\geq 0} p^v \cdot \mathcal{D}_p^{|U|} J^v_{|V|-|U|}(x,p) && (3.46)
\end{aligned}$$

where for F, a function in some variables including p, we define the operator $\mathcal{D}_p^m$ recursively as

$$\mathcal{D}_p^m F = p\cdot \frac{\partial \mathcal{D}_p^{m-1}F}{\partial p} \qquad \text{and} \qquad \mathcal{D}_p^0 F = F. \tag{3.47}$$

Thus, we simplify $\Phi_k^{(2)}$ as follows.

$$\begin{aligned}
\Phi_k^{(2)} &= \sum_{V\subset[k]} \frac{\partial}{\partial z([k]-V)}\left(\sum_{\vec{t}([k]-V)\in\mathbb{Z}_{\geq 0}^{k-|V|}} \frac{\partial F_k(\vec{t}|x,p,\vec{1})}{\partial y([k]-V)} \prod_{i\in[k]-V} z_i^{t_i}\right)\times\\
&\qquad \frac{\partial}{\partial z(V)} \sum_{\vec{t}(V)\in\mathbb{Z}_{\geq 0}^{|V|}} \Phi_V(\vec{t})\prod_{i\in V} z_i^{t_i} && (3.48)\\
&= \sum_{V\subset[k]} M_{k-|V|}(x,p)\cdot \sum_{\emptyset\neq U\subset V}\sum_{v\geq 0} p^v\cdot \mathcal{D}_p^{|U|} J^v_{|V|-|U|}(x,p) && (3.49)
\end{aligned}$$

where we set z_i to 1.

We will denote $\mathcal{D}_p^m\left(F(x,p)\right)\Big|_{p=1}$ simply as $\mathcal{D}_p^{|U|}F(x)$.

Theorem 25. *For $k \geq 1$, the following recurrence holds.*

$$M_k(x,p) = \sum_{U\subset V\subset[k]} M_{k-|V|}(x,p)\cdot \sum_{v\geq 0} p^v\cdot \mathcal{D}_p^{|U|} J^v_{|V|-|U|}(x,p). \tag{3.50}$$

Furthermore, for $k \geq 1$ *and* $m \geq 0$, *let*

$$S(k,m) = \{(a,b,c,d) \in \mathbb{Z}^4 : 0 \leq a \leq b \leq m, 0 \leq c \leq d \leq k\}$$

and

$$S'(k,m) = S(k,m) - \{(0,0,0,0),(0,m,0,k)\}.$$

The following recurrence also holds.

$$\mathcal{D}_p^m M_k(x) = \frac{x}{\sqrt{1-4x}} \cdot \sum_{(a,b,c,d)\in S'(k,m)} \binom{k}{d}\binom{d}{c} \sum_{v\geq 0} v^a \cdot \mathcal{D}_p^{m-b} M_{k-d}(x) \cdot \mathcal{D}_p^{c+b-a} J_{d-c}^v(x). \tag{3.51}$$

Proof. From Equation 3.37, we see that, for $k \geq 1$,

$$\begin{aligned} M_k(x,p) &= x\Phi_k^{(1)} + x\Phi_k^{(2)} \\ &= \sum_{V\subset[k]} M_{k-|V|}(x,p) \cdot \sum_{\emptyset\neq U\subset V} \sum_{v\geq 0} p^v \cdot \mathcal{D}_p^{|U|} J_{|V|-|U|}^v(x,p) \\ &\qquad + \sum_{V\subset[k]} \sum_{v\geq 0} p^v \cdot M_{k-|V|}(x,p) \cdot J_{|V|}^v(x,p) \\ &= \sum_{U\subset V\subset[k]} M_{k-|V|}(x,p) \cdot \sum_{v\geq 0} p^v \cdot \mathcal{D}_p^{|U|} J_{|V|-|U|}^v(x,p). \end{aligned}$$

Differentiating the above expression, we see that

$$\begin{aligned} \mathcal{D}_p^m M_k(x,p) &= x \sum_{U\subset V\subset[k]} \sum_{b=0}^{m} \sum_{a=0}^{b} \sum_{v\geq 0} \mathcal{D}_p^a(p^v) \cdot \mathcal{D}_p^{m-b} M_{k-|V|}(x,p) \times \\ &\qquad \mathcal{D}_p^{|U|+b-a} J_{|V|-|U|}^v(x,p) \\ \mathcal{D}_p^m M_k(x) &= x \sum_{(a,b,c,d)\in S(k,m)} \binom{k}{d}\binom{d}{c} \sum_{v\geq 0} v^a \cdot \mathcal{D}_p^{m-b} M_{k-d}(x) \cdot \mathcal{D}_p^{c+b-a} J_{d-c}^v(x) \\ \mathcal{D}_p^m M_k(x)(1-2xM_0(x)) &= x \sum_{(a,b,c,d)\in S'(k,m)} \binom{k}{d}\binom{d}{c} \sum_{v\geq 0} v^a \cdot \mathcal{D}_p^{m-b} M_{k-d}(x) \cdot \mathcal{D}_p^{c+b-a} J_{d-c}^v(x). \end{aligned}$$

Notice that $M_0(x)$ is the generating function counting plane trees by number of edges.

Thus $1 - 2xM_0(x) = \sqrt{1-4x}$. Thus, we arrive at the desired result.

$\square$

For simple subtree additive properties $\mathcal{P}^*$ where the property in question is not clear, we will denote the generating function for $\sum_{T \in \mathfrak{T}_n} \mathcal{P}^*(T)^k$ by $M_k(\mathcal{P}^*, x)$. Let $[x^n]\mathcal{D}_p^m M_k(\mathcal{P}^*, x) = M_{k,n}^{(m)}(\mathcal{P}^*)$ (or simply $M_{k,n}^{(m)}$ if there is no ambiguity). Note that $M_{0,n}^{(m)}(\mathcal{P}^*) = \sum_{T \in \mathfrak{T}_n} \mathcal{P}(T)^m$ where $\mathcal{P}$ is the additive property from which $\mathcal{P}^*$ is derived.

We now prove the following lemmas from which the main theorem of this section will follows. To do so, we utilize the following lemma (whose proof is deferred).

Lemma 26. *Let $a_1, a_2 \in \mathbb{R}$ and $n \in \mathbb{N}$ be large. For $\min\{a_1, a_2\} > -1$,*

$$\sum_{n_1+n_2=n n_1,n_2 \geq 1} n_1^{a_1} \cdot n_2^{a_2} = \Theta\left(n^{a_1+a_2+1}\right)$$

and, for $\min\{a_1, a_2\} < -1$,

$$\sum_{n_1+n_2=n n_1,n_2 \geq 1} n_1^{a_1} \cdot n_2^{a_2} = \Theta\left(n^{\max\{a_1,a_2\}}\right).$$

Lemma 27. *Let $\mathcal{P}$ be a non-negative integer valued additive property with toll function f and let their induced subtree additive property be $\mathcal{P}^*$. Further assume that there exists $\zeta \in \mathbb{N}$ such that for all $T \in \mathfrak{T}_{\geq 0}$, $f(T) \leq \zeta \in \mathbb{N}$. For all m, k, the following holds.*

$$M_{k,n}^{(m)} = \Theta\left(n^{\frac{2m+3k-3}{2}} 4^n\right).$$

Proof. We show this result by induction on k and m. The base case of $k = 0$ (and any m) holds by Lemma 19. We now consider (Equation 3.51). Let

$$\Psi(k, m, a, b, c, d) = \frac{x}{\sqrt{1-4x}} \binom{k}{d} \binom{d}{c} \sum_{v \geq 0} v^a \cdot \mathcal{D}_p^{m-b} M_{k-d}(x) \cdot \mathcal{D}_p^{c+b-a} J_{d-c}^v(x). \quad (3.52)$$

We now take the coefficients on both sides of the equation. Let $[x^n]\mathcal{D}_p^m J_k^v(x) = J_{k,n}^{(m)}(v)$.

$$[x^n]\Psi(k,m,a,b,c,d) \;=\; \binom{k}{d}\binom{d}{c}\sum_{v\geq 0} v^a \sum_{\substack{n_1+n_2+n_3=n-1\\ n_1,n_2,n_3\geq 1}} M_{k-d,n_1}^{(m-b)} \cdot J_{d-c,n_2}^{(c+b-a)}(v) \cdot \Theta\left(n_3^{-\frac{1}{2}} 4^{n_3}\right)$$

where $(a,b,c,d) \in S'(k,m)$.

Fix $k \geq 1$, $m \geq 0$. Assuming the the theorem holds for all smaller k and any m as well as for equal k and smaller m. Notice that the right hand side of Equation 3.53 depends precisely on terms for which the theorem holds. We now see using Lemma 26 that

$$\begin{aligned}
[x^n]\Psi(k,m,a,b,c,d) \;&\leq\; \binom{k}{d}\binom{d}{c}\zeta^a \sum_{\substack{n_1+n_2+n_3=n-1\\ n_1,n_2,n_3\geq 1}} M_{k-d,n_1}^{(m-b)} \cdot M_{d-c,n_2}^{(c+b-a)}(v) \cdot O\left(n_3^{-\frac{1}{2}} 4^{n_3}\right) \\
&\leq\; O\left(4^n \sum_{\substack{n_1+n_2+n_3=n-1\\ n_1,n_2,n_3\geq 1}} n_1^{a_1} \cdot n_2^{a_2} \cdot n_3^{-\frac{1}{2}}\right) \qquad (3.53)
\end{aligned}$$

where $a_1 = \frac{2(m-b)+3(k-d)-3}{2}$ and $a_2 = \frac{2(c+b-a)+3(d-c)-3}{2}$. Notice that for $(a,b,c,d) \in S'(k,m)$, $a_1, a_2 \geq -\frac{3}{2}$. We now consider the 2 possible cases:

Case 1: $(a,b,c,d) \in S'(k,m)$ such that $d = k, m = b$ or $d = c, c+b = a$. We see that $\min\{a_1,a_2\} = -\frac{3}{2}$ and $\max\{a_1,a_2\} = \frac{2m+3k-c-2a-3}{2}$. Thus we apply Lemma 26 twice, noting that $\min\{a_1,a_2\} = -\frac{3}{2} < -1$, to get

$$[x^n]\Psi(k,m,a,b,c,d) \;\leq\; O\left(n^{\frac{2m+3k-c-2a-2}{2}} 4^n\right). \qquad (3.54)$$

Note that we get the most significant upper bound when we minimize $c+2a$. The minimal $c+2a$ for which there is $(a,b,c,d) \in S'(k,m)$ for some k,m in this case is when $c = 1, a = 0$.

Case 2: All other cases. We see that $\min\{a_1,a_2\} > -1$. Thus we apply Lemma 26 twice to get

$$[x^n]\Psi(k,m,a,b,c,d) \;\; \leq \;\; O\left(n^{\frac{2m+3k-c-2a-3}{2}}4^n\right). \tag{3.55}$$

Note that we get the most significant upper bound when we minimize $c+2a$. The minimal $c+2a$ for which there is $(a,b,c,d)\in S'(k,m)$ in this case is when $c=0, a=0$. We note that when $k=1, m=0$ there is no $(a,b,c,d)\in S'(k,m)$ in this case.

Thus, for all $(a,b,c,d)\in S'(k,m)$, $[x^n]\Psi(k,m,a,b,c,d) \leq O\left(4^n n^{\frac{2m+3k-3}{2}}\right)$. From Equation 3.51, we hence get

$$\mathcal{D}_p^m M_k(x) \;\; = \;\; \sum_{(a,b,c,d)\in S'(k,m)} \Psi(k,m,a,b,c,d) \tag{3.56}$$

$$M_{k,n}^{(m)} \;\; \leq \;\; O\left(4^n n^{\frac{2m+3k-3}{2}}\right). \tag{3.57}$$

Towards the lower bound, we see when $a=0$,

$$\begin{aligned}[x^n]\Psi(k,m,a,b,c,d) \;\; &\geq \;\; \binom{k}{d}\binom{d}{c} \sum_{\substack{n_1+n_2+n_3=n-1\\ n_1,n_2,n_3\geq 1}} M_{k-d,n_1}^{(m-b)} \cdot M_{d-c,n_2}^{(c+b-a)} \cdot \Theta\left(n_3^{-\frac{1}{2}}4^{n_3}\right) \\ &\geq \;\; \Omega\left(4^n \sum_{\substack{n_1+n_2+n_3=n-1\\ n_1,n_2,n_3\geq 1}} n_1^{a_1}\cdot n_2^{a_2}\cdot n_3^{-\frac{1}{2}}\right) \end{aligned} \tag{3.58}$$

where $a_1 = \frac{2(m-b)+3(k-d)-3}{2}$ and $a_2 = \frac{2(c+b-a)+3(d-c)-3}{2}$. We break this into cases exactly as before. We notice that there is $(a,b,c,d)\in S'(k,m)$ where $a=0$ and $[x^n]\Psi(k,m,a,b,c,d) \geq \Omega\left(4^n n^{\frac{2m+3k-3}{2}}\right)$. Thus, from Equation 3.51, we get

$$M_{k,n}^{(m)}(\mathcal{P}^*) \;\; \geq \;\; \Omega\left(4^n n^{\frac{2m+3k-3}{2}}\right). \tag{3.59}$$

The result follows by induction.

□

Observation 28. *In the proof of Lemma 27, for any $k, m \geq 0$ and $(a, b, c, d) \in S'(k, m)$, when $a \geq 1$,*

$$[x^n]\Psi(k, m, a, b, c, d) \leq O\left(n^{\frac{2m+3k-4}{2}} 4^n\right).$$

Lemma 29. *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be non-negative integer valued additive properties with toll functions f_1 and f_2, respectively. Let the induced subtree additive properties of $\mathcal{P}_1$ and $\mathcal{P}_2$ be $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$, respectively. Further assume that there exists $\zeta \in \mathbb{N}$ such that for all $T \in \mathfrak{T}_{\geq 0}$, $f_1(T) \leq \zeta \in \mathbb{N}$ and $f_2(T) \leq \zeta \in \mathbb{N}$. Assume for all $m, n \geq 0$,*

$$\sum_{T \in \mathfrak{T}_n} \mathcal{P}_1(T)^m = \mu^m \cdot \sum_{T \in \mathfrak{T}_n} \mathcal{P}_2(T)^m + O\left(n^{\frac{2m-4}{2}} 4^n\right)$$

where μ is a constant. It then holds that for all $n, m, k \geq 0$,

$$M_{k,n}^{(m)}(\mathcal{P}_1^*) = \mu^{k+m} \cdot M_{k,n}^{(m)}(\mathcal{P}_2^*) + O\left(n^{\frac{2m+3k-4}{2}} 4^n\right).$$

Proof. We prove the result by induction on k and m. The base case of $k = 0$ (and any m) is true by assumption. Fix $k \geq 1$, $m \geq 0$. Assuming the the theorem holds for all smaller k and any m as well as for equal k and smaller m. We now apply Lemma 27 and Observation 28 regarding $\Psi(k, m, a, b, c, d)$ to get

$$\begin{aligned} M_{k,n}^{(m)}(\mathcal{P}_1^*) &= \sum_{(a,b,c,d)\in S'(k,m)} \binom{k}{d}\binom{d}{c} \sum_{v\geq 0} v^a \sum_{n_1+n_2+n_3=n-1} M_{k-d,n_1}^{(m-b)}(\mathcal{P}_1^*) \\ &\qquad \times J_{d-c,n_2}^{(c+b-a)}(\mathcal{P}_1^*, v) \cdot \binom{2n_3}{n_3} \\ &= \mu^{m+k} \sum_{\substack{(a,b,c,d)\in S'(k,m)\\ a=0}} \binom{k}{d}\binom{d}{c} \sum_{n_1+n_2+n_3=n-1} M_{k-d,n_1}^{(m-b)}(\mathcal{P}_2^*) \qquad (3.60) \\ &\qquad \times M_{d-c,n_2}^{(c+b-a)}(\mathcal{P}_2^*) \cdot \binom{2n_3}{n_3} \\ &\quad + \sum_{\substack{(a,b,c,d)\in S'(k,m)\\ a\geq 1}} \Psi(k,m,a,b,c,d) + O\left(n^{\frac{2m+3k-4}{2}} 4^n\right) \qquad (3.61) \\ &= \mu^{m+k} \cdot M_{k,n}^{(m)}(\mathcal{P}_2^*) + O\left(n^{\frac{2m+3k-4}{2}} 4^n\right) \qquad (3.62) \end{aligned}$$

Thus the result holds for all $k, m \in \mathbb{N}$ by induction.

□

Lemma 30. *Let $\mathcal{P}$ be a non-negative integer valued additive properties of plane trees with toll function f. Let the subtree additive property induced by $\mathcal{P}$ be $\mathcal{P}^*$. Further assume that there exist $\zeta \in \mathbb{N}$ such that for any $T \in \mathfrak{T}_{\geq 0}$, $f(T) \leq \zeta$. The limiting distribution of*

$$\frac{\mathcal{P}^*(\mathfrak{T}_n)}{\sqrt{n^3}}$$

is unique determined by its moments. Specifically, it satisfies the Carleman's condition.

Proof. Let $\mathcal{P}_1$ be the additive property with toll function f where $f(T) = \zeta$ for all $T \in \mathfrak{T}_{\geq 0}$. Let $\mathcal{P}_1^*$ be the subtree additive property derived from $\mathcal{P}_1$. We see that $\mathcal{P}(T) \leq \mathcal{P}_1(T)$, and hence, $\mathcal{P}^*(T) \leq \mathcal{P}_1^*(T)$ for all $T \in \mathfrak{T}_{\geq 0}$. From Example 2, $\mathcal{P}_1(T) = \zeta \cdot \mathcal{P}^v(T)$, thus

$\mathcal{P}_1^*(T) = \zeta \cdot \mathcal{P}^{PL}(T)$. Applying Theorem 5, we see that

$$\lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^*(\mathfrak{T}_n)}{\sqrt{2n^3}}\right)^k\right] \leq \lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\zeta \cdot \mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}}\right)^k\right] \sim \frac{6k}{\sqrt{2}}\left(\frac{k}{12e}\right)^{\frac{k}{2}} \cdot \zeta^k$$

as $k \to \infty$. Thus applying the Carleman Condition (Theorem 1), we get the desired result.

□

Proof of Theorem 17. We recall that for all $n, k \geq 0$ and $\mathcal{P}^* \in \{\mathcal{P}_1^*, \mathcal{P}_2^*\}$, $M_{k,n}^{(0)}(\mathcal{P}^*) = \sum_{T\in\mathfrak{T}_n} \mathcal{P}^*(T)^k$, hence $\mathbb{E}[\mathcal{P}^*(\mathfrak{T}_n)^k] = \frac{M_{k,n}^{(0)}(\mathcal{P}^*)}{C_n}$. We apply the assumption of the Theorem 17, Lemma 27 and Lemma 29 to achieve

$$\mathbb{E}\left[\left(\frac{\mathcal{P}_1^*(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right] = \mu^k \cdot \mathbb{E}\left[\left(\frac{\mathcal{P}_2^*(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right] + O\left(\frac{1}{\sqrt{n}}\right). \tag{3.63}$$

We now apply Lemma 30 to see that the limiting distribution of the properties are unique determined by their moments. Thus, we arrive at the desired result.

□

Corollary 31. *The total leaf to root distance of a random plane tree on n edges, $\mathcal{P}^{LR}(\mathfrak{T}_n)$, and the total internal node to root distance of a random plane tree on n edges, $\mathcal{P}^{IR}(\mathfrak{T}_n)$ is asymptotically Airy distributed, up to a scaling factor. Specifically, as $n \to \infty$,*

$$2 \cdot \frac{\mathcal{P}^{LR}(\mathfrak{T}_n)}{\sqrt{2n^3}} \quad \overset{d}{\longleftrightarrow} \quad 4 \cdot \frac{\mathcal{P}^{IR}(\mathfrak{T}_n)}{\sqrt{2n^3}} \quad \overset{d}{\longleftrightarrow} \quad \frac{\mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}}.$$

Proof. Let $F(x,p)$ be the generating function for plane trees weighted by number of edges and vertices. Let $F_{d_0}(x,p)$ be the generating function for plane trees weighted by number of edges and leaves. Let $F_{d_1}(x,p)$ be the generating function for plane trees weighted by number of edges and internal nodes. From Corollary 12,

$$F_{d_0}(x,p) = \frac{1 + (1-p)x - \sqrt{(1+(1-p)x)^2 - 4x}}{2x}$$

and

$$F_{d_1}(x,p) = \frac{1+(1-p)x - \sqrt{(1+(1-p)x)^2 - 4x(1-(p-1)x)}}{2x}.$$

Thus, we see that

$$[x^n]\frac{\partial F_{d_0}(x,1)}{\partial p^m} = \frac{1}{2^m} \cdot [x^n]\frac{\partial F(x,1)}{\partial p^m} + O\left((1-4x)^{\frac{2-2m}{2}}\right)$$

and

$$[x^n]\frac{\partial F_{d_1}(x,1)}{\partial p^m} = \frac{1}{4^m} \cdot [x^n]\frac{\partial F(x,1)}{\partial p^m} + O\left((1-4x)^{\frac{2-2m}{2}}\right).$$

Thus

$$\sum_{T\in\mathfrak{T}} \mathcal{P}^{d_0}(T)^m = \frac{1}{2^m} \cdot \sum_{T\in\mathfrak{T}} \mathcal{P}^{v}(T)^m + O\left(n^{\frac{2m-4}{2}} 4^n\right)$$

and

$$\sum_{T\in\mathfrak{T}} \mathcal{P}^{d_1}(T)^m = \frac{1}{4^m} \cdot \sum_{T\in\mathfrak{T}} \mathcal{P}^{v}(T)^m + O\left(n^{\frac{2m-4}{2}} 4^n\right).$$

We now apply Theorem 17 to get that, as $n \to \infty$,

$$2 \cdot \frac{\mathcal{P}^{LR}(\mathfrak{T}_n)}{\sqrt{2n^3}} \quad \xleftrightarrow{d} \quad 4 \cdot \frac{\mathcal{P}^{IR}(\mathfrak{T}_n)}{\sqrt{2n^3}} \quad \xleftrightarrow{d} \quad \frac{\mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}}.$$

From Theorem 5, we know that $\frac{\mathcal{P}^{PL}(\mathfrak{T}_n)}{\sqrt{2n^3}}$ converges weakly to an Airy random variable, thus the same holds for $2 \cdot \frac{\mathcal{P}^{LR}(\mathfrak{T}_n)}{\sqrt{2n^3}}$ and $4 \cdot \frac{\mathcal{P}^{IR}(\mathfrak{T}_n)}{\sqrt{2n^3}}$.

□

3.3.3 The Distribution of Classes of Subtree Additive Properties under NNTM

Let $f : \mathfrak{T}_{\geq 0} \times \mathfrak{T}_{\geq 0} \to \mathbb{R}$ be a polynomial in the number of edges, leaves and internal nodes of its input trees. Hence, $f(T', T) \in \mathbb{R}[t,p,q][n,\overline{p},\overline{q}]$ where t, p, q represents the number of edges, leaves and internal nodes of T' and $n, \overline{p}, \overline{q}$ represents the number of edges, leaves and internal nodes of T.

We now consider $\mathcal{P}^f$, the subtree additive property given by

$$\mathcal{P}^f(T) = \sum_{v \in \overline{\mathcal{V}}(T)} f(n(T_v), d_0(T_v), d_1(T_v), n(T), d_0(T), d_1(T)) = \sum_{v \in \overline{\mathcal{V}}(T)} f(T_v, T) \quad (3.64)$$

where, for $v \in \overline{\mathcal{V}}(T)$, the root vertex in T_v is allowed to count as a leaf or internal node.

Let f be a polynomial over $\mathbb{R}$ in the variables $x_1, \cdots, x_n$. We can write f in the form of $\sum_{j=1}^{L} w_j \cdot v_j$ where $w_j \in \mathbb{R}$ and v_j is a monic monomial in the variables such that the monomial are all distinct. We will call this the reduced form of f. It should be clear that f can be written uniquely in this form. For any such v_j and variables $x_1, \cdots, x_l$, let $\Delta(v_j|x_1, \cdots, x_l)$ be the degree of the monomial v_j if we consider all variables other than $x_1, \cdots, x_l$ to be constant. For a vector of variables $\vec{x} = (x_1, \cdots, x_l)$, let $\Delta(v_j|\vec{x}) = \Delta(v_j|x_1, \cdots, x_l)$. We also define $\Delta(f|x_1, \cdots, x_l) = \max_{1 \le j \le L} \Delta(v_j|x_1, \cdots, x_l)$.

Fix parameters $\alpha, \beta, \gamma \in \mathbb{R}$. Let

$$M_{k,n}(f, \alpha, \beta, \gamma) = \sum_{T \in \mathfrak{T}_n} \mathcal{P}^f(T)^k e^{-E(T)} = \mathbb{E}\left[\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}_n)^k\right] \cdot \mathcal{Z}_{(n,\alpha,\beta,\gamma)}$$

and

$$M_k(f, \alpha, \beta, \gamma)(x) = \sum_{n \ge 0} M_{k,n}(f, \alpha, \beta, \gamma) x^n.$$

We provide the following theorem that specifies the asymptotic form of $M_{k,n}(f, \alpha, \beta, 0)$.

Theorem 32. *Fix $k \in \mathbb{N}$, $\alpha, \beta \in \mathbb{R}$ and $f \in \mathbb{R}[t, p, q][n, \overline{p}, \overline{q}]$. Let $\delta = \Delta(f|t, n, p, \overline{p}, q, \overline{q})$. Let $V_k(f) = \frac{(2\delta+1)k-1}{2}$ if for every monomial of f, u, such that $\Delta(u|t, n, p, \overline{p}, q, \overline{q}) = \delta$, $\Delta(u|t, p, q) > 0$ and $V_k(f) = \frac{2(\delta+1)k-1}{2}$ otherwise. We have that*

$$M_k(f,\alpha,\beta,0)(x) = \frac{W(\alpha,\beta,f)}{(1-\rho x)^{V_k(f)}} + O\left(\frac{1}{(1-\rho x)^{V_k(f)-\frac{1}{2}}}\right),$$

where $\rho = e^{-\alpha} + e^{-\beta} + 2e^{-\frac{\alpha}{2}}$ and $W(\alpha,\beta,f)$ is a constant that depends on f,α,β.

Furthermore, assume every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q}) = \delta$ is such that $\Delta(u|t,n) = \delta_n$, $\Delta(u|p,\overline{p}) = \delta_{d_0}$ and $\Delta(u|q,\overline{q}) = \delta_{d_1}$ where δ_n, δ_{d_0} and δ_{d_1} are constants that depend only on f. We have that

$$W(\alpha,\beta,f) = Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)\cdot P_k(f)$$

where $Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)$ depends on $\delta_{d_0},\delta_{d_1}$ and $V_k(f)$, and $P_k(f)$ is a constant that depends on f and is independent of α and β.

We now describe a construction that will allow us to prove the above result. Fix $k \in \mathbb{N}$. Let $\mathcal{C}_k^p$ be the set of compositions of k into p parts, where a composition of an integer k into p parts is a p-tuple $(c_1,\cdots,c_p)$ such that $c_i \in \mathbb{N}$ and $\sum_{i=1}^p c_i = k$. Let

$$\mathcal{V}_k^* = \bigcup_{T\in\mathfrak{T}_{\geq 1}} \{T\}\times\overline{\mathcal{V}}(T)^k \qquad \text{and} \qquad \mathcal{F}_k^* = \bigcup_{1\leq i\leq k} \mathfrak{T}_i \times \mathcal{C}_k^i.$$

Let $(T,\vec{v}) \in \mathcal{V}_k^*$. We will define the *skeleton tree* of this tuple to be the tree formed by contracting the parent edge of all vertices except those in $\vec{v}$. We will denote the skeleton tree of the tuple as $\mathcal{S}(T,\vec{v})$. We define the composition of the tuple $(T,\vec{v})$ by $\mathcal{C}(T,\vec{v}) = (c_1^*,\cdots,c_\lambda^*)$, where c_i^* is the number of times the ith non-root vertex in the pre-ordering of $\mathcal{S}(T,\vec{v})$ appears in $\vec{v}$. Notice that $\mathcal{C}(T,\vec{v}) \in \mathcal{C}_k^\lambda$ and $\mathcal{S}(T,\vec{v}) \in \mathfrak{T}_\lambda$ where $\lambda \leq k$ is the number distinct vertices in $\vec{v}$.

We define $\phi : \mathcal{V}_k^* \to \mathcal{F}_k^*$ as $\phi(T,\vec{v}) = (\mathcal{S}(T,\vec{v}),\mathcal{C}(T,\vec{v}))$. (See the top of Figure 3.1.) We will denote the pre-image of $(T_s,C) \in \mathcal{F}_k^*$ under ϕ as $\phi^{-1}(T_s,C) \subset \mathcal{V}_k^*$. We will now show a construction for the elements of $\phi^{-1}(T_s,C)$ that will allow us to count the elements of $\phi^{-1}(T_s,C)$.

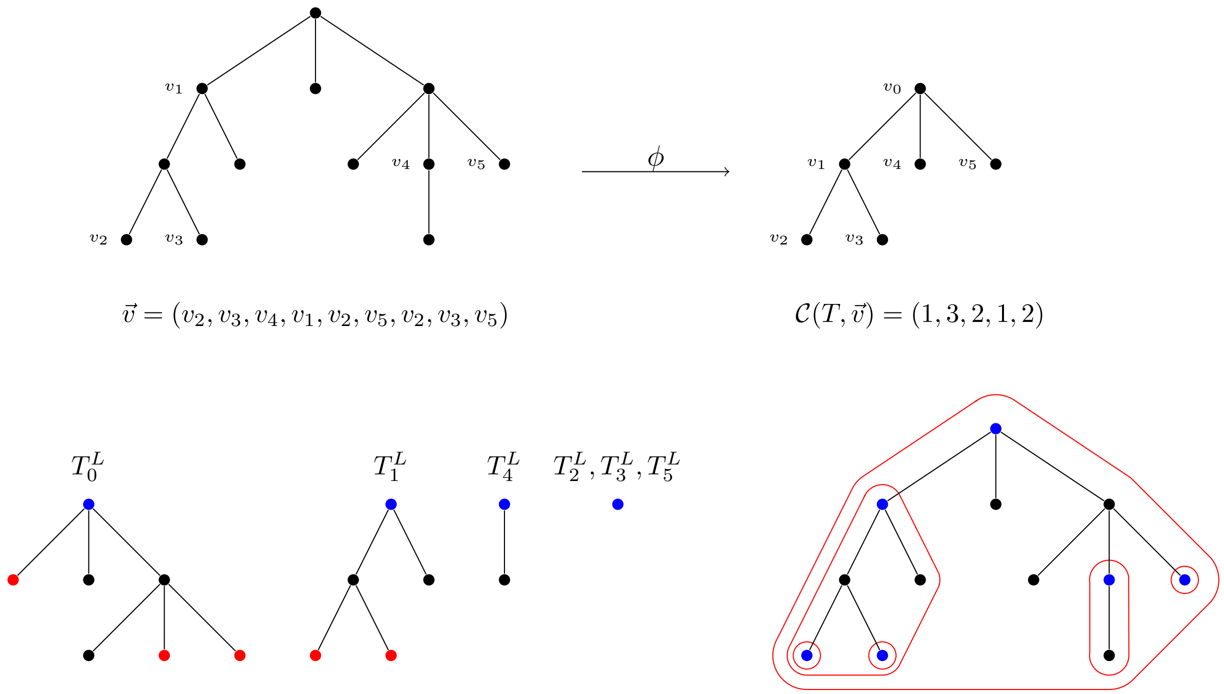


Figure 3.1: Illustration of the map ϕ and the construction of elements in $\phi^{-1}(T_s, C)$. We consider $k = 9$. The top left tree is $T \in \mathfrak{T}_{11}$ and the top right tree is $T_s = \mathcal{S}(T, \vec{v})$. To achieve $(T, \vec{v})$ from the construction we describe, we select local trees T_i^L as in the bottom left trees. The red vertices are the leaf vertices we must choose. The bottom rightmost tree is T formed by the construction. The resulting fused trees T_i^F are outlined in red. The blue vertices are the roots of the local trees.

Let $(T_s, C) \in \mathcal{F}_k^*$ such that $T_s \in \mathfrak{T}_\lambda$. We label each vertex in T_s. We assign 0 to the root vertex and assign i to the ith non-root vertex in the pre-ordering of T_s. From this point, we will refer to the vertices of T_s by their assigned number. Let m_i be the number of children of the ith vertex of T_s.

Each element of $\phi^{-1}(T_s, C)$ may be constructed as follows. To each vertex in T_s, we shall assign 2 trees. We shall call these trees the local tree and the fused tree assigned to the vertex. We now proceed recursively. Consider a vertex v, labelled i, in T_s with children to all of which a local and fused tree has been assigned. We select a tree with t_i edges, p_i leaves, q_i internal nodes and root degree y_i. This tree will be the local tree assigned to the vertex, denoted T_i^L.

We select m_i leaves in T_i^L. We then identify these leaves with the root vertices of the fused trees assigned to the children of v. We denote these root vertices $v_1, \cdots, v_{m_i}$, where v_i is the root vertex assigned to the fused tree of the ith child of v encountered in a pre-order traversal. We identify each root vertex to a unique leaf in such a way that v_i is encountered before v_{i+1} is a pre-order traversal of the new tree formed. This new tree will be the fused tree assigned to v, denoted T_i^F. We continue this process until all the vertices in T_s have a fused and local tree assigned to them. (See the bottom of Figure 3.1.)

The elements of $\phi^{-1}(T_s, C)$ this construction will correspond to, $(T, \vec{v})$, will have $T = T_0^F$. The vertices in $\vec{v}$ will be root vertices of each T_i^L for $i \neq 0$. The number of times each vertex appears in $\vec{v}$ is determined by $C = (c_1^*, \cdots, c_\lambda^*)$, where the root of T_i^L appears c_i^* times. We are now free to choose any arrangement of these vertices into a tuple. There are $\frac{k!}{c_1^*! \cdots c_\lambda^*!}$ such choices, which we will denote as $\binom{k}{C}$. Notice that every tuple formed by this process of building up a tree and arranging special vertices from the tree is unique.

Observation 33. *Let $S_i \subset \{0, \cdots, \lambda\}$ be the set indices j, such that the vertex i of T_s is an ancestor of vertex j. The number of edges in T_i^F is $\hat{t}_i = \sum_{j \in S_i} t_j$. The number of vertices in T_i^F which are leaves in T_0^F is $\hat{p}_i = \sum_{j \in S_i}(p_j - m_j)$ if we declare that T_j^L has 1 leaf* *if $T_j^L = T^*$ and vertex j is a leaf vertex of T_s. The number of vertices in T_i^F which are internal nodes in T_0^F is $\hat{q}_i = \sum_{j \in S_i} q_j$ if we declare that for all $j \neq 0$, T_j^L has an extra leaf if it has root degree 1.*

The adjustments are to account for the fact that the root of a proper sub-tree of a larger tree may be an internal node or a leaf relative to the larger tree but not relative to itself.

Recall the definition of $\mathcal{G}_n(d_0, d_1, r)$ in subsection 3.3.1. Let $\mathcal{G}'_n(d_0, d_1, r)$ be defined similarly to $\mathcal{G}_n(d_0, d_1, r)$ but where we declare that root vertices with down degree 1 are internal nodes. Let $\mathcal{G}''_n(d_0, d_1, r)$ be defined similarly to $\mathcal{G}'_n(d_0, d_1, r)$ but where we also declare that the tree on 0 edges has 1 leaf. Let $L(T_s) \subset \{1, \cdots, \lambda\}$ be the set of indices greater than 0 corresponding to leaf vertices in T_s. Let $\overline{L}(T_s) \subset \{1, \cdots, \lambda\}$ be the set of indices greater than 0 corresponding to non-leaf vertices in T_s.

Let $\vec{t} = (t_0, \cdots, t_\lambda)$, $\vec{p} = (p_0, \cdots, p_\lambda)$, $\vec{q} = (q_0, \cdots, q_\lambda)$ and $\vec{y} = (y_0, \cdots, y_\lambda)$. Let $\mathcal{H}(T_s, C, \vec{y}, \vec{t}, \vec{p}, \vec{q})$ be the set of all elements in $\phi^{-1}(T_s, C)$ with fixed t_i, p_i, q_i, y_i given by $\vec{t}, \vec{p}, \vec{q}, \vec{y}$. From the above construction, after making the adjustments specified in Observation 33, we see that

$$\begin{aligned} |\mathcal{H}(T_s, C, \vec{y}, \vec{t}, \vec{p}, \vec{q})| &= \binom{k}{C} \cdot \left[\binom{p_0}{m_0} \cdot |\mathcal{G}_{t_0}(p_0, q_0, y_0)| \right] \cdot \left[\prod_{i \in \overline{L}(T_s)} \binom{p_i}{m_i} \cdot |\mathcal{G}'_{t_i}(p_i, q_i, y_i)| \right] \times \\ & \left[\prod_{i \in L(T_s)} \binom{p_i}{m_i} \cdot |\mathcal{G}''_{t_i}(p_i, q_i, y_i)| \right] \end{aligned} \tag{3.65}$$

where $\binom{p_i}{m_i}$ counts the number of ways to pick the leaves which will be identified with the root vertices of the fused trees assigned to the children of vertex i. Notice that when $p_i < m_i$, $\binom{p_i}{m_i}$ forces the entire term to 0 thus we do not require any extra restriction on the properties of the trees chosen.

We now let x_i, a_i, b_i be variables that count the number of edges in T_i^L, the number of vertices in T_i^L which are leaves in T_0^F and the number of vertices in T_i^L which are internal nodes in T_0^F, respectively. Also, let c weigh the root degree of T_0^F. Let $\vec{x} = (x_0, \cdots, x_\lambda)$, $\vec{a} = (a_0, \cdots, a_\lambda)$ and $\vec{b} = (b_0, \cdots, b_\lambda)$. Let

$$R_i'(T_s, t_i, p_i, q_i, y_i) = x_i^{t_i} a_i^{p_i - m_i} b_i^{q_i} c^{y_i} \cdot \binom{p_i}{m_i} \cdot |\mathcal{G}_{t_i}(p_i, q_i, y_i)| \tag{3.66}$$

for $i = 0$,

$$R_i'(T_s, t_i, p_i, q_i, y_i) = x_i^{t_i} a_i^{p_i - m_i} b_i^{q_i} \cdot \binom{p_i}{m_i} \cdot |\mathcal{G}_{t_i}'(p_i, q_i, y_i)| \tag{3.67}$$

for $i \in \overline{L}(T_s)$, and

$$R_i'(T_s, t_i, p_i, q_i, y_i) = x_i^{t_i} a_i^{p_i - m_i} b_i^{q_i} \cdot \binom{p_i}{m_i} \cdot |\mathcal{G}_{t_i}''(p_i, q_i, y_i)| \tag{3.68}$$

for $i \in L(T_s)$. We now let

$$R_i(T_s, x_i, a_i, b_i, c) = \sum_{t_i, p_i, q_i, y_i \geq 0} R_i'(T_s, t_i, p_i, q_i, y_i). \tag{3.69}$$

For convenience, we make the following definitions.

$$\begin{aligned}
\Psi(x, a, b) &= [1 + (2 - a - b)x]^2 - 4x[1 - (b - 1)x] && (3.70) \\
X(x, a, b) &= -\frac{\partial \Psi(x, a, b)}{\partial x} && (3.71) \\
A(x, a, b) &= -\frac{\partial \Psi(x, a, b)}{\partial a} && (3.72) \\
B(x, a, b) &= -\frac{\partial \Psi(x, a, b)}{\partial b} && (3.73)
\end{aligned}$$

For conciseness, we let $\Psi_i = \Psi(x_i, a_i, b_i)$, $X_i = X(x_i, a_i, b_i)$, $A_i = A(x_i, a_i, b_i)$ and $B_i = B(x_i, a_i, b_i)$. We will simply write Ψ, X, A, B to refer to $\Psi(x, a, b)$, $X(x, a, b)$, $A(x, a, b)$ and $B(x, a, b)$, respectively.

Recall Theorem 11 and Corollary 12 and the definitions in subsection 3.3.1. From

Equation 3.69, we see that

$$R_i(T_s, x_i, a_i, b_i, c) = \frac{1}{m_i!} \cdot \frac{\partial G(x_0, a_0, b_0, c)}{\partial a_0^{m_0}} \tag{3.74}$$

for $i = 0$,

$$R_i(T_s, x_i, a_i, b_i, c) = \frac{1}{m_i!} \cdot \frac{\partial \left(G(x_i, a_i, b_i) + (b_i - 1)G^*(x_i, a_i, b_i)\right)}{\partial a_i^{m_i}} \tag{3.75}$$

for $i \in \overline{L}(T_s)$, and

$$R_i(T_s, x_i, a_i, b_i, c) = \frac{1}{m_i!} \cdot \frac{\partial \left(G(x_i, a_i, b_i) + (b_i - 1)G^*(x_i, a_i, b_i) + a_i - 1\right)}{\partial a_i^{m_i}} \tag{3.76}$$

for $i \in \overline{L}(T_s)$, where the difference between the cases is to account for whether or not trees with root degrees 0 and 1 as having an extra leaf and internal node, respectively. Note that

$$G(x_i, a_i, b_i) + (b_i - 1)G^*(x_i, a_i, b_i) + a_i - 1 = \frac{G^*(x_i, a_i, b_i)}{x_i}.$$

Recall the definition of the operator $\mathcal{D}_p^m$ in Equation 3.47.

Lemma 34. *Fix $(T_s, C) \in \mathcal{F}_k^*$ for $k \geq 1$. Let $u_x, u_a, u_b, \in \mathbb{Z}_{\geq 0}$ and $u = u_x + u_a + u_b$. Let $\mathbb{1}_n = -1$ for $n = 0$ and $\mathbb{1}_n = 1$ otherwise. Let $\mathbb{1}'_n = 1$ for $n = 0$ and $\mathbb{1}'_n = 0$ otherwise. We have that, for $0 \leq i \leq \lambda$,*

$$\mathcal{D}_{x_i}^{u_x}\mathcal{D}_{a_i}^{u_a}\mathcal{D}_{b_i}^{u_b}\left[a_i^{m_i} R_i(T_s, x_i, a_i, b_i, 1)\right] = \mathbb{1}'_{u+m_i} \cdot Z'_i + \mathbb{1}_{u+m_i} \cdot Z_i(u+m_i) \cdot (x_i X_i)^{u_x} \cdot (a_i A_i)^{u_a+m_i} \times (b_i B_i)^{u_b} \cdot \Psi_i^{\frac{1-2(u+m_i)}{2}} + O\left(p_i \cdot \Psi_i^{\frac{2-2(u+m_i)}{2}}\right), \tag{3.77}$$

where, $Z_0(n) = \frac{(2n-3)!!}{2^n \cdot m_0!} \cdot \frac{1}{2x_0}$, for $i > 0$, $Z_i(n) = \frac{(2n-3)!!}{2^n \cdot m_i!} \cdot \frac{1}{2x_i(1-(b_i-1)x_i)}$, for $i \in \overline{L}$, $Z'_i = \frac{1+(a_i-b_i)x_i}{2x_i(1-(b_i-1)x_i)}$ and p_i is a sum of quotients of polynomial whose denominators are the *products of powers of* x_i *and* $1-(b_i-1)x_i$.

Proof. Follows by differentiating $G(x,a,b)$, $\frac{G^*(x,a,b)}{x}$ and $\frac{G^*(x,a,b)}{x}+1-a$. We note that for $i \notin \overline{L}$, we omit stating Z_i' since since $u+m_i>0$.

□

Fix $\alpha, \beta \in \mathbb{R}$ and $f \in \mathbb{R}[t,p,q][n,\overline{p},\overline{q}]$. For $(T_s, C) \in \mathcal{F}_k^*$ such that $T_s \in \mathfrak{T}_\lambda$ and $C=(c_1^*, \cdots, c_\lambda^*)$, let

$$f_k(T_s,C) = \prod_{i=1}^{\lambda} f(\hat{t}_i, \hat{p}_i, \hat{q}_i, \hat{t}_0, \hat{p}_0, \hat{q}_0)^{c_i^*} \tag{3.78}$$

such that

$$f_k(T_s,C) = \sum_{j=1}^{L} w_j \cdot v_j \qquad \text{where} \qquad v_j = \prod_{i=0}^{\lambda} t_i^{v_n(i,j)} \cdot p_i^{v_{d_0}(i,j)} \cdot q_i^{v_{d_1}(i,j)} \tag{3.79}$$

where $w_j \in \mathbb{R}$ and $v_n(i,j)$, $v_{d_0}(i,j)$, $v_{d_1}(i,j) \in \mathbb{Z}_{\geq 0}$ for all $0 \leq i \leq \lambda$ and $1 \leq j \leq L$. Furthermore, let $\sigma_{i,j} = v_n(i,j) + v_{d_0}(i,j) + v_{d_1}(i,j) + m_i$ and $\{\sigma_{i,j}\}_{>0} = \{0 \leq i \leq \lambda : \sigma_{i,j} > 0\}$. Finally, let

$$R_{i,j}(T_s, x_i, a_i, b_i) = \frac{1}{a^{m_i}} \cdot \mathcal{D}_{x_i}^{v_n(i,j)} \mathcal{D}_{a_i}^{v_{d_0}(i,j)} \mathcal{D}_{b_i}^{v_{d_1}(i,j)} \left[a^{m_i} R_i(T_s, x_i, a_i, b_i, 1)\right] \tag{3.80}$$

for $0 \leq i \leq \lambda$.

Lemma 35. *For* $f_k(T_s,C) = \sum_{j=1}^{L} w_j \cdot v_j$ *as in (Equation 3.79), the following equation* *holds.*

$$M_k(f,\alpha,\beta,0)(x) = \sum_{(T_s,C)\in\mathcal{F}_k^*} \binom{k}{C} \sum_{j=1}^{L} w_j \cdot A^{\lambda} \cdot (e^{-\alpha}A)^{\sum_{i=0}^{\lambda} v_{d_0}(i,j)} \cdot (e^{-\beta}B)^{\sum_{i=0}^{\lambda} v_{d_1}(i,j)} \times$$

$$(xX)^{\sum_{i=0}^{\lambda} v_n(i,j)} \cdot \Psi^{\frac{|\{\sigma_{i,j}\}_{>0}|-2\sum_{i=0}^{\lambda}\sigma_{i,j}}{2}} \cdot (Z')^{1+\lambda-|\{\sigma_{i,j}\}_{>0}|} \cdot \left(\prod_{i\in\{\sigma_{i,j}\}_{>0}} Z_i(\sigma_{i,j})\right)$$

$$+\, O\left(p(x)\cdot \Psi^{\frac{1+|\{\sigma_{i,j}\}_{>0}|-2\sum_{i=0}^{\lambda}\sigma_{i,j}}{2}}\right), \quad (3.81)$$

where $Z' = \frac{1+(e^{-\alpha}-e^{-\beta})x}{2x(1-(e^{-\beta}-1)x)}$*, for* $i = 0$*,* $Z_i(n) = \frac{(2n-3)!!}{2^n \cdot m_i!} \cdot \frac{1}{2x}$*, for* $i > 0$*,* $Z_i(n) = \frac{(2n-3)!!}{2^n \cdot m_i!} \cdot \frac{1}{2x(1-(e^{-\beta}-1)x)}$ *and* $p(x)$ *is a sum of quotients of polynomial whose denominators are the products of powers of* x *and* $1-(e^{-\beta}-1)x$*.*

Proof. We observe that

$$\begin{aligned} M_{k,n}(f,\alpha,\beta,0) &= \sum_{T\in\mathfrak{T}_n}\left(\sum_{v\in\overline{\mathcal{V}}(T)} f(T_v,T)\right)^{k} e^{-E(T)} \\ &= \sum_{T\in\mathfrak{T}_n}\left(\sum_{\vec{v}\in\overline{\mathcal{V}}(T)^k} \prod_{i=1}^{k} f(T_{v_i},T)\right) e^{-E(T)} \\ &= \sum_{(T_s,C)\in\mathcal{F}_k^*}\left(\sum_{\substack{(T,\vec{v})\in\phi^{-1}(T_s,C)\\ T\in\mathfrak{T}_n}} e^{-E(T)} \prod_{i=1}^{k} f(T_{v_i},T)\right), \qquad (3.82) \end{aligned}$$

where $\vec{v} = (v_1,\cdots,v_k)$.

We now notice that for any $(T,\vec{v}) \in \phi^{-1}(T_s,C)$ formed by the construction we previously described, with fixed $\vec{t}$, $\vec{p}$, $\vec{q}$ and $\vec{y}$, $\prod_{i=1}^{k} f(T_{v_i},T) = f_k(T_s,C)$. We have also shown that the number of such $(T,\vec{v})$ is $|\mathcal{H}(T_s,C,\vec{t},\vec{p},\vec{q},\vec{y})|$. Hence, following from Equation 3.82, we achieve

$$\sum_{\substack{(T,\vec{e})\in\phi^{-1}(T_s,C)\\ T\in\mathfrak{T}_n}} e^{-E(T)}\prod_{i=1}^{k} f(T_{v_i},T) = \sum_{\sum_{i=0}^{\lambda} t_i=n}\sum_{\vec{p},\vec{q},\vec{y}\in\mathbb{Z}_{\geq 0}^{\lambda+1}} e^{-E(\hat{p}_0,\hat{q}_0)}|\mathcal{H}(T_s,C,\vec{t},\vec{p},\vec{q},\vec{y})|f_k(T_s,C) \tag{3.83}$$

where $\hat{t}_i,\hat{p}_i,\hat{q}_i$ are defined as in Observation Observation 33 and $E(\hat{p}_0,\hat{q}_0)=\alpha\hat{p}_0+\beta\hat{q}_0$. Recall the definitions in (Equation 3.79) and (Equation 3.80). We now show that

$$\begin{aligned} M_k(f,\alpha,\beta,0)(x) &= \sum_{(T_s,C)\in\mathcal{F}_k^*}\sum_{n\geq 0} x^n \sum_{\substack{(T,\vec{v})\in\phi^{-1}(T_s,C)\\ T\in\mathfrak{T}_n}} e^{-E(T)}\prod_{i=1}^{k}\mathcal{P}(T_{v_i},T) \\ &= \sum_{(T_s,C)\in\mathcal{F}_k^*}\binom{k}{C}\sum_{j=1}^{L} w_j\prod_{i=0}^{\lambda} R_{i,j}(T_s,x,e^{-\alpha},e^{-\beta}). \end{aligned} \tag{3.84}$$

We notice that to get $R_{i,j}(T_s,x_i,a_i,b_i)$, we first multiply by a^{m_i} to get the power of a_i to be p_i. The application of the differentiation operator $\mathcal{D}$ then creates the factor of $t_i^{v_n(i,j)}\cdot p_i^{v_{d_0}(i,j)}\cdot q_i^{v_{d_1}(i,j)}$ (without changing the power of x_i,a_i,b_i). We then divide by a^{m_i} to get the power of a_i to be p_i-m_i. Taking the product of all these terms, we now notice that if we set all the x_i to x, a_i to a and b_i to b, x, a and b now weight the number of edges, leaves and internal nodes in T for $(T,\vec{v})\in\phi^{-1}(T_s,C)$. Finally, we set a to $e^{-\alpha}$ and b to $e^{-\beta}$ to weight by $e^{-E(T)}$. We set c to 1 as, for $\gamma=0$, we are not weighting by the root degree.

By Lemma 34, we get an expression for $R_{i,j}(T_s,x,e^{-\alpha},e^{-\beta})$. We consider when $\sigma_{i,j}=0$ for some i. We notice that the most significant term of $R_{i,j}$ with respect to Ψ has order $\frac{1}{2}$. Thus we get a higher order term in $\prod_{i=0}^{\lambda}R_{i,j}(T_s,x,e^{-\alpha},e^{-\beta})$ by taking the term independent of Ψ for i such that $\sigma_{i,j}=0$. Furthermore, note that any i such that $\sigma_{i,j}=0$ must correspond to a leaf vertex of T_s since otherwise $\sigma_{i,j}\geq m_i\geq 1$. Thus $0\in\{\sigma_{i,j}\}_{>0}$ since the root of T_s cannot be a leaf (since $T_s\in\mathfrak{T}_{\geq 1}$).

$$\begin{aligned}\prod_{i=0}^{\lambda} R_{i,j}(T_s, x, e^{-\alpha}, e^{-\beta}) = A^{\sum_{i=0}^{\lambda} m_i} \cdot (e^{-\alpha}A)^{\sum_{i=0}^{\lambda} v_{d_0}(i,j)} \cdot (e^{-\beta}B)^{\sum_{i=0}^{\lambda} v_{d_1}(i,j)} \cdot (xX)^{\sum_{i=0}^{\lambda} v_n(i,j)} \\ \times \Psi^{\frac{1+\lambda-m-2\sum_{i=0}^{\lambda}\sigma_{i,j}}{2}} \cdot (Z')^m \cdot \left(\prod_{i\in\{\sigma_{i,j}\}_{>0}} Z_i(\sigma_{i,j})\right) \\ + O\left(p(x)\cdot \Psi^{\frac{2+\lambda-m-2\sum_{i=0}^{\lambda}\sigma_{i,j}}{2}}\right)\end{aligned} \tag{3.85}$$

where $p(x)$ is a sum of quotients of polynomials whose denominators are the products of powers of x and $1-(e^{-\beta}-1)x$ and m is the number of i such that $\sigma_{i,j}=0$ and is thus equal to $1+\lambda-|\{\sigma_{i,j}\}_{>0}|$.

To conclude the proof, we see that $\sum_{i=0}^{\lambda} m_i = \lambda$ since this is the sum of the number of children of all vertices of a tree on λ edges.

□

Notice that every monomial of f^k is of the form $\prod_{i=1}^{k} u_i$ where u_i is a monomial in f. To go from f^k to f_k, in each of the monomials $\prod_{i=1}^{k} u_i$, we set t, p, q to $\hat{t}_j, \hat{p}_j, \hat{q}_j$ and $n, \overline{p}, \overline{q}$ to $\hat{t}_0, \hat{p}_0, \hat{q}_0$, respectively, in u_i from some j for each i. After making this change, we can expand each of the $\hat{t}_j$, $\hat{p}_j$, $\hat{q}_j$ to express $\prod_{i=1}^{k} u_i$ as the sum of monomials in t_i, p_i, q_i. Notice that for every monomial formed in this way, v_j, $\Delta\left(v_j|\vec{t}\right) = \Delta\left(\prod_{i=1}^{k} u_i\middle| t, n\right)$, $\Delta\left(v_j|\vec{q}\right) = \Delta\left(\prod_{i=1}^{k} u_i\middle| q, \overline{q}\right)$ and $\Delta\left(v_j|\vec{p}\right) \leq \Delta\left(\prod_{i=1}^{k} u_i\middle| p, \overline{q}\right)$. We note that the inequality in the last relation occurs because $\hat{p}_j = \sum_{i\in S_j} p_i - m_i$ may have a constant term. We however also note that the inequality is tight.

Let $\delta = \Delta(f|t, p, q, n, \overline{p}, \overline{q})$. Towards simplifying the expression in Lemma 35, we establish the following lemmas.

Lemma 36. *Fix $k \in \mathbb{N}$. Consider $f^k = \sum_{j=1}^{L_k} w'_j \cdot v'_j$ in reduced form where v'_j is a monomial. For any $1 \leq j \leq L_k$, the inequality*

$$\Delta(v'_j|t, p, q, n, \overline{p}, \overline{q}) = \Delta(v'_j|t, n) + \Delta(v'_j|p, \overline{p}) + \Delta(v'_j|q, \overline{q}) \leq \delta k$$

holds and is tight.

Proof. The fact that δk is an upper bound for the degree of monomials in the reduced form of f^k should be clear. Every monomial of degree δk, must be achieved as the product of monomials of degree δ in f. Let $f = \sum_{j=1}^{L} v_j$ in reduced form such that the $v_{j_1} \cdots v_{j_m}$ are all the monomials on f of degree δk. Since $\mathbb{R}[t, p, q][n, \overline{p}, \overline{q}]$ is an integral domain, $(w_{j_1} \cdot v_{j_1} + \cdots + w_{j_m} \cdot v_{j_m})^k \neq 0$ since $w_{j_1} \cdot v_{j_1} + \cdots + w_{j_m} \cdot v_{j_m} \neq 0$. Thus, in reduced form, there is at least one monomial of f^k of degree δk.

□

Recall $f_k = \sum_{j=1}^{L} w_j \cdot v_j$. Since every monomial in f_k, v_j, is formed from a monomial f^k in the way we previously described, $\Delta(v_j|\vec{t}, \vec{p}, \vec{q}) \leq \delta k$ and this inequality is tight. Clearly $v_n(i, j) = \Delta(v_j|t_i)$, thus $\sum_{i=0}^{\lambda} v_n(i, j) = \Delta(v_j|\vec{t})$. Similarly, $\sum_{i=0}^{\lambda} v_{d_0}(i, j) = \Delta(v_j|\vec{p})$ and $\sum_{i=0}^{\lambda} v_{d_1}(i, j) = \Delta(v_j|\vec{q})$. Thus, we achieve

$$\sum_{i=0}^{\lambda} \sigma_{i,j} = \sum_{i=0}^{\lambda} v_n(i, j) + v_{d_0}(i, j) + v_{d_1}(i, j) + m_i \leq \delta k + \lambda \tag{3.86}$$

and this inequality is tight.

We will call v_j, a mononial of f_k, such that $\sum_{i=0}^{\lambda} \sigma_{i,j} = \delta k + \lambda$ a maximal monomial of f_k. We also call u, a monomial in f, such that $\Delta(u|t, p, q, n, \overline{p}, \overline{q}) = \delta$ a maximal monomial of f. We now consider, for a maximal monomial v_j, when $\sigma_{i,j} = 0$ for some $0 \leq i \leq \lambda$.

Lemma 37. *There exist a maximal monomial v_j where $\sigma_{i,j} = 0$ for some $0 \leq i \leq \lambda$ if and only if, in the reduced form, f has a maximal monomial, u, such that $\Delta(u|t, p, q) = 0$.*

Furthermore, when f has a maximal monomial, u, such that $\Delta(u|t,p,q) = 0$, there exist j where $\sigma_{i,j} = 0$ where vertex i is a leaf of T_s.

Proof. For each vertex i corresponding to a non-leaf of T_s, $m_i \geq 1$, hence $\sigma_{i,j} \geq 1$. Let vertex i be leaf vertex of T_s.

Assume that every maximal monomial in the the reduced form of f is such that $\Delta(u|t,p,q) \geq 1$. We have seen that v_j is derived from a monomial of f^k, $\prod_{i=1}^{k} u_i$, where u_i must be maximal monomial of f. To go from $\prod_{l=1}^{k} u_l$ to monomials in f_k, in at least one of u_l, t, p, q are set to $\hat{t}_i = \sum_{l \in S_i} t_l = t_i$, $\hat{p}_i = \sum_{l \in S_i} p_l - m_l = p_i$ and $\hat{q}_i = \sum_{l \in S_i} q_l = q_i$, respectively. Assume (WLOG) $\Delta(u|t) \geq 1$. Thus t_i divides every monomial of f_k from $\prod_{i=1}^{k} u_i$, including v_j. Thus $\sigma_{i,j} \geq v_n(i,j) \geq 1$.

Assume, in the reduced form, f has a maximal monomial, u, such that $\Delta(u|t,p,q) = 0$, thus $u(n, \overline{p}, \overline{q}) = n^{l_n} \overline{p}^{l_{d_0}} \overline{q}^{l_{d_1}}$ for some l_n, l_{d_0}, l_{d_1} such that $l_n + l_{d_0} + l_{d_1} = \delta$. Consider $\prod_{l=1}^{k} u_l$, where $u_l = u$ for all l.

Setting $n, \overline{p}, \overline{q}$ to $\hat{t}_0, \hat{p}_0, \hat{q}_0$, respectively, in $\prod_{l=1}^{k} u_l$ and expanding the term, we get

$$\prod_{l=1}^{k} u_l = \hat{t}_0^{kl_n} \hat{p}_0^{kl_{d_0}} \hat{q}_0^{kl_{d_1}} = \left(\sum_{i=0}^{\lambda} t_i \right)^{kl_n} \left(\sum_{i=0}^{\lambda} p_i \right)^{kl_{d_0}} \left(\sum_{i=0}^{\lambda} q_i - m_i \right)^{kl_{d_1}} = t_0^{kl_n} p_0^{kl_{d_0}} q_0^{kl_{d_1}} + \cdots .$$

Thus $t_0^{kl_n} p_0^{kl_{d_0}} q_0^{kl_{d_1}}$ is maximal monomial of f_k with $v_n(i,j) = v_{d_0}(i,j) = v_{d_1}(i,j) = 0$. Thus $\sigma_{i,j} = 0$. Notice that this holds whenever vertex i is a leaf vertex of T_s.

□

From Lemma 35, to achieve the most significant with respect to Ψ in $M_k(f, \alpha, \beta, 0)(x)$, we must minimize $|\{\sigma_{i,j}\}_{>0}|$. When f does not have a maximal monomial, u, such that $\Delta(u|t,p,q) = 0$, $|\{\sigma_{i,j}\}_{>0}| = \lambda + 1$. When f has a maximal monomial, u, such that $\Delta(u|t,p,q) = 0$, $|\{\sigma_{i,j}\}_{>0}|$ is at least the number of non-leaf vertices in T_s (and this is tight). Thus, taking T_s to be the tree where every non-root vertex is a leaf (the bush), we get $|\{\sigma_{i,j}\}_{>0}| = 1$.

Finally, we achive the following lemma that tells us that in all cases, the term of $M_k(f, \alpha, \beta, 0)(x)$ from non-maximal monomial of f_k are insignificant with respect to Ψ.

Lemma 38. *The most significant terms in the sum for $M_k(f, \alpha, \beta, 0)(x)$ in Lemma 35 must be achieve from maximal monomial of f_k and $T_s \in T_k$.*

Proof. When f has a maximal monomial, u, such that $\Delta(u|t, p, q) = 0$, By Lemma 37, the maximum order with respect to Ψ of a term in $M_k(f, \alpha, \beta, 0)(x)$ is $\frac{1-2(\delta k+k)}{2}$ (when T_s is the bush on k edges).

Assume f has no maximal monomial, u, such that $\Delta(u|t, p, q) = 0$. Notice that the order with respect to Ψ of a term in $M_k(f, \alpha, \beta, 0)(x)$ from a maximal monomial of f_k is $\frac{1-2\delta k-k}{2}$ (when $T_s \in T_k$). To possibly get a non-maximal monomial of f_k which leads to a term in $M_k(f, \alpha, \beta, 0)(x)$ of higher order with respect to Ψ, we must minimize $|\{\sigma_{i,j}\}_{>0}|$.

By similar argument to the latter part of the proof of Lemma 37, any monomial, v_j, of f_k achieved from a monomial of f^k, $\prod_{i=1}^k u_i$, where each of the u_i are maximal monomials of f, has $\sigma_{i,j} \geq 1$ for all i. Thus, we need only consider monomial, v_j, of f_k where achieved from a monomial of f^k, $\prod_{i=1}^k u_i$, where at least one of the u_i are maximal monomials of f.

Consider any monomial, v_j, of f_k achieved from a monomial of f^k, $\prod_{l=1}^k u_l$, where each of the u_l are monomials of f and at least one of which is non-maximal. Let $s \geq 1$ be the number of non-maximal u_l. By similar argument the former part of the proof of Lemma 37, the number of i where $\sigma_{i,j} = 0$ is at most s. Clearly $\Delta(u_l|n, t, p, \overline{p}, q, \overline{q}) \leq \delta - 1$ for all non maximal u_l. Thus we see that $\sum_{i=0}^{\lambda} \sigma_{i,j} \leq \delta(k-s) + (\delta-1)s + \lambda = \delta k - s + \lambda$. Considering the order with respect to Ψ of the term in $M_k(f, \alpha, \beta, 0)(x)$ from v_j, we see that

$$\frac{|\{\sigma_{i,j}\}_{>0}| - 2\sum_{i=0}^{\lambda} \sigma_{i,j}}{2} \geq \frac{1+s-2\delta k-\lambda}{2} > \frac{1-2\delta k-\lambda}{2}. \tag{3.87}$$

We notice that to minimize $\frac{1-2\delta k-\lambda}{2}$, we must maximize $\lambda \leq k$, which is achieved when

$T_s \in \mathfrak{T}_k$ proving the result.

$\square$

We now prove the main theorem using the above results.

Proof of Theorem 32. From Lemma 35, we have an expression for $M_k(f, \alpha, \beta, 0)(x)$. We assume (WLOG) that the maximal monomials of f_k are $v_1, \cdots, v_{L_{\max}}$. We now break the proof into 2 cases.

Case 1: All the maximal monomial of f, u, are such that $\Delta(u|t, p, q) > 0$. By Lemma Lemma 37, for every maximal monomial of f_k, v_k, $\sigma_{i,j} \geq 1$. By Lemma 38, to get the most significant term with respect to Ψ we need only consider the maximal monomials of f_k when $T_s \in \mathfrak{T}_k$. For $(T_s, C) \in \mathcal{F}_k^*$, when $T_s \in \mathfrak{T}_k$, $C = (1, 1, \cdots, 1)$. We now see that

$$\begin{aligned} M_k(f, \alpha, \beta, 0)(x) &= \sum_{(T_s,C)\in\mathcal{F}_k^*} \sum_{j=1}^{L} w_j \cdot \binom{k}{C} \prod_{i=0}^{\lambda} R_{i,j}(x, e^{-\alpha}, e^{-\beta}) && (3.88) \\ &= \sum_{T_s\in\mathfrak{T}_k} W_{T_s}(f, x, a, b)\Psi^{\frac{1-(2\delta+1)k}{2}} + O\left(\Psi^{\frac{2-(2\delta+1)k}{2}}\right) && (3.89) \\ &= W_f(x, \alpha, \beta)\Psi^{\frac{1-(2\delta+1)k}{2}} + O\left(\Psi^{\frac{2-(2\delta+1)k}{2}}\right) && (3.90) \end{aligned}$$

where, for $T_s \in \mathfrak{T}_k$,

$$\begin{aligned} W_{T_s}(f, x, a, b) = k! \sum_{j=1}^{L_{\max}} w_j \cdot A^k \left(e^{-\alpha} A\right)^{\sum_{i=0}^{\lambda} v_{d_0}(i,j)} \left(e^{-\beta} B\right)^{\sum_{i=0}^{\lambda} v_{d_1}(i,j)} \times \\ (xX)^{\sum_{i=0}^{\lambda} v_n(i,j)} \cdot \left(\prod_{0\leq i\leq k} Z_i(\sigma_{i,j})\right) \quad (3.91) \end{aligned}$$

and $W_f(x, a, b) = \sum_{T_s\in\mathfrak{T}} W_{T_s}(f, x, a, b)$.

We now assume further that for any maximal monomial u of f, $\Delta(u|t, n) = \delta_n$, $\Delta(u|p, \overline{p}) = \delta_{d_0}$ and $\Delta(u|q, \overline{q}) = \delta_{d_1}$ where δ_n, δ_{d_0} and δ_{d_1} are constants that depend only on f and not u. Thus for any maximal monomial of f_k, v_j, we see that $\sum_{i=0}^{\lambda} v_n(i, j) =$

$\Delta(v_j|\vec{t}) = \delta_n k$. Similarly, $\sum_{i=0}^{\lambda} v_{d_0}(i,j) = \delta_{d_0} k$ and $\sum_{i=0}^{\lambda} v_{d_1}(i,j) = \delta_{d_1} k$. Thus,

$$W_f(x,a,b) = \frac{A^k \left(e^{-\alpha}A\right)^{\delta_{d_0}k} \left(e^{-\beta}B\right)^{\delta_{d_1}k} (xX)^{\delta_n k}}{x^{k+1}(1-(e^{-\beta}-1)x)^k} \cdot \frac{k!}{2^{(2+\delta)k+1}} \times \sum_{T_s \in \mathfrak{T}} \sum_{j=1}^{L_{\max}} w_j \left(\prod_{0 \le i \le k} \frac{(2\sigma_{i,j}-3)!!}{m_i!} \right) \tag{3.92}$$

Case 2: There exists a maximal monomial of f, u, are such that $\Delta(u|t,p,q) = 0$. Combining Lemma Lemma 37 and Lemma 38, to get the most significant term with respect to Ψ we need only consider the maximal monomials of f_k when T_s is the bush on k edges. Thus $C = (1, 1, \cdots, 1)$. Assume $v_1, \cdots, v_{L_{\max}}$ are the maximal monomials of f_k where $\sigma_{i,j} = 0$ for all $i \neq 0$. We now see that

$$\begin{aligned} M_k(f,\alpha,\beta,0)(x) &= \sum_{(T_s,C)\in\mathcal{F}_k^*} \sum_{j=1}^{L} w_j \cdot \binom{k}{C} \prod_{i=0}^{\lambda} R_{i,j}(x, e^{-\alpha}, e^{-\beta}) && (3.93) \\ &= W_f(x,\alpha,\beta)\Psi^{\frac{1-(2\delta+2)k}{2}} + O\left(\Psi^{\frac{2-(2\delta+2)k}{2}}\right) && (3.94) \end{aligned}$$

where

$$W_f(x,a,b) = k! \sum_{j=1}^{L_{\max}} w_j \cdot A^k \left(e^{-\alpha}A\right)^{\sum_{i=0}^{\lambda} v_{d_0}(i,j)} \left(e^{-\beta}B\right)^{\sum_{i=0}^{\lambda} v_{d_1}(i,j)} \times (xX)^{\sum_{i=0}^{\lambda} v_n(i,j)} \cdot (Z')^k \cdot Z_0. \tag{3.95}$$

We now assume further that for any maximal monomial u of f, $\Delta(u|t,n) = \delta_n$, $\Delta(u|p,\overline{p}) = \delta_{d_0}$ and $\Delta(u|q,\overline{q}) = \delta_{d_1}$ where δ_n, δ_{d_0} and δ_{d_1} are constants that depend only on f and not u. Thus for any maximal monomial of f_k, v_j, we see that $\sum_{i=0}^{\lambda} v_n(i,j) =$

$\Delta(v_j|\vec{t}) = \lambda\delta_n$. Similarly, $\sum_{i=0}^{\lambda} v_{d_0}(i,j) = \lambda\delta_{d_0}$ and $\sum_{i=0}^{\lambda} v_{d_1}(i,j) = \lambda\delta_{d_1}$. Thus,

$$\begin{aligned} W_f(x,a,b) = \frac{A^k \left(e^{-\alpha}A\right)^{\delta_{d_0}k} \left(e^{-\beta}B\right)^{\delta_{d_1}k} \left(xX\right)^{\delta_n k}}{x} \cdot \left(\frac{1+(e^{-\alpha}-e^{-\beta})x}{x(1-(e^{-\beta}-1)x)}\right)^k \times \\ \frac{(2(1+\delta)k-3)!!}{2^{(2+\delta)k+1}} \cdot \sum_{j=1}^{L_{\max}} w_j. \end{aligned} \quad (3.96)$$

Let $\rho = e^{-\alpha} + e^{-\beta} + 2e^{-\frac{\alpha}{2}}$ and $\overline{\rho} = e^{-\alpha} + e^{-\beta} - 2e^{-\frac{\alpha}{2}}$. Notice that $M_k(f,\alpha,\beta,0)(x)$ is the sum of products of derivatives of $G(x,e^{-\alpha},e^{-\beta})$ and $G^*(x,e^{-\alpha},e^{-\beta})$. Thus, by Lemma 13, the dominant singularity in $M_k(f,\alpha,\beta,0)(z)$ occurs at $z = \frac{1}{\rho}$. Thus

$$M_k(f,\alpha,\beta,0)(z) = \frac{W_f(z,\alpha,\beta)}{(1-\overline{\rho}z)^{V_k(f)}} \cdot (1-\rho z)^{-V_k(f)} + O\left(p(x) \cdot (1-\rho z)^{\frac{1}{2}-V_k(f)}\right) \quad (3.97)$$

where $V_k(f) = \frac{(2\delta+1)k-1}{2}$ if for every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q}) = \delta$, $\Delta(u|t,p,q) > 0$ and $V_k(f) = \frac{2(\delta+1)k-1}{2}$ otherwise. Notice that $p(x)$ and $\frac{W_f(x,\alpha,\beta)}{(1-\overline{\rho}x)^{V_k(f)}}$ are analytic in the disk $R = \left\{z \in \mathbb{C} : |z| \leq \frac{1}{\rho}\right\}$. Thus by Taylor's Theorem,

$$M_k(f,\alpha,\beta,0)(z) \quad = \quad \frac{W_f\left(\frac{1}{\rho},\alpha,\beta\right)}{\left(1-\frac{\overline{\rho}}{\rho}\right)^{V_k(f)}} \cdot (1-\rho z)^{-V_k(f)} + O\left((1-\rho z)^{\frac{1}{2}-V_k(f)}\right) (3.98)$$

This completes the proof of the theorem. □

Remark 39. *Specifically, when $V_k(f) = \frac{(2\delta+1)k-1}{2}$,*

$$Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta) = \sqrt{\rho e^{-\frac{\alpha}{2}}} \cdot \left(\frac{(e^{-\frac{\alpha}{2}}+1)^{\delta_{d_0}-1} \cdot e^{-\frac{\alpha}{4}(2\delta_{d_0}-1)} \cdot e^{-\beta\delta_{d_1}}}{\sqrt{\rho^{2\delta_{d_0}+2\delta_{d_1}-1}}}\right)^k \quad (3.99)$$

and, when $V_k(f) = \frac{2(\delta+1)k-1}{2}$,

$$Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta) = \sqrt{\rho e^{-\frac{\alpha}{2}}} \cdot \left(\frac{(e^{-\frac{\alpha}{2}}+1)^{\delta_{d_0}} \cdot e^{-\frac{\alpha\delta_{d_0}}{2}} \cdot e^{-\beta\delta_{d_1}}}{\rho^{\delta_{d_0}+\delta_{d_1}}}\right)^k. \quad (3.100)$$

Corollary 40. *Fix $\alpha, \beta \in \mathbb{R}$ and $f \in \mathbb{R}[t,p,q][n,\overline{p},\overline{q}]$. Let $\delta = \Delta(f|t,n,p,\overline{p},q,\overline{q})$. Let $V'(f) = \frac{2\delta+1}{2}$ if for every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q}) = \delta$, $\Delta(u|t,p,q) > 0$ and $V'(f) = \frac{2(\delta+1)}{2}$ otherwise. There is a unique distribution with kth moment given by*

$$\lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^{f}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}}\right)^k\right].$$

Proof. We first note that $V_k(f) + \frac{1}{2} = V'(f)k$. Let

$$f = \sum_{i=1}^{L} w_i \cdot t^{a_{i,1}} p^{a_{i,2}} q^{a_{i,3}} n^{a_{i,4}} \overline{p}^{a_{i,5}} \overline{q}^{a_{i,6}}$$

where $w_i \in \mathbb{R}$, $a_{i,j} \in \mathbb{Z}_{\geq 0}$. We notice that for any tree, the number of leaves and the number of internal node is at most the number of edges in the tree. We thus see that for $t,p,q,n,\overline{p},\overline{q} \in \mathbb{Z}_{\geq 0}$ where $p,q \leq t$ and $\overline{p},\overline{q} \leq n$,

$$f \leq \sum_{i=1}^{L} |w_i| \cdot t^{a_{i,1}} p^{a_{i,2}} q^{a_{i,3}} n^{a_{i,4}} \overline{p}^{a_{i,5}} \overline{q}^{a_{i,6}} \tag{3.101}$$

$$\leq \sum_{i=1}^{L} |w_i| \cdot t^{\sum_{j=1}^{3} a_{i,j}} \cdot n^{\sum_{j=4}^{6} a_{i,j}}. \tag{3.102}$$

We now break the proof into 2 cases as we did for the proof of Theorem 32.

Case 1: All the maximal monomial of f, u, are such that $\Delta(u|t,p,q) > 0$. For $t \leq n$,

$$f \leq \sum_{i=1}^{L} |w_i| \cdot t \cdot n^{\sum_{j=1}^{6} a_{i,j} - 1} \leq \sum_{i=1}^{L} |w_i| \cdot t \cdot n^{\delta - 1}. \tag{3.103}$$

Let $f' = \omega \cdot t \cdot n^{\delta-1}$ and $f'' = t$ where $\omega = \sum_{i=1}^{L} |w_i|$. We thus see that

$$\mathcal{P}^{f}(T) \leq \mathcal{P}^{f'}(T) = \omega \cdot n(T)^{\delta-1} \sum_{v \in \overline{\mathcal{V}}} n(T_v) = \omega \cdot n(T)^{\delta-1} \cdot \mathcal{P}^{f''}(T)$$

for all $T \in \mathfrak{T}_{\geq 0}$. Thus

$$\mathbb{E}\left[\left(\frac{\mathcal{P}^{f}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}}\right)^k\right] \leq \mathbb{E}\left[\left(\frac{\omega \cdot n^{\delta-1} \cdot \mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{\frac{2\delta+1}{2}}}\right)^k\right] = \omega^k \cdot \mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right].$$

Let $\rho = e^{-\alpha}+e^{-\beta}+2e^{-\frac{\alpha}{2}}$ and $\overline{\rho} = e^{-\alpha}+e^{-\beta}-2e^{-\frac{\alpha}{2}}$. By the defintion of $M_{k,n}(f'', \alpha, \beta, 0)$,

$$\mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(\alpha,\beta,\gamma)}(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right] = \frac{M_{k,n}(f'', \alpha, \beta, 0)}{\mathcal{Z}_{(n,\alpha,\beta,0)} \cdot \sqrt{n^{3k}}}.$$

We now apply Theorem 9, Theorem 32, and Corollary 14, to see that

$$M_k = \lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right] = \frac{W_f\left(\frac{1}{\rho}, \alpha, \beta\right)}{\left(1-\frac{\overline{\rho}}{\rho}\right)^{\frac{3k-1}{2}}} \cdot \frac{2\sqrt{\pi}}{\sqrt{e^{-\frac{\alpha}{2}}\rho} \cdot \Gamma\left(\frac{3k-1}{2}\right)}.$$

Applying Remark Remark 39, we see that

$$\lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right] = c_0 \cdot c_1^k \cdot \lim_{n\to\infty} \mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(0,0,0)}(\mathfrak{T}_n)}{\sqrt{n^3}}\right)^k\right]$$

where c_0, c_1 are constants. Note that $\mathcal{P}^{f''}$ is a simple subtree additive property with bounded toll function. Thus by Lemma 30, the moments of $\frac{\mathcal{P}^{f''}_{(0,0,0)}(\mathfrak{T}_n)}{\sqrt{n^3}}$ satisfy Carleman's condition (Theorem 1). This thus implies that the moments of $\frac{\mathcal{P}^{f}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}}$ also satisfy Carleman's condition. Thus we get the desired result.

Case 2: There exists a maximal monomial of f, u, are such that $\Delta(u|t,p,q) = 0$. For $t \leq n$,

$$f \leq \sum_{i=1}^{L} |w_i| \cdot n^{\sum_{j=1}^{6} a_{i,j}} \leq \sum_{i=1}^{L} |w_i| \cdot n^{\delta}. \tag{3.104}$$

Let $f' = \omega \cdot n^{\delta}$ and $f'' = 1$ where $\omega = \sum_{i=1}^{L} |w_i|$. We note that $\mathcal{P}^{f''}_{(\alpha,\beta,0)}(T) = n$ for all

$T \in \mathfrak{T}_n$ (since this counted the number of subtrees in the tree). We thus see that

$$\mathcal{P}^f(T) \le \mathcal{P}^{f'}(T) = \omega \cdot n(T)^\delta \sum_{v \in \overline{\mathcal{V}}} 1 = \omega \cdot n(T)^\delta \cdot \mathcal{P}^{f''}(T)$$

for all $T \in \mathfrak{T}_{\ge 0}$. Thus

$$\mathbb{E}\left[\left(\frac{\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}}\right)^k\right] \le \mathbb{E}\left[\left(\frac{\omega \cdot n^\delta \cdot \mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{\frac{2(\delta+1)}{2}}}\right)^k\right] = \omega^k \cdot \mathbb{E}\left[\left(\frac{\mathcal{P}^{f''}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n}\right)^k\right] = \omega^k. \tag{3.105}$$

We now apply the Carleman's condition to get the desired result in this case.

□

Fix $h \in \mathbb{N}$. Let $\mathfrak{T}^h_n$ for $h \in \mathbb{N}$ be the set of plane trees on n edges with root degree at most h. We define

$$M_{k,n}(f, \alpha, \beta, \gamma |\, h) = \sum_{T \in \mathfrak{T}^h_n} \mathcal{P}^f(T)^k e^{-E(T)}$$

and

$$M_k(f, \alpha, \beta, \gamma |\, h)(x) = \sum_{n \ge 0} M_{k,n}(f, \alpha, \beta, \gamma |\, h) x^n.$$

Theorem 41. *Fix $k \in \mathbb{Z}_{\ge 0}$ and $\alpha, \beta, \gamma \in \mathbb{R}$. Let $f \in \mathbb{R}[t,p,q][n,\overline{p},\overline{q}]$. Let $\delta = \Delta(f|t,n,p,\overline{p},q,\overline{q})$. Let $V_k(f) = \frac{(2\delta+1)k-1}{2}$ if for every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q}) = \delta$, $\Delta(u|t,p,q) > 0$ and $V_k(f) = \frac{2(\delta+1)k-1}{2}$ otherwise. We have*

$$M_k(f, \alpha, \beta, \gamma |\, h)(x) = \frac{J_h(\alpha,\beta,\gamma) \cdot W(\alpha,\beta,f)}{(1-\rho x)^{V_k(f)}} + O\left(\frac{1}{(1-\rho x)^{V_k(f)-\frac{1}{2}}}\right)$$

where $\rho = e^{-\alpha} + e^{-\beta} + 2e^{-\frac{\alpha}{2}}$, $J_h(\alpha,\beta,\gamma)$ is a constant independent of f and $W(\alpha,\beta,f)$ is the same as in Theorem 32.

Proof. This proof proceeds very similarly to the proof of Theorem 32. Thus we will only describe how the proof of this theorem differs from the latter proof.

Restricting the construction in the latter theorem to tree of root degree at most h, from $(Equation\ 3.69)$, when $i = 0$,

$$R_{i,h}(T_s, x_i, a_i, b_i, c) = \frac{1}{m_i!} \cdot \frac{\partial G_h(x_0, a_0, b_0, c)}{\partial a_0^{m_0}}$$

where $G_h(x_0, a_0, b_0, c)$ is the generating function for tree with root degree at most h where the tree are weighted by number of edges, leaves, internal nodes and root degree. Adapting Theorem 11, we see that

$$\begin{aligned} G_h(x_0, a_0, b_0, c) &= \sum_{r=0}^{h} [cG^*(x_0, a_0, b_0)]^r \\ &= \sum_{r=0}^{h} c^r \left[\left(\frac{1 + (a_0 - b_0)x_0}{2(1 - (b_0 - 1)x_0)} \right)^r - r \left(\frac{1 + (a_0 - b_0)x_0}{2(1 - (b_0 - 1)x_0)} \right)^{r-1} \Psi^{\frac{1}{2}} \right] + O(\Psi). \end{aligned} \tag{3.106}$$

Recall that $m_0 \geq 1$ (since for $k \geq 0$, the root of a tree with k edges has degree at least 1). Thus for, $i = 0$,

$$R_{i,h}(T_s, x_i, a_i, b_i, c) = R_i(T_s, x_i, a_i, b_i, c) \cdot W_h(x_i, a_i, b_i, c) + O\left(\Psi^{\frac{2-2m_i}{2}}\right) \tag{3.107}$$

where $W_h(x_i, a_i, b_i, c) = \sum_{r=0}^{h} c^r \cdot r \cdot 2x_i \left(\frac{1+(a_i-b_i)x_i}{2(1-(b_i-1)x_i)} \right)^{r-1}$ and $R_i(T_s, x_i, a_i, b_i, c)$ is as in (Equation 3.74). Thus, we adapt Lemma 34 to get that for $i = 0$,

$$\begin{aligned} \mathcal{D}_{x_i}^{u_x} \mathcal{D}_{a_i}^{u_a} \mathcal{D}_{b_i}^{u_b} \left[a_i^{m_i} R_{i,h}(T_s, x_i, a_i, b_i, c_i) \right] &= W_h(x_i, a_i, b_i, c) \cdot \mathcal{D}_{x_i}^{u_x} \mathcal{D}_{a_i}^{u_a} \mathcal{D}_{b_i}^{u_b} \left[a_i^{m_i} R_i(T_s, x_i, a_i, b_i, c_i) \right] \\ &\quad + O\left(p_i(x_i) \cdot \Psi_i^{\frac{2-2(u+m_i)}{2}} \right), \end{aligned} \tag{3.108}$$

where $u_x, u_a, u_b, \in \mathbb{Z}_{\geq 0}$ and $u = u_x + u_a + u_b$.

From this point, it should be clear that

$$M_k(f,\alpha,\beta,\gamma|\,h)(x) = W_h(x,e^{-\alpha},e^{-\beta},e^{-\gamma})\cdot M_k(f,\alpha,\beta,\gamma)(x) + O\left((1-\rho x)^{\frac{1}{2}-V_k(f)}\right). \tag{3.109}$$

We see that $W_h(z,e^{-\alpha},e^{-\beta},e^{-\gamma})$ is analytic in the disk $R=\left\{z\in\mathbb{C}:|z|\le\frac{1}{\rho}\right\}$ (since $W_h(x,e^{-\alpha},e^{-\beta},e^{-\gamma})$ is a finite sum). Thus

$$M_k(f,\alpha,\beta,\gamma|\,h)(x) = W_h\left(\frac{1}{\rho},e^{-\alpha},e^{-\beta},e^{-\gamma}\right)\cdot M_k(f,\alpha,\beta,\gamma)(x) + O\left((1-\rho x)^{\frac{1}{2}-V_k(f)}\right). \tag{3.110}$$

which is the desired result.

□

Let $\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)$ be defined similarly to $\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}_n)$ conditioned on the tree chosen having root degree at most h. From the above theorem, we can conclude the following.

Theorem 42. *Let $\alpha,\beta,\gamma\in\mathbb{R}$ and $f\in\mathbb{R}[t,p,q][n,\overline{p},\overline{q}]$. Assume every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q})=\delta$ is such that $\Delta(u|t,n)=\delta_n$, $\Delta(u|p,\overline{p})=\delta_{d_0}$ and $\Delta(u|q,\overline{q})=\delta_{d_1}$ where δ_n, δ_{d_0} and δ_{d_1} are constants that depend only on f. Let $V'(f)=\frac{2\delta+1}{2}$ if for every monomial of f, u, such that $\Delta(u|t,n,p,\overline{p},q,\overline{q})=\delta$, $\Delta(u|t,p,q)>0$ and $V'(f)=\delta+1$ otherwise. We have*

$$\frac{\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{n^{V'(f)}} \overset{d}{\longleftrightarrow} \frac{\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}} \overset{d}{\longleftrightarrow} \frac{Q(f,\alpha,\beta)\cdot\mathcal{P}^f_{(0,0,0)}(\mathfrak{T}_n)}{n^{V'(f)}}$$

where, for $V'(f)=\frac{2\delta+1}{2}$,

$$Q(f,\alpha,\beta) = 2^{\delta_{d_0}+2\delta_{d_1}}\cdot\frac{(e^{-\frac{\alpha}{2}}+1)^{\delta_{d_0}-1}\cdot e^{-\frac{\alpha}{4}(2\delta_{d_0}-1)}\cdot e^{-\beta\delta_{d_1}}}{\sqrt{\rho^{2\delta_{d_0}+2\delta_{d_1}-1}}}$$

and, for $V'(f) = \delta + 1$*,*

$$Q(f,\alpha,\beta) = 2^{\delta_{d_0}+2\delta_{d_1}} \cdot \frac{(e^{-\frac{\alpha}{2}}+1)^{\delta_{d_0}} \cdot e^{-\frac{\alpha\delta_{d_0}}{2}} \cdot e^{-\beta\delta_{d_1}}}{\rho^{\delta_{d_0}+\delta_{d_1}}}.$$

Proof. We notice that $M_{k,n}(1,\alpha,\beta,\gamma) = n^k \mathcal{Z}_{(n,\alpha,\beta,\gamma)}$. For a random variable X, let $\mathrm{Mom}_k(X)$ the kth moment of X. Fixed $n \in \mathbb{Z}_{\geq 0}$. By definition,

$$\mathrm{Mom}_k(\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)) = \frac{M_{k,n}(f,\alpha,\beta,0)}{\mathcal{Z}_{(n,\alpha,\beta,0)}} = \frac{n^k \cdot M_{k,n}(f,\alpha,\beta,0)}{M_{k,n}(1,\alpha,\beta,0)}$$

We now apply Theorem 32 and use the standard expression for the asymptotic coefficients of the Taylor series expansion of $(1-x)^k$ for $k \in \mathbb{C}$. We use the Transfer Theorem from [34], to get

$$M_{k,n}(f,\alpha,\beta,0) = \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta) \cdot P_k(f) \cdot n^{V_k(f)-1} \cdot (e^{-\alpha}+e^{-\beta}+2e^{-\frac{\alpha}{2}})^n}{\Gamma(V_k(f))}$$

where we only consider the most significant terms.

Notice that $V_k(1) = \frac{2k-1}{2}$ and every maximal monomial of 1 is independent of t, p, q. Thus

$$\mathrm{Mom}_k(\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)) = \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta) \cdot P_k(f) \cdot \Gamma(V_k(1))}{Q_k(1,0,0,\alpha,\beta) \cdot P_k(1) \cdot \Gamma(V_k(f))} \cdot n^{V_k(f)+\frac{1}{2}} + O\left(n^{V_k(f)}\right)$$

Notice that $V'(f)k = V_k(f) + \frac{1}{2}$. Thus

$$\lim_{n\to\infty} \mathrm{Mom}_k\left(\frac{\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}}\right) = \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta) \cdot P_k(f) \cdot \Gamma(V_k(1))}{Q_k(1,0,0,\alpha,\beta) \cdot P_k(1) \cdot \Gamma(V_k(f))}.$$

We now see that

$$\mathrm{Mom}_k(\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)) = \frac{n^k \cdot M_{k,n}(f,\alpha,\beta,\gamma|h)}{M_{k,n}(1,\alpha,\beta,\gamma|h)}$$

Using Theorem 41 and nearly identical computation to above, we see that

$$\lim_{n\to\infty} \mathbf{Mom}_k \left(\frac{\mathcal{P}^f_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{n^{V'(f)}} \right) = \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)\cdot P_k(f)\cdot \Gamma(V_k(1))}{Q_k(1,0,0,\alpha,\beta)\cdot P_k(1)\cdot \Gamma(V_k(f))}.$$

Finally, we set that

$$\begin{aligned}
\lim_{n\to\infty} \mathbf{Mom}_k \left(\frac{\mathcal{P}^f_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{n^{V'(f)}} \right) &= \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)\cdot P_k(f)\cdot \Gamma(V_k(1))}{Q_k(1,0,0,\alpha,\beta)\cdot P_k(1)\cdot \Gamma(V_k(f))} \\
&= \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)\cdot Q_k(1,0,0,0,0)}{Q_k(f,\delta_{d_0},\delta_{d_1},0,0)\cdot Q_k(1,0,0,\alpha,\beta)} \\
&\qquad \times \lim_{n\to\infty} \mathbf{Mom}_k \left(\frac{\mathcal{P}^f_{(0,0,0)}(\mathfrak{T}_n)}{n^{V'(f)}} \right) \\
&= \lim_{n\to\infty} \mathbf{Mom}_k \left(Q(f,\alpha,\beta)\cdot \frac{\mathcal{P}^f_{(0,0,0)}(\mathfrak{T}_n)}{n^{V'(f)}} \right)
\end{aligned}$$

where $Q(f,\alpha,\beta) = \sqrt[k]{\frac{Q_k(1,0,0,0,0)}{Q_k(f,\delta_{d_0},\delta_{d_1},0,0)} \cdot \frac{Q_k(f,\delta_{d_0},\delta_{d_1},\alpha,\beta)}{Q_k(1,0,0,\alpha,\beta)}}$ is a constant independent of k.

□

Corollary 43. *Let $\alpha,\beta,\gamma\in\mathbb{R}$. For any $h\in\mathbb{N}$, we have*

$$\frac{\mathcal{P}^{PL}_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{\sqrt{2n^3}} \quad \overset{d}{\longleftrightarrow} \quad \frac{\mathcal{P}^{PL}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{2n^3}} \quad \overset{d}{\to} \quad \frac{\sqrt{\rho}}{(e^{-\frac{\alpha}{2}}+1)\cdot e^{-\frac{\alpha}{4}}}\int_0^1 e(t)\,dt$$

where $e(t)$ is a normalized Brownian excursion on $[0,1]$.

Proof. We take $f=t+1$ then apply Theorem $Theorem$ 42 and Theorem 5.

□

Corollary 44. *Let $\alpha,\beta,\gamma\in\mathbb{R}$. For any $h\in\mathbb{N}$, we have*

$$\frac{\mathcal{P}^{WI}_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{\sqrt{2n^5}} \overset{d}{\longleftrightarrow} \frac{\mathcal{P}^{WI}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{2n^5}}$$
$$\overset{d}{\to} \frac{\sqrt{\rho}}{(e^{-\frac{\alpha}{2}}+1)\cdot e^{-\frac{\alpha}{4}}} \int\int_{0<s<t<1} (e(s)+e(t)-2\min_{s\leq u\leq t} e(u))\, ds\, dt$$

where $e(t)$ is a normalized Brownian excursion on $[0,1]$.

Proof. We take $f=(t+1)(n-t)$ then apply Theorem $Theorem\ 42$ and Theorem 7.

□

Corollary 45. *Let $\alpha,\beta,\gamma\in\mathbb{R}$. For any $h\in\mathbb{N}$, we have*

$$\frac{\mathcal{P}^{LR}_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{\sqrt{2n^3}} \overset{d}{\longleftrightarrow} \frac{\mathcal{P}^{LR}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{2n^3}} \overset{d}{\to} \frac{e^{-\frac{\alpha}{4}}}{\sqrt{\rho}}\int_0^1 e(t)\, dt$$

and

$$\frac{\mathcal{P}^{IR}_{(\alpha,\beta,\gamma)}(\mathfrak{T}^h_n)}{\sqrt{2n^3}} \overset{d}{\longleftrightarrow} \frac{\mathcal{P}^{IR}_{(\alpha,\beta,0)}(\mathfrak{T}_n)}{\sqrt{2n^3}} \overset{d}{\to} \frac{e^{-\beta}}{\sqrt{\rho}\cdot e^{-\frac{\alpha}{4}}\cdot(e^{-\frac{\alpha}{2}}+1)}\int_0^1 e(t)\, dt$$

where $e(t)$ is a normalized Brownian excursion on $[0,1]$.

Proof. For $\mathcal{P}^{LR}$, we take $f=p$. For $\mathcal{P}^{IR}$, we take $f=q$. We then apply Theorem $Theorem\ 42$ and Corollary 31.

□

3.4 Other Results

3.4.1 Counting trees by leaves, internal nodes and root degree

In the study of the Nearest Neighbour Thermodynamic Model, the partition function, $\mathcal{Z}_{(n,\alpha,\beta,\gamma)}$, is of general interest. Thus, we provide enumeration results to assist with the computation this quantity. We define $\mathcal{T}_n(m, k, r)$ to be the number of trees on n vertices with m internal nodes, k leaves and root degree r. We will omit any of the above parameters in $\mathcal{T}_n(m, k, r)$ to denote the number of trees summed over the omitted parameters.

It is well-known that $\mathcal{T}_n(k)$ is given by the Narayana numbers [92, 93]. Hence,

Theorem 46. *For $n \geq k$,*

$$\mathcal{T}_n(k) = \frac{1}{n}\binom{n}{k}\binom{n}{k-1}.$$

From the work of Dershowitz and Zaks [94], we also know that

Theorem 47. *For $n \geq r$,*

$$\mathcal{T}_n(r) = \frac{r}{n}\binom{2n-1-r}{n-1}.$$

From the work of Donaghey and Shapiro [95], we know that the number of tree on n edges with no internal nodes is given by the $(n-1)$th Motzkin number. Hence, we have the following result.

Theorem 48. *For $n > m$,*

$$\mathcal{T}_n(m) = \binom{n-1}{m} M_{n-m-1},$$

where

$$M_n = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2k} C_k$$

are the Motzkin numbers.

The main result of this section is the closed form expression for $\mathcal{T}_n(m, k, r)$ given in the following theorem.

Theorem 49. *For $n - m > k > r$,*

$$\mathcal{T}_n(m, k, r) = \frac{r}{n}\binom{n}{k+m}\binom{k+m}{k}\binom{k-r-1}{n-m-k-1},$$

and for $n - m = k = r$,

$$\mathcal{T}_n(m, k, r) = \binom{n-1}{m}.$$

Before proving this theorem, we first observe the following useful lemmas.

Lemma 50. *The number of trees with root degree r, m internal nodes and k leaves is equal to the number of trees with leftmost path with r edge, m leaves with a left sibling and $n + 1 - k$ leaves in total.*

Proof. We briefly describe the following bijection from $\mathfrak{T}_n$ to $\mathfrak{T}_n$ given by Dershowitz and Zaks [94], which we will denote as ψ. We describe ψ recursively. For $T = T_1 \ltimes T_2 \in \mathfrak{T}_{\geq 1}$, $\psi(T) = \psi(T_2) \ltimes \psi(T_1)$ and $\psi(T^*) = T^*$. (See top left of Figure 3.2 for illustration.)

Fix $T \in \mathfrak{T}_n$. We denote a leaf edge to be the edge incident to a leaf. Notice that leaf edges in T become edges without a right sibling in $\psi(T)$ (the rightmost edge under each vertex). Notice that the number of rightmost edges in a tree is the number of vertices with at least 1 child, hence the number of non-leaf vertices. To describe the effect of ψ on internal node, for an internal node, v, of T, we denote its parent edge and child edge by $p(v)$ and $c(v)$. Since we assume the root cannot be internal, a vertex is an internal node if and only if it has a parent edge and a child edge which has no siblings. Notice that the edge $c(v)$ and $p(v)$ in T becomes a leaf in $\psi(T)$ and a left sibling of that leaf respectively. Furthermore, every leaf and left sibling pair induce an internal node. Finally notice that the root degree of T is the length of the path from the root to the leaf reached by moving down leftmost edges. (See Figure 3.2 for illustration.)

□

Lemma 51. *The number of trees with root degree r, no internal nodes and k leaves is equal to the number of balanced parenthesis sequences with an initial run of r opening parenthesis with $n + 1 - k$ occurrences of $'()'$ with the restriction that no $'()'$ is preceded by a closing parenthesis.*

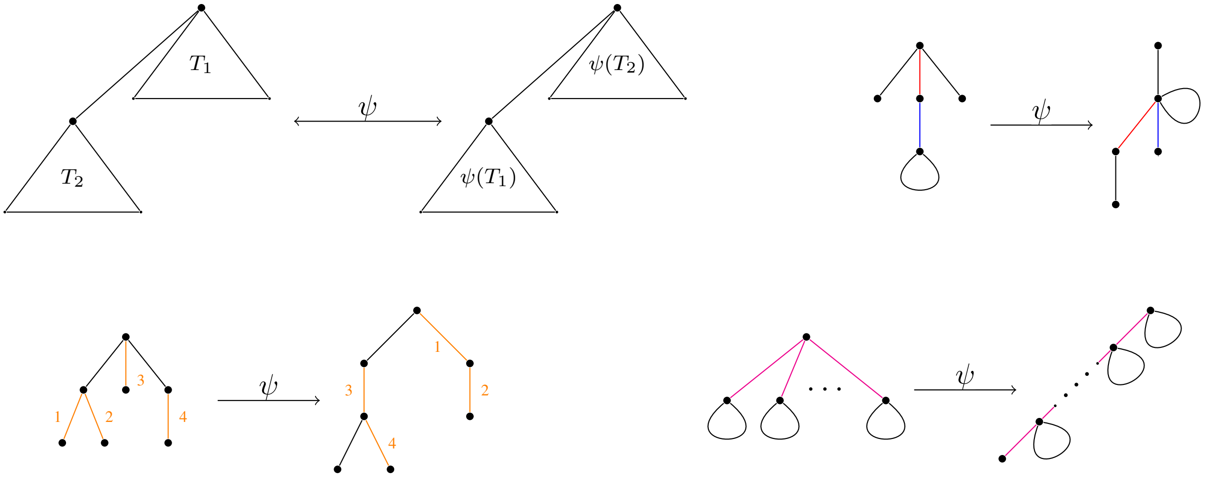


Figure 3.2: Illustrations of the effects of map ψ from plane trees to plane trees. The top left diagram illustrates the recurrence of ψ. The top right image illustrates what happens to internal nodes under ψ. The red and blue egdes are the parent and child edges of the internal node (and their new position under ψ). The bottom left image illustrates what happens to leaves under ψ. The numbered orange edges are leaves (and their new position under ψ). The bottom right image illustrates what happens to the root edges under ψ. The purple edges are root edges (and their new position under ψ).

Proof. We consider the standard bijection from plane trees to balances parenthesis sequences, denoted ψ_2. Fix $T \in \mathfrak{T}_n$. The length of the leftmost path of T is the length of the initial run of opening parenthesis of $\psi_2(T)$. A leaf in T corresponds to the sub-sequence $'()'$ in $\psi_2(T)$. A leaf of T with a left sibling corresponds to the sub-sequence $')()'$ in $\psi_2(T)$. Thus for T to have no leaves with left sibling, we must avoid the sub-sequence $')()'$. Applying Lemma 50, we achieve the result.

□

Lemma 52 (Cycle Lemma [96]). *For any sequence $p_0 p_1 \cdots p_{m+n-1}$ of m open parentheses and n close parentheses, where $m > n$, there exist exactly $m - n$ cyclic permutations*

$$p_i p_{i+1} \cdots p_{m+n-1} p_0 \cdots p_{i-1}$$

such that the number of opening parenthesis is always greater than the number of closing parenthesis. We will say these sequences are dominating.

We adapt the Cycle Lemma to aid the computation that will following.

Corollary 53. *For any sequence $p = p_0p_1 \cdots p_{m+n-1}$ of m open parentheses and n close parentheses, where $m > n$, among all distinct cyclic permutation of p, the ratio between the number of dominating sequences and total number of such permutations is $\frac{m-n}{m+n}$.*

Proof. We will denote the permutation $p_ip_{i+1} \cdots p_{m+n-1}p_0 \cdots p_{i-1}$ by $p(i)$ (where i is modulo $m + n$). Let $k \leq m + n$ be smallest natural number such that $p(0) = p(k)$. Notice that $p(0) = p(sk)$ for all $s \in \mathbb{Z}$.

We now show that for any $i, j \in \mathbb{Z}$, $p(i) = p(j)$ if and only if $i \equiv j \mod k$. It should be clear that $p(i) = p(i + sk)$ for all $s, i \in \mathbb{Z}$, since $p(0) = p(sk)$ and shifting $p(0)$ and $p(sk)$ by i, we get $p(i)$ and $p(i + sk)$, respectively. Alternatively, if we assume $i \not\equiv j \mod k$ (thus $i = j + sk + r$ where $0 < r < k$) and $p(i) = p(j)$, we see that $p(i) = p(j) = p(i + sk + r) = p(i + r)$. Shifting $p(i)$ and $p(i + r)$ by $-i$, we get that $p(0) = p(r)$, a contraction to the minimality of k.

Since $p(0) = p(m + n)$, k divides $m + n$. We thus see that the set of distinct cyclic permutations of p is $P = \{p(i) : 0 \leq i < k\}$ and each such permutation occurs $l = \frac{m+n}{k}$ times as a cyclic permutation of p. Let q be the number of element of P that are dominating sequences. By the Cycle Lemma, $\frac{m-n}{m+n} = \frac{lq}{lk} = \frac{q}{k}$, proving the result.

□

Proof of Theorem 49. We count the number of trees with root degree r, no internal nodes and k leaves using the bijection in Lemma 51. Let $l = n + 1 - k$. The beginning of all such balanced parenthesis sequences is fixed as a run of r opening parenthesis followed by a closing parenthesis. So we need to count the number of valid suffixes with $n - r$ opening parenthesis, $n-1$ closing parenthesis and $l-1$ occurrences of $'(()'$ and no other occurrences of $'()'$ or symmetrically, the number of valid prefixes with $n - 1$ opening parenthesis, $n - r$ closing parenthesis and $l - 1$ occurrences of $'())'$ and no other occurrences of $'()'$.

We count this via the cycle lemma. We count the number of dominating sequence with n opening parenthesis, $n-r$ closing parenthesis and $k'-1$ occurrences of $'())'$ and no other occurrences of $'()'$. We first count all possible cyclic permutations of all such sequences:

1. *Starting with* $'('$ *and ending in* $')'$.

   All such sequence are achieved by partitioning the n opening parenthesis into $l-1$ (ordered) parts of size 1 or more, denotes $x_1, \cdots, x_{l-1}$ and partitioning the $n-r$ closing parenthesis into $l-1$ parts of size 2 or more (since we want to guarantee that every $'()'$ is proceeded by a closing parenthesis), denotes $y_1, \cdots, y_{l-1}$. The formed sequence is then $x_1 y_1 \cdots x_{l-1} y_{l-1}$. Thus the number of sequences is $\binom{n-1}{l-2}\binom{n-r-l}{l-2}$.

2. *Starting with* $'('$ *and ending in* $'('$.

   Similarly to (1) with the extra step of adding another run of opening parenthesis of size at least 1 at the end of the sequence. These are sequences of the form $x_1 y_1 \cdots x_{l-1} y_{l-1} x_l$. The number of such sequences is $\binom{n-1}{l-1}\binom{n-r-l}{l-2}$.

3. *Starting with* $')'$ *and ending in* $'('$.

   We notice that we lose an occurrence of $'()'$ since it must be formed by the last opening parenthesis and the first closing parenthesis (in a cyclic shift). These sequences are similar to the sequences in (1) shifted so they are of the form $y_{l-1} x_1 y_1 \cdots x_{l-1}$. Thus the number of such sequences is $\binom{n-1}{l-2}\binom{n-r-l}{l-2}$.

4. *Starting with* $')'$ *and ending in* $')'$.

   These are sequences of the form $y_0 x_1 y_1 \cdots x_{l-1} y_{l-1}$. Since, we are counting cyclic permutations of dominating sequences of the desired type, the restriction of each run of closing parenthesis sequences being of size 2 or more is loosen for the first and last runs (since in any cyclic permutation that would lead to a dominating sequences, these 2 runs will become 1 run). Thus the number of sequences is $\binom{n-1}{l-2}\binom{n-r-l+1}{l-1}$.

Notice that we counted all the *distinct* cyclic permutations of the desired dominating sequences. Using Corollary 53, for any sequence counted above, in the set of its distinct cyclic permutations, $\frac{r}{2n-r}$ of members of the set are valid dominating sequences. Thus the number of tree with n edges, root degree of r and k leaves and no internal nodes is the sum of the terms derived for each of the above cases scaled by the term $\frac{r}{2n-r}$. After simplifying and setting $l = n+1-k$, we get this to be

$$\frac{r}{n}\binom{n}{k}\binom{k-r-1}{n-k-1}.$$

Adding internal nodes is now just a matter of deciding how many to put under each edge. Thus for each tree on $n-m$ edges with root degree of r and k leaves and no internal nodes, there are $\binom{n-1}{m}$ trees with n edges with root degree of r and k leaves and m internal nodes giving us the desired expression (after simplification).

Corollary 54. *For $n-m > k$,*

$$\mathcal{T}_n(m,k) = \frac{1}{n}\binom{n}{k+m}\binom{k+m}{k}\binom{k}{n-m-k+1},$$

and for $n-m = k$,

$$\mathcal{T}_n(m,k) = \binom{n-1}{m}.$$

Proof. Let $p, q \in \mathbb{N}$. We now evaluate the following sum.

$$\sum_{q\geq 0}\sum_{r\geq 0} r\binom{q-r}{p}x^q = x\sum_{r\geq 0} r x^{r-1}\sum_{q\geq 0}\binom{q-r}{p}x^{q-r} \tag{3.111}$$

$$= \frac{x^{p+1}}{(1-x)^{p+3}} = \sum_{q\geq 0}\binom{q+1}{p+2}x^q. \tag{3.112}$$

Thus we get

$$\sum_{r\geq 0} r\binom{q-r}{p} = \binom{q+1}{p+2}.$$

We now set $q = k - 1$ and $p = n - m - k - 1$ to achieve the desired result.

□

Corollary 55. *For $n > k > r$,*

$$\mathcal{T}_n(k, r) = \frac{r}{n}\binom{n}{k}\binom{n-r-1}{n-k-1},$$

and for $n = k = r$,

$$\mathcal{T}_n(k, r) = 1.$$

Proof. Let $p, q \in \mathbb{N}$. We now evaluate the following sum.

$$\begin{aligned} \sum_{m\geq 0}\sum_{p\geq 0}\binom{n-k}{m}\binom{q}{p-m}x^p &= \sum_{m\geq 0}\binom{n-k}{m}x^m\sum_{p\geq m}\binom{q}{p-m}x^{p-m} && (3.113)\\ &= (1+x)^{n-k+q} = \sum_{p\geq 0}\binom{n-k+q}{p}x^p. && (3.114)\end{aligned}$$

Thus we get

$$\sum_{m\geq 0}\frac{r}{n}\binom{n}{k+m}\binom{k+m}{k}\binom{q}{p-m} = \frac{r}{n}\binom{n}{k}\sum_{m\geq 0}\binom{n-k}{m}\binom{q}{p-m} = \frac{r}{n}\binom{n}{k}\binom{n-k+q}{p}.$$

We now set $q = k - r - 1$ and $p = n - k - 1$ to achieve the desired result.

□

3.5 Discussion

We have characterized the asymptotic behavior of many plane tree properties under a probability distribution where the weight of each tree depends on its number of leaves, number of internal nodes, and root degree. We have shown that, in the case where the root degree is bounded, the distribution of any subtree additive property under parameters (α, β, γ)

is simply constant multiple of the distribution of that same property under the parameter set $(0,0,0)$, where the constant depends on the property in question and the parameters (α,β,γ). The probability distribution and tree properties studied were inspired by questions from molecular biology, specifically RNA secondary structure. Of interest in the biological context, we have shown that the asymptotic distributions of total contact distance, total ladder distance, total leaf to root distance, and total internal node to root distance depend on the parameters (α,β,γ) in a relatively simple way. The fact that these asymptotic distributions can vary only up to a constant multiplier suggests limitations in the Nearest Neighbor Thermodynamic Model. The explicit form of the scaling constant given in Theorem 42 may enable further insights about the exact role of the parameters α and β.

The results are also of independent mathematical interest, as they allow for the examination of a large set of plane tree properties under many natural probability distributions, and show that the behavior of these distributions when changing the values of (α,β,γ) is actually quite simple. In particular, this means that, in order to understand the asymptotic distribution of a plane tree property where the plane trees are weighted according to a specific (α,β,γ), it is sufficient to understand the behavior of the property under parameters $(0,0,0)$. Since the $(0,0,0)$ case can often be determined through combinatorial techniques (or is already known), these theorems may be useful in studying combinatorial properties of plane trees under different probability distributions.

We conclude with a few open questions.

1. In Theorem 17, we characterize the asymptotic behavior of simple subtree additive properties with bounded toll functions. What (if anything) can be proved about the asymptotic behavior simple subtree additive properties with an unbounded toll function?

2. What is the asymptotic behavior of the distributions of subtree additive properties if we remove the restriction that the root degree of the tree must be bounded? Is a generalization of Theorem 42 possible?

3. The *maximum ladder distance* of a plane tree is the length of the longest path in the tree, a.k.a. its diameter. Maximum ladder distance is not a subtree additive property, but it is of interest in the molecular biology context. What is the asymptotic distribution of maximum ladder distance? The framework constructed here does not seem to address properties of this type, and we suspect an entirely different approach would be necessary to answer this question.

CHAPTER 4
MARKOV CHAIN-BASED SAMPLING FOR EXPLORING RNA SECONDARY STRUCTURE UNDER THE NEAREST NEIGHBOR THERMODYNAMIC MODEL AND EXTENDED APPLICATIONS

The content of this chapter has been published in the journal Mathematical and Computational Applications, with co-authors Cassie Mitchell, Chidozie Onyeze, and Prasad Tetali [85].

4.1 Introduction

Computational and mathematical applications play a critical role in the analysis of the structure and function of biological molecules, including ribonucleic acid (RNA). RNA is an essential biological polymer with many roles including information transfer, regulation of gene expression, and catalysis of chemical reactions. The *primary structure* of an RNA molecule may be understood as a sequence of amino acids: arginine, urasil, guanine, and cytosine. As is standard, we frequently abbreviate these as A, U, G, and C, respectively. RNA molecules are single-stranded and may therefore interact with themselves, forming A-U, G-U, and G-C bonds. The *secondary structure* of an RNA molecule is a set of such bonds.

The determination of secondary structure is an important step to understanding an RNA molecule's full shape and therefore its function [11, 10]. Accordingly, secondary structure information is commonly used in tertiary structure prediction algorithms, see e.g. [74, 75, 76, 77]. Identifying the secondary structure of RNA is crucial to understanding its function and mechanism in a cell [9]. Thus, the structure of RNA is critical to the development of biological and pharmaceutical therapeutics. Biologists use inexpensive and expedient means to sequence RNA, but experimental determination of structure is more difficult and time-consuming. Therefore, computational methods are the primary means to determine possible RNA secondary structures.

For decades, one of the main computational approaches for examining RNA structure and branching properties has been thermodynamic free energy minimization using Nearest Neighbor Thermodynamics Modeling (NNTM) [78, 79, 80]. This free energy is in turn used in algorithms to predict secondary structure given an RNA sequence, see, e.g., [81, 82, 83]. Under the NNTM, the free energy of a structure is computed as the sum of the free energy of its various substructures. Many common programs (e.g. mFold, RNAFold, RNA Structure, sFold, Vienna RNA, etc.) intake a single sequence to produce secondary structures based on NNTM energy minimizations performed via dynamic programming. Nearest neighbor parameter sets include both a set of rules, referred to as equations or features, and a set of parameter values used by the equations. Separate rules exist for predicting stabilities of helices, hairpin loops, small internal loops, large internal loops, bulge loops, multi-branch loops, and exterior loops. Other branching properties of interest include, but are not limited to, average ladder distance, maximum ladder distance, maximum branching degree, average contact distance, average branching degree, degree of branching at the exterior loop, number of multi-loops with n braches, etc. The online NNDB (nearest neighbor database) archives and stores complete nearest neighbor sets, including rules and corresponding parameter values [97].

A common challenge is inferring whether the predicted results of NNTM for a set of RNA structural features or branching properties are within expected dispersion thresholds for a given energy model. For example, is the number of hairpins more than 2-3 standard deviations greater than the expected mean for a given energy model? This challenge is particularly vexing if the sequence is relatively long (greater than 1000 nucleotides). If structural features or branching properties are determined to exceed expected energy model dispersion thresholds, it relays potential scientific and/or mechanistic insight. Continuing with our hairpin example, what if an NNTM model produces a result where the number of hairpins seems rather large for the given sequence length? If the number of hairpins exceeds the expected dispersion of the NNTM model, it might be inferred that the greater number of hairpins is evidence of natural selection.

The primary objective of the present study is to enable mathematical determination of the dispersion of RNA secondary structural features for a given sequence length. We present a Markov-based algorithm to provide samples of the branching structure under the NNTM and Gibbs distribution, but without reference to a particular sequence of nucleotides. The algorithm enables the determination of where the predicted feature or branching property for an actual sequence falls within this distribution, which in turn enables the determination of whether the predicted NNTM feature or branching property is within expected dispersion limits.

In particular, this work investigates RNA substructures called *multi-loops*, the places where three or more helices join. Though multi-loops are crucial to the overall shape of a secondary structure, the models used to predict them algorithmically do not produce accurate results [84]. This investigation builds on an existing model of RNA branching [12] and provides a theoretical grounding for a Markov chain which may be used to algorithmically investigate branching properties of secondary structure models. The investigational foundation is a model for RNA secondary structure developed by Hower and Heitsch [12], in which secondary structures are in bijection with plane trees and the minimal energy structures of the model have been previously characterized. The present study characterizes the full Gibbs distribution of possible structures. Notably, Bakhtin and Heitsch [86] analyzed a very similar model and determined degree sequence properties of the distribution of plane trees asymptotically. However, the present study utilizes a Markov chain-based sampling algorithm to investigate the Gibbs distribution in the finite case. A full explanation of the plane tree model as well as the derivation of the energy functions is provided in subsection 4.2.1.

4.2 Methods

The methods are divided into an overview of the RNA secondary structure NNTM plane tree model and energy functions (subsection 4.2.1) and an all-encompassing explanation of the mathematical preliminaries that lay the foundation for the derived results and corresponding algorithms (subsection 4.2.2).

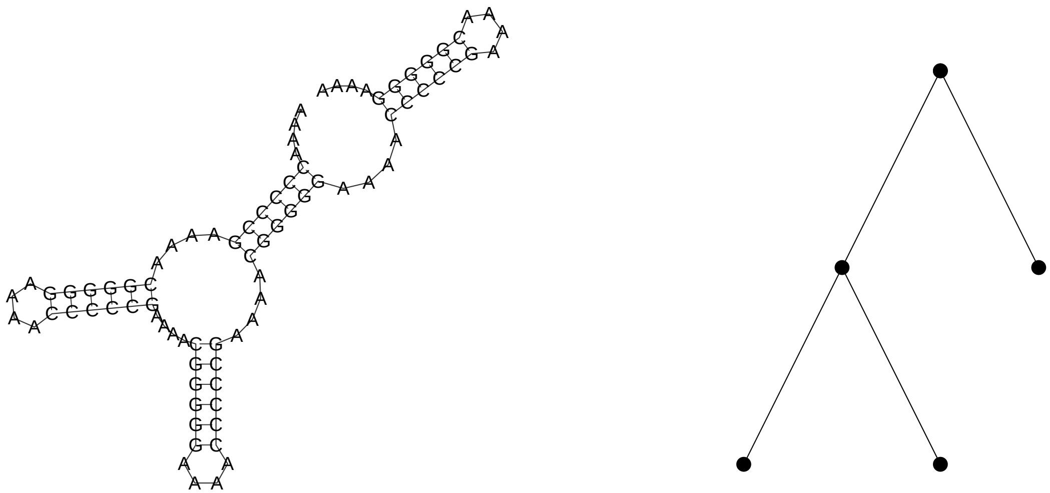
4.2.1 Derivation of Energy Functions

The energy function studied here is derived from the Nearest Neighbor Thermodynamic Model (NNTM). The numerical parameters from the NNTM can be found in the NNDB [97]. In calculating energy functions for the sequences, we consider thermodynamic parameter values published by Turner in 1989 [78], 1999 [79], and 2004 [80].

The plane trees that we study in this chapter come from two combinatorial RNA sequences, both of the form $A^4(Y^5ZA^4YZ^5A^4)^n$. The sequences of interest have $(Y, Z) = (C, G)$ or $(Y, Z) = (G, C)$. For both of these sequences, the set of maximally-paired secondary structures is in bijection with the set of plane trees of size n [98]. Figure 4.1 shows one example of a secondary structure and corresponding plane tree.

These specific combinatorial sequences are chosen because they allow for the study of the relationship between NNTM multiloop parameters and the branching behavior of secondary structures without interference from the energy contributions have specific base pairing combinations.. In particular, the only places where the free energy differs between different secondary structures (for the same sequence) is in the type and number of multiloops, the branching at the exterior loop, the number of hairpins, and the number of internal nodes. All of these energies directly relate to branching, not to specific base pairs. This simplification achieved by focusing only on multi-loops and branching both creates a model that is more amenable to theoretical analysis and speeds computation.

Note that these secondary structures should not be considered representative of natu-



(a) A maximally-paired secondary structure for $A^4(C^5GA^4CG^5A^4)^4$ has 4 helices.

(b) The corresponding plane tree has 4 edges and encodes the branching pattern seen in the secondary structure.

Figure 4.1: An RNA secondary structure for one of the combinatorial RNA sequences used in this work and its corresponding plane tree. The ordering of the edges in the plane tree is derived from the 3' to 5' ordering of the RNA sequence. Note that the exterior loop corresponds to the root of the plane tree. The diagram in Figure 4.1a was generated by ViennaRNA [99].

rally occurring secondary structures. Instead, the only properties of interest in these structures are branching-related properties.

Three constants determine the free energy contribution of multiloops under NNTM, a, b, and c. The value of a encodes the energy penalty per multiloop. The constant b specifies the energy penalty per single-stranded nucleotide in a multiloop. The value of c gives the energy penalty for each helix branching from a multiloop.

In addition to the multiloop parameters a, b, c discussed above, we must account for the energy contributions of stacking base pairs, hairpins, interior loops, and dangling energy contributions. The energy of one helix is given by h. The energy associated with a hairpin is f, and the energy contribution of an interior loop is i. Finally, the parameter g encodes the dangling energy contributions. All of these values can be computed directly from the parameters found in the NNTM.

We wish to compute the energy of the structure corresponding to plane tree t having (down) degree sequence $d_0, d_1, \ldots, d_{n-1}$ and root degree r. Note that the *down degree* of a node x is equal to the number of children of x, and, in the *down degree sequence*, d_i is the number of non-root nodes with exactly i children. The energy contribution of all hairpin loops will be $d_0 f$, and similarly the total energy of all interior loops will be $d_1 i$. For a multi-loop having down degree j, the energy contribution will be $a+4b(j+1)+c(j+1)+(j+1)g$, and so the contribution of all multi-loops is given by $\sum_{j=2}^{n} d_j(a + 4b(j + 1) + c(j + 1) + g(j + 1))$. The root vertex of the tree corresponds to the exterior loop and has energy contribution gr. Finally, our structure has n helices, each with energy h. Summing all of these components gives the total energy.

$$d_0 f + d_1 i + \sum_{j=2}^{n} d_j(a + 4b(j + 1) + c(j + 1) + g(j + 1)) + nh + gr \tag{4.1}$$

$$= (f - a - 4b - c - g)d_0 + (i - a - 8b - 2c - 2g)d_1 + (-4b - c)r$$
$$+ (a + 8b + 2c + h + 2g)n, \tag{4.2}$$

where we have used the facts $\sum_{k=0}^{n-1} d_k = n$ and $\sum_{k=0}^{n-1} kd_k = n - r$.

Set $\alpha = f - a - 4b - c - g$, $\beta = i - a - 8b - 2c - 2g$, $\gamma = -4b - c$, and $\delta = a + 8b + 2c + h + 2g$. Then, the energy function is $\alpha d_0 + \beta d_1 + \gamma r + \delta n$. Since n will be fixed, we disregard the term δn, giving

$$E(t) = \alpha d_0 + \beta d_1 + \gamma r. \tag{4.3}$$

Though we study these energy functions for arbitrary values of (α, β, γ), numerical values for both the input energy parameters from NNTM and the resulting energy function coefficients are given in Table 4.1.

Y	Z	Turner	a	b	c	h	f	i	g	α	β	γ
C	G	89	4.6	0.4	0.1	-10.9	3.8	3.0	-1.6	-0.9	-1.8	-1.7
G	C	89	4.6	0.4	0.1	-16.5	3.5	3.0	-1.9	-0.9	-1.2	-1.7
C	G	99	3.4	0	0.4	-12.9	4.5	2.3	-1.6	2.3	1.3	-0.4
G	C	99	3.4	0	0.4	-16.9	4.1	2.3	-1.9	2.2	1.9	-0.4
C	G	04	9.3	0	-0.9	-12.9	4.5	2.3	-1.1	-2.8	-3.0	0.9
G	C	04	9.3	0	-0.9	-16.9	4.1	2.3	-1.5	-2.8	-2.2	0.9

Table 4.1: NNTM parameters and resulting energy functions. Energy functions are of the form $\alpha d_0 + \beta d_1 + \gamma r$.

4.2.2 Mathematical Preliminaries

In subsection 4.2.2 we provide the necessary mathematical background, including a formal introduction of combinatorial objects and a review of the relevant Markov chain mixing results used to construct our resultant sampling Markov chain and corresponding mixing time proof in section 4.3.

Combinatorial Objects

A *plane tree* is a rooted, ordered tree. We will use $\mathfrak{T}_n$ to denote the set of plane trees with n edges. It is known that $|\mathfrak{T}_n|$ is given by the nth Catalan number $C_n = \frac{1}{n+1}\binom{2n}{n}$. In a plane tree, a *leaf* is a node with down degree 0, and an *internal node* is a non-root node with down degree 1. For a given plane tree t, we will use $d_0(t)$ to denote the number of leaves and $d_1(t)$ to denote the number of internal nodes.

For a plane tree t, the *energy* of the tree is given by

$$E(t) = \alpha d_0(t) + \beta d_1(t), \tag{4.4}$$

where α and β are real parameters of the energy function. Note that this function is a simplification of the model due to Hower and Heitsch [12] discussed in subsection 4.2.1. Making this simplification effectively disregards the energy contribution of the exterior loop, which is small in comparison to the total energy of a structure, especially for the longer sequences that are of interest to us. Other authors have made similar simplifications, e.g. [86].

For our purposes, we consider α and β to be arbitrary but fixed. We will consider a Gibbs distribution g on the set $\mathfrak{T}_n$, where the weight of each tree t is given by

$$\mathbf{g}(t) = \frac{e^{-E(t)}}{Z}, \tag{4.5}$$

where $Z = \sum_{y \in \mathfrak{T}_n} e^{-E(y)}$ is a normalizing constant.

A *Motzkin path* of length n is a lattice path from $(0, 0)$ to $(n, 0)$, which consists of steps along the vectors $U = (1, 1)$, $H = (1, 0)$, and $D = (1, -1)$ and never crosses below the x-axis. We can also represent Motzkin paths as strings from the alphabet $\{U, H, D\}$ where, in any prefix, the number of Us is greater than or equal to the number of Ds. The number of Motzkin paths of length n is given by the Motzkin numbers M_n where

$$M_n = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{2k} C_k. \tag{4.6}$$

Motzkin numbers and Motzkin paths have been well-studied in the combinatorics literature, see e.g. [100, 101, 102, 103, 104].

A *Dyck path* is a Motzkin path with no H steps. It is easy to see that a Dyck path must have even length, so we will use $\mathfrak{D}_n$ to denote the set of Dyck paths on length $2n$. It is well known that $|\mathfrak{D}_n| = C_n$ (see, e.g. [105]).

A *2-Motzkin path* is a Motzkin path in which $(1, 0)$ steps are given one of two distinguishable colors. Let $\mathfrak{M}^2_m$ be the set of all 2-Motzkin paths of length m. We can also represent 2-Motzkin paths as strings from the alphabet $\{U, H, I, D\}$, where as before, the number of Ds never exceeds the number of Us in any prefix. In a such a string x, we denote by $|x|_a$ the number of times the symbol a appears in x, where $a \in \{U, H, I, D\}$. Notice that we always have $|x|_U = |x|_D$. For any $x \in \mathfrak{M}^2_n$ and $k \in \{1, \cdots, n\}$, let $x(k)$ denote the symbol at index k in the string representation of x. Additionally, the *skeleton* of a 2-Motzkin path x is the Dyck path of Us and Ds which results from removing all Hs and Is from x. We will denote the skeleton of x by $\sigma(x)$.

A Bijection Between $\mathfrak{T}_n$ and $\mathfrak{M}^2_{n-1}$

We will use the particular bijection $\Phi\colon \mathfrak{T}_n \to \mathfrak{M}^2_{n-1}$ between plane trees and 2-Motzkin paths from Deutsch [106], which neatly encodes information about d_0 and d_1. For clarity, we will overview the bijection here.

For a given plane tree t with n edges, assign a label from the set $\{U, H, I, D\}$ to each edge e according to the following rules:

- If e is the leftmost edge off a non-root node of down degree at least 2, assign the label U.
- If e is the rightmost edge off a non-root node of down degree at least 2, assign the label D.
- If e is the only edge off a non-root node of degree 1, assign the label I.
- If e is an edge off the root node, or if e is neither the leftmost nor the rightmost edge off its parent node, assign the label H.

Now, if we traverse t in preorder reading off these labels, we get a 2-Motzkin path of length n. However, this path will always begin with H, so we define $\Phi(t)$ to be the 2-Motzkin path of length $n-1$ after this initial H is removed. Figure 4.2 gives an example of this labeling process. From Deutsch, we know not only that Φ is a bijection, but also that if $x = \Phi(t)$ then $|x|_I = d_1(t)$ and $|x|_U + |x|_H + 1 = d_0(t)$.

Using this bijection, it is natural to extend our energy function to 2-Motzkin paths. We define the energy of a 2-Motzkin path x to be

$$E(x) = \alpha(|x|_U + |x|_H + 1) + \beta|x|_I, \tag{4.7}$$

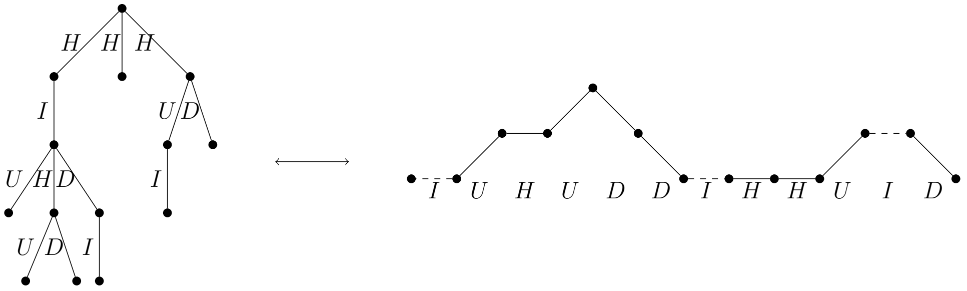


Figure 4.2: A plane tree with edges labeled according to the bijection Φ, along with its corresponding 2-Motzkin path.

and we extend our definition of the distribution g to $\mathfrak{M}_n^2$ accordingly. We note that, while this energy function does not capture all possible weightings on 2-Motzkin paths, it does capture all weightings possible under our simplification of the model due to Hower and Heitsch [12] after applying the bijection due to Deutsch [106].

Markov Chains

A *Markov chain* $\mathcal{M}$ is a sequence of random variables $X_0, X_1, X_2, \cdots$ taking values in a state space Ω subject to the condition that

$$\Pr(X_{t+1} = y \mid X_t = x,\, X_{t-1} = s_{t-1},\, \cdots,\, X_0 = s_0) = \Pr(X_{t+1} = y \mid X_t = x). \quad (4.8)$$

All Markov chains that we consider in this chapter will be implicitly *time-homogeneous* (meaning $\Pr(X_{t+1} = y \mid X_t = x)$ does not depend on t) and *finite* (meaning $|\Omega| < \infty$). The *transition matrix* of a time-homogeneous Markov chain is the matrix $P\colon \Omega \times \Omega \to [0,1]$ given by

$$P(x,y) = \Pr(X_{t+1} = y \mid X_t = x). \quad (4.9)$$

It is easy to see that if X_0 has distribution vector $\mathbf{x}$, then X_t has distribution vector $P^t\mathbf{x}$.

A finite Markov chain with transition matrix P is said to be *ergodic* if it has the following two properties.

1. *Irreducibility:* For any $x, y \in \Omega$, there is some integer $t \in \mathbb{N}$ for which $P^t(x,y) > 0$.

2. *Aperiodicity:* For any state $x \in \Omega$, we have $\gcd\{t \in \mathbb{N} \colon P^t(x,x) > 0\} = 1$.

It is well known that if $\mathcal{M}$ is ergodic, then there exists a unique distribution vector π, the *stationary distribution*, such that $P\pi = \pi$, and $\lim_{t\to\infty} P^t(x,y) = \pi(y)$ for any states $x, y \in \Omega$. Additionally, we call $\mathcal{M}$ *reversible* if for all states $x, y \in \Omega$, we have $\pi(x)P(x,y) = \pi(y)P(y,x)$.

For $\epsilon > 0$, the *mixing time* $\tau(\epsilon)$ of $\mathcal{M}$ is given by

$$\tau(\epsilon) = \min\left\{ t \in \mathbb{N} \colon \forall s \geq t,\ \max_{x\in\Omega} \left(\frac{1}{2} \sum_{y\in\Omega} |P^s(x,y) - \pi(y)| \right) < \epsilon \right\}. \tag{4.10}$$

Intuitively, the mixing time gives a measure of the number of steps required for $\mathcal{M}$ to get sufficiently close to its stationary distribution from any starting state.

Let $\mathcal{M}$ be a finite ergodic Markov chain over a state space Ω with transition matrix P. Let the eigenvalues of P be $\lambda_0, \lambda_1, \ldots, \lambda_{|\Omega|-1}$ such that $1 = \lambda_0 > |\lambda_1| \geq \ldots \geq |\lambda_{|\Omega|-1}|$. The *spectral gap* of $\mathcal{M}$ is given by $\mathrm{Gap}(\mathcal{M}) = 1 - |\lambda_1|$. As is standard, it will be convenient to denote the inverse of the spectral gap by *relaxation time* $\tau_{rel}(\mathcal{M}) := 1/\mathrm{Gap}(\mathcal{M})$.

Additionally, the spectral gap is given by the following functional definition [107].

$$\mathrm{Gap}(\mathcal{M}) = \inf_f \frac{\sum_{x,y\in\Omega} |f(x) - f(y)|^2 \pi(x) P(x,y)}{\sum_{x,y\in\Omega} |f(x) - f(y)|^2 \pi(x)\pi(y)}, \tag{4.11}$$

where the infimum is taken over all non-constant functions $f \colon \Omega \to \mathbb{R}$. A direct consequence of this definition of the spectral gap is the following lemma.

Lemma 56. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be ergodic Markov chains over Ω with the same stationary distribution. Let P_1 and P_2 be the transition matrices of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. If for all $x, y \in \Omega$ and for some constant $c > 0$ we have $P_1(x,y) \leq cP_2(x,y)$, then $\mathrm{Gap}(\mathcal{M}_1) \leq c\mathrm{Gap}(\mathcal{M}_2)$.*

Additionally, spectral gap is related to the mixing time by the following lemma [42].

Lemma 57. *Let $\mathcal{M}$ be an ergodic Markov chain with state space Ω, and let $\lambda_0, \lambda_1, \ldots, \lambda_{|\Omega|-1}$ be the eigenvalues of the transition matrix P as defined above. Then, for all $\epsilon > 0$ and $x \in \Omega$, we have*

$$\frac{|\lambda_1|}{\text{Gap}(\mathcal{M})} \log\left(\frac{1}{2\epsilon}\right) \leq \tau(\epsilon) \leq \frac{1}{\text{Gap}(\mathcal{M})} \log\left(\frac{1}{\pi(x)\epsilon}\right). \tag{4.12}$$

We say that a Markov chain $\mathcal{M}$, whose state space depends on a variable $n \in \mathbb{N}$, is *rapidly mixing* if $\tau(\epsilon)$ is bounded above by some polynomial in n and $\log(\epsilon^{-1})$. For the specific chains studied in this chapter, we will show that $\tau(\epsilon)(\mathcal{M})$ is bounded by a polynomial in n and $\log(\epsilon^{-1})$ if and only if $\tau_{rel}(\mathcal{M})$ is bounded by a polynomial in n and $log(\epsilon^{-1})$. Our next lemma presents sufficient conditions.

Lemma 58. *Let $\mathcal{M}$ be an ergodic Markov chain with state space Ω and let $\lambda_0, \lambda_1, \ldots, \lambda_{|\Omega|-1}$ be the eigenvalues of its transition matrix. Let $\epsilon > 0$. If $\tau(\epsilon)$ is bounded by a polynomial in n and $log(\epsilon^{-1})$, then τ_{rel} is also bounded by a polynomial in n and* $\log(\epsilon^{-1})$. *Further, suppose we have* $\log(1/\pi(x))$ *bounded by some polynomial $q(n)$ for all $x \in \Omega$. Then, $\tau_{rel}(\mathcal{M})$ being bounded by a polynomial in n and* $\log(\epsilon^{-1})$ *implies that $\tau(\epsilon)$ is also bounded by some polynomial in n and* $\log(\epsilon^{-1})$.

Proof. Suppose that $\tau(\epsilon) \leq p(n, \log(\epsilon^{-1}))$, where p is a polynomial. Beginning with the left hand side of Lemma 57, note that

$$\frac{|\lambda_1|}{1 - |\lambda_1|} \log\left(\frac{1}{2\epsilon}\right) = (\tau_{rel}(\mathcal{M}) - 1) \log\left(\frac{1}{2\epsilon}\right).$$

Then, applying Lemma 57 and the bound on $\tau(\epsilon)$,

$$\tau_{rel}(\mathcal{M}) \leq \frac{\tau(\epsilon)}{\log((2\epsilon)^{-1})} + 1 \leq \frac{p(n, \log(\epsilon^{-1}))}{\log((2\epsilon)^{-1})} + 1 \leq p'(n, \log(\epsilon^{-1})),$$

where p' is again a polynomial in n and $\log(\epsilon^{-1})$.

Turning now to converse, suppose that we have $\tau_{rel} \leq p(n, \log(\epsilon^{-1}))$, for some polynomial p. Additionally suppose $\log(1/\pi(x)) \leq q(n)$ for all $x \in \Omega$, for some polynomial q.

Applying Lemma 57,

$$\tau(\epsilon) \leq \tau_{rel}(\mathcal{M}) \log\left(\frac{1}{\pi(x)\epsilon}\right) \leq p(n, \log(\epsilon^{-1})) \log(\epsilon^{-1}) q(n) \leq p'(n, \log(\epsilon^{-1})),$$

where p' is some polynomial. □

Coupling

A *coupling* of a Markov chain $\mathcal{M}$ on Ω is a chain $(X_t, Y_t)_{t=0}^{\infty}$ on $\Omega \times \Omega$ for which the following properties hold.

1. Each chain $(X_t)_{t=0}^{\infty}$ and $(Y_t)_{t=0}^{\infty}$, when viewed in isolation, is a copy of $\mathcal{M}$, given initial states $X_0 = x$ and $Y_0 = y$.
2. Whenever $X_t = Y_t$, we have $X_{t+1} = Y_{t+1}$.

Formally, item one. above requires that the joint distribution of (X_t, Y_t) given (X_{t-1}, Y_{t-1}) should satisfy the property that the marginal of X_t (and also Y_t) is consistent with the probability transitions of $\mathcal{M}$. We define the *coupling time* T to be

$$T = \max_{x,y \in \Omega} \mathbb{E}\left[\min\{t \colon X_t = Y_t \mid X_0 = x,\, Y_0 = y\}\right] \tag{4.13}$$

The following lemma [108] is useful in bounding the coupling time T.

Lemma 59. *Suppose that $(X_t, Y_t)_{t=0}^{\infty}$ is a coupling of a Markov chain M. Let φ be an integer-valued distance function on $\Omega \times \Omega$ taking values in the range $[0, B]$, and suppose that $\varphi(x, y) = 0$ if and only if $x = y$. Let $\varphi(t) = \varphi(x_t, y_t)$. Suppose that the coupling satisfies $E\left(\varphi(t+1) - \varphi(t) | X_t, Y_t\right) \leq 0$. Additionally, suppose that whenever $\varphi(t) > 0$, $E\left(|\varphi(t+1) - \varphi(t)|^2 | X_t, Y_t\right) \geq V$. Then, the expected coupling time satisfies $E\left(T^{x,y}\right) \leq \varphi(0)(2B - \varphi(0))/V$.*

Coupling time and mixing time are then related by the following theorem [42].

Theorem 60. *A Markov chain M with coupling time T has mixing time $\tau(\epsilon)$ bounded by*

$$\tau(\epsilon) \leq \lceil Te \log \epsilon^{-1} \rceil. \tag{4.14}$$

Decomposition

We use two disjoint decomposition methods for bounding the spectral gap, one developed by Martin and Randall [43], and a very recent one given by Hermon and Salez [44], building on the work by Jerrum, Son, Tetali and Vigoda [109]. We use both theorems because, while the latter gives better bounds, the former has more relaxed conditions, which is necessary in one of our applications. The setup for both methods is the same.

Let $\mathcal{M}$ be an ergodic, reversible Markov chain over a state space Ω with transition matrix P and stationary distribution π. Suppose Ω can be partitioned into disjoint subsets $\Omega_1, \ldots, \Omega_m$. For each $i \in [m]$, let $\mathcal{M}_i$ be the *restriction* of $\mathcal{M}$ to Ω_i, which is obtained by rejecting any transition that would leave Ω_i. Let P_i be the transition matrix of $\mathcal{M}_i$ Additionally, we define $\overline{\mathcal{M}}$ to be the *projection chain* of $\mathcal{M}$ over the state space $[m]$ as follows. Let the transition matrix $\overline{P}$ of $\overline{\mathcal{M}}$ be given by

$$\overline{P}(i,j) = \frac{1}{\pi(\Omega_i)} \sum_{\substack{x \in \Omega_i \\ y \in \Omega_j}} \pi(x) P(x,y). \tag{4.15}$$

One can check that $\overline{\mathcal{M}}$ is reversible and has stationary distribution

$$\overline{\pi}(i) = \pi(\Omega_i),$$

while each $\mathcal{M}_i$ has stationary distribution

$$\pi_i(x) = \frac{\pi(x)}{\overline{\pi}(i)}.$$

With this notation, we have the following theorem by Martin and Randall [43].

Theorem 61. *Defining $\mathcal{M}_i$ and $\overline{\mathcal{M}}$ as above, we have*

$$\mathrm{Gap}(\mathcal{M}) \geq \frac{1}{2}\mathrm{Gap}(\overline{\mathcal{M}}) \min_{i\in[m]} \mathrm{Gap}(\mathcal{M}_i). \tag{4.16}$$

The theorem due to Hermon and Salez obtains better bounds if, for each pair $(i,j) \in [m] \times [m]$ with $\overline{P}(i,j) > 0$, we can find an effective joint distribution (often referred to as a "coupling") $\kappa_{ij} \colon \Omega_i \times \Omega_j \to [0,1]$ of the distributions π_i and π_j. In other words, we must have

$$\forall x \in \Omega_i, \qquad \sum_{y\in\Omega_j} \kappa_{ij}(x,y) = \pi_i(x), \tag{4.17}$$

$$\forall y \in \Omega_j, \qquad \sum_{x\in\Omega_i} \kappa_{ij}(x,y) = \pi_j(y). \tag{4.18}$$

The *quality* of the joint distribution κ is defined as

$$\chi := \chi(\kappa) := \min\left\{\frac{\pi(x)P(x,y)}{\overline{\pi}(i)\overline{P}(i,j)\kappa_{ij}(x,y)}\right\}, \tag{4.19}$$

where the minimum is taken over all (x,y,i,j) with $x \in \Omega_i, y \in \Omega_j$ for which $\overline{P}(i,j) > 0$ and $\kappa_{ij}(x,y) > 0$. Hermon and Salez [44] prove the following.

Theorem 62. *With P, $\bar{P}$, P_i, and χ defined as above,*

$$\mathrm{Gap}(\mathcal{M}) \geq \min\left\{\chi\mathrm{Gap}(\overline{\mathcal{M}}), \min_{i\in[m]} \mathrm{Gap}(\mathcal{M}_i)\right\}. \tag{4.20}$$

The utility of these decomposition theorems is that they allow us to break down a more complicated Markov chain into pieces that are easier to analyze. If we can show that the pieces rapidly mix, and the projection chain rapidly mixes, then we may conclude that the original chain rapidly mixes as well.

Additionally, to aid with the analysis of some projection chains, we will need another lemma from [43]. Let $\mathcal{M}_M$ be the Markov chain on $[m]$ with Metropolis transitions $P_M(i,j) = \frac{1}{2\Delta}\min\{1, \frac{\pi(\Omega_j)}{\pi(\Omega_i)}\}$ whenever $\overline{P}(i,j) > 0$, where Δ is the maximum degree of vertices in the transition graph of $\overline{M}$ Let $\partial_i(\Omega_j) = \{y \in \Omega_j \colon \exists x \in \Omega_i \text{ with } P(x,y) > 0\}$. Then we have the following

Lemma 63. *With $\mathcal{M}_M$ as defined above, suppose there exist constants $a > 0$ and $b > 0$ with*

1. *$P(x,y) \geq a$ for all x, y such that $P(x,y) > 0$.*
2. *$\pi(\partial_i(\Omega_j)) \geq b\pi(\Omega_j)$ for all i, j with $\overline{P}(i,j) > 0$.*

Then $\mathrm{Gap}(\overline{\mathcal{M}}) \geq ab \cdot \mathrm{Gap}(\mathcal{M}_M)$.

In order to help analyze the mixing time of $\mathcal{M}_M$, we will also require the following two lemmas. Note that Lemma 64 is used only in the proof of Lemma 65.

Lemma 64. *Let $(a_i)_{i=1}^m$ be a log concave sequence, with $a_i > 0$ for all $1 \leq i \leq m$. Then,*

$$\frac{a_{i+1}}{a_i} \geq \frac{a_{j+1}}{a_j} \tag{4.21}$$

for all $1 \leq i \leq j \leq m$.

Proof. In order to use induction, we will slightly reframe the statement. We will prove

$$\frac{a_{i+1}}{a_i} \geq \frac{a_{i+1+k}}{a_{i+k}}$$

for all $i + k \leq n$.

We now proceed by induction on k. The base case, $k = 0$, is trivial.

Now fix $l > 0$ and suppose that the induction hypothesis is true for $k = l - 1$, that is,

$$\frac{a_{i+1}}{a_i} \geq \frac{a_{i+l}}{a_{i+l-1}}.$$

By log concavity $a_{i+l}^2 \geq a_{i+l-1}a_{i+l+1}$, or, equivalently,

$$\frac{a_{i+l}}{a_{i+l-1}} \geq \frac{a_{i+l+1}}{a_{i+l}}.$$

Therefore,

$$\frac{a_{i+1}}{a_i} \geq \frac{a_{i+l}}{a_{i+l-1}} \geq \frac{a_{i+l+1}}{a_{i+l}},$$

where the first inequality follows from the induction hypothesis, and the second inequality follows from log concavity. □

Lemma 65. *Let π be a probability distribution on $[m]$. Let $\mathcal{M}$ be a Markov chain on $[m]$ with the transition probabilities*

$$P(i,j) = \begin{cases} \frac{1}{4}\min\left\{1, \frac{\pi(j)}{\pi(i)}\right\} & \text{if } |i-j| = 1 \\ 0 & \text{if } |i-j| > 1 \end{cases} \tag{4.22}$$

and the appropriate self-loop probabilities $P(i,i)$. If $\pi(i)$ is log concave in i, then $\mathcal{M}$ has mixing time (and hence also relaxation time) $O(m^2)$.

Proof. We define a coupling (X_t, Y_t) on $\mathcal{M}$ as follows. If $X_t \neq Y_t$, then at time step $t+1$, flip a fair coin.

- If heads, set $Y_{t+1} = Y_t$. Let l be either 1 or -1, each with probability $1/2$. If possible, let $X_{t+1} = X_t + l$ with probability $\frac{1}{2}\min\left\{1, \frac{\pi(X_t+l)}{\pi(X_t)}\right\}$. Otherwise, let $X_{t+1} = X_t$.
- If tails, set $X_{t+1} = X_t$, and update Y_{t+1} the same way as we did for X_{t+1} in the previous case.

Now, suppose that for some t we have $X_t = i$ and $Y_t = j$ for $i \neq j$. WLOG, assume that $i < j$. Let $\varphi(t) = \varphi(X_t, Y_t) = j - i$, and let $\Delta\varphi(t) = \varphi(t) - \varphi(t-1)$. Note that we have two moves, with probabilities $P(i, i-1)$ and $P(j, j+1)$, which will increase the distance φ by 1 and similarly two moves, with probabilities $P(i,i+1)$ and $P(j,j+1)$, will decrease the distance by 1. Then we have

$$\mathbb{E}(\Delta\varphi(t)) = -P(i,i+1) + P(i,i-1) + P(j,j+1) - P(j,j-1).$$

By the log-concavity of $\pi(i)$ and Lemma 64, we have $P(i,i+1) \geq P(j,j+1)$ and $P(i,i-1) \leq P(j,j-1)$. Therefore, the expected change in $\varphi(t)$ is always non-positive. We also have

$$\begin{aligned}
&\mathbb{E}\left((\Delta\varphi(t))^2 \,|X_t, Y_t\right) = P(j,j+1) + P(i,i+1) + P(j,j-1) + P(i,i-1)\\
&= \frac{1}{4}\left(\min\left\{1, \frac{\pi(j+1)}{\pi(j)}\right\} + \min\left\{1, \frac{\pi(i+1)}{\pi(i)}\right\}\right.\\
&\left. + \min\left\{1, \frac{\pi(j-1)}{\pi(j)}\right\} + \min\left\{1, \frac{\pi(i-1)}{\pi(i)}\right\}\right).
\end{aligned}$$

We claim that $E\left((\Delta\varphi)^2 \,|X_t, Y_t\right) \geq \frac{1}{4}$. Suppose, for contradiction, that the expectation is less than $\frac{1}{4}$. Then, for each of the minimum functions in the above expression, 1 must be the larger argument. Equivalently, $\pi(i-1) < \pi(i)$, $\pi(i) > \pi(i+1)$, $\pi(j-1) < \pi(j)$, and $\pi(j) > \pi(j+1)$. Therefore, $\pi(i)$ is not unimodal in i and is therefore also not log concave in i, contradicting our hypothesis. Therefore we have $E\left((\Delta\varphi)^2 \,|X_t, Y_t\right) \geq \frac{1}{4}$, as desired. □

4.3 Results

Here we present the constructed Markov chain and corresponding algorithms devised for the sampling task and the proof of an upper bound on the relaxation time - that the chain mixes rapidly. Collectively, the results illustrate an analytical approach to calculate dispersion of the secondary structure and corresponding branching properties of RNA based on the NNTM energy function minimization and without reference to a specific nucleotide sequence.

4.3.1 Our Markov Chain on $\mathfrak{M}_m^2$

We define a Markov chain $\mathcal{M} = X_0, X_1, X_2, \cdots$ on $\mathfrak{M}_m^2$ to sample 2-Motzkin paths as a representation of plane trees. Here, we use $m = n-1$ to denote the length of the 2-Motzkin paths corresponding to plane trees with n edges.

We define each step of $\mathcal{M}$ as follows. First, pick a random element l uniformly from $\{1, 2, 3, 4\}$. Now choose y as follows.

- If $l = 1$, pick a random pair of consecutive symbols in X_t, and call this pair s. If s is UD or HH, let s' be either UD or HH with probabilities $\frac{1}{1+e^{-\alpha}}$ and $\frac{e^{-\alpha}}{1+e^{-\alpha}}$ respectively. Let y be the string X_t with s replaced by s'. Otherwise, let $y = X_t$.

- If $l = 2$, pick i uniformly from $\{1, \cdots, m\}$. If $X_t(i)$ is H or I, choose a symbol c to be either H or I with probabilities $\frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}}$ and $\frac{e^{-\beta}}{e^{-\alpha}+e^{-\beta}}$ respectively. Let y be the 2-Motzkin path given by changing the symbol in $X_t(j)$ to c. Otherwise, we let $y = X_t$.

- If $l = 3$, pick i and j each uniformly from $\{1, \cdots, m\}$. If each of $X_t(i)$ and $X_t(j)$ are either U or D, let y be the string X_t with the symbols at indices i and j swapped. Otherwise, let $y = X_t$.

- If $l = 4$, pick a random pair of consecutive symbols in X_t, and call this pair s. If s is of the form ab or ba for some $a \in \{U, D\}$ and $b \in \{H, I\}$, let s' be the reverse of s, and let y be the string X_t with s replaced by s'. Otherwise, let $y = X_t$.

If y is a valid 2-Motzkin path, set $X_{t+1} = y$ with probability $\frac{1}{2}$. Otherwise, set $X_{t+1} = X_t$.

One can see that $\mathcal{M}$ is irreducible by noting that every path can be transformed to the path consisting of all H's. To make this transformation, first use the $l = 4$ rule to move all H's and I's to the end of the path. If there are any U's in the path, we must now have at least one consecutive pair UD. Use the $l = 1$ rule to convert the UD to a HH. From here we can repeat, again moving all H's to the end and replacing UD with HH, until only H's and I's remain. Finally, we can use the $l = 2$ rule to convert all I's to H's. Since all of these steps can also be taken in reverse, this gives a procedure to move between two arbitrary paths, demonstrating irreducibility. We can also conclude that $\mathcal{M}$ is aperiodic, due to the existence of self loops. Combined with irreducibility, this establishes that $\mathcal{M}$ is ergodic.

We claim that $\mathcal{M}$ is reversible with respect to the stationary distribution $\pi(x) = \frac{e^{-E(x)}}{Z}$, where $Z = \sum_{y \in \mathfrak{M}_m^2} e^{-E(y)}$. This can be easily verified by considering the 4 move types listed above. For example, for the first move type given above (transforming UD to HH and vice versa), let x and y be the states of interest. Suppose that y has the consecutive symbols HH where x contains UD. Then,

$$\begin{aligned}
\pi(x)P(x,y) &= \frac{e^{-\alpha\left(|x|_U + |x|_H + 1\right) - \beta|x|_I}}{Z} \cdot \frac{e^{-\alpha}}{1 + e^{-\alpha}} \\
&= \frac{e^{-\alpha\left((|y|_U + 1) + (|y|_H - 2) + 1\right) - \beta|y|_I}}{Z} \cdot \frac{e^{-\alpha}}{1 + e^{-\alpha}} \\
&= \frac{e^{-\alpha\left(|y|_U + |y|_H + 1\right) - \beta|y|_I}}{Z} \cdot \frac{1}{1 + e^{-\alpha}} \\
&= \pi(y)P(y,x).
\end{aligned}$$

One can verify that similar computations hold for the remaining 3 types of moves. Therefore, we conclude that the chain $\mathcal{M}$ has stationary distribution $\pi(x) = \frac{e^{-E(x)}}{Z}$.

The Markov chain $\mathcal{M}$ can be implemented in pseudocode as in Algorithm 1. Here, the Ber(p) function returns true with probability p, and false otherwise. We also use addition of strings to denote concatenation.

Additionally, in order to convert the 2-Motzkin path X_t into a plane tree, we use the algorithm in Algorithm 2, which assumes the existence of a Node object with children and parent attributes.

4.3.2 Mixing Time Results

Our main result is to prove the rapid mixing of the Markov chain defined in subsection 4.3.1. An upper bound on the relaxation time is achieved by bounding the spectral gap from below. A spectral gap bound for the complex chain at hand is obtained through the use of multiple decomposition theorems, which give bounds on the spectral gap of the complex chain in terms of the spectral gaps of multiple simpler chains. The disjoint decomposition theorem due to Martin and Randall [43] provides a flexible approach to decomposition of Markov chains. Very recent work by Hermon and Salez [44], building on the work of Jerrum, Son, Tetali, and Vigoda [109], proves a decomposition theorem with tighter bounds but stronger hypotheses.

Since this proof involves multiple decomposition steps, we provide an overview here. The primary tools used in this proof are the two decomposition theorems presented in subsubsection 4.2.2. We first partition the state space of all 2-Motzkin paths by the number of Us in the path. The projection chain from this first decomposition is linear and is proved to be rapidly mixing using a result of Martin and Randall [43] (Lemma 66). Each of the restriction chains are decomposed again, this time by the pattern of H and I symbols. The projection chains for this second decomposition are shown to be rapidly mixing by coupling (Lemma 67). The restriction chains are decomposed a third time, this time according to the skeleton of U and D steps. The projection chains for this third decomposition are shown to be rapidly mixing by comparison to the classic mountain valley moves chain on Dyck paths (Lemma 68). This last set of restriction chains are found to be rapidly mixing by isomorphism to the chain consisting of adjacent transpositions on binary strings (Lemma 69). Finally, starting from the most restricted chains, we use the decomposition theorems to obtain a bound on the spectral gap of the original chain (Theorem 70).

We now proceed with a formal presentation. We will use a series of decompositions of

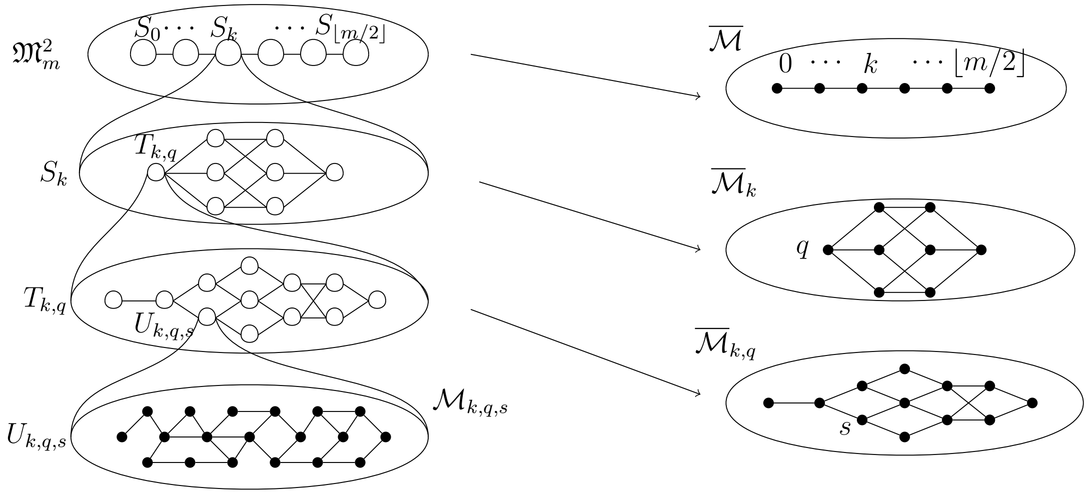

$\mathfrak{M}_m^2$
$S_0 \cdots S_k \cdots S_{\lfloor m/2 \rfloor}$
S_k
$T_{k,q}$
$T_{k,q}$
$U_{k,q,s}$
$U_{k,q,s}$
$\mathcal{M}_{k,q,s}$
$\overline{\mathcal{M}}$
$0 \cdots k \cdots \lfloor m/2 \rfloor$
$\overline{\mathcal{M}}_k$
q
$\overline{\mathcal{M}}_{k,q}$
s

Figure 4.3: The four level decomposition of $\mathfrak{M}_m^2$ (left), and the projection chains corresponding to each decomposition (right).

$\mathcal{M}$. We will first decompose our state space $\mathfrak{M}_m^2$ into $S_0, \cdots, S_{\rfloor m/2 \lfloor}$, where

$$S_k = \{x \in \mathfrak{M}_m^2 \colon |x|_U = k\}.$$

Let $\mathcal{M}_k$ denote the Markov chain $\mathcal{M}$ restricted to the set S_k, and let $\overline{\mathcal{M}}$ be the projection chain over this decomposition as outlined for Theorem 61.

Additionally, we will decompose each S_k into the sets $\{T_{k,q} \colon q \in (H+I)^{m-2k}\}$, where $(H+I)^{m-2k}$ denotes the set of strings with length $m-2k$ from the alphabet $\{H, I\}$. We define $T_{k,q}$ to be the set of 2-Motzkin paths $x \in S_k$ such that the substring of H and I symbols in x is q. Let $\mathcal{M}_{k,q}$ denote the chain $\mathcal{M}_k$ restricted to $T_{k,q}$, and let $\overline{M}_k$ be the projection chain of $\mathcal{M}_k$ over this decomposition.

Finally, we decompose each $T_{k,q}$ into the partition $\{U_{k,q,s} \colon s \in \mathfrak{D}_k\}$ based on the skeletons of the 2-Motzkin paths. For each $s \in \mathfrak{D}_k$, we define

$$U_{k,q,s} = \{x \in T_{k,q} \mid \sigma(x) = s\}.$$

As before, we let $\mathcal{M}_{k,q,s}$ be the Markov chain $\mathcal{M}_{k,q}$ restricted to $U_{k,q,s}$, and let $\overline{\mathcal{M}}_{k,q}$ be the appropriate projection chain. For clarity, this four-level decomposition is summarized in Figure 4.3.

Lemma 66. *$\overline{\mathcal{M}}$ has relaxation time $\tau_{rel}(\overline{\mathcal{M}}) = O(m^4)$.*

Proof. The chain $\overline{\mathcal{M}}$ is a linear chain with states k in $\{0, \dots, \lfloor m/2 \rfloor\}$, and with stationary distribution

$$\begin{aligned}\overline{\pi}(k) = \pi(S_k) &= \frac{C_k}{Z_m} \cdot \sum_{i=0}^{m-2k} \binom{m}{2k} \binom{m-2k}{i} e^{-\alpha(k+i+1)-\beta(m-2k-i)} \\ &= \frac{e^{-\alpha(k+1)}}{Z_m} \binom{m}{2k} C_k \cdot (e^{-\alpha} + e^{-\beta})^{m-2k},\end{aligned}$$

where $\overline{\pi}$ is defined as in subsubsection 4.2.2. Notice that transitions in $\mathcal{M}$ which move between the S_k sets are those which change a HH substring into a UD or DU substring, or vise versa. Thus, the transitions in $\overline{\mathcal{M}}$ only increase or decrease k by at most 1. We seek to apply Lemma 63. To choose a, notice that for $x \in S_k$ and $y \in S_{k\pm 1}$ with $P(x, y) > 0$, we have

$$P(x, y) = \frac{1}{4(m-1)} \frac{1}{1 + e^{\alpha}} \text{ or } P(x, y) = \frac{1}{4(m-1)} \frac{1}{1 + e^{-\alpha}}.$$

Note that the factor $1/4$ comes from the choice $l = 4$, and the factor $1/(m-1)$ comes from the fact that there are $m - 1$ adjacent pairs to pick from. Then,

$$P(x, y) \geq \frac{1}{4(m-1)(1 + e^{-|\alpha|)}}.$$

Thus, we pick $a = \frac{1}{4(m-1)(1+e^{-|\alpha|})}$.

To pick b, we let

$$\partial_-(S_k) = \{y \in S_k \colon \exists x \in S_{k-1}, P(x, y) > 0\}$$

for $k \in \{1, \cdots, \lfloor m/2 \rfloor\}$, and we let

$$\partial_+(S_k) = \{y \in S_k \colon \exists x \in S_{k+1}, P(x,y) > 0\}$$

for $k \in \{0, \cdots, \lfloor m/2 \rfloor - 1\}$.

Additionally, let A_k for $k \in \{1, \cdots, \lfloor m/2 \rfloor\}$ be the subset of S_k consisting of the 2-Motzkin paths in which the first D symbol appears immediately after a U. Let B_k for $k \in \{0, \cdots, \lfloor m/2 \rfloor - 1\}$ be the subset of S_k consisting of the 2-Motzkin paths in which a pair of adjacent H symbols occurs before all other H or I symbols. It is easy to see that $A_k \subset \partial_-(S_k)$ and $B_k \subset \partial_+(S_k)$. We have

$$\pi(A_k) = \frac{C_k e^{-\alpha(k+1)}}{Z_m} \binom{m-1}{2k-1} (e^{-\alpha} + e^{-\beta})^{m-2k},$$

as there are C_k ways to arrange the U and D symbols and $\binom{m-1}{2k-1}$ ways to insert $m - 2k$ H or I symbols (treating H and I as being identical for now) without placing anything between the first D and the U immediately before it. The energy contribution of the U and D symbols is given by $e^{-\alpha(k+1)}$, and the energy contribution of the H and I symbols is $\left(e^{-\alpha} + e^{-\beta}\right)^{m-2k}$. The required normalizing constant is Z_n. Similarly, we also get

$$\pi(B_k) = \frac{C_k e^{-\alpha(k+3)} e^{-2\beta}}{Z_m} \binom{m-1}{2k} (e^{-\alpha} + e^{-\beta})^{m-2k-2}$$

because there are C_k ways to arrange the U and D symbols and $\binom{m-1}{2k}$ ways insert $m - 2k - 1$ H or I symbols (treating the initial pair of H's as a single symbol gives us only $m - 2k - 1$ symbols to insert). The energy contribution of the U's, D's, and the initial two H's is given by $e^{-\alpha(k+3)} e^{-2\beta}$, and the energy contribution of the remaining H's and I's is $(e^{-\alpha} + e^{-\beta})^{m-2k-2}$. Finally, Z_m is again a normalizing constant.

Hence combining these two results, we have

$$\frac{\pi(\partial_-(S_k))}{\pi(S_k)} \geq \frac{\pi(A_k)}{\pi(S_k)} = \frac{2k}{m}$$

and

$$\frac{\pi(\partial_+(S_k))}{\pi(S_k)} \geq \frac{\pi(B_k)}{\pi(S_k)} = \frac{m-2k}{m}\left(\frac{e^{-\alpha}e^{-\beta}}{e^{-\alpha}+e^{-\beta}}\right)^2.$$

Thus, we may let $b = \frac{1}{m}\left(\frac{e^{-\alpha}e^{-\beta}}{e^{-\alpha}+e^{-\beta}}\right)^2$.

Applying Lemma 63, we get that $\mathrm{Gap}(\overline{\mathcal{M}}) \geq \frac{\mathrm{Gap}(\mathcal{M}_M)}{O(m^2)}$. Additionally, one can check that $\overline{\pi}(i)$ is log concave in i. Hence, using Lemma 65, we get $\tau_{rel}(\mathcal{M}_M) = O(m^2)$, and in turn $\tau_{rel}(\overline{\mathcal{M}}) = O(m^4)$, as claimed. □

Lemma 67. *$\overline{\mathcal{M}}_k$ has mixing time $\tau(\overline{\mathcal{M}}_k) = O(m \log m)$, for all k.*

Proof. Notice that $\overline{\mathcal{M}}_k$ appears as a chain with states q in the set $Q = (H+I)^{m-2k}$. Additionally, transitions in $\overline{\mathcal{M}}_k$ only occur between strings in Q that differ at only one index. The stationary distribution of $\overline{\mathcal{M}}_k$ is given by $\overline{\pi}_k(q) \propto e^{(\beta-\alpha)|q|_H}$, where we have intentionally used the constant of proportionality to remove all dependence on k, which we consider in this context to be fixed.

Additionally, for $q_1, q_2 \in Q$ which differ at exactly one index, we have the transition probability

$$\overline{P}_k(q_1,q_2) = \begin{cases} \frac{(m-2k)e^{-\alpha}}{4m(e^{-\alpha}+e^{-\beta})} & \text{if } |q_2|_H = |q_1|_H + 1 \\ \frac{(m-2k)e^{-\beta}}{4m(e^{-\alpha}+e^{-\beta})} & \text{if } |q_2|_H = |q_1|_H - 1 \end{cases}.$$

We may show that $\overline{\mathcal{M}}_k$ rapidly mixes by a simple coupling argument. Let $(X_t, Y_t)_{t=0}^{\infty}$ be our coupled Markov chain on $Q \times Q$. We define one step in this coupled chain as follows.

1. With probability $1 - \frac{m-2k}{4m}$, set $(X_{t+1}, Y_{t+1}) = (X_t, Y_t)$.

2. Otherwise, pick a random index $j \in [m-2k]$. Let $a \in \{H, I\}$ be a random symbol

such that $\Pr(a = H) = \frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}}$ and $\Pr(a = I) = \frac{e^{-\beta}}{e^{-\alpha}+e^{-\beta}}$. Now let X_{t+1} and Y_{t+1} be X_t and Y_t respectively, each with the jth symbol changed to a.

One can check that each of $(X_t)_t$ and $(Y_t)_t$ are indeed copies of $\overline{\mathcal{M}}_k$. Additionally, notice that we will have $X_t = Y_t$ after all $m - 2k$ possible indices j have been updated. By the Coupon Collector Theorem, we have the coupling time of this chain to be $T_{\overline{\mathcal{M}}_k} = \frac{4m}{m-2k} \cdot O((m - 2k) \log(m - 2k)) = O(m \log m)$. Thus, using Theorem 60, we have the mixing time (and the relaxation time) also $O(m \log m)$. □

Lemma 68. *$\overline{\mathcal{M}}_{k,q}$ has relaxation time $\tau_{rel}(\overline{\mathcal{M}_{k,q}}) = O(m^2)$, for all pairs (k, q).*

Proof. Notice that all $x \in T_{k,q}$ have equal energy, and that $|U_{k,q,s}| = \binom{m}{2k}$ for all s. Thus, $\overline{\mathcal{M}}_{k,q}$ has a uniform stationary distribution. If we represent each set $U_{k,q,s}$ by the Dyck path s, we can think of $\overline{\mathcal{M}}_{k,q}$ as a chain over $\mathfrak{D}_k$. Since all the transitions in $\mathcal{M}_{k,q}$ that move between the $U_{k,q,s}$ sets are moves that exchange the positions of a U and a D, the transitions in $\overline{\mathcal{M}}_{k,q}$ are simply the moves on elements of $\mathfrak{D}_k$ which exchange a U with a D. We call these moves on the elements of $\mathfrak{D}_k$, *transposition moves*.

For each $s_1, s_2 \in \mathfrak{D}_k$ that differ by a transposition move, the transition probabilities in our projection chain are given by

$$\begin{aligned}\overline{P}_{k,q}(s_1, s_2) &= \frac{1}{\pi(U_{k,q,s_1})} \sum_{\substack{x \in U_{k,q,s_1} \\ y \in U_{k,q,s_2}}} \pi(x) P(x, y) = \frac{1}{|U_{k,q,s}|} \sum_{\substack{x \in U_{k,q,s_1} \\ y \in U_{k,q,s_2}}} P(x, y) \\ &= \frac{1}{\binom{m}{2k}} \sum_{\substack{x,y \\ P(x,y)>0}} \frac{1}{4m^2} = \frac{1}{4m^2}.\end{aligned}$$

The last equality above relies on counting the number of terms in the sum. Notice that for each $x \in U_{k,q,s_1}$, there is a unique $y \in U_{k,q,s_2}$ for which $P(x, y) > 0$. Therefore, the number of terms is simply $|U_{k,q,s_1}| = \binom{m}{2k}$. Compare this chain to the traditional mountain valley Markov chain on $\mathfrak{D}_k$, which we will denote by $\mathcal{M}'$. The transition probabilities of

$\mathcal{M}'$ are given by $P'(s_1, s_2) = \frac{1}{k^2}$ for each pair (s_1, s_2) which differ by a mountain-valley move. It is known from Cohen [110] that $\mathrm{Gap}(\mathcal{M}') = \frac{1}{O(k^2)}$. Thus, applying Lemma 56 to $\overline{\mathcal{M}}_{k,q}$ and $\mathcal{M}'$, we see that $\mathrm{Gap}(\overline{\mathcal{M}}_{k,q}) = \frac{1}{O(m^2)}$. □

Lemma 69. *$\mathcal{M}_{k,q,s}$ has relaxation time $\tau_{rel}(\mathcal{M}_{k,q,s}) = O(m^3)$, for all valid triples (k, q, s).*

Proof. Notice that transitions in $\mathcal{M}_{k,q,s}$ consist only of moves which involve swapping an H or an I with an adjacent U or D. Additionally, all 2-Motzkin paths in $U_{k,q,s}$ have equal energy, so for all $x, y \in U_{k,q,s}$ such that $P(x, y) > 0$, we have $P(x, y) = \frac{1}{8(m-1)}$.

To determine the mixing time of $\mathcal{M}_{k,q,s}$, consider an isomorphic chain. Let U' be the set of all binary strings of length m with $2k$ zeros and $m - 2k$ ones. Let $\mathcal{M}'$ be the Markov chain on U' where each step does nothing with probability $7/8$ and swaps a random pair of adjacent (potentially identical) digits with probability $1/8$. From Wilson [111], we know that the spectral gap of $\mathcal{M}'$ is $\frac{1}{O(m^3)}$. □

Finally, we can combine our bounds on the spectral gaps of all of these chains to prove our main result.

Theorem 70. *The Markov chain $\mathcal{M}$ has relaxation time $\tau_{rel}(\mathcal{M}) = O(m^7)$, for all $\alpha, \beta \in \mathbb{R}$.*

Proof. We use Lemma 69 and Lemma 68 along with Theorem 62 to obtain a bound on $\mathrm{Gap}(\mathcal{M}_{k,q})$. We define a coupling κ_{s_1,s_2} for each pair $(s_1, s_2) \in \mathfrak{D}_k \times \mathfrak{D}_k$ with $\overline{P}_{k,q}(s_1, s_2) > 0$. For each such pair, notice that the set of pairs $(x, y) \in U_{k,q,s_1} \times U_{k,q,s_2}$ with $P(x, y) > 0$ is a perfect matching. Thus, we may set

$$\kappa_{s_1,s_2}(x, y) = \begin{cases} \frac{1}{\binom{m}{2k}} & \text{if } P(x, y) > 0 \\ 0 & P(x, y) = 0 \end{cases}.$$

To compute χ, we begin by observing $\pi(x) = \pi(y)$ for all $x, y \in \mathcal{M}_{k,q}$. Also note $|U_{k,q,s}| = \binom{m}{2k}$ for all skeletons s of length $2k$. Before computing χ, we start by finding

$\overline{P}(s_1, s_2)$.

$$\begin{aligned}
\overline{P}(s_1, s_2) &= \frac{1}{\pi(U_{k,q,s_1})} \sum_{x \in U_{k,q,s_1}, y \in U_{k,q,s_2}} \pi(x) P(x, y) \\
&= \frac{1}{\pi(U_{k,q,s_1})} \sum_{x \in U_{k,q,s_1}, y \in U_{k,q,s_2}} \frac{\pi(x)}{\frac{1}{4}\binom{m}{2}} \\
&= \frac{1}{\pi(U_{k,q,s_1})} \left| U_{k,q,s_1} \right| \frac{4\pi(x)}{\binom{m}{2}} \\
&= \frac{4}{\binom{m}{2}}.
\end{aligned}$$

We now proceed with the calculation of χ. Recall that the minimum is taken over all tuples x, y, s_1, s_2 where $\overline{P}(s_1, s_2) > 0$ and $\kappa_{0_1,s_2}(x, y) > 0$.

$$\begin{aligned}
\chi &= \min \left\{ \frac{\pi(x) P\left(x, y\right)}{\overline{\pi}(s_1) \overline{P}(s_1, s_2) \kappa_{s_1,s_2}(x, y)} \right\} \\
&= \min \left\{ \frac{\pi(x) \frac{4}{\binom{m}{2}}}{\pi(U_{k,q,s_1}) \frac{4}{\binom{m}{2}} \frac{1}{\binom{m}{2k}}} \right\} \\
&= \frac{\binom{m}{2k}}{\binom{m}{2k}} = 1.
\end{aligned}$$

Theorem 62 then gives

$$\begin{aligned}
\mathrm{Gap}(\mathcal{M}_{k,q}) &\geq \min \left\{ \chi \mathrm{Gap}(\overline{\mathcal{M}}_{k,q}), \min_s \mathrm{Gap}(\mathcal{M}_{k,q,s}) \right\} \\
&\min \left\{ \frac{1}{O(m^2)}, \frac{1}{O(m^3)} \right\} \\
&= \frac{1}{O(m^3)}.
\end{aligned}$$

Similarly, we define a coupling κ_{q_1,q_2} for each pair $(q_1, q_2) \in (H+I)^{m-2k} \times (H+I)^{m-2k}$ with $\overline{P}_k(q_1, q_2) > 0$ to apply Theorem 62 to $\overline{M}_k$. Notice that once again, the set of pairs

$(x,y) \in T_{k,q_1} \times T_{k,q_2}$ for which $P(x,y) > 0$ forms a perfect matching. Thus, we take

$$\kappa_{q_1,q_2}(x,y) = \begin{cases} \frac{1}{\binom{m}{2k} C_k} & \text{if } P(x,y) > 0 \\ 0 & P(x,y) = 0 \end{cases}.$$

To compute χ for this coupling, we again begin with a few preliminary computations. In all of the following, let $x \in T_{k,q_1}, y \in T_{k,q_2}$ with $\overline{P}(q_1,q_2) > 0$. Note that q_1 and q_2 have the same length and differ at only one index. We will show the computations for the case where q_1 has a I where q_2 has a H. The computations for the other case are nearly identical.

Note that $P(x,y) = \frac{e^{-\alpha}}{e^{-\alpha}+a^{-\beta}}$. Also note

$$\overline{\pi}(q_1) = \pi(T_{k,q_1}) = \pi(x)\,|T_{k,q_1}| = \pi(x) C_k \binom{m}{2k}$$

and

$$\begin{aligned} \overline{P}(q_1,q_2) &= \frac{1}{\pi(T_{k,q_1})} \sum_{x' \in T_{k,q_1}, y' \in T_{k,q_2}} P(x',y') \\ &= \frac{1}{|T_{s,q_1}|} \cdot \frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}} \, |T_{s,q_1}| \\ &= \frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}}. \end{aligned}$$

Now we can compute

$$\begin{aligned} \chi &= \min \left\{ \frac{\pi(x)P(x,y)}{\overline{\pi}(q_1)\overline{P}}(q_1,q_2)\kappa_{q_1,q_2}(x,y) \right\} \\ &= \min \left\{ \frac{\pi(x)\frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}}}{\pi(x)C_k\binom{m}{2k}\frac{e^{-\alpha}}{e^{-\alpha}+e^{-\beta}} \cdot \frac{1}{C_k\binom{m}{2k}}} \right\} \\ &= 1. \end{aligned}$$

Applying Theorem 62 then gives

$$\begin{aligned}\mathrm{Gap}(\mathcal{M}_k) &\geq \min\left\{\chi\mathrm{Gap}(\overline{\mathcal{M}_k}), \min_q \mathrm{Gap}(\mathcal{M}_{k,q})\right\} \\ &= \min\left\{\frac{1}{O(m\log m)}, \frac{1}{O(m^3)}\right\} \\ &= \frac{1}{O(m^3)}.\end{aligned}$$

Unfortunately, we have not been able to find a useful coupling for $\overline{\mathcal{M}}$, so for the last step of our decomposition, we apply Theorem 61. Since $\mathrm{Gap}(\overline{\mathcal{M}}) = O\left(\frac{1}{m^4}\right)$ and $\mathrm{Gap}(\mathcal{M}_k) = O(\frac{1}{m^3})$ for all k, we have

$$\begin{aligned}\mathrm{Gap}(\mathcal{M}) &\geq \frac{1}{2}\mathrm{Gap}(\overline{\mathcal{M}}) \min_{k\in[m/2]} \mathrm{Gap}(\mathcal{M}_k) \\ &= \frac{1}{2O(m^4)O(m^3)} \\ &= \frac{1}{O(m^7)},\end{aligned}$$

establishing Theorem 70. □

Finally, an application of Lemma 58 allows us to conclude that the mixing time is also polynomially-bounded.

Corollary 71. *$\mathcal{M}$ is rapidly mixing.*

Proof. In order to apply Lemma 58, we need to obtain a polynomial bound on $\log(1/\pi(x))$

for all $x \in \Omega$. Let $t \in \Omega$ have maximum energy among all elements of Ω. For any $x \in \Omega$,

$$\begin{aligned}
\log\left(\frac{1}{\pi(x)}\right) &= \log\left(\frac{\sum_{y\in\Omega} e^{-\alpha d_0(y)-\beta d_1(y)}}{e^{-\alpha d_0(x)-\beta d_1(x)}}\right)\\
&\leq \log\left(\frac{C_n e^{-\alpha d_0(t)-\beta d_1(t)}}{e^{-\alpha d_0(x)-\beta d_1(x)}}\right)\\
&\leq \log\left(\frac{C_n e^{-\alpha n-\beta n}}{e^{-\alpha}}\right)\\
&= \log\left(C_n e^{-\alpha(n-1)} e^{-\beta n}\right)\\
&\leq n\log(2n) + \log\left(\frac{1}{n+1}\right) - \alpha(n-1) - \beta n.
\end{aligned}$$

This gives us the required polynomial bound, and therefore Lemma 58 implies that $\mathcal{M}$ is rapidly mixing. □

4.4 Discussion and Conclusions

The goal of this work was to identify a Markov chain and construct a corresponding algorithm by which to examine the non-uniform distribution and dispersion properties of NNTM RNA secondary structures and branching properties independent of a specific nucleotide sequence. This study successfully identifies the existence of a Markov chain, with a provably polynomial mixing time, which generates a Gibbs distribution on plane trees. This stationary probability distribution models branching characteristics of RNA secondary structure under the NNTM. While exploration of sampled structures obtained from this algorithm are beyond the scope of the presented results, pseudocode (see subsection 4.3.1) is provided to facilitate future work in this area. Below we discuss the direct applications and implications of this work to RNA modeling, the possibility of implementing a dynamic programming approach, the possibility of an approach using stochastic context free grammars, other biological applications of this work, contributions of this work towards independent mathematical research interests, and limitations and future directions of the present work.

4.4.1 Applications to RNA modeling

The most straightforward application of this work is in understanding the background distribution of the branching behavior for secondary structures predicted under the NNTM. While the NNTM is widely used to predict secondary structures from sequence data, little is known about the general branching characteristics of the predicted structures, independent of a specific input sequence. Quantities such as the number of hairpins, the maximum branching in a multiloop, the average branching in a multiloop, and the maximum ladder distance of the structure [9, 88] help to characterize the branching behavior and could be computed from samples obtained from this algorithm. These quantities also have been studied in native structures and/or could be easily obtained from databases such as the RNA Secondary Structure and Statistical Analysis Database (RNA STRAND) [112]. The parameter values of α, β, and γ corresponding to various revisions of the NNTM are given in Table 4.1 in subsection 4.2.1. The Markov chain and corresponding algorithms presented will enable biologists to calculate the dispersion of key branching properties for a specific energy function. As described with the detailed hairpin dispersion example in the Introduction (section 4.1), knowing whether branching properties fall within acceptable dispersion limits is crucial for deducing potential functional insight or hypothesizing other scientific ramifications.

Another key application to RNA modeling of the presented algorithms is the ability to explore the parameter space of possible values for α and β. While the various revisions of the NNTM correspond to specific values for these parameters, in principle any real-valued parameters could be used. Finding values for these parameters that approximate reality remains an open question. Yet, determination of how differences in parameter values change the distribution of NNTM branching properties, such as maximum ladder distance, is crucial. Moreover, parameter space exploration is necessary to identify and further explore the phase transitions that exist. The presented Markov chain and corresponding algorithms expedite such future computational experimentation. Therefore, collectively, the presented algorithm enables exploration that will greatly improve understanding of NNTM-basesd RNA secondary structures and branching properties, as well as identify potential limitations or specific branching structures where the NNTM models do not sufficiently emulate reality. For example, NNTM-based free energy minimization algorithms achieved an accuracy of at least 60% in only 9% of 16S secondary structures analyzed by Doshi et. al. [84].

The algorithm presented here can only sample under an energy function of the form $\alpha d_0 + \beta d_1$, and this does not capture the entirety of the model presented in [12], which considers energy functions of the form $\alpha d_0 + \beta d_1 + \gamma r$. However, the missing term, γr, represents the energy contribution of exterior loop, and the exterior loop contributes less of the total free energy as sequence length increases. Therefore, when interested in sequences of at least moderate length, this algorithm may be able to provide insight, as long as information about the exterior loop is not the specific object of study. Also note that other authors have made similar simplifications with respect to the exterior loop, e.g. [86].

4.4.2 Possibility of a dynamic programming approach

This sampling problem to calculate the dispersion of NNTM RNA secondary structure and properties utilized Markov chain techniques. However, is it possible to utilize a dynamic programming algorithm? It is straightforward to sample Dyck paths under a uniform probability distribution using dynamic programming techniques. However, it is not clear whether a similar technique could be used for the Gibbs distribution we define here, due to the complexity of the energy function. In particular, large numeric computations may be required to handle the variable k, the number of U steps in a path. While Alonso presents a way to sample from the unweighted distribution $\Pr(k = l) \propto \binom{m}{2l} C_l$ in $O(n)$ time without large computations [113], it is unclear if a similar method may be used for the present application.

4.4.3 Possibility of an SCFG approach

Stochastic context free grammars (SCFGs) have been widely used in the field of RNA secondary structure prediction, e.g. [114, 115, 116, 117]. Most commonly, the probabilities for production rules in an SCFG are determined by training on a set of known secondary structures, often including covariance information from homologous structures. These approaches are not immediately applicable to the problem we study here, as they do not give any insight into the NNTM multiloop energy parameters.

However, some authors have constructed SCFGs based on the NNTM. In particular, Nebel and Scheid [114] construct a SCFG with 29 distinct production rules to mirror the NNTM features. They also present a sampling algorithm allowing for sampling structures of a fixed size using the grammar. However, they do not actually compute probabilities for the production rules that would allow one to sample from a Gibbs distribution (with NNTM energy) and instead rely on training on a set of known structures. Indeed, it is not clear from the paper whether such a set of probabilities must exist.

Even in the case of the simplified model we present in this chapter, it is not clear how to assign probabilities to production rules in an SCFG so that the probability of obtaining a given structure matches the Gibbs probability under the NNTM. See section 4.5 for more details.

Even if a suitable SCFG could be formulated, the SCFG approach is not necessarily superior. The sampling algorithm presented by Nebel and Scheid has time complexity $O(n^3)$ and space complexity $O(n^2)$. While the algorithm we present does have large time complexity, it only requires linear space, which may be an advantage for some applications.

Even though we cannot easily formulate a SCFG, it is reasonable to consider whether a context free grammar (such as that presented in section 4.5) could nonetheless be used as the basis for a dynamic programming algorithm. In fact, this is possible. The key idea is to create a table for each non-terminal symbol X and then populate entry k of the table with

$$\sum e^{E(t)},$$

where the sum is taken over all trees $t \in \mathfrak{T}_k$ which can be derived from symbol X.

Once the tables been populated with these (non-normalized) probabilities, a stochastic backtracking procedure can be used to obtain samples.

However, as in subsection 4.4.2, an assumption that each arithmetic operation can be performed in unit time is not appropriate here. Because the elements of our dynamic programming tables are in fact parts of the partition function, we can conclude that the numbers involved could have up to $O(n)$ digits. Each arithmetic operation therefore becomes much more expensive. While a polynomial time dynamic programming algorithm based on a context free grammar is possible, an efficient dynamic programming algorithm would require substantially more work.

4.4.4 Extended applications

The Markov chain mixing analysis techniques explored in this chapter have potential for useful application in a variety of fields. Markov chain Monte Carlo algorithms are widely used in several fields including, machine learning [118], econometrics [119], and Bayesian Statistics [120]. In virtually all applications, an understanding of mixing time increases confidence in the results. In some situations, an understanding of mixing time may also allow for more efficient algorithm selection and implementation.

While many Markov chains with nonuniform stationary distributions have been used for biological applications (e.g. [35, 36, 37, 38]), theoretical guarantees on the mixing time are generally not known. Instead, researchers must rely on convergence heuristics, and if fact many introductions to Markov chain Monte Carlo written for biologists explain such heuristic techniques [121, 39, 40, 41]. Of course, heuristics can be misleading, and rigorous mixing time guarantees would be significantly preferable. The same techniques used in this work might be used to generate algorithms with rigorous mixing time bounds for other biological problems concerning a nonuniform distribution.

The mathematical techniques used in this chapter have been widely used in mathematics, physics, and computer science, demonstrating their broader applicability. For numerous examples, we direct the reader to the books of Levin, Peres, and Wilmer [122]; Montenegro and Tetali [123]; and Jerrum [45].

As an example where similar techniques have found utility in biological applications, it is interesting to briefly consider the study of cladograms, which arise from phylogenetic trees. Mathematically, a cladogram is a binary tree with n labeled leaves and $n-2$ unlabeled internal nodes. While an explicit formula is known for the exact number of cladograms of a given size, mixing time under certain dynamics has also been studied. For example, Aldous [124] studied a Markov chain where a leaf is removed at random and then attached to a random edge in the tree, obtaining a proof that the mixing time is bounded below by $O(n^2)$ and bounded above by $O(n^3)$. Further work by Schweinsberg [125] later proved an upper bound of $O(n^2)$, closing the gap between the upper and lower bounds.

4.4.5 Independent mathematical research interests

The plane trees examined as a model for RNA secondary structure are of independent mathematical interest. As Catalan objects, they have been studied combinatorially (see, for example, [126, 105]), and Markov chains on Catalan objects have received significant attention over the years [110, 127, 128, 129, 111], but with very few results providing tight estimates on the corresponding mixing times; most commonly these are discussed in the language of Dyck paths. Cohen's thesis [110] gives an overview of the known mixing time results for chains on Catalan objects. All of the chains surveyed there have uniform distribution over the Catalan-sized state space as their stationary distribution. Among these, essentially the only known chain with tight bounds (upper and lower bounds differing by a small multiplicative constant) is due to Wilson [111] and gives the relaxation time of $O(n^3)$ for the walk consisting of adjacent transpositions on Dyck paths. In comparison, in [127] the chain using *all* (allowed) transpositions has been shown to have relaxation time of $O(n^2)$, and further conjectured to have $O(n)$ as the relaxation time, in analogy with the random transposition shuffle of n cards.

Judging from the lack of progress on several of these chains, it is evident that determining mixing or relaxation time for these chains is typically a challenging problem, even in the case where the stationary distribution is uniform.

In the current work, the RNA secondary structure modeling naturally leads to a state space on Catalan objects with a nonuniform distribution, making the corresponding mixing time analysis even more challenging. Another example where mixing times are estimated for Markov chains on Catalan objects with nonuniform stationary distribution is the work of Martin and Randall [43], which examines a Gibbs distribution on Dyck paths weighted by the number of returns to the x-axis.

4.4.6 Limitations and Future Directions

While the mixing time proved here is polynomial, it is almost certainly too large to allow for any practical computational sampling experiments. However, we conjecture the actual mixing time to be much smaller, and future work may provide a better bound. Even without additional theoretical results, interesting work is possible using the algorithm we present and heuristic methods for evaluating Markov chain mixing. See [130, Ch. 8] for a discussion of heuristic methods for monitoring Markov chain convergence.

The results of this study provide an important mathematical foundation for examining dispersion of RNA secondary structures and branching properties using a Markov chain. However, more work is necessary to optimize the developed computational application for incorporation into the software utilized by biologists that study RNA. Example questions that strongly compel further investigation include:

1. Can the mixing time bound in our main result be improved?

2. Is there a rapidly mixing chain, with the same stationary distribution studied here, whose transitions correspond naturally to moves on the set plane trees? Mixing time bounds on the chain of matching exchange moves, as defined in [131], would be especially interesting, as such a chain may relate to RNA folding kinetics.

3. Is there a rapidly mixing chain converging to the Gibbs distribution using the full energy function for the utilized NNTM model [12]? The chain presented here uses only the parameters α and β, setting $\gamma = 0$.

4. Is there a stochastic context free grammar which generate secondary structures (in our simplified model or using the full NNTM) according to a Gibbs distribution with NNTM energy?

4.5 Supplement: SCFG

This section attempts to illustrate it is not apparent how to formulate stochastic context free grammar based on the work of Rivas and Eddy [116]and Nebel and Scheid [114] that generates a given NNTM Gibbs distribution. To do so, we will attempt to formulate such a grammar, making the most natural or logical choices at each step. We do not claim that finding such a grammar is impossible; we only claim that is not apparent from the existing literature.

We first give a description of a stochastic context free grammar similar to that described by Nebel and Scheid [114] but restricted to the plane tree model of RNA secondary structure. Notably, each of the production rules corresponds to a specific change in free energy under the NNTM.

Plane tree are represented as strings of parentheses, using the typical Catalan bijection. The notation for the free energy is consistent with subsection 4.2.1.

The alphabet of terminal symbols is $\{(,)\}$, and the non-terminals are s, t, u, with s being the initial state. We additionally use ϵ to denote the empty string.

For the time being, we leave the probabilities undetermined.

probability	**production rule**	**description**	**free energy**
s_1	$s \to (t)s$	branch on exterior loop	g
s_2	$s \to \epsilon$	end of exterior loop	0
t_1	$t \to (t)u$	first branch on a multiloop	$a + 8b + 2c$
t_2	$t \to (t)$	internal node	i
t_3	$t \to \epsilon$	hairpin	f
u_1	$u \to (t)u$	additional branches on a multiloop	$4b + c$
u_2	$u \to (t)$	last branch on a multiloop	$4b + c$

4.5.1 Determination of production rule probabilities

Given specific values for the free energy parameters a, b, c, f, g, i, we need to pick specific values for the production rule probabilities $s_1, s_2, t_1, t_2, t_3, u_1, u_2$ so that the probability of generating a given string with the grammar, given the length of the string, is the Gibbs probability.

That is, given a plane tree with n edges, root degree r, and down degree sequence (excluding the root) $d_0, d_1, \ldots, d_{n-1}$, the free energy should be given by

$$gr + id_1 + fd_0 + \sum_{i=2}^{n-1} \left(d_i(a + 8b + 2c) + d_i(i-1)(4b + c)\right),$$

and hence the probability (given the number of edges n) should be

$$\frac{e^{-\left(gr+id_1+fd_0+\sum_{i=2}^{n-1}(d_i(a+8b+2c)+d_i(i-1)(4b+c))\right)}}{Z_n}, \tag{4.23}$$

where Z_n is the appropriate normalizing constant:

$$Z_n = \sum_{t \in \mathcal{T}_n} e^{-\left(gr(t)+id_1(t)+fd_0(t)+\sum_{i=2}^{n-1}(d_i(t)(a+8b+2c)+d_i(i-1)(4b+c))\right)}.$$

Note that a plane tree with n edges, degree sequence $d_0, d_1, \ldots, d_{n-1}$, and root degree r has probability in the grammar

$$s_1^r s_2 t_2^{d_1} t_3^{d_0} \prod_{i=2}^{n-1} t_1^{d_i} u_2^{d_i} u_1^{(i-1)d_i}.$$

Conditioning on the requirement that the length of the string be $2n$ (or, equivalently, that the plane tree has n edges) gives

$$\frac{s_1^r s_2 t_2^{d_1} t_3^{d_0} \prod_{i=2}^{n-1} t_1^{d_i} u_2^{d_i} u_1^{(i-1)d_i}}{Y_n}, \tag{4.24}$$

where Y_n is the probability that the grammar generates a string of length $2n$.

Based on the similarity between Equation 4.23 and Equation 4.24, it is natural to try setting

$$\begin{aligned}
s_1 &= e^{-g}/Z_s \\
s_2 &= 1/Z_s \\
t_1 &= e^{-(a+8b+2c)}/Z_t \\
t_2 &= e^{-i}/Z_t \\
t_3 &= e^{-f}/Z_t \\
u_1 &= e^{-(4b+c)}/Z_u \\
u_2 &= e^{-(4b+c)}/Z_u,
\end{aligned}$$

where

$$\begin{aligned} Z_s &= e^{-g} + 1 \\ Z_t &= e^{-(a+8b+2c)} + e^{-i} + e^{-f} \\ Z_u &= 2e^{-(4b+c)} \end{aligned}$$

are normalizing constants which ensure $s_1 + s_2 = 1$ as well as $t_1 + t_2 + t_3 = 1$ and $u_1 + u_2 = 1$.

We note that, if not for these normalizing constants Z_s, Z_t, Z_u, we would be able to obtain equality between the probability given by the grammar and that given by the Gibbs distribution. However, the normalizing constants are necessary to satisfy the definition of a stochastic context free grammar.

Proceding with the definitions of $s_1, s_2, t_1, t_2, t_3, u_1, u_2$ given above, we see that the probability that a given tree is generated by the grammar is

$$\begin{aligned} &\frac{s_1^r s_2 t_2^{d_1} t_3^{d_0} \prod_{i=2}^{n-1} t_1^{d_i} u_2^{d_i} u_1^{(i-2)d_i}}{Y_n} \\ &= \frac{e^{-rg} e^{-d_1 i} e^{-d_0 f}}{Y_n Z_s^{r+1} Z_t^{d_1+d_0}} \prod_{i=2}^{n-1} \left(\frac{e^{d_i(a+8b+2c)} e^{-d_i(4b+c)} e^{-(i-2)d_i(4b+c)}}{Z_t^{d_i} Z_u^{(i-1)d_i}} \right) \\ &= \frac{e^{-\left(rg+d_1 i+d_0 f+\sum_{i=2}^{n-1}(d_i(a+8b+2c)+d_i(i-1)(4b+c))\right)}}{Y_n Z_s^{r+1} Z_t^{n} Z_u^{d_0-r}} \end{aligned}$$

Note that the numerator now matches exactly with the numerator in Equation 4.23. In order to obtain equality for the whole expression, we would need

$$Y_n Z_s^{r+1} Z_t^n Z_u^{d_0-r} = Z_n$$

for all trees with n edges.

The left hand side expands to

$$Y_n\left(e^{-g}+1\right)^{r+1}\left(e^{-(a+8b+2c)}+e^{-i}+e^{-f}\right)^n\left(2e^{-(4b+c)}\right)^{d_0-r}.$$

Even without an explicit expression for Y_n, we know that Y_n is constant for fixed n. However, the non-constant portion

$$\left(e^{-g}+1\right)^{r+1}(2e)^{-(4b+c)(d_{0-r})}$$

clearly varies among trees with n edges. Hence, the denominator we obtain when computing the probability of obtaining a string using the stochastic context free grammar clearly cannot be equal to the constant Z_n.

Therefore, we have shown that one clear approach to formulating a stochastic context free grammar based on the work of Rivas and Eddy (1999) and Nebel and Scheid (2011) fails. We do not claim that no such grammar exists. We only claim that the approach which seems most obvious does not work.

Algorithm 1 The main Markov chain algorithm. This pseudocode calculates X_t given X_0.

```
Input: $X_0$ is a valid 2-Motzkin path of length $m$.
  $x \leftarrow X_0$
  for $s = 1 \to t$ do
    $y \leftarrow x$
    $l \leftarrow$ randInt$(1, 4)$
    if $l = 1$ then
      $i \leftarrow$ randInt$(1, m - 1)$
      if $x[i : i + 1] = UD$ and Ber $\left(\frac{e^{-\alpha}}{2(1+e^{-\alpha})}\right)$ then
        $y[i : i + 1] \leftarrow HH$
      else if $x[i : i + 1] = HH$ and Ber $\left(\frac{1}{2(1+e^{-\alpha})}\right)$ then
        $y[i : i + 1] \leftarrow UD$
    else if $l = 2$ then
      $i \leftarrow$ randInt$(1, m)$
      if $x[i] = I$ and Ber $\left(\frac{e^{-\alpha}}{2(e^{-\alpha}+e^{-\beta})}\right)$ then
        $y[i] \leftarrow H$
      else if $x(i) = H$ and Ber $\left(\frac{e^{-\beta}}{2(e^{-\alpha}+e^{-\beta})}\right)$ then
        $y[i] \leftarrow I$
    else if $l = 3$ then
      $i \leftarrow$ randInt$(1, m)$
      $j \leftarrow$ randInt$(1, m)$
      if ($x[i] \in \{U, D\}$ and $x[j] \in \{U, D\}$) and Ber $\left(\frac{1}{2}\right)$ then
        $y[i] \leftarrow x[j]$
        $y[j] \leftarrow x[i]$
        if $y$ is not a valid 2-Motzkin path then
          $y \leftarrow x$
    else if $l = 4$ then
      $i \leftarrow$ randInt$(1, m - 1)$
      if ($x[i] \in \{U, D\}$ and $x[j + 1] \in \{H, I\}$) or ($x[i] \in \{H, I\}$ and $x[j + 1] \in \{U, D\}$)
      and Ber $\left(\frac{1}{2}\right)$ then
        $y[i : i + 1] \leftarrow x[j + 1] + x[j]$
    $x \leftarrow y$
  return $x$
```

Algorithm 2 Algorithm to convert a sampled 2-Motzkin path to a plan tree. The pseudocode calculates $\Phi^{-1}(x)$.

```
Input: $x$ is a valid 2-Motzkin path of length $m$.
  root ← new Node()
  // $u$ will be where a new node will be added for an $H$ or $D$ symbol
  $u$ ← root
  // $v$ will be always the last node added
  $v$ ← new Node()
  // the stack will keep track of previous values of $u$
  stack = new Stack()
  root.children.append($v$)
  for $i = 1 \to m$ do
    node ← new Node()
    if $x[i] = U$ then
      $v$.children.append(node)
      stack.push($u$)
      $u \leftarrow v$
    else if $x[i] = I$ then
      $v$.children.append(node)
    else if $x[i] = H$ then
      $u$.children.append(node)
    else if $x[i] = D$ then
      $u$.children.append(node)
      $u$ ← stack.pop()
    $v$ ← node
  return root
```

CHAPTER 5
EXPLORING OPTIMIZATIONS TO HETESIM FOR COMPUTING RELATEDNESS IN HETEROGENEOUS INFORMATION NETWORKS

The content of this chapter is in preparation for submission to a journal, with co-authors Chidozie Onyeze, David Kartchner, Stephen Allegri, Davi Nakajima-An, Evie Davalbhakta, and Cassie Mitchell.

5.1 Introduction

5.1.1 Background and Motivation

Heterogeneous information networks, or knowledge graphs, are valuable tools for collecting and analyzing insights from the vast number of papers published in the biomedical sciences. Informally, a heterogeneous information network is a directed graph in which each node corresponds to a biomedical concept and each directed edge encodes a relationship between concepts. Additionally, each node in the graph has an associated type. Heterogeneous information networks are further constrained by a schema which lists all possible edge types and the allowed node types which may serve as the source and target for a given edge type. Formal definitions for these concepts are given in subsection 5.1.2.

SemNet is a heterogeneous information network with approximately 300,000 nodes and 20,000,000 edges built from the abstracts of papers in the PubMed database [15]. SemNet has proven useful in ongoing unpublished research, as well as in a published study of link prediction for drug discovery [132]. However, long algorithm runtimes have limited the application of this valuable tool.

This work will build on the SemNet codebase with the goal of making revisions and introducing new algorithms that can reduce algorithm runtimes. Throughout this work, the version of SemNet described by Sedler and Mitchell in [15] will be referred to as SemNet version 1. The new version of SemNet, after the applying improvements described in this manuscript, will be referred to as SemNet version 2.

This work will focus on the similarity score HeteSim [14]. HeteSim-based similarity scoring on heterogeneous information networks has been successfully applied to multiple biomedical research problems [133, 134, 135, 136, 137, 138] Therefore, the implementation of a faster HeteSim scoring algorithm will have the potential for significant benefit to the biomedical research community.

Several techniques will be used to improve the performance of SemNet. First, investigation of runtimes in SemNet will highlight bottlenecks, especially the reliance on Neo4j to store the knowledge graph. Based on this insight, optimizations in data structures will lead to significant performance improvements.

Algorithmic improvements will also be investigated. In particular, approximation algorithms using randomness will be explored. An approximation algorithm is an algorithm which returns a value within a specified error (generally additive or multiplicative) of the true answer, with some known or bounded probability. The power of approximation algorithms lies in their ability, for some problems, to provide a fast approximation to a solution even when computing the exact solution requires exponential time (assuming $\mathbf{P} \neq \mathbf{NP}$). Though approximation algorithms have existed in the literature for some time, Garey, Graham, and Ullman [46] and Johnson [47] both introduced the idea formally in 1973 and 1974, respectively. Since then, the computer science and combinatorics literature has featured many advancements in the field of randomized approximation algorithms. For an overview of basic techninques and more recent results, see [48, 49, 50].

In addition to reducing the required computation time for HeteSim scoring in SemNet, this paper will also address a flaw in SemNet version 1, namely the reliance on ULARA [139] for aggregating HeteSim scores over multiple metapaths. subsection 5.2.1 will explain the flaw in ULARA and propose an alternative, which will be implemented and further discussed in subsequent sections.

Throughout this manuscript, we will consider metapaths between the node for Alzheimer's disease (CUI C0002395) and several possible source nodes. Specifically, we will consider insulin (CUI C0021641), hypothyroidism (CUI C0020676), and amyloid (CUI C0002716) as possible source nodes. These examples were selected because of their relevance to other ongoing work on Alzheimer's disease.

5.1.2 Definitions and Mathematical Preliminaries

In this section, we will formally define a schema and a knowledge graph / heterogeneous information network. A schema tells us which node and edge types may be present in our knowledge graph, while the knowledge graph tells us which relations apply to specific concepts nodes.

Definition 1. *A* schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ *is a set* $\mathcal{A}$ *of node types and a set* $\mathcal{R}$ *of relations. Each relation* $R \in \mathcal{R}$ *has a* source type $A \in \mathcal{A}$ *and a* target type $B \in \mathcal{A}$.

Definition 2. *Let* $S = (\mathcal{A}, \mathcal{R})$ *be a schema with* $|\mathcal{A}| > 1$ *. Then, a* heterogeneous information network *(also called a* knowledge graph*) is a directed graph* $G = (V, E)$ *with an object type mapping function* $\varphi : V \to \mathcal{A}$ *and a link type mapping function* $\psi : E \to \mathcal{R}$*. If* $e = (u, v) \in E$*, then the source type of* $\psi(e)$ *must be* $\varphi(u)$ *and similarly the target type of* $\psi(e)$ *must be* $\varphi(v)$*.*

Relations are a key concept in understanding knowledge graphs. We may understand both individual edges and entire metapaths as relations. We start by defining the simplest relation, the self relation.

Definition 3. *The relation* I *is the* self-relation. *So,* $a \xrightarrow{I} b$ *if and only if* $a = b$*. We also define the function* δ *by* $\delta(a, b) = 1$ *if* $a = b$ *and* $\delta(a, b) = 0$ *otherwise.*

We now define our primary object of study: the metapath. Note that the metapath may be viewed as a list of node and edge types or as the relation equivalent to the composition of all individual relations in the metapath.

Definition 4. *Let $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ be a schema. Then, a* metapath *$\mathcal{P}$ is a sequence of node and edge types, denoted $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$, with $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$. The* length *of $\mathcal{P}$ is l. Note that a metapath may also be understood as the composition of the relations given by its metaedges: $R = R_1 \circ R_2 \circ \cdots \circ R_l$. Let $p = a_1 a_2 \dots a_{l+1}$ with $a_i \in V$ and $(a_i, a_{i+1}) \in E$ be a path in G. Then, p is a* path instance *of the metapath $\mathcal{P}$ if $\varphi(a_i) = A_i \forall i \leq l+1$ and $\psi((a_i, a_{i+1})) = R_i \forall i \leq l$. We denote the fact that p is a path instance of $\mathcal{P}$ by $p \in \mathcal{P}$.*

Given these definitions, we are nearly ready to define the function of interest: $HeteSim$, which was defined by Shi et. al. [14]. We start by defining a function h which is a non-normalized version of HeteSim.

Definition 5. *Let $l > 0$. Let $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$. Let $\varphi(s) = A_1$ and $\varphi(t) = R_{l+1}$. Then the non-normalized HeteSim score between s and t with respect to the relevance path $\mathcal{P}$ is defined recursively as follows. When $R_1 \circ R_2 \circ \dots R_l \neq I$,*

$$\begin{aligned} &h(s, t | R_1 \circ R_2 \circ \cdots \circ R_l) \\ &= \frac{1}{|O(s|R_1)|\,|I(t|R_l)|} \sum_{a \in O(s|R_1)} \sum_{b \in I(t|R_l)} h(a, b | R_2 \circ R_3 \circ \cdots \circ R_{l-1}), \end{aligned} \tag{5.1}$$

where $O(s|R_1)$ is the set of out-neighbors of node s based on relation R_1, and $I(t|R_l)$ is the set of in-neighbors of node t based on the relation R_l.

In the base case, we define

$$h(a, b | I) = \delta(a, b). \tag{5.2}$$

Note that this definition only works for relevance paths of even length. We will need an extension for paths of odd length. We briefly explain the definition of HeteSim for odd paths here. For more detail, see Shi et. al. [14].

The basic idea to define h for paths of odd length is to transform those paths into paths of even length. Suppose we have a relevance path of odd length $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$. We now modify $\mathcal{P}$ by adding a new object type E and two new relation types R_E and R_F. We then define $\mathcal{P}' = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_{\frac{l+1}{2}-1}} A_{\frac{l+1}{2}} \xrightarrow{R_E} E \xrightarrow{R_F} A_{\frac{l+1}{2}+1} \xrightarrow{R_{\frac{l+1}{2}+1}} \dots \xrightarrow{R_l} A_{l+1}$. Additionally, in the underlying graph G, for any edge $g = (u, v)$ with $\psi(g) = R_{\frac{l+1}{2}}$, we add a new node, E_g and 2 new edges: $e_1 = (u, E_g)$ and $e_2 = (E_g, v)$. We additionally assign $\varphi(E_g) = E$, $\psi(e_1) = R_E$, and $\psi(e_2) = R_F$. This procedure allows us to transform any odd path into an even path, giving a definition for the non-normalized HeteSim score h for odd length paths.

As a final step, $HeteSim$ is normalized so that the normalized score for any two nodes lies in the interval $[0, 1]$. To do so, we will cast the problem in the language of transition matrices.

Definition 6. *Given a relation $A \xrightarrow{R} B$, let W_{AB} be an adjacency matrix between type A and type B. Let U_{AB} be W_{AB} normalized along each row vector. That is, U_{AB} is the transition probability matrix $A \rightarrow B$ based on relation R where each allowed transition is given equal probability. Similarly, let V_{AB} be a normalized form of the matrix W_{AB}, this time normalized along its column vectors. So, V_{AB} is the transition probability matrix for $B \rightarrow A$ based on relation R^{-1}. Note that $U_{AB} = V_{BA}^T$.*

Definition 7. *Given a metapath $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$, the* reachable probability matrix *PM for that metapath is given by*

$$PM_{\mathcal{P}} = U_{A_1A_2}U_{A_2A_3}\dots U_{A_lA_{l+1}}. \tag{5.3}$$

Note that $PM_{\mathcal{P}}(i, j)$ gives us the probability of object $i \in A_1$ reaching object $j \in A_{l+1}$ under the path $\mathcal{P}$, under the assumption that at each step all valid transitions have equal probability.

The following lemma is implicit in [14], but it is stated here for clarity.

Lemma 72. *Let $s \in A_1$, $t \in A_{l+1}$. Let $\mathcal{P} = (A_1 A_2 \ldots A_{l+1})$ be a metapath. Then,*

$$h(s, t|\mathcal{P}) = PM_{\mathcal{P}_L}(s, :)(PM_{\mathcal{P}_R^{-1}}(t, :))^T, \tag{5.4}$$

where $PM_{\mathcal{P}_L}(a, :)$ is used to denote the ath row of the matrix $PM_{\mathcal{P}}$, and $\mathcal{P} = \mathcal{P}_L \mathcal{P}_R$ is the decomposition of $\mathcal{P}$ into two paths of equal length.

Proof. First, notice that we only need to prove this result for even values of l. We proceed by induction.

In the base case, we have $l = 0$. This is the trivial metapath, and its corresponding relation is the self relation. We have

$$h(s, s) = \delta(s, s) = 1, \tag{5.5}$$

and

$$PM_{\mathcal{P}_L}(s, :)(PM_{\mathcal{P}_R^{-1}}(s, :))^T = 1 \cdot 1 = 1. \tag{5.6}$$

Therefore, the base case holds.

For the induction step, let $k \geq 2$ be an even integer. Assume that the lemma holds for all metapaths of length k. We will prove the lemma for paths of length $k + 2$. Beginning with the definition of h, we have

$$\begin{aligned} & h(s, t|R_1 \circ R_2 \circ \cdots \circ R_{k+2}) \\ &= \frac{1}{|O(s|R_1)|\,|I(t|R_{k+2})|} \sum_{a \in O(s|R_1)} \sum_{b \in I(t|R_{k+2})} h(a, b|R_2 \circ \cdots \circ R_{k+1}) \qquad (5.7) \\ &= \frac{1}{|O(s|R_1)|\,|I(t|R_{k+2})|} \sum_{a \in O(s|R_1)} \sum_{b \in I(t|R_{k+2})} PM_{\mathcal{P}'_L}(a, :) \left(PM_{(\mathcal{P}')_R^{-1}}(b, :)\right)^T, \qquad (5.8) \end{aligned}$$

where $\mathcal{P}' = R_2 \circ \cdots \circ R_{k+1}$, and the second equality follows from the induction hypothesis.

Recalling the interpretation of $PM_{\mathcal{P}}$ as the product of transition matrices, we see

$$\begin{aligned}
&\frac{1}{|O(s|R_1)|\,|I(t|R_{k+2})|} \sum_{a\in O(s|R_1)} \sum_{b\in I(t|R_{k+2})} PM_{\mathcal{P}'_L}(a,:) \left(PM_{(\mathcal{P}')_R^{-1}}(b,:)\right)^T \\
&= \sum_{a\in O(s|R_1)} \frac{1}{|O(s|R_1)|} PM_{\mathcal{P}'_L}(a,:) \sum_{b\in I(t|R_{k+2})} \frac{1}{|I(t|R_{k+2})|} \left(PM_{(\mathcal{P}')_R^{-1}}(b,:)\right)^T \qquad (5.9)\\
&= \left(U_{A_1A_2} PM_{\mathcal{P}'_L}(s,:)\right) \left(V_{A_{k+1}A_{k+2}} PM_{(\mathcal{P}\prime)_R^{-1}}(t,:)\right)^T \qquad (5.10)\\
&= PM_{\mathcal{P}_L}(s,:) \left(PM_{\mathcal{P}_R^{-1}}(t,:)\right)^T, \qquad (5.11)
\end{aligned}$$

which establishes the result. □

Finally, the HeteSim score is given by the cosine of the angle θ defined by vectors $PM_{\mathcal{P}_L}(s,:)$ and $PM_{\mathcal{P}_R^{-1}}(t,:)$.

Definition 8. *The* normalized HeteSim score *between two objects a and b based on the relevance path $\mathcal{P}$ is*

$$HS\,(s,t|\mathcal{P}) = cos(\theta) = \frac{PM_{\mathcal{P}_L}(s,:)(PM_{\mathcal{P}_R^{-1}}(t,:))^T}{\left|PM_{\mathcal{P}_L}(s,:)\right|\left|(PM_{\mathcal{P}_R^{-1}}(t,:))^T\right|}. \qquad (5.12)$$

The above definition uses the multiplication of transition matrices to obtain reachable probability matrices, which in turn give the HeteSim score with respect to a given metapath. We can recast this matrix multiplication in the language of random walks. Consider the example graph and metapath given in Figure 5.1. Beginning with node s, we assign the probability value 1, since this is the specified source node. Next, we distribute that probability among all neighbors of s with type A_2 joined by an edge of type R_1. These neighbors are a, b and c, and each of these three nodes gets labeled with the probability $1/3$. We repeat the same process with the neighbors of a, b, c having type A_3 and joined by an edge of type R_2. The probability $1/3$ assigned to node a is split between its neighbors d and f, with each neighbor receiving $1/6$. Node b has no eligible neighbors, and so its probability mass does not propagate to the next layer of the graph. Node c splits its probability mass of $1/3$

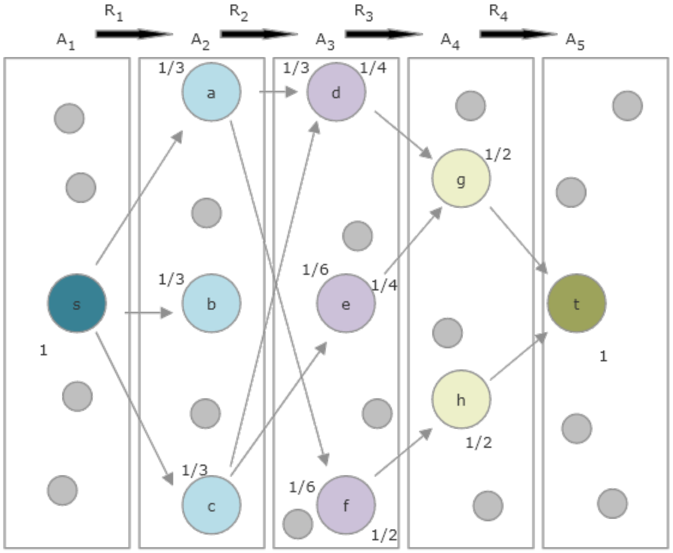


Figure 5.1: Example graph, metapath, and HeteSim computation.

between d and e. Therefore, d is labeled with probability mass $1/3$, with $1/6$ coming from a and $1/6$ from c. Node e only receives probability mass from c and is therefore labeled with $1/6$. Similarly, node f receives probability mass only from a, and therefore has total probability mass $1/6$. This computation, which is equivalent to the matrix multiplication described above, gives

$$PM_{\mathcal{P}_L}(s,:) = \begin{bmatrix} 1/3 \\ 1/6 \\ 1/6 \end{bmatrix}. \tag{5.13}$$

To obtain $PM_{\mathcal{P}_R^{-1}}(t:)$, we repeat the same procedure on the second half of the metapath, this time working backwarde towards A_3 from t. To start, t gets probability mass label 1. That probability is split among its 2 neighbors in A_4, giving g and h each probability mass $1/2$. The mass of g is split evenly among d and e, so both of these nodes have probability mass $1/4$. All of the probability mass of h goes to f, giving f a probability mass $1/2$.

Note that we have now labeled nodes d, e and f twice, once from the left and once from the right. While the labels from the left gave us $PM_{\mathcal{P}_L}(s\ :)$, the labels from the right give

$$PM_{\mathcal{P}_R^{-1}}(t,:) = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/2 \end{bmatrix}. \tag{5.14}$$

Finally, we can compute

$$\text{HS}\,(s,t|\mathcal{P}) = \frac{PM_{\mathcal{P}_L}(s,:)(PM_{\mathcal{P}_R^{-1}}(t,:))^T}{\left|PM_{\mathcal{P}_L}(s,:)\right|\left|(PM_{\mathcal{P}_R^{-1}}(t,:))^T\right|} = \frac{1/4}{1/2 \cdot \sqrt{6}/4} = \frac{\sqrt{6}}{3}. \tag{5.15}$$

5.1.3 Overview of SemNet's existing HeteSim implementation

The implementation of HeteSim in SemNet version 1 includes more than just the single-metapath HeteSim computation described in subsection 5.1.2. In SemNet, HeteSim is not just used to give a score of the relatedness of two specific nodes with respect to a fixed metapath. Instead, it is used as a tool to rank a set of candidate source nodes based on their relatedness to a fixed target node.

Figure 5.2 gives an overview of this ranking algorithm as it exists in SemNet version 1. As input, the algorithm accepts a set of candidate source nodes S and a single target node t. In step 1, the set of all metapaths $\mathcal{MP}$ which have an instance joining some element of S to t is enumerated. This enumeration depends upon the underlying knowledge graph, which is stored in Neo4j. Step 2 is the computation of HeteSim scores for each triple (s, t, m) for $s \in S, m \in \mathcal{MP}$. For any fixed metapath $m \in \mathcal{MP}$, the results from step 2 induce a ranking on the source nodes S by HeteSim score. Step 3 takes these $|\mathcal{MP}|$ rankings and combines them to form a single ranking using a technique called ULARA (see [139]). Finally, this combined ranking is returned to the user and is used as an indication of which nodes from S are most closely related to t.

In this work, we will keep the overall structure of the HeteSim algorithm outlined in

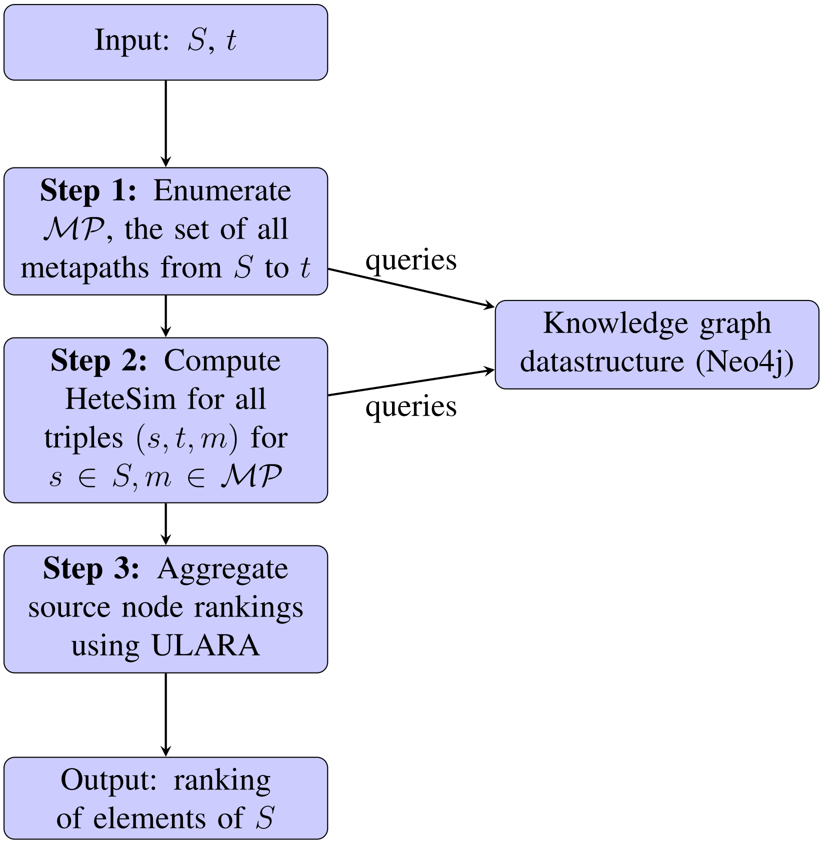

Input: S, t
Step 1: Enumerate $\mathcal{MP}$, the set of all metapaths from S to t
queries
Knowledge graph datastructure (Neo4j)
Step 2: Compute HeteSim for all triples (s, t, m) for $s \in S, m \in \mathcal{MP}$
queries
Step 3: Aggregate source node rankings using ULARA
Output: ranking of elements of S

Figure 5.2: Overview of SemNet version 1 HeteSim implementation.

Figure 5.2, but will make several substantial changes to the various subroutines. First, we will replace the knowledge graph data structure using Neo4j with one based solely on Python dictionaries. Second, we will explore algorithms using randomization as candidate replacements for Step 2. Finally, we will discuss a flaw in ULARA and will replace Step 3 with the generation of a ranking based on mean HeteSim score over all metapaths. We will also explore an approximate version of Step 3 where only a subset of metapaths are selected for inclusion in the mean.

5.2 Methods

5.2.1 A new method for combining HeteSim scores from multiple metapaths

SemNet version 1 outputs a ranking of many candidate source nodes with respect to a fixed target node. This ranking is intended to reflect the overall relatedness of each source node to the target node. SemNet version 1 computes the HeteSim scores for all requested source nodes and for all possible metapaths (up to some length bound) joining those source nodes to the target node. Each metapath induces a ranking of the source nodes according to HeteSim score. In order to combine these many rankings into a single ranking, SemNet version 1 uses a technique called ULARA (Unsupervised Learning Algorithm for Rank Aggregation) [139]. Due to a flaw in ULARA, this work replaces ULARA with a ranking based on mean HeteSim scores.

Before proceeding to describe the flaw in ULARA, it is necessary to give some background. We explain ULARA in the full generality with which it is presented in [139] in order to explain the flaw, but note that SemNet version 1 does not require the full generality of ULARA and may be thought of as using a special case of ULARA. Let X be a set of objects to be ranked, and let Q be a set of valid queries. Let $x, x' \in X$, $q \in Q$. Let $r : Q \times X \to \mathbb{N}$ be a ranking function, so that $r(q,x) < r(q,x')$ means that x has a higher ranking than x' with respect to the query q.

Let $N \in \mathbb{N}$. Given a set of ranking functions $\{r_i\}_{i=1}^N$, ULARA produces a ranking function of the form

$$R(q,x) = \sum_{i=1}^{N} w_i r_i(q,x), \tag{5.16}$$

for some real numbers $\{w_i\}_{i=1}^N$ satisfying $0 \leq w_i \leq 1$ for all $1 \leq i \leq N$ and $\sum_{i=1}^N w_i = 1$. The value of each w_i is determined by an optimization problem.

Let

$$\mu(q,x) = \frac{\sum_{i:r_i(q,x)\leq\kappa_i} r_i(q,i)}{|\{i : r_i(q,x) \leq \kappa_i\}|}, \tag{5.17}$$

where κ_i is a threshold which allows for the possibility that not every ranking function returns a rank for every $x \in X$. The function $\mu(q, x)$ is intended to represent the mean ranking of element x with respect to query q over all ranking functions r_i. Let

$$\sigma_i = \left(r_i(q, x) - \mu(q, x)\right)^2 . \tag{5.18}$$

This variance-like function is used to measure the agreement of ranking functions with each other, with the goal of giving ranking functions that agree with the mean a higher weight. Let

$$\delta_i(q, x) = w_i \sigma_i(q, x). \tag{5.19}$$

We can now finally state the optimization problem at the center of ULARA:

$$\operatorname*{arg\,min}_{w_1,\ldots,w_N} \sum_{q \in Q} \sum_{x \in X} \sum_{i=1}^{N} \delta_i(q, x), \tag{5.20}$$

subject to the constraints

$$\sum_{i=1}^{n} w_i = 1 \text{ and } \forall i, w_i > 0. \tag{5.21}$$

ULARA solves this optimization problem using gradient descent, the details which are not relevant here. The flaw in ULARA can be seen simply by examining the optimization problem itself. Let

$$a_i = \sum_{q \in Q} \sum_{x \in X} \sigma_i(q, x). \tag{5.22}$$

Then, the optimization problem becomes

$$\operatorname*{arg\,min}_{w_1,\ldots,w_N} \sum_{i=1}^{N} w_i a_i, \tag{5.23}$$

subject to the constraints

$$\sum_{i=1}^{N} w_i = 1 \text{ and } \forall i, w_i > 0. \tag{5.24}$$

Let j be such that $a_j = \min_i a_i$. Then, an optimal solution is given by

$$w_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \tag{5.25}$$

Further, the solution is unique if a_j is the unique minimum of the set $A = \{a_1, \ldots, a_N\}$. The case where the optimization problem does not have a unique solution is not mentioned in [139], and it seems this case should be rare in practice. Therefore, any unique optimal solution of the ULARA optimization problem places all of the available weight on a single ranking function. That is, ULARA does not give an aggregation of ranking functions; it simply selects a single raking function which shows most agreement with the others. In the language of SemNet, this means that only one metapath is used to give the final ranking of source nodes.

In the case of SemNet version 1, the implementation of ULARA was reexamined, and a bug was found that resulted in the algorithm terminating before the gradient descent had converged. As a result, a linear combination of multiple ranking functions (with nonzero coefficients) was actually returned, and multiple metapaths therefore are reflected in the rankings given by SemNet. This likely explains the observed utility of SemNet version 1 in spite of the flaw in ULARA.

As a replacement for ULARA, in SemNet version 2 the mean HeteSim score of a source node with respect to all metapaths is used to generate a ranking of source nodes.

5.2.2 Computational analysis of HeteSim runtimes: SemNet version 1

To better understand the run time of the HeteSim computation, the Python module time [140] was used to record the time required to compute HeteSim for each of the metapaths from the studied source nodes to Alzheimer's Disease. Additionally, the total time spent on the required Neo4j queries was recorded for each metapath. This allows separate analysis of the time required to query the graph and the time required to perform the HeteSim computations.

5.2.3 Development, implementation, and testing of algorithms

The core development work for this project can be divided into three general categories: reimplementation of the knowledge graph data structure, development and implementation of algorithms, and testing.

Knowledge graph data structure

SemNet version 1 used Neo4j to store the knowledge graph. After preliminary testing showed that Neo4j was likely a significant bottleneck, the knowledge graph data structure was re-implemented using nested Python dictionaries. Because these dictionaries use hashing for lookup, they have average lookup time $O(1)$ (see, e.g. [141]). As a result, dictionaries allow for quickly examining the neighborhood of a node in the knowledge graph, restricted to edge and node types of interest. Consequently, it is also efficient to traverse paths within the graph.

After testing on artificial examples, a knowledge graph object was built using an edge set derived from SemMedDB. This is an updated version of the edge set, and is not identical to the edge set from SemNet version 1.

Development of approximation algorithms

In addition to the data structure improvements, approximation algorithms based on randomization were explored as a way of further increasing performance. In particular, approximation algorithms were investigated as possible replacements for the computation of HeteSim on a single metapath (step 2 in Figure 5.2) and aggregation of rankings (step 3 in Figure 5.2).

Implementation and Testing

All code with implemented in Python 3. Testing was performed using Jupyter Notebook 5.5.0 [142] and Python 3.6.10 [143]. All code was run on a server with 1 NVIDIA TESLA v100 GPU with 32 GB RAM and 48 core CPU with 320 GB RAM.

For all code not involving randomization, the correctness of implementation was assessed using unit tests, which may be found in the source code repository. The one randomized function of significant complexity, randomized pruned HeteSim, was assessed on artificially-constructed example knowledge graphs. These examples were constructed by hand by the authors, and the full examples may be found in the source code repository. The algorithm was run on each graph 100 times with parameters $\epsilon = 0.05$ and $r = 0.95$. As with the SemNet version 1 implementation, the speed of the new implementation was assessed using the Python time module [140].

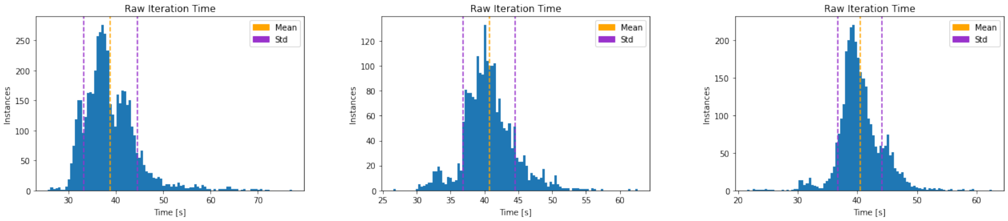
5.3 Results

5.3.1 Computational Analysis of HeteSim runtimes: SemNet version 1

For each of the three source nodes, the run time of the HeteSim computation on each metapath from the source node to Alzheimer's disease was recorded. The computation time results are given in Table 5.1, and the distribution of runtimes is depicted graphically in Figure 5.3. Note that SemNet version 1 incorporated parallelization, allowing multiple HeteSim computations for different metapaths to occur simultaneously. Therefore, the computation time per metapath times the number of metapaths does not equal the total computation time. Time required for the Neo4j graph queries was also measured and is displayed in Figure 5.4.

source node	insulin	hypothyroidism	amyloid
number of metapaths	4873	2148	3095
total computation time (min)	79.1	36.5	52.3
computation time per metapath (s)	38.9 ± 5.7	40.7 ± 3.8	40.5 ± 3.7
Neo4j query time, per metapath (s)	38.0 ± 4.8	39.9 ± 3.2	39.8 ± 3.2
time per metapath, excluding query (s)	0.9 ± 2.5	0.8 ± 1.8	0.6 ± 1.6

Table 5.1: SemNet version 1 HeteSim computation times for all metapaths between each of the three source nodes and Alzheimer's disease. Per-metapath values are given a mean ± standard deviation.



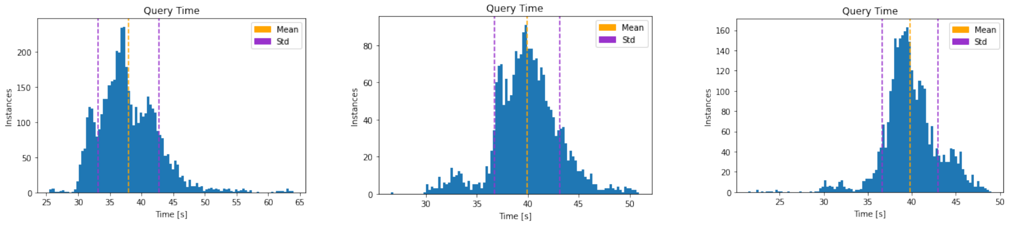
(a) Insulin (b) Hypothyroidism (c) Amyloid

Figure 5.3: Distribution of SemNet version 1 HeteSim computation times for all metapaths joining the given source node and Alzheimer's disease.

5.3.2 Algorithms

In this section, we present several algorithms for computing HeteSim and variants. Proofs of correctness are also given where appropriate.

We consider two main algorithms for computing HeteSim on a single metapath and two algorithms for aggregating HeteSim scores across multiple metapaths. For computing HeteSim on a single metapath, we consider the deterministic HeteSim algorithm used in SemNet version 1 and a new algorithm, randomized pruned HeteSim. For aggregating HeteSim scores over multiple metapaths we consider computing the exact mean over all metapaths and also an algorithm which approximates the mean by taking the mean over a random subset of metapaths. We also combine these algorithms to get 3 distinct algorithms for computing (an approximation to) the mean HeteSim score: deterministic HeteSim with exact mean, deterministic HeteSim with approximate mean, and randomized pruned HeteSim with approximate mean. Using approximate mean HeteSim as an example, an overview of



(a) Insulin (b) Hypothyroidism (c) Amyloid

Figure 5.4: Distribution of Neo4j query times in SemNet version 1 HeteSim computation for all metapaths joining the given source node and Alzheimer's disease.

the new algorithm structure, emphasizing changes, is shown in Figure 5.5.

Deterministic HeteSim

For completeness, we summarize the deterministic algorithm for computing HeteSim. While this same algorithm is used in SemNet version 1, SemNet version 2 significantly improves the implementation by changing the underlying data structure for the knowledge graph. Where version 1 used Neo4j, version 2 uses a knowledge graph object built from Python dictionaries.

Given a source node s, a target node t, and a metapath $\mathcal{P}$, the deterministic HeteSim algorithm begins by splitting $\mathcal{P}$ into two halves: $\mathcal{P}_L$ and $\mathcal{P}_R$. If $\mathcal{P}$ has odd length, the construction described in subsection 5.1.2 is applied before constructing $\mathcal{P}_L$ and $\mathcal{P}_R$. An identical subroutine is now applied to both $\mathcal{P}_L$ and $\mathcal{P}_R^{-1}$. The following exposition will consider only $\mathcal{P}_L$.

Recall that the algorithm must compute $PM_{\mathcal{P}_L}(s,:)$, which may be understood as the probability that a random walk along the given metapath starting from s arrives at a given node in $A_{l/2}$. The algorithm iteratively computes the probability of arriving at each node in A_i for step i of the metapath for $1 \leq i \leq l/2$.

Let $v_i(x)$ be the probability of arriving at node x of type A_i at step i of the metapath.

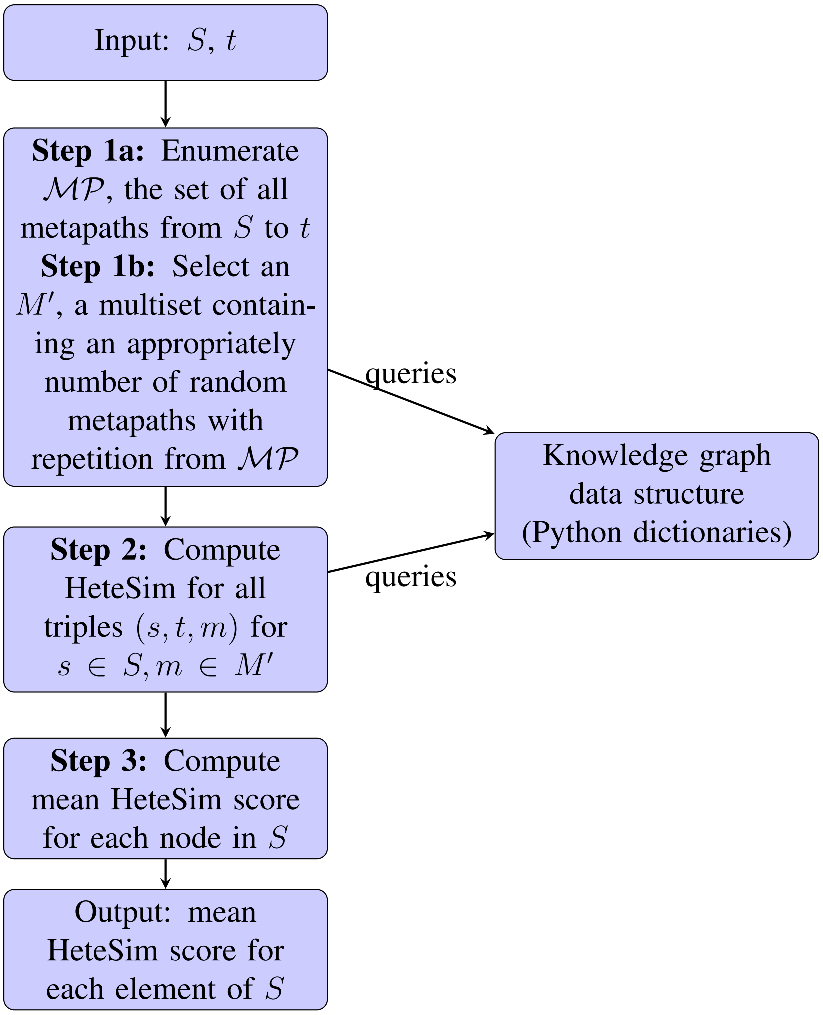

Input: S, t
Step 1a: Enumerate $\mathcal{MP}$, the set of all metapaths from S to t
Step 1b: Select an M', a multiset containing an appropriately number of random metapaths with repetition from $\mathcal{MP}$
queries
Knowledge graph data structure (Python dictionaries)
Step 2: Compute HeteSim for all triples (s, t, m) for $s \in S, m \in M'$
queries
Step 3: Compute mean HeteSim score for each node in S
Output: mean HeteSim score for each element of S

Figure 5.5: Overview of SemNet version 2 approximate mean HeteSim implementation.

To compute v_i for $i > 1$, note that it is sufficient to know v_{i-1}, as

$$v_i(x) = \sum_{y \in \delta^-_{R_{i-1}}(x)} \frac{1}{\delta^+_{R_{i-1}}(y)} v_{i-1}(y). \tag{5.26}$$

Therefore, beginning with $v_1(s) = 1$, the algorithm iteratively computes $v_2, \ldots, v_{l/2}$, and $PM_{\mathcal{P}_L} = v_{l/2}$.

After completing the analogous computation for $\mathcal{P}_R^{-1}$, the algorithm returns

$$\frac{PM_{\mathcal{P}_L}(a,:)(PM_{\mathcal{P}_R^{-1}}(b,:))^T}{\left|PM_{\mathcal{P}_L}(a,:)\right|\left|(PM_{\mathcal{P}_R^{-1}}(b,:))^T\right|}. \tag{5.27}$$

Pseudocode is given in Algorithm 3 and Algorithm 4.

Algorithm 3 HeteSim

```
Input: start node $s$, end node $t$, metapath $\mathcal{P}$ of even length {odd relevance paths must be
  preprocessed}
Output: HeteSim score
  Construct $\mathcal{P}_L, \mathcal{P}_A$
  $v_L \leftarrow oneSidedHS(s, \mathcal{P}_L)$
  $v_R \leftarrow oneSidedHS(t, \mathcal{P}_R^{-1}$
  return $(v_L \cdot v_R)/(|v_L||v_R|)$
```

Algorithm 4 oneSidedHS subroutine

```
Input: start node $s$, metapath $\mathcal{P}$
Output: vector $v_{\text{length}(\mathcal{P})}$, the one-sided HeteSim vector
  for $i = 1$ to length($\mathcal{P}$)/2 do
    $v_i \leftarrow [0]^{\|A_i\|}$ {Vector of zeros, indexed by elements of $A_i$}
  $v_1[s] = 1$
  for $i = 2$ to length($\mathcal{P}$) do
    for $x \in A_i$ do
      $v_i[x] \leftarrow \sum_{y \in \delta^-_{R_{i-1}}} \frac{1}{\delta^+_{R_{i-1}}[y]} v_{i-1}[y]$
  return $v_{\text{length}(\mathcal{P})}$
```

Pruning the graph

Given a metapath $\mathcal{P}_{\mathcal{L}} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_{\frac{l}{2}-1}} A_{\frac{l}{2}}$, a random walk starting from $s \in A_1$ may arrive at node $u \in A_i$ such that the out degree of u along edges of type R_i is 0. Informally speaking, the random walk has reached a dead end. As an example, node b in Figure 5.1 is a dead end . The presence of these dead ends reduces the probability that a random walk starting from s actually reaches any node of type $A_{\frac{l}{2}}$. In fact, we can construct graphs that make this probability arbitrarily small. Therefore, a basic random

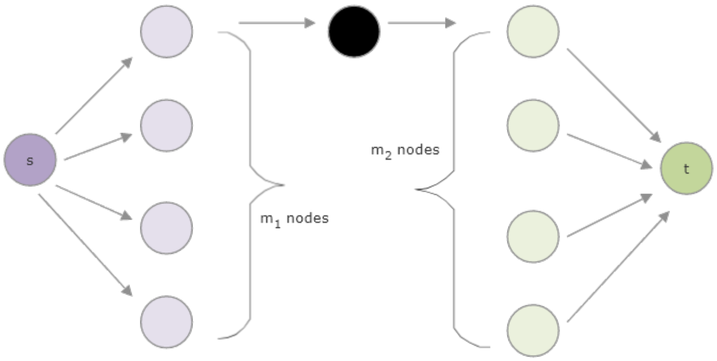


Figure 5.6: An example knowledge graph. Here, we use the convention that nodes are organized by type into vertical columns in the order that they appear in the metapath. We also only show edges that may appear in some metapath instance. This example has $m_1 - 1$ dead end nodes on the left and $m_2 - 1$ dead end nodes on the right. The HeteSim score of s and t with respect to the metapath is 1 for all values of m_1 and m_2.

walk algorithm may have arbitrarily long run time. We will address this limitation by defining a new but closely related quantity: pruned HeteSim.

Before proceeding, we provide two additional examples to explore the effect of dead ends on HeteSim scores. In Figure 5.6, a simple knowledge graph is shown, organized according to one metapath. The nodes are organized into columns by type, and the columns are given in the order that those types appear in the metapath. The only edges shown are those which appear in some instance of the metapath. This graph has $m_1 - 1$ dead end nodes on the left-hand side and $m_2 - 1$ dead end nodes on the right-hand side. We can compute its HeteSim score as follows.

$$\text{HS}\left(s, t|\mathcal{P}\right) = \frac{1 \cdot 1}{1 \cdot 1} = 1. \tag{5.28}$$

Note that this score does not change with m_1 or m_2. In particular, the HeteSim score with the given graph is identical to the HeteSim score when all dead ends are removed from the graph. As we will later see, this result generalizes to all metapaths of length less than or equal to 4.

In contrast, the metapath and knowledge graph depicted in Figure 5.7 create a situation

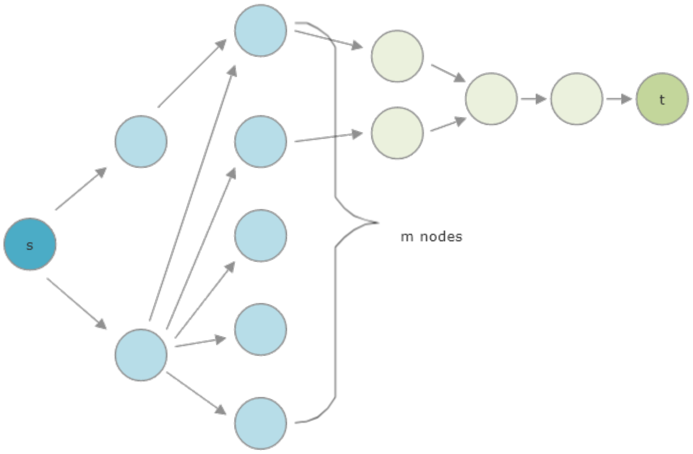


Figure 5.7: An example metapath and knowledge graph, drawn with the same conventions as in Figure 5.6. Note that, in this example, the removal of dead ends does change the HeteSim score.

where the removal of dead ends does change the HeteSim score. If we take $m = 2$, then we have removed all dead end nodes. In this case, the HeteSim score is

$$\text{HS}\left(s, t|\mathcal{P}\right) = \frac{\begin{bmatrix} 3/4 & 1/4 \end{bmatrix}\left(\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}\right)^T}{\left|\begin{bmatrix} 3/4 & 1/4 \end{bmatrix}\right|\left|\begin{bmatrix} 1/2 & 1/2 \end{bmatrix}\right|} = \frac{1/2}{\sqrt{5/8}\sqrt{1/2}} = \frac{2\sqrt{5}}{5}. \tag{5.29}$$

If we instead take $m = 3$, then the HeteSim score is $\frac{5\sqrt{34}}{34}$, and, in the limit as $m \to \infty$, the HeteSim score approaches $\frac{\sqrt{2}}{2}$.

We now introduce a new score: Pruned HeteSim. This new score is identical to HeteSim on relevance paths of length at most 4. To rigorously define Pruned HeteSim, we must first formally define a dead end node at step i of a given metapath and with respect to nodes s and t.

Let $G = (V, E)$ be a heterogeneous information network, and let $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ be a metapath in G. Let $s \in V$ with $\psi(s) = A_1$ and $t \in V$ with $\psi(t) = A_l$. Let C_1 be the set of nodes of type $A_{l/2}$ reachable from s along metapath $\mathcal{P}_L$. Similarly, let C_2 be the set of nodes of type $A_{l/2}$ reachable from t along metapath $\mathcal{P}_R^{-1}$. Let $C = C_1 \cap C_2$, and label the elements of C so that $C = \{c_1, c_2, \ldots, c_j\}$. For $i \leq j$, let X_i be the event that a random walk starting at s along $\mathcal{P}_L$ ends at node c_i. Similarly, let Y_i be the event that a random walk starting at t along $\mathcal{P}_R^{-1}$ ends at node c_i. Let $x_i = P\left(X_i\right)$ and $y_i = P\left(Y_i\right)$. Let $x = (x_1, x_2, \ldots, x_j)$ and let $y = (y_1, y_2, \ldots, y_j)$.

Let Z be the event that a random walk starting from s along $\mathcal{P}_L$ reaches some node in C. Similarly, let W be the event that a random walk starting from t along $\mathcal{P}_R^{-1}$ reaches some node in C.

Definition 9. *For a node v belonging to any of $A_1, A_2, \ldots, A_{l/2}$, we define a dead-end as follows. Let metapath $\mathcal{P}$ and source node s be fixed. Let A be the event that a random walk beginning from s and following metapath $\mathcal{P}_L$ contains node v at step i (so that the type of v is A_i). Then, v is a* dead end *at step i of metapath $\mathcal{P}$ and with respect to source node s if and only if $P\left(Z|A\right) = 0$. For a node w belonging to any of $A_{l/2+1}, \ldots, A_{l+1}$, the definition is analogous. Let metapath $\mathcal{P}$ and target node t be fixed. Let B be the event that a random walk starting from t and following metapath $\mathcal{P}_R^{-1}$ contains node w at step i. Then, w is a* dead end *with respect to step i of metapath $\mathcal{P}$ and target node t if and only if $P\left(W|B\right) = 0$.*

For fixed nodes s, t and fixed metapath $\mathcal{P}$, let D_i be the set of dead end nodes at step i of metapath $\mathcal{P}$ with respect to source node s and target node t.

Informally, this definition means that a node v is a dead end at step i of a metapath if no random walk which reaches the set of central nodes C has v as its ith node.

Recall that non-normalized HeteSim is defined by

$$\begin{aligned} & h(s,t|R_1 \circ R_2 \circ \cdots \circ R_l) \\ & = \frac{1}{|O(s|R_1)|\,|I(t|R_l)|} \sum_{a \in O(s|R_1)} \sum_{b \in I(t|R_l)} h(a,b|R_2 \circ R_3 \circ \cdots \circ R_{l-1}), \end{aligned} \tag{5.30}$$

where $O(s|R_1)$ is the set of out-neighbors of node s based on relation R_1, and $I(t|R_l)$ is the set of in-neighbors of node t based on the relation R_l.

To define the non-normalized version of pruned HeteSim, we simply exclude dead end nodes from the sets of neighbors.

Definition 10. *Let $\mathcal{P} = R_1 \circ R_2 \circ \cdots \circ R_l$ be a metapath in some graph G. Let s, t belong to the vertex set of G, and let D_i be the set of dead end nodes at step i of metapath $\mathcal{P}$. Then, the non-normalized pruned HeteSim score is given by*

$$\begin{aligned} & g(s, t|R_1 \circ R_2 \circ \cdots \circ R_l) \\ & = \frac{1}{|O(s|R_1) \setminus D_1| \, |I(t|R_l) \setminus D_l|} \sum_{a \in O(s|R_1) \setminus D_1} \sum_{b \in I(t|R_l) \setminus D_l} h(a, b|R_2 \circ R_3 \circ \cdots \circ R_{l-1}), \end{aligned} \tag{5.31}$$

where $O(s|R_1)$ is the set of out-neighbors of node s based on relation R_1, and $I(t|R_l)$ is the set of in-neighbors of node t based on the relation R_l.

The normalization of pruned HeteSim proceeds exactly like that for HeteSim. We obtain a restricted adjacency matrix $W'_{AB,i}$ for the relation $A \xrightarrow{R_i} B$ by removing any 1's in W_{AB} corresponding to a dead end node in B at step i of the metapath. As before, we normalize $W'_{AB,i}$ along its row vectors to obtain $U'_{AB,i}$. As before, we can obtain a reachable probability matrix by multiplying the normalized restricted adjacency matrices:

$$PM'_{\mathcal{P}} = U'_{A_1 A_2, 2} U'_{A_2 A_3, 3} \dots U'_{A_l A_{l+1}, l+1}. \tag{5.32}$$

Definition 11. *The* normalized pruned HeteSim score *is given by*

$$\mathit{PHS}\,(a, b|\mathcal{P}) = \frac{PM'_{\mathcal{P}_L}(a,:)(PM'_{\mathcal{P}_R^{-1}}(b,:))^T}{\sqrt{\left|PM'_{\mathcal{P}_L}(a,:)\right| \left|(PM'_{\mathcal{P}_R^{-1}}(b,:))^T\right|}}. \tag{5.33}$$

Note that, for metapaths with no repeated node types, pruned HeteSim may be computed by simply removing all dead end nodes from the graph and then computing HeteSim on this pruned graph.

Importantly, pruned HeteSim has value equal to plain HeteSim for metapaths of length at most 4. Since these shorter paths are often the ones of most interest in small-diameter knowledge graphs, pruned HeteSim may be thought of as a replacement for HeteSim in these circumstances.

Additionally, note that Equation 5.33 gives rise to a deterministic algorithm for computing pruned HeteSim, much like the deterministic algorithm for HeteSim. The algorithm now requires 2 passes over the data structure. In the first pass over the data, dead ends are identified. In a second pass, Equation 5.33 allows for the computation of the non-normalized pruned HeteSim score. Normalization is applied as the final step. Because our computational focus in this manuscript is on short paths of length at most four, and because HeteSim and pruned HeteSim have the same values for paths of length at most four, we do not pursue the deterministic algorithm for pruned HeteSim further. For these short paths, a deterministic computation of HeteSim is faster than a deterministic computation of pruned HeteSim.

Theorem 73. *Let* $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$ *be a metapath with length* $l \leq 4$*. Then,*

$$\textit{PHS}\left(s, t|G, \mathcal{P}\right) = \textit{HS}\left(s, t|G, \mathcal{P}\right). \qquad (5.34)$$

Proof. First, note that we only need to consider metapaths with even length, as odd metapaths will simply be transformed to even length metapaths before HeteSim is computed. Next, note that the result is trivial for metapaths with length 2, as these can have no dead ends. We may therefore focus only on the case where the metapath has length 4.

Let $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3 \xrightarrow{R_3} A_4 \xrightarrow{R_4} A_5$ be a metapath in G. Note that there can be no dead ends of type A_3. Additionally, if s or t is a dead end, then HS $\left(s, t|G, \mathcal{P}\right) = 0 =$ PHS $\left(s, t|G, \mathcal{P}\right)$. Therefore, we may assume that all dead ends are of type A_2 or A_4.

Recall that X_i is the event that a random walk in G from s reaches node c_i, and similarly Y_i is the event that a random walk in G starting at t arrives at node c_i. Let X_i' be the event that a random walk in G' along metapath $\mathcal{P}_L$ starting from s arrives at node c_i. Similarly let Y_i' be the event that a random walk in G' along metapath $\mathcal{P}_R^{-1}$ arrives at node c_i. Let

p_L be the probability that a random walk starting from s arrives at a dead end node in A_2. Similarly, let p_R be the probability that a random walk beginning at t will arrive at a dead end in A_4. Note that, once a random walk has reached a non-dead end node of type A_2 or A_4, that random walk must reach some node of type A_3. Therefore,

$$P\left(X_i\right) = (1 - p_L)P\left(X_i'\right) \tag{5.35}$$

and

$$P\left(Y_i\right) = (1 - p_R)P\left(Y_i'\right). \tag{5.36}$$

Letting $x_i = P\left(X_i\right)$, $y_i = P\left(Y_i\right)$, $x_i' = P\left(X_i'\right)$, and $y_i' = P\left(Y_i'\right)$, observe

$$\begin{aligned}
\text{HS}\left(s, t|G, \mathcal{P}\right) &= \frac{\sum_{i=1}^{k} x_i y_i}{\sqrt{\sum_{i=1}^{k} x_i^2 \sum_{i=1}^{k} y_i^2}} && (5.37)\\
&= \frac{\sum_{i=1}^{k}(1-p_L)x_i'(1-p_R)y_i'}{\sqrt{\sum_{i=1}^{k}(1-p_L)^2(x_i')^2 \sum_{i=1}^{k}(1-p_R)^2(y_i')^2}} && (5.38)\\
&= \frac{\sum_{i=1}^{k} x_i' y_i'}{\sqrt{\sum_{i=1}^{k}(x_i')^2 \sum_{i=1}^{k}(y_i')^2}} && (5.39)\\
&= \text{PHS}\left(s, t|G, \mathcal{P}\right). && (5.40)
\end{aligned}$$

□

Pruned HeteSim

We now present alternate algorithm for computing a variant of the HeteSim score. This algorithm is much more computationally tractable, and we have shown that the HeteSim and pruned HeteSim scores are identical for relevance paths of length at most 4.

Let $\mathcal{P}$ be a metapath, and let s and t be source and target notes, respectively. Let N be a positive integer, the required value of which will be determined later. Starting from s the algorithm takes N random walks along $\mathcal{P}_R$, never visiting any node that has been marked as a dead end for the current step of the metapath. At any point, if the algorithm encounters a dead end, it marks the current node as a dead end for the current step of the metapath and then retraces its steps until a non-dead end node is reached, marking dead ends along the way as necessary. Note that any dead end at a given step in the metapath will only need to be marked once, and the algorithm will avoid it for all future random walks. The same algorithm is repeated along metapath $\mathcal{P}_L^{-1}$ starting from t.

The frequency vectors of the terminal nodes of the random walks give an approximation for $PM'_{\mathcal{P}_L}$ and $PM'_{\mathcal{P}_R^{-1}}$, which are used to approximate the pruned HeteSim score. Psuedocode is given in Algorithm 5, Algorithm 6, and Algorithm 7. Analysis of the algorithm, determination of N, and a formal proof of correctness are given in subsubsection 5.3.2.

Algorithm 5 Randomized Pruned HeteSim

```
Input: start node $s$, end node $t$, relevance path $\mathcal{P}$ of even length, error tolerance $\epsilon$, success
  probability $r$ {odd relevance paths must be preprocessed}
Output: approximate HeteSim score
  $S \leftarrow$ breadthFirstSearch($s$, $\mathcal{P}_L$)
  $T \leftarrow$ breadthFirstSearch($t$, $\mathcal{P}_R^{-1}$)
  $k \leftarrow |S \cup T|$
  $c \leftarrow (5 + 4\sqrt{2})/4$
  $C \leftarrow 2(c + \sqrt{c^2 + 2\epsilon})^2 + \epsilon(c + \sqrt{c^2 + 2\epsilon})$
  $N \leftarrow \lceil \frac{C}{\epsilon^2} \rceil k \ln(4k/(1 - r))$
  return RandomizedPrunedHeteSimGivenN($s$, $t$, $\mathcal{P}$, $N$)
```

Runtime Analysis of the Pruned HeteSim Algorithm

We now provide guarantee on the number of random walks required to approximate pruned HeteSim with a given error tolerance ϵ and success probability r.

Let $\mathcal{S}_k = \{v \in \mathbb{R}^k : \sum_i v_i = 1 \text{ and } v_i \geq 0\}$. We consider arbitrary $v, w \in \mathcal{S}_k$ for fixed k where $v = PM'_{\mathcal{P}_L}(s, :)$ and $w = PM'_{\mathcal{P}_R^{-1}}(t, :)$. We will show that if all the entries in the vectors are sufficiently close to their true value, then the cosine will be sufficiently close to the true value. We consider $\hat{v}$, a random approximation of v after some number of steps. Notice that $\hat{v} = v + \lambda$ where $\lambda \in \mathbb{R}^k$ such that $\sum_{i=1}^{k} \lambda_i = 0$ and $v_i + \lambda_i \geq 0$ (since $\hat{v}$ is

Algorithm 6 RandomizedPrunedHeteSimGivenN subroutine

```
Input: start node $s$, end node $t$, relevance path $\mathcal{P}$ of even length, number of iterations $N$
Output: approximate HeteSim score
  for $i = 0$ to length($\mathcal{P}_L$)) do
    B[i] ← $\emptyset$
  $v_L \leftarrow [0]^k$ {array of 0's indexed by elements of $K$}
  $v_R \leftarrow [0]^k$
  {random walks from s}
  for $n = 1$ to $N$ do
    $(B, x) \leftarrow$ restrictedRandomWalkOnMetapath(s, $\mathcal{P}_L$, B)
    $v_L[x] = v_L[x] + 1$
  {random walks from t}
  for $i = 0$ to length($\mathcal{P}_L$)) do
    B ← $\emptyset$
  for $n = 1$ to $N$ do
    $(C, x) \leftarrow$ restrictedRandomWalkOnMetapath(t, $\mathcal{P}_R^{-1}$, B)
    $v_R[x] \leftarrow v_R[x] + 1$
  {compute approximate probability vectors and approximate pruned HeteSim}
  $v'_L \leftarrow v_L/N$
  $v'_R \leftarrow v_R/N$
  return $(v'_L \cdot v'_R)/(|v'_L||v'_R|)$
```

Algorithm 7 restrictedRandomWalkOnMetapath subroutine

```
Input: start node $s$, metapath $\mathcal{P}$, badNodes $B$
Output: (B, node), where node is the final node reached, and B is the updated list of
  dead-end nodes
  $i \leftarrow 1$
  nodeStack ← [ ]
  $x \leftarrow s$
  while $i > 0$ do
    $Y \leftarrow$ neighbors($x$, $R_i$) $\backslash B[i]$
    if $Y \neq \emptyset$ then
      {pick a neighbor with probability proportional to edge weight}
      $w \leftarrow \sum_{y \in Y}$ edgeweight($x$, $y$)
      $z \leftarrow$ SelectWithProbability( [($y$, edgeWeight($x, y$)/$w$) for $y \in Y$] )
      nodeStack.push(x)
      $x \leftarrow z$
      $i \leftarrow i + 1$
    else
      {$x$ is a dead end}
      B[i-1] ← B[i-1] $\cup \{x\}$
      $x \leftarrow$ nodeStack.pop()
      $i \leftarrow i - 1$
  return $(B, x)$
```

always a probability vector). Let

$$\mathcal{E}_k(v, \delta, \alpha, \beta) =$$

$$\{w \in W_k : v_i + w_i \geq 0, v_i \geq \alpha \Rightarrow |w_i| \leq \delta |v_i|, \text{ and } v_i < \alpha \Rightarrow |w_i| \leq \beta\delta\}, \tag{5.41}$$

where $W_k = \{w \in \mathbb{R} : \sum_i w_i = 0\}$. We now consider $\lambda \in \mathcal{E}_k(v, \delta, \alpha, \beta)$. Note that the bound imposed by $\mathcal{E}_k(v, \delta, \alpha, \beta)$ treats small entries and large entries in v differently. This will be important to achieve an $O(k \log k)$ bound on the number of required random walks (N) later in the section.

We start by giving sufficient conditions for a bound on $|\cos\theta' - \cos\theta|$, where θ' is the angle between $\hat{v}$ and $\hat{w}$ and θ is the angle between v and w.

Theorem 74. *Fix $\epsilon > 0$. Let $0 \leq \beta, \overline{\beta} \leq 1$. Let $\alpha, \overline{\alpha} \geq 0$. Let $v, w \in \mathcal{S}_k$. Let*

$$b = \frac{2 + \frac{k\beta^2}{2|v|^2}}{1 + \frac{1}{|v|\sqrt{k}}} + \sqrt{\frac{k\beta^2}{|v|^2} + 1} \qquad \textit{and} \qquad a = \frac{\frac{k\beta^2}{|v|^2} + 1}{1 + \frac{1}{|v|\sqrt{k}}},$$

and

$$\overline{b} = \frac{2 + \frac{k\overline{\beta}^2}{2|w|^2}}{1 + \frac{1}{|w|\sqrt{k}}} + \sqrt{\frac{k\overline{\beta}^2}{|w|^2} + 1} \qquad \textit{and} \qquad \overline{a} = \frac{\frac{k\overline{\beta}^2}{|w|^2} + 1}{1 + \frac{1}{|w|\sqrt{k}}}.$$

Let $\delta = \frac{\epsilon}{b + \sqrt{b^2 + 2a\epsilon}}$ and $\overline{\delta} = \frac{\epsilon}{\overline{b} + \sqrt{\overline{b}^2 + 2\overline{a}\epsilon}}$. If $\lambda \in \mathcal{E}_k(v, \delta, \alpha, \beta)$ and $\overline{\lambda} \in \mathcal{E}_k(w, \overline{\delta}, \overline{\alpha}, \overline{\beta})$ then

$$\left| \frac{(v + \lambda) \cdot (w + \overline{\lambda})}{|v + \lambda||w + \overline{\lambda}|} - \frac{v \cdot w}{|v||w|} \right| \leq \epsilon.$$

Proof. Follows from Lemma 88 in subsection 5.5.1 and the triangle inequality. □

We now need to understand the probability that any given entry of $\hat{v}$ (or $\hat{w}$) is close to the corresponding entry of v (or w). Since the number of walks arriving at a given node is binomial, we apply a Chernoff bound (Lemma 75) to the binomial distribution to get Corollary 76.

Lemma 75 (Chernoff Bound [144])**.** *Let $X \sim Binom(n,p)$. Let $\mu = \mathbb{E}(X) = np$. For $\delta > 0$,*

$$P\left(X \leq (1-\delta)\mu\right) \leq exp\left(-\frac{\delta^2\mu}{2}\right)$$

and

$$P\left(X \geq (1+\delta)\mu\right) \leq exp\left(-\frac{\delta^2\mu}{2+\delta}\right).$$

Corollary 76. *Let $X \sim Binom(n,p)$. For $\delta > 0$,*

$$P\left(\left|\frac{X}{n} - p\right| > \delta p\right) \leq 2 \cdot exp\left(-\frac{n\delta^2 p}{2+\delta}\right)$$

and

$$P\left(\left|\frac{X}{n} - p\right| > \delta\right) \leq 2 \cdot exp\left(-\frac{n\delta^2}{2p+\delta}\right).$$

Having bounded the probability of any one vector entry having small error, we now use a union bound to bound the probability that all entries have small error.

Lemma 77. *Fix $n, k \in \mathbb{N}$. Fix $\delta \geq 0$ and $0 \leq \alpha, \beta \leq 1$. Let $v = (v_1, \cdots, v_k)$ such that $v_i \geq 0$ and $\sum_i v_i = 1$. Let $X_i \sim Binom(n, v_i)$ such that $\sum_i X_i = n$. Let $\lambda_i = \frac{X_i}{n} - v_i$ and let $\lambda = (\lambda_1, \cdots, \lambda_k)$. We have that*

$$P\left(\lambda \notin \mathcal{E}_k(v, \delta, \alpha, \beta)\right) \leq 2k\, exp\left(-n\delta^2 \cdot \min\left\{\frac{\beta^2}{2\alpha+\delta\beta}, \frac{\alpha}{2+\delta}\right\}\right). \tag{5.42}$$

Proof. Since $X_i \geq 0$, $v_i + \lambda_i \geq 0$. We now apply the Chernoff bound. For $v_i \geq \alpha$, we see that

$$P\left(|\lambda_i| \geq \delta v_i\right) = P\left(\left|\frac{X_i}{n} - v_i\right| \geq \delta v_i\right) \tag{5.43}$$

$$\leq 2 \cdot \exp\left(-\frac{n\delta^2 v_i}{2+\delta}\right) \leq 2 \cdot \exp\left(-\frac{n\delta^2\alpha}{2+\delta}\right). \tag{5.44}$$

For $v_i < \alpha$, we see that

$$P(|\lambda_i| \geq \beta\delta) = P\left(\left|\frac{X_i}{n} - v_i\right| \geq \beta\delta\right) \tag{5.45}$$

$$\leq 2 \cdot \exp\left(-\frac{n\beta^2\delta^2}{2v_i + \beta\delta}\right) \leq 2 \cdot \exp\left(-\frac{n\beta^2\delta^2}{2\alpha + \beta\delta}\right). \tag{5.46}$$

The result then follows by the union bound.

□

Finally, we can combine the previous results to bound the required number of random walks, given error tolerance ϵ and success probability r.

Lemma 78. *Let* $\epsilon > 0$ *and* $0 < r < 1$. *For* $c(\epsilon) = 2(c+\sqrt{c^2+2\epsilon})^2 + \epsilon(c+\sqrt{c^2+2\epsilon})$ *and* $c = \frac{5+4\sqrt{2}}{4}$. *Let* δ *as in Theorem 74. After making* n *(non-deadend) walks in the randomized pruned HeteSim algorithm,*

$$P\left(\lambda \notin \mathcal{E}_k\left(v, \delta, \frac{|v|}{\sqrt{k}}, \frac{|v|}{\sqrt{k}}\right)\right) \leq 2k \, exp\left(-\frac{n}{k} \cdot \frac{\epsilon^2}{c(\epsilon)}\right).$$

Proof. We apply Lemma 77 and Theorem 74. We set $\alpha = \beta = \frac{|v|}{\sqrt{k}} \leq 1$ and $\delta = \frac{\epsilon}{b+\sqrt{b^2+2a\epsilon}}$. Thus,

$$P(\lambda \notin \mathcal{E}_k(v, \delta, \alpha, \beta)) \leq 2k \exp\left(-n \cdot \frac{|v|}{\sqrt{k}} \cdot \frac{\epsilon^2}{2(b+\sqrt{b^2+2a\epsilon})^2 + \epsilon(b+\sqrt{b^2+2a\epsilon})}\right). \tag{5.47}$$

We notice that the content of the exponent is a decreasing function in $|v|$ (for $|v| > 0$). Thus,

$$P(\lambda \notin \mathcal{E}_k(v, \delta, \alpha, \beta)) \leq 2k \exp\left(-\frac{n}{k} \cdot \frac{\epsilon^2}{2\left(c+\sqrt{c^2+2\epsilon}\right)^2 + \epsilon(c+\sqrt{c^2+2\epsilon})}\right). \tag{5.48}$$

□

Corollary 79. *Under the same assumptions as Lemma 78, let* $n > \frac{c(\epsilon)}{\epsilon^2} \cdot k \ln(\frac{4k}{1-r})$*, after making* n *(non-deadend) walks in the randomized pruned HeteSim algorithm (on both sides of the computation),*

$$P\left(\left|PHS\,(a, b|\mathcal{P}) - \widetilde{PHS}\,(a, b|\mathcal{P})\right| < \epsilon\right) > r.$$

Proof. Follows from Lemma 78 (applied to both sides of the computation), Theorem 74 and the union bound. □

Deterministic aggregation

In order to rank the overall relatedness of source nodes to a fixed target node, SemNet version 2 uses the mean HeteSim score between the source and target node, averaged over all metapaths which exist for any source node in the set under study. For completeness, pseudocode for computing exact mean HeteSim scores is given in Algorithm 8.

Algorithm 8 Exact Mean HeteSim score

```
Input: set of start nodes $S$, end node $t$, path length p
Output: vector of mean HeteSim scores $h$, indexed by elements of $S$
  Construct $M$, the set of all metapaths between any element of $S$ and $t$
  for $s \in S$ do
    HSscores = []
    for $m \in M$ do
      HSscores.append(HeteSim($s$, $t$, $m$))
    $h[s]$ = mean(HSscores)
  return $h$
```

Randomized Aggregation

As an alternative to taking the exact mean HeteSim score over all metapaths, we also consider an approximation to the mean given by the mean over a random subset of metapaths. Let S be a set of source nodes in the graph and T be a set of target nodes. Let $\mathcal{MP}_{ST}$ be the set of all metapaths in the knowledge graph with at least one instance between some node in S and some node in T. Let $(s, t) \in S \times T$. Let $\mathcal{P} = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{+1}$

by a metapath. Recall that HS $(s, t|\mathcal{P})$ is the HeteSim score between s and t relative to the metapath $\mathcal{P}$. Similarly, let PHS $(s, t|\mathcal{P})$ be the Pruned HeteSim score between s and t relative to the metapath $\mathcal{P}$.

The aggregated HeteSim score of a source-target pair (s, t) is defined to be

$$Q(s,t) = \frac{1}{|\mathcal{MP}_{\mathcal{ST}}|} \sum_{\mathcal{P} \in \mathcal{MP}} \text{HS}\,(s, t|\mathcal{P})\,, \tag{5.49}$$

and the aggregated Pruned HeteSim Score is defined to be

$$R(s,t) = \frac{1}{|\mathcal{MP}|} \sum_{\mathcal{P} \in \mathcal{MP}} \text{PHS}\,(s, t|\mathcal{P})\,. \tag{5.50}$$

Notice that if we select a metapath from $\mathcal{MP}$ uniformly at random and took the HeteSim score relative to that metapath, the expected value of the score is precisely $Q(s, t)$. Thus, we may approximate $Q(s, t)$ by taking m independent and uniformly chosen mathpaths, $\mathcal{P}_1, \cdots, \mathcal{P}_m$, and taking the mean of the HeteSim scores relative to these metapaths. Let

$$\hat{Q}(s,t) = \frac{1}{m} \sum_{i=1}^{m} \text{HS}\,(s, t|\mathcal{P}_i)\,. \tag{5.51}$$

Hence, $\mathbb{E}(\tilde{R}(s,t)) = R(s,t)$.

Let $\widetilde{\text{PHS}}\,(s, t|\mathcal{P})$ be the approximation of PHS $(s, t|\mathcal{P})$ derived from our randomized algorithm after taking $n(s, t|\mathcal{P})$ random walks. Let $k(s, t|\mathcal{P})$ be the number of reachable nodes of type $A_{l/2+1}$ when considering source s, target t and metapath $\mathcal{P}$. Let $k_{\max} = \max\{k(s, t|\mathcal{P}_1), \cdots, k(s, t|\mathcal{P}_m)\}$, for $\mathcal{MP}_{ST} = \{\mathcal{P}_1, \ldots, \mathcal{P}_m\}$. By the construction of the algorithm, $\mathbb{E}(\widetilde{\text{PHS}}\,(s, t|\mathcal{P})) = \text{PHS}\,(s, t|\mathcal{P})$ for a fixed $\mathcal{P}$. Let

$$\tilde{R}(s,t) = \frac{1}{m} \sum_{i=1}^{m} \text{PHS}\,(s, t|\mathcal{P}_i)$$

and

$$\hat{R}(s,t) = \frac{1}{m}\sum_{i=1}^{m} \widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right). \tag{5.52}$$

Similarly to above, $\mathbb{E}(\tilde{R}(s,t)) = R(s,t)$. We now see that

$$\begin{aligned}\mathbb{E}(\hat{R}(s,t)) &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}(\mathbb{E}(\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)|\mathcal{P}_i)) = \mathbb{E}\left(\frac{1}{m}\sum_{i=1}^{m} \text{PHS}\left(s,t|\mathcal{P}_i\right)\right) \\ &= \mathbb{E}(\tilde{R}(s,t)) = R(s,t). \end{aligned} \tag{5.53}$$

We now provide bounds on the number of random metapaths (m) we require to have $\hat{Q}(s,t)$ and $\hat{R}(s,t)$ be within some error of $Q(s,t)$ and $R(s,t)$, respectively, with at least some probability.

Lemma 80 (Bounded Differences Inequality [145]). *Let $Z_1, \cdots, Z_k$ be independent random variables such that $Z_i \in \Lambda_i$. Let $f : \Lambda_1 \times \cdots \times \Lambda_k \to \mathbb{R}$. Assume there exist $c_1, \cdots, c_k \in \mathbb{R}$ such that, for all i,*

$$|f(a_1, \cdots, a_{i-1}, a_i, a_{i+1}, \cdots a_k) - f(a_1, \cdots, a_{i-1}, a_i', a_{i+1}, \cdots a_k)| \leq c_i$$

for all $a_j \in \Lambda_j$ and $a_i' \in \Lambda_i$. Let $X = f(Z_1, \cdots, Z_k)$. We have that

$$P\left(|X - \mathbb{E}(X)| \geq t\right) \leq 2\, exp\left(-\frac{2t^2}{\sum_{i=1}^{k} c_i^2}\right). \tag{5.54}$$

Lemma 81. *For all $(s,t) \in S \times T$,*

$$P\left(\left|\hat{Q}(s,t) - Q(s,t)\right| \geq \epsilon\right) \leq 2e^{-2m\epsilon^2} \tag{5.55}$$

and

$$P\left(\left|\tilde{R}(s,t) - R(s,t)\right| \geq \epsilon\right) \leq 2e^{-2m\epsilon^2}. \tag{5.56}$$

.

Proof. Fix $(s,t) \in S \times T$. We utilize the bounded differences inequality. We take $\mathcal{P}_1, \cdots, \mathcal{P}_m$ to be our independent random variables. Let

$$\hat{Q}(\mathcal{P}_1, \cdots \mathcal{P}_k \cdots, \mathcal{P}_m)(s,t) = \frac{1}{m} \sum_{i=1}^{m} \text{HS}\left(s,t|\mathcal{P}_i\right).$$

Notice that for any $k \in [m]$,

$$\begin{aligned} &\left|\hat{Q}(\mathcal{P}_1, \cdots \mathcal{P}_k \cdots, \mathcal{P}_m)(s,t) - \hat{Q}(\mathcal{P}_1, \cdots \mathcal{P}'_k \cdots, \mathcal{P}_m)(s,t)\right| \\ &= \left|\frac{\text{HS}\left(s,t|\mathcal{P}_k\right) - \text{HS}\left(s,t|\mathcal{P}'_k\right)}{m}\right| \leq \frac{1}{m}. \end{aligned} \tag{5.57}$$

Thus $c_i = \frac{1}{m}$ is sufficient to apply the bounded differences inequality. Hence,

$$P\left(|\hat{Q}(s,t) - \mathbb{E}(\hat{Q}(s,t))| \geq \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{m} c_i^2}\right) = 2e^{-2m\epsilon^2}.$$

A similar argument holds for $\tilde{R}(s,t)$.

□

Corollary 82. *For* $m = \frac{1}{2\epsilon^2}\ln\left(\frac{2|S||T|}{r}\right)$*, with probability at least* $1-r$*,*

$$\left|\hat{Q}(s,t) - Q(s,t)\right| < \epsilon$$

for all $(s,t) \in S \times T$*.*

Proof. Applying Equation 81, we see that

$$\begin{aligned} P\left(\bigcup_{(s,t)\in S\times T} |\hat{Q}(s,t) - Q(s,t)| \geq \epsilon\right) &\leq \sum_{(s,t)\in S\times T} P\left(|\hat{Q}(s,t) - Q(s,t))| \geq \epsilon\right) \\ &\leq 2|S||T|e^{-2m\epsilon^2}. \end{aligned} \tag{5.58}$$

Thus the probability that $\left|\tilde{R}(s,t) - R(s,t)\right| < \epsilon$ for all $(s,t) \in S \times T$ is at least $1 - 2|S||T|e^{-2m\epsilon^2}$. To have this probability at least $1 - r$, it is hence sufficient to have $2|S||T|e^{-2m\epsilon^2} = r$, proving the result.

$\square$

Theorem 83. *Fix* $0 < \epsilon, r < 1$. *For*

$$n(s,t|\mathcal{P}_i) = \frac{4c\left(\frac{\epsilon}{2}\right) \cdot k(s,t|\mathcal{P}_i)}{\epsilon^2} \ln\left(\frac{4m|S||T|k_{\max}}{r_1}\right), \tag{5.59}$$

and

$$m = \frac{2}{\epsilon^2} \ln\left(\frac{2|S||T|}{r - r_1}\right), \tag{5.60}$$

where $r_1 = r \cdot \frac{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max}}{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max}+\epsilon^2}$, *with probability at least* $1 - r$,

$$|\hat{R}(s,t) - R(s,t)| < \epsilon$$

for all $(s,t) \in S \times T$.

The proof of this result is deferred to subsection 5.5.2.

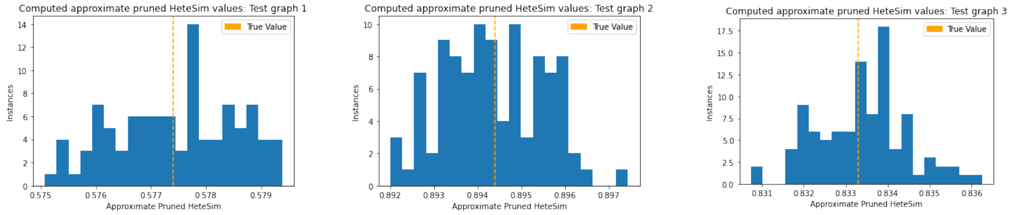
The results from this section give rise to 2 algorithms for computing approximations to mean HeteSim scores. First, Corollary 82 gives an algorithm for approximating the mean HeteSim score using the deterministic HeteSim algorithm given in Algorithm 3. Pseudocode for this approximate mean HeteSim computation is given in Algorithm 9. Second, Theorem 83 shows how to compute an approximation to the mean pruned HeteSim score, and pseudocode for this computation is given in Algorithm 10.

Algorithm 9 Approximate Mean HeteSim score

```
Input: set of start nodes $S$, end node $t$, path length p, approximation parameters $\epsilon$ and $r$
Output: vector of approximate mean HeteSim scores $h$, indexed by elements of $S$, with
  error bounds as in Corollary 82
  $m \leftarrow \frac{1}{2\epsilon^2} \ln\left(\frac{2|S|}{r}\right)$
  Construct $M$, the set of all metapaths of length $p$ between any element of $S$ and $t$
  if $m < M$ then
    select $M' \subseteq M$ with $|M'| = m$ uniformly at random
  else
    $M' \leftarrow M$
  for $s \in S$ do
    HSscores = []
    for $m \in M'$ do
      HSscores.append(HeteSim($s$, $t$, $m$))
    $h[s]$ = mean(HSscores)
  return $h$
```

Algorithm 10 Approximate Mean Pruned HeteSim score

```
Input: set of start nodes $S$, end node $t$, path length p, approximation parameters $\epsilon$ and $r$
Output: vector of approximate mean HeteSim scores $h$, indexed by elements of $S$, with
  error bounds as in Theorem 83
  $r_1 \leftarrow r \cdot \frac{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max}}{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max}+\epsilon^2}$
  $m \leftarrow \frac{2}{\epsilon^2} \ln\left(\frac{2|S|}{r-r_1}\right)$
  $N \leftarrow \frac{4c\left(\frac{\epsilon}{2}\right)\cdot k(s,t|\mathcal{P}_i)}{\epsilon^2} \ln\left(\frac{4m|S||T|k_{\max}}{r_1}\right)$
  Construct $M$, the set of all metapaths of length $p$ between any element of $S$ and $t$
  if $m < M$ then
    select $M' \subseteq M$ with $|M'| = m$ uniformly at random
  else
    $M' \leftarrow M$
  for $s \in S$ do
    PHSscores = []
    for $m \in M'$ do
      PHSscores.append(RandomizedPrunedHeteSimGivenN($s$, $t$, $m$, $N$))
    $h[s]$ = mean(PHSscores)
  return $h$
```



(a) Test graph 1

(b) Test graph 2

(c) Test graph 3

Figure 5.8: Computed approximate pruned HeteSim values for each of the three test graphs.

5.3.3 Algorithm Runtimes: SemNet version 2

Verification of randomized algorithm performance

For each of the three test graphs and corresponding metapaths, the randomized pruned HeteSim algorithm was run 100 times, with $\epsilon = 0.05$ and $r = 0.95$. For each of the three test graphs, an error less than ϵ was observed in all 100 iterations. Histograms showing the distribution of computed values are given in Figure 5.8.

Comparison of Algorithm Runtimes

For two of the three main algorithm variants, runtime on length 2 metapaths was measured, using Alzheimer's disease as a target node and a set of three source nodes: insulin, hypothyroidism, and amyloid. Each of these source nodes has some amount of real-world domain significance; all three have, at some point, acted as a source node to the target node Alzheimer's disease in other ongoing research in the authors' lab. This ongoing work aims to investigate and discover causes and treatments (re-purposed or otherwise) within the active body of biomedical academic literature. As a more specific example, SemNet version 1 was used to investigate how hypothyroidism and Alzheimer's disease are related via the combined rankings of shared source nodes. This is a slightly different application than what is being investigated in this manuscript, but the results definitively show that hypothyroidism and Alzheimer's disease are closely related. These previous runs have historically been extremely slow while utilizing SemNet version 1, taking up to an hour to complete (see Table 5.3). Decreasing runtime is the main motivation for the new algorithms and implementations.

Of the two chosen algorithms associated with SemNet version 2, each computation was repeated 10 times. The results are summerized in Table 5.2. The third algorithm variant, approximate mean pruned HeteSim, was not run on the actual knowledge graph, due to excessive runtime when using realistic values for ϵ and r.

Algorithm	Runtime (sec)
Mean exact HeteSim	0.37 ± 0.0082
Approximate mean of exact HeteSim	0.29 ± 0.0017

Table 5.2: Run times for algorithms over all metapaths of length 2, in seconds. Each of the algorithms was run with target node Alzheimer's disease and a set of three source nodes: insulin, hypothyroidism, and amyloid. Values given are the mean and standard deviation obtained from running each computation 10 times, and each value is stated with 2 significant figures. For the approximation algorithm, parameters $\epsilon = 0.1$ and $r = 0.9$ were used.

For the fastest algorithm, approximate mean HeteSim, time spent on each of the three steps show in Figure 5.5 was also recorded. Results are given in Table 5.3.

Additionally, the time to compute HeteSim using the new data structure for a single metapath was analyzed. For comparison, 5 unique metapaths were computed between source and target nodes for each target node, using an iterator based on a standard path finding algorithm. The randomized pruned HeteSim algorithm was run on these metapaths with approximation algorithm parameters $\epsilon = 0.1$ and $r = 0.9$. Randomized pruned HeteSim was not run on all metapaths due to excessive runtime. Results are given in Table 5.4. Further detail on the randomized pruned HeteSim results, including the number of iterations is given in Table 5.5.

For each source node, 20 metapaths of length 4 were generated, and the deterministic HeteSim algorithm was run on each metapath. Runtimes are shown in Table 5.6.

Source node	Insulin	Hypothyroidism	Amyloid
Num metapaths	1069	864	825
SemNet 1: Step 1 mean (s)	81	35	84
SemNet 1: Step 1 std (s)	5.3	2.4	5.3
SemNet 1: Step 2 mean (s)	$220,000$	$96,000$	$220,000$
SemNet 1: Step 2 std (s)	2300	270	2700
SemNet 1: Step 3 mean (s)	0.80	0.39	0.80
SemNet 1: Step 3 std (s)	0.0021	0.0093	0.014
SemNet 2: Step 1 mean (s)	0.077	0.042	0.042
SemNet 2: Step 1 std (s)	0.0010	0.00026	0.00019
SemNet 2: Step 2 mean (s)	0.0025	0.0020	0.0021
SemNet 2: Step 2 std (s)	0.000046	0.00038	0.000046
SemNet 2: Step 3 mean (s)	0.00010	9.9×10^{-5}	0.00010
SemNet 2: Step 3 std (s)	1.9×10^{-6}	1.7×10^{-6}	3.3×10^{-6}
Runtime ratio: Step 1	1100	830	2000
Runtime ratio: Step 2	8.8×10^{7}	4.8×10^{7}	1.0×10^{8}
Runtime ratio: Step 3	8000	3900	8000

Table 5.3: Runtimes for approximate mean HeteSim algorithm in SemNet version 2, broken down by step as in Figure 5.5 and runtimes for HeteSim algorithm in SemNet version 1, broken down by step as in Figure 5.2. For approximate mean HeteSim, approximation parameters were $\epsilon = 0.1$ and $r = 0.9$. All values are given with 2 significant figures. For each algorithm, means were taken over 10 iterations. Number of metapaths are given for SemNet version 2.

5.4 Discussion

The results presented in this manuscript show that algorithmic and data structure changes have reduced the runtime of HeteSim computations, while fixing an error in the rank aggregation algorithm. However, the need to compute all metapaths between the specified nodes is still a computational bottleneck.

5.4.1 Computational improvements

Both the mean HeteSim score and approximate mean HeteSim score show runtime reductions compared to SemNet version 1. These improvements are evident both in the overall algorithm runtimes (Table 5.1 and Table 5.2) and in the speed of the deterministic HeteSim subroutine (Table 5.1 and Table 5.4). Note that, though the number of metapaths

Source Node	Deterministic HeteSim	Randomized pruned HeteSim
Insulin	$9.2 \times 10^{-5} \pm 2.9 \times 10^{-4}$	1400 ± 2800
Hypothyroidism	$5.8 \times 10^{-5} \pm 9.3 \times 10^{-5}$	8.9 ± 11
Amyloid	$5.5 \times 10^{-5} \pm 9.9 \times 10^{-5}$	76 ± 130

Table 5.4: Mean and standard deviation of runtime for HeteSim algorithms on a single length 2 metapath. For the deterministic HeteSim algorithm, means are computed over all metapaths of length 2 between source and target nodes. For the randomized pruned HeteSim algorithm, the means are computed over 5 unique metapaths. Each value is stated with 2 significant figures.

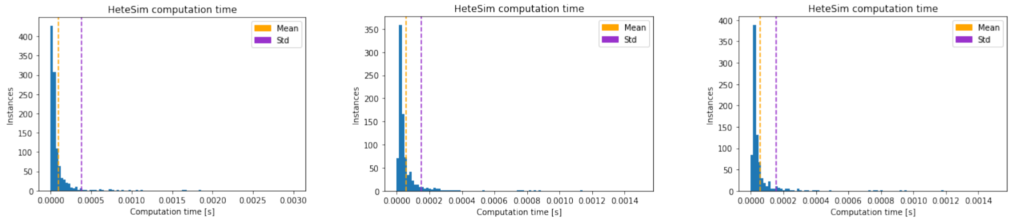
Source node	Insulin	Hypothyroidism	Amyloid
Max num iterations (N)	$40,485,730$	$2,208,823$	$9,464,267$
Min num iterations (N)	$50,937$	$117,988$	$117,988$
Mean num iterations (N)	$8,927,127$	$590,025$	$2,545,180$
Max runtime (s)	$7,100$	32	340
Min runtime (s)	0.35	2.7	0.91
Mean runtime (s)	$1,400$	8.9	76

Table 5.5: Computation details for the randomized pruned HeteSim algorithm on a single metapath. For each source node, 5 distinct metapaths were used as input to the algorithm. Approximation parameters were $\epsilon = 0.1$ and $r = 0.9$. Runtimes and runtime means are given to 2 significant figures. Iteration counts are exact. Iteration means are rounded to the nearest whole number.

decreased in the graph used to test SemNet version 2 and this reduction must account for some speedup, computation time per metapath also decreased. However, we have not collected any information about the complexity of the metapaths (e.g. how many instances of each metapath exist). Table 5.3 shows that the largest improvement happened in step 2, likely because the implementation of step 2 in SemNet version 1 used many Neo4j queries. Since we have already shown that Neo4j queries made up the majority of the runtime in SemNet version 1 (see Table 5.1), it is likely that the substitution of the Python dictionary-

Source node	Single metapath HeteSim runtime
Insulin	0.034 ± 0.034
Hypothyroidism	0.024 ± 0.031
Amyloid	0.013 ± 0.013

Table 5.6: SemNet version 2 HeteSim runtime for a single length 4 metapath, averaged over 20 distinct metapaths for each source node. All results are given with 2 significant figures.



(a) Insulin (b) Hypothyroidism (c) Amyloid

Figure 5.9: HeteSim computation times per metapath for all metapaths of length 2 from the given source node to Alzheimer's disease, using the deterministic HeteSim implementation from SemNet version 2.0.

based data structure for the knowledge graph was the largest source of runtime reduction for step 2. Similarly, step 1 involves querying the knowledge graph, and the replacement of Neo4j with a custom dictionary-based data structure is likely the largest source of improvement here as well.

Step 3 is a bit different because the changes here were motivated by the replacement of a flawed rank aggregation technique, rather than runtime considerations. As a ratio, we do see an improvement reduction in runtime of over 1000, but the absolute runtime values for step 3 are quite small in relation to the entire algorithm. The most important result regarding step 3 is the replacement ULARA with a sensible alternative (mean HeteSim score) that is also is amenable to approximation based on randomization.

In the length 2 metapath tests reported in Table 5.2, the approximate mean HeteSim algorithm achieves a runtime of approximately 20% less than the exact mean HeteSim score computation. This reduction is due to the need to run the HeteSim subroutine on fewer metapaths. Since the bound on the number of metapaths for which HeteSim must be computed depends only on the number of candidate source nodes and the approximation parameters ϵ and r (see Corollary 82), the performance advantage of the approximate mean computation should be even larger in situations involving more metapaths. In particular, computations on longer metapaths will generally involve more metapaths and therefore will benefit even more from the approximate mean algorithm.

It should also be noted that the use of approximation algorithms is appropriate in this context. The knowledge graph is inherently noisy, as it is generated using natural language processing techniques on biomedical paper abstracts. Additionally, the primary use of SemNet is in hypothesis generation. Both of these factors make the trade off of some accuracy for speed an acceptable one.

5.4.2 Mathematical Limitations

In Corollary 79, we prove that it is sufficient to take $O\left(\frac{1}{\epsilon^2}k\ln\left(\frac{k}{1-r}\right)\right)$ random walks in the randomized Pruned HeteSim algorithm. As illustrated by Table 5.5, the bound we achieved may, at times, result in a large number of required walks, when considering realistic knowledge graphs and modest values for ϵ and r. We acknowledge that the bound achieved may be crude, especially in our frequent use of the, generally loose, union bound. Hence, we leave open the possibility of substantial improvement to both the constant we achieve ($c(\epsilon) \leq 71$) and the order with respect to the various variables.

One possible area of improvement is in the order with respect to k. We conjecture that the required number of walks is at least order k, thus leaving room for the possibility of the true value to be between order k and $k\log k$ (inclusive). Considering the order with respect to ϵ, we note that most standard general concentration inequalities necessitate $O\left(\frac{1}{\epsilon^2}\right)$. This being said, the distribution we are considering is binomial. While the authors are not aware of any stronger results for the binomial distribution, we are also not aware of any reason why such a result could not exist.

We also note that to achieve Lemma 78, we utilize an error allocation scheme that bounds large entries with error proportional to the value of the entry but bounds small entries with a fixed bound. This is just one possible scheme which leaves open the possibility of achieving tighter results using another, possibly more individualized, scheme.

5.4.3 Limitations and future directions

The knowledge graph used to test SemNet version 2 has substantially fewer edges than the knowledge graph used in SemNet version 1, as seen by the reduced number of metapaths between vertices of interest (see Table 5.1 and Table 5.3). Future work will address this limitation and give more accurate runtime comparisons by building a knowledge graph of comparable size to that used in SemNet version 1.

Though the new implementation has significantly reduced the runtime required to enumerate metapaths, metapath enumeration remains a computational bottleneck. This bottleneck is a barrier to HeteSim computations on longer metapaths. Since counting the number of paths between two specified nodes in a directed graph is $\#\mathbf{P}$-complete [146], metapath enumeration is likely also a computationally hard problem. In order to make further progress, future work will need to address this metapath enumeration problem. One possible approach is to devise an algorithm for sampling metapaths under a uniform (or other useful) probability distribution, perhaps using a Markov chain Monte Carlo technique. If such an algorithm could be devised, it could be used directly with the randomized aggregation scheme described in Algorithm 9.

5.5 Technical Lemmas and Proofs of Theorems

5.5.1 Technical Lemmas

Lemma 84. *For $v \in \mathcal{S}_k$, $\frac{1}{\sqrt{k}} \leq |v| \leq 1$.*

Proof. By method of Lagrange Multipliers. □

Lemma 85. *Let $\delta > 0$, $\alpha > 0$ and $0 < \beta \leq 1$. Let $v, w \in \mathcal{S}_k$ and $\lambda \in \mathcal{E}_k(v, \delta, \alpha, \beta)$. We have that*

$$|\lambda \cdot v| \leq \delta \left(|v|^2 + \frac{k\beta^2}{4} \right) \qquad \text{and} \qquad \left| \lambda \cdot \frac{w}{|w|} \right| \leq |\lambda| \leq \delta \sqrt{k\beta^2 + |v|^2}.$$

Proof. Assume v_i has m entries less than α and these are the first m entries. Clearly, $m \leq k$. We see that

$$\begin{aligned}
\lambda \cdot v &\leq \sum_{i=1}^{k} \lambda_i \cdot v_i \leq \sum_{i=1}^{m} \beta\delta \cdot v_i + \sum_{i=m+1}^{k} \delta \cdot v_i^2 && (5.61)\\
&\leq \delta \sum_{i=1}^{m} v_i(\beta - v_i) + \delta \sum_{i=1}^{k} \cdot v_i^2 && (5.62)\\
&\leq \delta |v|^2 + \delta \frac{k\beta^2}{4}. && (5.63)
\end{aligned}$$

We get the lower bound similarly. Clearly, $m \leq k$. Thus we also see that

$$\begin{aligned}
\left| \lambda \cdot \frac{w}{|w|} \right| &\leq \left| |\lambda| \cdot \frac{\lambda}{|\lambda|} \cdot \frac{w}{|w|} \right| \leq |\lambda| && (5.64)\\
&\leq \sqrt{\sum_{i=1}^{m} (\beta\delta)^2 + \delta^2 \sum_{i=m+1}^{k} v_i^2} && (5.65)\\
&\leq \delta \sqrt{k\beta^2 + |v|^2}. && (5.66)
\end{aligned}$$

□

Lemma 86. *For* $v, \lambda \in \mathbb{R}^k$ *such that* $v, v + \lambda \in \mathcal{S}_k$*,*

$$||v + \lambda| - |v|| \leq \frac{2|\lambda \cdot v| + |\lambda|^2}{|v| + \frac{1}{\sqrt{k}}}.$$

Proof. We first see that

$$||v + \lambda| - |v|| = \frac{||v + \lambda| + |v|| \cdot ||v + \lambda| - |v||}{||v + \lambda| + |v||} = \frac{||v + \lambda|^2 - |v|^2|}{||v + \lambda| + |v||} \leq \frac{||v + \lambda|^2 - |v|^2|}{|v| + \frac{1}{\sqrt{k}}}.$$

Next, note

$$\begin{aligned}
|v+\lambda|^2 &= (v+\lambda)\cdot(v+\lambda) && (5.67)\\
&= v\cdot v + 2\lambda\cdot v + \lambda\cdot\lambda && (5.68)\\
&= |v|^2 + 2\lambda\cdot v + |\lambda|^2 && (5.69)\\
|v+\lambda|^2 - |v|^2 &\leq 2|\lambda\cdot v| + |\lambda|^2 && (5.70)
\end{aligned}$$

Similarly, we get that $|v|^2 - |v+\lambda|^2 \leq 2|\lambda\cdot v| + |\lambda|^2$. The desired result then follows.

□

Lemma 87. *Let $\beta \leq 1$. Let $0 < \delta$ and $\alpha, \beta \geq 0$. For $v \in \mathcal{S}_k$ and $\lambda \in \mathcal{E}_k(v, \delta, \alpha, \beta)$,*

$$\left|\frac{(v+\lambda)\cdot w}{|v+\lambda||w|} - \frac{v\cdot w}{|v||w|}\right| \leq \delta\left(2 + \frac{k\beta^2}{2|v|^2} + \sqrt{\frac{k}{|v|^2}\beta^2 + 1}\right) + \delta^2\left(\frac{k\beta^2}{|v|^2} + 1\right) \qquad (5.71)$$

Proof. We see that

$$\begin{aligned}
\frac{(v+\lambda)\cdot w}{|v+\lambda||w|} - \frac{v\cdot w}{|v||w|} &= \frac{(v+\lambda)\cdot w}{|v+\lambda||w|} - \frac{(v+\lambda)\cdot w}{|v||w|} - \frac{\lambda\cdot w}{|v||w|} && (5.72)\\
&= \frac{(v+\lambda)\cdot w}{|w|}\left(\frac{1}{|v+\lambda|} - \frac{1}{|v|}\right) - \frac{\lambda\cdot w}{|v||w|} && (5.73)\\
&= \frac{(v+\lambda)\cdot w}{|v+\lambda||w|}\cdot\frac{|v| - |v+\lambda|}{|v|} - \frac{1}{|v|}\cdot\left(\lambda\cdot\frac{w}{|w|}\right) && (5.74)\\
\left|\frac{(v+\lambda)\cdot w}{|v+\lambda||w|} - \frac{v\cdot w}{|v||w|}\right| &\leq \frac{1}{|v|}\left(||v| - |v+\lambda|| + \left|\lambda\cdot\frac{w}{|w|}\right|\right) && (5.75)\\
&\leq \frac{2\delta\left(|v|^2 + \frac{k\beta^2}{4}\right) + \delta^2\left(k\beta^2 + |v|^2\right)}{|v|^2 + \frac{|v|}{\sqrt{k}}} + \delta\sqrt{\frac{k\beta^2}{|v|^2} + 1} && (5.76)\\
&\leq \frac{\delta\left(2 + \frac{k\beta^2}{2|v|^2}\right) + \delta^2\left(\frac{k\beta^2}{|v|^2} + 1\right)}{1 + \frac{1}{|v|\sqrt{k}}} + \delta\sqrt{\frac{k\beta^2}{|v|^2} + 1} && (5.77)
\end{aligned}$$

The above inequality follows from $\left|\frac{v\cdot w}{|w||v|}\right| \leq 1$, Lemma 85 and Lemma 86. □

Lemma 88. *Fix ϵ. Let $\beta \leq 1$ and $\alpha \geq 0$. For $v \in \mathcal{S}_k$ and $\lambda \in \mathcal{E}_k(v, \alpha, \delta, \delta')$. Let*

$$b = \frac{2 + \frac{k\beta^2}{2|v|^2}}{1 + \frac{1}{|v|\sqrt{k}}} + \sqrt{\frac{k\beta^2}{|v|^2} + 1} \qquad \text{and} \qquad a = \frac{\frac{k\beta^2}{|v|^2} + 1}{1 + \frac{1}{|v|\sqrt{k}}}.$$

For $\delta \leq \frac{\epsilon}{b + \sqrt{b^2 + 2a\epsilon}}$,

$$\left| \frac{(v + \lambda) \cdot w}{|v + \lambda||w|} - \frac{v \cdot w}{|v||w|} \right| \leq \frac{\epsilon}{2}.$$

Proof. The result follows from Equation 87. □

5.5.2 Proofs of Theorems

Theorem 83. *Fix $0 < \epsilon, r < 1$. For*

$$n(s, t|\mathcal{P}_i) = \frac{4c\left(\frac{\epsilon}{2}\right) \cdot k(s, t|\mathcal{P}_i)}{\epsilon^2} \ln\left(\frac{4m|S||T|k_{\max}}{r_1}\right), \tag{5.59}$$

and

$$m = \frac{2}{\epsilon^2} \ln\left(\frac{2|S||T|}{r - r_1}\right), \tag{5.60}$$

where $r_1 = r \cdot \frac{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max}}{4\ln\left(\frac{2|S||T|}{r}\right)k_{\max} + \epsilon^2}$, with probability at least $1 - r$,

$$|\hat{R}(s, t) - R(s, t)| < \epsilon$$

for all $(s, t) \in S \times T$.

Proof. For $\epsilon_1, \epsilon_2 > 0$ such that $\epsilon_1 + \epsilon_2 = \epsilon$, we see that

$$P\left(|\hat{R}(s,t) - R(s,t)| \geq \epsilon\right) \leq P\left(\left|\hat{R}(s,t) - \tilde{R}(s,t)\right| + \left|\tilde{R}(s,t) - R(s,t)\right| \geq \epsilon\right) \tag{5.78}$$

$$\begin{aligned} P\left(|\hat{R}(s,t) - R(s,t)| \leq \epsilon\right) &\geq P\left(\left|\hat{R}(s,t) - \tilde{R}(s,t)\right| + \left|\tilde{R}(s,t) - R(s,t)\right| \leq \epsilon\right) \\ &\geq P\left(\left|\hat{R}(s,t) - \tilde{R}(s,t)\right| \leq \epsilon_1 \bigcap \left|\tilde{R}(s,t) - R(s,t)\right| \leq \epsilon_2\right) \end{aligned} \tag{5.79}$$

$$\begin{aligned} P\left(|\hat{R}(s,t) - R(s,t)| \geq \epsilon\right) &\leq P\left(\left|\hat{R}(s,t) - \tilde{R}(s,t)\right| \geq \epsilon_1 \bigcup \left|\tilde{R}(s,t) - R(s,t)\right| \geq \epsilon_2\right) \\ &\leq P\left(\left|\hat{R}(s,t) - \tilde{R}(s,t)\right| \geq \epsilon_1\right) \\ &+ P\left(\left|\tilde{R}(s,t) - R(s,t)\right| \geq \epsilon_2\right) \end{aligned} \tag{5.80}$$

Recall from Equation 81 that

$$P\left(\left|R(s,t) - \tilde{R}(s,t)\right| \geq \epsilon_2\right) \leq 2 \exp\left(-2m \cdot \epsilon_2^2\right).$$

Furthermore, from Lemma Equation 81,

$$P\left(\left|\hat{R}(s,t)-\tilde{R}(s,t)\right|\geq\epsilon_1\right)=P\left(\left|\frac{1}{m}\sum_{i=1}^{m}\text{PHS}\left(s,t|\mathcal{P}_i\right)-\frac{1}{m}\sum_{i=1}^{m}\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\geq\epsilon_1\right) \tag{5.81}$$

$$\leq P\left(\frac{1}{m}\sum_{i=1}^{m}\left|\text{PHS}\left(s,t|\mathcal{P}_i\right)-\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\geq\epsilon_1\right) \tag{5.82}$$

$$P\left(\left|\hat{R}(s,t)-\tilde{R}(s,t)\right|\leq\epsilon_1\right)\geq P\left(\frac{1}{m}\sum_{i=1}^{m}\left|\text{PHS}\left(s,t|\mathcal{P}_i\right)-\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\leq\epsilon_1\right) \tag{5.83}$$

$$\geq P\left(\bigcap_{i=1}^{m}\left|\text{PHS}\left(s,t|\mathcal{P}_i\right)-\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\leq\epsilon_1\right) \tag{5.84}$$

$$P\left(\left|\hat{R}(s,t)-\tilde{R}(s,t)\right|\geq\epsilon_1\right)\leq P\left(\bigcup_{i=1}^{m}\left|\text{PHS}\left(s,t|\mathcal{P}_i\right)-\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\geq\epsilon_1\right) \tag{5.85}$$

$$\leq\sum_{i=1}^{m}P\left(\left|\text{PHS}\left(s,t|\mathcal{P}_i\right)-\widetilde{\text{PHS}}\left(s,t|\mathcal{P}_i\right)\right|\geq\epsilon_1\right) \tag{5.86}$$

$$\leq\sum_{i=1}^{m}4k(s,t|\mathcal{P}_i)\,\text{exp}\left(-\frac{n(s,t|\mathcal{P}_i)}{k(s,t|\mathcal{P}_i)}\cdot\frac{\epsilon_1^2}{c(\epsilon_1)}\right). \tag{5.87}$$

Hence, for all $(s,t)\in S\times T$,

$$P\left(|\hat{R}(s,t)-R(s,t)|\geq\epsilon\right)\leq 2\ \mathrm{exp}\left(-2m\epsilon_2^2\right) + \sum_{i=1}^{m} 4k(s,t|\mathcal{P}_i)\ \mathrm{exp}\left(-\frac{n(s,t|\mathcal{P}_i)}{k(s,t|\mathcal{P}_i)}\cdot\frac{\epsilon_1^2}{c(\epsilon_1)}\right) \tag{5.88}$$

$$P\left(\bigcup_{(s,t)\in S\times T}|\hat{R}(s,t)-R(s,t)|\geq\epsilon\right)\leq\sum_{(s,t)\in S\times T}P\left(|\hat{R}(s,t)-R(s,t)|\geq\epsilon\right) \tag{5.89}$$

$$\leq 2|S||T|\ \mathrm{exp}\left(-2m\cdot\epsilon_2^2\right) + \sum_{(s,t)\in S\times T}\sum_{i=1}^{m}4k(s,t|\mathcal{P}_i)\ \mathrm{exp}\left(\frac{-n(s,t|\mathcal{P}_i)\epsilon_1^2}{k(s,t|\mathcal{P}_i)c(\epsilon_1)}\right) \tag{5.90}$$

Fix $r_1, r_2 > 0$ such that $r_1 + r_2 = r$. We now see that for

$$n(s,t|\mathcal{P}_i)=\frac{c(\epsilon_1)\cdot k(s,t|\mathcal{P}_i)}{\epsilon_1^2}\ln\left(\frac{1}{r_1}\sum_{(s,t)\in S\times T}\sum_{i=1}^{m}4k(s,t|\mathcal{P}_i)\right) \tag{5.91}$$

$$\leq\frac{c(\epsilon_1)\cdot k(s,t|\mathcal{P}_i)}{\epsilon_1^2}\ln\left(\frac{4m|S||T|k_{\max}}{r_1}\right) \tag{5.92}$$

and

$$m=\frac{1}{2\epsilon_2^2}\ln\left(\frac{2|S||T|}{r_2}\right), \tag{5.93}$$

we have

$$P\left(\bigcup_{(s,t)\in S\times T}|R(s,t)-\tilde{R}(s,t)|\geq\epsilon\right)\leq r. \tag{5.94}$$

We now notice that the total number of walks taken to run the algorithm (ignoring dead ends) is at most $m\cdot\max\{n(s,t|\mathcal{P}_i)\}=mn$ where $n=\frac{c(\epsilon_1)\cdot k_{\max}}{\epsilon_1^2}\ln\left(\frac{4m|S||T|k_{\max}}{r_1}\right)$. We aim to minimize nm by setting $\epsilon_1=\frac{\epsilon}{2}$ and $r_1=r\cdot\frac{2m_0k_{\max}}{2m_0k_{\max}+1}$ where $m_0=\frac{1}{2\epsilon_2^2}\ln\left(\frac{2|S||T|}{r}\right)$ is some approximation for m. (We do not claim that these choices are optimal.)

□

5.5.3 Analysis of Just-in-Time (JIT) Dead end Removal

In our given algorithm, whenever a dead end node is found, it is removed from the graph for all future walks. We model this as follows. Assume there are $m \in \mathbb{N}$ dead end nodes. Let $w \in \mathbb{R}_{\geq 0}$ be the maximum probability of reaching any single dead end. Thus, the probability of reaching a dead end is at most mw. Let $\alpha \in \mathbb{R}_{\geq 0}$ be the probability of any given walk not ending in a dead end and let $\beta = mw + \alpha$.

We now analyse the number of non-dead end walks we expect to take by the time we hit some fixed number of dead ends and the number of dead ends we expect to take by the time we hit some fixed number of non-dead ends.

In the JIT algorithm, whenever we hit a dead end, the probability of hitting a dead end in the future is affect as follows. Let $X_1, \cdots \in \{0, 1\}$ where $X_i = 1$ is the ith walk is not a dead end and $X_i = 0$ otherwise. For all i,

$$P(X_i = 1) = \frac{\alpha}{\beta - wY_i},$$

where Y_i is the number of $X_j = 0$ for $j < i$. (Thus, treating w as the weight of each dead end and α as the weight on non-dead ends, each time we hit a dead end, the weight of the dead end hit is lost as we can no longer get to that dead end. This means that overtime the probability of hitting a dead end decreases.)

Let S_i be the number of $X_j = 1$ before the i-th $X_j = 0$. Let T_i be the number of $X_j = 0$ before the i-th $X_j = 1$.

Theorem 89. *For all $i \in \mathbb{N}$,*

$$S_i = \sum_{j=0}^{i-1} Z_j - i$$

where Z_j is geometrically distributed with parameter $\frac{(m-j)w}{\beta - wj}$.

Proof. Notice that after the kth $X_j = 0$, there probability of $X_j = 1$ is $\frac{(m-k)w}{\beta - wk}$. Thus the number of $X_j = 1$ between the k-th $X_j = 0$ and $(k+1)$-th $X_j = 0$ is geometrically distributed with parameter $\frac{(m-k)w}{\beta - wk}$. (We subtract i to not count the $X_j = 0$.) □

Theorem 90.

$$\mathbb{E}(S_i) = \frac{\alpha}{w} \sum_{j=0}^{i-1} \frac{1}{m-j}$$

and

$$\mathit{Var}(S_i) = \frac{\alpha}{w^2} \sum_{j=0}^{i-1} \frac{\alpha + w(m-j)}{(m-j)^2}$$

Proof. Follows from linearity of expectation and standard results about the geometric distribution. □

Remark 91. *We can get a bound of the deviation from the mean using Chebyshev's inequality.*

Lemma 92. *For $i \geq 2$ and $k_{i-1} \leq k_i \leq m$,*

$$P\left(T_i = k_i | T_{i-1} = k_{i-1}\right) = \frac{\alpha}{\beta - wk_i} \cdot \prod_{t=k_{i-1}}^{k_i - 1} \frac{(m-t)w}{\beta - tw}$$

and

$$P\left(T_1 = k_1\right) = \frac{\alpha}{\beta - wk_1} \cdot \prod_{t=0}^{k_1 - 1} \frac{(m-t)w}{\beta - tw}$$

We now see that

$$\begin{aligned} P\left(T_n = k_n, \cdots, T_1 = k_1\right) &= P\left(T_1 = k_1\right) \prod_{i=2}^{n} P\left(T_i = k_i | T_{i-1} = k_{i-1}\right) && (5.95) \\ &= \alpha^n \left(\prod_{i=1}^{n} \frac{1}{\beta - wk_i}\right) \cdot \left(\prod_{t=0}^{k_n - 1} \frac{(m-t)w}{\beta - tw}\right) && (5.96) \end{aligned}$$

and

$$P\left(T_n = k_n\right) = \left(\frac{\alpha}{\beta}\right)^n \left(\prod_{t=0}^{k_n - 1} \frac{(m-t)w}{\beta - tw}\right) \sum_{k_{n-1}=0}^{k_n} \cdots \sum_{k_1=0}^{k_2} \left(\prod_{i=1}^{n} \frac{1}{1 - \frac{w}{\beta} k_i}\right) \qquad (5.97)$$

Lemma 93. *For all $n \geq 1$,*

$$\sum_{k_{n-1}=0}^{k_n} \cdots \sum_{k_1=0}^{k_2} \left(\prod_{i=1}^{n} \frac{1}{1 - \frac{w}{\beta} k_i} \right) \leq \frac{(-1)^{n-1}}{(n-1)!} \left(\frac{\beta}{w} \right)^{n-1} \frac{\ln \left(1 - \frac{w}{\beta}(k_n + n - 1) \right)^{n-1}}{1 - \frac{w}{\beta} k_n}.$$

Proof. We note that for increasing f,

$$\int_{a-1}^{b} f(x)\, dx \leq \sum_{k=a}^{b} f(k) \leq \int_{a}^{b+1} f(x)\, dx.$$

We prove this result inductively. Assume the result hold for $n = m$. For the upper bound, we now see that

$$\begin{aligned} \sum_{k_{n-1}=0}^{k_{m+1}} \cdots \sum_{k_1=0}^{k_2} \left(\prod_{i=1}^{m+1} \frac{1}{1 - \frac{w}{\beta} k_i} \right) &= \frac{1}{1 - \frac{w}{\beta} k_{m+1}} \sum_{k_m=0}^{k_{m+1}} \left(\sum_{k_{m-1}=0}^{k_m} \cdots \sum_{k_1=0}^{k_2} \left(\prod_{i=1}^{m} \frac{1}{1 - \frac{w}{\beta} k_i} \right) \right) \\ \leq \frac{1}{1 - \frac{w}{\beta} k_{m+1}} \cdot \frac{1}{(m-1)!} \left(\frac{\beta}{w} \right)^{m-1} &\sum_{k_m=0}^{k_{m+1}} \frac{(-1)^{m-1} \ln \left(1 - \frac{w}{\beta}(k_m + m - 1) \right)^{m-1}}{1 - \frac{w}{\beta} k_m}. \end{aligned} \tag{5.98}$$

We note that

$$\frac{\left[- \ln \left(1 - \frac{w}{\beta}(k_m + m - 1) \right) \right]^{m-1}}{1 - \frac{w}{\beta} k_m}$$

is increasing as a function in k_m and positive for $k_m \geq 0$ for all $m \in \mathbb{N}$. Hence,

$$\begin{aligned}
&\sum_{k_{n-1}=0}^{k_{m+1}} \cdots \sum_{k_1=0}^{k_2} \left(\prod_{i=1}^{m+1} \frac{1}{1-\frac{w}{\beta}k_i} \right) \\
&\leq \frac{1}{1-\frac{w}{\beta}k_{m+1}} \cdot \frac{(-1)^{m-1}}{(m-1)!} \left(\frac{\beta}{w}\right)^{m-1} \int_0^{k_{m+1}+1} \frac{\ln\left(1-\frac{w}{\beta}(x+m-1)\right)^{m-1}}{1-\frac{w}{\beta}x}\, dx \\
&\leq \frac{1}{1-\frac{w}{\beta}k_{m+1}} \cdot \frac{(-1)^{m-1}}{(m-1)!} \left(\frac{\beta}{w}\right)^{m-1} \int_0^{k_{m+1}+1} \frac{\ln\left(1-\frac{w}{\beta}(x+m-1)\right)^{m-1}}{1-\frac{w}{\beta}(x+m-1)}\, dx \\
&\leq \frac{(-1)^m}{m!} \left(\frac{\beta}{w}\right)^m \frac{\left[\ln\left(1-\frac{w}{\beta}(k_{m+1}+m)\right)^m - \ln\left(1-\frac{w}{\beta}(m-1)\right)^m\right]}{1-\frac{w}{\beta}k_{m+1}} \\
&\leq \frac{(-1)^m}{m!} \left(\frac{\beta}{w}\right)^m \frac{\ln\left(1-\frac{w}{\beta}(k_{m+1}+m)\right)^m}{1-\frac{w}{\beta}k_{m+1}}.
\end{aligned}$$

□

Theorem 94.

$$P\left(T_n = k_n\right) \leq \left(\frac{\alpha}{w}\right)^n \left(\prod_{t=0}^{k_n-1} \frac{(m-t)w}{\beta - tw}\right) \frac{(-1)^{n-1}}{(n-1)!} \cdot \frac{\ln\left(1-\frac{w}{\beta}(k_n+n-1)\right)^{n-1}}{\frac{\beta}{w}-k_n} \tag{5.99}$$

Proof. Follows from Lemma 93. □

CHAPTER 6
CONCLUSION

The previous four chapters have presented examples of combinatorial models in predictive medicine. Several different biomedical topics motivate the chapters, and these topics together relate to multiple key goals of predictive medicine. The Markov models for Alzheimer's disease progression developed in chapter 2 contribute to improving patient care by providing insight into which variables best predict the rate of progression of the disease, enabling clinicians to provide better information to patients and families. The plane tree model for RNA secondary structure analyzed in chapter 3 and chapter 4 can aid research into the mechanisms of disease through more accurate structural predictions. The algorithmic improvements presented in chapter 5 will enable further research into the causes of and treatments for disease.

The mathematical techniques employed in this thesis are somewhat more unified. Randomized algorithms play central roles in chapter 4 and chapter 5, and stochasticity is a central part of the models studied in chapter 2, chapter 3, and chapter 4.

Taken together, these chapters show the potential for mutual benefit among the fields of predictive medicine and combinatorics. Combinatorics is a valuable tool which can provide useful modeling frameworks, explore the boundaries of models, and devise improved algorithms for use in predictive medicine. The problems arising from predictive medicine can, in turn, inspire new combinatorial questions and results.

Beyond predictive medicine, combinatorial models have the potential to shed light in many fields dealing with complex systems. Indeed, the published literature contains many examples. Random walks have been used as a model for the movement of animals in an ecosystem [147, 148, 149, 150]. Random trees have been used in the study of evolution and phylogenetics [151, 152, 153, 154]. Hidden Markov models have been used as the basis for speech recognition software [155, 156]. These are, of course, just a few examples.

The power of combinatorial models to simplify complex systems so that meaningful analysis and/or computation is possible is evident both from the content of this thesis and from the many other cases were combinatorial modeling has been applied. Combinatorics is a valuable tool with the potential to shed light on problems in fields ranging from medicine to ecology to engineering. Future work applying combinatorial methods to these and other fields holds promise in building understanding of complex natural and man-made systems.

REFERENCES

[1] R. Wilson, *Combinatorics: A very short introduction*. Oxford University Press, 2016.

[2] G. Berman and K. D. Fryer, *Introduction to combinatorics*. Elsevier, 2014.

[3] F. Roberts and B. Tesman, *Applied combinatorics*. CRC Press, 2009.

[4] A. D. Weston and L. Hood, “Systems biology, proteomics, and the future of health care: Toward predictive, preventative, and personalized medicine,” *Journal of proteome research*, vol. 3, no. 2, pp. 179–196, 2004.

[5] L. Hood, J. R. Heath, M. E. Phelps, and B. Lin, “Systems biology and new technologies enable predictive and preventative medicine,” *Science*, vol. 306, no. 5696, pp. 640–643, 2004.

[6] G. W. Small, “Early diagnosis of Alzheimer’s disease: Update on combining genetic and brain-imaging measures,” *Dialogues in clinical neuroscience*, vol. 2, no. 3, p. 241, 2000.

[7] G. W. Ross, R. D. Abbott, H. Petrovitch, K. H. Masaki, C. Murdaugh, C. Trockman, J. D. Curb, and L. R. White, “Frequency and characteristics of silent dementia among elderly Japanese-American men: The Honolulu-Asia Aging Study,” *Jama*, vol. 277, no. 10, pp. 800–805, 1997.

[8] M. S. Chong and S. Sahadevan, “Preclinical Alzheimer’s disease: Diagnosis and prediction of progression,” *The Lancet Neurology*, vol. 4, no. 9, pp. 576–579, 2005.

[9] A. Borodavka, S. W. Singaram, P. G. Stockley, W. M. Gelbart, A. Ben-Shaul, and R. Tuma, “Sizes of long RNA molecules are determined by the branching patterns of their secondary structures,” *Biophysical Journal*, vol. 111, no. 10, pp. 2077–2085, 2016.

[10] I. Tinoco Jr and C. Bustamante, “How RNA folds,” *Journal of molecular biology*, vol. 293, no. 2, pp. 271–281, 1999.

[11] J. A. Doudna, “Structural genomics of RNA,” *Nature Structural Biology*, vol. 7, no. 11, pp. 954–956, 2000.

[12] V. Hower and C. E. Heitsch, “Parametric analysis of RNA branching configurations,” *Bulletin of Mathematical Biology*, vol. 73, no. 4, pp. 754–776, 2011.

[13] C. for International Organization of Medical Sciences and I. C. for Laboratory Animal Science, *International guiding principles for biomedical research involving animals*, 2012.

[14] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, “Hetesim: A general framework for relevance measure in heterogeneous networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.

[15] A. R. Sedler and C. S. Mitchell, “SemNet: Using local features to navigate the biomedical concept graph,” *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 156, 2019.

[16] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[17] A. Bureau, J. P. Hughes, and S. C. Shiboski, “An S-plus implementation of hidden markov models in continuous time,” *Journal of Computational and Graphical Statistics*, vol. 9, no. 4, pp. 621–632, 2000.

[18] P. Metzner, E. Dittmer, T. Jahnke, and C. Schütte, “Generator estimation of Markov jump processes,” *Journal of Computational Physics*, vol. 227, no. 1, pp. 353–375, 2007.

[19] I. L. MacDonald and W. Zucchini, *Hidden Markov and other models for discrete-valued time series*. CRC Press, 1997, vol. 110.

[20] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*. Springer Science & Business Media, 2006.

[21] Y.-Y. Liu, S. Li, F. Li, L. Song, and J. M. Rehg, “Efficient learning of continuous-time hidden Markov models for disease progression,” *Advances in neural information processing systems*, vol. 28, p. 3599, 2015.

[22] C. H. Jackson and L. D. Sharples, “Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients,” *Statistics in medicine*, vol. 21, no. 1, pp. 113–128, 2002.

[23] J. H. Klotz and L. D. Sharples, “Estimation for a Markov heart transplant model,” *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 43, no. 3, pp. 431–438, 1994.

[24] R. Kay, “A Markov model for analysing cancer markers and disease states in survival studies,” *Biometrics*, pp. 855–865, 1986.

[25] I. M. Longini Jr, W. S. Clark, R. H. Byers, J. W. Ward, W. W. Darrow, G. F. Lemp, and H. W. Hethcote, "Statistical analysis of the stages of HIV infection using a Markov model," *Statistics in medicine*, vol. 8, no. 7, pp. 831–843, 1989.

[26] G. A. Satten and I. M. Longini Jr, "Markov chains with measurement error: Estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 45, no. 3, pp. 275–295, 1996.

[27] G. Marshall and R. H. Jones, "Multi-state models and diabetic retinopathy," *Statistics in medicine*, vol. 14, no. 18, pp. 1975–1983, 1995.

[28] P. K. Andersen, "Multistate models in survival analysis: A study of nephropathy and mortality in diabetes," *Statistics in medicine*, vol. 7, no. 6, pp. 661–670, 1988.

[29] C. H. Jackson *et al.*, "Multi-state models for panel data: The msm package for r," *Journal of statistical software*, vol. 38, no. 8, pp. 1–29, 2011.

[30] M. E. Cowen, M. Chartrand, and W. F. Weitzel, "A Markov model of the natural history of prostate cancer," *Journal of clinical epidemiology*, vol. 47, no. 1, pp. 3–21, 1994.

[31] M. Momenzadeh, M. Sehhati, and H. Rabbani, "Using hidden Markov model to predict recurrence of breast cancer based on sequential patterns in gene expression profiles," *Journal of Biomedical Informatics*, vol. 111, p. 103 570, 2020.

[32] B. C. Soper, M. Nygård, G. Abdulla, R. Meng, and J. F. Nygård, "A hidden Markov model for population-level cervical cancer screening data," *Statistics in Medicine*, vol. 39, no. 25, pp. 3569–3590, 2020.

[33] T. Carleman, *Les Fonctions quasi analytiques: leçons professées au College de France*. Gauthier-Villars et Cie, 1926.

[34] P. Flajolet and R. Sedgewick, *Analytic combinatorics*. Cambridge University press, 2009.

[35] J. P. Huelsenbeck, B. Larget, and M. E. Alfaro, "Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo," *Molecular Biology and Evolution*, vol. 21, no. 6, pp. 1123–1133, Jun. 2004.

[36] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?" *Journal of Biological Systems*, vol. 10, no. 04, pp. 337–357, 2002.

[37] D. Lusseau, “Effects of tour boats on the behavior of bottlenose dolphins: Using Markov chains to model anthropogenic impacts,” *Conservation Biology*, vol. 17, no. 6, pp. 1785–1793, 2003.

[38] M. R. Said, A. V. Oppenheim, and D. A. Lauffenburger, “Modeling cellular signal processing using interacting Markov chains,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03).*, vol. 6, 2003, pp. VI–41.

[39] G. Hamra, R. MacLehose, and D. Richardson, “Markov chain Monte Carlo: An introduction for epidemiologists,” *International Journal of Epidemiology*, vol. 42, no. 2, pp. 627–634, 2013.

[40] F. F. Nascimento, M. dos Reis, and Z. Yang, “A biologist’s guide to Bayesian phylogenetic analysis,” *Nature Ecology & Evolution*, vol. 1, no. 10, pp. 1446–1454, 2017.

[41] D. Van Ravenzwaaij, P. Cassey, and S. D. Brown, “A simple introduction to Markov chain Monte–Carlo sampling,” *Psychonomic Bulletin & Review*, vol. 25, no. 1, pp. 143–154, 2018.

[42] D. Randall, “Rapidly mixing Markov chains with applications in computer science and physics,” *Computing in Science and Engg.*, vol. 8, no. 2, pp. 30–41, 2006.

[43] R. Martin and D. Randall, “Sampling adsorbing staircase walks using a new Markov chain decomposition method,” *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 492–502, 2000.

[44] J. Hermon and J. Salez, *Modified log-Sobolev inequalities for strong-Rayleigh measures*, 2019. arXiv: 1902.02775 `[math.PR]`.

[45] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[46] M. Garey, R. Graham, and J. Ullman, “An analysis of some packing algorithms,” *Combinatorial Algorithms*, pp. 39–47, 1973.

[47] D. S. Johnson, “Approximation algorithms for combinatorial problems,” *Journal of computer and system sciences*, vol. 9, no. 3, pp. 256–278, 1974.

[48] D.-Z. Du, K.-I. Ko, and X. Hu, *Design and analysis of approximation algorithms*. Springer Science & Business Media, 2011, vol. 62.

[49] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[50] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge university press, 2011.

[51] K. A. Matthews, W. Xu, A. H. Gaglioti, J. B. Holt, J. B. Croft, D. Mack, and L. C. McGuire, "Racial and ethnic estimates of Alzheimer's disease and related dementias in the United States (2015–2060) in adults aged $\geq$ 65 years," *Alzheimer's & Dementia*, vol. 15, no. 1, pp. 17–24, 2019.

[52] R. Brookmeyer, N. Abdalla, C. H. Kawas, and M. M. Corrada, "Forecasting the prevalence of preclinical and clinical Alzheimer's disease in the united states," *Alzheimer's & Dementia*, vol. 14, no. 2, pp. 121–129, 2018.

[53] M. Weiner and G. B. Frisoni, "Alzheimer's Disease Neuroimaging Initiative (ADNI) studies," *Neurobiology of aging*, vol. 31, no. 8, 2010.

[54] W. G. Rosen, R. C. Mohs, and K. L. Davis, "A new rating scale for Alzheimer's disease.," *The American journal of psychiatry*, 1984.

[55] M. D. Lezak, D. B. Howieson, D. W. Loring, J. S. Fischer, *et al.*, *Neuropsychological assessment*. Oxford University Press, USA, 2004.

[56] J. K. Kueper, M. Speechley, and M. Montero-Odasso, "The Alzheimer's disease assessment scale–cognitive subscale (ADAS-Cog): Modifications and responsiveness in pre-dementia populations. a narrative review," *Journal of Alzheimer's Disease*, vol. 63, no. 2, pp. 423–444, 2018.

[57] C. P. Hughes, L. Berg, W. Danziger, L. A. Coben, and R. L. Martin, "A new clinical scale for the staging of dementia," *The British journal of psychiatry*, vol. 140, no. 6, pp. 566–572, 1982.

[58] C. Lynch, C. Walsh, A. Blanco, M. Moran, R. Coen, J. Walsh, and B. Lawlor, "The clinical dementia rating sum of box score in mild dementia," *Dementia and geriatric cognitive disorders*, vol. 21, no. 1, pp. 40–43, 2006.

[59] S. E. O'Bryant, S. C. Waring, C. M. Cullum, J. Hall, L. Lacritz, P. J. Massman, P. J. Lupo, J. S. Reisch, R. Doody, T. A. R. Consortium, *et al.*, "Staging dementia using Clinical Dementia Rating Scale Sum of Boxes scores: A Texas Alzheimer's research consortium study," *Archives of neurology*, vol. 65, no. 8, pp. 1091–1095, 2008.

[60] A. D. Roses and A. M. Saunders, "APOE is a major susceptibility gene for Alzheimer's disease," *Current opinion in biotechnology*, vol. 5, no. 6, pp. 663–667, 1994.

[61] M. Safieh, A. D. Korczyn, and D. M. Michaelson, "ApoE4: An emerging therapeutic target for Alzheimer's disease," *BMC medicine*, vol. 17, no. 1, pp. 1–17, 2019.

[62] D. M. Michaelson, “APOE ε4: The most prevalent yet understudied risk factor for Alzheimer’s disease,” *Alzheimer’s & Dementia*, vol. 10, no. 6, pp. 861–868, 2014.

[63] K. A. Johnson, N. C. Fox, R. A. Sperling, and W. E. Klunk, “Brain imaging in Alzheimer disease,” *Cold Spring Harbor perspectives in medicine*, vol. 2, no. 4, a006213, 2012.

[64] C. Ledig, A. Schuh, R. Guerrero, R. A. Heckemann, and D. Rueckert, “Structural brain imaging in Alzheimer’s disease and mild cognitive impairment: Biomarker analysis and shared morphometry database,” *Scientific reports*, vol. 8, no. 1, pp. 1–16, 2018.

[65] T. A. team, *ADNIMERGE: Alzheimer’s Disease Neuroimaging Initiative*, R package version 0.0.1, 2021.

[66] J. Sluimer, H. Vrenken, M. Blankenstein, N. Fox, P. Scheltens, F. Barkhof, and W. Van Der Flier, “Whole-brain atrophy rate in Alzheimer disease: Identifying fast progressors,” *Neurology*, vol. 70, no. 19 Part 2, pp. 1836–1841, 2008.

[67] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo, M. Filippi, A. D. N. Initiative, *et al.*, “Automated classification of Alzheimer’s disease and mild cognitive impairment using a single MRI and deep neural networks,” *NeuroImage: Clinical*, vol. 21, p. 101 645, 2019.

[68] J. B. Bae, S. Lee, W. Jung, S. Park, W. Kim, H. Oh, J. W. Han, G. E. Kim, J. S. Kim, J. H. Kim, *et al.*, “Identification of Alzheimer’s disease using a convolutional neural network model based on T1-weighted magnetic resonance imaging,” *Scientific Reports*, vol. 10, no. 1, pp. 1–10, 2020.

[69] D. Shigemizu, S. Akiyama, S. Higaki, T. Sugimoto, T. Sakurai, K. A. Boroevich, A. Sharma, T. Tsunoda, T. Ochiya, S. Niida, *et al.*, “Prognosis prediction model for conversion from mild cognitive impairment to Alzheimer’s disease created by integrative analysis of multi-omics data,” *Alzheimer’s research & therapy*, vol. 12, no. 1, pp. 1–12, 2020.

[70] Y.-Y. Song and L. Ying, “Decision tree methods: Applications for classification and prediction,” *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in Python,” *The Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[72] J. A. Doudna, “Structural genomics of RNA,” *Nature Structural Biology*, vol. 7, no. 11, pp. 954–956, 2000.

[73] I. Tinoco Jr and C. Bustamante, "How RNA folds," *Journal of molecular biology*, vol. 293, no. 2, pp. 271–281, 1999.

[74] C. Massire and E. Westhof, "MANIP: An interactive tool for modelling RNA," *Journal of Molecular Graphics and Modelling*, vol. 16, no. 4, pp. 197–205, 1998.

[75] M. G. Seetin and D. H. Mathews, "Automated RNA tertiary structure prediction from secondary structure and low-resolution restraints," *Journal of Computational Chemistry*, vol. 32, no. 10, pp. 2232–2244, 2011.

[76] Y. Zhao, Z. Gong, and Y. Xiao, "Improvements of the hierarchical approach for predicting RNA tertiary structure," *Journal of Biomolecular Structure and Dynamics*, vol. 28, no. 5, pp. 815–826, 2011, PMID: 21294592.

[77] Y. Zhao, Y. Huang, Z. Gong, Y. Wang, J. Man, and Y. Xiao, "Automated and fast building of three-dimensional RNA structures," *Scientific Reports*, vol. 2, p. 734, 2012.

[78] J. A. Jaeger, D. H. Turner, and M. Zuker, "Improved predictions of secondary structures for RNA," *Proceedings of the National Academy of Sciences*, vol. 86, no. 20, pp. 7706–7710, 1989.

[79] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner, "Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure," *Journal of Molecular Biology*, vol. 288, no. 5, pp. 911–940, 1999.

[80] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner, "Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure," *Proceedings of the National Academy of Sciences*, vol. 101, no. 19, pp. 7287–7292, 2004.

[81] Y. Ding, C. Y. Chan, and C. E. Lawrence, "Sfold web server for statistical folding and rational design of nucleic acids," *Nucleic Acids Research*, vol. 32, no. suppl_2, W135–W141, Jul. 2004.

[82] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster, "Fast folding and comparison of RNA secondary structures," *Monatshefte für Chemie/Chemical Monthly*, vol. 125, no. 2, pp. 167–188, 1994.

[83] A. Mathuriya, D. A. Bader, C. E. Heitsch, and S. C. Harvey, "GTfold: A scalable multicore code for RNA secondary structure prediction," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 981–988.

[84] K. J. Doshi, J. J. Cannone, C. W. Cobaugh, and R. R. Gutell, "Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters

for RNA secondary structure prediction," *BMC Bioinformatics*, vol. 5, no. 1, p. 105, 2004.

[85] A. Kirkpatrick, K. Patton, P. Tetali, and C. Mitchell, "Markov chain-based sampling for exploring RNA secondary structure under the Nearest Neighbor Thermodynamic Model and extended applications," *Mathematical and Computational Applications*, vol. 25, no. 4, p. 67, 2020.

[86] Y. Bakhtin and C. E. Heitsch, "Large deviations for random trees and the branching of RNA secondary structures," *Bulletin of Mathematical Biology*, vol. 71, no. 1, pp. 84–106, 2009.

[87] A. Borodavka, S. W. Singaram, P. G. Stockley, W. M. Gelbart, A. Ben-Shaul, and R. Tuma, "Sizes of long RNA molecules are determined by the branching patterns of their secondary structures," *Biophysical journal*, vol. 111, no. 10, pp. 2077–2085, 2016.

[88] A. M. Yoffe, P. Prinsen, A. Gopal, C. M. Knobler, W. M. Gelbart, and A. Ben-Shaul, "Predicting the sizes of large RNA molecules," *Proceedings of the National Academy of Sciences*, vol. 105, no. 42, pp. 16 153–16 158, 2008.

[89] S. Janson, "Asymptotic normality of fringe subtrees and additive functionals in conditioned Galton-Watson trees," *Random Structures & Algorithms*, vol. 48, no. 1, pp. 57–101, 2016.

[90] L. Takács, "A Bernoulli excursion and its various applications," *Advances in Applied Probability*, vol. 23, no. 3, pp. 557–585, 1991.

[91] S. Janson, "The Wiener index of simply generated random trees," *Random Structures & Algorithms*, vol. 22, no. 4, pp. 337–358, 2003.

[92] J. Riordan, *Combinatorial identities*. Wiley, 1968.

[93] T. Venkata Narayana, "Sur les treillis formés par les partitions d'un entier et leurs applications à la théorie des probabilités," *C. R. Acad. Sci. Paris*, vol. 240, pp. 1188–1189, 1955.

[94] N. Dershowitz and S. Zaks, "Enumerations of ordered trees.," *Discrete Mathematics*, vol. 31, no. 1, pp. 9–28, 1980.

[95] R. Donaghey and L. W. Shapiro, "Motzkin numbers," *Journal of Combinatorial Theory, Series A*, vol. 23, no. 3, pp. 291–301, 1977.

[96] N. Dershowitz and S. Zaks, "The cycle lemma and some applications," Computer Science Department, Technion, Tech. Rep., 1982.

[97] D. H. Turner and D. H. Mathews, "NNDB: The nearest neighbor parameter database for predicting stability of nucleic acid secondary structure," *Nucleic Acids Research*, vol. 38, no. suppl_1, pp. D280–D282, Oct. 2009.

[98] C. Heitsch and S. Poznanović, "Combinatorial insights into RNA secondary structure," in *Discrete and topological models in molecular biology*, Springer, 2014, pp. 145–166.

[99] R. Lorenz, S. H. Bernhart, C. H. Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker, "ViennaRNA package 2.0," *Algorithms for molecular biology*, vol. 6, no. 1, p. 26, 2011.

[100] R. Donaghey and L. W. Shapiro, "Motzkin numbers," *J. Combinatorial Theory Ser. A*, vol. 23, no. 3, pp. 291–301, 1977.

[101] F. R. Bernhart, "Catalan, Motzkin, and Riordan numbers," *Discrete Math.*, vol. 204, no. 1-3, pp. 73–112, 1999.

[102] S.-P. Eu, T.-S. Fu, J. T. Hou, and T.-W. Hsu, "Standard Young tableaux and colored Motzkin paths," *J. Combin. Theory Ser. A*, vol. 120, no. 7, pp. 1786–1803, 2013.

[103] J.-L. Baril, S. Kirgizov, and A. Petrossian, "Motzkin paths with a restricted first return decomposition," *Integers*, vol. 19, Paper No. A46, 19, 2019.

[104] W. Fang, "A partial order on Motzkin paths," *Discrete Math.*, vol. 343, no. 5, pp. 111802, 9, 2020.

[105] R. P. Stanley, *Enumerative Combinatorics: Volume 1*, 2nd. New York, NY, USA: Cambridge University Press, 2011.

[106] E. Deutsch and L. W. Shapiro, "A bijection between ordered trees and 2-Motzkin paths and its many consequences," *Discrete Mathematics*, vol. 256, no. 3, pp. 655–670, 2002, LaCIM 2000 Conference on Combinatorics, Computer Science and Applications.

[107] N. Madras and D. Randall, "Markov chain decomposition for convergence rate analysis," *Ann. Appl. Probab.*, vol. 12, no. 2, pp. 581–606, May 2002.

[108] M. Luby, D. Randall, and A. Sinclair, "Markov chain algorithms for planar lattice structures," *SIAM Journal on Computing*, vol. 31, no. 1, pp. 167–192, 2001.

[109] M. Jerrum, J.-B. Son, P. Tetali, and E. Vigoda, "Elementary bounds on Poincaré and log-Sobolev constants for decomposable Markov chains," *Ann. Appl. Probab.*, vol. 14, no. 4, pp. 1741–1765, 2004.

[110] E. Cohen, "Problems in Catalan mixing and matchings in regular hypergraphs," Ph.D. dissertation, Georgia Institute of Technology, 2016.

[111] D. B. Wilson, "Mixing times of lozenge tiling and card shuffling Markov chains," *Ann. Appl. Probab.*, vol. 14, no. 1, pp. 274–325, Feb. 2004.

[112] M. Andronescu, V. Bereg, H. H. Hoos, and A. Condon, "RNA STRAND: The RNA secondary structure and statistical analysis database," *BMC bioinformatics*, vol. 9, no. 1, p. 340, 2008.

[113] L. Alonso, "Uniform generation of a Motzkin word," *Theoretical Computer Science*, vol. 134, no. 2, pp. 529–536, 1994.

[114] M. E. Nebel and A. Scheid, "Evaluation of a sophisticated SCFG design for RNA secondary structure prediction," *Theory in Biosciences*, vol. 130, no. 4, pp. 313–336, 2011.

[115] E. Rivas and S. R. Eddy, "Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs," *Bioinformatics*, vol. 16, no. 7, pp. 583–605, 2000.

[116] ——, "A dynamic programming algorithm for RNA structure prediction including pseudoknots," *Journal of molecular biology*, vol. 285, no. 5, pp. 2053–2068, 1999.

[117] B. Knudsen and J. Hein, "Pfold: RNA secondary structure prediction using stochastic context-free grammars," *Nucleic acids research*, vol. 31, no. 13, pp. 3423–3428, 2003.

[118] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to MCMC for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[119] S. Chib, "Introduction to simulation and MCMC methods," in *The Oxford Handbook of Bayesian Econometrics*, 2011.

[120] S. Jackman, "Estimation and inference via Bayesian simulation: An introduction to Markov chain Monte Carlo," *American journal of political science*, pp. 375–404, 2000.

[121] A. Gelman and D. B. Rubin, "Markov chain Monte Carlo methods in biostatistics," *Statistical Methods in Medical Research*, vol. 5, no. 4, pp. 339–355, 1996.

[122] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. American Mathematical Society, Providence, RI, 2009, pp. xviii+371, With a chapter by James G. Propp and David B. Wilson, ISBN: 978-0-8218-4739-8.

[123] R. R. Montenegro and P. Tetali, *Mathematical aspects of mixing times in Markov chains*. Now Publishers Inc, 2006.

[124] D. J. Aldous, “Mixing time for a markov chain on cladograms,” *Combinatorics, Probability and Computing*, vol. 9, no. 3, pp. 191–204, 2000.

[125] J. Schweinsberg, “An $o(n^2)$ bound for the relaxation time of a Markov chain on cladograms,” *Random Structures & Algorithms*, vol. 20, no. 1, pp. 59–70, 2002.

[126] N. Dershowitz and S. Zaks, “Ordered trees and non-crossing partitions,” *Discrete Mathematics*, vol. 62, no. 2, pp. 215–218, 1986.

[127] E. Cohen, P. Tetali, and D. Yeliussizov, “Lattice path matroids: Negative correlation and fast mixing,” *arXiv preprint arXiv:1505.06710*, 2015.

[128] L. McShine and P. Tetali, “On the mixing time of the triangulation walk and other Catalan structures,” in *Randomization Methods in Algorithm Design (Princeton, NJ, 1997)*, ser. DIMACS Ser. Discrete Math. Theoret. Comput. Sci. Vol. 43, Amer. Math. Soc., Providence, RI, 1999, pp. 147–160.

[129] M. Sohoni, “Rapid mixing of some linear matroids and other combinatorial objects,” *Graphs Combin.*, vol. 15, no. 1, pp. 93–107, 1999.

[130] C. P. Robert, G. Casella, and G. Casella, *Introducing Monte Carlo methods with R*. Springer, 2010, vol. 18.

[131] C. E. Heitsch and P. Tetali, “Meander graphs,” in *23rd International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2011)*, ser. Discrete Math. Theor. Comput. Sci. Proc., AO, Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2011, pp. 469–480.

[132] K. McCoy, S. Gudapati, L. He, E. Horlander, D. Kartchner, S. Kulkarni, N. Mehra, J. Prakash, H. Thenot, S. V. Vanga, *et al.*, “Biomedical text link prediction for drug discovery: A case study with COVID-19,” *Pharmaceutics*, vol. 13, no. 6, p. 794, 2021.

[133] X. Zeng, Y. Liao, Y. Liu, and Q. Zou, “Prediction and validation of disease genes using HeteSim scores,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 14, no. 3, pp. 687–695, 2016.

[134] Y. Xiao, J. Zhang, and L. Deng, “Prediction of lncRNA-protein interactions using HeteSim scores based on heterogeneous networks,” *Scientific reports*, vol. 7, no. 1, pp. 1–12, 2017.

[135] J. Qu, X. Chen, Y.-Z. Sun, Y. Zhao, S.-B. Cai, Z. Ming, Z.-H. You, and J.-Q. Li, "In silico prediction of small molecule-miRNA associations based on the HeteSim algorithm," *Molecular Therapy-Nucleic Acids*, vol. 14, pp. 274–286, 2019.

[136] X. Chen, W. Shi, and L. Deng, "Prediction of disease comorbidity using HeteSim scores based on multiple heterogeneous networks," *Current gene therapy*, vol. 19, no. 4, pp. 232–241, 2019.

[137] C. Fan, X. Lei, L. Guo, and A. Zhang, "Predicting the associations between microbes and diseases by integrating multiple data sources and path-based HeteSim scores," *Neurocomputing*, vol. 323, pp. 76–85, 2019.

[138] J. Wang, Z. Kuang, Z. Ma, and G. Han, "GBDTL2E: Predicting lncRNA-EF associations using diffusion and HeteSim features based on a heterogeneous network," *Frontiers in genetics*, vol. 11, p. 272, 2020.

[139] A. Klementiev, D. Roth, and K. Small, "An unsupervised learning algorithm for rank aggregation," in *European Conference on Machine Learning*, Springer, 2007, pp. 616–623.

[140] The Python Software Foundation, *Time*, version 3.6.10, Jan. 7, 2020.

[141] M. Gorelick and I. Ozsvald, *High Performance Python: Practical Performant Programming for Humans*. O'Reilly Media, 2020.

[142] Project Jupyter, *Jupyter notebook*, version 5.5.0, May 23, 2018.

[143] The Python Software Foundation, *Python*, version 3.6.10, Jan. 7, 2020.

[144] N. Alon and J. H. Spencer, *The Probabilistic Method*. John Wiley & Sons, 2004.

[145] C. McDiarmid, "On the method of bounded differences," *Surveys in combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[146] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[147] G. M. Viswanathan, V. Afanasyev, S. Buldyrev, E. Murphy, P. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.

[148] F. Bartumeus, M. G. E. da Luz, G. M. Viswanathan, and J. Catalan, "Animal search strategies: A quantitative random-walk analysis," *Ecology*, vol. 86, no. 11, pp. 3078–3087, 2005.

[149] F. Bartumeus, J. Catalan, U. Fulco, M. Lyra, and G. Viswanathan, “Optimizing the encounter rate in biological interactions: Lévy versus Brownian strategies,” *Physical Review Letters*, vol. 88, no. 9, p. 097 901, 2002.

[150] A. Mårell, J. P. Ball, and A. Hofgaard, “Foraging and movement paths of female reindeer: Insights from fractal analysis, correlated random walks, and Lévy flights,” *Canadian Journal of Zoology*, vol. 80, no. 5, pp. 854–865, 2002.

[151] J. B. Slowinski, “Probabilities of n-trees under two models: A demonstration that asymmetrical interior nodes are not improbable,” *Systematic Zoology*, vol. 39, no. 1, pp. 89–94, 1990.

[152] W. P. Maddison and M. Slatkin, “Null models for the number of evolutionary steps in a character on a phylogenetic tree,” *Evolution*, vol. 45, no. 5, pp. 1184–1197, 1991.

[153] M. Kirxpatrick and M. Slatkin, “Searching for evolutionary patterns in the shape of a phylogenetic tree,” *Evolution*, vol. 47, no. 4, pp. 1171–1181, 1993.

[154] M. Fischer, M. Galla, L. Herbst, and M. Steel, “The most parsimonious tree for random data,” *Molecular phylogenetics and evolution*, vol. 80, pp. 165–168, 2014.

[155] B. H. Juang and L. R. Rabiner, “Hidden Markov models for speech recognition,” *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[156] M. Gales and S. Young, “The application of hidden Markov models in speech recognition,” 2008.