INFERENCE OF ANCESTRAL PROTEIN-PROTEIN INTERACTIONS USING METHODS FROM ALGEBRAIC STATISTICS

by

Ashok Rajaraman

B.Tech (Metal. & Mat. Eng.), Indian Institute of Technology (Roorkee), 2009

THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN THE

DEPARTMENT OF MATHEMATICS

FACULTY OF SCIENCE

SIMON FRASER UNIVERSITY
Summer 2011

APPROVAL

Name: Ashok Rajaraman

Degree: Master of Science

Title of Thesis: Inference of Ancestral Protein-Protein Interactions using methods from Algebraic Statistics

Examining Committee: Dr. Marni Mishna (Chair)

Dr. Cedric Chauve (Senior supervisor)

Dr. Nilima Nigam (Supervisor)

Dr. Oliver Schulte (Supervisor)

Dr. Jason Bell (Examiner)

Date Approved: August 4, 2011

Abstract

Protein-protein interactions are important catalysts for many biological functions. The interaction networks of different organisms may be compared to investigate the process of evolution through which these structures evolve. The parameters used for inference models for such evolutionary processes are usually hard to estimate.

This thesis explores approaches developed in algebraic statistics for parametric inference in probabilistic models. Here, we apply the parametric inference approach to Bayesian networks representing the evolution of protein interaction networks. More precisely, we modify the belief propagation algorithm for Bayesian inference for a polytope setting. We apply our program to analyze both simulated and real protein interaction data and compare the results to two well known discrete parsimony inference methods.

To Mom and Dad. I promise I will take the time to explain this- one day.

Acknowledgments

I would like to thank my senior supervisor Dr. Cedric Chauve, who introduced me to evolutionary models, patiently went through the many iterations of this thesis and offered insights that helped shape the structure and content of this thesis.

I would also like to thank Dr. Michael Joswig of Technische Universität Darmstadt, who provided us with the SVN version of the polymake software, and helped us greatly in using the same.

The computation performed made extensive use of resources provided by Westgrid, as well as those at the Pacific Institute of Mathematical Sciences, SFU. I extend my thanks to the system administrators for these resources, who were kind enough to make special allocation for the massive computational resources used.

I also believe I have shaped into a better mathematician through the efforts of all my teachers here and in the past, and a better person thanks to my friends. I thank them for the knowledge, mathematical or otherwise, that I have learned from them.

Contents

List of Tables

List of Figures

Introduction

Protein-protein interactions are important biological phenomena, which participate in many major functions in cells [11]. Most proteins carry out their functions by interacting with other proteins. The interactions in a single species form a biological network. The study of these interactions is crucial to understanding such networks.

Evolution is an important field of research in biology, and nothing in biology makes sense except in the light of evolution [13]. However, understanding the evolutionary history of biological networks, such as the network of protein-protein interactions, is still a widely open problem [26].

There has been tremendous progress in data acquisition for molecular biology, and through this, the protein-protein interaction networks for current species have been made available to us [27]. This opens the way for methods in computational biology to be used to infer the evolutionary history of protein-protein interactions networks.

The purpose of this thesis is to give a brief overview of the work done on the inference of the evolutionary history of these interaction networks, and to serve as a preliminary exploration of an algebraic statistics approach to the problem of inferring them [33].

The first part of the thesis is an introduction to mathematical models in molecular and evolutionary biology. The basic concepts of protein-protein interactions and structures used to model their evolution are given here. Furthermore, we discuss various techniques used to infer ancestral protein-protein interactions.

The main emphasis is on probabilistic graphical models, in particular Bayesian networks on trees, which are the objects of interest for the thesis. A probabilistic approach to inference is desirable as probabilistic models are more realistic models of evolution. Bayesian networks have been used for the prediction of ancestral protein-protein interactions, and have compared well with other techniques [35]. Apart from probabilistic techniques, we also discuss some deterministic approaches to the problem, and the principles that govern them.

The use of probabilistic models also implies that a variety of efficient algorithms are available to us for inference. For example, the forward algorithm for hidden Markov models is a special case of a family of algorithms known as sum-product algorithms. Apart from simple inference, there are also optimization techniques available to us on probabilistic models. This makes these models versatile and relatively easy to use.

The second part of the thesis an algebraic statistics point of view on probabilistic inference, introduced by Sturmfels et al. [33]. Approaching computational biology through algebraic statistics is a relatively new idea, which develops naturally from the use of probabilistic models for inference, and has been applied to sequence alignment using hidden Markov models [5]. The main motivation behind this approach is that sum-product algorithms yield polynomials when the parameters are treated as formal variables, and every probabilistic model can be represented by a polynomial map. The algorithms for probabilistic graphical models translate well when we move the problem of inference to algebraic statistics, which leads to parametric inference algorithms.

The final part of the thesis applies methods in algebraic statistics to Bayesian networks which describe the evolution of protein-protein interactions. The methods are applied to both simulated data and data from real interactions. The results are compared to well known deterministic approaches to inference, and to theoretical calculations of complexity bounds. A brief summary of our results and the major issues we face is given at the end.

Part I

Computational Biology

Chapter 1

Introduction to Computational Molecular Biology

The objective of the thesis is to reconstruct ancestral protein interaction networks. To do so, we need a working definition of certain biological terms which occur frequently in bioinformatics research, and which serve as a basis to the system we are concerned with.

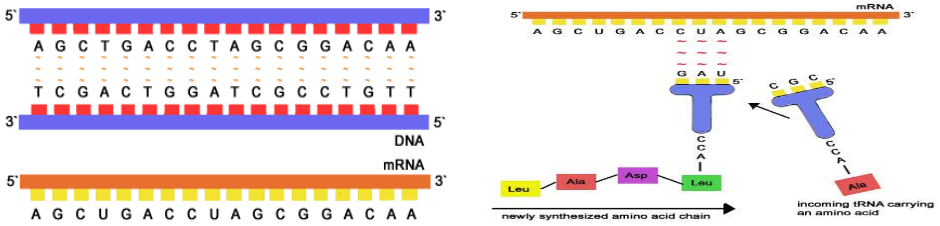
1.1 Genomes, genes and proteins

A *genome* is a molecule of DNA made of four nucleic acids. It is composed of *chromosomes*, which carry genes. The genome is present in the nucleus of each cell of an organism [23]. It is the support of genetic material.

A *gene* is a genome segment that encodes a protein [23]. Genes can be ‘read’, through a process known as *transcription*, to create a *messenger-RNA*. The RNA molecule is *translated* then into a protein.

Proteins are macromolecules formed by sequences of amino acids. They are important units for biological functions, often used as catalysts for biological reactions, providing structure to components, or signalling cells etc.

As stated before, proteins are formed by translation of an RNA molecule. Substrings of size 3 of


5'
3'
A G C T G A C C T A G C G G A C A A
T C G A C T G G A T C G C C T G T T
3'
5'
DNA
mRNA
5'
3'
A G C U G A C C U A G C G G A C A A
mRNA
5'
3'
A G C U G A C C U A G C G G A C A A
G A U
5'
C G C
5'
C C A
C C A
Leu
Ala
Asp
Leu
Ala
newly synthesized amino acid chain
incoming tRNA carrying an amino acid

(a) Transcription from DNA to create m-RNA

(b) Translation of m-RNA to protein

Figure 1.1: From genes to proteins

the RNA, known as *codons*, encode amino acids, and a series of codons encodes the whole protein.

1.2 Protein-protein interactions

A protein-protein interaction occurs when two or more proteins bind together to carry out a biological function [11, 25]. Protein-protein interactions (which we shall refer to by protein interaction for convenience) participate in many major biological processes, and this makes their study interesting.

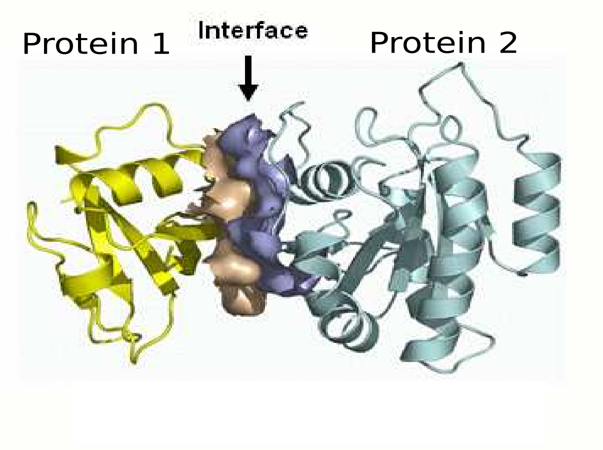


Figure 1.2: A Protein-Protein Interaction [43]

Protein interactions are an example of *networks* in biology [26]. Each protein is represented by a vertex, and an edge is present between two vertices if and only if the corresponding proteins interact.

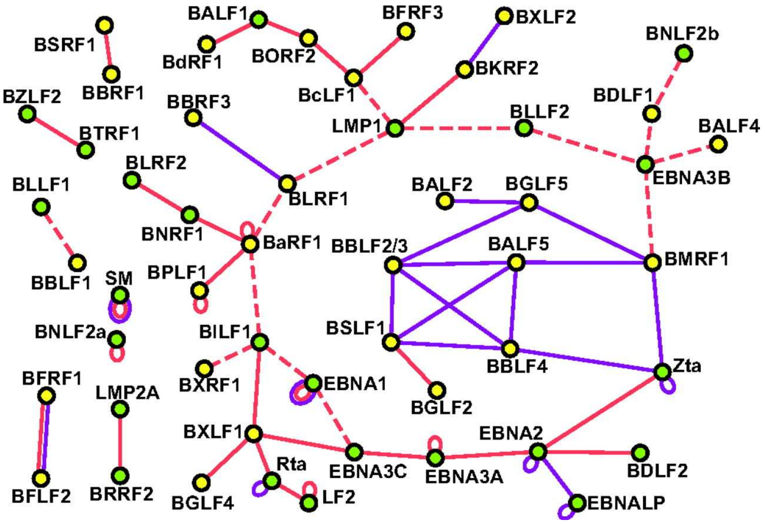


Figure 1.3: A Protein-Protein Interaction network in the Epstein-Barr virus [6]

Some proteins that evolve from the same ancestral protein, called a *protein family*, may show a large number of interactions with proteins of the same type.

1.3 Evolution

Evolution is the process through which inherited traits in organisms change over time. Evolutionary information is stored in the genome, and is inherited by the child from the parent.

Speciation is the process through which a species evolves into two or more descendant species. Each species has its representative genome. Once a speciation occurs, each species evolves along its own branch, independently of the other species.

Genes within a genome evolve through duplication, speciation, and loss [23]. Duplication creates two copies of the gene in the same genome. Through speciation, the two new genomes inherit the gene.

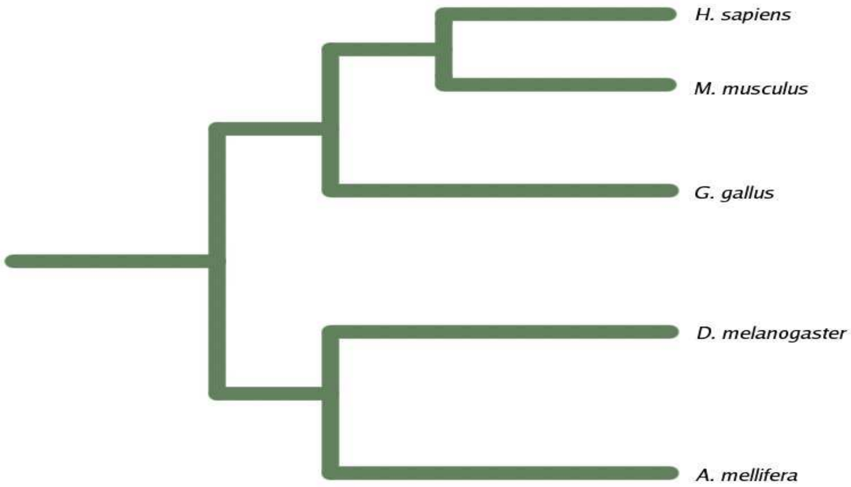


Figure 1.4: Evolution of humans, mice, chicken, fruit flies and bees from a common ancestor

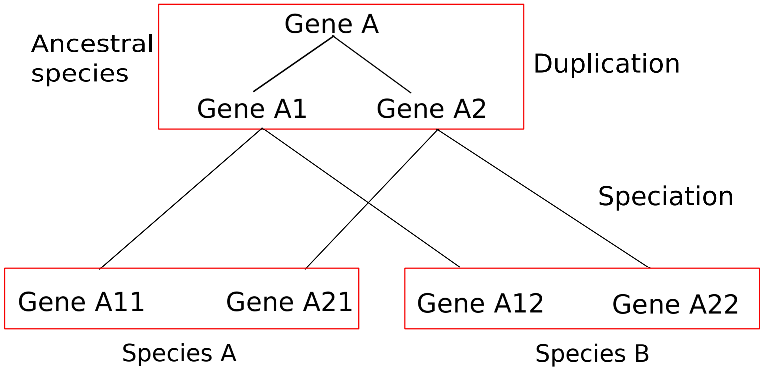


Figure 1.5: Gene evolution through speciation and duplication

Gene loss occurs after a species inherits a gene. The gene generally gets duplicated, and, due to the accumulation of mutations on one of the duplicated copies, the copy either loses its function and becomes a *pseudogene*, or it develops a new function [23]. Genes, through messenger RNA, produce proteins. It is reasonable to assume that changes in the gene sequence will lead to changes in the protein produced by that gene. So, proteins are assumed to evolve in parallel with genes.

The evolution of proteins also affects protein-protein interactions. Following a speciation event of

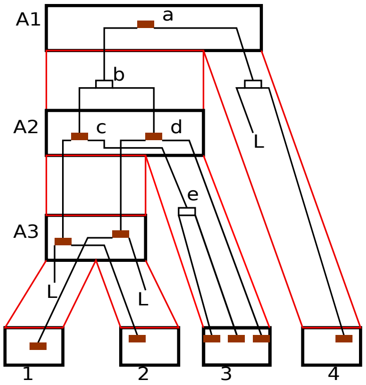


Figure 1.6: Gene loss. Genes have been lost over the branches marked with L at the leaves.

two proteins which may have interacted in the ancestral species, their immediate descendants in each of the new species may start interacting. If a protein is lost in a new species during speciation, then no new interactions with that protein are possible. The duplication of a protein means that the two resulting copies of the protein in the species can potentially interact with every other protein that their parent was interacting with.

A major question asked by biologists is what biological information we can infer about species that are now extinct. The main obstacle to answering this question is that we do not know the evolutionary process well enough to predict the path of evolution with certainty.

1.4 Mathematical models in evolutionary biology

This section discusses basic mathematical models that are used to represent the evolution of species, genes, proteins, and protein interactions.

1.4.1 Phylogenetic trees

The main combinatorial object of interest in phylogenetics is the tree.

Definition 1. *A tree is a connected, undirected, acyclic graph.*

Nodes of a tree with degree greater than 1 are called *internal nodes*, and nodes with degree 1 are called *leaf nodes*. If a node of the tree is oriented in such a way that all edges are orientated towards or away from it, the node is called the *root* of the tree. We can also define directed trees, by assigning a direction to each edge. The source of a directed edge is called a *parent*, and the sink is called a *child* of the parent.

We say that a tree $\mathcal{T}$ is defined on the alphabet of leaves L if the set L forms the set of leaves of the tree.

Trees are a very natural and simple models for evolution. Given a rooted tree, and an orientation of the edges such that each edge is directed away from the root, along each edge, the child is assumed to evolve from the parent. For an internal node v in the tree, its descendants are defined to be the set of nodes that lie on the directed paths from v to the leaves in the subtree rooted at v, including the leaves.

Species tree

In biology, the set of species that are currently alive are called *extant species*. The set of species that have died out, and through which evolution progressed are called *extinct species*. We define a species tree as follows.

Definition 2. *Let X be a set of extant species. A* species tree *is a tree defined on the alphabet of leaves X.*

Alternately, a species tree is a tree defined on a set of hypothetical extinct species Y, with leaves X, such that $Y \cap X = \emptyset$, and a directed edge exists between each pair of nodes $v_1, v_2 \in X \cup Y$ when the species v_2 is a direct descendant of the species v_1. Thus, every internal node in the species tree represents an ancestral species, and the branching at an internal node to two edges represents a speciation. For example, Figure 1.4 is a species tree over 5 species. Species trees with full information about the species are rooted and binary. However, full information might be hard to obtain, and very often, we resort to non-binary or unrooted species trees on a set of extant species.

Gene tree

The evolution of individual genes can also be modelled by a tree. We can define a gene tree as follows.

Definition 3. *Let X be a set of genes belonging to extant species. A* gene tree *is a tree defined on the alphabet of leaves X.*

Each internal node in the gene tree represents an ancestral gene, and the branching at each internal node is either a speciation event or a duplication event.

Gene trees, like species trees, may be rooted or unrooted. The root of a gene tree, if it exists, corresponds to the most recent common ancestral gene of all the genes at the leaf nodes.

Gene trees are often constructed with weights on the edges. These weights represent the amount of sequence divergence between the two genes on either end of the edge.

Reconciliation

An important problem is the identification of the species that each gene in the gene tree belongs to. For the leaves, this is straightforward, since we only have extant genes, and we know the source. However, the internal nodes of the gene tree are not labelled with species names. Thus, we use the species tree for the set of extant species, and identify the internal nodes with ancestral species in the tree.

At each of the internal nodes in the gene tree, we could have had a gene duplication, speciation, or loss. If we had a duplication, the children will belong to the same species as the parent. If the node was a speciation node, its children will belong to different species. Losses occur at nodes in which one of the two children is lost, i.e. the gene is not present in that branch of the tree.

The process of reconciliation identifies each internal node of a gene tree with a species in the species tree, and associates a speciation or duplication event to each node in the tree [7–9]. This is often done with respect to some optimization criterion.

In the following example, the gene tree is given, with white boxes labelled with small letters a, b, c, d, e representing ancestral genes, and the solid boxes representing extant genes. In the species tree, the capital letters A_1, A_2, A_3 represent ancestral species, and the numbers represent extant species. The numbers at the leaves of the gene tree identify each gene to the species it belongs to. Denote the species tree by S, and the gene tree by G. The edges of a tree T will be given by $E(T)$, and the vertices by $V(T)$. Let $L_S(X)$ be the set of species of the leaves in the subtree rooted at $X \in V(S)$ of the species tree. Similarly, let $L_G(X)$ be the set of species of the leaves in

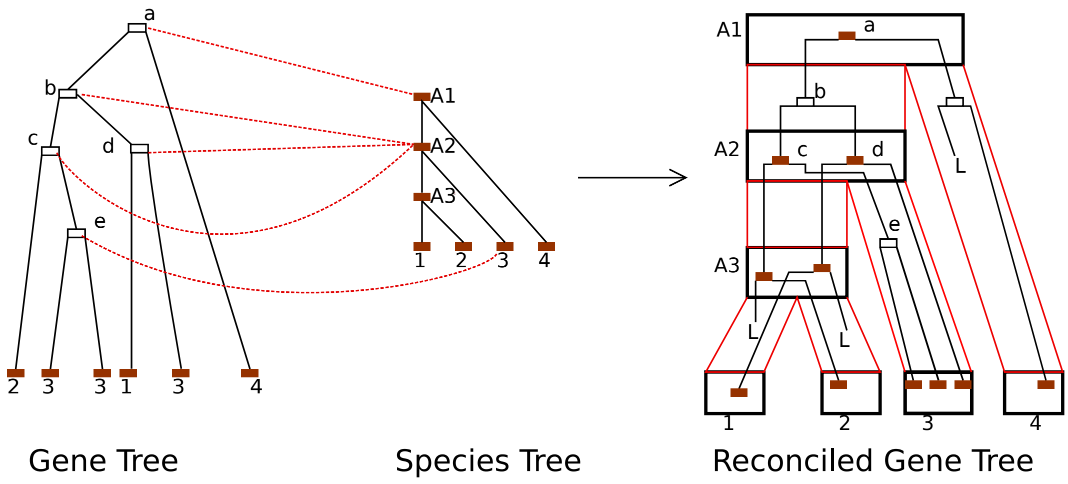


Figure 1.7: Reconciliation of a gene tree with a species tree. Solid nodes are speciations, and empty nodes are duplications.

the subtree rooted at $X \in V(G)$ of the gene tree.

Definition 4. [9] *A reconciliation is a mapping $LCA : V(G) \mapsto V(S)$ such that $LCA(X) = U$ for $X \in V(G)$, and $U \in V(S)$ is the lowest node of S such that $L_G(X) = L_S(U)$.*

This reconciliation technique minimizes the number of duplications, losses, and the total number of duplication and loss events [9]. In the reconciled gene tree in the example, solid internal nodes represent speciating genes, and white ones represent duplicating genes. Branches marked L represent losses. Node d in the gene tree has to at least map to species $A2$ in the species tree or higher, since genes of species 3 could not have evolved from $A3$. If d was mapped to $A1$, however, the number of duplication and loss events would increase.

1.4.2 Protein interaction networks

The protein interaction network for a set of proteins in a species can be modelled by an *interaction graph*.

Interaction graph

Definition 5. *Let V be a set of proteins in some organism, and E be the set of all interactions, such that for any two proteins u and v in V, we say that $\{u, v\} \in E$ if we have an interaction between u and v. The interaction graph for that set of proteins is the graph $G = (V, E)$ with vertex set V and edge set E.*

If the set V is the entire set of proteins in the organism, then we get the entire protein interaction network for the organism. Small families of proteins may show dense subgraphs with few edges with proteins in other families.

Proteins can also interact with copies of themselves. Such interactions are called *homodimer interactions*. If proteins interact with other proteins, the interactions are called *heterodimer interactions*.

Given a set of protein interaction networks of different species, finding the protein interaction network in the ancestor would naively translate into identifying similar proteins in all species, and finding the induced subgraph. This is almost akin to solving the subgraph isomorphism problem, which is NP-complete. This, of course, does not take into account protein duplication and loss.

Interaction tree

A more useful idea to model the evolution of protein interactions involves making the assumption that each protein interaction is independent of other interactions. This leads to the concept of an interaction tree, first described by Pinney et al. [35].

Definition 6. *An interaction tree is a rooted, directed tree of maximum degree outdegree 3, which describes the evolution of protein interactions. The nodes of an interaction tree represent possible protein interactions. The branches of an interaction tree represent the effect of duplication, speciation and loss of proteins on the evolution of protein-protein interactions.*

Interaction trees are constructed for one or more families of proteins over the same set of extant species from the corresponding gene trees that represent the evolution for these families. The maximum outdegree condition stems from using rooted gene trees with branch lengths given for each edge, as will be seen during the construction of the interaction tree.

Given a rooted, edge-weighted, reconciled gene tree G for a gene family, and the corresponding proteins, it is possible to construct the interaction tree for the proteins in this family as follows.

(i) For every two proteins $A, B \in V(G)$, not necessarily distinct, in the same species, add the node $\{A, B\}$ to the vertex set of a new graph $\mathcal{I}$.

(ii) For a duplication, where the edge to node A is shorter than the edge to node B in the gene tree, to give proteins A_1 and A_2, add edges from the node $\{A, B\}$ to the nodes $\{A_1, B\}$ and $\{A_2, B\}$ to $\mathcal{I}$.

(iii) For a speciation of node A to A_1 and A_2, and of node B to B_1 and B_2, where the proteins labelled 1 and 2 belong to different species, add edges from $\{A, B\}$ to $\{A_1, B_1\}$ and $\{A_2, B_2\}$.

(iv) For a node $\{A, A\}$, if A duplicates to give A_1 and A_2, add edges from $\{A, A\}$ to $\{A_1, A_1\}$, $\{A_1, A_2\}$ and $\{A_2, A_2\}$.

(v) Delete all isolated nodes.

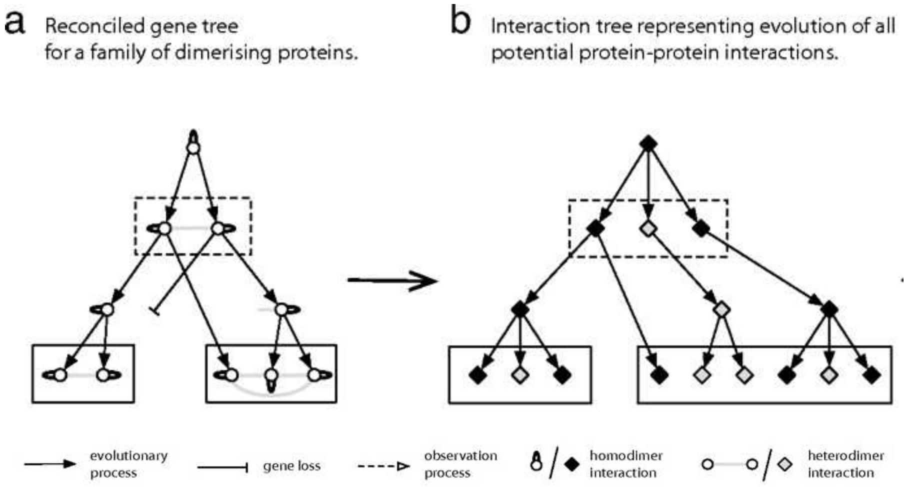

a
Reconciled gene tree
for a family of dimerising proteins.
b
Interaction tree representing evolution of all
potential protein-protein interactions.
evolutionary process
gene loss
observation process
homodimer interaction
heterodimer interaction

Figure 1.8: Constructing an interaction tree from a gene tree [35]

The tree $\mathcal{I}$ thus constructed is the interaction tree of the two protein families. The evolution of homodimer duplications is represented in the tree by nodes with an outdegree of 3.

Similar models have also been proposed for describing the interactions between two different families of proteins [15]. In the next chapter, we shall see how these models are used to infer the evolutionary history of ancestral protein-protein interactions.

Chapter 2

Inference of Ancestral Characters

Inference of ancestral characters in evolutionary biology makes use of the tree structure of evolution. Such a structure implies that the evolution of two disjoint branches is independent of each other. Inference techniques can be broadly classified into deterministic and probabilistic approaches.

2.1 Deterministic Approaches- Parsimony

The principle of parsimony states that the process of evolution would be carried out with the minimal number of character changes in the evolutionary tree [17]. The change of a character from a parent to a child is called a transition.

2.1.1 Fitch Parsimony

Fitch parsimony is a simple concept which states that the path of evolution is the one with the least number of changes [18]. This means that there is no preference for any transition. The algorithm for constructing an evolutionary scenario which obeys Fitch parsimony minimizes the number of such transitions in the model.

Algorithm for Fitch Parsimony

```
Input: Tree $\mathcal{T}$, character $C_L$ at each leaf $L$ in the tree, set $\mathcal{C}_X$ of possible characters at each
       node $X$.
Output: Character $C_X$ at each internal node $X$ of the tree, such that the number of
        transitions is minimized.
foreach Node $X$ in $\mathcal{T}$ do
    if $X \notin Leaves\,(\mathcal{T})$ then
        if $\bigcap_{Y \in \textit{Children of } X} \mathcal{C}_Y \neq \emptyset$ then
            $\mathcal{C}_X = \bigcap_{Y \in \text{Children of } X} \mathcal{C}_Y$;
        end
        else if $\bigcap_{Y \in \textit{Children of } X} \mathcal{C}_Y == \emptyset$ then
            $\mathcal{C}_X = \bigcup_{Y \in \text{Children of } X} \mathcal{C}_Y$;
        end
    end
end
$changes = 0$;
foreach Node $X$ in $\mathcal{T}$ do
    if $X == Root\,(\mathcal{T})$ then
        Choose character $c \in \mathcal{C}_X$;
        $\mathcal{C}_X = \{c\}$;
    end
    else
        if $\mathcal{C}_X \cap \mathcal{C}_{Parent(X)} \neq \emptyset$ then
            Choose character $c \in \mathcal{C}_X \cap \mathcal{C}_{Parent(X)}$;
            $C_X = \{c\}$;
        end
        else
            Choose character $c \in \mathcal{C}_X$;
            $C_X = \{c\}$;
            changes$++$;
        end
    end
end
```

The algorithm for Fitch parsimony on a tree is executed in a two step process. The upward pass compiles a set of all possible characters at a node. The downward pass chooses characters from these sets that minimize the number of transitions. The example given shows a Fitch parsimonious labelling of the tree obtained from the algorithm. Given the evidence at the leaves, we have only two gains in the entire tree.

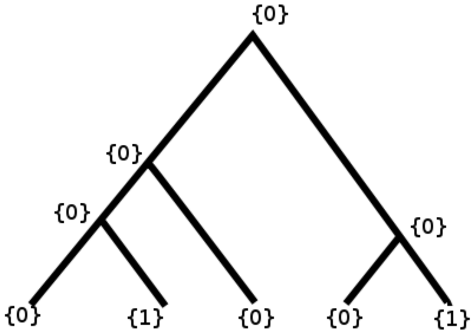


Figure 2.1: A tree labelled with Fitch parsimony, given the labels at the leaves

Fitch parsimony is the term used for applying the concept to binary trees. For non-binary trees, the corresponding concept is called Fitch-Hartigan parsimony.

2.1.2 Sankoff Parsimony

The main drawback of Fitch-Hartigan parsimony is that the transitions of all characters are considered equally likely. Sankoff parsimony seeks to remedy this limitation by assigning costs to each transition, and stating that the most likely scenario would have been one which yields the least total cost [37, 38].

The Sankoff parsimony scenario can be computed using dynamic programming. It then remains to determine the cost of each transition. In particular, the case in which all transitions are assigned equal costs reduces to Fitch-Hartigan parsimony.

2.1.3 Dollo Parsimony

Another special case of Sankoff parsimony is called Dollo parsimony. The Dollo principle states that *complex characters* which were formed during evolution are very hard to gain, but relatively easy to lose. Dollo parsimony is a condition on the existence or absence of a complex character. Thus, we only have binary transitions, from 0(absence) to 1(existence), or vice-versa. Furthermore, since characters are considered hard to gain, evolutionary scenarios are restricted to have at most one gain, while minimizing the number of losses.

In terms of Sankoff parsimony costs, the Dollo argument corresponds to the condition that the cost of going from 0 to 1 is infinite. On a tree describing evolution, this means that if the character is present at two leaves, since we could have had at most one gain, the character must be present at each node which lies on the path between the two leaves.

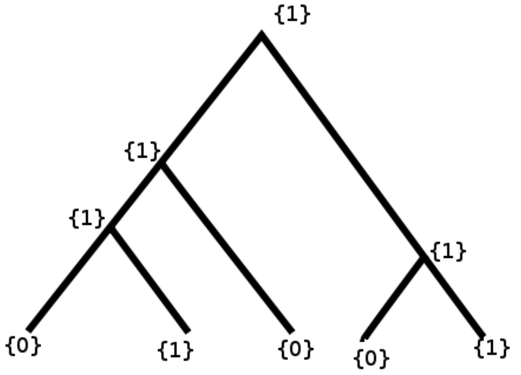


Figure 2.2: A tree labelled with Dollo parsimony, given the labels at the leaves

The example given is the same tree and leaf characters used to illustrate Fitch parsimony. However, since Dollo parsimony does not allow more than one gain, we are forced to have three losses instead of just two gains. Also note that all the nodes on a path between two nodes with the character 1 also have the character 1.

Algorithm for Dollo Parsimony

The following algorithm for Dollo parsimony takes advantage of the fact that the character is present at every node on a path between two nodes that already have the character.

```
Input: Tree $\mathcal{T}$, binary characters $C_L$ at each leaf $L$ in the tree.
Output: Binary characters $C_X$ at each internal node $X$ of the tree, such that the number of
        $0 \rightarrow 1$ transitions is at most $1$, and the number of $1 \rightarrow 0$ transitions is minimized.
foreach Leaf $X$ in $\mathcal{T}$ do
    if $C_X == \{1\}$ then
        foreach Leaf $Y$ in $\mathcal{T}$, $Y \neq X$ do
            if $C_Y == \{1\}$ then
                foreach Node $N$ on the path from $X$ to $Y$ do
                    $C_N = \{1\}$;
                end
            end
        end
    end
end
foreach Node $K$ in $\mathcal{T}$ do
    if $C_K \neq \{1\}$ then
        $C_K = \{0\}$;
    end
end
```

2.2 Probabilistic Approaches- Bayesian networks on Trees

Probabilistic inference techniques aim to compute the probability of existence of an ancestral character. The key idea is that characters in evolutionary biology evolve along the branches of a tree, and each character evolves from its parent through speciations, duplications and losses.

Along a branch, the character at the parent node will affect the character at the other end of the edge. Evolution over the edge can be modelled by a stochastic transition matrix, by associating each end of the edge with a random variable. For example, on an edge $X \rightarrow Y$, we can associate X and Y to a random variable, where the character at X can take values x_0, x_1, x_2 and the character at Y can take the values y_0 and y_1. The transition matrix over the edge, which describes a *conditional probability distribution* function, is

$$\begin{array}{c c} & \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{array}{c} y_0 \\ y_1 \end{array} & \begin{pmatrix} s_{x_0 \rightarrow y_0} & s_{x_1 \rightarrow y_0} & s_{x_2 \rightarrow y_0} \\ s_{x_0 \rightarrow y_1} & s_{x_1 \rightarrow y_1} & s_{x_2 \rightarrow y_1} \end{pmatrix} \end{array},$$

where each $s_{x_i \rightarrow y_j}$, which we may also call $s_{x_i y_j}$ for convenience, represents the probability $Pr\left(Y = y_j | X = x_i\right)$ (See Appendix B for a short introduction to notation used in probability theory). It is immediately apparent that the columns sum to 1, i.e. $\sum_j s_{x_i \rightarrow y_j} = 1$. At the root, we have a *prior probability table* $[p_1 p_2 \ldots p_t]$ instead of the matrix, which gives us the probability of each state of the root. This table meets the condition that $\sum_{i=1}^{t} p_i = 1$.

The object we now have is a directed, rooted tree, with each node associated with a random variable, a conditional probability distribution over each edge, and a prior probability distribution at the root. This is a *probabilistic graphical model*. More precisely, the model we obtain is a *Bayesian network* on a rooted, directed tree [30].

Probabilistic graphical models, which include hidden Markov models and Markov chains, have been widely studied, and applied to problems in machine learning, social networks etc. In the field of computational biology, these models are used for sequence alignment, inferring ancestral population structures etc [1]. Algorithms to apply on these models have also been well developed, and make them very attractive to use.

The directed edges on the underlying graph of a Bayesian network represent a causal relationship between the two events associated with the nodes. For example, an edge from node X to node Y means that the outcome of event X directly influences the outcome of event Y. Also, for a path $X \rightarrow Y \rightarrow Z$, if there is no other path from X to Z and the outcome of Y is fixed, then the outcome of event X does not influence the outcome of event Z. Since we shall be working with directed rooted tree models, there is at most one unique path between any two nodes, and each node (except the root) has a unique parent.

2.2.1 Joint and prior probability distributions

The *joint probability distribution* of the Bayesian network describes the probability of all random variables in the network being assigned a specific value. Formally,

Definition 7. [30] *Let a set of n discrete random variables $V = \{X_1, X_2, \ldots, X_n\}$ be specified such that each X_i has a countably infinite space. A function, that assigns a real number $Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$ to every n-tuple $(x_1, x_2, \ldots, x_n)$, such that x_i is chosen from the space of X_i, is called a joint probability distribution of the random variables V if it satisfies the following conditions.*

(i) For every n-tuple $(x_1, x_2, \ldots, x_n)$,

$$0 \leq Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \leq 1.$$

(ii) If we sum over all possible n-tuples $(x_1, x_2, \ldots, x_n)$,

$$\sum_{(x_1, x_2, \ldots, x_n)} Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = 1.$$

For a general case, when we do not have a Bayesian network, the joint probability $Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$ can be written as follows,

$$Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} Pr\,(X_i = x_i | X_{i+1} = x_{i+1}, \ldots, X_n = x_n).$$

For a Bayesian network, however, since the outcome of each event is directly dependent only on its parent,

$$Pr\,(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} Pr\,(X_i = x_i | X_p = x_p, \text{ where } X_p \text{ is the parent of } X_i).$$

Using the conditional probability distribution matrices given along each edge, we can find the joint probability distribution of all the random variables V in the Bayesian network.

Having completely defined a Bayesian network, we can proceed to extract information from it. In the absence of any evidence, i.e. when every random variable can take any possible value, we can create a prior probability distribution over the network. This can be done by iterating the following steps.

1. For an edge $X \rightarrow Y$, find $Pr\left(X = x\right)$ for every value x that the random variable X can take.

2. Set $Pr\left(Y = y\right)$ for every value y that Y can take to be

   $\sum_{\forall\ values\ of\ x} Pr\left(Y = y|X = x\right).Pr\left(X = x\right)$, where the conditional probability $Pr\left(Y = y|X = x\right)$ can be looked up from the probability matrix on the edge.

3. Repeat for every child of Y.

This gives us the probability of every value at each node when we do not know the state of any random variable in the system. This is known as the *prior probability distribution* of the network. The next section deals with the case when we do have some information about the state of the system.

2.2.2 Inference of marginal probabilities

An *evidence* is an assignment x to a random variable X. It corresponds to saying that $Pr\left(X = x\right) = 1$ and $Pr\left(X \neq x\right) = 0$. An *evidence set* $\mathbf{Y}$ is a set of random variables which have been assigned some evidence. We will use the notation e for a $|\mathbf{Y}| -$ tuple that represents the assignment of each random variable in $\mathbf{Y}$, and will denote this by $\mathbf{Y} = e$. The entire set of possible assignments e that can be given to $\mathbf{Y}$ will be denoted by $\mathcal{E}$. This set is called the *state space* of the model.

Definition 8. *The* marginal posterior probability *of a node X being in state x, given an evidence set $\mathbf{Y}$ and evidence e, is the probability that we observe the random variable X to be in state x, conditioned on the evidence, or $Pr\left(X = x|\mathbf{Y} = e\right)$.*

Given a Bayesian network on a tree $\mathcal{T}$ on the set of nodes V, with evidence set $\mathbf{Y}$, and evidence e, it is possible to find the marginal posterior probability of a random variable/node $X \notin \mathbf{Y}$ being in state x. This is equivalent to letting the other variables (i.e. $V \backslash \mathbf{Y} \cup \{X\}$) attain any value, which can be done by summing over all other cases, and restricting the values of X and the variables in $\mathbf{Y}$. To understand this, let us introduce some notation. In this section, unless stated, the random variable of interest at node X will also be called X, and the evidence set will be $\mathbf{Y}$.

We define a *labelling* of the nodes of the graph as an assignment of all random variables. Thus, labellings are $|V|-$tuples that represent the outcomes for all possible events. The set of all possible labellings will be denoted by $\mathcal{L}$. For any labelling $l \in \mathcal{L}$, the label of a single node Q, i.e. the value of the random variable associated to Q according to that labelling, will be denoted by l_Q .

An e-*consistent labelling* is a labelling l in the set $\mathcal{L} : e$, which denotes the set of all labellings in $\mathcal{L}$ such that the evidence nodes in $\mathbf{Y}$ have been labelled with $e \in \mathcal{E}$. We can extend this definition to an $e \cup \{x\}$ consistent labelling in the set $\mathcal{L} : e \cup \{x\}$, the subset of labellings in $\mathcal{L} : e$ such that the node X is labelled x in all elements of that set. We define the probability of a labelling $l \in \mathcal{L}$ as follows

$$Pr\,(l) = p_{l_{root}} \prod_{P \to Q \in E} s_{l_P l_Q}, \tag{2.1}$$

where $s_{l_I l_J}$ denotes the transition probability of going from the label l_I of the node I, to the label l_J of its child J, and p_{lroot} is the probability that the labelling of the root is l_{root}. We can then define the marginal probability of $X = x$ as

$$Pr\,(X = x|\mathbf{Y} = e) = \frac{1}{\sum_{l \in \mathcal{L}:e} Pr\,(l)} \sum_{l \in \mathcal{L}:e \cup \{x\}} Pr\,(l).$$

Thus, the marginal probability of $X = x$ is simply the probability of observing x at X conditional on the evidence. We sum over all possible states of the random variables except for X and for the evidence nodes.

Example

In the following example, the edges are oriented away from the root. Each random variable is binary, the conditional probability distribution matrix, T, is assumed to be the same over each edge, and the prior probability at the root is taken to be 0.5. The evidence nodes in $\mathbf{Y}$ are (A, B), and the evidence is $(1, 0)$. The probability of node D being in state 1 and the evidence, is

$$Pr\,(D = 1, (A, B) = (1, 0)) = \sum_{R=\{0,1\}} \sum_{C=\{0,1\}} Pr\,(D = 1, A = 1, B = 0, R, C).$$

On normalizing this with the following term, we get the marginal probability of D being in state 1 conditioned on the evidence.

$$Pr\,(A = 1, B = 0) = \sum_{D=\{0,1\}} \sum_{R=\{0,1\}} \sum_{C=\{0,1\}} Pr\,(D = 1, A = 1, B = 0, R, C).$$

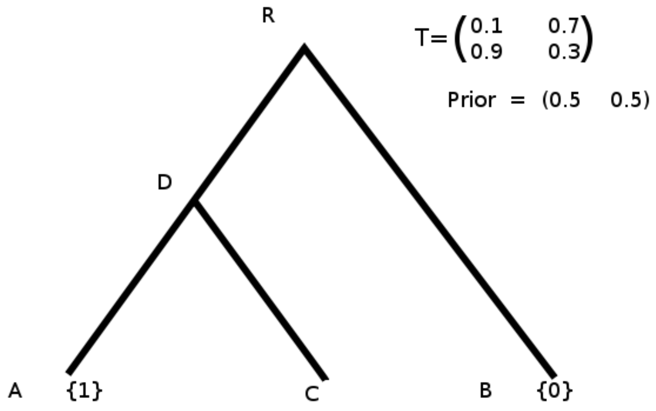


Figure 2.3: A small Bayesian network, with the conditional probability matrix over each node, prior probability at the root, and evidence.

Thus the marginal probability of D being in state 1 conditioned on the evidence is

$$Pr\left(D = 1 |\left(A, B\right) = \left(1, 0\right)\right) = \frac{1}{6}$$

The normalization term is called the *marginal probability of the evidence* $Pr\left(\mathbf{Y} = e\right)$. Formally, the marginal probability of the evidence for any evidence $\mathbf{Y} = e$ is given by

$$Pr\left(\mathbf{Y} = e\right) = \sum_{x \in \text{Sample space of } X} Pr\left(X = x, \mathbf{Y} = e\right) = \sum_{l \in \mathcal{L}:e} Pr\left(l\right). \tag{2.2}$$

This is a constant for an evidence e, irrespective of the node X we are summing over.

Since all the steps, excluding the normalization, consist of only sums and products, we can infer the marginal probability $Pr\left(X = x, \mathbf{Y} = e\right)$ through repeated sums and products. By using the fact that an initialized node induces conditional independence of nodes connected through through it, to consider the probability of $X = x$, we can take the product of the probability of the tree rooted at X, when $X = x$, and the probability of the rest of the graph, both conditional on $\mathbf{Y} = e$. Each of these can be recursively calculated. The stopping condition for the recursion is specified by the evidence, which fixes the probability at the nodes in the evidence set. Thus, we get a *sum-product algorithm* to infer marginal probabilities. One variant of this algorithm, proposed by Pearl [34], is the belief-propagation algorithm, which passes the output of the sums and products as information to be used for the next level of recursion.

Sum-product algorithms are generalizations of many widely used algorithms for probabilistic graphical models, such as the forward algorithm for Hidden Markov models.

2.2.3 Inference of maximum a posteriori labelling

We defined the marginal probability of an evidence e as $\sum_{l\in\mathcal{L}:e} Pr\left(l\right)$. Since the probability of a labelling l is $Pr\left(l\right) = \prod_{P\to Q\in E} s_{l_P l_Q}$, each summand in $Pr\left(\mathbf{Y} = e\right)$ corresponds to an explanation of the evidence. We get a unique marginal probability of evidence if and only if we have no directed path from one evidence node to another. Otherwise, due to the Markov property, the probability of the evidence of those two nodes will be independent of each other.

One optimization question that we could ask is which labelling maximizes the probability of seeing the evidence. This question is almost equivalent to finding the most probable evidence consistent labelling. The second question is answered by the largest summand in $\sum_{l\in\mathcal{L}:e} Pr\left(l\right)$. This labelling is not necessarily unique.

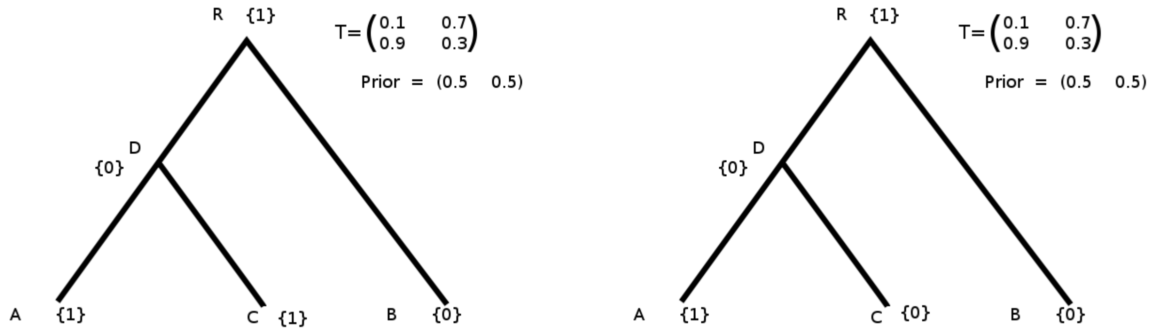
Definition 9. *The* maximum a posteriori probability labelling *of an evidence e ($MAP\left(e\right)$) is a labelling of the nodes of the Bayesian network which maximizes the probability of seeing the evidence.*

$$MAP\left(e\right) = \underset{l\in\mathcal{L}:e}{\operatorname{argmax}} \{Pr\left(l\right)\}. \tag{2.3}$$

There can be more than one internal labelling which gives us the same maximum a posteriori probability for a given evidence. This is can be seen from the fact that $Pr\left(l\right)$ is simply the product of the transition probabilities along each edge, and if we change the order of these transitions, we will still get the same probability.

There may be also be more labellings that maximize the probability of the evidence than evidence consistent labellings of maximum probability. To illustrate this, let us look at the case of the Bayesian network in Figure 2.3. The labelling of the internal nodes which maximizes the probability of the evidence is $R = 1$ and $D = 0$. Notice that C is not an evidence node, nor is it an internal node.

Since both labellings give us a maximum probability of evidence, which equals 0.2205, and since this probability is independent of the label at C, we can effectively prune the tree at C, and look at the rest of the tree. At the same time, the probabilities of the labellings are different. When C is



(a) C labelled with 1

(b) C labelled with 0

Figure 2.4: Example of non-uniqueness of optimal labellings

labelled 1, the probability of the labelling is 0.19845, while the probability of the labelling with C labelled 0 is 0.02205.

One way to resolve this ambiguity is to take the most probable labelling instead of looking for labellings that maximize the probability of the evidence.

The belief-propagation algorithm can be adapted to do solve the MAP problem, by using a max-product formulation or a max-sum formulation on the log-parameter space instead of the sum-product formulation. Then, the probabilities at the root gives us the maximum a posteriori probability of the evidence. By backtracking, we can find the possible labellings that give us the same maximum a posteriori probability. As in the case of the sum-product algorithm, the max-product formulation of belief propagation is a generalization of other algorithms used in probabilistic graphical models, such as the Viterbi algorithm.

2.3 Inference in Ancestral Protein-Protein Interaction Networks

The inference techniques discussed in the previous section can all be used for inference of ancestral protein interactions. To apply them, we work on the interaction tree, which is created from a rooted, binary gene tree which has been reconciled with the species tree. The critical point is that the tree structure removes the dependence of an interaction on its siblings.

2.3.1 Input

The binary information at the leaves, indicating the presence or absence of an interaction in the extant species, is usually computed using sequence alignment techniques. A cut-off score is decided using statistical data representing the strength of interactions as a function of the score. If the score of an interaction is greater than the cut-off, then the interaction is assumed to be present (binary label 1), else it is assumed to be absent (binary label 0). Such techniques can also be used to reconstruct ancestral protein sequences, and estimate the strength of an ancestral interaction [19, 35].

2.3.2 Parsimony on the interaction tree

Parsimonious techniques can be directly applied to the interaction tree using the evidence. It is common to use non-parametric versions of parsimony, such as Fitch or Dollo parsimony, for inference. Other non-parametric variants have been used for comparison against probabilistic models and inference through sequencing data [29, 35].

2.3.3 Bayesian inference

A probabilistic approach to the inference of ancestral protein-protein interactions is desirable since we have to infer data that we can not compare to what actually happened. Thus, a probabilistic inference technique gives us an estimate of whether an interaction was present or absent, instead of outright postulating its existence, as in parsimony.

The work of Pinney et al. on bZIP transcription factors [35] is based on a well studied family of proteins. These are proteins that bind to specific DNA sequences and control the transcription process from DNA to messenger RNA.

Dutkowski and Tiuryn [14, 15] worked on protein-protein interactions in many families of proteins, and differentiated between duplicating and speciating nodes.

The graph

The probabilistic inference technique is centred around the fact that the interaction tree we constructed in Chapter 1 forms the underlying graph of a Bayesian network. Each node represents an interaction, and we associate it with a binary random variable. Thus, an interaction is present if the random variable is 1, and it is absent if it is 0.

Since we assumed during construction that the interactions in a given species are independent of each other, we have no causal relations between them, and get an underlying cycle free undirected graph. Since there is an identified root, and a natural direction of evolution from the root, we can assign a direction to each edge, pointing away from the root. This enforces the property that interactions within a single species evolve independently of each other, which is critical to the construction of a Bayesian network. The paper by Dutkowski and Tiuryn [15] does not explicitly construct an interaction tree, but their model can be interpreted as one.

Parameter selection

A major obstacle in computational biology in general is to estimate parameters to fit a model. This is especially true for evolutionary models, since we have no data to infer from. So, we have to rely on experimental data that we often hope is back compatible with the true evolutionary scenario.

In the case of protein-protein interactions, given an interaction tree, we can fit the following parameters to our model.

(i) We can estimate the gain and loss probabilities of an interaction over each edge. So, the number of parameters in our model is twice the number of edges.

(ii) We can distinguish duplicating and speciating nodes, and fit 2 parameters to each of them. The number of parameters in this case is 4.

(iii) We can treat all edges as identical, and fit 2 parameters to the whole model.

Pinney et al used sequencing data based on the paper by Fong, Keating and Singh [19] to fit parameters to all edges in their interaction tree. Experimental scores were calculated for the strength of human protein-protein interactions, and the probabilities of gain and loss fitted to their model were estimated from this by modelling these probabilities as logistic functions of sequence divergence on the gene tree. The interactions predicted by the scores were also used as a basis to compare their probabilistic techniques. The parameters used by them are given in Appendix A.

The paper by Dutkowski and Tiuryn used data based on the paper of Sole et al. [40], which considers a specified model of evolution, and estimates parameters based on that model instead of using direct empirical data.

In the absence of reliable empirical data, the probabilistic inference techniques available to us cannot be used. It is, therefore, desirable to have some method to get an overall, parametric view of the Bayesian network, using which we can make an informed choice about the parameters to use for the model. This leads us to the field of algebraic statistics.

Part II

Algebraic Statistics

A major issue in evolutionary biology is the inference of parameters for evolutionary models. These parameters may be the cost matrices for a discrete algorithm, or the transition matrices or probability distributions for Markov models and Bayesian networks. Parameter estimation is often done through empirical methods, such as sequence analysis.

The goal of this part is to introduce the field of algebraic statistics and related terminology. Viewing statistical models as algebraic objects allows us to examine the parameter space of these models. In particular, a translation of the models to *tropical geometry* provides nice geometric interpretations of the MAP problem.

Chapter 3 discusses the algebraic interpretation of statistical models. It lays emphasis on toric models, such as the one we deal with. It also lays the foundation behind algebraic statistics, and discusses some basic algebraic concepts that we will need.

Chapter 4 introduces tropical geometry, and provides a relation between it and polytopes. In particular, this chapter is intended to provide a natural transition from classical arithmetic to tropical arithmetic for polynomials. The Newton polytopes of polynomials are established to be objects that can be interpreted as generalizations of the tropical semiring in one dimension. It is also made clear that the Newton polytope of a given polynomial can be constructed using Minkowski sum and convex hull operations on the Newton polytopes of the factors of the polynomial.

Chapter 5 uses tropical geometry to answer a parametric MAP problem on statistical models. It establishes our problem and the approach we use in the experiments section. Bounds for the size of the polytopes constructed are provided in this chapter, and the translation of the sum-product algorithm to polytope algebra is made clear.

Chapter 3

Statistical models as Algebraic objects

Parameter estimation in evolutionary biology is an important and generally hard problem. A novel way to approach it is to compute algebraic varieties that define statistical models of evolution. These allow us to obtain a parameter independent representation of these models.

3.1 Polynomial maps of statistical models

Formally, a statistical model is a family of probability distributions on a set of possible observed outcomes, called a *state space*. For our purposes, we shall only consider finite state spaces. Following the convention of Chapter 2, we shall call our state space of observations $\mathcal{E}$, and the cardinality of the state space will be denoted by m. An element of this space $e \in \mathcal{E}$ will be called an *evidence configuration*.

Definition 10. *A* probability distribution *on the state space $\mathcal{E}$ is a point $(p_1, p_2, \ldots, p_m)$ in the probability simplex in $m-1$ dimensions, Δ_{m-1}*

$$\Delta_{m-1} = \left\{ (p_1, p_2, \ldots, p_m) : \sum_{i=1}^{m} p_i = 1, 0 \leq p_i \leq 1 \; \forall \; i \right\}.$$

The element p_i in a probability distribution in the simplex denotes the probability of the i^{th} outcome in the state space $\mathcal{E}$.

Recall that we defined the marginal probability for the evidence in a tree-like Bayesian network $G = (V, E)$, with evidence set $\mathbf{Y} \subset V$ and a parameter matrix S to be the given by the following expression

$$Pr\left(\mathbf{Y} = e \in \mathcal{E}\right) = \sum_{l \in \mathcal{L}: e} \prod_{uv \in E(\mathbf{Y})} s_{l_u \to l_v}, \tag{3.1}$$

where $E\left(\mathbf{Y}\right)$ denotes the set of all edges in E which belong to a directed path from the root to one of the evidence nodes in $\mathbf{Y}$, and each distinct $s_{i \to j}$ is an entry in S. For convenience, we shall refer to $s_{i \to j}$ by s_{ij}.

Now, assuming that we do not have preset parameters s_{ij}, we can treat these as formal variables x_{ij}, and obtain a polynomial in these variables. Let us represent such a polynomial by f_e. Thus,

$$f_e = \sum_{l \in \mathcal{L}: e} \prod_{uv \in E(\mathbf{Y})} x_{l_u l_v}. \tag{3.2}$$

If each node in $\mathbf{Y}$ can take one of c values, then we can say that the total number of possible evidence configurations $e \in \mathcal{E}$, i.e. the cardinality of $\mathcal{E}$, which we called m, is $c^{|\mathbf{Y}|}$. Thus, we can define at least m polynomials f_e. Formally, and more generally for all statistical models, we can state the following:

For a statistical model defined on d parameters, and state space $\mathcal{E}$, with cardinality m, we can define a positive polynomial map $\mathbf{f} : \mathbb{R}^d \mapsto \mathbb{R}^m$.

Since each f_e corresponds to the probability of a possible evidence, we also have the property that $\sum_{e \in \mathcal{E}} f_e = 1$ and the condition that $f_e > 0$. The structure of the statistical model may impose other conditions on the polynomial map. The function $\mathbf{f}$ defines an *algebraic statistical model* with d-parameters. This definition holds even if we do not have an underlying graphical model.

Toric models

Consider the probabilistic tree model $G = (V, E)$ with 4 parameters $s_{00}, s_{01}, s_{10}, s_{11}$, and a prior of 0.5 at the root. An explanation l corresponds to a fixed labelling of the underlying tree . Its probability, as stated in Chapter 2, is given by

$$Pr\left(l\right) = 0.5 \prod_{P \to Q \in E} s_{l_P l_Q}.$$

On multiplying this out, we get a monomial $0.5s_{00}^{\theta_1}s_{01}^{\theta_2}s_{10}^{\theta_3}s_{11}^{\theta_4}$. Taking the logarithm, we get the following log-probability,

$$\ln\left(Pr\left(l\right)\right) = \ln\left(0.5\right) + \theta_1 \ln\left(s_{00}\right) + \theta_2 \ln\left(s_{01}\right) + \theta_3 \ln\left(s_{10}\right) + \theta_4 \ln\left(s_{11}\right).$$

This is a linear function in the logarithm of the model parameters. Many graphical probabilistic models have polynomial maps that are exhibit this property.

Definition 11. *Algebraic models in which the logarithm of the probability of a single explanation of an evidence $e \in \mathcal{E}$ can be expressed as a linear function of the model parameters are called* toric models.

Since this is a linear function in the $\log$-space, toric models are also called *log-linear* models.

Our interest in these models arises from the fact that they describe a wide range of graphical probabilistic models, including acyclic Bayesian networks. To discuss the algebraic properties of these models, we shall first discuss some basic algebraic concepts.

In order to work with a more general class of polynomials, we shall assume that each polynomial f_e belongs to the polynomial ring $\mathbb{Q}\left[x_1, x_2, \ldots, x_d\right]$, where the variables can take values from the field of complex numbers. Thus, the map we shall study is $\mathbf{f} : \mathbb{C}^d \mapsto \mathbb{C}^m$. This lets us discuss the algebraic interpretation of a statistical model without going into methods involving real algebraic geometry.

3.2 Ideals and Varieties

Let $\mathbb{Q}\left[\mathbf{x}\right] = \mathbb{Q}\left[x_1, x_2, \ldots, x_m\right]$ be the polynomial ring with coefficients in the rational numbers, and over m variables, $x_1, x_2, \ldots, x_m \in \mathbb{C}$. Since this ring also behaves like an $\mathbb{Q}$-vector space, we can define a *distinguished* $\mathbb{Q}$*-linear basis* of this ring as the set of monomials

$$\left\{x_1^{\theta_1} x_2^{\theta_2} \ldots x_m^{\theta_m} : \theta_1, \theta_2, \ldots, \theta_m \in \mathbb{N}\right\}.$$

3.2.1 Variety

For every polynomial $f \in \mathbb{Q}\left[\mathbf{x}\right]$, we can define a zero set $V\left(f\right)$

$$V\left(f\right) = \left\{\mathbf{z} = \left(z_1, z_2, \ldots, z_m\right) \in \mathbb{C}^m : f\left(\mathbf{z}\right) = 0\right\}.$$

$V\left(f\right)$ is a hypersurface in $\mathbb{C}^m$. For a subset S of $\mathbb{C}^m$, we define $V_S\left(f\right) = V\left(f\right) \cap S$ as the set of points in S that belong to the zero set.

Assume $\mathcal{F} \subset \mathbb{Q}\left[\mathbf{x}\right]$ is a subset of the polynomial ring. Then, we can define an intersection of hypersurfaces in $\mathbb{C}^m$

$$V\left(\mathcal{F}\right) = \left\{\mathbf{z} = \left(z_1, z_2, \ldots, z_m\right) \ : \ f\left(\mathbf{z}\right) = 0 \ \forall \ f \in \mathcal{F}\right\}.$$

Alternately,

$$V\left(\mathcal{F}\right) = \left\{\bigcap_{f \in \mathcal{F}} V\left(f\right)\right\}.$$

This set is called the *variety* of the set $\mathcal{F}$ over the set $\mathbb{C}^m$ [10]. We can define a restricted variety V_S as a subset of the variety such that all elements lie in the set $S \subseteq \mathbb{C}^m$. When $m = 1$, the variety is simply the set of all zeros of a polynomial in one variable.

3.2.2 Ideal

For a subset $\mathcal{F} \subset \mathbb{Q}\left[\mathbf{x}\right]$, the ideal generated by $\mathcal{F}$, denoted by $\langle\mathcal{F}\rangle$ is defined as follows [10]

$$\langle\mathcal{F}\rangle = \left\{\sum_{f_i \in \mathcal{F}} h_i f_i \ : \ \forall \ h_i \in \mathbb{Q}\left[\mathbf{x}\right]\right\}.$$

Ideals are not unique to the set, i.e. it is possible for $\mathcal{F}, \mathcal{F}' \in \mathbb{Q}\left[\mathbf{x}\right] \ \ \mathcal{F} \neq \mathcal{F}'$ to exist such that $\langle\mathcal{F}\rangle = \langle\mathcal{F}'\rangle$. If so, then we have the following relation between the varieties of the two sets

$$V\left(\mathcal{F}\right) = V\left(\mathcal{F}'\right).$$

A major result in algebraic geometry is Hilbert's basis theorem.

Theorem 1. ***(Hilbert's basis theorem)*** *Every infinite set $\mathcal{F}$ of polynomials in a ring $\mathbb{Q}\left[\mathbf{x}\right]$ contains a finite subset $\mathcal{F}'$ such that $\langle\mathcal{F}\rangle = \langle\mathcal{F}'\rangle$.*

This implies that every variety can be represented as the intersection of finitely many hypersurfaces.

An ideal I is called a *prime ideal* if, for two polynomials $g, h \in \mathbb{Q}\left[\mathbf{x}\right]$ such that $f = g \cdot h \ \in I$, then either $g \in I$ or $h \in I$. This generalizes the concept of prime numbers to polynomials.

3.3 Algebraic interpretation

Having defined the polynomial map $\mathbf{f} : \mathbb{C}^d \mapsto \mathbb{C}^m, m = |\mathcal{E}|$, of a statistical model, the image of the map $\mathbf{f}$ is the following set

$$\mathbf{f}\left(\mathbb{C}^d\right) = \{(p_1, p_2, \ldots, p_m) \in \mathbb{C}^m : \text{Set of conditions on } p_i\text{'s defined by the statistical model}\}.$$

This set can be interpreted as a Boolean combination of algebraic varieties, i.e. composed of unions, intersections and exclusions. If we take the topological closure of the set, we get another algebraic variety.

For example, let us look at the polynomial map $\mathbf{f} : \mathbb{C}^2 \mapsto \mathbb{C}^3, (x_1, x_2) \mapsto \left(x_1^2, x_1 \cdot x_2, x_1 \cdot x_2\right)$. The image of this map is the following set,

$$\mathbf{f}\left(\mathbb{C}^2\right) = \left\{(p_1, p_2, p_3) \in \mathbb{C}^3 : p_2 = p_3 \text{ and } p_2 = 0 \text{ if } p_1 = 0\right\}.$$

In terms of varieties,

$$\mathbf{f}\left(\mathbb{C}^2\right) = \left(V\left(p_2 - p_3\right) \backslash V\left(p_1, p_2 - p_3\right)\right) \cup V\left(p_1, p_2, p_3\right),$$

which is not an algebraic variety. Geometrically, this is the plane $p_2 - p_3 = 0$, excluding its projection on $p_1 = 0$, but including the origin. However, the closure $\overline{\mathbf{f}\left(\mathbb{C}^2\right)} = V\left(p_2 - p_3\right)$, which contains the limit points of $\mathbf{f}\left(\mathbb{C}^2\right)$, which satisfy $p_2 \neq 0, p_1 = 0$, is an algebraic variety.

This result, which holds over the complex numbers, but not over the reals, can be stated as follows.

Theorem 2. [33] *The image of a polynomial map $\mathbf{f} : \mathbb{C}^d \mapsto \mathbb{C}^m$ is a Boolean combination of algebraic varieties in $\mathbb{C}^m$. The topological closure $\overline{\mathbf{f}\left(\mathbb{C}^d\right)}$ of the image $\mathbf{f}\left(\mathbb{C}^d\right)$ in $\mathbb{C}^m$ is an algebraic variety.*

The elements of this variety correspond to points that satisfy the conditions imposed by the model. The real elements of this variety are those that lie in the probability simplex $\sum_{i=1}^m p_i = 1$. Methods to find these elements lie in the domain of *real algebraic geometry* [4]. It is also common to disregard points that lie in the closure, but not in the image of the map $\mathbf{f}$, to simplify arguments [33].

If we consider all polynomials in $\mathbb{Q}\left[p_1, p_2, \ldots, p_m\right]$ that vanish on the image of $\mathbf{f}$, we can compactly represent them by an ideal $I_{\mathbf{f}}$ in $\mathbb{Q}\left[p_1, p_2, \ldots, p_m\right]$. Thus, a point in the simplex will always send the polynomials in this ideal to zero. Furthermore, this happens to be a prime ideal which represents the closure $\overline{\mathbf{f}\left(\mathbb{C}^d\right)}$ by definition. The members of $I_{\mathbf{f}}$ are called *model invariants*.

The problem of finding the probability distributions in the simplex that satisfy a given statistical model is well defined and translates to finding a finite set of generators, $\mathcal{F}$, which generate the ideal $I_{\mathbf{f}}$. These generators are independent of the model parameters, being defined only in terms of polynomials in the ring $\mathbb{Q}\left[p_1, p_2, \ldots, p_m\right]$. These generators will completely describe a parameter independent version of the statistical model, i.e. they will be be the same set of conditions imposed by the statistical model on the polynomial map, represented in terms of polynomials in $p_1, p_2, \ldots, p_m$.

The main problem is to find these generators, and in particular, it is desirable to get a *Gröbner basis* of the ideal. This is a set of generators such constructed that the leading terms of the polynomials in $I_{\mathbf{f}}$, according to some term ordering, are generated by the leading terms of the polynomials in the generating set. However, this is usually hard when the number of parameters and m are large.

Since we can describe the statistical model as a polynomial map, we can also look at these polynomials in the $\min$-plus algebra, taking parameters in the $\log$-space. While the map in classical algebra provides us with solutions to the marginal probability problem, the $\min$-plus algebra, as we stated before, is used to solve the maximum a posterori probability problem. To discuss the algebraic interpretation of the MAP problem, the next chapter introduces the concept of tropical geometry.

Chapter 4

Tropical Geometry

The maximum a posteriori problem for statistical models is a case of moving the marginal probability problem to a *tropical* setting. By this, it means we replace the classical algebra $(\mathbb{R}, +.\times)$ by the tropical semiring $(\mathbb{R}, \min, +)$. This algebra has a well defined geometric interpretation, and this property can be exploited to solve parametric inference problems. This chapter introduces some concepts in tropical geometry and about polytopes.

4.1 The tropical semiring

The object we shall be working with is the tropical semiring [36]. It is defined as follows.

Definition 12. *The tropical semiring over a totally ordered field* $\mathbb{K}$*,* $(\mathbb{K} \cup \{\infty\}, \oplus, \odot)$ *is defined by the following operations*

$$x \oplus y = \min\{x, y\} \qquad \textit{and} \qquad x \odot y = x + y \tag{4.1}$$

$\forall\, x, y \in \mathbb{K}$.

Since we need a total order on the elements of the field, we generally work over the field of reals. The operation $\oplus$ is called the tropical sum, while the operation $\odot$ is called the tropical product. Both operations are commutative.

$$x \oplus y = y \oplus x \qquad \text{and} \qquad x \odot y = y \odot x.$$

The tropical product is distributive over the tropical sum.

$$z \odot (x \oplus y) = (z \odot y) \oplus (z \odot x) .$$

Each operation has an identity element, or a neutral element.

$$x \oplus \infty = x \qquad \text{and} \qquad x \odot 0 = x.$$

We can, define a polynomial over the tropical semiring. Let $x_1, x_2, \ldots, x_d$ be elements in the tropical semiring. A *tropical monomial* is a finite tropical product of these elements, with repetition allowed. For example

$$x_1 \odot x_1 \odot x_2 \odot x_3 = x_1^2 x_2 x_3.$$

In terms of classical arithmetic, this translates into the following expression

$$x_1 + x_1 + x_2 + x_3 = 2x_1 + x_2 + x_3.$$

This is always a linear function with integer coefficients.

Definition 13. *A tropical polynomial is a finite tropical linear combination of tropical monomials, with coefficients in the real numbers*

$$g(x_1, x_2, \ldots, x_d) = c_1 \odot x_1^{i_{11}} x_2^{i_{12}} \ldots x_d^{i_{1d}} \oplus \ldots \oplus c_l \odot x_1^{i_{l1}} x_2^{i_{l2}} \ldots x_d^{i_{ld}},$$

where $i_{11}, i_{12}, \ldots, i_{l1}, j_{l2}, \ldots$ *are non-negative integers.*

In terms of classical arithmetic, we get a function g that returns the minimum of a finite number of linear functions

$$g(x_1, x_2, \ldots, x_d) = \min(c_1 + i_{11}x_1 + i_{12}x_2 + \ldots + i_{1d}x_d, \ldots, c_l + i_{l1}x_1 + i_{l2}x_2 + \ldots + i_{ld}x_d) .$$

Thus, the function $g : \mathbb{R}^d \mapsto \mathbb{R}$ has the following properties:

(i) It is continuous.

(ii) It is piece-wise linear.

(iii) It is concave.

Based on this, we can define the tropical hypersurface $\mathcal{T}(g)$ of g.

Definition 14. *The* tropical hypersurface $\mathcal{T}(g)$ *of a tropical polynomial* g *is the set of all points* $s \in \mathbb{R}^d$ *at which* g *attains a minimum value at least twice.*

Thus, it is the set of points at which g is non-linear. A point $s \in \mathbb{R}^d$ that lies on the $\mathcal{T}(g)$ exhibits the following property

$$\begin{aligned} & c_p + i_{p1}s_1 + i_{p2}s_2 + \ldots + i_{pd}s_d = c_q + i_{q1}s_1 + i_{q2}s_2 + \ldots + i_{qd}s_d \\ \leq & c_k + i_{k1}s_1 + i_{k2}s_2 + \ldots + i_{kd}s_d. \end{aligned}$$

where $i_p, i_q, i_k \in \mathbb{N}^d$, such that the monomial $c_k \odot x_1^{i_{k1}} x_2^{i_{k2}} \ldots \odot x_d^{i_{kd}}$ (respectively for i_p and i_q) occurs in g, $i_p \neq i_q$, and i_k is not equal to i_p or i_q.

4.2 Polytopes

The geometric representation of tropical hypersurfaces is related to the cones and fans of polytopes. Furthermore, the operations in the tropical semiring have very natural analogous operations when we deal with polytopes. The notation and terminology have been borrowed from the book by Sturmfels [41].

4.2.1 Definitions and notation

Definition 15. *Given* n *points* $v_1, v_2, \ldots, v_n$ *in* $\mathbb{R}^d$*, the convex hull of this set of points is the set*

$$P = \left\{ \sum_{i=1}^{n} \lambda_i v_i \ \in \ \mathbb{R}^d : \lambda_1, \lambda_2, \ldots, \lambda_n \geq 0 \textit{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\}.$$

This set is called a convex polytope. *The* dimension *of the polytope* P*, dim* (P)*, is the dimension of its affine span* $\{\sum_{i=1}^{n} \lambda_i v_i \ : \ \sum_{i=1}^{n} \lambda_i = 1\}$.

In this thesis, we shall always talk about convex polytopes, and so we may use the more general term ‘polytope’ to refer to them.

A polytope can be represented by either a unique set of points whose convex hull yields us the polytope, or by the finite set of closed half-spaces whose intersection includes all the points in

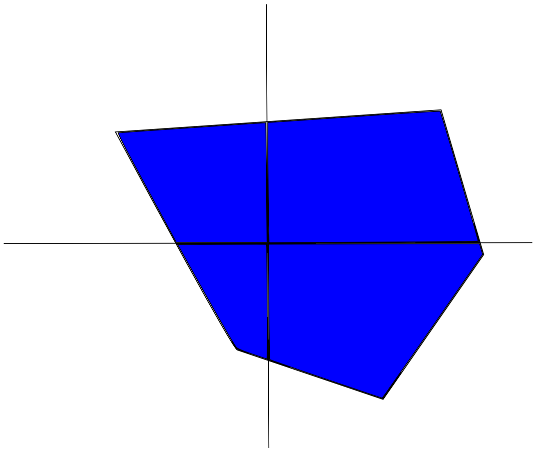


Figure 4.1: A convex polytope

the convex hull. Given an $m \times d$ matrix A, and a column vector $b \in \mathbb{R}^m$, each row of A and the corresponding entry in b will define a half-space in $\mathbb{R}^d$. Thus, we can define an intersection of the half-spaces defined by A and b, which may or may not be bounded, by the equation $P = \left\{x \in \mathbb{R}^d : A.x \geq b\right\}$. A subset of $\mathbb{R}^d$ of this form is called a *convex polyhedron*. The following theorem establishes the alternative definition of convex polytopes.

Theorem 3 (Weyl-Minkowski Theorem)**.** *Convex polytopes are bounded convex polyhedrons.*

A polytope also defines other objects, namely faces, normal cones and a normal fan.

Definition 16. *Given a polytope $P \subset \mathbb{R}^d$, and a vector $w \in \mathbb{R}^d$, we define the* face *of the polytope with respect to w as the set of all points x in P at which the linear functional $x \mapsto x.w$ attains a minimum,*

$$\textit{face}_w(P) = \{x \in P \ : \ x.w \leq y.w \ \forall \ y \in P\}.$$

Since this is a subset of the polytope itself, each face of P is a polytope. If $w = 0$, then we recover P. Thus, every polytope is a face of itself. A face of dimension 0 is called a vertex of the polytope, and a face of dimension 1 is called an edge of the polytope. A face of dimension $\dim(P) - 1$ is called a facet of the polytope P.

Definition 17. *Let F be a face of the polytope P. The* normal cone *of P at F is the following set*

$$N_P(F) = \left\{w \in \mathbb{R}^d \ : \ \textit{face}_w(P) = F\right\}.$$

The normal cone at F contains all linear functionals w that are minimized at every point in F. The dimension of the normal cone, dim $(N_P(F))$ is given by $d - \dim(F)$. Thus, if F is chosen to be a vertex, then the normal cone has dimension d. Cones which are not contained in cones of higher dimension, are called *maximal cones*.

Definition 18. *The union of all cones $N_P(F)$ as F runs over all faces of P is called the* normal fan *of P,*

$$\mathcal{N}(P) = \left\{ \bigcup N_P(F) \ : \ F = \textit{face}_w(P) \ \forall \, w \in \mathbb{R}^d \right\}.$$

Since the union of all cones will cover the whole space, the normal fan $\mathcal{N}(P)$ is a partition of $\mathbb{R}^d$ into maximal cones, which are in bijection with the vertices of P.

4.2.2 Polytope algebra

Let $\mathcal{P}_d$ be the set of all polytopes in $\mathbb{R}^d$. We can define the polytope algebra $(\mathcal{P}_d, \oplus, \odot)$ as the commutative ring with the following operations for any $P, Q \in \mathcal{P}_d$. The sum of two polytopes is defined as the convex hull of the union of the point sets of P and Q,

$$\begin{aligned} P \oplus Q &= \text{conv}\,(P \cup Q) && (4.2) \\ &= \left\{ \lambda p + (1-\lambda)\, q \, \in \mathbb{R}^d \ : \ p \in P, \, q \in Q, \, 0 \le \lambda \le 1 \right\}. && (4.3) \end{aligned}$$

The product of two polytopes is defined as the Minkowski sum of the two polytopes,

$$\begin{aligned} P \odot Q &= P + Q && (4.4) \\ &= \left\{ p + q \in \mathbb{R}^d \ : \ p \in P, \, q \in Q \right\}. && (4.5) \end{aligned}$$

Both operations yield convex polytopes in $\mathbb{R}^d$, so we get a closed algebra. This algebra is commutative in both sum and product, and holds the distributive property of multiplication over addition, i.e. $P \odot (Q \oplus R) = (P \odot Q) \oplus (P \odot R)$ for all $P, Q, R \in \mathcal{P}_d$.

4.2.3 Relation to the tropical semiring

The one-dimensional polytope algebra, $(\mathcal{P}_1, \oplus, \odot)$, is the geometric interpretation of the tropical semiring $(\mathbb{R}, \odot, \oplus)$. A member of $\mathcal{P}_1$ can be represented by $[a, b]\,, a \le b, a, b \in \mathbb{R}$, a segment on the real line. Then, for $[a, b], [c, d] \in \mathcal{P}_1$, we can define the polytope algebra operations of convex hull and Minkowski sum as follows,

$$[a, b] \oplus [c, d] = [\min(a, c), \max(b, d)]$$
$$[a, b] \odot [c, d] = [a + c, b + d].$$

This yields a definition that agrees with the corresponding operations on the tropical semiring. In higher dimensions, polytope algebra simply becomes a generalization of the tropical semiring.

4.2.4 Tropical varieties and polytopes

The concept of varieties in polynomial rings can be defined almost analogously for the tropical semiring. We have already defined the tropical hypersurface $\mathcal{T}(g)$ of a tropical polynomial g in Definition 14.

We first have to define the tropicalization of a polynomial. Let $f = \sum_{i=1}^{m} a_i x_1^{\theta_{i_1}} x_2^{\theta_{i_2}} \dots x_d^{\theta_{i_d}} \in \mathbb{Q}[\mathbf{x}]$, be a classical polynomial with real variables and constant coefficients. Then, we can define the *tropicalization* of f to be the following operation,

$$trop(f) = \bigoplus_{i=1}^{m} l_{a_i} \odot l_{x_1}^{\theta_{i_1}} \odot l_{x_2}^{\theta_{i_2}} \odot \dots \odot l_{x_d}^{\theta_{i_d}},$$

where $l_{a_i}, l_{x_1}, l_{x_2}, \dots, l_{x_d}$ etc., are the tropical semiring analogues to $a_i, x_1, x_2, \dots, x_d$. Thus, we simply replace the products in the original polynomial by sums, and the sums by $\min$. This defines a tropical hypersurface $\mathcal{T}(trop(f))$ for any tropicalized polynomial $trop(f)$.

We can define an ideal I in $\mathbb{Q}[\mathbf{x}]$ as the ideal generated by a set of polynomials $\mathcal{F}$. The ***tropical variety*** $\mathcal{T}(trop(I))$ of the ideal I is defined as follows [42].

$$\mathcal{T}(trop(I)) = \bigcap_{f \in I} \mathcal{T}(trop(f)).$$

Since every ideal can be finitely generated, the tropical variety can be described as the intersection of finitely many tropical hypersurfaces. It is known that the tropical variety of an ideal I in some polynomial ring $\mathbb{Q}[x_1, x_2, \dots, x_d]$ is a polyhedral fan in d-dimensions [36]. This means that the cones of the polytope indicate which tropical polynomial in the ideal attains minimum value. In particular, this definition establishes a connection between tropical polynomials and polytopes, which proves important in the techniques used in algebraic statistics. A more general definition of the tropical variety exists [42], but for our purposes, we shall not require it.

4.3 Newton Polytopes

Let us define the polynomial ring $\mathbb{Q}[\mathbf{x}]$, where $\mathbf{x}$ is the set of variables $x_1, x_2, \ldots, x_d$. Let us also represent a monomial in any polynomial $f \in \mathbb{Q}[\mathbf{x}]$ by $c_i \mathbf{x}^{\theta_i}$, which represents the monomial $c_i x_1^{\theta_{i1}} x_2^{\theta_{i2}} \ldots x_d^{\theta_{id}}$, where c_i is a constant belonging to the field $\mathbb{Q}$. Then, the polynomial f, with m monomials can be represented by

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i \mathbf{x}^{\theta_i}, \tag{4.6}$$

where none of the c_i is zero, and $\theta_i \in \mathbb{N}^d$ for $i = 1, 2, \ldots, m$. Each θ_i is called an exponent vector of $f(\mathbf{x})$.

Definition 19. *The Newton polytope NP (f) of the polynomial $f(\mathbf{x})$ is the convex hull of the exponent vectors of $f(\mathbf{x})$,*

$$\textit{NP}(f) = \textit{conv}\left(\{\theta_i = (\theta_{i1}, \theta_{i2}, \ldots, \theta_{id}), 0 \leq i \leq m\} : f(\mathbf{x}) = \sum_{i=1}^{m} c_i \mathbf{x}^{\theta_i}\right).$$

For example, the Newton polytope of the polynomial over 2 variables, $f(x_1, x_2) = x_1^4 + 19x_2^6 + 2x_1^3x_2^2 - x_1^2x_2^3 + x_1x_2^2$ is given below. Note that the point $(2, 3)$ is hidden within the polytope. It is important to note that there is no way to retrieve the coefficients of the polynomial from the Newton polytope.

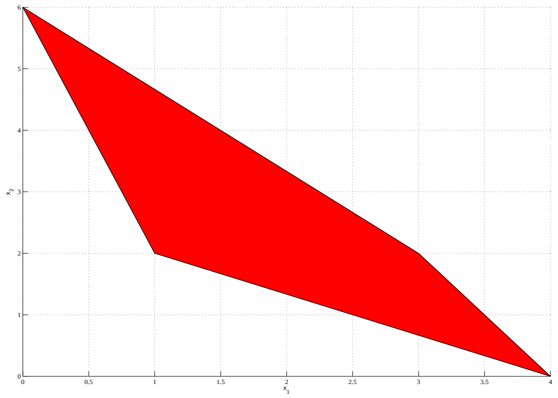


Figure 4.2: Newton polytope

4.3.1 Tropical geometry of Newton Polytopes

Given a polynomial $f = \sum_{i=1}^{m} a_i x_1^{\theta_{i_1}} x_2^{\theta_{i_2}} \ldots x_d^{\theta_{i_d}} \in \mathbb{Q}[\mathbf{x}]$ in d variables, we can ask the question: *which monomial of f attains the maximum value for some value of* $(x_1, x_2, \ldots, x_d)$? Let us consider a set of values $\mathbf{s} = (s_1, s_2, \ldots, s_d)$ for which we are asking this question. Then, the problem becomes finding a monomial g such that,

$$g = \max_i \left\{ a_i s_1^{\theta_{i_1}} s_2^{\theta_{i_2}} \ldots s_d^{\theta_{i_d}} \right\}.$$

We can also formulate this question as follows: find g such that

$$-\ln g = \min_i \{-\ln a_i - \theta_{i_1} \ln s_1 - \theta_{i_2} \ln s_2 - \ldots - \theta_{i_d} \ln s_d\}.$$

We have a one-to-one mapping between $-\ln(x)$ and x, define $-\ln(x) = l_x$, and rewrite this as follows:

$$l_g = \min_i \{l_{a_i} + \theta_{i_1} l_{s_1} + \theta_{i_d} l_{s_d} + \ldots + \theta_{i_d} l_{s_d}\}.$$

The key point is to notice that this can be reformulated as a problem on the tropical semiring $(\mathbb{R}, \oplus, \odot)$,

$$l_g = \bigoplus_i l_{a_i} \odot l_{s_1}^{\theta_{i_1}} \odot l_{s_2}^{\theta_{i_2}} \odot \ldots \odot l_{s_d}^{\theta_{i_d}}. \tag{4.7}$$

At the same time, we can define a Newton polytope, NP(f), whose vertices will be a subset of the set $\left\{\theta_i : x_1^{\theta_{i_1}} x_2^{\theta_{i_1}} \ldots x_d^{\theta_{i_d}} \text{ is a monomial in } f\right\}$. Since vertices are defined as faces of dimension zero, this means that the vertex v minimizes the functional $w \cdot v$ for some vector $w \in \mathbb{R}^d$. This is precisely the tropical monomial for the values w assigned to the variables. Thus, the vertices of the Newton polytope of a polynomial f encode the exponent vectors for which the tropical polynomial is minimized.

Let us call the vertex set V. Then, for $v = (v_1, v_2, \ldots, v_d) \in V$, the normal cone will include those vectors w at which the linear functional $w.v$ is minimized. Taking a point $(w_1, w_2, \ldots, w_d)$ from the cone, the linear functional will become $w_1 v_1 + w_2 v_2 + \ldots + w_d v_d$. Since v is a set of exponents of a monomial of f, this is equivalent to the value of the tropical polynomial for the variables given by w. Thus, the cone of a vertex gives us the set of variables for which the tropical polynomial is equal to the monomial corresponding to the vertex v, assuming that all the coefficients are 1. Furthermore, if we take the union of all the cones, the entire parameter space is covered, and the different cones will provide a partition of the space into regions that yield different optimal solutions. As stated in Section 4.2.4, this is a consequence of the correspondence between tropical varieties and polyhedral fans. Thus, the cones in the normal fan of NP (f) are the regions over which the tropical polynomial $trop\,(f)$ is linear.

4.3.2 Construction of Newton Polytopes from other Newton Polytopes

Suppose we are given a finite set of polynomials $p_1, p_2, \ldots, p_l$. Then, we can find the Newton polytope corresponding to any sum-product combination of these polynomials without calculating the polynomial itself. The main theorem which we use to formulate the process for this is as follows.

Theorem 4. [33] *Let f and g be polynomials in $\mathbb{Q}\left[x_1, x_2, \ldots, x_d\right]$. Then,*

$$\mathit{NP}\,(f \cdot g) = \mathit{NP}\,(f) \odot \mathit{NP}\,(g) \qquad and \qquad \mathit{NP}\,(f+g) \subseteq \mathit{NP}\,(f) \oplus \mathit{NP}\,(g)\,. \tag{4.8}$$

If all the coefficients of f and g are positive, then NP $(f+g) = \mathit{NP}\,(f) \oplus \mathit{NP}\,(g)$.

Proof. Let $f = \sum_{i=1}^{m} c_i \mathbf{x}^{\theta_i}$, and $g = \sum_{i=1}^{n} c'_i \mathbf{x}^{\theta'_i}$. For any vector $w \in \mathbb{R}^d$, define the *initial form* of f, $\text{in}_w\,(f)$ as the subsum of all the monomials $c_i \mathbf{x}^{\theta_i}$, such that $\theta_i \cdot w$ is minimized. By the definition of a face of a polytope, we get the following identity

$$\text{NP}\,(\text{in}_w\,(f)) = \text{face}_w\,(\text{NP}\,(f))\,. \tag{4.9}$$

The initial form of the product of f and g can be obtained by taking the product of the initial forms of f and g individually, as follows

$$\text{in}_w\,(f \cdot g) = \text{in}_w\,(f) \cdot \text{in}_w\,(g)\,. \tag{4.10}$$

Each monomial will be of the form $c_i c'_j \mathbf{x}^{\theta_i + \theta'_j}$, the coefficient being the product of the coefficients of the corresponding monomials from f and g, and the exponent being the sum of the exponents of the same monomials. For any $w \in \mathbb{R}^d$, we will get a single monomial of this form, which minimizes $\left(\theta_i + \theta'_j\right) \cdot w$.

Consider the operator $\text{face}_w\,(\cdot)$. If we apply this operator on the polytope NP $(f) \odot$ NP (g), then we get the set of points $\theta_i + \theta'_j$ which minimize the functional $\left(\theta_i + \theta'_j\right) \cdot w$. We can distribute this product over the sum of vectors. Thus, we get the following identity

$$\text{face}_w\left(\text{NP}\left(f\right)\odot\text{NP}\left(g\right)\right)=\text{face}_w\left(\text{NP}\left(f\right)\right)\odot\text{face}_w\left(\text{NP}\left(g\right)\right). \tag{4.11}$$

These three identities lead to the following result for $w\in\mathbb{R}^d$.

$$\text{NP}\left(\text{in}_w\left(f\cdot g\right)\right)=\text{NP}\left(\text{in}_w\left(f\right)\right)\odot\text{NP}\left(\text{in}_w\left(g\right)\right).$$

But since this holds for all w, we can surmise that both $\text{NP}\left(f\cdot g\right)$ and $\text{NP}\left(f\right)\odot\text{NP}\left(g\right)$ have the same set of vertices, this proving the first identity.

To prove the second identity, we notice that $\text{NP}\left(f\right)\oplus\text{NP}\left(g\right)$ gives us the convex hull of the set $\{\theta_1,\theta_2,\ldots,\theta_m,\theta'_1,\theta'_2,\ldots,\theta'_n\}$. Since every monomial in $f+g$ has its exponent in this set, their convex hull will contain $\text{NP}\left(f+g\right)$, proving the identity. If both f and g consist of only positive coefficients, then there are no cancellations, and we get an equality. □

This theorem allows us to substitute any sequence of operations in the tropical semiring by the corresponding polytope algebra. This also means that as long as we have a polynomial consisting of only positive coefficients, and which can be factored into polynomials of smaller degree, we can construct its Newton polytope from the Newton polytopes of its factors. Furthermore, while each addition-multiplication operation could have caused the number of monomials in the polynomial to grow exponentially, the number of vertices at each step turns out to be polynomial in the number of operations.

We can now use the concepts discussed in this chapter and extend the interpretation of statistical objects as algebraic varieties to tropical arithmetic and polytope algebra. This provides us with a technique to attack the problem of parametric inference in probabilistic graphical models, as we shall see in Chapter 5.

Chapter 5

Tropical Geometry of Probabilistic Graphical Models

In Chapter 3, we discussed the polynomials associated with a graphical model as the marginal probability of the evidence. We now try to interpret each monomial in the probability polynomial, and in doing so, move to an optimization problem.

5.1 Inference functions

For a model $G = (V, E)$, with an evidence set $\mathbf{Y}$ and a parameter set S with d parameters, with none of the parameters determined, we can define a positive polynomial map $f_e : \mathbb{R}^d \mapsto \mathbb{R}$ for each assignment e given to the nodes in $\mathbf{Y}$. This polynomial will correspond to the probability $Pr\left(Y = e\right)$. Since we sum up over all possible internal labellings of the relevant hidden nodes $\mathbf{X}$ that are consistent with the observation e, we can write this as,

$$Pr\left(\mathbf{Y} = e\right) = \sum_{l \backslash e \in \mathcal{L} : e} Pr\left(\mathbf{X} = l \backslash e, \mathbf{Y} = e\right).$$

Each $Pr\left(\mathbf{X} = l, \mathcal{E} = e\right)$ can be written as a monomial in the elements of S, such that the degree of all such monomial is equal. A labelling of the form $l \backslash e : \mathcal{L} : e$ is called an *explanation* of e. Consequently, we can say that each monomial corresponds to an explanation of e. The question we are interested in is which monomial maximizes the probability, $Pr_{max}\left(\mathbf{Y} = e\right)$ of seeing the evidence e. This is the maximum a posteriori probability problem for the graphical model given the evidence e. Naively, the answer to this question will almost be the same as the answer to the question of the most probable labelling consistent with the evidence. However, as stated in Chapter 2, there may be more labellings that maximize the probability of seeing the evidence than the number of labellings of maximum probability that are evidence consistent. In the case when we treat the parameters as formal variables, these are not equivalent. We shall consider the former probabilities as the ones we want to maximize.

Thus, we can formulate the problem as follows

$$Pr_{max}\left(\mathbf{Y}=e\right) = \max_{l\backslash e:\mathcal{L}:e} \{Pr\left(X=l\backslash e, \mathbf{Y}=e\right)\}.$$

Alternatively, we can take the negative logarithms on both sides, and formulate this as a minimization problem. Furthermore, we can write $Pr\left(X=l\backslash e, \mathbf{Y}=e\right)$ as a monomial $s_1^{\theta_{l1}} s_2^{\theta_{l2}} \ldots s_d^{\theta_{ld}}$, where $\sum_{i=1}^{d} \theta_{li}$ is a constant, as stated before. Thus,

$$-\ln Pr_{max}\left(\mathbf{Y}=e\right) = \min_{l\backslash e:\mathcal{L}:e} \{-\theta_{l1}\ln s_1 - \theta_{l2}\ln s_2 - \ldots - \theta_{ld}\ln s_d\}. \tag{5.1}$$

Let us call this function g_e. It is continuous, piecewise linear in the variables $-\ln s_1, -\ln s_2, \ldots,$ $-\ln s_d$, and concave, satisfying all the properties of a tropical polynomial. The vector $(\theta_{l1}, \theta_{l2}, \ldots, \theta_{ld})$ represents the number of times each event of probability $s_1, s_2, \ldots, s_d$ occurs.

So, our problem reduces to finding the tropical hypersurface of the function g_e. Such a function, which gives us the explanation that maximizes the probability of seeing the evidence is called an *inference function*.

An elegant result by Elizalde and Wood [16] states that the number of inference functions of a probabilistic graphical model is polynomially bounded.

Theorem 5. [16] *In a graphical model with E edges and d parameters, the number of inference functions for the graphical model is at most $O\left(E^{d(d-1)}\right)$.*

Vertices in the Newton polytope of an inference function

Since an inference function g_e of a statistical model with d-parameters is basically a tropicalization of the marginal probability polynomial f_e, we can encode the inference function in a space of d-dimensions by constructing the Newton polytope of f_e. Each exponent vector $(\theta_{l1}, \theta_{l2}, \ldots, \theta_{ld})$, as stated in the previous section, represents the number of times each event of probability $s_1, s_2, \ldots, s_d$ occurs, and the vertices of the Newton polytope will encode a set of transition events which yield the evidence.

It can be proved that the number of vertices in the Newton polytope of f_e is at most polynomial in the size of the model. The result depends on the following theorem by Andrews in 1963.

Theorem 6. [2] *If P is a D-dimensional strictly convex lattice polytope, with N vertices, then*

$$N < C_D \cdot \textit{vol}\,(P)^{\frac{D-1}{D+1}},$$

where vol (P) is the volume of the polytope, and C_D is a constant that depends only on D.

Since the polytope of an inference function f_e must lie in the space $[0, E]^D$, where E is the number of edges in the graphical model, and D is the number of parameters in the model, we can thus bound the number of vertices by $C_D \cdot E^{D(D-1)/(D+1)}$. However, this result only holds for full dimensional polytopes. If the polytope P lies in a d-dimensional affine subspace of $\mathbb{R}^D$, then we need to consider the following lemma.

Lemma 1. [33] *Let $\mathcal{S}$ be a d-dimensional linear subspace of $\mathbb{R}^D$. Then, there exists a subset $\{i_1, i_2, \ldots, i_d\}$ of the D coordinate axes of $\mathbb{R}^D$ such that the projection $\phi : \mathcal{S} \mapsto \mathbb{R}^d$, given by $\phi\left((x_1, x_2, \ldots, x_D)\right) = (x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ is injective.*

Proof. Choose $v_1, v_2, \ldots, v_d \in \mathbb{R}^D$ to be a basis for the subspace $\mathcal{S}$. Then, we can construct a $D \times d$ matrix A of rank d, whose columns are the vectors $v_1, v_2, \ldots, v_d$. Assume that for any choice of indices $\{i_1, i_2, \ldots, i_d\}$, there does not exist a mapping $\phi\left((x_1, x_2, \ldots, x_D)\right) = (x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ which is injective on $\mathcal{S}$. Then, the $d \times d$ minor of A, choosing the rows indexed by $\{i_1, i_2, \ldots, i_d\}$, must necessarily have a rank strictly less than d, since we can find two vectors in $\mathbb{R}^D$ with the same entries in $i_1, i_2, \ldots, i_d$. This contradicts the fact that the rank of A is d, thus proving the lemma. □

This lemma leads us to the next theorem.

Theorem 7. [33] *Let f be a polynomial of degree n in D variables. If the dimension of $\mathcal{P} = NP\,(f)$ is d, then the total number of vertices in the Newton polytope will be bounded by $C_d \cdot n^{d(d-1)/(d+1)}$, where C_d is a constant that depends only on d.*

Proof. Consider $\mathcal{S}$ to be a d-dimensional affine span of the polytope. Using Lemma 1, we can find a set of d coordinate axes of $\mathbb{R}^D$ such that the projection ϕ of $\mathcal{S}$ onto the space determined by those d axes is injective. Thus, $\phi(\mathcal{P})$, the image of all points of the polytope $\mathcal{P}$, will be a d-dimensional polytope with integer coordinate vertices, every point in one-to-one correspondence with a point of $\mathcal{P}$, and the vertices of $\mathcal{P}$ mapping to the vertices of the projection. Since f has degree n, $\phi(\mathcal{P})$ must lie in the d-dimensional hypercube of volume n^d. Using Theorem 6, the total number of vertices in $\mathcal{P}$ will be bounded by $C_d \cdot n^{d(d-1)/(d+1)}$. □

Putting these results together, we get the following theorem for the number of vertices in the Newton polytope of an inference function of a graphical model.

Theorem 8. [32] *Consider a graphical model with E edges, with d parameters, and state space $\mathcal{E}$. For a fixed evidence $e \in \mathcal{E}$, the number of vertices in the Newton polytope of the polynomial map f_e, is bounded above as follows*

$$\begin{aligned}\textit{Number of vertices in NP}\left(f_e\right) &\leq c \cdot E^{d(d-1)/(d+1)} \\ &\leq c \cdot E^{(d-1)},\end{aligned}$$

where c is a constant.

Thus, while the number of monomials in the polynomial f_e can grow exponentially with the number of sum and product operations, only a polynomial number of these terms can be maximal, and those are the terms we are interested in.

Interpretation of the cones and the fans

In Section 4.3.1, we discussed the interpretation of cones of a vertex v in a Newton polytope in d dimension as the set of points $N_P(v)$ such that they minimize the functional $w \cdot v = w_1 v_1 + w_2 v_2 + \ldots + w_d v_d$ for all $w \in N_P(v)$.

Since the vertices now represent sets of transitions, the cone of a vertex θ_l represents points that yield the minimum value of $\min_{l \setminus e : \mathcal{L} : e} \{-\theta_{l1} w_1 - \theta_{l2} w_2 - \ldots - \theta_{ld} w_d\}$, where w is a point in the cone. Comparing with Equation 5.1, this means that the cone encodes the set of parameters for which the set of transitions represented by the vertex is an optimal solution. These parameters are encoded as negative logarithms of the actual probabilities used in the model. Furthermore, from

Section 4.2.1, we also see that the the cones of the vertices, which form a fan, partition the entire parameter space. This leads to an elegant method to explore the parameter space for probabilistic models, which involves constructing the Newton polytopes of the marginal probability polynomials.

5.2 Computation

In Chapter 2, we discussed the maximum a posteriori probability problem. It is a natural tropicalization of the sum product algorithm, in which we replace the products by sums in the negative log-parameter space, and sums by taking the minimum. If we can backtrack through the algorithm, then we get the labelling which maximizes the probability of the evidence. Since there is an established relation between the tropical semiring and the polytope algebra, the algorithm can be modified to a polytope setting, and this gives us a useful way to explore the parameter space.

5.2.1 Polytope propagation

The sum-product algorithm to compute the marginal posterior probability of an event was an operation carried out on the classical arithmetic semiring $(\mathbb{R}, +, \times)$. A similar algorithm, carried out on the tropical semiring, solves the MAP problem. An example of the latter is the Viterbi algorithm for sequence alignment.

In order to solve the parametric a posteriori maximum likelihood problem, a *polytope propagation* algorithm was proposed by Sturmfels and Pachter [32]. This generalizes the tropical semiring algorithm to the space of all parameters in the negative logarithmic space. It involves the same dynamic programming technique as the sum-product algorithm, but moves to polytope algebra by constructing the Newton polytopes of the marginal probability polynomials. The switch between algebras is illustrated in Table 5.1.

There is some loss of information when we use polytope algebra, since we cannot encode the coefficients of the polynomials in the Newton polytope. However, if the coefficient of a certain monomial happens to be greater than 1, it simply means that that there is more than one evidence consistent labelling that can be obtained through the same set of transitions. Thus, the probability of a single consistent labelling, when represented through a monomial, will have coefficient 1.

Sum-product	Polytope algebra analogue
Transition probability	Unit vector along a coordinate axis
$\times$	Minkowski Sum
$+$	Convex hull
1	Origin
0	Empty polytope

Table 5.1: From sum-product to polytope propagation

5.2.2 Vertices of a subset of the parameters

As stated in Section 5.1, the cone of a vertex encodes the set of parameters that would yield a maximum a posteriori labelling which is consistent with the transitions encoded by the vertex. Often, we might have a rough idea about the parameters used for the model, and would like to find the sets of transitions that would be optimal for a certain subset of parameters. The naive method of doing this would be to sample a large number of parameters and find the MAP labelling for each of them. However, this gives us no guarantee that all optimal scenarios for that subset will be covered.

In a polytope setting, this problem would correspond to finding the vertices of the polytope whose cones contain the subset of parameters. A preliminary examination seems to suggest that this is a hard problem, even if we are given the cones that cover the subset, because the construction of a cone depends not only on its vertex v, which minimizes the functional $w \cdot v$ for all points w in the cone, but also on all the other points in the polytope, which never includes a point that might yield a smaller functional.

Polytope propagation has been used successfully for sequence alignment [5, 12, 31]. The elegance of this framework is that it can be applied to all sum-product min-plus algorithms. The work of Dewey et al. on Drosophilia genomes [12] used a non-probabilistic framework and found optimal scenarios for the Needleman-Wunsch algorithm for various parameters. The strength of taking a non-probabilistic point of view is that they do not have to deal with parameter dependencies, and can work in a lower dimensional space. By discarding a subset of parameters which is biologically unreasonable, they were able to get a small set of optimal scenarios for sequence alignment. These scenarios were then compared with scenarios provided by well known sequence alignment software called BLASTZ [39], and were shown to agree well. This comparison also lets allows other users of BLASTZ to assess if the default parameters used by the software is reasonable for their data. The cones of the vertices also provide robustness measures by partitioning the parameter space.

Finally, the paper of Dewey et al. also discusses the reconstruction of phylogenetic trees. Since this reconstruction depends on the branch length, which in turn is inferred through the alignment of the genomes at the leaves for a given set of parameters, it would be useful to have a parametric view of the problem. With this in mind, they compute the set of optimal alignments of the genomes, i.e. at the vertices of the alignment polytope they obtain, and use this to propose a parametric reconstruction scheme for phylogenetic trees.

In the case of Bayesian networks, and for most probabilistic graphical models, we have to deal with intersections of the polytope space with non-algebraic curves. The next three chapters will discuss the implementation of polytope propagation for Bayesian networks and application to both real and simulated data.

Part III

Experiments

The final section of the thesis expands on Chapter 5, and discusses the implementation of a polytope propagation scheme for Bayesian networks used in evolutionary biology. The work builds upon applications of polytope propagation to hidden Markov models for sequence alignment.

Chapter 6 discusses the well known belief propagation algorithm for inference of marginal probabilities in Bayesian networks. This algorithm is translated to an algorithm in polytope algebra using the scheme provided in Table 5.1. The chapter also discusses the input for the problem and issues that we face when applying the algorithm to a Bayesian network.

Chapter 7 uses the techniques discussed in the previous chapter to explore the evolutionary scenarios of the bZIP transcription factors. Statistics for runtime and the size of the polytope are provided, and the output is compared to Dollo and Fitch parsimony results on the same data.

Chapter 8 includes the last set of experiments to be performed, which involves generating simulated evolutionary scenarios for the bZIP network, and applying polytope propagation on these scenarios. The same statistics for runtime and polytope size are provided, along with comparisons with deterministic approaches. In addition, the simulated data is used to examine the effect of a model with greater number of parameters on the polytope propagation algorithm for Bayesian networks.

Chapter 6

Implementation

The polytope propagation algorithm for acyclic Bayesian networks with binary nodes was implemented in polymake, a C++ and perl based interactive software for handling complex polytopes [22]. The main advantages of using polymake were that it was open source, and allowed us to program at a high level, without bothering about the background algorithms for convex hulls and Minkowski sums. The current SVN version of polymake also provided tools to construct cones and fans, which would be the ultimate goal of the whole project.

6.1 Algorithm

The algorithm used for the sum-product decomposition for polytopes was a direct translation of the belief propagation algorithm for marginal probabilities. By using the dictionary provided in Chapter 5, we could program polytope propagation in perl script.

6.1.1 Classical belief propagation

In the classical belief propagation algorithm [30, 34], a tree data structure is created, with each node A having the following attributes. For the rest of the discussion, we shall always assume binary random variables at each node.

1. λ values: The probabilities of 0 or 1 at A based purely on evidence from nodes in the tree

rooted at A. Denoted by $\lambda\left(A=a\right)$ for node A in state a.

2. π values: The probabilities of 0 or 1 at A based purely on evidence from nodes in the tree above A. Denoted by $\pi\left(A=a\right)$ for node A in state a.

3. A *lambda message* $\lambda_A\left(p\right)$ to the parent, assuming the parent is in state p, informing it about the evidence coming from the nodes in the tree rooted at A.

4. A *pi message* $\pi_C\left(a\right)$ to a child C of A, assuming node A is in state a, informing them about the evidence coming from the nodes in the tree above A.

5. The value of the random variable, set to 0, 1, or some other value, indicating that the node is not initialized, i.e. it is not an evidence node.

To illustrate the algorithm, we show the messages needed to compute the probability of node D being in state 1 in the following example. In this example, $p+q=1$ and $s_{00}+s_{01}=s_{10}+s_{11}=1$.

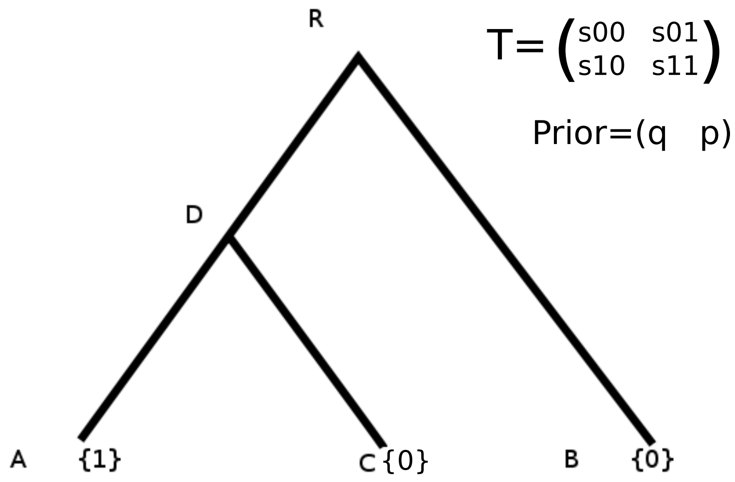


Figure 6.1: Example for belief propagation

After initializing node A as 1, we pass a lambda message from A to D for every possible value of the random variable at D. This lambda message, in the case when D takes the value 1, $\lambda_A\left(1\right)$, will be $s_{10}Pr\left(A=0,D=1\right)+s_{11}Pr\left(A=1,D=1\right)$. The first term goes to zero, since node A can never take the value 0, according to the evidence. The lambda message from C to D, when D takes the value 1, $\lambda_C\left(1\right)$, will be $s_{10}Pr\left(C=0,D=1\right)+s_{11}Pr\left(C=1,D=1\right)$. In this case, the second term goes to 0, since C can never be 1. The probability that D is in state 1, based on evidence from the nodes in the tree rooted at D, i.e. $\lambda\left(D=1\right)$, is the product of these two messages.

The lambda message from B to R, for the case when $R=0$, i.e. $\lambda_B\left(0\right)$, will be $s_{00}Pr\left(B=0,R=0\right)$ $+\ s_{01}Pr\left(B=1,R=0\right)$, with the second term vanishing. For the case when $R=1$, the message $\lambda_B\left(1\right)$ will be $s_{10}Pr\left(B=0,R=1\right)+s_{11}Pr\left(B=1,R=1\right)$, and the second term vanishes again. The pi-message from R to D, in the case $R=1$, $\pi_D\left(1\right)$, will be $\pi\left(R=1\right)\cdot\lambda_B\left(1\right)$. In terms of the prior probabilities, this is $p\cdot\lambda_B\left(1\right)$. This is the probability that R is in state 1 based solely on evidence from branches other than the branch going to D. The corresponding message when $R=0$, $\pi_D\left(0\right)$, will be $q\cdot\lambda_B\left(0\right)$. To calculate the π-value of node D when $D=1$, $\pi\left(D=1\right)$, we multiply the pi-messages by transition probabilities that take the node D to the state 1, and add them up, which gives us $s_{01}\pi_D\left(0\right)+s_{11}\pi_D\left(1\right)$. This is the probability of $D=1$ based on evidence coming from the parent.

The marginal probability of node D being in state 1, and the evidence, will be $\lambda\left(D=1\right)\cdot\pi\left(D=1\right)$. Similarly, the marginal probability for $D=0$ can also be calculated. Furthermore, when the evidence is restricted to the leaves, the marginal probability of the evidence is $\sum_{j\in\{0,1\}}\lambda\left(j\right)\cdot\pi\left(j\right)$ at any non-evidence node. So, we can just calculate this value at the root.

If we treat the parameters are formal variables, then it is clear that we will get a polynomial for the marginal probability of the evidence. The lambda and pi messages are factors of the final marginal probability polynomial at any node. Grouping the factors in this fashion allows the construction of a polynomial time algorithm to infer the marginal probabilities at each node in the tree.

6.1.2 Belief propagation for polytope-propagation

The polytope propagation algorithm for Bayesian networks is simply the belief propagation algorithm, with all the values and messages replaced by convex polytopes. We also change the operations according to the dictionary. Then, the operation $\lambda\left(0\right)\odot\pi\left(0\right)\oplus\lambda\left(1\right)\odot\pi\left(1\right)$ gives us the final propagated polytope at any node.

For our purposes, we make a major modification in the belief propagation algorithm. Since we are not interested in the individual labels at the nodes, we only propagate evidence messages towards the root. There is no feedback from a node to its children, and there is no need for the pi-messages. While this means that the nice ‘universal marginal probability of evidence’ structure that we got while using conventional belief propagation is lost, this saves considerably on computational time, since we only need to cover each edge once, and the polytope at the root gives us the correct propagated polytope.

6.1.3 Input and output

The algorithm takes a tree and the binary evidence at the leaves as input. The tree is provided in the form of an adjacency list, and the evidence is a list of nodes with their labelling given alongside. This model assumes that the same transition matrix is present along every edge, and that all nodes are binary. The transition matrix has four entries, s_{00}, s_{10}, s_{01} and s_{11}, where the subscripts follow the convention of Chapter 2. Then, since there are four parameters, we make the following substitutions when moving to polytope algebra.

$$\begin{bmatrix} s_{00} & s_{10} \\ s_{01} & s_{11} \end{bmatrix} \rightarrow \begin{bmatrix} (1000) & (0100) \\ (0010) & (0001) \end{bmatrix}$$

As an option, a third input, which indicates whether each node duplicates or speciates, can be provided. In this case, the program moves to an 8 parameter setting, with different transition matrices for duplicating and speciating nodes.

The program output is a polytope in either $\mathbb{R}^4$ or $\mathbb{R}^8$, depending on the input. This is the propagated polytope at the root.

6.1.4 Constructing the fan

After running the algorithm, the cones of the vertices of the polytope can be found. The maximal cones of the polytope, i.e. the cones of maximal dimension, which are not contained in any other cones and are in bijection with the vertices, partition the parameter space. For each vertex, a set of parameters chosen from its cone will give an evidence consistent labelling which maximizes the probability of seeing the evidence and has a transition set which is represented by the vertex.

6.1.5 Restricting the space of parameters

When using polytope propagation, we disregard the dependencies between parameters, and work outside the probability simplex, i.e. the parameters need not be in the region $[0, 1]$. Thus, while the fan gives us a partition of the whole parameter space into cones, some of these will be redundant.

In order to restrict the parameter space to the probability simplex, we note that, in a single parameter, taking the negative logarithm will map the point 1 to the origin, and will map all the points in $(0, 1]$ to the set $[0, \infty)$. The set $(1, \infty)$, which is outside the probability simplex, maps onto the negative real line.

Since all our parameters are probabilities, the positive cube in d-dimensions, $(0, 1]^d$, will map onto the first quadrant when we tropicalize the statistical model. Thus, we only need to consider maximal cones that lie in the first quadrant.

The second problem we face is that of including dependencies between the parameters. In particular, we need only consider parameters in which the columns add up to 1. In the tropical space, the parameters $s_{00}, s_{10}, s_{01}, s_{11}$ would translate into the parameters $x_{00}, x_{10}, x_{01}, x_{11}$, where $x_{ij} = -\ln s_{ij}$. Then, we would have to consider the conditions that $s_{00} + s_{01} = s_{10} + s_{11} = 1$, since the columns in the stochastic matrix along each edge should sum up to 1.

Including this in the polytope propagation scheme is hard, because the tropical semiring and polytope algebra have no analogous operation for subtraction. This means that we have to move to a non-algebraic setting to resolve this problem, and take intersections of the cone encoding the parameters with the curves in 4-space, $e^{-x_{00}} + e^{-x_{01}} = 1$ and $e^{-x_{10}} + e^{-x_{11}} = 1$, would give us the set of parameters which are probabilities that yield an MAP labelling which agrees with the transitions represented by the vertex.

6.2 Computational complexity

Worst case complexity results for the polytope propagation algorithm are derived by studying the sum-product decomposition of the marginal probability polynomial. This tells us exactly how many Minkowski sum and convex hull operation we will have to perform.

Theorem 8, tells us that the polytopes generated during the algorithm for a model with D parameters cannot have more than $C_d n^{d(d-1)/(d+1)}$ vertices, where n is the degree of the final marginal polynomial, and d is the dimension of the affine space that the Newton polytope of this polynomial lies on. Let us call this bound N. Assume, now, that we have k steps in the sum-product decomposition of the polynomial, with at most l additions and l multiplications. This translates to l convex hull operations and l Minkowski additions. The value of k will vary with the model.

The bound on the number of vertices in the polytopes means that the number of points in the Minkowski sum of two polytopes will be at most N^2. Calculating the sum will take $O\left(N^2 D\right)$ time, since we have D components to add in each vector. The problem with naively taking this sum is that the number of points will grow exponentially with the number of sum operations. This causes a memory overflow error, even on very small models. To avoid this, we take the convex hull of the N^2 points after each summing operation.

The same bound works for stand alone convex hull operations, since the convex hull of two polytopes can have at most $2N$ different vertices. Clearly, the Minkowski sum operation will dominate the convex hull operation, since, in a worst case scenario, it involves a convex hull of $O\left(N^2\right)$ points.

In the case of convex hull operations following a Minkowski sum, we have to check a set of at most N^2 points, and see which of these points lies on a hyperplane in $\mathbb{R}^D$ which separates it from the rest of the set. This translates to a series of linear constraints, and yields a linear optimization problem with N^2 linear programs in D variables and N^2 constraints. There are a number of algorithms to solve this problem, and in particular, the algorithm by Megiddo [28] solves it in linear time in N^2 for a fixed number of parameters, albeit with a constant of proportionality which is exponential in D. Another algorithm by Kachiyan [24], solves it in polynomial time in N^2, with a constant which is polynomial in D. Depending on the algorithm, we can assume that the complexity of computing the convex hull of N^2 points will be $\nu_D\left(N^2\right)$.

The overall complexity for the sum-product algorithm will then be $O\left(klDN^2 + kl\nu_D\left(N^2\right)\right)$, since stand alone convex hull operations on at most $2N$ points will never dominate the Minkowski sum operation. The complexity will vary, since polymake uses a variety of convex hull algorithms [3,20] depending on the case it is being applied to.

Chapter 7

Experiments on the bZIP network

The first part of our experiments was to apply the methods discussed to real data. For this, we used the bZIP interaction network. We include the details of the bZIP network, including the parameters used by Pinney et al. [35], in Appendix A. For this dataset, we restricted ourselves to the four-parameter model. The eight-parameters model proved computationally intractable to be applied on our data.

7.1 Input

The bZIP gene tree has 383 genes, over 7 species in the species tree. The extant species are *Ciona intestinalis*, *Takifugu rubripes* (pufferfish), *Danio rerio* (zebrafish), and *Homo sapiens* (humans). The internal species are *Chordata,* which is the common ancestor for all the species, *Vertebrata*, which is the common ancestor to humans and the two fishes, and *Teleosti*, which is the common ancestor to both fishes. The gene tree is reconciled, and has sequence divergence scores over each edge.

The interaction tree constructed from this has 6850 nodes. This tree is too large for efficient computation. In order to continue with the analysis, the tree was broken down into 327 smaller trees, rooted at a speciating *Chordata* interaction. The parameters and initialization of the model relied on the work of Fong, Keating and Singh [19]. The bZIP data has been well studied and there are sequence-based methods to infer the strength of interactions.

The interactions at the extant species were initialized based on Fong-Singh interaction scores [19]. These scores are based on sequencing data, and were used to predict both the extant interactions, as well as the interactions in extinct species based on sequence reconstruction. The scheme is explained in Appendix A. The sequencing data is considered reliable enough to use as a standard to measure other techniques against.

The approach used by Pinney et al. fixed the parameters for the transition matrices over each node using the interaction scores, as explained in Appendix A, and used classical belief propagation to calculate the marginal probabilities at each node of the whole interaction tree.

The input to the algorithm was given in the form of an adjacency list for the tree, and a list of leaves with the evidence on them.

7.2 Output

Ideally, the model should have only 2 parameters, i.e. the probabilities of gain or loss over an edge. However, as stated in Section 6.1.5, the absence of an analogue for subtraction in the tropical semiring and polytope algebra means that expressions such as $s_{00} = 1 - s_{01}$ cannot be represented nicely. The polytope that was created through polytope propagation for the model, thus, lay in 4-dimensional space.

7.3 Results

The runtime for the algorithm and the size of the polytope obtained were the main statistics of interest. Apart from them, we also compared the output to Dollo and Fitch parsimony results on the tree.

7.3.1 Runtime

The computation time, on a system with Quad-socket, 6-core AMD Istanbul processor, with 256 GB of memory, takes about 8 hours. The runtime for different trees in our set, in seconds, is plotted against the size of the tree in Figure 7.2. The plot has been reduced to trees of size 60 or lower, to emphasize the variation of runtime.

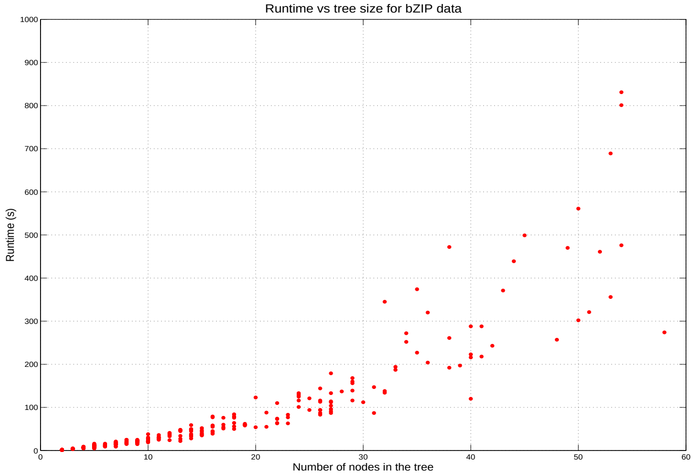


Figure 7.1: Distribution of runtime vs tree size

The runtime increased with tree depth, and for trees of comparable size, polytope propagation took longer on trees with greater depth. This was expected behaviour, as greater depth would indicate Minkowski sums of progressively larger polytopes as the messages are passed to the root.

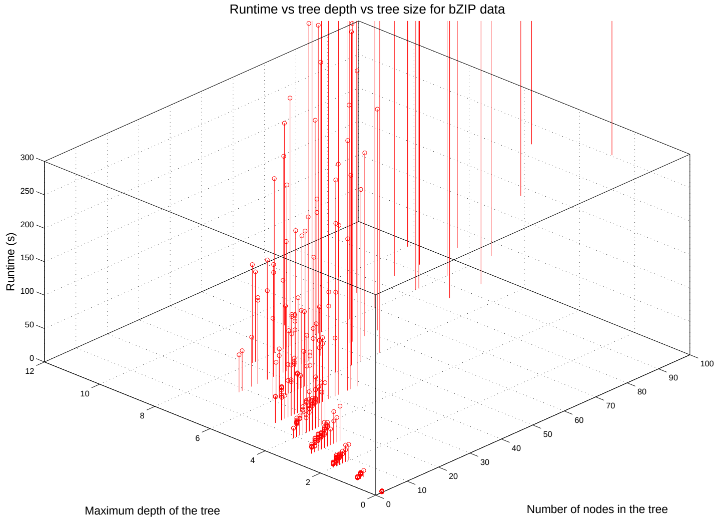


Figure 7.2: Distribution of runtime vs tree depth

It was also expected that the runtime would increase with the number of homodimer duplications, since we need 3 Minkowski addition operations at each such node, but no appreciable increase was noticed.

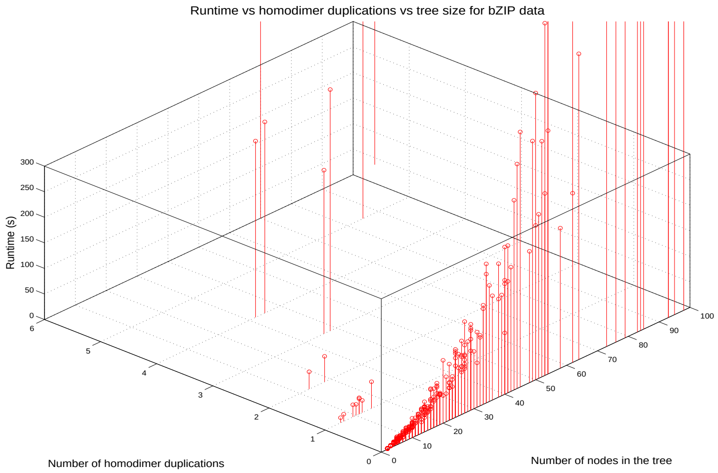


Figure 7.3: Distribution of runtime vs homodimer duplications

7.3.2 Polytope size

The number of vertices in the polytope, in the worst case, should not exceed $O\left(n^3\right)$, where n is the size of the tree. This follows from Theorem 8. In fact, since our polytope will lie on a three-dimensional affine space in four dimensions, we can say that the number vertices will be bounded by $O\left(n^2\right)$, where n is the number of edges in the tree, using Lemma 1. On running the algorithm, we find experimental evidence for this bound. The number of vertices also depends on the tree topology. Thus, trees with smaller size may be associated to polytopes with greater number of vertices than trees of comparatively larger size.

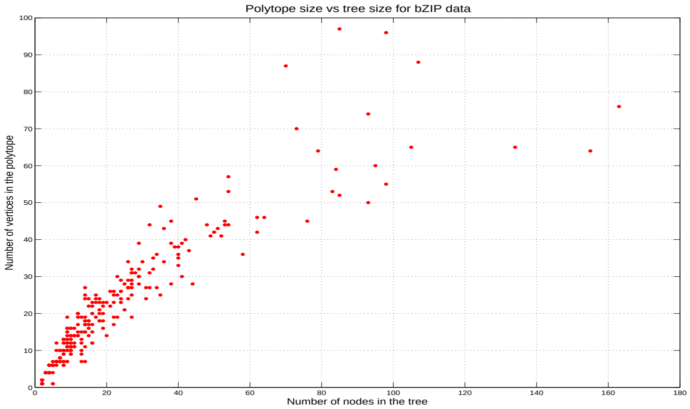


Figure 7.4: Distribution of polytope size vs tree size

Interestingly, the size of the polytope also increases with the tree depth. Greater depth would mean that the number of Minkowski sum operations along the branch will increase, and the subsequent polytopes generated by them will be larger.

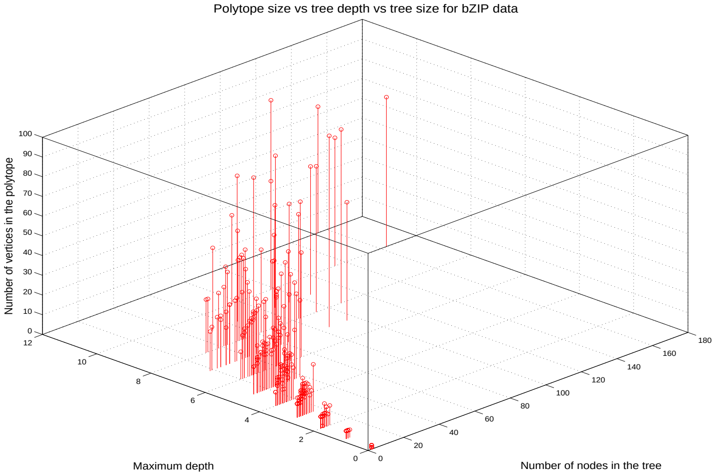


Figure 7.5: Distribution of polytope size vs tree depth

This corroborates our runtime results, since larger polytopes at any stage would imply that the

Minkowski sum and subsequent convex hull operations will take longer.

7.3.3 Comparison to Dollo parsimony

It was also of interest to contrast the results for polytope propagation with the results for Dollo parsimony on the same trees. The idea was to find the number of $0 \to 0,\ 0 \to 1,\ 1 \to 0,$ and $1 \to 1$ transitions in the labelling obtained through Dollo parsimony. This was called the *Dollo signature* of the tree for that evidence.

Since the vertices of the polytope corresponded to similar signatures that, in some sense, maximize the probability of seeing the evidence, we computed the Hamming distance from each vertex to the Dollo signature, and found the minimum distance over all vertices. We plotted this against tree size and the number of vertices in the polytope.

When comparing the Dollo signatures of the data to the vertices of the polytope, there are a large number of cases with at least one vertex corresponding to the Dollo signature, the maximum hamming distance is 7, for a 26 node tree which yields a 24 vertex polytope. In total, there are 322 trees with a Dollo hamming distance of 0. This means that there is one vertex in their polytopes whose cone encodes parameters that have a low probability of gain. This indicates that there is often

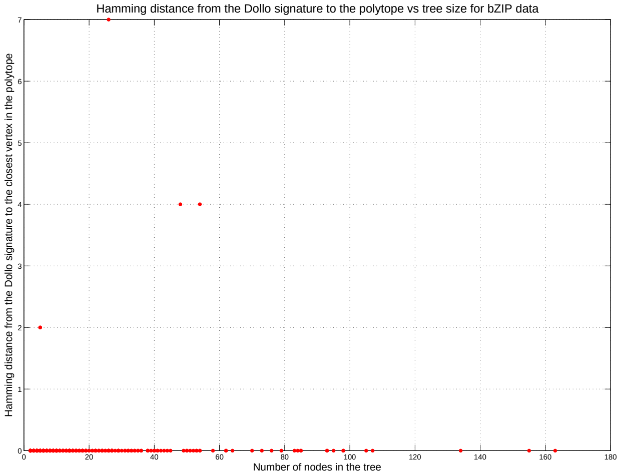


Figure 7.6: Distribution of Dollo hamming distance vs tree size

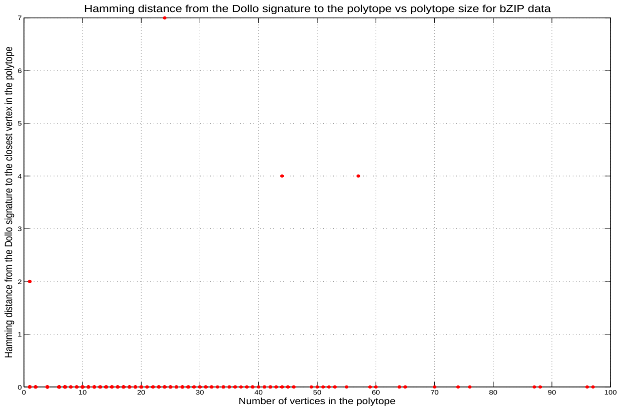


Figure 7.7: Distribution of Dollo hamming distance vs polytope size

a scenario in which there is a single gain, since Dollo parsimony only allows a single gain during evolution.

7.3.4 Comparison to Fitch-Hartigan parsimony

As in the case of Dollo parsimony, we define the *Fitch signature* of the tree to be the vector of the number of transitions of each type when we compute a Fitch parsimonious scenario.

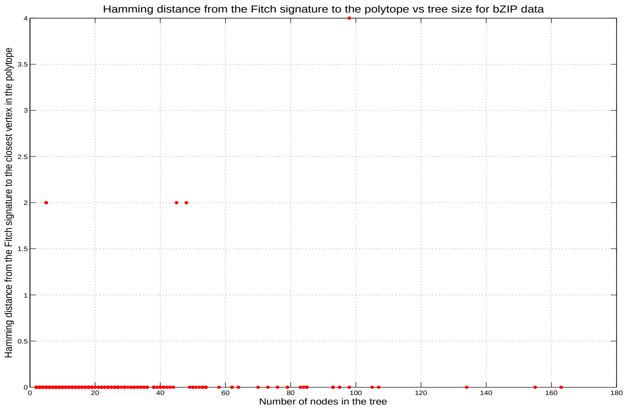


Figure 7.8: Distribution of Fitch hamming distance and tree size

The Fitch signatures of the trees using the bZIP data yielded good results when compared to the vertices of the propagated polytopes. There were a very large number of trees with a vertex corresponding to the Fitch signature.

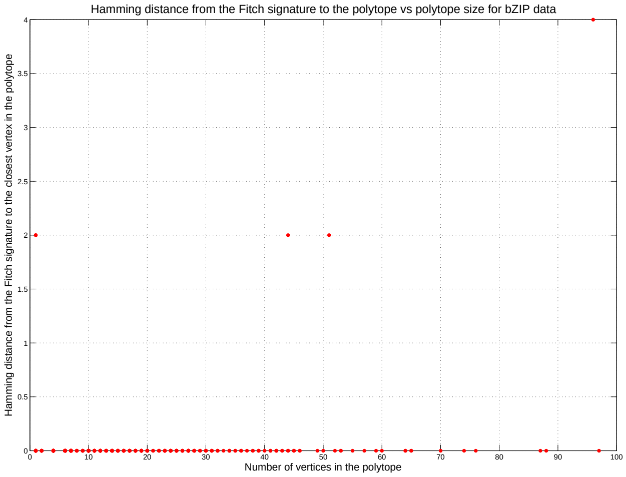


Figure 7.9: Distribution of Fitch hamming distance and polytope size

In all, there are 5 trees with non-zero Hamming distance. The maximum hamming distance was 4, on a tree of size 98, with 96 vertices in the polytope. As far as the experiments are concerned, there is often a probable set of transitions which is an optimal MAP solution for some set of parameters, and also corresponds to a Fitch-Hartigan parsimonious reconstruction of the trees. It is also interesting that there are only 5 cases in which the Fitch and Dollo hamming distances do not match each other, and there are 2 cases when they do match, but have non-zero hamming distance. This means that in most cases, the Fitch and Dollo signatures are the same, and there is a vertex in the propagated polytope which is equal to that signature.

Intersecting the cones of these vertices for 90 of the largest polytopes, with more than 14 vertices, we get a subspace of the affine 4-space in which the parameters might yield an MAP transition signature which is equivalent to the Fitch and Dollo signatures. Intersections of this space with the non-algebraic curves in 4-space, as stated in Section 6.1.5, would give us the actual set of parameters that yield a parsimonious MAP labelling and also correspond to transition probabilities.

7.4 Summary

These results tell us that a parsimonious scenario for bZIP evolution often corresponds to a maximum a posteriori probability scenario for some set of parameters. In view of the fact that there are an exponential number of possible evolutionary scenarios, and thus, a possibly exponential number of monomials in the marginal probability polynomial, this is a surprising result.

Since the number of vertices is polynomial in the number of nodes in the Bayesian network, it is also possible to explore the space of all optimal scenarios for evolution and compare the effect of the parameters from the cones of these vertices when used for classical belief propagation. The next chapter discusses polytope propagation on simulated evolutionary scenarios.

Chapter 8

Simulations

The polytope propagation algorithm was also run on simulated data generated from the bZIP transcription factor interaction tree. The experimental data from the paper from Pinney et al. [35] had concluded that the probability of gain and loss are generally small, with the probability of gain tending to be smaller. The aim of the simulation was to compare the results over simulated data with the real data.

8.1 Four parameter model

For the first run of simulations, we assumed that all the edges carried the same 4 parameters. Thus, the statistical model was defined on 4 parameters.

8.1.1 Random data

The probabilities of gain and loss (note that we need only 2 parameters to generate a simulation scenario) were picked with uniform probability over the $[0, 1]$ interval. The prior probability was taken to be 0.5. At the root, we picked a number in $[0, 1]$ with uniform probability and labelled the root 1 if the number was greater than 0.5 and labelled it 0 otherwise. Then, over each edge, we picked a number in $[0, 1]$ with uniform probability, and depending on the label of the parent, we labelled the child 1 if the parent was labelled 1 and the number picked was less than the probability of loss, or if the parent was labelled 0 and the number was greater than the probability of gain.

Similarly, if the parent was labelled 0 and the number was less than the probability of gain, or if the parent was labelled 1, and the number was greater than the probability of loss, we labelled the child 0.

The process mentioned above gives us a complete labelling of the set of interaction trees we had mentioned in the previous chapter. The input we were interested in was the labelling at the leaves which represented interactions in the extant species. This was taken to be our evidence, and the labelling on the rest of the tree was hidden.

8.1.2 Results

The polytope propagation algorithm was carried out on the interaction trees using the evidence set created above.

Runtime

The input is provided as an adjacency list of the edges in the tree, and a list of leaves with evidence. The computing specifications are the same as that for the bZIP data. The computation took a maximum of 18 hours. As in the previous case, the graph is truncated for trees up to the size of 60 to show the runtime trend.

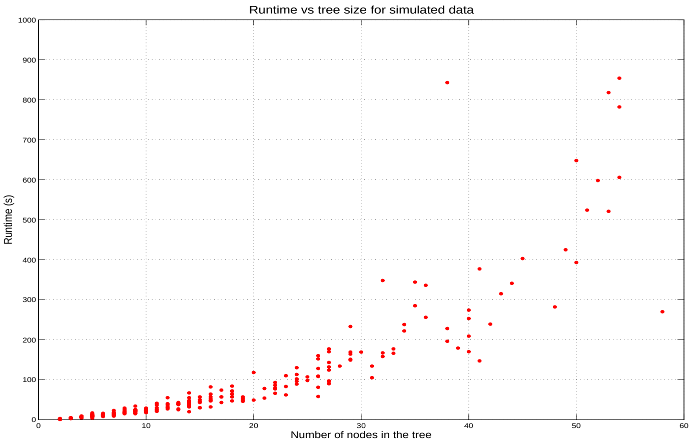


Figure 8.1: Distribution of runtime vs tree size for 4 parameters

Here again, the size of the tree is not the only parameter that affects the computation time. A major parameter to consider was the depth of the tree. A plot of the runtime versus the tree depth again shows that for trees of comparable size, the polytope propagation algorithm takes longer to run on the tree with greater depth.

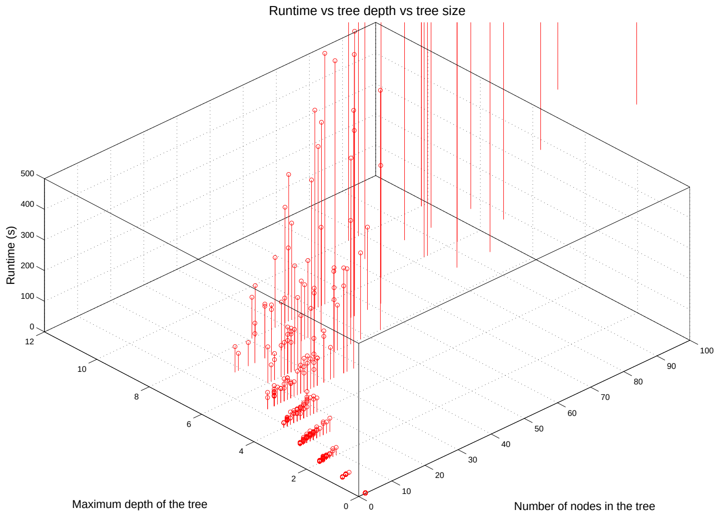


Figure 8.2: Distribution of runtime vs tree depth for 4 parameters

We also expected the runtime to increase with the number of homodimer duplications in the tree, since each homodimer duplication is a node of outdegree 3, and there would be two Minkowski sum operations at each such node. But such a correlation, if any, was weak.

Polytope size

The polytope size shows great similarity to the results obtained from the bZIP data.

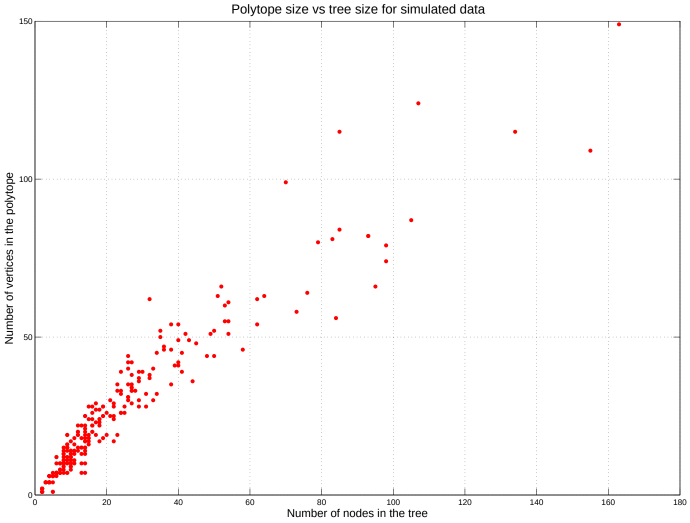


Figure 8.3: Distribution of polytope size vs tree size for 4 parameters

Again, we have support for the $O\left(n^2\right)$ upper bound for number of vertices. This statistic also shows a gradual upward trend as we keep the size of the tree fixed and vary the tree depth. This is illustrated in the following graph.

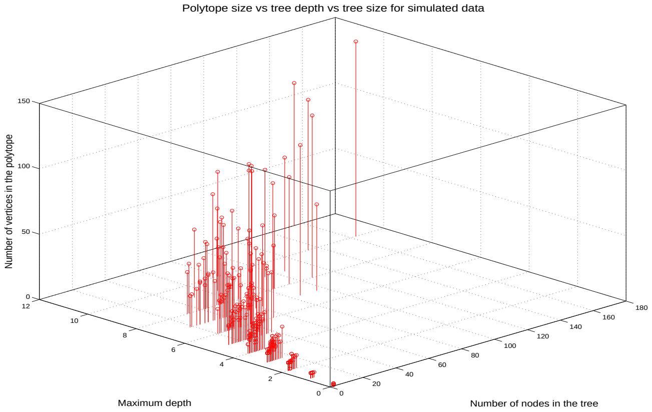


Figure 8.4: Distribution of polytope size vs tree depth for 4 parameters

As stated before, this is a result of progressively larger Minkowski sum operations.

8.1.3 Comparison to Dollo parsimony

As with the bZIP data, the Dollo signatures were computed for each tree with simulated input. This signature was compared to the vertices of the propagated polytope of the tree.

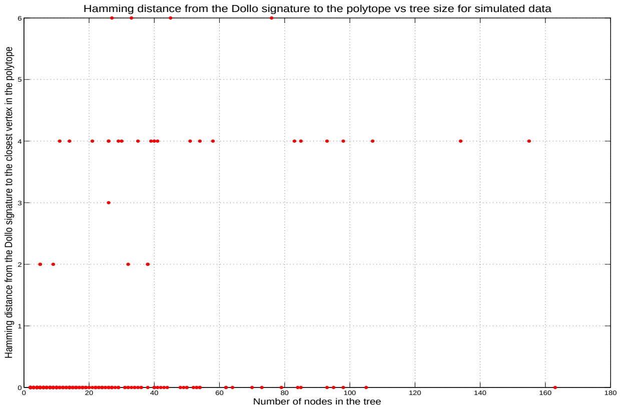


Figure 8.5: Distribution of Hamming distance from Dollo signature vs tree size

The Hamming distance for many trees was greater than 0, which meant that there were fewer cases in which at least one optimal scenario with only a single gain existed than in the case of the real data.

The maximum Hamming distance was 6, for four trees, the largest of which had 76 nodes and yielded a polytope with 64 vertices, and the smallest of which had 27 nodes and had a propagated polytope with 29 vertices.

8.1.4 Comparison to Fitch-Hartigan parsimony

As in the case of the bZIP data, we also computed Fitch signatures for the simulated scenarios and compared them to the vertices of the polytope.

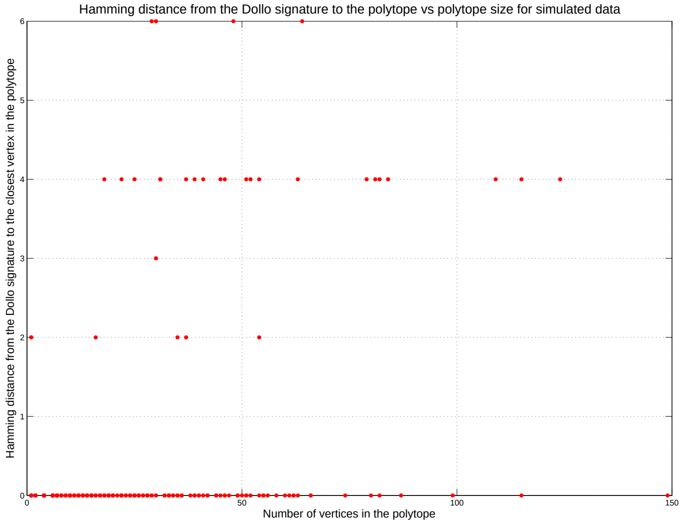


Figure 8.6: Distribution of Hamming distance for Dollo signature vs polytope size

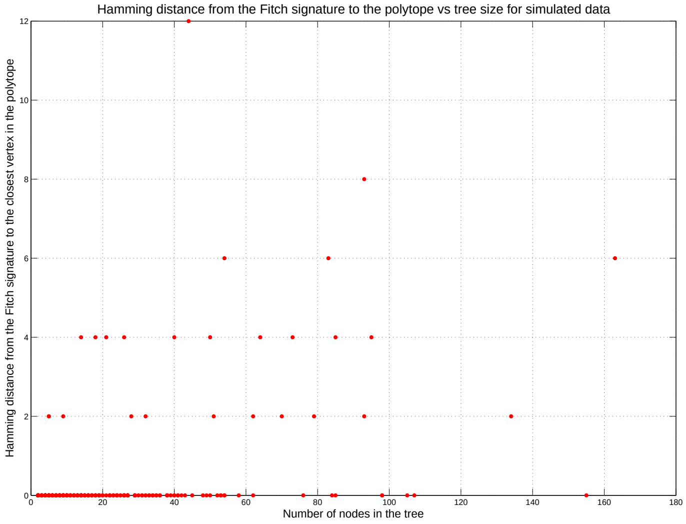


Figure 8.7: Distribution of Hamming distance from Fitch signature vs tree size

The maximum Hamming distance observed for Fitch-Hartigan parsimony, compared to the vertices

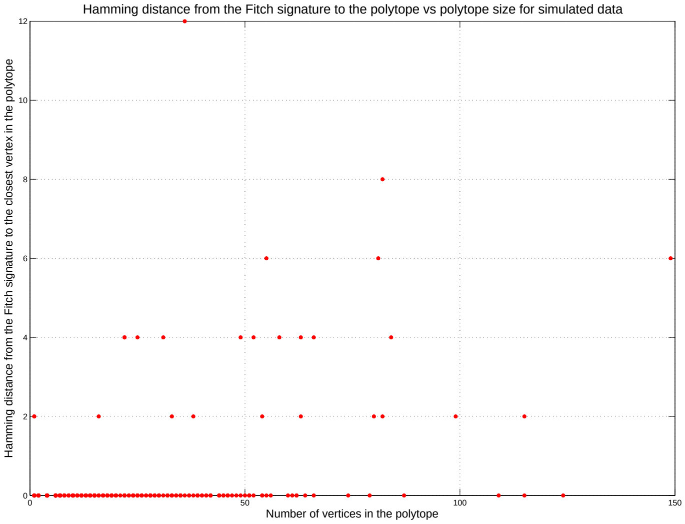


Figure 8.8: Distribution of Hamming distance for Fitch signature vs polytope size

of the polytope, was 12, was found for a tree on just 44 nodes, which implied that the simulation for that tree did not yield a parsimonious scenario which minimized the number of transitions along the edges.

There were 301 trees in which there was a vertex corresponding to the Fitch signatures, i.e. they had a Fitch hamming distance of 0, while 295 trees had a vertex corresponding to their Dollo signatures. Thus, parsimonious MAP scenarios of evolution were rarer for the simulated data.

8.2 Eight parameter model

The eight parameter model is based on the scheme provided by Dutkowski and Tiuryn [15]. The probabilities of gain and loss are taken to be different for speciating and duplicating interactions, giving rise to two different transition matrices. Thus, instead of 4 parameters for the entire model, we have 8.

8.2.1 Random data

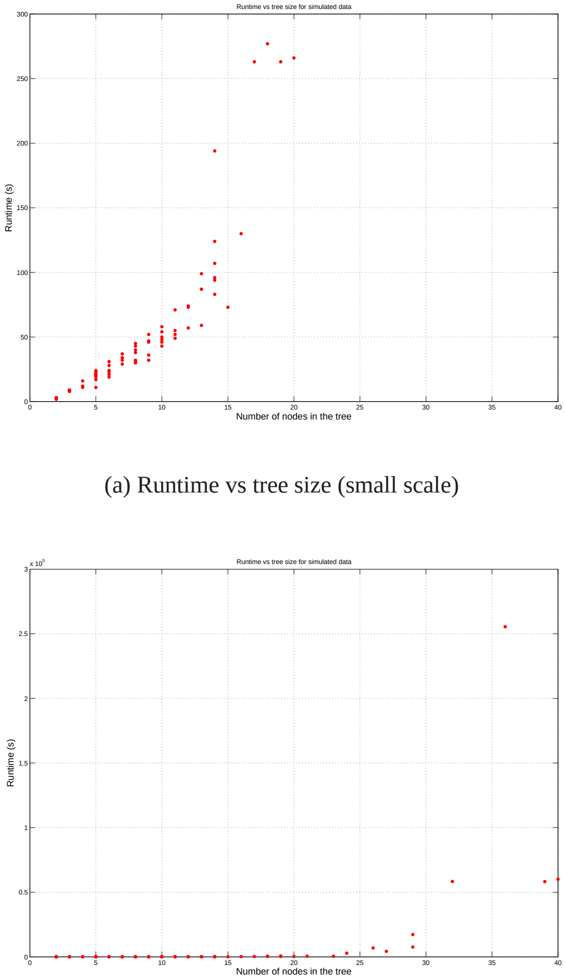
The process for simulation in the 8 parameter model was almost exactly the same as that for the 4 parameter model. However, we kept track of the additional information on whether an interaction duplicated or if it evolved into two new species. Instead of 2 numbers in $[0, 1]$, we picked 4, representing the probabilities of gain and loss for a speciating interaction, and for a duplicating interaction.

8.2.2 Polytope propagation for eight parameters

The algorithm took an extra input parameter for the 8 parameter model. We had to provide information on whether a node was duplicating or speciating. Also, the propagated polytope for this model was constructed in $\mathbb{R}^8$, and it lay on an affine space of $\mathbb{R}^7$. The number of vertices was bound by $O\left(n^6\right)$. Furthermore, the complexity increases considerably in eight dimensions, and the computation for even small trees (less than 45 nodes) takes over 3 weeks on an Intel(R) Xeon(R) E5520 processor, clocked at 2.27 GHz, with 37 GB of memory.

8.2.3 Runtime

The runtime taken for polytope propagation using the eight parameter model was considerably greater than that for the four parameter model. The plots below show the trend as the size of the tree grows from 0 to 20 nodes, and from 0 to 40 nodes.



(a) Runtime vs tree size (small scale)

(b) Runtime vs tree size

Figure 8.9: Runtime characteristics for the eight parameter model

Note the rapid increase in runtime. While computations for the four parameter model finished in under 500 seconds for trees smaller than 40 nodes, the same computation here takes as long as 30000 seconds. This is in agreement with the theoretical result that runtime is exponential in the dimension of the affine space that the polytope lies in.

8.2.4 Polytope size

The number of vertices in the propagated polytope also showed a noticeable increase. The trend was no longer sub-quadratic, as we had observed in the four-parameter model. However, only 93 of the 327 trees were used to generate these statistics, as the computation proved to be too intractable for trees of size larger than 45 nodes.

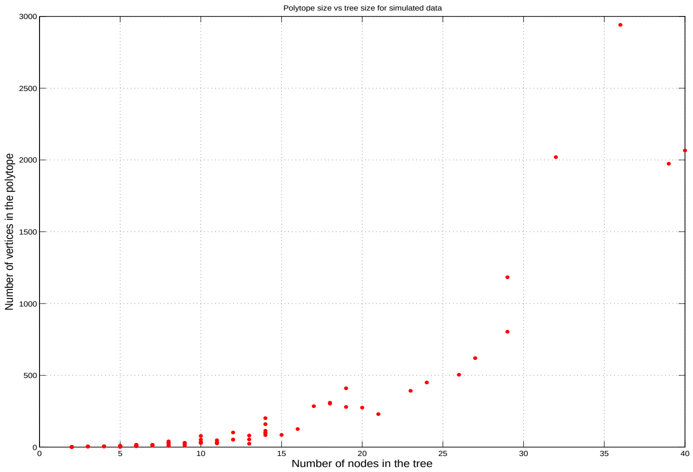


Figure 8.10: Polytope size vs tree size for eight parameters

8.3 Summary

Experiments on simulated data show a marked difference in the Fitch signature results when compared to the results from the bZIP data. The bZIP data almost always had a vertex corresponding to the Fitch signature of the tree, while that is not true of the simulations. This indicates that the probabilities of gaining or losing a character was probably low during bZIP evolution.

Other trends on the statistics of polytope size and runtime were in agreement with the real data. The eight parameter model was an interesting example of the problems involved in moving polytope propagation to higher dimensions. At the moment, there does not seem to be a tractable way to use polytope propagation on probabilistic models with a large number of parameters.

Chapter 9

Conclusions

In this thesis, we discussed the main computational techniques used for the inference of ancestral protein-protein interactions and the preliminary use of an framework algebraic statistics framework to analyze their evolutionary history.

Inference techniques in computational biology are generally classified into deterministic and probabilistic techniques. Both of them are very well studied, and have been applied to the inference of ancestral protein-protein interactions. The crux of the thesis lies in the interpretation of probabilistic models, such as those used for evolutionary models on trees, as algebraic varieties.

The work of Sturmfels et al [21, 32] also extended this interpretation to an interpretation of probabilistic models in tropical algebra. Such an interpretation yields a natural translation of the model into polytopes, with the cones of the vertices of the polytopes encoding parameters that yield optimal scenarios. Furthermore, it is possible to construct these polytopes through successive Minkowski sum and convex hull calculations.

The technique of polytope propagation, which is an extension of the belief propagation algorithm to polytopes, was applied to both, real data on the bZIP interaction networks, and simulated data for protein-protein interactions, and the results were compared to the theoretical bounds that were known. They were also compared to well known deterministic models of evolution on trees. This comparison gave us an insight into the possible evolutionary path of the interactions. The experiments also included an extension of the polytope propagation algorithm to an evolutionary model with a larger number of parameters.

The results obtained indicate that while polytope propagation provides an elegant mathematical framework, using it on Bayesian networks for evolution is fraught with difficulties. Evolutionary tree models in biology usually rely on branch lengths to specify parameters. For example, in the paper by Pinney et al. [35], the branch length of each edge on the interaction tree is used as a parameter to compute the probabilities of gain and loss over that edge. The current implementation of polytope propagation for the same tree cannot incorporate this scale of complexity.

There are other models, such as the one by Dutkowski and Tiuryn [15], which do not use branch length to estimate evolutionary parameters. However, the prospect of extending polytope propagation to a full-fledged parametric inference technique, even for small Bayesian trees, seems distant at the moment.

The extension of linear hidden Markov models to polytope propagation has been well studied and has been used successfully on real data [5, 12, 31]. However, in the absence of a linear graphical structure, such as one for sequence alignment or recombination, as in Bayesian networks that model evolution, this extension is encumbered by the necessity to take intersections with non-algebraic hypersurfaces in the space of the parameters. This was a major obstacle in the approach taken, and warrants further research to find a scheme that can reduce the number of parameters.

Another area of further research would be to identify a method to handle probabilistic models with a large number of parameters. Convex hull and Minkowski sum computation in higher dimensions is a field of major research, and it may be possible to extend such results to polytope propagation.

The problem of finding the set of optimal evolutionary scenarios for a subset of the parameter space is still open, as stated in Section 5.2.2. The naive experimental method of finding the optimal solutions is not guaranteed to give all the possible scenarios, and it might be useful, though difficult, to find an algebraic statistics approach to the same.

Finally, the translation of belief propagation to polytope propagation also means that while we get possible transition scenarios in the Bayesian network that may have led to the evidence, we cannot know which or how many internal labellings yield the same transition scenario. It would be useful to devise a backtracking algorithm which associates each vertex in the propagated polytope with the internal labellings that yield the set of transitions represented by the vertex.

To summarize, an algebraic statistics approach to inference in evolutionary biology is still far from complete. While the mathematical background is well laid, the implementation of efficient methods to compute and analyze relevant statistics for the the problem needs further investigation. A few possible directions to approach this include massive parallelization of the polytope propagation algorithm and the implementation of more efficient Minkowski sum solvers. It would be interesting to examine how a low parameter model computed through algebraic statistics compares with high parameter models of evolution. Apart from this, a non-probabilistic approach to parametric inference, such as the one taken by Dewey et al. [12], may also provide an idea which lets us handle parametric inference in evolutionary models. Certain discrete algorithms, such as Sankoff parsimony, can be naturally translated to polytope propagation, and this solves the problems of high dimensionality and non-algebraic intersections.

Appendix A

The bZIP transcription factors

The data set of interest to us is the protein interactions occurring in the family of the bZIP transcription factors. They are a family of proteins involved in the regulation of development, metabolism, circadian rhythm, and other cellular processes. The family exhibits a high rate of gene duplication, and the bZIP subfamilies have broadly conserved interaction patterns with each other. There are accurate genome-scale experimental data for the family, and a process to estimate the strength of interactions based on amino acid sequences exists, which makes the bZIP family particularly useful for investigating methods for reconstruction ancestral networks.

The paper by Pinney et al. [35] investigates the reconstruction of ancestral protein interactions in the bZIP family by using a Bayesian network modelled by the interaction tree.

A.1 The interaction tree

The reconciled gene tree for the bZIP family is already provided. Using this, we can easily construct the interaction tree, starting with the assumption that the protein at the root could have been self-interacting. The gene tree has 383 nodes, and yields an interaction tree with 6850 nodes, of which 2227 are interactions in extant species.

A.2 Parameters

The paper selects parameters for gain and loss experimentally, based on the true-positive and true-negative extant human bZIP interactions, by considering all possible moves in the sequence space, from each strongly interaction pair, or each non-interacting pair, and modelling the probabilities of loss and gain, respectively. as a function of the sum of branch lengths of the two genes corresponding to the interacting proteins. Then, they fitted this data to logistic functions of the sum of branch lengths. At the root, the prior was chosen to be 0.5. The functions that were fitted to the data were $0.0809/\left(1+\exp\left(-2.9495\left(d-1.6409\right)\right)\right)$ for the probability of gain and $0.9219/\left(1+\exp\left(-5.886\left(d-1.2887\right)\right)\right)$ for the probability of loss, where d is the sum of evolutionary distances from their parents for both proteins in an interaction. This yields 6849 different stochastic matrices along each edge, with different gain and loss percentages over each edge.

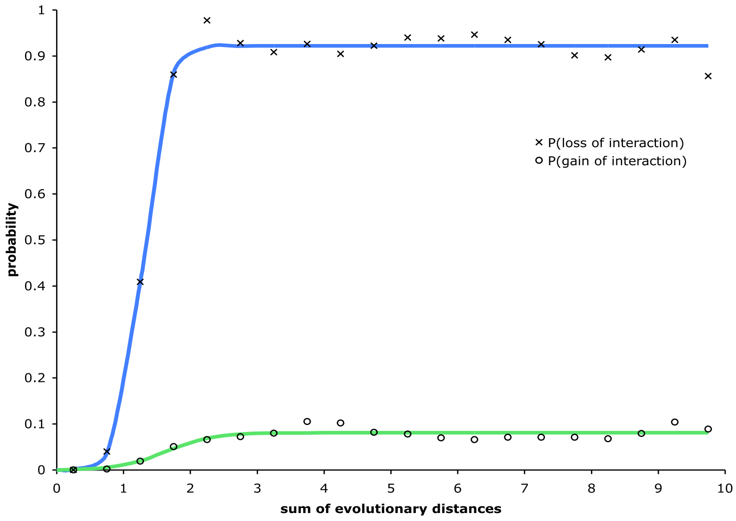


Figure A.1: Probabilities of gain and loss for human interactions versus sum of evolutionary distances [35]

A.3 Evidence

Using the true-positive and true-negative human interactions, score distributions were derived for strongly interaction and non-interacting protein pairs. These distributions were fitted to normal distributions that varied over the Fong-Singh scores [19] for the interaction protein pairs in the extant species.

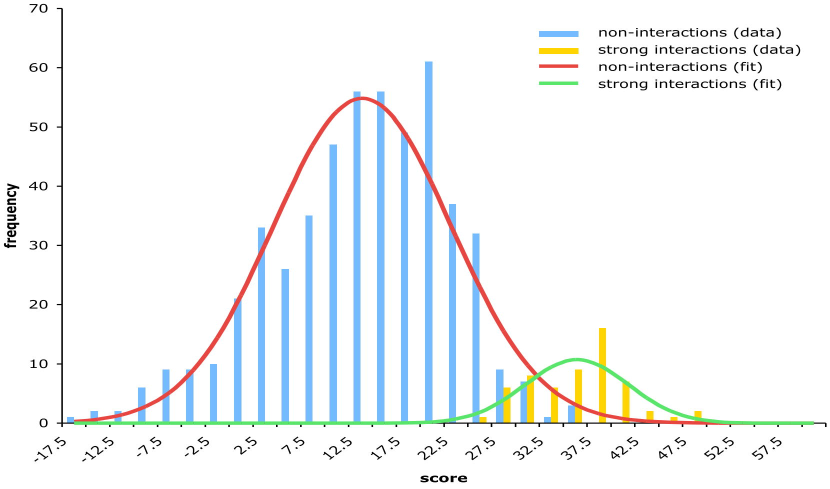


Figure A.2: Fong-Singh predictions for strong and weak interactions [35].

A score of 30.6 was found to correspond to the probability of an interaction being 0.5, and it was taken as the cut-off score for the binary evidence.

Appendix B

Basics of probability

This appendix is supposed to provide a brief overview of probability theory, and the terminology used in it.

B.1 Probability space

Suppose we define an experiment with finite sample space $\Omega = \{e_1, e_2, \ldots, e_n\}$, where each e_i is an outcome of the experiment. A subset of this sample space is called an *event*.

Definition 20. [30] *A function that assigns a real number $Pr\left(E\right)$ to each event $E \subseteq \Omega$, is called a probability function on the set of subsets of Ω if it satisfies the following conditions:*

1. $0 \leq Pr\left(\{e_i\}\right) \leq 1$ *for* $1 \leq i \leq n$.
2. $\sum_{i=1}^{n} Pr\left(\{e_i\}\right) = 1$.
3. *For each E, as long as E is not a singleton set,*

$$Pr\left(E\right) = \sum_{e_k \in E} Pr\left(\{e_k\}\right).$$

We say that (Ω, P) define a probability space.

The number $Pr\left(E\right)$ assigned to an event $E \subseteq \Omega$ is called the probability of E, and Pr is said to be a map on Ω.

A *random variable* is a function on Ω which assigns a unique value to each element in the sample space. The set of values a random variable X can assume is called the *space* of X.

The following theorem holds for all probability spaces.

Theorem 9. [30] *For any probability space* Ω, Pr*,*

1. $Pr\left(\Omega\right) = 1$.
2. $0 \leq Pr\left(E\right) \leq 1$ *for any* $E \subseteq \Omega$.
3. *For* $E \subset \Omega$ *and* $F \subset \Omega$*, if* $E \cap F = \emptyset$*, then,*

$$Pr\left(E \cup F\right) = Pr\left(E\right) + Pr\left(F\right).$$

These properties are called the *axioms of probability*.

B.2 Conditional probability and independence

Definition 21. *For any two events* E *and* F *in* Ω*, if* $Pr\left(F\right) \neq 0$*, the* conditional probability *of* E *given* F*,* $Pr\left(E|F\right)$*, is given by*

$$Pr\left(E|F\right) = \frac{Pr\left(E \cap F\right)}{Pr\left(F\right)}$$

If the condition $E \cap F = \emptyset$ holds, the two events E and F are said to be mutually exclusive.

In terms of probability functions, two events E and F are independent if one of the following holds.

1. $Pr\left(E|F\right) = Pr\left(E\right)$ and $Pr\left(E\right) \neq 0, Pr\left(F\right) \neq 0$.
2. $Pr\left(E\right) = 0$ or $Pr\left(F\right) = 0$.

If two events E and F are independent, then $Pr\left(E \cap F\right) = Pr\left(E\right) \cdot Pr\left(F\right)$.

Based on the definitions of conditional probability and independence, we can now define conditional independence as follows.

Definition 22. [30] *Two events* E *and* F *are said to be conditionally independent given another event* G *and* $Pr\left(G\right) \neq 0$ *if one of the following holds:*

1. $Pr\left(E|F \cap G\right) = Pr\left(E|G\right)$ *and* $Pr\left(E|G\right) \neq 0, Pr\left(F|G\right) \neq 0$.
2. $Pr\left(E|G\right) = 0$ *or* $Pr\left(F|G\right) = 0$.

B.3 Bayes Theorem

Using the definitions of conditional probabilities and conditional independence, we can prove the following theorem.

Theorem 10. [30] *[Bayes' theorem] Given two events E and F, such that $Pr\left(E\right) \neq 0$ and $Pr\left(F\right) \neq 0$, then*

$$Pr\left(E|F\right) = \frac{Pr\left(F|E\right)Pr\left(E\right)}{Pr\left(F\right)}$$

This is the central theorem in Bayesian inference. It states, very roughly, that given the outcome of a certain event, it is possible to find the probability of an event which may have led to that outcome.

Bibliography

[1] E. M. Airoldi, *Getting started in probabilistic graphical models*, PLoS Comput Biol **3** (200712), no. 12, e252.

[2] G. E. Andrews, *A lower bound for the volume of strictly convex bodies with many boundary lattice points*, Trans. Amer. Math. Soc. **106** (1963), 270–279. MR0143105 (26 #670)

[3] D. Avis and K. Fukuda, *A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra*, Discrete Comput. Geom. **8** (1992), no. 3, 295–313. ACM Symposium on Computational Geometry (North Conway, NH, 1991). MR1174359 (93h:68137)

[4] S. Basu, R. Pollack, and M. F. Roy, *Algorithms in real algebraic geometry*, Second, Algorithms and Computation in Mathematics, vol. 10, Springer-Verlag, Berlin, 2006. MR2248869 (2007b:14125)

[5] N. Beerenwinkel, C. N. Dewey, and K. M. Woods, *Parametric inference of recombination in HIV genomes*, eprint arXiv:q-bio (December 2005), available at `arXiv:q-bio/0512019`.

[6] M. A. Calderwood, K. Venkatesan, L. Xing, M. R. Chase, A. Vazquez, A. M. Holthaus, A. E. Ewence, N. Li, T. Hirozane-Kishikawa, D. E. Hill, M. Vidal, E. Kieff, and E. Johannsen, *Epstein-barr virus and virus human protein interaction maps*, Proceedings of the National Academy of Sciences **104** (2007), no. 18, 7606–7611, available at `http://www.pnas.org/content/104/18/7606.full.pdf+html`.

[7] C. Chauve, J. P. Doyon, and N. El-Mabrouk, *Inferring a duplication, speciation and loss history from a gene tree (extended abstract)*, Comparative genomics, 2007, pp. 45–57. 10.1007/978-3-540-74960-8_4.

[8] ———, *Gene family evolution by duplication, speciation, and loss*, J. Comput. Biol. **15** (2008), no. 8, 1043–1062. MR2461961 (2009m:92077)

[9] C. Chauve and N. El-Mabrouk, *New perspectives on gene family evolution: losses in reconciliation and a link with supertrees*, Research in computational molecular biology, 2009, pp. 46–58.

[10] D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms*, Third, Undergraduate Texts in Mathematics, Springer, New York, 2007. An introduction to computational algebraic geometry and commutative algebra. MR2290010 (2007h:13036)

[11] J. De Las Rivas and C. Fontanillo, *ProteinâĂŞprotein interactions essentials: Key concepts to building and analyzing interactome networks*, PLoS Comput Biol **6** (201006), no. 6, e1000807.

[12] C. N Dewey, P. M Huggins, K. Woods, B. Sturmfels, and L. Pachter, *Parametric alignment of drosophila genomes*, PLoS Comput Biol **2** (200606), no. 6, e73.

[13] T. Dobzhansky, *Nothing in biology makes sense except in the light of evolution*, The American Biology Teacher **35** (1973), no. 3, 125–129.

[14] J. Dutkowski, *Evolution of protein-protein interaction networks*.

[15] J. Dutkowski and J. Tiuryn, *Identification of functional modules from conserved ancestral proteinâĂŞprotein interactions*, Bioinformatics **23** (2007), no. 13, i149–i158, available at `http://bioinformatics.oxfordjournals.org/content/23/13/i149.full.pdf+html`.

[16] S. Elizalde and K. Woods, *Bounds on the number of inference functions of a graphical model*, Statist. Sinica **17** (2007), no. 4, 1395–1415. MR2398601 (2009e:62028)

[17] J. Felsenstein, *Inferring phytogenies*, Sunderland, Massachusetts: Sinauer Associates (2004).

[18] W.M. Fitch, *Toward defining the course of evolution: minimum change for a specific tree topology*, Systematic Biology **20** (1971), no. 4, 406.

[19] J. Fong, A. Keating, and M. Singh, *Predicting specificity in bzip coiled-coil protein interactions*, Genome Biology **5** (2004), no. 2, R11.

[20] K. Fukuda and A. Prodon, *Double description method revisited*, Combinatorics and computer science (Brest, 1995), 1996, pp. 91–111. MR1448924 (98c:90108)

[21] L. D. Garcia, M. Stillman, and B. Sturmfels, *Algebraic geometry of bayesian networks*, Arxiv preprint math/0301255 (2003).

[22] E. Gawrilow and M. Joswig, *polymake: a frame work for analyzing convex polytopes*, Polytopes—combinatorics and computation (Oberwolfach, 1997), 2000, pp. 43–73. MR1785292 (2001f:52033)

[23] D. Graur and W. H. Li, *Fundamentals of molecular evolution*, Vol. 7, Sinauer Associates Sunderland, MA, 2000.

[24] L. G. Hačijan, *Polynomial algorithms in linear programming*, Zh. Vychisl. Mat. i Mat. Fiz. **20** (1980), no. 1, 51–68, 260. MR564776 (81j:90079)

[25] L. Hakes, J.W. Pinney, D.L. Robertson, and S.C. Lovell, *Protein-protein interaction networks and biologyâĂŤwhat's the connection?*, Nature biotechnology **26** (2008), no. 1, 69–72.

[26] C. G. Knight and J. W. Pinney, *Making the right connections: biological networks in the light of evolution*, BioEssays **31** (2009), no. 10, 1080–1090.

[27] A. C. F. Lewis, R. Saeed, and C. M. Deane, *Predicting protein–protein interactions in the context of protein evolution*, Mol. BioSyst. **6** (2009), no. 1, 55–64.

[28] N. Megiddo, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach. **31** (1984), no. 1, 114–127. MR821388 (87b:90082)

[29] B. Mirkin, T. Fenner, M. Galperin, and E. Koonon, *Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes*, BMC Evolutionary Biology **3** (2003), no. 1, 2.

[30] R. E. Neapolitan, *Learning Bayesian Networks*, illustrated edition, Prentice Hall, 2003.

[31] L. Pachter and B. Sturmfels, *Parametric inference for biological sequence analysis*, Proc. Natl. Acad. Sci. USA **101** (2004), no. 46, 16138–16143 (electronic). MR2114587

[32] ______, *Tropical geometry of statistical models*, Proc. Natl. Acad. Sci. USA **101** (2004), no. 46, 16132–16137 (electronic). MR2114586

[33] ______ (ed.), *Algebraic statistics for computational biology*, Cambridge University Press, New York, 2005. MR2205865 (2006i:92002)

[34] J. Pearl, *Reverend bayes on inference engines: A distributed hierarchical approach*, Proceedings of the national conference on artificial intelligence, 1982, pp. 133–136.

[35] J. W. Pinney, G. D. Amoutzias, M. Rattray, and D. L. Robertson, *Reconstruction of ancestral protein interaction networks for the bzip transcription factors*, Proceedings of the National Academy of Sciences **104** (2007), no. 51, 20449–20453, available at `http://www.pnas.org/content/104/51/20449.full.pdf+html`.

[36] J. Richter-Gebert, B. Sturmfels, and T. Theobald, *First steps in tropical geometry*, Idempotent mathematics and mathematical physics, 2005, pp. 289–317. MR2149011 (2006d:14073)

[37] D. Sankoff, *Minimal mutation trees of sequences*, SIAM Journal on Applied Mathematics **28** (1975), no. 1, 35–42.

[38] D. Sankoff and P. Rousseau, *Locating the vertices of a steiner tree in an arbitrary metric space*, Mathematical Programming **9** (1975), no. 1, 240–246.

[39] S. Schwartz, W.J. Kent, A. Smit, Z. Zhang, R. Baertsch, R.C. Hardison, D. Haussler, and W. Miller, *Humanïźč mouse alignments with blastz*, Genome research **13** (2003), no. 1, 103.

[40] R. V. Sole, R. Pastor-Satorras, E. Smith, and T. B. Kepler, *A model of large-scale proteome evolution*, Arxiv preprint cond-mat/0207311 (2002).

[41] B. Sturmfels, *Gröbner bases and convex polytopes*, University Lecture Series, vol. 8, American Mathematical Society, Providence, RI, 1996. MR1363949 (97b:13034)

[42] ______, *Solving systems of polynomial equations*, CBMS Regional Conference Series in Mathematics, vol. 97, Published for the Conference Board of the Mathematical Sciences, Washington, DC, 2002. MR1925796 (2003i:13037)

[43] H. Zhu, F. Domingues, I. Sommer, and T. Lengauer, *Noxclass: prediction of protein-protein interaction types*, BMC Bioinformatics **7** (2006), no. 1, 27.