UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik

Modellierung von Reverse Engineering Strategien zur Identifizierung genetischer Netzwerke aus unvollständigen Genexpressionsdaten

Diplomarbeit

Aufgabenstellung und Betreuung:

Prof. Dr. M. Löffler
Dr. D. Drasdo

Leipzig, 31. März 2003

vorgelegt von:

Missal, Kristin
geb. am: 14. Juni 1977

Studiengang
Medizininformatik

Zusammenfassung

Genetische Netzwerke zeigen wie Gene über ihre Produkte wieder andere Gene regulieren. Sind die Netzwerktopologie und die Art der Einflüsse bekannt, können Vorhersagen über das dynamische Verhalten der individuellen genetischen Expression von Zellen getroffen werden. Mögliche Modelle für genetische Netzwerke sind *Boolesche Netze* und *Dynamische Bayessche Netze*. Genregulationsnetzwerke zu analysieren und zu verstehen, ist auf einer abstrakten Ebene mit Hilfe eines Computers und dieser Modelle möglich.

In der vorliegenden Arbeit wird auf der Basis von in-silico Experimenten analysiert, wie ein Modell für genetische Netzwerke aus Genexpressionsdaten von einzelnen Zellen gelernt werden kann, wenn nur unvollständiges Wissen über die initialen Genexpressionszustände vorliegt. Der initiale Expressionszustand wird unvollständig festgelegt, indem die Expressionsstärke einiger Gene gezielt manipuliert wird.

Boolesche Netze repräsentieren das genetische Netzwerk der in-silico Zellen. Ihre Regeln sind deterministischer Art und sind bei vollständig gegebenen Daten mit dem Reverse Engineering Algorithmus REVEAL einfach rekonstruierbar. REVEAL hat keinen Ansatz für unbeobachtete Werte in den Daten. Es wird gezeigt, dass die Inputelemente und Booleschen Regeln für Elemente lernbar sind, deren Anzahl an Inputelementen kleiner oder gleich der manipulierbaren Gene ist. Durch Rauschen in den Daten ist es jedoch unmöglich deterministische Beziehungen korrekt zu charakterisieren. Deshalb wird angestrebt, aus den künstlichen Expressionsdaten ein Dynamisch Bayessches Netz zu lernen. Es modelliert die verbleibende Unsicherheit über die Abhängigkeiten in dem genetischen Netzwerk. Eine Analyse des Verfahrens Strukturelle Erwartungswert Maximierung (SEM) ergab, dass die fehlenden Beobachtungen umgangen werden müssen.

Eine getrennte Auswertung der Experimente, die sich in den manipulierten Genen unterscheiden, ist ein Weg ein gutes Modell zu lernen, wenn mindestens zwei Gene gleichzeitig manipulierbar sind. Kann die Expressionsstärke nur von einem Gen festgelegt werden, sind mit dieser Strategie die regulierenden Gene identifizierbar, die unabhängig von den anderen regulierenden Genen den Expressionszustand des Zielgens wesentlich bestimmen.

Qualitatives Vorwissen über das interessierende genetische Netzwerk kann eine umfangreiche Verringerung des notwendigen Stichprobenumfangs herbeiführen.

Danksagung

Ich möchte mich bei all den Beteiligten für das Gelingen dieser Arbeit bedanken.
Ein ganz besonderer Dank geht an Dr. D. Drasdo für die umfassende Betreuung, die anregenden Diskussionen, die hilfreichen Hinweise und für die Bereitschaft, sich mit meinen Fragen und Problemen auseinanderzusetzen.
Vielen Dank an Prof. Dr. M. Löffler für die sehr spannende Aufgabenstellung.
Besonderer Dank gebührt ebenfalls Dr. M. Cross. Er hat stets bereitwillig und mit viel Geduld meine Fragen über die Biologie beantwortet.
Meinen herzlichsten Dank auch an Dr. D. Hasenclever für die fruchtbaren Diskussionen über statistische Fragestellungen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die vorliegende Arbeit untergliedert sich in fünf Kapitel. Das erste Kapitel dient zur Einführung in die biologische Fragestellung und gibt einen Überblick über Modelle und Reverse Engineering Algorithmen für genetische Netzwerke. Außerdem formuliert es die Ziel- und Problemstellung.
Das zweite Kapitel geht kurz auf Maße der Informationstheorie ein, die in dieser Arbeit eine zentrale Rolle spielen.
Im dritten Kapitel werden die ausgwählten Modelle und Reverse Engineering Strategien detailliert beschrieben.
Das vierte Kapitel dient zur Darstellung der durchgeführten Simulationen und Ergebnisse.
Im fünften Kapitel wird untersucht, ob das genetische Netzwerk von multipotenten hämatopoetischen Stammzellen rekonstruiert werden kann, wenn unvollständige Genexpressionsdaten gegeben sind.

1.1 Spezifische Genexpression

Mehrzellige Lebewesen sind aus verschiedenen Zelltypen aufgebaut, die jeweils eine charakteristische Morphologie haben und auf eine bestimmte Funktion spezialisiert sind. Die Entwicklung und Erhaltung des differenzierten Zustandes beruht zum Teil auf spezifischer Genexpression. Spezifische genetische Expression bedeutet, dass jede Zelle eines Organismus die gleichen Gene besitzt, die genetische Information aber nur von einer kleinen Teilmenge von Genen abgelesen wird. Verschiedene Zelltypen benutzen verschiedene Gene. Ein *Gen* ist ein Abschnitt auf der *DNA* (engl.: Desoxyribonucleinacid), welcher die Information für die Synthese eines Proteins trägt [17]. Die DNA befindet sich im Zellkern und tritt meist als Doppelstrang auf. Sie ist ein Polynukleotid, wobei die einzelnen Nukleotide aus einer Base, Desoxyribose und Phosphorsäure bestehen. Die Basen sind Adenin, Thymin, Guanin und Cytosin.

Jeweils zwei Basen bilden ein sogenanntes *Basenpaar* (Adenin - Thymin und Guanin - Cytosin). Die Desoxyribose- und Phosphorsäurereste sind das Rückgrad eines Einzelstranges. Die zwei DNA Stränge sind durch Wasserstoffbrücken zwischen den komplementären Basen verknüpft. Die beiden Einzelstränge befinden sich in entgegengesetzter Polarität zueinander, denn das 3‘-Hydroxylende der einen Kette, steht dem 5‘-Phosphatende der anderen Kette gegenüber. Der Doppelstrang ist zu einer Spirale (*Doppelhelix*) aufgerollt und bildet zusammen mit Proteinen das *Chromatin*, die Bestandteile des *Chromosoms*. Das Ablesen der in der Reihenfolge der Basenpaare enthaltenen Information und ihre Verarbeitung im Zellkern und Zellplasma zu einem Protein ist ein mehrstufiger Prozess. In eukaryotischen Zellen gliedert er sich in *Transkription* und *Translation*.
Während der Transkription werden die Basen eines DNA-Einzelstranges in eine *nukleare mRNA* (engl.: nuclear messenger Ribonucleinacid) kopiert, indem der DNA-Doppelstrang in zwei Einzelstränge aufgebrochen wird und die komplementären Basen zu einem Einzelstrang erzeugt werden. Außerdem erfolgt die Anhängung einer Adenin Sequenz (*polyA*-Strang) an das 3‘ Ende und eines modifizierten Nukleotides an das 5‘ Ende, wodurch die nukleare mRNA stabilisiert wird [17]. Die nukleare mRNA besteht aus zwei Typen von Sequenzen, *Exons* und *Introns*. Im Verlauf der zweiten Stufe, der Verarbeitung von nuklearer mRNA in mRNA, werden die Introns entfernt und die Exons miteinander verbunden (engl.: *Splicing*). Es entsteht die *mRNA* (engl.: messenger RNA), die nur noch die Exons enthält. Ein Exon ist ein Teilabschnitt eines Gens, welcher Information für die Proteinsynthese trägt. Die mRNA wird danach aus dem Zellkern in das Zellplasma überführt, wo die Translation, das Übersetzen der Basensequenz der mRNA in die Aminosäuresequenz eines Proteins, einsetzt. Auf jeder dieser Stufen finden komplexe Regulierungen statt, welche die Synthese von individuellen Proteinen in verschiedenen Zellen bewirken.

Die Regulierung während der Transkription beinhaltet die Entscheidung, welche der Gene in nukleare mRNA überführt werden. Die zwei wichtigsten regulierenden Elemente sind die *cis-Regulatoren* und die *trans-Regulatoren* [17]. Unter cis-Regulatoren (*Promoter*, *Enhancer* und *Silencer*) versteht man spezifische DNA Sequenzen auf einem Chromosom, die die Expression nachfolgender Gene regulieren. Der Promoter liegt typischerweise genau vor dem DNA-Abschnitt, an dem die Transkription beginnt, wohingegen Enhancer und Silencer sich überall auf einem DNA-Doppelstrang befinden können. Alle drei binden trans-Regulatoren. Trans-Regulatoren (*Transkriptionsfaktoren*) sind Moleküle, die auf Gene, auf dem gleichen oder einem anderen Chromosom wirken [17].
Transkription findet statt, indem sich ein Enzym, die RNA-Polymerase, an den Promoter bindet und die nukleare mRNA erzeugt. Die RNA-Polymerase kann sich jedoch nicht von selbst effizient an die Promoterregion binden. Dafür ist ein komplexes Zusammenspiel von trans- und cis-Regulatoren nötig. An den Promoter bindet

sich ein Komplex von Transkriptionsfaktoren, welcher mit der RNA-Polymerase interagiert und somit die nukleare mRNA-Synthese initiiert. Für die effiziente Bildung des Promoterkomplexes benötigen die meisten Gene zusätzlich an die Enhancer gebundene Komplexe von Transkriptionsfaktoren. Zum anderen können an Silencer gebundene Transkriptionsfaktoren die Transkription an dem Promoter verhindern. Die Entscheidung, welches Gen in nukleare mRNA übersetzt wird, hängt also wesentlich davon ab, ob und welche spezifischen Komplexe von Transkriptionsfaktoren an den Enhancern, Silencern und dem Promoter entstehen können [17]. Gene, die zell-spezifische Transkriptionsfaktoren kodieren, haben meist sehr komplexe Enhancer und Promotoren. Deshalb werden sie nur in bestimmten Zellen exprimiert. Es entsteht also ein Mechanismus, der die Synthese von Transkriptionsfaktoren über wiederum andere Transkriptionsfaktoren reguliert (*Genregulationsnetzwerk*).
Neben den regulierenden Prozessen auf der Ebene der Gene, wird die Transkription auch von übergeordneten Strukturen beeinflusst. Die DNA bildet zusammen mit Proteinen Chromosome, die sich teilweise in einem sehr kompakten und schwer zugänglichen Zustand im Zellkern befinden. Die Grundeinheiten der Chromatin-Struktur sind die *Nukleosome*. Sie bestehen aus *Histonen* und aus mit ihnen verbundenen Basenpaaren der DNA. Transkription kann erst beginnen, wenn die Promoterregion und die Enhancerregionen von den Histonen getrennt sind. Andererseits ist es für trans-Regulatoren unmöglich, sich zu binden, außer sie sind fähig diese Verbindung selbst kurzzeitig aufzulösen [17].

Die erfolgreiche Synthese einer nuklearen mRNA garantiert nicht, dass daraus auch ein Protein erzeugt wird. Es muss zunächst die Verarbeitung in mRNA und ihr Transport aus dem Zellkern in das Zellplasma stattfinden. Alternatives Splicing erlaubt es, verschiedene Proteine vom gleichen Gen zu synthetisieren. Die Erkennung der Introns und Exons erfolgt nicht in allen Zellen gleich, sondern selektiv. Alternative Kombinationen von Exons resultieren deshalb in unterschiedlicher mRNA. Eine weitere Regulierung ist gegeben, indem unter Umständen nicht die gesamte mRNA im Zellkern in das Zellplasma transportiert wird [17].

Der letzte Schritt der Proteinsynthese, die Translation der mRNA in ein Polypeptid, erfordert die Bindung von *tRNA* (engl.: transfer RNA) an enzymatisch aktivierter Aminosäuren und die Bildung von mRNA-Ribosomen Komplexen. Es sind meist mehrere Ribosomen durch den mRNA Strang miteinander verknüpft. Sie bilden ein *Polysom*. Ein Ribosom besteht im wesentlichen aus einer kleinen und einer großen Untereinheit. Die kleine ribosomale Untereinheit bindet sich an die mRNA. Eine tRNA muss das Triplet auf der im Ribosom befindlichen mRNA finden, das seiner bestimmten Aminosäure entspricht. Die Aminosäuren werden in der großen ribosomalen Untereinheit durch eine Peptidbindung zu einem Polypeptid verkettet. Indem sich mehrer Ribosomen gleichzeitig an die gleiche mRNA binden können,

erfolgt eine Mehrfachproduktion des gleichen Proteins. Das ist eine mögliche Regulierung der Translation. Eine weitere ist durch die unterschiedliche Lebensdauer der mRNA im Zellplasma gegeben. Kann eine mRNA stabilisiert werden, so dass sie länger aktiv im Zellplasma verbleibt, ermöglicht das eine längere Produktion eines Proteins.
Nachdem ein Protein synthetisiert wurde, nimmt es einen individuellen Platz in einem komplexen System ein. Es wird in Beziehungen mit anderen Proteinen gestellt, welche einen Einfluss darauf haben, ob das Protein aktiv oder inaktiv ist. Die Information ist zwar in der Zelle vorhanden, ihre Nutzung hängt aber von bestimmten zellulären Bedingungen ab [17].

Wie zu erkennen, erfolgt die Regulierung der Genexpression auf vielen komplexen Ebenen. Eine ist die spezifische Transkription, die wesentlich vom Zusammenspiel der Transkriptionsfaktoren beeinflusst wird. Ein Genregulationsnetzwerk definiert diese Interaktion. Ein konkretes Verständnis für ein genetisches Netzwerk würde einen tiefen Einblick in die Entwicklung von verschiedenen Zellen geben. Jedoch muß bedacht werden, dass darin viele Einflußfaktoren unberücksichtigt bleiben.

1.2 Das Stammzellenkonzept und zelluläre Differenzierung bei der Hämatopoese

Billionen von Blutzellen sterben im menschlichen Körper während eines Tages und müssen ersetzt werden. Insgesamt gibt es acht unterschiedliche Zelltypen im Blut, die alle von einem einzigen Zelltyp, den *hämatopoetischen Stammzellen*, durch Differenzierung hervorgehen.

Definition 1.1 (Stammzelle [29]) *Stammzellen heißen Zellen, falls sie undifferenziert sind und fähig sind, sich zu vermehren, ihre Population zu erhalten, differenzierte Tochterzellen zu bilden, die Population der differenzierten Zellen nach starkem Verlust zu regenerieren, flexibel in diesen Funktionen sind und sich in eine spezifische Wachstumsumgebung integrieren.*

Die Existenz von hämatopoetischen Stammzellen wurde 1961 von Till und McCulloch nachgewiesen. Bei einer Zellteilung kann eine Stammzelle Tochterzellen, die wiederum Stammzellen sind, oder Zellen, die sich in ihren intrazellulären Prozessen und Morphologien von einer hämatopoetischen Stammzelle unterscheiden, hervorbringen. Diese können sich wiederum in Tochterzellen mit einer dritten Funktionalität und Morphologie teilen. Ein Modell für die Dynamik der Stammzellentwicklung ([17]) besagt, dass Stammzellen, die sich aktiv im Zellzyklus befinden, meistens in

Tochterzellen, die ebenfalls Stammzellen sind, geteilt werden. Es besteht andererseits die Möglichkeit, dass sich Tochterzellen bilden, die keine Stammzellen mehr sind. Es handelt sich in dem Fall um transiente Zellen in einem Zwischenstadium. Bei ihrer Zellteilung können Zellen im gleichen Zwischenstadium entstehen. Meistens teilen sie sich jedoch in andere transiente Zellen, die sich in einem weiteren Zwischenstadium befinden. Diese können wiederum ihr Kompartment erhalten oder sich in weitere transiente Zellen teilen. Die Wahrscheinlichkeit eine Zelle in einem weiteren Zwischenstadium zu erzeugen, steigt bei jedem neuen Typ von transienten Zellen. Schließlich entstehen so transiente Zellen, deren Tochterzellen ausschließlich differenzierte Zellen sind. Sie haben sich auf eine bestimmte Zelllinie festgelegt. Die Stammzellpopulation produziert Stammzellen, um ihr eigenes Kompartment zu erhalten, oder transiente Zellen, die in die Differenzierung eintreten [17]. Wie diese Entscheidung getroffen wird, ist bis heute weitgehend ungeklärt.

Definition 1.2 (Differenzierung) *Der Prozess, bei dem sich aus Stammzellen morphologisch und funktionell unterschiedliche Tochterzellen entwickeln, wird Differenzierung genannt.*

Differenzierung ist ein gradueller Prozess, der bei sukzessiver Zellteilung auftritt und jede Generation schrittweise mehr differenziert ist. Dabei ist zu beobachten, dass Zellvermehrung offensichtlich vor dem terminalen Stadium der Differenzierung stattfindet und dass sich ausdifferenzierte Zellen kaum teilen [45]. Der Genexpressionszustand in dem sich eine ausdifferenzierte Zelle befindet, ist sehr stabil. Nur bestimmte, relativ extreme, Veränderungen der Zellumgebung, können eine Veränderung des Zelltyps herbeiführen.
Ein Maß für die Vielfalt der Differenzierungsmöglichkeiten einer Zelle ist das *Differenzierungspotential*. Kann sich eine Stammzelle in morphologisch und funktional sehr unterschiedliche Zellen teilen, ist das Differenzierungspotential hoch. *Pluripotente hämatopoetische Stammzellen* sind Stammzellen, die ein sehr hohes Differenzierungspotential besitzen. Stammzellen, die schon auf eine bestimmte Zelllinie festgelegt sind, werden *vorgebundene Stammzellen* oder *multipotente Stammzellen* genannt und haben ein niedrigeres Differenzierungspotential. Während die Differenzierung fortschreitet, werden die Tochterzellen immer mehr an einem bestimmten Differenzierungspfad gebunden und das Differenzierungspotential nimmt zunehmend ab.
Differenzierte Zellen sind insbesondere durch die Proteine, die sie synthetisieren charakterisiert. Die roten Blutzellen produzieren z.B. Hämoglobin und die Hautzellen Keratin. Sie besitzen demnach eine spezialisierte Funktion und Morphologie. Die Erklärung dafür ist die spezifische genetische Expression einer ausdifferenzierten Zelle, die zu der Synthese von zellspezifischen Proteinen führt.

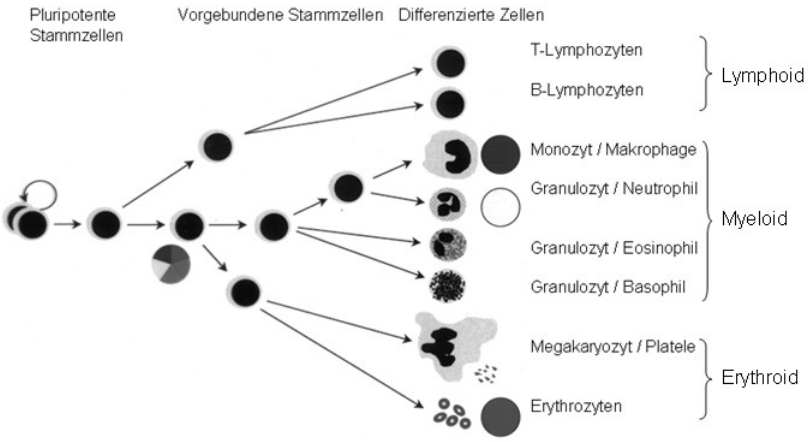


Abbildung 1.1: Mögliche Differenzierungspfade während der Hämatopoese. Es gibt drei Differenzierungswege, lymphoid, myeloid und erythroid. Eine vorgebundene Stammzelle, auch multipotente Stammzelle genannt, enthält das Potential für mehrere Zelltypen (Durch mehrfarbige Kreise dargestellt), Abbildung nach [8].

Bisher hat man nur modellhafte Vorstellungen über den möglichen Differenzierungsablauf von hämatopoetischen Stammzellen. Tiefere Erkenntnisse über die molekularen Mechanismen, die für die Wahl eines bestimmten Pfades verantwortlich sind, fehlen weitgehend. Hämatopoetische Stammzellen differenzieren im wesentlichen auf drei Zelllinien, die *myolide*, *erythroide* und die *lymphoide* Zelllinie. Abbildung 1.1 zeigt ein schematisches Modell für mögliche Differenzierungspfade.
Die Regulierung des Differenzierungsprozesses erfolgt über zwei Einflussfaktoren:

- *Intrazelluläre Signale (Genexpression)*
- *Extrazelluläre Signale (z.B. Wachstumsfaktoren)*

Wachstumsfaktoren sind Proteine, die auf Stammzellen von außen wirken. Jede Zelllinie, in die eine pluripotente Stammzelle differenzieren kann, reagiert auf spezifische Wachstumsfaktoren. Die Art ihrer Wirkung ist vielfältig und hängt davon ab, in welcher Kombination sie auftreten.
In der Vergangenheit wurden zwei kontroverse Theorien diskutiert. Die erste besagt, dass allein äußere Faktoren den Differenzierungspfad einer Zelle bestimmen. Welchen Differenzierungspfad die Nachkommen einer pluripotenten Stammzelle einschlagen, glaubte man, hängt von der Kombination von Wachstumsfaktoren, die auf die Zelle wirken, ab. In der zweiten Theorie wurde behauptet, dass allein der intrazelluläre Zustand den Differenzierungsprozess bestimmt.
Die Interaktion von zellinneren und -äußeren molekularen Faktoren gibt eine bessere Erklärung für die Regulierung des Differenzierungspfades [11]. Denn nach [7]

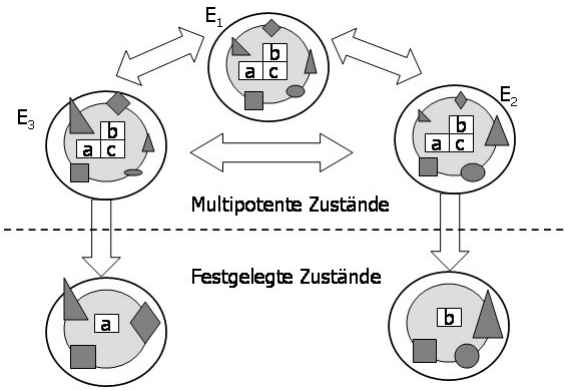


Abbildung 1.2: Modell für die Regulierung des Differenzierungspfades. Abbildung nach [7]. Es sind drei mögliche Expressionszustände von multipotenten Blutstammzellen zu sehen, die durch Fluktuationen der Expressionsstärken ineinander übergehen können.

sind intrazelluläre Mechanismen die primären Steuerungsfaktoren, aber man ist sich auch über den zusätzlichen Einfluss von externen Signalen einig ([7], [11] und [33]). Ein hybrides Modell vereint diese beiden Ansichten. Multipotente hämatopoetische Stammzellen besitzen ein aktives Genregulationsnetzwerk, dessen verschiedene Expressionszustände die Voraussetzung für die Differenzierung auf verschiedenen Pfaden sind [7]. Über die Art der externen Einflüsse gibt es noch keine genauen Kenntnisse. Vermutet wird, dass Wachstumsfaktoren während der Festlegung auf eine bestimmte Zelllinie keine Rolle spielen. Ihnen wird vielmehr eine selektive Funktion zugeschrieben [7]. Sie unterstützen schon auf eine spezifische Zelllinie festgelegte Zellen in ihrer Vermehrung und Entwicklung. Jedoch kann ein möglicher Einfluss von Wachstumsfaktoren nicht völlig ausgeschlossen werden, denn die Experimente lassen nur den Schluß zu, dass auch ohne zelllinien-spezifische Wachstumsfaktoren die Festlegung auf eine Zelllinie stattfindet. Das ist aber noch kein Beweis dafür, dass Wachstumsfaktoren für diesen Prozess völlig überflüssig sind ([11], [33] und [7]).
In [7] wurde ein Modell für die Regulierung des Differenzierungspfades von multipotenten hämatopoetischen Stammzellen erläutert (Abb. 1.2). Die Gene des für die Wahl der Zelllinie verantwortlichen Genregulationsnetzwerkes befinden sich in primären Expressionszuständen, welche innerhalb eines Intervalls fluktuieren. Je nach individueller Expressionsstärke werden verschiedene Proteine unterschiedlich stark synthetisiert. Kleine geometrische Symbole in Abbildung 1.2 bedeuten eine geringe Konzentration des entsprechenden Proteins. Große geometrische Symbole repräsentieren dagegen eine hohe Konzentration. Übersteigt die Expressionsstärke von zelllinien-spezifischen Genen ein bestimmtes Level, resultiert daraus ein auf eine

Zelllinie festgelegter Expressionszustand. Die für die gewählte Zelllinie spezifischen Gene bleiben aktiv und die nicht-spezifischen Gene werden inaktiv [7]. Die Fluktuation der Expressionsstärken wird wesentlich von der gegenseitigen Regulierung der Aktivatoren und Repressoren erklärt. Ein weiterer Einflussfaktor kann das Wirken von bisher noch unbekannten externen Proteinen auf die Transkriptionsfaktoren sein.
Normalerweise behalten interne und externe Einflüsse ihr Gleichgewicht bei. Durch spontane Transkription eines anderen Aktivator- oder Repressor-Proteins oder durch veränderte Umweltbedingungen (z.B. hoher Blutverlust) kann das Gleichgewicht gestört werden. Eine mögliche Folge einer solchen Störung ist die Überexpression von linien-spezifischen Genen, abhängend von dem spezifischen Expressionszustand der multipotenten Zelle. (In Abbildung 1.2 sind das die Zustände E_2 oder E_3. Befindet sich die Zelle gerade im Zustand E_1, bleibt die Störung ohne Wirkung, da dort alle Zustände unterexprimiert sind.) Das hybride Modell beinhaltet: Nur wenn sich die Zelle in einem bestimmten Expressionszustand befindet und bestimmte externe und interne Signale im gleichen Moment auf das Genregulationsnetzwerk wirken, kann die Expressionsstärke von zelllinien-spezifischen Genen ein Level übersteigen und es erfolgt die Festlegung auf einen bestimmten Differenzierungspfad. Die festgelegten Zellen werden dann von Wachstumsfaktoren in ihrer Vermehrung und Entwicklung gefördert.
Die Wahl des Differenzierungspfades hängt also primär von den regulierenden Interaktionen der zelllinien-spezifischen Transkriptionsfaktoren ab [7]. Eine Hypothese für das Genregulationsnetzwerk in multipotenten hämatopoetischen Stammzellen ist in Abbildung 1.3 zu sehen. Es umfaßt im Kern vier Gene, deren Produkte die Transkriptionsfaktoren GATA-1, GATA-2, PU.1 und SCL sind. In multipotenten Blutstammzellen liegen sie exprimiert vor und während der Wahl eines Differenzierungspfades verändert sich das Genexpressionsmuster dahingehend, dass die linienspezifischen Gene überexprimiert und alle anderen gehemmt werden. Eine Überexpression von PU.1 unterstützt die Wahl einer multipotenten Blutstammzelle für den myeloiden Pfad [27] und eine Überexpression von SCL blockiert genau diesen Weg und begünstigt die erythroide Differenzierung [8]. Einmal auf dem myoeloiden Pfad, entscheidet die Expression von PU.1 zwischen dem granulozyten oder makrophagen Differenzierungspfad. Bleibt PU.1 überexprimiert, fördert dies die makrophage Zelllinie gegenüber der granulozyten [30] und eine Hemmung von GATA-1 [27]. GATA-1 und GATA-2 sind in ihrem Verhalten weitgehend redundant und deshalb ist es schwierig, ihnen spezifische Auswirkungen auf die Wahl des Differenzierungspfades zuzuordnen. Eine Überexpression von GATA-1 führt aber dazu, dass sich eine Stammzelle für den erythroiden Differenzierungspfad entscheidet [27].

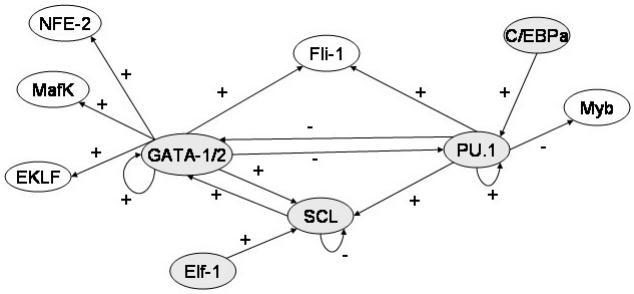


Abbildung 1.3: Hypothese für das Genregulationsnetzwerk in multipotenten Blutstammzellen. Es sind Transkriptionsfaktoren und ihre Interaktionen dargestellt. Transkriptionsfaktoren GATA-1 und GATA-2 wurden zu GATA-1/2 zusammengefaßt, da sie bisher ähnliches Verhalten aufwiesen. Schattierte Elemente bilden das Kernnetzwerk. Die Hypothese ist das Ergebnis einer Literaturrecherche in [8]. + bedeutet aktivierende Interaktion und − hemmende Interaktion. Es ist davon auszugehen, dass diese Hypothese kein vollständiges Bild liefert, sondern vielmehr der Einfluss von bisher unbekannten Transkriptionsfaktoren sehr wahrscheinlich ist.

1.3 Expressionsanalyse auf Basis von Einzelzellen

Eine Herangehensweise zur Konstruktion eines Genregulationsnetzwerkes ist die Gewinnung der Information für die Topologie und Regeln aus Genexpressionsdaten. Genexpresssionsdaten werden von Zellpopulationen (z.B. Microarray-Experimente) oder von einzelnen Zellen (z.B. *polyA-PCR*) erhalten. Damit ein Modell für genetische Netzwerke rekonstruiert werden kann, ist es notwendig das Expressionsmuster an mindestens zwei Zeitpunkten zu charakterisieren.
Insofern sich das Genregulationsnetzwerk synchron verhält und die Zellen sich nicht in ihrem Genregulationsnetzwerk unterscheiden, kann das dynamische Verhalten mit Populationsdaten zuverlässig charakterisiert werden. Verhalten sich andererseits die Genregulationsnetzwerke einer Population asynchron, ist dies nicht mehr möglich. Denn das Ergebnis der Expressionsanalyse einer Zellpopulation ist der Mittelwert über alle verschiedenen Expressionszustände jeder einzelnen Zelle. Sind z.B. zwei Zellen mit einem identischen Genregulationsnetzwerk gegeben und angenommen ein Gen A wird in der ersten Zelle aktiviert und in der zweiten Zelle gehemmt, ist es unmöglich mit einer Expressionsanalyse auf Basis von beiden Zellen diesen Unterschied zu erkennen (Abb. 1.4). Eine weitere Ungenauigkeit ergibt sich, falls auf dem Genexpressionspfad in der ersten Zelle Gen A kurz vor Gen B aktiviert wird und in einer anderen Zelle erfolgt die Aktivierung von B kurz vor A (Abb. 1.5). Dann zeigt der gemittelte Expressionszustand die gleiche Zeitabhängigkeit für beide Gene [43]. M. Cross [private Mitteilung] nimmt an, dass sich das Genregulationsnetzwerk in multipotenten Blutstammzellen asynchron verhält. Deshalb ist eine Populations-

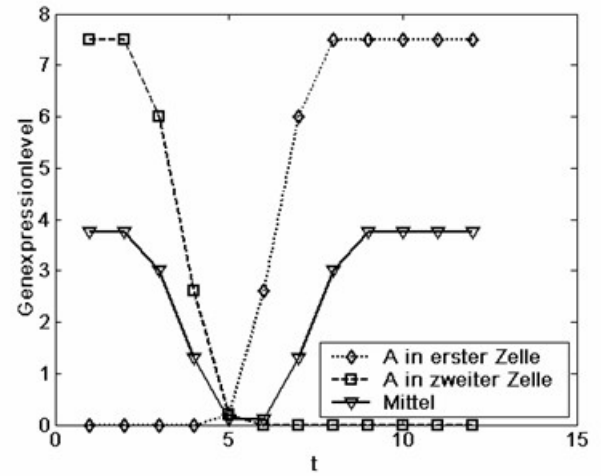


Abbildung 1.4: Gene A und B werden in zwei Zellen unterschiedlich reguliert. Im Mittel ist dies nicht mehr zu erkennen.

analyse nicht angebracht und es sollte das Expressionsmuster der einzelnen Zellen charakerisiert werden.
Die getrennte Charakterisierung der Expression einzelner Zellen hat den Vorteil, dass anschließend die Expression jeder einzelnen Zelle individuell bekannt ist. Dadurch sind Unterschiede in dem Genexpressionsmuster für unterschiedliche Zellen derselben Zellpopulation identifizierbar.
Eine für Einzelzellanalysen geeignete experimentelle Technik ist *polyA-PCR* (engl.: Polyadenylated - Polymerase Chain Reaction) [4]. Sie ist eine Variante von *RT-PCR* (engl.: Reverse Transcription - PCR). PCR ist eine Methode, um viele Kopien eines bestimmten DNA Fragmentes herzustellen [17]. Bei Expressionsanalysen soll jedoch die mRNA Konzentration gemessen werden. Bevor PCR angewendet werden kann, müssen deshalb aus mRNA mit Reverser Transkription (RT) *cDNA*-Einzelstränge (engl.: copied DNA) gewonnen werden. RT-PCR wird benutzt um festzustellen, ob aktiv mRNA produziert wird, also ein spezifisches Gen abgelesen wird. PolyA-PCR ist eine Methode, die auch bei geringem Ausgangsmaterial eine große Menge an kopierten cDNA Fragmenten erzeugt und deshalb für die Expressionsanalyse auf Basis von Einzellzellen genutzt wird.
Bei polyA-PCR [4], werden zuerst kurze spezifische Sequenzen (3-500 Basenpaare) am 3‘ Ende von allen mRNA Einzelsträngen in der Zelle zu cDNA Einzelstränge kopiert. Jeder cDNA Strang wird um eine kurze Adenin-Sequenz verlängert. Danach findet eine globale PCR statt, bei der alle cDNA Einzelstränge amplifiziert werden. Dies erfolgt, indem mit Hilfe von DNA Polymerase die Einzelstränge in Doppelstränge, durch Erzeugung von komplementären cDNA-Strängen, transformiert

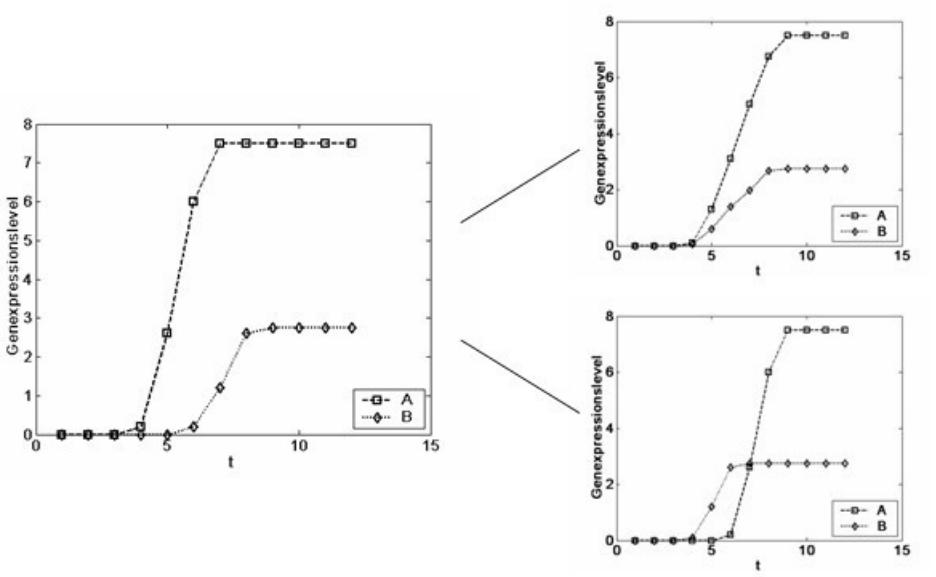


Abbildung 1.5: Gene *A* und *B* haben in zwei Zellen verschiedene Zeitabhängigkeiten, im Mittel jedoch die gleiche Zeitabhängigkeit. Beispiel nach [43].

werden. Die cDNA-Doppelstränge werden wieder getrennt und Thymin Sequenzen binden sich an den Adenin Strang am 3‘ Ende und initiieren die Bildung eines neuen komplementären cDNA Stranges. Nach einigen Durchläufen ist genügend cDNA in der Zelle vorhanden, um die Expressionsstärke eines spezifischen Gens messen zu können. Dafür wird entweder eine spezifische PCR oder *Hybridisierung* durchgeführt.
Bei spezifischer PCR werden spezifische Oligonukleotid Primer hinzugefügt. Sie sind komplementär zu einem charakteristischen Teil der spezifischen cDNA, die verstärkt werden soll. Erkennen die Oligonukleotide diese charakteristischen Sequenzen in dem korrespondierenden cDNA Einzelstrang, war die gesuchte mRNA in der Zelle vorhanden. Der cDNA-Einzelstrang wird kopiert und ergibt somit einen weiteren cDNA Einzelstrang, der für weitere spezifische PCR Durchläufe zur Verfügung steht.
Hybridisierung ermittelt anhand von spezifischen Genproben die Konzentrationsstärke der entsprechenden cDNA, indem sich die Genproben an die komplementären cDNA Einzelstränge binden. Danach ist die Konzentration der cDNA Doppelstränge durch das Auftragen auf Nitrozellulose-Papier ermittelbar.
Die Fehlerrate bei der Expressionsanalyse von einzelnen Zellen liegt bei ca. 5% [M. Cross, private Mitteilung].
Die auf diese Art und Weise verstärkte Konzentration von spezifischer mRNA wird gemessen und gibt die Expressionsstärke des korrespondierenden Gens an. Eine ho-

he mRNA Konzentration bedeutet eine hohe Aktivität des Gens, also eine hohe Expressionsstärke. Geringe mRNA Konzentration ist dagegen ein Beleg für wenig aktive Expression.
Der wesentliche Nachteil dieser Prozedur liegt darin, dass die Zelle verbraucht wird. Es ist unmöglich, die genetische Expression einer Zelle zweimal zu messen. Aber genau das wäre für die Analyse der Dynamik des Genregulationsnetzwerkes wünschenswert. In [8] wird zu diesem Problem eine Lösung vorgeschlagen. Könnte man den Expressionszustand der interessierenden Gene gezielt manipulieren, wäre die Expression dieser Gene zum Zeitpunkt der Manipulation bekannt und PolyA-PCR würde folglich den Expressionszustand der Zelle zum späteren Zeitpunkt messen.
Für die Überexpression von spezifischen Genen werden Plasmide konstruiert, die zu diesen Genen korrespondierende DNA-Abschnitte enthalten. Wird ein Plasmid in eine Zelle transferiert, werden die darauf befindlichen Gene aktiv. Es enthält außerdem ein weiteres Gen, welches bei Aktivierung des spezifischen Gens der Zelle einen grünen Fluoreszenzfarbstoff synthetisiert. In Zellen mit diesem Farbstoff, sind die spezifischen Gene zu 99% überexprimiert.
Die Hemmung eines spezifischen Gens wird durch die Zugabe von inhibitorischen Genen erreicht [19]. Diese können in einem Plasmid integriert in die Zelle eingeführt werden. Daraufhin erfolgt eine Transkription von störenden RNA Sequenzen. Diese Sequenzen heißen siRNAs (engl.: small interfering RNAs) und binden sich als komplementäre Sequenzen an die mRNA des zu hemmenden Gens. Die Translation der mRNA wird somit verhindert [19]. Die Hemmungseffizienz hängt von der Wahl der inhibitorischen RNA Sequenz ab. Eine Verringerung der aktiven mRNA Konzentration um 90% sollte ausreichen, um die regulierende Aktivität des Gens effektiv auszuschalten [M. Cross, private Mittelung]. Die Trennung von Zellen mit gehemmtem Gen von Zellen mit aktivem Gen erfolgt auch über ein grün leuchtendes Protein. Bei einer gut gewählten inhibitorischen RNA Sequenz beträgt die Fehlerrate ca. 1% [M. Cross, private Mitteilung].
Eine gezielte Manipulation ist nur für ein oder maximal zwei Gene gleichzeitig möglich. Die genetische Ausgangsexpression ist somit nur unvollständig bekannt!
Der genaue Ablauf eines Experimentes versteht sich wie folgt. Es wird ein biologisches Modell, die *FDCPmix* Zellen, für multipotente Blutstammzellen ausgewählt. Das sind Zellen, die sich in alle myeloide Differenzierungspfade entwickeln können. FDCPmix Zellen differenzieren über eine Periode von $7-9$ Tagen. In den ersten $5-6$ Tagen vermehren sich die Zellen und ca. am 3. Tag findet bei ihnen die Entscheidung für eine Zelllinie statt. Eine FDCPmix Kolonie wird in drei verschiedene Kulturen aufgeteilt, die jeweils verschiedenen Medien ausgesetzt werden. Ein Medium unterstützt die Vermehrung, ein anderes die Entscheidung für den erythroiden/myeloiden Differenzierungspfad und das letzte die Entscheidung für den granulozyten/makrophagen Differenzierungspfad. Für die Identifizierung des Netz-

Eigenschaft	Boolesche Netze	DGL	Bayessche Netze	Dyn. Bayessche Netze
Variablen	dis	kont	dis/kont	dis/kont
Relationen	det	det	stoch	stoch
Zeitl. Verhalten	dyn	dyn	stat	dyn

Tabelle 1.1: Klassifikation möglicher Modelle für Genregulationsnetzwerke. dis=diskret, kont=kontinuierlich, det=deterministisch, stoch=stochastisch, dyn=dynamisch, stat=statisch. Die Relationen zwischen den Elementen in Bayesschen Netzen und Dynamisch Bayesschen Netzen sind stochastischer Art, womit auch deterministische Beziehungen des tatsächlichen Modells approximiert werden können. (Tabelle nach [22])

werkes von multipotenten FDCPmix Zellen werden aus den sich differenzierenden Kulturen Populationen am ersten und dritten Tag entnommen. In jede Zelle dieser Populationen wird ein Plasmid transferiert. Zellen mit gewünschter Expression der spezifischen Gene werden von den anderen getrennt. Ein Teil von ihnen unterliegt der weiteren Beobachtung, in welche Zelllinie sie differenzieren oder ob sie womöglich sterben. Der Genexpressionszustand der restlichen Zellen mit gewünschter Expression wird mit polyA-PCR zum späteren Zeitpunkt gemessen. Es sollte sicher gestellt sein, dass bis dahin das genetische Netzwerk auf die durchgeführte Manipulation reagierte.

1.4 Modellierung und Reverse Engineering

Es gibt eine Vielfalt von Modellierungsmethoden und entsprechende Reverse Engineering Strategien für genetische Netzwerke. Die Modelle unterscheiden sich vor allem darin, ob die verschiedenen Zustände der Gene diskret oder stetig abgebildet sind und ob die regulierenden Beziehungen zwischen den Genen deterministisch oder stochastisch modelliert werden. Eine zusammenfassende Klassifikation der wichtigsten Modelle ist in Tabelle 1.1 in Anlehnung an [22] dargestellt. Das Entwerfen von Modellen für eine Domäne erfordert fundierte Kenntnisse eines Experten. Unter Umständen ist die Domäne so komplex, dass auch ein Experte nicht fähig ist, diese Aufgabe zu lösen. In solchen Fällen sind Algorithmen notwendig, die die Modellstruktur und die Modellparameter aus Trainingsdaten ableiten können. Solche Algorithmen sind *Reverse Engineering Algorithmen*.
Erste Modellierungsansätze für Genregulationsnetzwerke betrafen *Boolesche Netze* ([24]) und *Differentialgleichungen* (DGL) ([18], [32], [44]).

Indem Boolesche Netze vereinfachende Annahmen über die Struktur und die Dynamik des biologischen Systems machen, können regulierende Prozesse effizient analysiert werden. Von Stuart Kaufman [24] wurden sie erstmals auf biologische Themen

angewendet. In [38] wird ein Boolesches Modell für den Differenzierungsprozeß von hämatopoetischen Stammzellen beschrieben. Attraktoren modellieren Stammzellen, da sie ein wiederholendes Zustandsmuster besitzen. Die abstrakte Modellierungsebene von Booleschen Netzen hat den Vorteil, dass schon mit geringen Rechenleistungen die Dynamik von komplexen Systemen auf dem Computer simuliert werden kann. Sie besitzen wenige freie Parameter und eignen sich insbesondere für die Modellierung von einfachen Relationen. Dem stehen jedoch entscheidende Nachteile bei Folgerungen aus dem Modell in die reale Welt gegenüber. Die diskrete Idealisierung der Genproduktkonzentration entspricht nicht der biologischen Realität und es gehen Informationen, die in den Daten enthalten sind, verloren. Außerdem sind kleine Änderungen in der Genproduktkonzentration nicht darstellbar. So ist z.B. die Amplifizierung und die Addition und Subtraktion von Signalen nicht möglich.
Reverse Engineering Algorithmen für Boolesche Netze wurden erst viel später entwickelt. Das Ergebnis waren Algorithmen, wie *REVEAL* [28], *BOOL-1* und *BOOL-2* ([1], [2]). REVEAL ist ein Spezialfall von allgemeinen Methoden für das Lernen eines Modells von Daten (sog. *Trainingsdaten*) aus einer beliebigen Quelle. Anhand von Maßen aus der Informationstheorie wird die Stärke der Korrelationen zwischen den zu untersuchenden Variablen bewertet und somit Abhängigkeiten erkannt. BOOL-1 und BOOL-2 versuchen aus allen Booleschen Funktionen die herauszufinden, die mit den gegebenen Trainingsdaten konsistent sind. Bei BOOL-1 werden nur die Booleschen Funktionen, die mit allen Zustandsübergangspaaren konsistent sind, als relevante Funktion für bestimmte Variablen identifiziert. BOOL-2 fordert nur eine Konsistenz für eine vorher festgelegte Anzahl von Zustandsübergangspaaren. Dadurch werden auch bei verrauschten Daten noch Funktionen identifiziert. Der Nachteil von BOOL-1 und BOOL-2 gegenüber REVEAL ist, dass der benutzte Ansatz eine Suche über den gesamten Raum der Booleschen Funktionen erfordert, was bei großen Netzen nicht mehr möglich ist. Beide Methoden wurden bisher ausschließlich für vollständig gegebene Trainingsdaten untersucht. Im Abschnitt 4.2 wird untersucht, ob sich REVEAL für unvollständige Daten eignet.

Im Gegensatz zu Booleschen Netzen können Differentialgleichungen Konzentrationsschwankungen der Genprodukte mit kontinuierlichen Werten beschreiben. Für die Modellierung von Genregulationsnetzwerken mit DGL's gibt es mehrere Varianten [22]. Sie haben gemeinsam, dass die Expressionslevel der Input-Gene additiv auf die Änderungsrate der Expressionsstärke des Output-Gens wirken und die Stärke des Einflusses durch eine Gewichtsmatrix gesteuert wird. Den einfachsten Fall stellen die *Linearen Modelle* dar. Bei ihnen wird die Änderung der Produktkonzentration eines Gens zu einer bestimmten Zeit durch lineare Differentialgleichungen beschrieben. Eine andere Modellklasse bilden die *Nicht-Linearen Modelle*, die die Änderung der Produktkonzentration z.B. durch eine sigmoide Funktion darstellen ([32], [18], [44]). In [44] wird ein Reverse Engineering Algorithmus (*Reverse Engineering of Matrices*

(REM)) für nicht-lineare Modelle vorgestellt. Die Dynamik des Systems wird durch Differenzengleichungen beschrieben. Das Ziel des Reverse Engineering Algorithmus ist es, eine entsprechende Gewichtsmatrix aus den gegebenen Trainingsdaten zur rekonstruieren. Reverse Engineering wird auf ein algebraisches Problem zurückgeführt, indem jede Zeile der Gewichtsmatrix die Lösung eines linearen Gleichungssystems ist.

Spätere Modellierungsansätze umfaßten *Bayessche Netze* ([14]) und *Dynamisch Bayessche Netze* (DBN) ([35], [37]). In [14] wurden erstmals Bayessche Netze verwendet, um Korrelationen zwischen Genen aus Genexpressionsdaten zu modellieren. Ihre Methode erlaubt jedoch keine Aussagen darüber, wie sich die Gene gegenseitig abhängig von der Zeit regulieren. Dafür sind Dynamisch Bayessche Netze zu verwenden, die erstmals in [37] auf Zeitserien von Genexpressionsdaten angewendet wurden.
Bayessche Netze und Dynamisch Bayessche Netze sind eine Technik aus der Künstlichen Intelligenz für die Modellierung von Unsicherheit über eine Domäne.
Die die Genregulation steuernden Prozesse sind noch nicht vollständig aufgedeckt und es sind nicht alle bereits bekannten Einflussfaktoren immer messbar. Deshalb erscheinen die in der Genregulation stattfindenden Prozesse stochastisch und ein stochastisches Modell wie Bayessche Netze eignet sich besser als deterministische Modelle. Es werden Vermutungen eines Beobachters über eine Domäne modelliert. Informationen über einige Komponenten dieser Domäne werden zur Aktualisierung von Vermutungen über andere Komponenten herangezogen. Der Beobachter oder Agent kann dann Entscheidungen treffen und auf den neuen Wissensstand reagieren.
Biologische Experimente zur Genexpression unterliegen Fehlern. Deshalb sind die Genexpressionsdaten verrauscht. Dadurch können Inkonsistenzen in den Daten enthalten sein und es gibt unter Umständen kein Modell, welches die Inkonsistenzen modellieren kann. Lernalgorithmen für Bayessche Netze und Dynamische Bayessche Netze suchen daher nach dem wahrscheinlichsten Modell gegeben der Daten [35].
Dynamische Bayessche Netze erlauben zusätzlich die Modellierung der Systemevolution. Deshalb sind sie für Genregulationsnetzwerke interessanter als ihre Verallgemeinerung, die Bayesschen Netze.
Neben Rauschen in den Daten, erschweren fehlende Beobachtungen das Lernen eines guten Modells. Für Bayessche Netze und Dynamisch Bayessche Netze stehen Reverse Engineering Algorithmen zur Verfügung, die in einem Toleranzbereich auch fehlende Werte gut verarbeiten können ([12], [13]). Gibt es jedoch zu viele fehlende Beobachtungen, haben auch diese Algorithmen ihre Schwierigkeiten. Die Lernmethoden für Bayessche Netze verwenden verschiedene Heuristiken, um ein optimales Modell zu finden. Sie basieren auf gut fundierten Methoden aus der Statistik und des Maschinellen Lernens. Bei vielen Sachverhalten haben die Beobachter bzw. Experten

Vermutungen darüber, welche Komponenten auf welche Art und Weise miteinander in Beziehung stehen. Werkzeuge, die es ermöglichen solches Vorwissen in den Lernprozess mit einfließen zu lassen, sind sehr wertvoll. Bei Bayesschen Netzen bieten Bayessche Lernmethoden die Möglichkeit eine prior-Verteilung zu definieren.

Neben den hier vorgestellten Modellierungsmethoden sind weitere *Qualitative Differentialgleichungen* und *Stochastische Mastergleichungen* [22].

1.5 Problemstellung und Zielsetzung

Ein biologisches Experiment zur Identifizierung der genetischen Expression im Sinne dieser Arbeit ist:

Definition 1.3 ((biologisches) Experiment) *Ein (biologisches) Experiment beinhaltet die Störung der Expression von einem, maximal von zwei Genen, einer Zelle und die Charakterisierung des Expressionszustandes der interessierenden Gene in dieser Zelle, nachdem sich das Genregulationsnetzwerk einmal aktualisierte.*

Das Ergebnis eines Experimentes ist ein Zustandsübergangspaar. Es besteht aus einem Anfangszustand und einem Folgezustand. Der Anfangszustand ist nur teilweise bekannt, wohingegen der Folgezustand vollständig identifizierbar ist. Dies entspricht genau der experimentellen Datensituation aus Abschnitt 1.3. Die Daten mit denen ein Modell gelernt wird, sind die sogenannten *Trainingsdaten*. Sie sind in den Analysen dieser Arbeit unvollständig gegeben und haben konkret folgende Form:

Definition 1.4 (unvollständige Trainingsdaten) *Ein unvollständiges Trainingsdatum besteht aus einem unvollständig definierten Anfangszustand und einem vollständig definierten unmittelbaren Folgezustand des genetischen Netzwerkes. Der unvollständig definierte Anfangszustand enthält mindestens den Expressionszustand von einem Gen und maximal die Expressionszustände von zwei Genen. Alle anderen Expressionszustände sind im Anfangszustand unbekannt.*

Das Ziel der Arbeit ist es, eine optimale Strategie für die Durchführung solcher Experimente zu erarbeiten, so dass ausreichend Information für die Identifizierung des Genregulationsnetzwerkes vorhanden ist. Von besonderem Interesse sind Angaben über die maximal notwendige Anzahl von Experimenten und wieviel Gene manipuliert werden müssen.
Dies wird erreicht, indem mit Hilfe eines geeigneten mathematischen Modells für Genregulationsnetzwerke Experimente in-silico simuliert werden. Denn reale biologische Daten stehen bis dato noch nicht zur Verfügung. Die Untersuchungen dienen

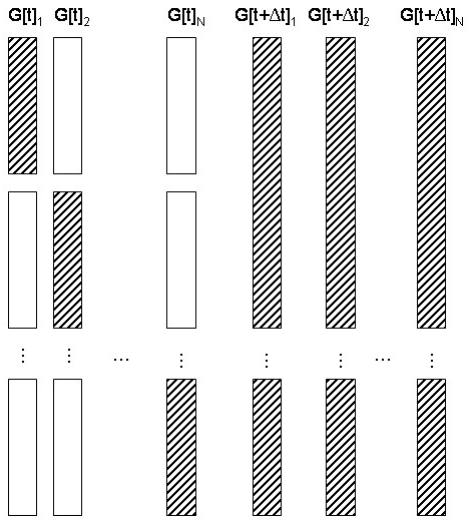


Abbildung 1.6: Trainingsdaten, falls in einem Experiment die Anfangszustände nicht gemeinsam bekannt sind. Δt ist in der vorliegenden Arbeit gleich 1.

der theoretischen Analyse und Klärung der Frage, ob es möglich ist, anhand solcher unvollständigen Trainingsdaten ein gutes Modell zu lernen. Auf die künstlich erzeugten Trainingsdaten wird ein Reverse Engineering Algorithmus zur Identifizierung des Netzwerkes angewendet. Die gelernten Modelle werden mit dem originalen künstlichen Genregulationsnetzwerk verglichen, um den angewendeten Algorithmus zu evaluieren.
Die einfachsten Beziehungen zwischen auftretenden Variablen sind deterministischer Art. Innerhalb dieser Arbeit erfolgt eine Fokussierung auf simulierte Daten, die von einem deterministischen Modell, hier ein Boolesches Netz, erzeugt werden. Das gelernte Modell muss dagegen keine deterministischen Beziehungen widerspiegeln, denn durch Rauschen in den Daten oder infolge fehlender Daten sind deterministische Beziehungen oft nicht erkennbar. Ziel ist es vielmehr das wahrscheinlichste Modell aus den gegeben Daten zu lernen. Es werden Methoden analysiert, mit denen ein Dynamisch Bayessches Netz aus unvollständig gegebenen Trainingsdaten gelernt werden kann.
In den resultierenden Reverse Engineering Algorithmus muß mögliches Vorwissen der Experten leicht integrierbar sein.

Kapitel 2

Entropie und Mutual Information

Um die Struktur für ein Modell aus Trainingsdaten zu identifizieren, kann auf das Maß *Mutual Information* (MI) zurückgegriffen werden. Die *Mutual Information* ist ein zentraler Begriff dieser Arbeit. Dieses Maß der Informationstheorie gibt an, wie unabhängig die Zufallsvariablen X und Y sind.
Claude Shannon gilt mit seinem Artikel [41] als der Begründer der Informationstheorie. Er führte diese Wissenschaft als die mathematische Theorie für Kommunikation ein. Das fundamentale Problem der Kommunikation ist, dass eine Nachricht, die von einem Sender an einen Empfänger über einen gestörten Kanal gesendet wird, vom Empfänger exakt oder annähernd exakt reproduziert werden muss. Der Sender wird auch als Informationsquelle bezeichnet und erzeugt eine diskrete bzw. kontinuierliche Sequenz von Symbolen oder Ereignissen, wobei jedes Symbol oder Ereignis x_i mit einer bestimmten Wahrscheinlichkeit p_i auftritt. Mathematische Modelle für solche Systeme bilden stochastische Prozesse. Quellen sind dabei mit diskreten bzw. stetigen Zufallsvariablen beschrieben. Eine diskrete Informationsquelle fasst man als einen diskreten Markov Prozess auf.
Diese Vorstellung von Informationsübertragung wird, ein wenig abgeändert, auf Genregulationsnetzwerke angewendet. Gene, die über ihre Produkte wiederum die Expressionsstärke anderer Gene beeinflussen, fungieren als *Sender*. Diskrete oder stetige Sequenzen von Symbolen beschreiben die fluktuierenden Konzentrationen der Genprodukte in der Zelle. Unterschiedliche Symbole stehen für verschiedene Konzentrationen. Die in Experimenten gemessene Sequenz von Konzentrationsschwankungen ist nichts anderes als eine Nachricht. Der Empfänger der Nachricht muss diese hier jedoch nicht reproduzieren, sondern sein Verhalten steht vielmehr unter dem Einfluss dieser Nachricht. Gene, deren Expressionsstärke durch die Konzentrationsunterschiede beeinflusst werden, sind die *Empfänger*. Die Information, die von einem Sender-Gen auf ein Empfänger-Gen übertragen wird, kann mit Gesetzen der Informationstheorie analysiert werden und so die Struktur des Genregulationsnetz-

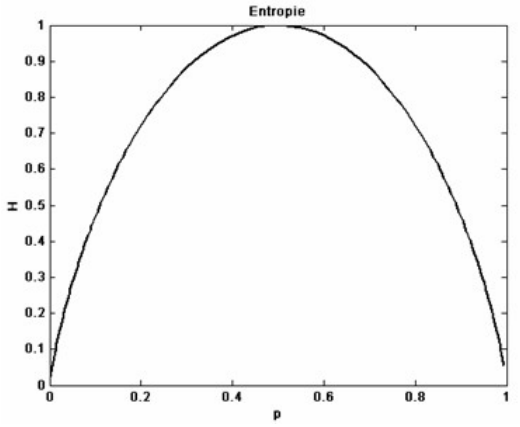


Abbildung 2.1: Entropie für eine Quelle, die eine Sequenz aus zwei Symbolen mit Wahrscheinlichkeiten p bzw. $1-p$ erzeugt.

werkes aus Trainingsdaten identifizieren.
Das zentrale Maß der Informationstheorie ist die *Entropie*. Sie gibt die Unsicherheit über die generierte Sequenz von Symbolen einer Quelle X an. Die Entropie einer Zufallsvariablen X wird folgendermaßen definiert:

Definition 2.1 (Entropie) $H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$.

p_i entspricht der Wahrscheinlichkeit ein Symbol x_i innerhalb der von X generierten Sequenz zu beobachten. n bezeichnet die Anzahl unterschiedlicher Symbole, die auftreten können. Eine Sequenz ist umso informationsreicher, je unsicherer man über deren syntaktischen Inhalt ist. Der unsicherste Fall ist gegeben, falls die Symbole der Sequenz gleichverteilt sind, denn es kann nicht vorhergesagt werden, welches Symbol als nächstes von der Quelle generiert wird. In solchen Fällen ist die Entropie maximal.
Werden dagegen die Wahrscheinlichkeiten verschoben, so dass ein Symbol oder Ereignis wahrscheinlicher als die anderen Symbole ist, verringert sich $H(X)$. Im Falle, dass sicher vorhergesagt werde kann, welches Symbol als nächstes von der Quelle generiert wird, ist $H(X) = 0$. Es herrscht minimale Unsicherheit über X. Dies ist der Fall, sobald ein Symbol j sicher in der Sequenz auftritt ($p_j = 1$) und die restlichen Symbole nie auftreten ($p_i = 0 \ \forall i \neq j$).
Zieht man N Zufallsvariablen $X_1, ..., X_N$ mit den Wahrscheinlichkeiten $P(x_1, \ldots, x_N)$ ihrer vereinigten Ereignisse $\bigcup_{i=1}^{N} x_i$ heran, entspricht die Entropie der Vereinigung:

$$H(X_1, \ldots, X_N) = -\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_N=1}^{n_N} P(x_1, \ldots, x_N) \log_2 P(x_1, \ldots, x_N). \qquad (2.1)$$

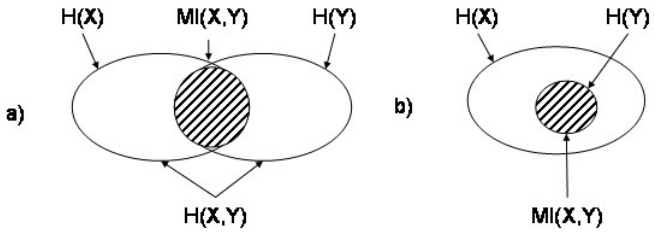


Abbildung 2.2: a) Mengentheoretische Beziehungen zwischen der Entropie einer Zufallsvariablen $H(X)$, der Entropie zweier Zufallsvariablen $H(X,Y)$ und der *Mutual Information* $MI(X,Y)$. b) Mengentheoretische Beziehungen zwischen Entropien und Mutual Information, falls die Zufallsvariablen X und Y immer kovariieren.

Die Zufallsvariable X_i kann n_i verschiedene Zuweisungen annehmen.
Ein anderes wichtiges Entropiemaß ist die Mutual Information. Zwei Zufallsvariablen sind unabhängig, falls $P(X,Y) = P(X) * P(Y)$. Die *Mutual Information* gibt ein Maß für die gegenseitige Unabhängigkeit von X und Y vor und ist definiert durch:

Definition 2.2 (*Mutual Information*) $MI(X,Y) = \sum_i^n \sum_j^m p_{ij} \log_2 \frac{p_{ij}}{p_i p_j}$.

Daraus folgt, dass $MI(X,Y) = 0$ ist, falls zwei Zufallsvariablen unabhängig sind. Mit steigender Abhängigkeit zweier Zufallsvariablen, wird auch die *Mutual Information* größer. Diese ist maximal, falls die beiden Zufallsvariablen immer kovariieren.
Impliziert zum Beispiel die maximale Expression des Gens A, dass Gen B minimal exprimiert wird, liegt eine maximale Kovarianz vor. In diesem Falle ist $P(A = 1 \wedge B = 0) = P(A = 1) = P(B = 0)$ bzw. $P(A = 0 \wedge B = 1) = P(A = 0) = P(B = 1)$ und $P(A = 1 \wedge B = 1) = P(A = 0 \wedge B = 0) = 0$, also ist entweder $p_{ij} = p_i = p_j$ oder $p_{ij} = 0$. Eingesetzt in 2.2 folgt $MI(X,Y) = \sum_i^n \sum_j^m f(i,j)$ mit

$$f(i,j) = \begin{cases} p_j \log_2 \frac{p_i}{p_i p_j} & \text{falls} \quad p_{ij} = p_i = p_j \\ 0 & \text{falls} \quad p_{ij} = 0 \end{cases}$$

([10]), und dies entspricht nichts anderem als

$$MI(X,Y) = -\sum_j^n p_j \log_2 p_j = H(Y). \tag{2.2}$$

Wird danach gefragt, ob X das Element Y vollständig festlegt, reicht es also aus zu testen, ob $MI(X,Y) = H(Y)$ gilt. Dieser Test kann unkompliziert mit Trainingsdaten durchgeführt werden, indem man aus ihnen Schätzungen $\hat{p}$ für die nötigen Wahrscheinlichkeiten gewinnt und die *Mutual Information* mit 2.3 berechnet:

$$MI(X,Y) = H(X) + H(Y) - H(X,Y). \tag{2.3}$$

Kapitel 3

Modelle für genetische Netzwerke

In diesem Kapitel wird zuerst auf Boolesche Netze und auf einen möglichen Reverse Engineering Algorithmus für Boolesche Netze eingegangen.
Danach erfolgt eine Einführung in die Dynamisch Bayesschen Netze und in ihre Reverse Engineering Strategien.

3.1 Boolesche Netze (BN)

Boolesche Netze modellieren die komplexe Dynamik von genetischen Netzen auf einer abstrakten Ebene. Ihr wesentlicher Nachteil ist jedoch, dass kontinuierliche Expressionsänderungen eines Gens nicht darstellbar sind. Unabhängig davon sollen sie in dieser Arbeit als Modell herangezogen werden, um künstliche unvollständige Zustandsübergangsdaten zu erzeugen. Denn die so erhaltenen Trainingsdaten resultieren aus deterministischen Relationen und sind leicht rekonstruierbar, falls vollständige Beobachtungen vorliegen. Außerdem sind mit ihnen Studien über das dynamische Verhalten eines biologischen Systems möglich. Vorwiegend wegen des ersten Punktes soll angenommen werden, dass das genetische Netzwerk in multipotenten Blutstammzellen mit einem Booleschen Netz darstellbar ist.
Boolesche Netze sind folgendermaßen definiert:

Definition 3.1 *Ein **Boolesches Netz** $G = (V, F)$ ist ein gerichteter Graph. Er besteht aus einer Menge $V = v_1, \ldots, v_N$ von Booleschen Variablen (Elemente) und einer Menge $F = f_1, \ldots, f_N$ von Booleschen Funktionen (Kanten). Eine Boolesche Funktion $f_i(v_{i_1}, \ldots, v_{i_k})$ ist der Booleschen Variable v_i zugeordnet und bestimmt deren Zustand zur Zeit $t + 1$ in Abhängigkeit der Zustände von $v_{i_1}, \ldots, v_{i_k}$ zur Zeit t und $v_{i_k} \in \{0, 1\}$.*

Der Zustand von v_i zur Zeit $t+1$ wird auch *Output* und die Zustände von $v_{i_1}, \ldots, v_{i_k}$ zur Zeit t als *Input* bezeichnet. Dementsprechend wird v_i auch *Outputelement* und $v_{i_1}, \ldots, v_{i_k}$ nennt man auch *Inputelemente*. Die Anzahl der Elemente in G wird mit N notiert und die Anzahl der Inputelemente für eine Boolesche Funktion mit k. Die maximal mögliche Anzahl von Inputelementen wird mit K angegeben. Die Booleschen Variablen repräsentieren Genprodukte, z.B. Transkriptionsfaktoren, und ihre Zustände die vorhandenen Konzentrationen der Genprodukte in einer Zelle. Im allgemeinen besteht der diskrete Zustandsraum eines Booleschen Netzes aus zwei Werten, 0 oder 1. Befindet sich eine Boolesche Variable im Zustand 1, dann modelliert sie ein aktives Gen, dessen Genprodukt in der Zelle sehr konzentriert vorkommt. Im Zustand 0 repräsentiert sie ein inaktives Gen, dessen Produkt in der Zelle wenig konzentriert vorkommt bzw. nicht vorhanden ist. Der Zustandsraum eines Booleschen Netzes enthält 2^N globale Zustände, da jedes Element nur zwei Werte annehmen kann. Die Dynamik des Systems wird beschrieben, indem der globale Zustand eines Booleschen Netzes zur Zeit $t+1$ durch den globalen Zustand zur Zeit t und durch Anwendung der Booleschen Funktionen eindeutig bestimmt wird. Boolesche Netze modellieren also deterministische Beziehungen. Die Zustände der Elemente v_i werden synchron aktualisiert und eine Aktualisierung wird *Zustandsübergang* genannt.

Definition 3.2 (Trajektorie) *Eine **Trajektorie** ist eine Sequenz von globalen Zuständen von G, die durch eine zeitliche Folge von Zustandsübergängen verbunden sind (vgl. Abb. 3.2).*

Das dynamische Verhalten eines Booleschen Netzes kann grafisch anhand von *Wiring Diagrammen* dargestellt werden (vgl. Abb. 3.1). Ein Wiring Diagramm definiert die regulierenden Verbindungen zwischen den Elementen. Existiert eine Verbindung zwischen den Elementen v_1 und v_2, dann bestimmt der Zustand von v_1 zur Zeit t den Zustand von v_2 zur Zeit $t+1$.

Definition 3.3 (Attraktor) *Befindet sich eine Trajektorie in einem über die Zeit stabilen Zyklus oder in einem stationären Zustand, dann wird dieser Zyklus **Attraktor** genannt.*

Attraktoren enthalten also wenige Zustände, die ein sich wiederholendes Muster ergeben. Demzufolge werden sie benutzt, um Zellen mit einem stabilen Genexpressionsmuster zu modellieren. Solche Zellen können ausdifferenzierte Zellen, aber auch multipotente Stammzellen sein. Jede Trajektorie wird schließlich in einen Attraktor führen, denn es gibt nur eine endliche Anzahl von möglichen globalen Zuständen. Das System wird also nach einer endlichen Zeit einen globalen Zustand annehmen, den es schon zuvor hatte.

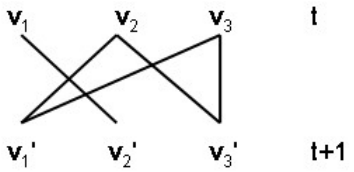


Abbildung 3.1: Wiring-Diagramm für das Boolesche Netz $v_1' = v_2 \wedge v_3$, $v_2' = v_1$, $v_3' = v_2 \vee v_3$.

t	1	2	3	4	5
v_1	1	0	1	0	1
v_2	1	1	0	1	0
v_3	0	1	1	1	1

Abbildung 3.2: Die Trajektorie eines Booleschen Netzes mit den Elementen v_1, v_2 und v_3. Bei $t = 4$ befindet sich das Netz in einem zyklischen Attraktor

In [24] wird argumentiert, dass nur ein Bruchteil der 2^N möglichen Booleschen Funktionen biologisch relevant ist. Solche Funktionen heißen *Canalyzing Funktionen* und werden wie folgt definiert:

Definition 3.4 (Canalyzing Inputelement [24]) *Ein Inputelement für eine Boolesche Funktion ist* ***canalyzing****, falls wenigstens einer seiner Zustände einen Zustand des Outputelementes garantiert, unabhängig von den Zuständen der anderen Inputelemente.*

Definition 3.5 (Canalyzing Funktion [24]) *Ist mindestens ein Inputelement nach Definition 3.4 einer Booleschen Funktion canalyzing, dann ist sie eine* ***Canalyzing Funktion****.*

Beispiele für Canalyzing Funktionen sind die AND, OR und IF Funktionen. Biologisch bedeutet (v_1 AND v_2), dass die Expressionsstärke eines Gens von der Kombination der Transkriptionsfaktoren v_1 und v_2 abhängt. Ist also ein Transkriptionsfaktor nicht an der Promoterregion des Gens gebunden, dann ist das Gen inaktiv, unabhängig davon ob der andere Faktor vorhanden ist. (v_1 OR v_2) beschreibt dagegen zwei mögliche Aktivatoren. Sobald ein Aktivator am Promoter gebunden ist, ist das Gen aktiv. Die Bindung eines Repressors oder Aktivators wird mit (IF v_1) beschrieben.
Ein Grund für die biologische Relevanz von *Canalyzing Funktionen* wird darin gesehen, dass sie eine besondere chemische Einfachheit aufweisen. So ist es z.B. chemisch einfach eine OR Funktion zu realisieren. Denn diese benötigt an einem Protein nur eine Bindungsstelle. Sobald sich eines von zwei möglichen Proteinen an diese Bindungsstelle gekoppelt hat, reagiert das Zielprotein und verändert seine chemischen Eigenschaften. Für die XOR Funktion sind dagegen zwei Bindungsstellen mit einer komplexen Interaktion notwendig [24]. Für die Realisierung der AND Funktion sind zwar auch zwei Bindungsstellen erforderlich, aber ihre Interaktion muss nicht

A	B	C	A'	B'	C'
1	1	1	1	1	1
1	1	0	0	1	1
1	0	1	0	1	1
1	0	0	0	1	0
0	1	1	1	0	1
0	1	0	0	0	1
0	0	1	0	0	1
0	0	0	0	0	0

Globale Zustände einer Zelle i zur Zeit t und t+1. Sie repräsentieren einen Zustandsübergang (Input/Output Paar).

Abbildung 3.3: Zustandsübergangstabelle für das Boolesche Netz $A' = B \wedge C$, $B' = A$, $C' = B \vee C$.

so komplex wie bei XOR sein. Solange eine Bindungsstelle frei ist, werden sich die chemischen Eigenschaften des Proteins nicht verändern.

3.2 Reverse Engineering eines Booleschen Netzes

Der bekannteste Algorithmus für die Identifizierung von Genregulationsnetzwerken aus Zustandsübergangsdaten, unter Betrachtung des Booleschen Modells, ist *REVEAL* (Reverse Engineering Algorithmus) ([28], [42]). Liang et. al. benutzen die Shannon Entropie und das Korrelationsmaß *Mutual Information*, um zu berechnen, wieviel Information einer Kombination von Inputelementen effektiv auf ein Outputelement übertragen wird (vgl. Abb. 2.2). Die experimentelle Bestimmung der Genexpressionsdaten von N Genen erfolgt an zwei aufeinander folgenden Zeitpunkten. Es wird angenommen, dass die Gene über ihre Genprodukte in einem Netzwerk von gegenseitigen Einflüssen stehen. Dadurch verändern sich die Expressionsstärken der Gene. Diese Veränderungen werden in einer *Zustandsübergangstabelle*, welche aus zwei Spalten besteht, zusammengefasst. Die linke Spalte enthält die approximierten Expressionsstärken der Gene zur Zeit t, die rechte die approximierten Expressionsstärken zur Zeit $t+1$. Idealerweise beträgt der Zeitabstand zwischen den Spalten 1. Ansonsten sind die direkten Einflüsse zwischen den Genen nicht bestimmbar. Eine Zeile in einer Zustandsübergangstabelle zeigt den approximierten Zustand einer ganzen Zelle zur Zeit t (Input) und zur Zeit $t+1$ (Output). Jede Zeile bildet somit ein *Input/Output Paar*, was wiederum einen Zustandsübergang repräsentiert (vgl. Abb. 3.3).
Die Argumentation ist, dass der aus den dynamischen Veränderungen des Inputs erhaltene Informationsgewinn auf die Fluktuationen des Outputs übertragen wird. Ist die Mutual Information zwischen einer Kombination von k Inputelementen $\boldsymbol{X} =$

[1,2,3]

[1,2] [1,3] [2,3]

[1] [2] [3]

[∅]

Abbildung 3.4: Bsp. für einen Suchraum: Die Teilmengen von [1,2,3]. Die k-te Zeile repräsentiert die Teilmengen mit k Elementen ($0 \leq k \leq 3$). Abbildung nach [35]

.

$[X_1, \ldots, X_k]$ und einem Outputelement Y_j'

$$MI(\boldsymbol{X}, Y_j') = H(Y_j'), \tag{3.1}$$

dann bestimmen die Veränderungen der Inputelemente die Veränderungen des Outputelementes Y_j' vollständig. Auf Basis von Gleichung (3.1) wird bei REVEAL eine Entscheidung getroffen. Die Elemente $X_1, ..., X_k$ sind somit die eindeutigen Eltern für Y_j'. Eine verbesserte Laufzeit des Algorithmus kann erreicht werden, indem nicht (3.1) sondern

$$H(\boldsymbol{X}, Y_j') = H(\boldsymbol{X}) \tag{3.2}$$

getestet wird, denn für $H(\boldsymbol{X}, Y') = H(\boldsymbol{X})$, folgt aus Gleichung (2.3)

$$MI(\boldsymbol{X}, Y') = H(Y'). \tag{3.3}$$

Für den Umkehrschluss gilt, dass, falls $MI(\boldsymbol{X}, Y') = H(Y')$, wiederum aus Gleichung (2.3)

$$H(\boldsymbol{X}) = H(\boldsymbol{X}, Y') \tag{3.4}$$

folgt. Daher:

$$H(\boldsymbol{X}, Y') = H(\boldsymbol{X}) \Leftrightarrow MI(\boldsymbol{X}, Y') = H(Y). \tag{3.5}$$

Für k gegebene Inputelemente gibt es insgesamt $2^{(2^k)}$ Boolesche Funktionen. Von ihnen kann REVEAL aber nur diejenigen identifizieren, bei denen das Outputelement effektiv von allen k Inputelementen abhängt:

Definition 3.6 (k_{eff}-Regeln [28]) *Kann eine Boolesche Funktion, dargestellt in einer Regeltabelle, nicht auf eine Boolesche Funktion mit weniger Inputelementen reduziert werden, dann haben alle ihre Inputelemente einen effektiven Einfluss auf das Outputelement. Sie ist eine k_{eff}-Boolesche Funktionen.*

Es gilt $k_{eff} \leq k$ und im folgenden bezeichnet k_{eff} die Anzahl der Inputelemente, von denen das Outputelement effektiv abhängt.
In einem Booleschen Netz mit N Elementen gibt es für jedes Element $\sum_{k=0}^{N} \binom{N}{k}$ Mengen von potentiellen Eltern. Dies ergibt durch Anwendung des Binomischen Lehrsatzes einen Suchraum von insgesamt 2^N Kombinationen (s. Abb. 3.4) [13]. Theoretisch müsste der gesamte Raum durchsucht werden, um die Eltern für ein Element eindeutig zu bestimmen. REVEAL startet am Boden des Gitters in Abb. 3.4 und berechnet für jede Kombination von Inputelementen die Entropie. Die Suche endet, sobald Gleichung (3.2) zutrifft. Ist der Algorithmus in dem obersten Level des Gitters angelangt und hat noch keine Elternelemente gefunden, dann wird das Outputelement nicht von diesem Booleschen Netz beeinflusst. Dies hat nicht zu bedeuten, dass es nicht selbst ein Elternelement sein kann. Vielmehr weist diese Tatsache darauf hin, dass dieses Outputelement von einem unbekanntem Element abhängen kann. Es wäre also sinnvoll, dass Boolesche Netz um eine Variable zu erweitern, um zu prüfen ob sie den unbekannten Einfluss repräsentiert.
Das Ergebnis von REVEAL ist die Zuordnung der Eltern zu jedem Netzwerkelement (*Wiring Diagram*). Die Booleschen Funktionen können nachfolgend bestimmt werden, falls genügend verschiedene Belegungen des Booleschen Netzes in den Trainingsdaten gegeben sind. Dies soll geschehen, indem die beobachteten Zusammenhänge in Regeltabellen eingetragen werden.
Mit Abbruchbedingung (3.2) liegt eine zu harte Bedingung vor, die nur erreicht wird, falls ungestörte Trainingsdaten zur Verfügung stehen. Nur dann ist gewährleistet, dass die Beziehung (3.2) aufgedeckt werden kann. Bei realistischen Anwendungen werden jedoch die Trainingsdaten immer gestört sein bzw. rauschen. Folglich ist REVEAL in seiner ursprünglichen Form nicht mehr anwendbar. Deshalb sollten statistische Tests herangezogen werden. Miller liefert in seinem Artikel [34] eine mögliche Approximation der Mutual Information zur χ^2-Statistik. Damit kann dann die Nullhypothese, dass das Outputelement unabhängig von den Inputelementen ist, getestet werden [1].

[1] Für nähere Ausführungen siehe dazu Kapitel 4.4

3.3 Dynamische Bayessche Netze (DBN)

Dynamische Bayessche Netze eignen sich gut, um aktuelles Wissen bzw. Unsicherheit über das dynamische Verhalten von Genregulationsnetzwerken darzustellen. Es ist insbesondere zu beachten, dass Trainingsdaten aus realen biologischen Systemen immer gewissen Störungen unterliegen und deshalb nur stochastische Relationen zwischen den Genen identifizierbar sind, auch wenn die realen Beziehungen deterministisch wären. Ein Boolesches Netz zu rekonstruieren ist demnach nicht möglich. Es soll analysiert werden, ob ein DBN, das gegeben der Daten sehr wahrscheinlich ist, identifiziert werden kann. Bayessche Netze haben außerdem den Vorteil, dass sie mit gut fundierten Methoden gelernt werden können.
Stochastische Zufallsvariablen repräsentieren in Bayesschen Netzen die Expressionsstärke und stochastische Relationen die Beziehungen zwischen diesen Zufallsvariablen. Die erste Relation ist die *bedingte Wahrscheinlichkeit* und modelliert die Abhängigkeit der Expressionsstärke eines Gens von der Expression anderer Gene. Die bedingte Wahrscheinlichkeit zwischen zwei Zufallsvariablen A und B ist durch die *Bayessche Regel*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.6}$$

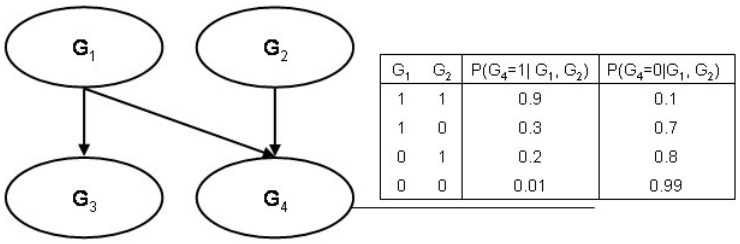
definiert. Die zweite stochastische Relation ist die *Unabhängigkeit*. Zwei Zufallsvariablen sind unabhängig, falls

$$P(A, B) = P(A)P(B). \tag{3.7}$$

Sind zwei Zufallsvariablen unabhängig, dann gibt uns der Wert der einen Zufallsvariablen keine Information über den Wert der anderen Zufallsvariablen, also:

$$P(A|B) = P(A). \tag{3.8}$$

Eine weitere Eigenschaft ist die *bedingte Unabhängigkeit* zwischen zwei Zufallsvariablen. Sie bedeutet, dass zwei Zufallsvariablen nur dann unabhängig sind, wenn der Zustand einer dritten Zufallsvariable bekannt ist. Dieser Sachverhalt sei hier an einem Beispiel erläutert. Es sind drei Zufallsvariablen gegeben, die eine Krankheit K und zwei Tests, T_1 und T_2, repräsentieren. Die Tests sind verschiedene Indikatoren dafür, ob der Patient die Krankheit besitzt. Die Krankheit beeinflusst die Ergebnisse der Tests. Fällt T_1 positiv aus, dann erhöht sich für einen Beobachter die Wahrscheinlichkeit des Auftretens der Krankheit und somit auch die Wahrscheinlichkeit das T_2 ein positives Ergebnis bringt. T_1 und T_2 sind also nicht unabhängig. Ist dagegen



G_1	G_2	$P(G_4=1 \mid G_1, G_2)$	$P(G_4=0 \mid G_1, G_2)$
1	1	0.9	0.1
1	0	0.3	0.7
0	1	0.2	0.8
0	0	0.01	0.99

Abbildung 3.5: Beispiel für ein Bayessches Netz. Die Knoten repräsentieren die Expressionsstärke der Gene und es wird angenommen, dass die Genexpressionsstärke entweder 0 oder 1 ist.

bekannt, dass der Patient die Krankheit hat, dann beeinflusst ein positives Ergebnis von T_1 nicht unseren Glauben über das Ergebnis von T_2, also

$$P(T_2|K = ja, T_1 = positiv) = P(T_2|K = ja).$$

Gilt diese Beziehung für alle Zustände der Zufallsvariablen, dann sind T_1 und T_2 bedingt unabhängig, gegeben K.
Bayessche Netze enthalten einen qualitativen Teil, die grafische Struktur, und einen quantitativen Teil, die Parameter. Im folgenden ist eine formale Definition für Bayessche Netze gegeben:

Definition 3.7 *Ein Bayessches Netz $B = (G, \mathbf{\Theta})$ besteht aus einem Graphen G und einer Parametermenge $\mathbf{\Theta}$. G ist ein gerichteter azyklischer Graph, dessen Knoten zu den Zufallsvariablen $X_1, \ldots, X_N$ korrespondieren. Eine Belegung für eine Zufallsvariable X_i wird mit x_i notiert. Eine gerichtete Kante zwischen zwei Knoten $X_i \rightarrow X_j$ bedeutet, dass X_j direkt von X_i beeinflusst wird. G definiert die statistischen Unabhängigkeitsbeziehungen zwischen den Zufallsvariablen. $\mathbf{\Theta}$ enthält für jede Zufallsvariable X_i den Parameter $\theta_{ij_ik_i} = P(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i)$ für jeden Wert k_i von X_i und jeder möglichen Menge von Werten j_i von $\boldsymbol{Pa}(X_i)$. $\boldsymbol{Pa}(X_i)$ ist die Menge der Eltern von X_i.*

Die Struktur eines Bayesschen Netzes repräsentiert die bedingte Unabhängigkeit der Zufallsvariablen.

Annahme 3.1 (Bedingte Unabhängigkeit [12]) *Jeder Knoten in einem Bayesschen Netz ist unabhängig von allen nicht nachfolgenden Knoten, gegeben seiner Eltern.*

	$X_1 = 1$	$X_1 = 2$	$X_1 = 3$
$X_2 = 1$	0.03	0.05	0.06
$X_2 = 2$	0.07	0.12	0.34
$X_2 = 3$	0.11	0.14	0.08

Tabelle 3.1: Beispiel einer Verbundwahrscheinlichkeit

Kindsknoten können zwar nicht den Wert der Elternknoten verändern, aber unseren Glauben über die Elternelemente.
Ein Beispiel für ein Bayessches Netz zeigt Abb. 3.5. Die Parameter $\theta_{ij_ik_i}$ für das Gen G_4 sind in einer Tabelle zusammengefasst. Diese Tabelle definiert die *Bedingte Wahrscheinlichkeitsverteilung (engl.: Conditional Probability Distribution)* (CPD) für Gen G_4. Eine CPD ist eine Wahrscheinlichkeitsverteilung über die Zustände einer Zufallsvariablen in Abhängigkeit der bedingenden Zufallsvariablen. Übertragen in die Domäne der Genregulationsnetzwerke ist eine CPD die Verteilung über die Expressionsstärken eines Gens in Abhängigkeit der Expressionsstärken der regulierenden Gene. Wird eine diskrete CPD als Tabelle dargestellt, heißt sie eine *Bedingte Wahrscheinlichkeitstabelle (engl.: Conditional Probability Table)* (CPT). Jede Zeile eines CPT enthält die bedingte Wahrscheinlichkeit für alle Zustände einer Zufallsvariablen für eine spezielle Belegung der bedingenden Zufallsvariablen.
Die Verteilung $P(X_1, \ldots, X_N)$ weist allen möglichen Belegungen $x_1, \ldots, x_N$ der Zufallsvariablen Wahrscheinlichkeiten zu und wird *Verbundwahrscheinlichkeitsverteilung (engl.: Joint Probability Distribution)* (JPD) genannt. Angenommen es wird eine Domäne mit zwei Zufallsvariablen X_1 und X_2, die jeweils drei Zustände annehmen können, betrachtet. Die Tabelle 3.1 zeigt eine mögliche Verbundwahrscheinlichkeitsverteilung für diese Domäne. Wobei die Summe der Einträge 1 beträgt.
Die Wahrscheinlichkeit für ein atomares Ereignis kann aus den Informationen aus $B = (G, \mathbf{\Theta})$ berechnet werden. Ein solches Ereignis kann z.B. sein, dass die Gene G_1 bis G_3 exprimieren und G_4 nicht (Abb. 3.5). Die Wahrscheinlichkeit für dieses Ereignisses ist $P(G_1 = 1, G_2 = 1, G_3 = 1, G_4 = 0)$ [2]. Darauf die Kettenregel angewendet, erhält man:

$$\begin{aligned} P(G_3 = 1, G_4 = 0, G_1 = 1, G_2 = 1) \quad = \quad & P(G_3 = 1 | G_4 = 0, G_1 = 1, G_2 = 1) * \\ & P(G_4 = 0 | G_1 = 1, G_2 = 1) * \\ & P(G_1 = 1 | G_2 = 1) * P(G_2 = 1). \end{aligned}$$

[2]Würde nach dem Ereignis $P(G_3 = 1, G_4 = 0)$ gefragt, dann müßte über die möglichen binären Zuweisungen von G_1 und G_2 summiert werden, denn diese beeinflussen das Auftreten dieses Ereignisses. Es folgt $P(G_3 = 1, G_4 = 0) = \sum_{x=0}^{1} \sum_{y=0}^{1} P(G_1 = x, G_2 = y, G_3 = 1, G_4 = 0)$

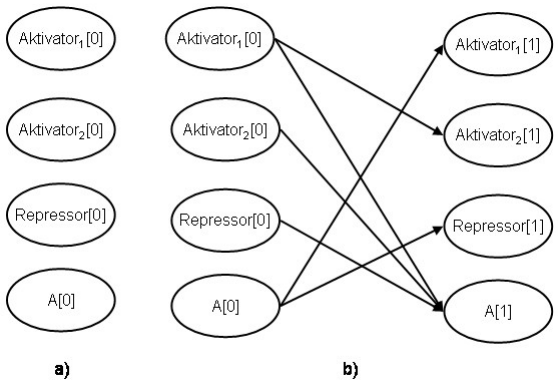


Abbildung 3.6: Beispiel für die Struktur eines Dynamisches Bayessches Netz. Die Startstruktur ist in a) zu sehen und die Übergangsstruktur in b).

Dieses Produkt kann weiter vereinfacht werden, indem die Information über die bedingten Unabhängigkeiten (Annahme 3.1) aus der Struktur des Bayesschen Netzes herangezogen wird:

$$\begin{aligned} P(G_3 = 1, G_4 = 0, G_1 = 1, G_2 = 1) &= P(G_3 = 1|G_1 = 1) * \\ &\quad P(G_4 = 0|G_1 = 1, G_2 = 1) * \\ &\quad P(G_1 = 1) * P(G_2 = 1). \end{aligned}$$

Daraus ist erkennbar, dass jeder Eintrag in der Verbundwahrscheinlichkeitsverteilung ein Produkt der passenden Werte der einzelnen CPTs ist und jedes beliebige Ereignis $x_1, \ldots, x_N$ durch die Formel

$$P_B(x_1, \ldots, x_N) = \prod_{i=1}^{N} P_B(x_i|\boldsymbol{pa}_i). \tag{3.9}$$

berechnet werden kann[3]. x_i ist eine Belegung für die Zufallsvariable X_i und $\boldsymbol{pa}_i$ eine Belegung ihrer Eltern. Dies wiederum zeigt, dass die CPTs nur eine Zerlegung der Verbundwahrscheinlichkeitsverteilung sind.
Mit Inferenz in Bayesschen Netzen können Entscheidungen getroffen werden.

[3]Gleichung (3.9) erhält man nur, wenn die Ereignisse in der Reihenfolge, Kinder zuerst und nachfolgend die Eltern, notiert werden. Andererseits hängen die Wahrscheinlichkeiten nicht nur von den Eltern, sondern auch von den Kindern ab (Annahme 3.1).

Definition 3.8 (Inferenz [39]) *Inferenz oder Schließen in einem Bayesschen Netz ist die Berechnung der posterior-Wahrscheinlichkeitsverteilung für eine Menge von Variablen (Anfrage), vorausgesetzt dass Werte für andere Variablen (Beleg) beobachtet wurden.*

Es existieren standardisierte Inferenzalgorithmen, die u.a. in [39] und [25] beschrieben sind.
Ein klassisches Bayessches Netz kann keine zeitliche Evolution beschreiben. Es eignet sich nur für statische Modellierung. Für die Modellierung der Dynamik eines Genregulationsnetzwerkes, sind die *Dynamischen Bayesschen Netze* zu benutzen [35]. Sei $\mathbf{X} = \{X_1, \ldots, X_N\}$ die Menge der Variablen, deren Zustände sich über die Zeit ändern. $X_i[t]$ ist die Zufallsvariable, die den Wert der Variable X_i zur Zeit t bestimmt und $\mathbf{X}[t]$ die Menge der Zufallsvariablen $X_i[t]$. In der vorliegenden Arbeit wird angenommen, dass der dynamische Prozess eines Genregulationsnetzwerkes ein Markov Prozess ist, also $P(\mathbf{X}[t+1]|\mathbf{X}[0], \ldots, \mathbf{X}[t]) = P(\mathbf{X}[t+1]|\mathbf{X}[t])$. Außerdem wird angenommen, dass es sich um einen stationären Prozess handelt, d.h. die Übergangswahrscheinlichkeit $P(\mathbf{X}[t+1])|\mathbf{X}[t])$ ist unabhängig von t. Diese Annahmen bewirken eine wesentliche Vereinfachung des Modells. Es besteht aus zwei Strukturen, der *Startstruktur* $G^{(0)}$ und der *Übergangsstruktur* $G^{\rightarrow}$ mit den dazugehörenden Verteilungen [13]. $G^{(0)}$ repräsentiert die initiale Struktur mit der initialen Verteilung für die Variablenbelegungen. Die Übergangsstruktur $G^{\rightarrow}$ beschreibt die bedingten Unabhängigkeiten für einen Zeitschritt. Mit den Variablen in $\mathbf{X}[0]$ von $G^{\rightarrow}$ sind keine Eltern und CPTs assoziiert. Sei B ein DBN, dann ist

$$B = (B_0, B_{\rightarrow}) = (\{G^{(0)}, \mathbf{\Theta}^{(0)}\}, \{G^{\rightarrow}, \mathbf{\Theta}^{\rightarrow}\}).$$

Damit ist nun die Verbundwahrscheinlichkeitsverteilung $P_B(\mathbf{X}[0], \ldots, \mathbf{X}[T])$ definiert [13]. Die Wahrscheinlichkeit für eine Belegung $\mathbf{x}[0], \ldots, \mathbf{x}[T]$ ist:

$$P_B(\mathbf{x}[0], \ldots, \mathbf{x}[T]) = P_{B_0}(\mathbf{x}[0]) \prod_{t=0}^{T-1} P_{B_{\rightarrow}}(\mathbf{x}[t+1]|\mathbf{x}[t]) \tag{3.10}$$

mit der Verteilung für die Übergangswahrscheinlichkeiten

$$P_{B_{\rightarrow}}(\mathbf{x}[t+1]|\mathbf{x}[t]) = \prod_{i=1}^{N} P_{B_{\rightarrow}}(X_i[t+1] = k_i|\mathbf{Pa}(X_i[t+1]) = j_i). \tag{3.11}$$

Die Eltern von $X_i[t+1]$ sind die Variablen in t und $t+1$, die zu den Eltern von $X_i[1]$ in $B_{\rightarrow}$ korrespondieren.

3.4 Reverse Engineering eines Dynamisch Bayesschen Netzes

3.4.1 Identifizieren der Parameter aus Zustandsübergangsdaten

Ist die Struktur bekannt und müssen nur noch die Parameter, also der quantitative Teil des DBN geschätzt werden, dann handelt es sich um ein einfach zu lösendes Reverse Engineering Problem. Sind zusätzlich die Trainingsdaten vollständig gegeben, gibt es mehrere gut verstandene Methoden die Parameter

$$\begin{aligned}\theta^{(0)}_{ij_ik_i} &= P_{B_0}(X_i[0] = k_i|\mathbf{Pa}(X_i[0]) = j_i)\\ \theta^{\rightarrow}_{ij_ik_i} &= P_{B_\rightarrow}(X_i[t+1] = k_i|\mathbf{Pa}(X_i[t+1]) = j_i)\end{aligned}$$

zu schätzen. Die Trainingsdaten D bestehen aus N_R Trajektorien. Jede Trajektorie enthält die beobachteten Werte für alle N Variablen für insgesamt T Zeitschritte[4]. Für eine vereinfachte Notation wird angenommen, dass die Trajektorien von gleicher Länge sind. Eine Trajektorie R_i hat demnach folgende Form:

$$R_i = \begin{bmatrix} x_1[0] & \dots & x_1[T-1] \\ \vdots & \ddots & \vdots \\ x_N[0] & \dots & x_N[T-1] \end{bmatrix} \tag{3.12}$$

und D

$$D = \{R_1, \dots, R_{N_R}\}. \tag{3.13}$$

Eine Zeile in R_i beschreibt den Zustandsverlauf einer Variable über T Zeitschritte und eine Spalte den globalen Zustand des DBN zu einem Zeitschritt t ($t = 0, \dots, T-1$). Da, wie angenommen, die Übergangswahrscheinlichkeiten in einem DBN unabhängig von der Zeit sind, stehen für einen Parameter $\theta^{\rightarrow}_{ij_ik_i}$ insgesamt $M^{\rightarrow} = N_R * (T-1)$ Trainingsvektoren/Stichprobenwerte zur Verfügung. Für die Parameter $\theta^{(0)}_{ij_ik_i}$ gibt es demzufolge nur $M^{(0)} = N_R$ Trainingsvektoren. $M^{(0)}$ enthält die initialen Zustände ($t = 0$) der Zufallsvariablen und $M^{\rightarrow}$ die Zustände auf den Trajektorien. Werden im folgenden *Daten* D erwähnt, dann sind damit diese *Trainingsvektoren* gemeint. Die Parameter in $B_{(0)}$ und $B_\rightarrow$ sind unabhängig voneinander und daher werden sie auch getrennt voneinander geschätzt. Deshalb bezeichnet im folgenden $\theta_{ij_ik_i}$ entweder $\theta^{(0)}_{ij_ik_i}$ oder $\theta^{\rightarrow}_{ij_ik_i}$. Analog dazu steht M entweder für $M^{(0)}$ oder $M^{\rightarrow}$.

[4]In dieser Arbeit ist $T = 2$.

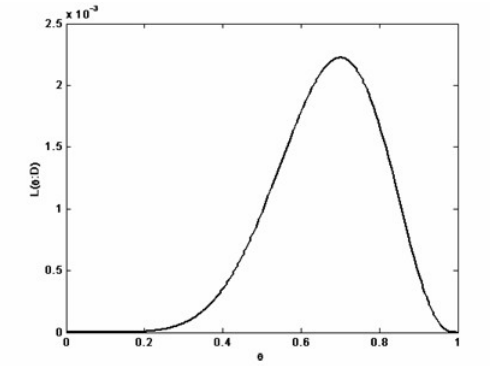


Abbildung 3.7: Die Likelihood Funktion, falls 7 mal Kopf und 3 mal Zahl bei dem Wurf einer Münze beobachtet wurde. $\theta_K = 0.7$ ist der Maximum Likelihood Schätzer.

Für das Lernen der Parameter sei vorausgesetzt, dass die Trajektorien unabhängig voneinander sind und der gleichen Verteilung unterliegen. Nur dann hängt D von dem gleichen Parametervektor $\theta = \{\theta^{(0)}, \theta^{\rightarrow}\}$ ab. Die Verteilung der Daten definiert somit für jeden Trainingsvektor die Wahrscheinlichkeit $P(R_i|\boldsymbol{\theta})$. Zusammenfassend kann die gestellte Aufgabe als

$$D = \{R_1, \ldots, R_{N_R}\} \rightarrow \hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}$$

aufgefasst werden. Es wird also nach den Schätzern $\hat{\theta}_{ij_ik_i}$ gesucht, die die vorhandenen Daten gut beschreiben[5]. Anders ausgedrückt: Sind die Trainingsdaten gegeben $\hat{\theta}_{ij_ik_i}$ sehr wahrscheinlich, dann sind $\hat{\theta}_{ij_ik_i}$ gute Schätzer für die Parameter der Grundmenge. Für das Schätzen ohne die Hinzunahme von Vorwissen (*Frequentismus*) liefert die *Likelihood Funktion* genau diese Schätzungen.

Definition 3.9 *Sei eine Stichprobe $D = \{d_1, \ldots, d_M\}$ gegeben. Dann ist*

$$L(\theta : D) = L(D|\theta) = P(D|\theta) = \prod_{m=1}^{M} P(d_m|\theta)$$

*die **Likelihood Funktion** für die Beobachtungen D und einem Parameter θ.*

Der natürlichste Ansatz ist nach dem Parameter θ zu suchen, welcher die Likelihood Funktion maximiert. Denn es interessiert der Parameter, unter dem die Daten am wahrscheinlichsten sind. Das ist der *Maximum Likelihood Schätzer* ($\hat{\theta}^{MLE}$):

$$\hat{\theta}^{MLE} = \max_{\theta} P(D|\theta). \tag{3.14}$$

[5] $\hat{X}$ ist der Schätzer der Größe X.

Das einfachste Beispiel, was zur Veranschaulichung herangezogen werden kann, ist der Wurf einer Münze. Es soll der Parametervektor $\boldsymbol{\theta} = \{\theta_K, \theta_Z\}$ mit
$\theta_K = P(\text{Kopf liegt oben})$ und $\theta_Z = P(\text{Zahl liegt oben})$ geschätzt werden. Dafür wird eine Folge von Würfen ausgeführt, deren Ergebnisse die Trainingsdaten liefern. Nach 10 Würfen liegen zum Beispiel die Daten $\{K, K, Z, K, Z, K, K, K, Z, K\}$ vor. Die daraus resultierende Likelihood Funktion ist $L(\theta : D) = \theta_K^7 * \theta_Z^3 = \theta_K^7 * (1-\theta_K)^3$ und ihr Verlauf in Abbildung 3.7 zu sehen.
Der Maximum Likelihood Schätzer ist jedoch nur sinnvoll, wenn ausreichend viele Trainingsdaten zur Verfügung stehen, so dass darin jede mögliche Belegung der Modellvariablen genügend oft vorkommt. Anderenfalls schließt man aus den Trainingsdaten, dass bestimmte Werte durch die Variablen nicht angenommen werden, obwohl dies eigentlich der Fall ist. Denn Werte die in den Daten nicht beobachtet wurden, erhalten durch den Maximum Likelihood Schätzer die Wahrscheinlichkeit 0. In dem vorangegangen Beispiel wurde für die Schätzung der Parameter zu wenige Würfe unternommen. Denn nach der Erfahrung eines Beobachters über den Wurf einer Münze hätte er $\theta_K = \theta_Z = 0.5$ und nicht $\theta_K = 0.7$ und $\theta_Z = 0.3$ erwartet. Um solche Fehlschlüsse zu umgehen, würde er Bayessche Lernmethoden[6] vorziehen, mit der er sein eventuell vorhandenes Vorwissen integrieren kann.

3.4.2 Identifizieren der Struktur aus Zustandsübergangsdaten

Ein schwierigeres Reverse Engineering Problem ist, dass neben den Parametern auch noch die Struktur, also $G^{(0)}$ und $G^{\rightarrow}$, des DBN unbekannt ist. Es soll nun der beste Schätzer $\hat{G} = (\hat{G}^{(0)}, \hat{G}^{\rightarrow})$ gefunden werden. Die Strategien für das Identifizieren der Struktur soll anhand von statischen Bayesschen Netzen hergeleitet werden. Sie werden jeweils auf Dynamische Bayessche Netze mit den Trainingsdaten wie in Bez. (3.13) erweitert, was sich wegen Gleichung (3.10) nachvollziehen lässt. Daher wird die gestellte Aufgabe hier zunächst für ein statisches Bayessches Netz als

$$D = \begin{bmatrix} x_{11} & \dots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{N1} & \dots & x_{NM} \end{bmatrix} \rightarrow \hat{G}$$

[6] In der Statistik gibt es zwei wesentliche Theorien. Die *Frequentistische Theorie* nimmt an, dass es einen festen Parameter gibt und schätzt diesen mit bestimmter Konfidenz. Die *Bayesschen Methoden* definieren dagegen einen unbekannten Parameter θ als eine Zufallszahl und es soll die Erwartung des Beobachters für θ in der $n+1$-ten Beobachtung bestimmt werden. Dafür wird über die Werte für θ eine prior-Verteilung $P(\theta)$ gelegt. Sie repräsentiert die Unsicherheit die der Beobachter hat, also sein a-priori Wissen über den unbekannten Parameter. Beobachtete Daten revidieren diese Unsicherheit über ihre Likelihood $P(D|\theta)$ und die aktualisierte Unsicherheit ist durch die *posterior-Verteilung* $P(\theta|D)$ beschrieben.

aufgefasst. M ist dann die Anzahl der Trainingsvektoren und N die Anzahl der Variablen in einer beliebigen Struktur G. Jede Spalte ist also eine Realisierung für alle N Variablen. Ein Schätzer $\hat{G}$ für die wahre Struktur ist dann ein optimaler Schätzer, wenn er gegeben den Trainingsdaten am wahrscheinlichsten ist. Um einen guten Schätzer zu finden, muss eigentlich der gesamte Raum von Strukturen durchsucht werden. Dieser Raum hat jedoch eine exponentielle Anzahl von Strukturen, wodurch die Suche nach der optimalen Struktur NP-hart ist. Eine Annäherung an die optimale Lösung kann daher nur durch die Definition eines Suchraumes und der Anwendung einer heuristischen Suche erreicht werden. Außerdem ist eine Scoring-Funktion notwendig, die eine Struktur gegeben der Daten bewertet. Somit umfasst eine Strategie zur Identifizierung der Struktur jeweils eine Scoring-Funktion, einen Suchraum (Operatoren für die Modifizierung der Strukturen) und einen Suchalgorithmus. Es soll hier auf die unterschiedlichen Scoring-Funktionen und kurz auf die Suchalgorithmen näher eingegangen werden.
Die drei wichtigsten Scoring-Funktionen für das Lernen ohne Vorwissen sind der *Likelihood Score*, der zu ihm äquivalente *Mutual Information Score* (MIS) und das *Bayessche Informations Kriterium* (BIC) (engl.: Bayesian Information Criterion).
Der Likelihood Score für eine Struktur G mit N Elementen, einer Trainingsmenge D mit unabhängigen Trainingsvektoren und der Maximum Likelihood Schätzungen $\hat{\mathbf{\Theta}}_G$ für die Parameter von G ist

Definition 3.10 (Likelihood)

$$L(G : D|\hat{\mathbf{\Theta}}_G) \;=\; L(D|G, \hat{\mathbf{\Theta}}_G) = P(D|G, \hat{\mathbf{\Theta}}_G) = \prod_{m=1}^{M} P(\bar{x}_m|G, \hat{\mathbf{\Theta}}_G).$$

Dies ist nichts anderes als die Likelihood Funktion für G, also die Wahrscheinlichkeit der Daten gegeben G und der wahrscheinlichsten Instanz von Parametern $\hat{\mathbf{\Theta}}_G$. Der Score vereinfacht sich wesentlich, wenn anstatt der Likelihood Funktion die *Log Likelihood Funktion* genommen wird:

Definition 3.11 (Log Likelihood)

$$\log L(G : D|\hat{\mathbf{\Theta}}_G) = \sum_{m=1}^{M} \log P(\bar{x}_m|G, \hat{\mathbf{\Theta}}_G).$$

Jeder Trainingsvektor $\bar{x}_m$ ist eine Belegung für das Bayessche Netz. Zu jeder Belegung ist aufgrund der geschätzten Verbundwahrscheinlichkeitsverteilung $\hat{\mathbf{\Theta}}_G$ genau eine Verteilung zugeordnet:

$$P(\bar{x}_m|G, \hat{\mathbf{\Theta}}_G) = \prod_{i=1}^{N} P(x_{im}|\mathbf{pa}_{im}). \tag{3.15}$$

Somit ist die Likelihood:

$$L(G : D|\hat{\mathbf{\Theta}}_G) = \prod_{i=1}^{N} \prod_{m=1}^{M} P(x_{im}|\mathbf{pa}_{im}). \tag{3.16}$$

Daraus folgt, dass sich die Likelihood entsprechend der Struktur G zerlegen läßt. Es sei $N_{ij_ik_i}$ die Anzahl der Vektoren in D, bei denen $X_i = k_i$ und $\mathbf{Pa}(X_i) = j_i$. Wenn die Variablen multinomial verteilt sind, ist für alle möglichen Belegungen k_i und j_i [13]:

$$\begin{aligned} L(G : D|\hat{\mathbf{\Theta}}_G) &= \prod_{i=1}^{N} \prod_{j_i=1}^{q_i} \prod_{k_i=1}^{r_i} P(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i)^{N_{ij_ik_i}} \\ &= \prod_{i=1}^{N} \prod_{j_i=1}^{q_i} \prod_{k_i=1}^{r_i} \hat{\theta}_{ij_ik_i}^{N_{ij_ik_i}}. \end{aligned} \tag{3.17}$$

Das zweite Produkt ist über die q_i unterschiedlichen Zustände der Eltern von X_i und das dritte Produkt über die r_i unterschiedlichen Zustände von X_i. Für $\hat{\theta}_{ij_ik_i}$ wird der MLE Schätzer $\frac{N_{ij_ik_i}}{N_{ij_i}}$ mit $N_{ij_i} = \sum_{k_i=1}^{r_i} N_{ij_ik_i}$ genommen. Aus diesen Überlegungen ergibt sich die letztendliche Formel für den Likelihood Score, basierend auf der Log Likelihood [25]:

Definition 3.12 (Likelihood Score)

$$\log L(G : D|\hat{\mathbf{\Theta}}_G) = \sum_{i=1}^{N} \sum_{j_i=1}^{q_i} \sum_{k_i=1}^{r_i} \log \hat{\theta}_{ij_ik_i}^{N_{ij_ik_i}} = \sum_{i=1}^{N} \sum_{j_i=1}^{q_i} \sum_{k_i=1}^{r_i} N_{ij_ik_i} \log \frac{N_{ij_ik_i}}{N_{ij_i}}.$$

Alle Umformungen beruhen auf der Annahme, dass in den Trainingsdaten keine Werte fehlen. Anderenfalls sind die konkreten Häufigkeiten $N_{ij_ik_i}$ nicht bekannt und die Parameter sind nicht mehr unabhängig gegeben der Daten [13]. Die obigen

Zerlegungen sind in solchen Fällen inkorrekt[7].
Der Likelihood Score für Dynamische Bayessche Netze ist wegen (3.10) [13]:

Definition 3.13 (Likelihood Score - DBN)

$$\log L(G : D|\hat{\Theta}_G) = \sum_{i=1}^{N} \sum_{j_i'=1}^{q_i'} \sum_{k_i'=1}^{r_i'} N_{ij_i'k_i'}^{(0)} \log \frac{N_{ij_i'k_i'}^{(0)}}{N_{ij_i'}^{(0)}} + \sum_{i=1}^{N} \sum_{j_i=1}^{q_i} \sum_{k_i=1}^{r_i} N_{ij_ik_i}^{\rightarrow} \log \frac{N_{ij_ik_i}^{\rightarrow}}{N_{ij_i}^{\rightarrow}}.$$

Eine alternative Formulierung für den Log Likelihood Score liefern Entropie und Mutual Information (2) [25]:

$$\log L(G : D|\hat{\Theta}_G) = \sum_{i=1}^{N} MI(X_i, \boldsymbol{Pa}(X_i)) - \sum_{i=1}^{N} H(X_i). \tag{3.18}$$

Die äquivalente Formulierung des Likelihood Score in Glg. (3.18) ist der **MIS Score**. Für eine genauere Herleitung siehe Anhang A. Der letzte Term ist eine Konstante, denn er ist unabhängig von der Modellstruktur. Er hat auf die Optimierungsschritte keinen Einfluß. Der MIS Score ist eine Verallgemeinerung der Beziehung $MI(X_i, \mathbf{Pa}(X_i)) = H(X_i)$ die in REVEAL verwendet wird. Denn die Gleichheitsbeziehung gilt nur, falls die Variablen in $\mathbf{Pa}(X_i)$ X_i vollständig voraussagen. Sonst gilt immer $MI(X_i, \mathbf{Pa}(X_i)) < H(X_i)$. Daraus folgt für die Differenz immer: $MI(X_i, \mathbf{Pa}(X_i)) - H(X_i) \leq 0$. Eine Suche nach dem Maximum dieser Differenz ist demzufolge der MIS Score.
Je größer der Likelihood Score einer geschätzten Struktur $\hat{G}$, desto wahrscheinlicher ist $\hat{G}$ eine gute Beschreibung für die originalen Zusammenhänge. Es ist infolgedessen sinnvoll nach der Struktur zu suchen, die einen maximalen Likelihood Score besitzt. Zwei Zufallsvariablen sind nur dann unabhängig, wenn ihre MI gleich 0 ist. Durch Ungenauigkeiten in den Daten ist es sehr unwahrscheinlich, dass für zwei unabhängige Zufallsvariablen auch tatsächlich $MI(X, Y) = 0$ erhalten wird. Die Mutual Information von zwei Zufallsvariablen ist daher immer kleiner oder gleich als die von drei Zufallsvariablen, also $\hat{MI}(X; Y) \leq \hat{MI}(X; Y, Z)$, auch wenn X eigentlich nur mit Y und nicht mit Z kovariiert. Deshalb resultiert eine zusätzliche Kante fast immer in einem besseren Likelihood Score. Demnach ist die Struktur mit maximalen Likelihood Score meistens die vollständig verknüpfte Struktur. Das Ziel ist es aber nur die Kanten in die Modellstruktur zu integrieren, die einen signifikanten Einfluss repräsentieren. Ein Vorteil liegt darin, dass weniger und bessere Schätzer für die Modellparameter berechnet werden. Denn um einen Parameter für

[7] Siehe Abschnitt 3.4.3.

k Eltern zu schätzen, müssen die Trainingsdaten in 2^k Gruppen gegliedert werden. Eine Gruppe für jede Kombination von Belegungen der Eltern. Ist k groß, dann gibt es viele Gruppen, die jeweils nur wenige Trainingsdaten enthalten. Wären dagegen nur $k-1$ Eltern gegeben, dann können Gruppen, die sich nur in diesem einem Elternteil unterscheiden, zusammengefasst werden. Die Anzahl der Trainingsdaten in einer Gruppe erhöht sich und die empirische Schätzung für den Parameter wird genauer (Gesetz der großen Zahlen).
Das Lernen einer komplexen Struktur kann durch das Einführen eines Strafterms verhindert werden. Das Resultat ist der BIC Score für ein Bayessches Netz [13]:

Definition 3.14 (BIC)

$$\begin{aligned} BIC(G:D|\hat{\mathbf{\Theta}}_G) &= \log P(D|G,\hat{\mathbf{\Theta}}_G) - \frac{\log M}{2} * dim_G \\ &= \sum_{i=1}^{N}\sum_{j_i=1}^{q_i}\sum_{k_i=1}^{r_i} N_{ij_ik_i}\log\frac{N_{ij_ik_i}}{N_{ij_i}} - \frac{\log M}{2} * dim_G. \end{aligned}$$

$dim_G = \sum_{i=1}^{N} q_i(r_i-1)$ ist die Anzahl der Parameter im Modell und M, wie oben, die Anzahl der Trainingsvektoren. Infolgedessen ist der BIC Score für Dynamische Bayessche Netze [13]:

Definition 3.15 (BIC - DBN)

$$\begin{aligned} BIC(G:D|\hat{\mathbf{\Theta}}_G) &= \sum_{i=1}^{N}\sum_{j_i'=1}^{q_i'}\sum_{k_i'=1}^{r_i'} N^{(0)}_{ij_i'k_i'}\log\frac{N^{(0)}_{ij_i'k_i'}}{N^{(0)}_{ij_i'}} - \frac{\log M^{(0)}}{2} * dim_{G^{(0)}} + \\ &\quad \sum_{i=1}^{N}\sum_{j_i=1}^{q_i}\sum_{k_i=1}^{r_i} N^{\rightarrow}_{ij_ik_i}\log\frac{N^{\rightarrow}_{ij_ik_i}}{N^{\rightarrow}_{ij_i}} - \frac{\log M^{\rightarrow}}{2} * dim_{G^{\rightarrow}}. \end{aligned}$$

Bedingt durch den Strafterm bevorzugt der BIC Score weniger komplexe Modelle. Wird eine Variable als Elternelement hinzugefügt, erhöht sich der Log Likelihood Term und sogleich der Strafterm. Es werden nur noch Elemente als Eltern identifiziert, falls die Erhöhung in dem Log Likelihood Term dies rechtfertigt. Je mehr Trainingsvektoren gegeben sind, desto mehr überwiegt der Log Likelihood Term und der Strafterm verliert an Bedeutung, denn die Likelihood wächst linear mit M und der Strafterm nur logarithmisch mit M. Wird ein Zusammenhang zwischen zwei Elementen bei großen M erkannt, ist dies kaum auf statistische Schwankungen als vielmehr auf die wahre Verteilung zurückzuführen und es ist verantwortlich diesen

Zusammenhang in G zu integrieren.
Alternativ zu dem BIC Score können statistische Tests die Signifikanz einer Abhängigkeit in den Daten bewerten. In [34] ist beschrieben, dass für Zufallsvariablen X_i und $\mathbf{Pa}(X_i)$ die Approximation

$$\chi^2(X_i, \mathbf{Pa}(X_i)) \sim \hat{MI}(X_i, \mathbf{Pa}(X_i)) * M * \ln 4 \quad i \neq j \tag{3.19}$$

gilt. X_i kann r_i Belegungen und $\mathbf{Pa}(X_i)$ kann q_i Belegungen annehmen. Ob ein Zusammenhang zwischen Zufallsvariablen X_i und $\mathbf{Pa}(X_i)$ besteht, kann durch den Vergleich der Beziehung (3.19) mit den tabellierten Werten der χ^2-Verteilung ermittelt werden. Wird die Nullhypothese, $MI(X_i, \mathbf{Pa}(X_i)) = 0$, auf dem Signifikanzniveau α abgelehnt, also $\hat{MI}(X_i, \mathbf{Pa}(X_i)) * M * \ln 4 > \chi^2_{(r_i-1)*(q_i-1);1-\alpha}$, dann gilt als widerlegt, dass X_i und $\mathbf{Pa}(X_i)$ keinen Zusammenhang besitzen. Die Elemente in $\mathbf{Pa}(X_i)$ werden in die Menge der Eltern von X_i aufgenommen.

Die meisten Suchalgorithmen im Raum der Strukturen nutzen eine wichtige Eigenschaft.

Definition 3.16 ([20]) *Ein Score heißt zerlegbar, falls er sich in ein Produkt von lokalen Scores zerlegen lässt. Ein lokaler Score ist eine Funktion von nur einer einzigen Variablen und ihrer Eltern, also*

$$Score(G : D) = \prod_{i=1}^{N} Score(X_i, \mathbf{Pa}(X_i) : D)$$

Diese Eigenschaft hat den Vorteil, dass nach einer lokalen Strukturänderung, nicht die gesamte Struktur neu evaluiert werden muss. Es reicht aus die lokale Änderung zu bewerten und ihren positiven oder negativen Beitrag auf den alten Score einzubeziehen. Für alle oben beschriebenen Scores gilt diese Eigenschaft.

Die generelle Idee für die Suche nach der optimalen Struktur ist, eine gewählte Startstruktur so zu modifizieren, dass ein besserer Score resultiert. Sie sind iterative Algorithmen, die lediglich Wissen über ihren aktuellen Zustand und der unmittelbaren Nachbarzustände haben. Wie kann unter diesen Bedingungen das globale Maximum einer Scorefunktion gefunden werden, die unter Umständen mehrere lokale Maxima besitzt? Ein möglicher Suchalgorithmus ist *Hill Climbing*. Hill Climbing evaluiert in jeder Iteration alle möglichen Strukturänderungen und vollzieht die mit der größten Verbesserung im Score. Gibt es mehrere Kandidaten, dann wird eine Änderung zufällig unter ihnen ausgewählt. Die Suche wird beendet, wenn keine Änderung zu einem verbesserten Score führt. Der wesentlichste Nachteil liegt darin, dass der Algorithmus in lokalen Maxima stecken bleibt und die Suche abbricht. Das globale Optimum wird also nicht gefunden. Ein weiterer Nachteil ist, dass die

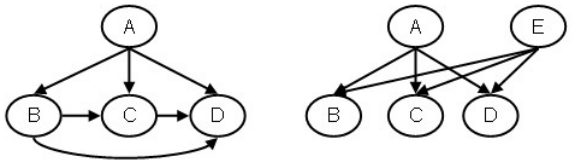


Abbildung 3.8: Das linke Bayessche Netz zeigt eine Region von hoher Konnektivität, nachdem das Modell aus Daten mit diesen 4 Elementen gelernt wurden. Dies ist ein Hinweis, dass die Elemente B, C, D von einer zweiten Variable beeinflusst sind. Wurde das Modell neu, nun mit 5 Elementen gelernt, dann ergab sich das rechte Modell [25].

Suche in einem Plateau der Scorefunktion nur noch zufällig erfolgt und auch hier die Suche abbricht. Abgeänderte Algorithmen, wie *Random restart Hill Climbing* und *Tabu-Search* versuchen diese Probleme zu vermeiden. Ist die Suche in einem lokalem Maxima gefangen, dann wird bei *Random restart Hill Climbing* die aktuelle Struktur in zufällig ausgewählten Kanten verändert und die Suche beginnt neu. Konnte nach einer festgelegten Anzahl von Neustarts keine Verbesserung im Score festgestellt werden, wird die Suche abgebrochen. Bei Tabu-Search kennt der Algorithmus dagegen seine vergangenen K' Schritte/Strukturen und führt die beste Strukturänderung aus, die nicht in den letzten K' Schritten gesehen wurde. Resultiert diese Strukturänderung in keinem besseren Score wird abgebrochen. Die Suche entkommt somit Plateaus nicht größer als K' Strukturen und lokalen Maxima. Eine Kombination von beiden Methoden ist auch möglich.

3.4.3 Unvollständige Trainingsdaten

In dem vorangegangenen Abschnitt wurden Strategien erläutert, wie aus Trainingsdaten die Struktur und die Parameter eines DBN Modells für Genregulationsnetzwerke gelernt werden können. Sie basieren auf der Annahme, dass die Trainingsdaten vollständig zur Verfügung standen. Es ist jedoch oftmals nicht möglich alle Variablen eines Modells in jedem Trainingsvektor zu beobachten. Außerdem wird in der vorliegenden Arbeit von unvollständigen initialen Beobachtungen ausgegangen.
Für solche Situationen gibt es zwei Begründungen. Einerseits können in Trainingsvektoren die Werte für einige Variablen fehlen. Es handelt sich dabei um fehlende Daten. Andererseits ist es auch vorstellbar, dass Werte für Variablen generell nicht beobachtet werden. Demzufolge fehlt diese Variable im Modell und es ist bis dahin nicht bekannt, dass sie in das Modell integriert werden müsste. Ein Hinweis auf die fehlenden Variablen (*Hidden Variablen*) sind Regionen von hoher Konnektivität im Modell (s. Abb. 3.8). Es ist sinnvoll ein Modell auf Hidden Variablen zu überprüfen,

da sich durch das Einfügen dieser Variablen die notwendige Zahl von Parametern und somit die Komplexität des Modells wesentlich verkleinert [35].
Angenommen alle Variablen der Domäne sind bekannt und einige Trainingsvektoren sind unvollständig. Der Fakt, dass in einem Trainingsvektor der Wert einer bestimmten Variable nicht vorhanden ist, kann von den Zuständen der Variablen im Modell abhängen. Ein Beispiel für diesen Sachverhalt ist, dass ein Patient der an einer medizinischen Studie teilnimmt, zu einer Untersuchung nicht kommt, da es ihm gesundheitlich zu schlecht geht. Womöglich bedingt durch die Nebenwirkungen der eingenommenen Medikamente. Dann fehlen die Daten für die Untersuchung nicht zufällig. Um von unvollständigen Trainingsdaten lernen zu können, muss folgende Annahme gemacht werden können:

Annahme 3.2 *Die Wahrscheinlichkeit, dass der Wert von X_i fehlt, ist unabhängig von den Werten der beobachteten Variablen.*

Der Schlüssel für das Schätzen von Parametern ist die Likelihood Funktion. Die funktionale Form der Likelihood verändert sich jedoch, falls die Trainingsdaten unvollständig gegeben sind. Sie besitzt nun nicht mehr ein einziges Maximum sondern es können vielmehr mehrere lokale Maxima auftreten. Denn für die fehlenden Beobachtungen gibt es mehrere Möglichkeiten diese zu ergänzen. Die resultierende Likelihood Funktion ist das gewichtete Mittel aus den Likelihoods für jede Ergänzung und ist meistens eine multimodale Funktion, was die Suche nach einem optimalen Schätzer erschwert. Je mehr Beobachtungen fehlen, desto unnatürlicher gestaltet sich die Likelihood. Den Schätzer $\hat{\theta}^{MLE}$ zu finden, ist nun ein nicht-lineares Optimierungsproblem für eine komplexe multimodale Funktion. Außerdem lässt sich die Likelihood Funktion nicht mehr in unabhängige Produkte zerlegen, weil die Parameterwahl gegeben den Daten nicht mehr unabhängig voneinander sind [13]. Wird zum Beispiel der Wert der Eltern von X_i in einem Trainingsvektor nicht gesehen, dann ist die Wahl für die Parameter für X_i nicht mehr unabhängig. Denn die optimale Wahl für einen Parameter hängt von der Wahl der anderen Parameter ab [13]. Indem die fehlenden Werte aber auf eine bestimmte Weise ergänzt werden, kann eine gute Lösung für die Parameterschätzer gefunden werden.
Für das Optimierungsproblem von multimodalen Likelihood-Funktionen gibt es einen zugeschnittenen Algorithmus, die Erwartungswert Maximierung (engl.: *Expectation Maximization*) (EM). Für das Lernen aus unvollständigen Daten sind zwei Probleme gleichzeitig zu lösen. Zum einem müssen die Werte für die nicht beobachteten Variablen abgeschätzt, zum anderen muss das Modell gelernt werden. EM löst diese Probleme, indem mit einem willkürlichen Startpunkt für die Parameter begonnen wird. Diese beschreiben das aktuelle Modell, mit dessen Hilfe eine Ergänzung der nicht beobachteten Variablen erfolgt. Die ergänzten Trainingsdaten behandelt man wie

die realen Daten und lernt aus ihnen ein neues Modell u.s.w.. Der EM-Algorithmus soll hier für das MLE Schätzverfahren erläutert werden. Bei unvollständigen Trainingsdaten sind die Häufigkeiten $N_{ij_ik_i}$ nicht bekannt. Anstatt der realen Häufigkeiten nutzt EM die *erwarteten Häufigkeiten*. Sei $C(d_m)$ die Menge der Ergänzungen für den m-ten Trainingsvektor und d_m^+ sei eine solche Ergänzung mit dem Gewicht $P(d_m^+|d_m, \theta)$, wobei d_m die beobachteten Daten sind [25]. Dann ergibt sich für die erwarteten Häufigkeiten

$$\bar{N}_{ij_ik_i} = \sum_{m=1}^{M} \sum_{d_m^+ \in C(d_m)} P(d_m^+|d_m, \boldsymbol{\theta}) * I(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i|d_m^+). \tag{3.20}$$

Die Indikatorfunktion $I(y|d_m^+)$ beträgt 1, falls y in der Ergänzung d_m^+ gesehen wurde, sonst 0. Die innere Summe wird also nur über die Ergänzungen ausgeführt, die konsistent mit $X_i = k_i$ und $\mathbf{Pa}(X_i) = j_i$ sind. Sie definieren die Wahrscheinlichkeit $P(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i|d_m, \theta)$. Demzufolge ist [25]

$$\bar{N}_{ij_ik_i} = \sum_{m=1}^{M} P(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i|d_m, \boldsymbol{\theta}). \tag{3.21}$$

Wären vollständige Trainingsdaten gegeben, dann würden die Formeln folgendermaßen aussehen:

$$N_{ij_ik_i} = \sum_{m=1}^{M} I(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i|d_m, \boldsymbol{\theta}). \tag{3.22}$$

Die Formel für die erwarteten Häufigkeiten ist demzufolge identisch zu der Berechnung von realen Häufigkeiten, ausgenommen die Ersetzung der Indikatorfunktion durch eine Wahrscheinlichkeit. Sind die Daten nicht verfügbar, dann wird also für das fehlende Datum eine Wahrscheinlichkeit für dessen Auftreten mit einem bestimmten Wert eingefügt. Andererseits ist die Wahrscheinlichkeit 1 oder 0, je nachdem ob der Wert i aus $i = 1, \ldots, k$ möglichen Werten gesehen oder nicht gesehen wurde.
Daraus folgt die formale Darstellung des EM Algorithmus für Bayessche Netze [25]:

- ***Expectation (E Schritt):*** *Nimm die aktuellen Parameterzuweisungen* $\hat{\boldsymbol{\theta}}^s$*, um die erwarteten Häufigkeiten zu berechnen:*

- *Berechne für jeden Trainingsvektor, jedes X_i und jedes $\mathbf{Pa}(X_i)$ die Verbundwahrscheinlichkeit $P(X_i, \mathbf{Pa}(X_i)|d_m, \hat{\boldsymbol{\theta}}^s)$ mit Inferenzalgorithmen. Die beobachteten Daten d_m fungieren dabei als Beleg für die Anfrage (X_i,$\mathbf{Pa}(X_i)$).*
- *Berechne*

$$\bar{N}_{ij_ik_i} = \sum_{m=1}^{M} P(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i|d_m, \hat{\boldsymbol{\theta}}^s)$$

- ***Maximization (M Schritt):*** *Benutze die erwarteten Häufigkeiten, um die neuen Parameter abzuleiten.*

$$\hat{\boldsymbol{\theta}}^{s+1}_{ij_ik_i} = \frac{\bar{N}_{ij_ik_i}}{\bar{N}_{ij_i}}$$

An der lokalen Position θ^s der gemittelten Likelihood Funktion wird diese durch eine konvexe unimodale Funktion approximiert (E Schritt), deren Maximum einfach zu berechnen ist (M Schritt). Ihr Maximum ist die neue Position θ^{s+1}. Die Approximation ist nichts anderes als die Likelihood Funktion resultierend aus den berechneten erwarteten Häufigkeiten an Position θ^s. EM ist ein Gradienten Anstiegsverfahren und kann in lokalen Maxima gefangen bleiben.

Das Lernen der Struktur aus unvollständigen Trainingsdaten ist direkt aus EM für Parameterschätzung und der Strategien für das Lernen von Strukturen ableitbar. Demnach wird, wie im Abschnitt 3.4.2 beschrieben, die Suche nach der optimalen Struktur mit einer zufällig oder nach bestimmten Kriterien ausgewählten Struktur G_0 begonnen. Um sie und ihre Nachfolger evaluieren zu können, sind jedoch optimale Schätzer für die Parameter notwendig. Es ist also für jeden möglichen Nachfolger von G_0 notwendig den EM Algorithmus für eine angemessene Zahl von Iterationen laufen zu lassen, um gute Schätzer für die Parameter in der Struktur zu erhalten. Diese Strategie ist jedoch sehr rechenaufwendig, denn jede Iteration in EM verlangt die Anwendung eines *Inferenzmechanismus* für jeden nicht gesehenen Stichprobenwert, um die Daten zu ergänzen. Wenn für jede Struktur mehrere EM Iterationen ausgeführt werden müssen, sind dies für die praktische Anwendung zu hohe Kosten. Friedman entwickelte eine Version von EM zugeschnitten auf die Suche nach der optimalen Struktur - strukturelle Erwartungswert Maximierung (engl.: *Structural Expectation Maximization*) (SEM) [12]. Das Prinzip hinter SEM ist, dass ein initiales Bayessches Netz B_0 gewählt wird, um die Daten mit EM iterativ zu vervollständigen, (E Schritt) und aktuelle optimale Schätzer θ^s für die Parameter von B_0 zu finden (M Schritt). Diese optimalen Schätzer sind dann die aktuelle Parametrisierung der Nachfolger von B_0. Dadurch fallen die vielen EM-Iterationen für jeden einzelnen Nachfolger weg, um eine gute erste Parameterwahl θ^s zu finden. Um den

Score zu berechnen, wird anhand dieser aktuellen Parameter für jeden Nachfolger ein letzter EM Schritt ausgeführt.
Dieses Vorgehen wird damit begründet, dass für neue EM Iterationen anhand einer nachfolgenden Struktur keine wesentlichen Änderungen in der Parameterwahl erwartet werden [25]. Nach einigen lokalen Strukturänderungen jedoch, wird das neue Bayessche Netz B_k wesentlich anders als B_0 sein und eine erneute Anwendung von EM ist angebracht.
Wie erfolgt die Parametrisierung eines Nachfolgers genau? Sei $B_C = (G_{B_C}, \boldsymbol{\theta}_{B_C})$ das Bayessche Netz auf dessen Grundlage die aktuelle Parameterwahl getroffen wurde. Für die Berechnung des Scores für das aktuelle Netz B' sind z.B. MLE Schätzer notwendig. Sie werden über die relativen Häufigkeiten gewonnen. Diese ergeben sich aus den erwarteten Häufigkeiten $\bar{N}_{ij_ik_i}$, welche sich relativ zu B_C berechnen lassen:

$$\bar{N}_{ij_ik_i} = \sum_{m=1}^{M} P(X_i = k_i, \boldsymbol{Pa}_{B'}(X_i) = j_i | d_m, B_C, \boldsymbol{\theta}_{B_C}). \tag{3.23}$$

Die Menge der Eltern $\mathbf{Pa}_{B'}(X_i)$ sind dabei die Eltern von X_i in B' und nicht die Eltern in B_C. Die Verbundwahrscheinlichkeitsverteilung kann durch *Inferenzalgorithmen* berechnet werden. Letztendlich ergibt sich daraus der SEM-Algorithmus für den MLE Schätzer [25]:

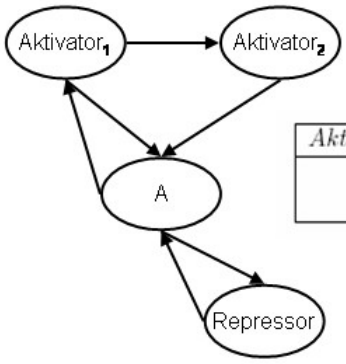
- *Setze* $(G, \boldsymbol{\theta}) = (G_0, \hat{\boldsymbol{\theta}}_0)$
- *Wiederhole bis Konvergenz*
  - *Wähle zufällig zwischen Punkte 1 und 2 aus.*
    - ***1.*** *Setze* $\hat{\boldsymbol{\theta}} = EM(G, \hat{\boldsymbol{\theta}}, D)$
    - ***2.*** *Bestimme für jede zufällige lokale Strukturänderung*[8] *das involvierte Element* X_i *und die neuen Eltern* $\mathbf{Pa}_{G'}(X_i)$ *und berechne die erwarteten Häufigkeiten:*

      $$\bar{N}_{ij_ik_i} = \sum_{m=1}^{M} P(X_i = k_i, \boldsymbol{Pa}_{G'}(X_i) = j_i | d_m, G, \hat{\boldsymbol{\theta}})$$

      Berechne anhand von $\bar{N}_{ij_ik_i}$ *den Score. Definiere* G *als die Struktur mit maximalen Score neu und berechne* $\hat{\boldsymbol{\theta}}^{MLE}$ *aus* $\bar{N}^{G}_{ij_ik_i}$.

Indem die Berechnung der optimalen Parameter aus der inneren Schleife herausgenommen ist, werden die EM Iterationen gespart, die notwendig gewesen wären die aktuell zu evaluierende Struktur zu parametrisieren. Es werden vielmehr Schätzer,

[8] Lokale Strukturänderungen können das Entfernen, Hinzufügen oder die Änderung der Richtung einer Kante sein. Für DBN ist nur das Entfernen oder Hinzufügen einer Kante sinnvoll.



$Aktivator_1$	$P(Aktivator_2 = 1)$	$P(Aktivator_2 = 0)$
0	1	0
1	0	1

Abbildung 3.9: Ein CPT, der eine Boolesche Funktion repräsentiert und der Graph eines Booleschen Netzes, der aus der Übergangsstruktur des DBN in Abb. 3.6 resultiert.

die zu diesem Zeitpunkt optimal sind, vorausgesetzt und anhand von ihnen der Score für G' berechnet. Nach einigen Schritten kann dann mit der aktuellen Parameterschätzung und einer neuen optimalen Struktur neue optimale Parameterschätzer hergeleitet werden (1). Woraufhin wieder mit den neuen Parametern eine neue optimale Struktur gesucht wird (2). Es werden also einmal die Parameter und ein anderes mal die Struktur iterativ optimiert.
Im originalen SEM-Algorithmus ist der die lokale Strukturänderung bewertende Score der BIC Score [12]. Die Anwendung für den Likelihood Score ist daraus leicht ableitbar.

3.5 Zusammenhang zwischen DBN und Boolesche Netze

Die Übergangsstruktur eines DBN kann in den gerichteten Graphen eines Booleschen Netzes überführt werden. Denn sie beschreibt die Dynamik des Netzes über definierte Zeitpunkte. Ist die gleiche Struktur für unendlich viele Zeitpunkte gültig, dann repräsentiert sie die gleiche Dynamik wie der Graph eines Booleschen Netzes. Die Kanten eines Booleschen Netzes definieren die Beziehungen zwischen zwei aufeinander folgenden Zeitpunkten und die Knoten die Zustände der Gene zu einem Zeitpunkt. Diese Interpretationen sind analog zu denen einer Übergangsstruktur eines DBN. Wird beispielsweise angenommen, dass die Übergangstruktur aus Abb. 3.6 für unendliche viele Zeitpunkte gilt, dann kann diese in den Graph eines Booleschen Netzes überführt werden (vgl. Abb. 3.9). Enthalten alle Zeilen eines CPT nur einen einzigen positiven Wert, also 1, dann ist die CPT auf eine deterministi-

sche Regel reduziert worden (vgl. Abb. 3.9). Die hier aufgeführte CPT beschreibt nichts anderes als die Boolesche Funktion $Aktivator_2 = \neg Aktivator_1$. Dynamische Bayessche Netze, deren Zufallsvariablen nur zwei diskrete Werte annehmen können und deren CPT's nur die Wahrscheinlichkeiten 1 und 0 enthalten, sind äquivalent zu Booleschen Netzen [35]. Aus diesem Grund ist der Versuch ein Dynamisches Bayessches Netz zu lernen gerechtfertigt, auch wenn das in Wirklichkeit zugrundeliegende System ein Boolesches Netz ist. Insbesondere wenn die Trainingsdaten gestört sind.

Kapitel 4

Reverse Engineering Strategien

Der erste Abschnitt dieses Kapitels befaßt sich damit, wie mit REVEAL ein Boolesches Netz aus unvollständigen Trainingsdaten erlernbar ist und welche Lösungen für auftretende Probleme geeignet sind.
In den weiteren Abschnitten wird auf allgemeinere Reverse Engineering Strategien eingegangen, die zum Teil die Probleme von REVEAL lösen. Das Ziel ist es nun nicht mehr ein Boolesches Netz, sondern ein Dynamisch Bayessches Netz zu lernen. Eine mögliche Heuristik für die Suche nach einem optimalen Modell beginnt mit einer initialen Struktur und Parameterverteilung und führt eine lokale Suche, z.B. *random restart Hill Climbing*, durch.
Eine zweite Heuristik analysiert, ob Teilmengen von Elementen unabhängig von den anderen Elementen einen signifikanten Einfluss auf ein Outputelement haben. Die Vereinigung der signifikanten Inputelemente bilden die Elternmenge für ein Outputelement. Die Parameter werden nachträglich für die gelernte Struktur geschätzt. Die Optimierung der Parameter und der Struktur erfolgt getrennt, im Gegensatz zur ersten Heuristik.
Diese Heuristiken werden in den ersten beiden Abschnitten dieses Kapitels analysiert. Dabei wird sich darauf konzentriert, wie durch sie korrekte Strukturen erlernbar sind. Je akkurater eine Struktur identifiziert wird, desto weniger Parameter sind notwendig. Zuerst wird gezeigt, dass die erste Heuristik für Trainingsdaten nach Definition 1.4 ungeeignet ist. Die zweite eignet sich besser, denn die gesehenen Anfangszustände können unabhängig von den fehlenden Beobachtungen untersucht werden. Auf ihre Analyse wird im zweiten Abschnitt genauer eingegangen.
Der dritte Abschnitt befaßt sich mit der Frage, ob die Parameter geschätzt werden können, falls die Struktur bekannt ist.
Die Analysen erfolgen auf optimal gegebene Trainingsdaten, d.h. sie sind ungestört und es befindet sich keine Zelle im Attraktor. Es ist jedoch anzunehmen, dass Zellen zum Zeitpunkt der Expressionsanalyse in einem Attraktor sind. Dies hat zur Folge,

dass sich ihre genetischen Zustände nur in einem begrenzten Raum bewegen und für das Lernen des Modells wenige unterschiedliche Expressionszustände zur Verfügung stehen. Die Auswirkungen auf die Lernalgorithmen werden im vierten Abschnitt behandelt.
Im letzten Abschnitt wird untersucht, wie die Algorithmen auf gestörte Trainingsdaten angewendet werden können.

4.1 Evaluierungsmaße

Wie gut ein Reverse Engineering Algorithmus eine Modellstruktur rekonstruiert, wird mit den Wahrscheinlichkeiten P_{Input}, $P_{identified}(k)$ und mit $P_{positive}(k)$ bewertet.

Definition 4.1 ($P_{Input}(k)$) *P_{Input} ist die Wahrscheinlichkeit, dass die k Inputelemente eines Outputelementes nicht korrekt identifiziert wurden.*

P_{Input} wird entweder **M**, der Anzahl Beobachtungen einer Kombination von bekannten Anfangszuständen oder **Z**, der Anzahl der Zustandsübergangspaare gegenübergestellt.

Definition 4.2 ($P_{identified}(k)$) *$P_{identified}(k)$ ist definiert als der Anteil von originalen Elternelementen für ein Outputelement, die auch in der gelernten Modellstruktur vorhanden sind. Das Outputelement hat k Inputelemente.*

Definition 4.3 ($P_{positive}(k)$) *$P_{positive}(k)$ ist definiert als der Anteil der Eltern eines Outputelementes in der gelernten Modellstruktur, die auch Elternelemente in der originalen Modellstruktur sind. Sie sind die echt positiven Eltern. Das Outputelement hat k Inputelemente.*

$P_{positive}(k)$ kleiner als 1, weist auf *falsch positive* Eltern hin, d.h. auf identifizierte Inputelemente, die in der originalen Struktur keine Eltern sind. Hat $P_{identified}(k)$ einen Wert kleiner als 1, bedeutet dies, dass noch nicht jedes Elternelement erkannt werden konnte. Es existieren noch *falsch negative* Eltern. Ein Reverse Engineering Algorithmus sollte hohe Werte für $P_{positive}(k)$ und $P_{identified}(k)$ besitzen.
Nachdem eine Parameterverteilung rekonstruiert wurde, soll bewertet werden, wie wahrscheinlich das gelernte Modell gegeben der Daten ist. Dafür wird mit der *Relativen Entropie* analysiert, wie das gelernte Modell die Daten generieren kann, die von dem originalen Modell erzeugt wurden.

Definition 4.4 (Relative Entropie [21])

$$
\begin{aligned}
H_{relative}(P,Q) &= \sum_{x_1,\ldots,x_N} P(X_1,\ldots,X_N)\log_2\frac{P(X_1,\ldots,X_N)}{Q(X_1,\ldots,X_N)} \qquad (4.1)\\
&= \sum_{i=1}^{N}\sum_{j_i=1}^{q_i}\sum_{k_i=1}^{r_i} P(X_i=k_i,\mathbf{Pa}(X_i)=j_i)\log_2\frac{P(X_i=k_i|\mathbf{Pa}(X_i)=j_i)}{Q(X_i=k_i|\mathbf{Pa}(X_i)=j_i)}
\end{aligned}
$$

$P(X_i = k_i|\mathbf{Pa}(X_i) = j_i)$ und $P(X_i = k_i, \mathbf{Pa}(X_i) = j_i)$ sind Verteilungen für die Testmenge mit Daten von dem originalen Modell und $Q(X_i = k_i|\mathbf{Pa}(X_i) = j_i)$ für die Parameter im gelernten Modell. P ergibt sich aus der Testmenge, indem der Maximum Likelihood Schätzer für jeden Parameter berechnet wird. Die relative Entropie ist immer positiv oder gleich 0. Kleine Werte der relativen Entropie korrespondieren zu einer gelernten Parameterverteilung, die der originalen Verteilung sehr ähnelt und damit die Daten gut erklären kann. Nur in Fällen, bei denen die Verteilungen identisch sind, wird die relative Entropie gleich 0.

4.2 REVEAL

Es wird analysiert, wie mit REVEAL ein Boolesches Netz aus unvollständigen Daten gelernt werden kann. Sind unvollständige Trainingsdaten gegeben, bedeutet dies, dass REVEAL nicht das gesamte Gitter in Abb. 3.4 erfolgreich durchläuft. Die Entropie einer Kombination von Inputelementen, mit mindestens einem Inputelement, dessen Zustand unbekannt ist, kann nicht berechnet werden. Damit REVEAL funktionsfähig bleibt, ist es notwendig, die fehlenden Werte zu ergänzen. Würde das zugrunde liegende Modell bekannt sein, wäre es möglich, die fehlenden Daten zu vervollständigen. Es könnte der wahrscheinlichste Wert das fehlende Datum ersetzen und die so entstehenden Daten würden wie vollständig gegebene Beobachtungen behandelt. Dies würde aber bedeuten, dass REVEAL nicht zwischen beobachteten sicheren Daten und unbeobachteten unsicheren Daten unterscheidet. Die unbeobachteten Zuweisungen werden mit der gleichen Konfidenz bewertet, wie Zuweisungen, die gesehen wurden. Eine bessere Variante ist, die Wahrscheinlichkeiten für die Beobachtung der fehlenden Werte zur Ergänzung heranzuziehen.
Die Schwierigkeit liegt jedoch darin, dass das Modell nicht bekannt ist, sondern gelernt werden soll. Damit es gelernt werden kann, sind Daten notwendig, die viel Information liefern. Der konkrete Typ von unvollständigen Daten, der in dieser Arbeit untersucht wird, hat aber nur zu einer begrenzten Anzahl von Elementen zur Zeit t lückenlose Information, ungestörte Daten vorausgesetzt. Deshalb ist es nicht trivial, das Modell zu lernen.

Für die Analyse von REVEAL und unvollständigen Daten soll eine konkrete Zuordnung eines Zustandes zu einem fehlenden Datum geschehen. Die Zuordnung erfolgt ohne Berücksichtigung der beobachteten Daten in einem Trainingsvektor. Sie ist zufällig und deshalb sehr fehleranfällig. Es wird erwartet, dass es für REVEAL unmöglich ist, die korrekte Struktur zu finden. Die Störung durch falsche Ergänzungen wird zu groß sein. Andere Methoden, die von einer Startkonfiguration aus iterativ das Modell lernen und die Daten entsprechend des aktuellen Modells ergänzen, werden in einem späteren Abschnitt betrachtet.

4.2.1 Implementierung

Die mit REVEAL durchgeführten Computersimulationen implementieren 300 Boolesche Netze. Jedes Boolesche Netz besteht aus 20 Elementen. Die Kanten zwischen den Elementen sind so konstruiert, dass die Anzahl der Inputelemente zu gleichen Anteilen 1, 2, 3 oder 4 beträgt. Die Auswahl der Inputelemente erfolgt zufällig. Die Boolesche Regel zu einem Element und seinen Inputelementen ist zufällig aus den k_{eff}-Regeln ausgewählt[1]. Ein so konstruiertes Boolesches Netz repräsentiert das Genregulationsnetzwerk einer Zelle. Auf eine Population von gleichartigen Zellen wird REVEAL angewendet und die Wahrscheinlichkeit $P_{Input}(k)$ $(k \in 1, 2, 3, 4)$ bestimmt. Ob die Inputelemente für ein Element richtig gefunden werden, hängt bei einer vollständig gegebenen Zustandsübergangstabelle im wesentlichen von der Stichprobengröße ab. Der Algorithmus in Pseudocode lautet wie folgt:

FOR $b := 1$ *TO* $b = 300$

Create Boolean Network b_{net} *with* N *nodes randomly*
Create Z *cells with* b_{net}
Assign unique initial state to each cell
FOR EACH cell i
FOR EACH $node_j$
IF state of $node_j$ *in* b_{net_i} *is known*
$transition\_table[t] \leftarrow$ *state of* $node_j$
ELSE
$transition\_table[t] \leftarrow$ *choose randomly between states* 0 *and* 1
Update cell once
FOR EACH $node_j$
$transition\_table[t+1] \leftarrow$ *state of* $node_j$

[1] Siehe Kapitel 3.2

D:=transition_table
$G \leftarrow$ *REVEAL(D)*
Evaluate G

Calculate $P_{Input}(k)$

PROCEDURE REVEAL:

FOR $j := 1$ *TO* $j = N$

FOR EACH combination $\mathbf{Pa}_j$ *of* j *input elements*

FOR ALL output elements o

IF $|MI(\mathbf{Pa}_j, o) - H(o)| < \epsilon$ */*Choose an adequate* ϵ **/*

$parents[o] \leftarrow \mathbf{Pa}_j$

$G \leftarrow parents[o]$

Return G

Eine erste Simulation setzt voraus, dass die Zustandsübergangstabelle vollständig gegeben ist, d.h. ohne fehlende Daten. Die Grundmenge, aus der die Startzustände für die linke Spalte der Zustandsübergangstabelle gezogen werden, ist die Menge aller 2^N globalen Zustände bei N Elementen. Die Zustände werden zufällig gezogen, jedoch so, dass ein globaler Zustand nicht mehrmals auftritt. Es ist also keine Redundanz bei den globalen Zuständen gegeben. Das Ergebnis bringt keine neuen Erkenntnisse, da es nur einer Reimplementierung des originalen REVEAL in [28] entspricht. Der Graph in Abbildung 4.1 zeigt, dass sich P_{Input} logarithmisch zu den Zustandsübergängen verhält. Als nächstes sind nun Zustandsübergangstabellen gegeben, bei denen in der linken Spalte nicht mehr alle Elemente Zustände zugewiesen bekamen. Es gibt Elemente, denen in keiner Zeile ein Zustand zugeordnet ist[2]. Alle anderen Elemente besitzen in jeder Zeile einen gemessenen Zustand. Die fehlenden Daten werden mit konkreten Werten ergänzt. Ein fehlendes Datum wird zufällig auf 1 oder 0 gesetzt, wobei beide Werte als gleich wahrscheinlich vorausgesetzt sind. Diese Annahme ist plausibel, wenn die Zelle zur Zeit t alle 2^N Zustände besitzen kann.

4.2.2 Ergebnisse

Anhand von Simulationen soll herausgefunden werden, wie REVEAL falsch ergänzte Werte verarbeitet. Die Abbildung 4.2 zeigt das Ergebnis für eine Simulation mit 19 gemessenen Genen zur Zeit t. Wie zu sehen, kann REVEAL schon bei einem

[2]Siehe Definition 1.4 für unvollständige Daten in dieser Arbeit.

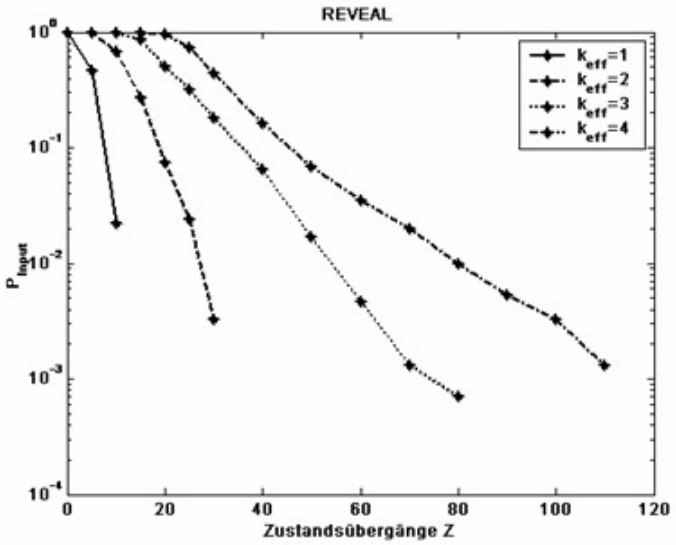


Abbildung 4.1: Wahrscheinlichkeit die Inputelemente nicht korrekt zu finden, wenn alle Zustände zu jedem Zeitpunkt bekannt sind ($N = 20$, $K = 4$). Der kleinste erkennbare Fehler ist $\frac{1}{1200}$, denn in jedem Netz gab es vier Elemente mit k Eltern und 300 Netze wurden simuliert. Ab 10 Zustandsübergängen wird das Inputelement für alle Regeln mit $k_{eff} = 1$ identifiziert. Für $k_{eff} = 2$, $k_{eff} = 3$ und $k_{eff} = 4$ wird dies erste bei ca. 30, 80 und 110 Zustandsübergängen erreicht.

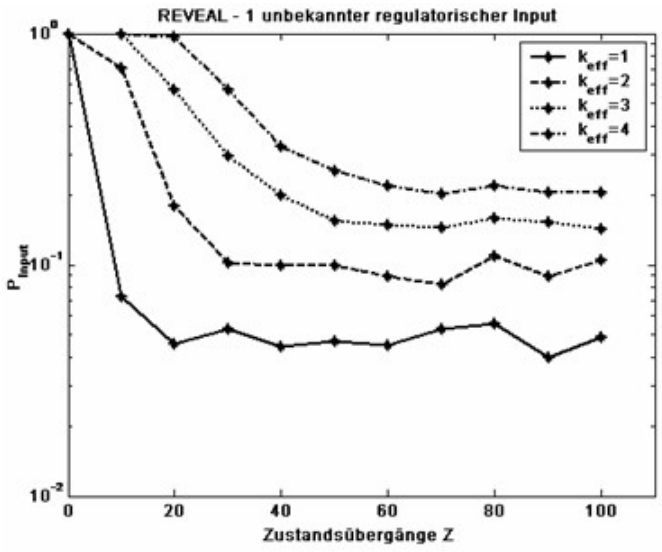


Abbildung 4.2: Wahrscheinlichkeit die Inputelemente nicht korrekt zu finden, falls zum Zeitpunkt t der Zustand eines Elementes nicht messbar war ($N = 20$, $K = 4$). Die Konvergenzwerte können berechnet werden (vgl. Tab. 4.1).

regulierenden Gen, dessen Zustände nicht gegeben sind, den Graphen des Booleschen Netzes nicht mehr korrekt bestimmen. Auch bei steigender Anzahl der Zustandsübergänge verbessern sich die Wahrscheinlichkeiten $1 - P_{Input}(k)$ nicht mehr. Die Anzahl der Inputelemente mit unbekanntem Zustand legt $P_{Input}(k)$ fest. Für die Erklärung dieses Ergebnisses sei folgendes Gedankenexperiment eingeführt: Es sei ein Boolesches Netz mit Elementen, aber ohne Kanten gegeben. Die Eltern für ein Element X werden konstruiert, indem aus den Elementen dieses Netzes zufällig k Elemente ausgewählt werden. Ist einmal ein Element als Input ausgewählt, wird es kein zweites Mal für X gezogen. Dieses Experiment entspricht einem statistischen Urnenmodell mit N unterschiedlichen Kugeln ohne Kugeln zurückzulegen. Unter diesen N Elementen gibt es einige, deren Zustände nicht gemessen wurden. Die Frage nach der Wahrscheinlichkeit, beim Ziehen von k Elementen mindestens eine Kugel mit unbekanntem Zustand zu erhalten, beantwortet die Frage nach der Wahrscheinlichkeit, dass REVEAL die k Inputelemente nicht identifizieren kann. Das Ereignis, mindestens ein Element mit unbekanntem Zustand zu ziehen, setzt sich aus mehreren Teilereignissen zusammen. Diese sind das Ereignis, ein Element mit unbekanntem Zustand zu ziehen, das Ereignis zwei Elemente mit unbekanntem Zustand zu ziehen, ... und das Ereignis k Elemente mit unbekanntem Zustand zu ziehen. Die Wahrscheinlichkeit für ein beliebiges Teilereignis läßt sich aus der Hypergeometrischen Verteilung bestimmen:

$$P(X = j) = \frac{\binom{u}{j}\binom{N-u}{k-j}}{\binom{N}{k}}. \tag{4.2}$$

Sie definiert die Wahrscheinlichkeit, für Element X genau j Inputelemente mit unbekanntem Zustand zu ziehen. N entspricht der Anzahl der Elemente im Netz und k der Anzahl der Ziehungen und somit der Anzahl der Elternelemente von X. Die Variable u ist die Anzahl der Elemente mit unbekanntem Zustand im Netz und j die Anzahl der Elternelemente mit unbekanntem Zustand. Die Wahrscheinlichkeit, dass das Element X mindestens ein Elternelement mit unbekanntem Zustand besitzt, ergibt sich wie folgt:

$$P_{Input}(k) = \sum_{j=1}^{u_{max}} \frac{\binom{u}{j}\binom{N-u}{k-j}}{\binom{N}{k}}. \tag{4.3}$$

N	u	k	$P_{Input}(k)$
20	1	1	$\frac{\binom{1}{1}\binom{19}{0}}{\binom{20}{1}} = 0.05$
20	1	2	$\frac{\binom{1}{1}\binom{19}{1}}{\binom{20}{2}} = 0.1$
20	1	3	$\frac{\binom{1}{1}\binom{19}{2}}{\binom{20}{3}} = 0.15$
20	1	4	$\frac{\binom{1}{1}\binom{19}{3}}{\binom{20}{4}} = 0.2$
12	2	1	$\frac{\binom{2}{1}\binom{10}{0}}{\binom{12}{1}} = 0.16$
12	2	2	$\frac{\binom{2}{1}\binom{10}{1}}{\binom{12}{2}} + \frac{\binom{2}{2}\binom{10}{0}}{\binom{12}{2}} = 0.318$
12	2	3	$\frac{\binom{2}{1}\binom{10}{2}}{\binom{12}{3}} + \frac{\binom{2}{2}\binom{10}{1}}{\binom{12}{3}} = 0.455$
12	3	1	$\frac{\binom{3}{1}\binom{9}{0}}{\binom{12}{1}} = 0.25$
12	3	2	$\frac{\binom{3}{1}\binom{9}{1}}{\binom{12}{2}} + \frac{\binom{3}{2}\binom{9}{0}}{\binom{12}{2}} = 0.455$
12	3	3	$\frac{\binom{3}{1}\binom{9}{2}}{\binom{12}{3}} + \frac{\binom{3}{2}\binom{9}{1}}{\binom{12}{3}} + \frac{\binom{3}{3}\binom{9}{0}}{\binom{12}{3}} = 0.618$

Tabelle 4.1: Beispielhafte Berechnungen für die Wahrscheinlichkeit, die Inputelemente mit REVEAL bei unvollständigen Trainingsdaten falsch zu bestimmen. Die ersten vier Zeilen enthalten die Wahrscheinlichkeiten für Abbildung 4.2.

Die Teilereignisse sind unabhängig und können deshalb aufsummiert werden. Die Variable u_{max} steht für die Anzahl der Ziehungen bei denen Elternelemente mit unbekannten Zuständen gezogen werden können und hängt von u und k ab:

$$u_{max} = \begin{cases} u & falls \quad u \leq k \\ k & falls \quad u > k \end{cases}$$

Die Tabelle 4.1 enthält Beispiellösungen für (4.3). Mit der Formel (4.3) können die Werte der y-Achse in Abbildung 4.2 ermittelt werden, bei denen die Sättigung der Kurven eintritt.

Ist der Zustand mindestens eines Elternelementes unbekannt, sind alle Elternelemente und somit die Kanten zu ihnen nicht identifizierbar! Durch falsche Ergänzungen der fehlenden Werte sind die Entropien $H(\mathbf{X}, Y_j')$ und $H(\mathbf{X})$ mit Fehlern belastet und die Gleichheitsbeziehung $H(\mathbf{X}, Y_j') = H(\mathbf{X})$ gilt nicht mehr. Daraus läßt sich schließen, dass die originale Version von REVEAL bei einer Ergänzung von konkreten Werten scheitert.
Ein möglicher Lösungsansatz ist leicht erkennbar. Bisher wurde nur eine einzige Zustandsübergangstabelle aufgebaut. Sie enthielt für $(N-u)$ Gene vollständig gegebene Zustände. Nach Definition 1.4 sind während eines Experimentes u Gene nicht messbar. Diese können sich bei verschiedenen Experimenten unterscheiden. Deshalb ist es möglich mehrere Experimente durchzuführen, bei denen jeweils eine andere Kombination von u Genen nicht gemessen wird. REVEAL hat, im Gegensatz zu oben, mehrere Zustandsübergangstabellen zur Verfügung, genau sind es $\binom{N}{N-u}$. Nun bleibt noch zu fragen, ob mindestens eine Kombination von k Elementen existiert, die nur Elemente mit gemessenen Zuständen enthält, um die Eltern eines Outputelementes mindestens einmal vollständig zu sehen. Gibt es so eine Kombination, kann REVEAL auf die korrespondierende Zustandsübergangstabelle angewendet werden und bei einer genügend großen Stichprobengröße die k Eltern erkennen. Es existieren also $\binom{N}{N-u}$ Kombinationen von gemessenen Genen, unter denen die Anzahl von Kombinationen, die k Gene umfassen, gesucht wird. Anders formuliert, wird nach den Möglichkeiten gesucht, eine Kombination mit k Elementen auf $(N-u)$ zu erweitern. Für eine solche Erweiterung stehen noch $(N-k)$ Elemente zur Verfügung und es müssen $((N-u)-k)$ Elemente hinzugefügt werden. Es gibt insgesamt

$$\binom{N-k}{(N-u)-k} \tag{4.4}$$

Möglichkeiten, k Elemente in Kombinationen der Größe $(N-u)$ zu finden. Zur Verdeutlichung soll ein kurzes Beispiel dienen. Ist $N = 5$ und $u = 1$, dann gibt es folgende Kombinationen mit 4 Elementen:

1234, 1235, 1245, 1345, 2345.

Jede Kombination mit 2 Elementen tritt dreimal auf.
Dieses Beispiel zeigt, dass es möglich ist für ein Element X, dessen Anzahl von Eltern $(N-u)$ Gene nicht überschreitet, mindestens eine Zustandsübergangstabelle zu erstellen, bei der alle k Eltern bekannt sind. Hat ein Element jedoch $k > (N-u)$ Inputelemente, müssen andere Strategien erarbeitet werden.
Die originale Version von REVEAL ordnet Inputelemente auf der Basis der Beziehung $MI(\mathbf{X}, Y_j') = H(Y_j')$ zu. Sie sagt aus, dass die Elemente $\mathbf{X}$ die Zustände des Outputelementes Y_j' vollständig bestimmen. Eine Abschwächung könnte nützlich

sein, um mit unvollständigen Trainingsdaten zu arbeiten. Anstatt nach den Elementen zu suchen, die das Outputelement vollständig bestimmen, muss bei unvollständigen Daten nach Elementen gesucht werden, die das Outputelement unvollständig, aber signifikant, festlegen. Die Bedeutung liegt nun darin zu fragen, ob

$$MI(\boldsymbol{X}, Y_j') > 0. \tag{4.5}$$

Ist die Mutual Information signifikant größer als 0, sind die Elemente in $\mathbf{X}$ Inputelemente von Y_j', denn es gilt: Elemente die unabhängig voneinander sind, haben eine Mutual Information von 0 (s. Kapitel 2).

Die Identifizierung eines Genregulationsnetzwerkes ist nicht allein durch das Finden der Struktur abgeschlossen, sondern es müssen auch die Regeln erkannt werden. Eine Boolesche Regel wird mit einer Regeltabelle repräsentiert, die zu jeder Belegung der Inputelemente eine eindeutige Belegung des Outputelementes zuordnet. In [28] werden die Regeltabellen aufgestellt, indem zu jeder Belegung der Inputelemente die korrespondierende Belegung des Outputelementes aus den Trainingsdaten herausgesucht und in der Regeltabelle zueinander in Beziehung gestellt wird. Sind die Trainingsdaten unvollständig, ist es unmöglich, die Regeltabellen auf diese Art und Weise zu konstruieren. Die Regeltabellen für Elemente mit $k \leq (N - u)$ Eltern sind ohne weitere Probleme anhand der gesehenen Daten von mehreren Experimenten aufstellbar. Für Elemente mit einer größeren Anzahl von Eltern sei angenommen, dass die korrekte lokale Struktur bekannt ist. Trotzdem ist es nicht möglich, die Regeltabelle anhand der Beobachtungen komplett zu erkennen. Denn es wurde eine vollständige Belegung aller Elternelemente nicht gesehen und deshalb scheitert die eindeutige Zuordnung zu einer Belegung des Outputelementes. Es ist also notwendig andere Methoden heranzuziehen. Eine davon ist der *EM-Algorithmus*[3].

Es wurde untersucht, wie sich REVEAL auf unvollständige Daten anwenden lässt. Das Ergebnis war, dass die Struktur und die Booleschen Regeln für Elemente bestimmt werden können, deren Anzahl von Eltern kleiner oder gleich der messbaren Elemente zur Zeit t ist. Dafür sind mehrere Experimente notwendig, die sich jeweils in den Elementen mit gemessenen Zuständen unterscheiden. Außerdem wurde gezeigt, dass es nicht möglich ist, mit REVEAL die Struktur und Regeln für Elemente zu identifizieren, die mehr Eltern besitzen, als während eines Experimentes gleichzeitig gemessen werden. Die Tatsache dass REVEAL Methoden aus der Künstlichen Intelligenz verwendet, zeigt den Weg für mögliche Lösungen auf. In der Künstlichen Intelligenz werden u.a. auch Maße aus der Informationstheorie benutzt, um Modelle von Beobachtungen aus der realen Welt zu lernen. Ein Beispiel dafür ist

[3] S. Kapitel 3.4.3

der *MIS Score*[4] für das Lernen von Strukturen. Letztendlich verwendet REVEAL nichts anderes als einen Spezialfall des MIS Scores, um die Eltern zu rekonstruieren. REVEAL ist deshalb nur eine Spezialisierung von schon bekannten Methoden [35]. Deshalb liegt es nahe, die in dieser Arbeit betrachteten besonderen Art von Trainingsdaten, mit Methoden aus der Künstlichen Intelligenz weiter zu analysieren. Eine weitere Motivation folgt aus der Tatsache, dass Boolesche Netze auch nur ein Spezialfall der Dynamischen Bayesschen Netze sind [35].

Bei der obigen Analyse von REVEAL wurden die unvollständigen Daten unabhängig von den gemessenen Zuständen ergänzt. Die Überzeugung des Beobachters über die Werte der fehlenden Daten wird aber von den gemessenen Zuständen verändert. Reverse Engineering Strategien für Dynamisch Bayessche Netze sind fähig diesen Sachverhalt zu berücksichtigen. Es gibt zwei mögliche Wege, einen Lernalgorithmus für ein DBN zu gestalten.
Einerseits wird von einer Startkonfiguration ausgegangen. Die Startkonfiguration wird anhand der Daten iterativ verändert. Die Idee ist, dass das Modell die Daten entsprechend den Parametern ergänzt und die Daten wiederum das Modell verbessern. Das Startmodell kann dabei zufällig gewählt oder anhand des Wissens von Experten aufgebaut werden.
Andererseits werden mehrere Experimente durchgeführt, die sich in den messbaren Elementen unterscheiden. Das Modell soll gelernt werden, indem charakterisiert wird, wie groß der Einfluss der aktuell gemessenen Inputelemente auf die Outputelemente ist. Eine Ergänzung der fehlenden Daten ist nicht notwendig, da nur der Einfluss von den gerade gemessenen Inputelementen auf das Outputelement charakterisiert werden soll.
Die zwei folgenden Abschnitte beschäftigen sich näher mit diesen beiden Sachverhalten.

4.3 SEM

4.3.1 Bewertungsfunktion

Als Bewertungsfunktion wird das Bayessche Informations Kriterium (BIC) gewählt. Es vermeidet komplexe Strukturen, indem eine Kante nur in das Modell integriert wird, wenn es die Likelihood Funktion gegenüber dem Strafterm rechtfertigt. Die beste Übergangsstruktur eines DBN, gegeben der Daten, soll aus Zustandsübergangstabellen wie Tabelle 4.2 gelernt werden.
Das Verfahren Erwartungswert Maximierung wird angewendet, um sinnvolle Schätzer für unbeobachtete Häufigkeiten $N_{ij_ik_i}$ und N_{ij_i} zu erhalten. Mit einem vorhandenen

[4]Siehe Kapitel 3.4.2

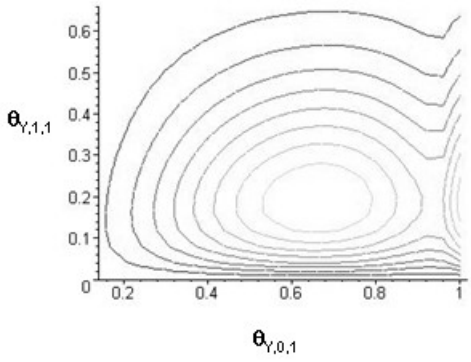
$G_1[t]$	$G_2[t]$	...	$G_N[t]$	$G_1[t+1]$	$G_2[t+1]$	...	$G_N[t+1]$
x	?	...	?	x	x	...	x
⋮	⋮	...	⋮	⋮	⋮	...	⋮
x	?	...	?	x	x	...	x
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
?	?	...	**x**	x	x	...	x
⋮	⋮	...	⋮	⋮	⋮	...	⋮
?	?	...	**x**	x	x	...	x

Tabelle 4.2: Aufbau der Zustandsübergangstabelle, wenn $l = 1$ (l bezeichnet die Anzahl der Elemente, deren Anfangszustand in einem Experiment gemessen werden kann.). x kennzeichnet einen gemessenen Zustand und „?“ einen nicht gemessenen. Jede Kombination von bekannten Anfangszuständen tritt genau M-mal auf. $G_i[t]$ ist eine Zufallsvariable, die den Expressionszustand von Gen i zur Zeit t definiert.

Modell wären die fehlenden Werte ohne Probleme ergänzbar. Mit vollständigen Daten dagegen, wäre es leicht, ein Modell zu lernen. Da jedoch beides nicht gegeben ist, wird mit einem zufällig gewählten Modell begonnen, mit dessen Hilfe sich die Erwartungen $\bar{N}_{ij_ik_i}$ und $\bar{N}_{ij_i}$ ergeben, falls ein involvierter Anfangszustand nicht gemessen wurde. Aus diesen Erwartungen wird dann wiederum ein neues Modell gelernt.

4.3.2 Suchraum

Die Suche beginnt mit einer zufällig erzeugten Übergangsstruktur und Parameterverteilung. Mit lokalen Strukturänderungen wird nach einem optimalen Modell gesucht, d.h. nach einer optimalen Übergangsstruktur mit optimalen Parametern. Lokale Strukturänderungen beinhalten das Hinzufügen oder Entfernen von Kanten zwischen einem Element und seinen aktuellen Eltern. In der Übergangsstruktur eines DBN ist es nicht sinnvoll, die Richtung einer Kante zu verändern. Die Änderung mit höchstem Score wird vollzogen (*Greedy Hill Climbing*). Aufgrund der Zerlegbarkeit der Bewertungsfunktion (vgl. Def. 3.16) ist es möglich, auf diese Art und Weise eine Struktur mit maximalem Score zu finden. EM stellt sicher, dass die Bewertungsfunktion auch bei fehlenden Daten zerlegbar bleibt. Ist die Suche in einem lokalen Optima gefangen, wird die Struktur zehnmal in 5 zufällig ausgewählten Kanten verändert und die Suche beginnt neu. Die $TABU$-Liste der nicht mehr zu betrachteten Strukturen wird auf 5 gesetzt. Die Suche wird abgebrochen, falls weitere Strukturänderung zu keinem besseren Score führen.



X	Y
?	1
?	1
?	1
?	1
?	0
1	1
1	0
1	0

Abbildung 4.3: Höhenlinien der gemittelten Likelihood Funktion für die Parameter $\theta_{Y,1,1}$ und $\theta_{Y,0,1}$ und festem Parameter $\theta_X = 0.1$ für das Bayessche Netz $X \rightarrow Y$. Es wurden 8 Beobachtungen gesehen, bei denen 5 für X fehlten. Es sind zwei Maxima zu erkennen.

4.3.3 Implementierung

Für jedes in-silico Experiment zur Genexpressionsanalyse an zwei aufeinanderfolgenden Zeitpunkten wird eine Population von in-silico Zellen mit gleichen Genregulationsnetzwerken betrachtet. Ein Genregulationsnetzwerk enthält $N = 8$ Gene und wird durch ein Boolesches Netz modelliert. Die maximal mögliche Zahl von Inputelementen für eine Boolesche Funktion beträgt $K = 4$. Die Anzahl der Elternelemente ist zu gleichen Teilen im Netz verteilt, d.h. jeweils zwei Outputelemente hatten 1, 2, 3 und 4 Eltern. Damit können Aussagen über die Identifizierbarkeit der Eltern abhängig von ihrer Anzahl k gemacht werden. Die Konstruktion eines Booleschen Netzes erfolgte zufällig. Es wurden aber nur Boolesche Funktionen ausgewählt, die nach St. Kauffman [24] biologisch relevant sind. Die Anzahl der Eltern für ein Element wird mit k_{bio} angegeben.
Zu Beginn eines Experimentes kann eine Zelle einen beliebigen Expressionszustand aus den 2^N möglichen Zuständen annehmen. Es wird ein Zellpopulation zusammengestellt, bei der jede Kombination von Genen mit l bekannten Anfangszuständen M-mal auftritt. Die bekannten Anfangszustände werden in eine Zustandsübergangstabelle eingetragen. Alle restlichen Zustände sind unbekannt. Danach wird gewartet bis sich jede Zelle einmal aktualisiert hat und der Expressionszustand ihres Genregulationsnetzwerkes vollständig charakterisiert ist. Der aktuelle Zustand wird in die Spalten für die Folgezustände in der Zustandsübergangstabelle eingetragen. Das Ergebnis eines Experimentes ist eine Zustandsübergangstabelle, wie sie in Abb. 4.2 für $l = 1$ zu sehen ist. Auf Basis solcher Zustandsübergangstabellen wird versucht eine optimale Modellstruktur zu finden. Für die durchgeführten Simulationen wurde die SEM Implementierung von [16] genommen. Der Algorithmus in Pseudocode lautet

wie folgt:

FOR $b := 1$ *TO* $b = 150$

Create Boolean Network b_{net} *with* N *nodes randomly*
FOR EACH combination $\mathbf{Pa}_l$ *of* l *known input elements*

Create M *cells with* b_{net}
Assign unique initial state to each cell
FOR EACH cell i

FOR EACH $node_j$

IF state of $node_j$ *in* b_{net_i} *is known*

$transition\_table[t] \leftarrow$ *state of* $node_j$

ELSE

$transition\_table[t] \leftarrow$ *label state as unknown*

Update cell once
FOR EACH $node_j$

$transition\_table[t+1] \leftarrow$ *state of* $node_j$

D:=transition_table
$G \leftarrow$ *SEM(D)*
Evaluate G

Calculate $P_{Input}(k)$

PROCEDURE SEM [16]:

Set $G = G_0$ *and* $\boldsymbol{\theta} = \boldsymbol{\theta_0}$
Repeat until convergence

Choose randomly between

Optimization of parameters:

$\theta = EM(G, \boldsymbol{\theta}, D)$

Optimization of structure:

Set TABU list to 5 *structures*
Determine for each local change in G *the new structure* G'
If G' *not in TABU determine the element* X_i *and it's parents* $\mathbf{Pa}(X_i)$ *that were involved in the local change*
Calculate for each local change expected counts $\bar{N}_{ij_ik_i}^{G'}$ *and BIC score*
$G := G'$ *with local change which had maximal score*

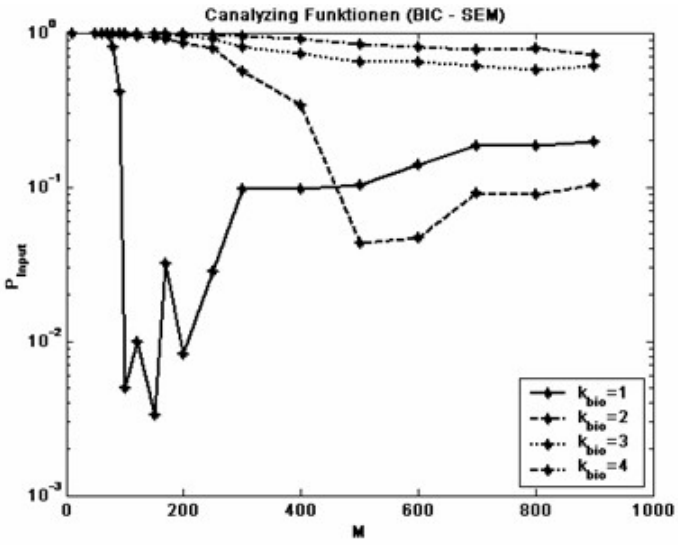


Abbildung 4.4: SEM (l=1, K=4).

IF no change in score

Change 10 *times* 5 *edges of* G *randomly and restart search*

Define $\boldsymbol{\theta}_G$ *using MLE over* $\bar{N}^G_{ij_ik_i}$

Return G

4.3.4 Ergebnisse

Die Likelihood Funktion der Daten D ist aufgrund der nicht beobachteten Werte eine Mischung aus mehreren Likelihood Funktionen. Eine Funktion steht für jede mögliche Ergänzung von D. Für N Variablen, deren Anfangszustände nicht in Gruppen von l Variablen zusammen beobachtet werden können, ergeben sich Trainingsvektoren, bei denen immer $(N-l)$ Beobachtungen der Anfangszustände fehlen. h bezeichnet unbeobachtete Werte und läuft über alle $(2^{(N-l)})^M$ möglichen Ergänzungen in der gesamten Datenmenge, o sind die gesehenen Werte. $P(h|o, \boldsymbol{\Theta}, G)$ ist die Wahrscheinlichkeit für das Auftreten einer Ergänzung h, gegeben der beobachteten Daten. Die Komplexität möglicher Ergänzungen wächst exponentiell mit Anzahl fehlender Beobachtungen. Als Resultat ist eine Likelihood Funktion mit mehreren lokalen Maxima möglich, in die der Optimierungsprozess konvergieren kann. Z.B. ist ein DBN $X \rightarrow Y$ und eine Zustandsübergangstabelle mit 8 Vektoren gegeben, bei denen X fünfmal unbekannt ist (vgl. Abb. 4.3). Dieses Bayessche Netz besitzt die Parameter θ_X, $\theta_{Y,1,1}$ und $\theta_{Y,0,1}$. Die gemittelte Likelihood Funktion für die Parameter

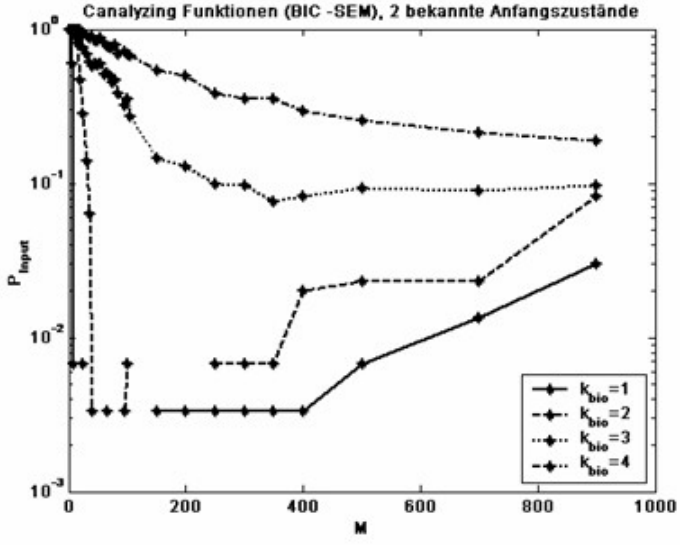


Abbildung 4.5: SEM (l=2, K=4). Der kleinste erkennbare Fehler ist $\frac{1}{300}$, denn in jedem Netz gab es zwei Elemente mit k Eltern und 150 Netze wurden simuliert. Fehlende Datenpunkte bedeuten, dass in allen 150 Netzen die Eltern für k_{bio} korrekt identifiziert wurden.

$\theta_{Y,1,1}$ und $\theta_{Y,0,1}$, falls $\theta_X = 0.1$, ist in Abbildung 4.3 zu sehen. In diesem Fall hat die Likelihood Funktion schon bei fünf fehlenden Werten für X zwei lokale Maxima.
Erwartungswert Maximierung (EM) approximiert die gemittelte Likelihood Funktion lokal an der aktuellen Position θ mit der Likelihood, entsprechend den erwarteten Häufigkeiten. Fehlt jedoch ein signifikanter Anteil von Beobachtungen, ist es schwierig, eine sinnvolle Extrapolation für die originalen Daten zu erhalten. Je mehr Beobachtungen fehlen, desto mehr Fehler treten bei der Bewertung der Ergänzungen auf. Beobachtungen für Kombinationen mit mindestens $l+1$ Anfangszuständen und einem Folgezustand sind in den Trainingsdaten nicht vorhanden. Auf ihre Häufigkeit in den originalen Daten zu schließen, ist daher unmöglich.
Das bedeutet, dass es für jeden Trainingsvektor 2^{N-l} mögliche Ergänzungen gibt, falls die Variablen zwei Zustände annehmen können. Daraus folgt, dass es für M Trainingsvektoren $(2^{(N-l)})^M$ Möglichkeiten gibt. Die gemittelte Likelihood Funktion ist

$$L(o|\mathbf{\Theta}, G) = \sum_h P(h|o, \mathbf{\Theta}, G) L(o, h|\mathbf{\Theta}, G). \tag{4.6}$$

Diese Überlegungen lassen vermuten, dass es nicht möglich ist, ein gutes Modell zu rekonstruieren. Eine Erhöhung des Stichprobenumfangs führt nicht nur zu einer Erhöhung des Wissens für die Variablen mit gesehenen Anfangswerten, sondern

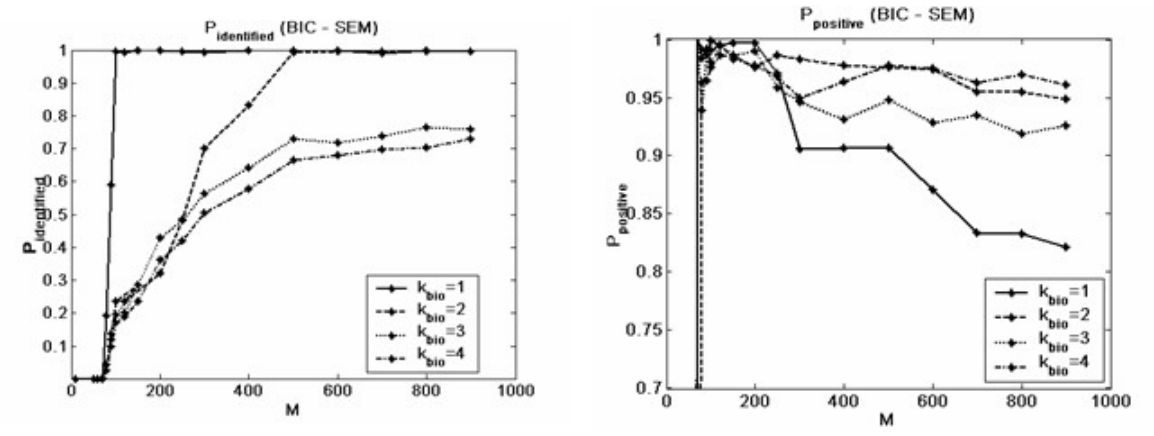


Abbildung 4.6: SEM - $P_{identified}(k)$ und $P_{positive}(k)$ (l=1, K=4). Für Outputelemente mit $k_{bio} = 1$ und $k_{bio} = 2$ werden alle Inputelemente mit einer hohen Wahrscheinlichkeit korrekt gefunden, aber auch falsch positive Inputelemente.

auch des Unwissens für alle anderen Variablen. Daraus folgt, dass für jeden neuen Trainingsvektor mit fehlenden Beobachtungen die Form der gemittelten Likelihood komplizierter und somit die Optimierung schwieriger wird. Zusätzliche Daten müssen deshalb nicht unbedingt zu einer Verbesserung führen.
Abbildungen 4.4 und 4.6 zeigen die Ergebnisse von Simulationsstudien, wenn der Anfangszustand nur eines Gens bekannt ist. Die Wahrscheinlichkeit, die Eltern für Elemente mit $k_{bio} = 1$ nicht korrekt zu lernen, verringert sich anfangs. Ab einer bestimmten Stichprobengröße steigt sie wieder. Die Graphen für $P_{identified}(k)$ und $P_{positive}(k)$ lassen erkennen, dass ab einer Stichprobengröße von 200 das Inputelement zwar korrekt erkannt wird, aber zusätzlich auch falsch positive Eltern. Die Erklärung ist darin zu sehen, dass aufgrund der fehlenden Beobachtungen die Likelihood gegenüber dem Strafterm die Integration einer zusätzlichen Kante in das Modell fälschlicherweise rechtfertigt. Es werden zu viele Eltern identifiziert, denn die Daten liefern keine Information über Kombinationen mit mehr als einem Inputelement und einem Outputelement. Das Modell wird deshalb nicht untermauert aber auch nicht verworfen. Analoges Verhalten ist bei der Wahrscheinlichkeit $P_{Input}(k_{bio} = 2)$ zu beobachten. Die Eltern von Elementen mit $k_{bio} = 2$ werden korrekt gefunden, aber außerdem auch falsch identifizierte Inputelemente. Mit Erhöhung der Stichprobengröße vergrößert sich die Wahrscheinlichkeit, falsch positive Inputelemente in die Menge der Eltern für ein Element X_i aufzunehmen. Es verbessern sich geringfügig die Wahrscheinlichkeiten, die Eltern für Elemente mit $k_{bio} = 3$ und $k_{bio} = 4$ zu finden. Je komplexer die gelernte Übergangsstruktur ist, desto wahrscheinlicher wird es, die Eltern für Elemente mit vielen Inputelementen identifiziert zu haben.

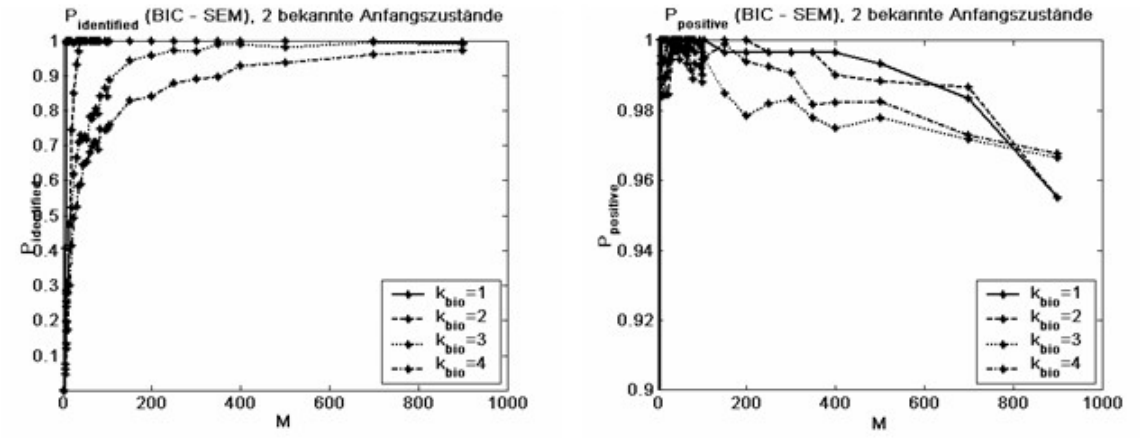


Abbildung 4.7: SEM - $P_{identified}(k)$ und $P_{positive}(k)$ (l=2, K=4). Es werden mit einer hohen Wahrscheinlichkeit die Inputelemente korrekt identifiziert, aber auch falsch positive Inputelemente.

Unter der Annahme, dass in einem biologischen Experiment die Anfangszustände von zwei Genen gemeinsam bekannt sind, verhält sich der Reverse Engineering Algorithmus ähnlich (Abbildungen 4.5 und 4.7). Die zusätzlichen bekannten Anfangszustände bewirken, dass auch für $k_{bio} = 3$ und $k_{bio} = 4$ alle Eltern gefunden werden. Andererseits werden aber auch sehr oft falsch positive Eltern identifiziert.
Die Anwendung von SEM auf Zustandsübergangstabellen wie Tabelle 4.2 ist keine gute Strategie. Die fehlenden Beobachtungen stellen ein Problem dar, was so nicht gelöst werden kann. Der hohe Anteil an fehlenden Beobachtungen und das Problem, dass Kombinationen von mehreren Inputelementen nicht zusammen gesehen werden, führen zu einem Scheitern von EM[5]. EM bzw. SEM konvergieren in lokale Maxima und verlassen diese auch durch zufällige Strukturänderungen und erneute Suche nicht. Eine gute Wahl der Startparameter könnte womöglich zu einer Verbesserung führen. Die Verteilungen der Anfangszustände sind durch die Expressionsanalyse einer Teilpopulation von Zellen erhaltbar und könnten die Wahl der Startparameter für die Inputelemente erleichtern. Aber für die Startparameter der Übergangswahrscheinlichkeiten sind weiterhin keine Anhaltspunkte vorhanden. Sie müssen durch eine zufällige Wahl oder durch das Wissen von Biologen gewählt werden.
Das Scheitern von EM ist nicht überraschend. Beobachtungen für Kombinationen von Inputelementen, die mehr Elemente enthalten, als Anfangszustände bekannt sind, werden nicht gesehen. Aus nicht gemachten Beobachtungen, sind auch keine Rückschlüsse auf das Modell möglich.

[5]Dies gilt auch für $l = 2$, wenn Kombinationen von mehr als zwei Inputelementen nicht gemeinsam beobachtet werden.

4.4 Partielles Lernen der Struktur

Im letzten Abschnitt wurde erläutert, weshalb SEM für das Lernen eines Modells bei unvollständig gegebenen Daten nach Definition 1.4 ungeeignet ist. Das wesentliche Problem liegt in möglichen Fehlern bei der Extrapolation der fehlenden Werte. Es ist notwendig, die fehlenden Beobachtungen zu umgehen. Das wird durch eine Reverse Engineering Strategie erreicht, in der jedes biologische Experiment separat von den anderen betrachtet wird. Es ergeben sich Zustandsübergangstabellen, wie in Abbildung 4.3 zu sehen. Die Übergangsstruktur des DBN folgt aus der Vereinigung aller gelernten Eltern, deren Einfluss jeweils unabhängig voneinander als signifikant bewertet wurde. Diese Herangehensweise hat allerdings den Nachteil, dass individuelle Fehlentscheidungen bei der separaten Analyse zu einem ungenauen Modell führen können und außerdem nicht mehr der gesamte Suchraum zur Verfügung steht.

4.4.1 Bewertungsfunktion

Als Bewertungsfunktion wurde der MIS Score in Verbindung mit dem χ^2-Test verwendet, um die Stärke des Einflusses zu quantifizieren. Der BIC Score ist eine äquivalente Bewertungsfunktion (vgl. Anhang A). Ein Strafterm bevorzugt weniger komplexer Modelle. Die Wahl fiel auf den MIS Score. Er ermöglicht, auf einfache Weise eventuelles Vorwissen über die Modellstruktur zu integrieren[6].
Im Abschnitt 4.2 wurde erläutert, dass REVEAL einen Spezialfall des MIS Score als Bewertungsfunktion benutzt, um die Abhängigkeit zwischen Netzwerkelementen zu charakterisieren. Es werden nur dann Kanten von $\mathbf{Pa}(X_i)$ [7] nach X_i im Modell aufgenommen, falls X_i und die Variablen in $\mathbf{Pa}(X_i)$ immer kovariieren, die Mutual Information also maximal ist:

$$MI(X_i, \mathbf{Pa}(X_i)) = H(X_i) \quad . \tag{4.7}$$

Diese Methode, die Elternelemente zu bestimmen, ist nur dann geeignet, wenn die Anfangszustände aller Eltern bekannt sind und ungestörte Trainingsdaten vorliegen[8]. Sobald die Trainingsdaten den Anfangszustand eines Elternelementes nicht enthalten, wird die Mutual Information nicht maximal und die Identifizierung der Eltern scheitert. Allerdings kann eine Abschwächung der Gleichheitsbeziehung (4.7) zu besseren Ergebnissen führen. Gilt

$$MI(X_i, \mathbf{Pa}(X_i)) \leq H(X_i) \quad \wedge \quad MI(X_i, \mathbf{Pa}(X_i)) > 0, \tag{4.8}$$

[6] Siehe Kapitel 5.
[7] $\mathbf{Pa}(X_i)$ bezeichnet die Eltern von X_i in der aktuell zu evaluierenden Teilstruktur.
[8] Siehe Abschnitt 4.7.

sind X_i und $\mathbf{Pa}(X_i)$ abhängig und es ist gerechtfertigt, Kanten von $\mathbf{Pa}(X_i)$ nach X_i einzuführen. Die so identifizierten Mengen $\mathbf{Pa}(X_i)$ erklären die Zufallsvariable X_i nicht immer vollständig, aber teilweise. Da $H(X_i)$ der maximale mögliche Wert für $MI(X_i, \mathbf{Pa}(X_i))$ ist, reicht es letztendlich aus, nach

$$MI(X_i, \mathbf{Pa}(X_i)) > 0 \tag{4.9}$$

zu fragen.
Die aus Trainingsdaten berechnete Mutual Information $\hat{MI}(X_i, \mathbf{Pa}(X_i))$ ist jedoch immer nur ein Schätzer für die wahre Mutual Information $MI(X_i, \mathbf{Pa}(X_i))$ der Grundmenge. Eine geschätzte Mutual Information die größer 0 ist, darf nicht ohne weiteres als Beleg für Abhängigkeiten zwischen Zufallsvariablen gelten. Ein statistischer Test muss herangezogen werden, um zu prüfen, ob die Ungleichung (4.9) signifikant größer 0 ist und nicht nur aufgrund von geringfügigen Abhängigkeiten in den Trainingsdaten eine positive *Mutual Information* festgestellt wurde. Die Mutual Information kann mit der χ^2-Verteilung approximiert werden (vgl. 3.19). Anhand der tabellierten Werten für die χ^2-Verteilung wird getestet, ob die Nullhypothese

$$H_0: \quad MI(X_i, \mathbf{Pa}(X_i)) = 0 \tag{4.10}$$

abgelehnt werden kann[9]. Eine Ablehnung bedeutet, dass die Zufallsvariablen nicht unabhängig sind und sich für die Alternativhypothese, $\mathbf{Pa}(X_j)$ sind Elternelemente von X_i, entschieden wird. Wenn $\mathbf{Pa}(X_i)$ nur ein Element enthält, ist nichts weiter zu beachten. Andererseits muß klar sein, dass wenn die Mutual Information einer Kombination von mehreren Elementen signifikant größer 0 ist, der Einfluss eigentlich nur von wenigen Elementen ausgehen kann. Dieser Sachverhalt macht einen weiteren Test notwendig.
Für jede Teilmenge $\mathbf{Pa}_T(X_i) \subset \mathbf{Pa}(X_i)$, die ein Element weniger als $\mathbf{Pa}(X_i)$ besitzt, wird getestet, ob die Abhängigkeit zwischen $\mathbf{Pa}(X_i)$ und X_i signifikant größer ist, als die Abhängigkeit zwischen $\mathbf{Pa}_T(X_i)$ und X_i [6]. Ist

$$(MI(X_i, \boldsymbol{Pa}(X_i)) - MI(X_i, \boldsymbol{Pa}_T(X_i))) * M * \ln 4 \tag{4.11}$$

signifikant größer 0, bedeutet dies, dass sich $MI(X_i, \mathbf{Pa}(X_i))$ und $MI(X_i, \mathbf{Pa}_T(X_i))$ signifikant unterscheiden und die Elemente in $\mathbf{Pa}_T(X_i)$ nicht allein den Zustand von X_i beeinflussen. Es gilt [6]:

$$(MI(X_i, \boldsymbol{Pa}(X_i)) - MI(X_i, \boldsymbol{Pa}_T(X_i))) * M * \ln 4 \sim \chi^2_{df;1-\alpha_T}, \tag{4.12}$$

[9]Für einen Stichprobenumfang kleiner als 40 ist der exakte Fisher-Test anzuwenden. Für die durchgeführten Simulationen in dieser Arbeit ist der notwendige Stichprobenumfang größer, so dass ein χ^2-Unabhängigkeitstest verwendet wird.

$$\{X_1[t], X_2[t]\} \quad \{X_1[t], X_3[t]\} \quad \{X_2[t], X_3[t]\}$$

$$\{X_1[t]\} \qquad \{X_2[t]\} \qquad \{X_3[t]\}$$

$$\{\emptyset\}$$

Abbildung 4.8: Bsp. für einen Suchraum, wenn Struktur partiell gelernt wird. Zur Zeit t können maximal die Zustände von zwei Elementen bekannt sein. Es fehlt die Menge $\{X_1, X_2, X_3\}$, da sie nicht messbar ist.

mit den Freiheitsgraden $df = df(X_i, \mathbf{Pa}(X_i)) - df(X_i, \mathbf{Pa}_T(X_i))$. Eine Ablehnung der Nullhypothese

$$H_0: \quad MI(X_i, \boldsymbol{Pa}(X_i)) - MI(X_i, \boldsymbol{Pa}_T(X_i)) = 0 \tag{4.13}$$

rechtfertigt die Erweiterung von $\mathbf{Pa}_T(X_i)$ auf $\mathbf{Pa}(X_i)$. Kann die Nullhypothese nicht abgelehnt werden, darf keine Kante von dem zusätzlichen Element in $\mathbf{Pa}(X_i)$ nach X_i führen. Es ist notwendig diese Analyse für jede Teilmenge durchzuführen.
Durch die Verwendung von statistischen Tests zur Erkennung von signifikanten Abhängigkeiten können falsche Entscheidungen getroffen werden. Für einen kleinen Stichprobenumfang M werden viele Belegungen der zwei Zufallsvariablen nicht beobachtet und somit ihre geschätzten Wahrscheinlichkeiten $\hat{p}_{ij}$ auf 0 gesetzt. Die Prüfgröße $\hat{MI}(X_i, \mathbf{Pa}(X_i)) * M * \ln 4$ ist in solchen Fällen sehr klein und die Nullhypothese wird auch dann nicht abgelehnt, wenn in Wahrheit ein Zusammenhang zwischen den Zufallsvariablen existiert (Fehler 2. Art). Um dies zu verhindern, gilt es, die Trainingsmenge zu vergrößern, was wiederum dazu führen kann, dass die Nullhypothese abgelehnt wird, obwohl sie richtig war (Fehler 1. Art, Irrtumswahrscheinlichkeit). Der Fehler 1. Art wird über das Signifikanzniveau α gesteuert. Ein großes α bedeutet eine große Irrtumswahrscheinlichkeit und das bevorzugt die Identifizierung von falsch positiven Elternelementen. Ein kleines α dagegen verringert die Irrtumswahrscheinlichkeit und führt zu einer vergrößerten Anzahl von falsch negativen Elternelementen. Dementsprechend ist es wichtig, einen Mittelweg zwischen dem Stichprobenumfang und α zu finden.

4.4.2 Suchraum

Die Suche nach einem guten Schätzer für die Übergangsstruktur eines DBN kann erfolgen, indem mit einer leeren Struktur begonnen wird und nacheinander Kanten hinzugefügt werden. Eine Kante wird nur dann eingeführt, falls dies aufgrund der

Bewertungsfunktion gerechtfertigt ist. Kann $\mathbf{Pa}(X_i)$ nur maximal l Variablen enthalten, z.B. wenn Zustandsübergangsdaten nach Def. 1.4 gegeben sind, besteht der Suchraum nur noch aus

$$\sum_{i=1}^{l} \binom{N}{i}$$

Teilstrukturen für jedes Outputelement. Aktuelle Kandidaten können nur Inputelemente mit bekanntem Expressionszustand sein. Diese Herangehensweise ist sinnvoll, wenn die Elemente mit bekannten Anfangszuständen variieren. Angenommen, es können von zwei Elementen die Anfangszustände gleichzeitig gemessen werden, ergibt sich ein Suchraum wie in Abb. 4.8. Die Suche beginnt, analog zu REVEAL, am Boden des Gitters mit einer Struktur ohne Kanten. Daraufhin wird über Teilstrukturen mit einem Kandidaten bis zu Teilstrukturen mit zwei Kandidaten nach der optimalen Elternmenge gesucht. Jedoch ist es unmöglich, die Anfangszustände von drei Inputelementen gleichzeitig zu messen und somit auch die Evaluierung von Teilstrukturen mit drei Kandidaten durchzuführen.
Es wurden zwei Kriterien eingeführt, die den Algorithmus abbrechen:

Kriterium 1: *Teste für jede Menge* $\mathbf{Pa}(X_i)$ *die Nullhypothese (4.10) mit* $\alpha = 0.001$. *Enthält die Menge* $\mathbf{Pa}(X_i)$ *mehr als ein Element, ist zusätzlich die Prüfung der Nullhypothese (4.13) mit* $\alpha_T = 0.0001$ *für jede Teilmenge der Kandidaten notwendig. Können (4.10) und (4.13) abgelehnt werden, sind die Kanten von* $\mathbf{Pa}(X_i)$ *nach* X_i *gerechtfertigt.*

Kriterium 2: *Beende Suche nach Elternelement für angemessenes* ϵ, *falls*

$$|\hat{MI}(\mathbf{Pa}(X_i), X_i) - \hat{H}(X_i)| < \epsilon.$$

Sonst teste die Nullhypothesen (4.10) und (4.13).

Die Signifikanzniveaus wurden sehr niedrig gewählt, um die Irrtumswahrscheinlichkeit sehr klein zu halten. Die in das Modell aufgenommenen Eltern sollen mit einer hohen Wahrscheinlichkeit einen realen Einfluß auf das Outputelement haben. Abbildung 4.15 zeigt eine Beispielsimulation, bei der ein Signifikanzniveau von $\alpha = 0.01$ für das erste Kriterium gewählt wurde. Im Vergleich zur Abbildung 4.12 haben sich die Wahrscheinlichkeiten, die Eltern korrekt zu identifizieren, wesentlich verschlechtert. Denn die Irrtumswahrscheinlichkeit hat sich erhöht und es werden oft falsch positive Eltern identifiziert (z.B. $P_{positive} \approx 85\%$ für $k_{bio} = 1$ und $P_{positive} \approx 91\%$ für $k_{bio} = 2$). Für die Prüfung der Nullhypothese (4.13) wurde eine noch geringere Irrtumswahrscheinlichkeit gewählt, da ein Inputelement in mehreren Mengen $\mathbf{Pa}(X_i)$

auftritt und dadurch mehrmals auf Unabhängigkeit mit dem selben Outputelement getestet wird. Dadurch wirkt die Wahrscheinlichkeit einen Fehler 1. Art zu begehen multiplikativ. Ein sehr kleines α_T ist notwendig.
Es wird für jede Zufallsvariable $X_i[t+1]$ eine leere Elternmenge angenommen und Zufallsvariablen, die einen signifikanten Einfluss aufzeigen, als Eltern von $X_i[t+1]$ identifiziert. Fällt die Wahl auf das zweite Kriterium, wird auf Basis des statistischen Tests entschieden, falls die Kandidaten $\mathbf{Pa}(X_i[t+1])$ das Outputelement X_i nicht oder nur unvollständig erklären. Ansonsten bilden sie die vollständige Elternmenge und die Suche wird abgebrochen[10]. Das zweite Kriterium ist für Elemente sinnvoll, die maximal so viele Eltern haben, wie Elemente mit bekannten Anfangszuständen existieren. Ihre Eltern werden aufgrund dieser Abbruchbedingung mit einer sehr hohen Wahrscheinlichkeit vollständig und korrekt, bei einer genügend großen Anzahl von Trainingsvektoren, identifiziert. Dieser Schluss lässt sich aus den Simulationen zu REVEAL in [28] und Kapitel 4.2 ziehen. Würde dagegen nur aufgrund der statistischen Tests entschieden, besteht die Gefahr, Testfehlern zu unterliegen.

4.4.3 Implementierung

Bei in-silico Experimenten werden Zellen mit einem Genregulationsnetzwerk von $N = 20$ Genen anhand eines zufällig konstruierten Booleschen Netzes modelliert. Die Anzahl der Elternelemente ist wieder zu gleichen Teilen im Netz verteilt, um Aussagen über die Identifizierbarkeit der Eltern abhängig von ihrer Anzahl k zu erhalten. Die maximal mögliche Zahl von Inputelementen für eine Boolesche Funktion ist $K = 4$. Die Zellpopulation für ein Experiment besteht ausschließlich aus Zellen mit unterschiedlichen Startzuständen und ihre Größe ist mit M festgelegt. Die Gene mit bekannten Anfangszuständen dürfen sich zwischen den ausgewählten Zellen nicht unterscheiden. Daraus ergeben sich Zustandsübergangstabellen, wie die Tabellen 4.3, wenn bei verschiedenen Experimenten verschiedene Anfangszustände bekannt sind. Auf Basis dieser Zustandsübergangstabellen sollen die strukturellen Beziehungen zwischen den regulierenden und den regulierten Elementen rekonstruiert werden. Für jede aktuelle Kombination von bekannten Anfangszuständen wird die Informationsübertragung auf die Folgezustände charakterisiert. Besitzt ein Inputelement eine signifikante Informationsübertragung auf ein Outputelement, wird es als sein Inputelement identifiziert. Die Inputelemente aus jeder einzelnen Zustandsübergangstabelle werden zusammengefaßt und bilden schließlich die Elternmenge für das entsprechende Outputelement.
Der Algorithmus in Pseudocode lautet:

FOR $b := 1$ *TO* $b = 300$

[10]Diese Abbruchbedingung ist die gleiche wie sie REVEAL benutzt.

$G_1[t]$	$G_1[t+1]$	$G_2[t+1]$	...	$G_N[t+1]$
x	x	x	...	x
⋮	⋮	⋮	⋮	⋮
x	x	x	...	x

⋮

$G_N[t]$	$G_1[t+1]$	$G_2[t+1]$	...	$G_N[t+1]$
x	x	x	...	x
⋮	⋮	⋮	⋮	⋮
x	x	x	...	x

Tabelle 4.3: Aufbau der Zustandsübergangstabellen, wenn $l = 1$. x kennzeichnet einen gemessenen Zustand. Jede Kombination von bekannten Anfangszuständen tritt genau M-mal auf. $G_i[t]$ ist eine Zufallsvariable, die den Expressionszustand von Gen i zur Zeit t definiert.

Create Boolean Network b_{net} with N nodes randomly
FOR $j := 1$ TO $j = l$
FOR EACH combination $\mathbf{Pa}_j$ of j known input elements
Create M cells with b_{net}
Assign unique initial state to each cell
FOR EACH cell i
FOR EACH $node_j$
IF state of $node_j$ in b_{net_i} is known THEN transition_table[t] ← state of $node_j$
Update cell once
FOR EACH $node_j$
transition_table[t+1] ← state of $node_j$
D:=transition_table
G ← LEARN_STRUCTURE_PARTLY(D,$\mathbf{Pa}_j$)
Evaluate G
Calculate $P_{Input}(k)$

PROCEDURE LEARN_STRUCTURE_PARTLY:

FOR EACH output element o
Choose between

Criterion 1:

IF $\hat{MI}(\mathbf{Pa}_j, o) * M * \ln 4 > \chi^2_{(q_i-1)*(r_i-1);1-\alpha}$ *AND no. elements in* $\mathbf{Pa}_j = 1$

parents[o] $\leftarrow \mathbf{Pa}_j$

IF $\hat{MI}(\mathbf{Pa}_j, o) * M * \ln 4 > \chi^2_{(q_i-1)*(r_i-1);1-\alpha}$ *AND no. elements in* $\mathbf{Pa}_j > 1$

FOR EACH element X in $\mathbf{Pa}_j$

$\mathbf{Pa}_{j_T} = \mathbf{Pa}_j \setminus X$

IF $(\hat{MI}(\mathbf{Pa}_j, o) - \hat{MI}(\mathbf{Pa}_{j_T}, o)) * M * \ln 4 > \chi^2_{df-df_T;1-\alpha_T}$

THEN parents[o] $\leftarrow X$

*ELSE RETURN /*Other subsets has been checked already*/*

Criterion 2:

IF $|\hat{MI}(\mathbf{Pa}_j, o) - H(o)| < \epsilon$ */*Choose adequate* ϵ **/*

parents[o] $\leftarrow \mathbf{Pa}_j$

ELSE

GO TO Criterion 1

RETURN G

4.4.4 Ergebnisse

Ein Nachteil dieser Methode ist, dass Eltern, die sich nicht in Teilmengen bis zu l Elementen mit einer signifikanten positiven Mutual Information unterteilen lassen, nicht identifizierbar sind. Im Fall $l = 1$ sind nur die Eltern erkennbar, die unabhängig von allen anderen eine Auswirkung auf das Outputelement haben. Es ist von Interesse, die Anzahl der Booleschen Regeln zu kennen, bei denen sich die Inputelemente in verschiedene *Quellen* mit höchstens l Elementen aufsplitten lassen und jede Quelle für sich einen erkennbaren Einfluss auf den Output hat. Nur diese Boolesche Funktionen können durch die Analyse von Teilstrukturen mit maximal l Inputelementen vollständig erkannt werden. Die Inputelemente für alle anderen Booleschen Regeln sind mit der hier beschriebenen Methode nicht identifizierbar.
In [42] wurde untersucht, wieviel Information ein Inputelement effektiv auf den Output einer Booleschen Regel überträgt. Die Übertragung von Information kann auch in Verbindung mit weiteren Inputelementen geschehen. Das Resultat sind die k_{eff} Regeln gewesen. Die Beschränkung der Booleschen Regeln auf die k_{eff}-Regeln ist sinnvoll, denn Inputelemente die keine Information zum Output übertragen, sind nicht notwendig, um die Regel eindeutig zu beschreiben. Da solche Inputelemente keine Beziehung zu dem Outputelement haben, kann zudem auch keine Relation anhand der Trainingsdaten erkannt werden. Interessant ist die Frage nach dem Anteil

k	Regeln	k_{eff}	k_1	$k_{1,2}$	k_{can}
1	4	2	2 (100%)	2 (100%)	2
2	16	10	8 (80%)	10 (100%)	8
3	256	218	136 (62%)	216 (99%)	16
4	65536	64594	36864 (57%)	63424 (98%)	32

Tabelle 4.4: Analyse der k_{eff}-Regeln. Es wurde untersucht wieviel Regeln es, abhängig von k gibt, bei denen sich der Input in Gruppen mit höchstens 1 und 2 Elementen mit signifikanten Einfluss aufteilen läßt. Außerdem wurde die Anzahl der Regeln bestimmt, deren Inputelemente alle *canalyzing* sind. Spalte $\mathbf{k_{eff}}$ enthält die Anzahl der Regeln mit effektivem Input für k Inputelemente/Eltern. Spalte $\mathbf{k_1}$ enthält die Anzahl der k_{eff}-Regeln bei denen jedes Inputelement eine separate Auswirkung auf den Output hat (vgl. Def. 4.5). Spalte $\mathbf{k_{1,2}}$ enthält die Anzahl der k_{eff}-Regeln bei denen jedes Inputelement entweder eine separate Auswirkung oder in Verbindung mit einem weiteren Inputelement auf den Output hat (vgl. Def. 4.6). Spalte $\mathbf{k_{can}}$ enthält die Anzahl der Regeln bei denen jedes Inputelement ein *canalyzing* Element ist.

von Booleschen Funktionen, deren Input sich aus unabhängigen Informationsquellen mit einem Element (k_1) und maximal mit zwei Elementen ($k_{1,2}$) zusammensetzt.

Definition 4.5 (k_1-Regeln) *Boolesche Funktionen, bei denen jedes Inputelement unabhängig von den anderen Inputelementen Information auf den Output überträgt, sind k_1-Regel. Ein Inputelement $X_j[t]$ überträgt separat Information auf den Output $X_i[t+1]$ einer Booleschen Funktion, falls $MI(X_j[t], X_i[t+1]) > 0$.*

Definition 4.6 ($k_{1,2}$-Regeln) *Boolesche Funktionen, bei denen jedes Inputelement unabhängig von den anderen oder in Verbindung mit einem zweiten Inputelement Information auf den Output überträgt, sind $k_{1,2}$-Regel. Ein Inputelement $X_{j_1}[t]$ überträgt nur gemeinsam mit einem zweiten Inputelement $X_{j_2}[t]$ Information auf den Output $X_i[t+1]$ einer Booleschen Funktion, falls*

$$MI(X_{j_1}[t], X_i[t+1]) = 0 \text{ und}$$
$$MI(X_{j_1}[t], X_{j_2}[t]; X_i[t+1]) > 0 \text{ und}$$
$$|MI(X_{j_1}[t], X_{j_2}[t]; X_i[t+1]) - MI(X_{j_2}[t], X_i[t+1])| > 0.$$

In Tabelle 4.4, sind die Ergebnisse der numerischen Untersuchungen zu sehen.
Der Anteil der Regeln mit ausschließlich effektiven Inputelementen wurde analog zu [42] bestimmt. Jedes Inputelement einer Booleschen Regel wurde getrennt danach getestet, ob es eine Auswirkung auf den Output hat. Ein Inputelement $X_j[t]$ ($j =$

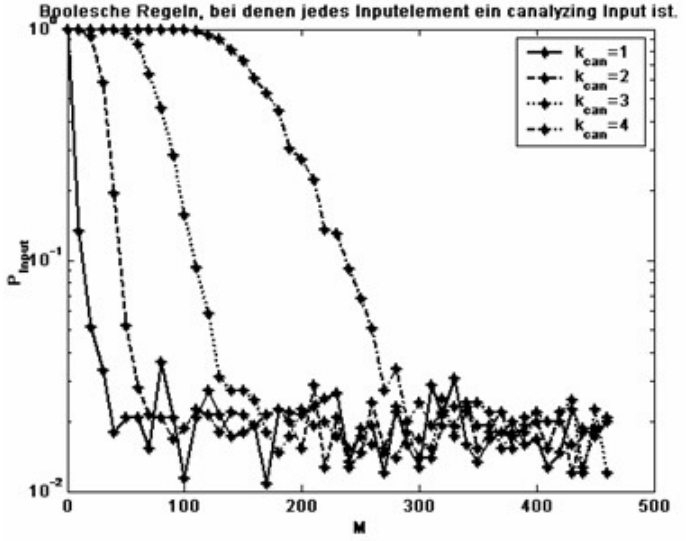


Abbildung 4.9: Reverse Engineering mit Booleschen Regeln bei denen jedes Inputelement *canalyzing* ist. Als Abbruchbedingung wurde das Kriterium 1 gewählt ($N = 20$, $K = 4$, $l = 1$). Aufgrund von Fehlern des χ^2-Unabhängigkeitstests konvergieren die Kurven nicht gegen Null. Der Konvergenzwert ergibt sich daraus, dass ab einem bestimmten M immer alle Inputelemente korrekt gefunden wurden ($P_{identified}(k)$ von 100% für alle k) und falsch positive Eltern auftreten können ($P_{positive}(k)$ ist ca. 98% für alle k).

$1, \ldots, k$) ist nicht überflüssig, wenn es Information mit mindestens einem anderen Inputelement oder dem Output $X_i[t+1]$ teilt, also:

$$MI(\{X_1[t], \ldots, X_k[t]\} \setminus X_j[t], X_i[t+1]; X_j[t]) > 0. \tag{4.14}$$

Wie gut die Struktur anhand der hier vorgestellten Reverse Engineering Strategie gelernt werden kann, erschließt sich durch den Anteil von Inputelementen einer Booleschen Funktion, die unabhängige Informationsquellen sind (vgl. Tab. (4.4)). Folglich lassen sich nur 62% der $k_{eff} = 3$-Regeln in drei einelementige Quellen mit signifikanter Informationsübertragung zerlegen. Daraus folgt, dass nur in 62% der Fälle die Eltern für $k_{eff} = 3$-Regeln identifizierbar sind. Das gilt, wenn nur von einem Element der Anfangszustand bekannt ist. Eine Erhöhung der bekannten Zustände auf 2 bewirkt, dass die Eltern nun schon in 99% der Fälle korrekt gelernt werden. Anhand von Simulationsstudien kann man abschätzen, wieviel Zustandsübergänge benötigt werden. Dafür ist es notwendig in-silico Experimente durchzuführen für

jede bekannte Anfangskombination, bestehend aus l Genen. Eine Wiederholung dieser Experimente für verschiedene Stichprobenumfänge M macht es möglich, den Reverse Engineering Algorithmus zu evaluieren.

Es ist zu erwarten, dass Inputelemente mit hoher Informationsübertragung schon bei kleinem Stichprobenumfang korrekt erkannt werden können. Ein Inputelement determiniert den Output vollständig, falls beide Zustände des Inputelementes den Zustand des Outputelementes vollständig festlegen. Solche Inputelemente treten jedoch nur in $k_{eff} = 1$-Regeln auf. Andererseits übertragen auch die anderen Inputelemente Information auf den Output und ein Inputelement legt den Zustand des Outputs nicht mehr lückenlos fest. Demzufolge können bei Regeln mit $k_{eff} > 1$ nicht beide Zustände eines Inputelementes den Output garantieren. Es ist aber möglich, dass ein Inputelement mit einem Zustand einen Zustand des Zielelementes garantiert. Ein *Canalyzing Input*[11] hat genau diese Eigenschaft. In diesen Fällen ist die Informationsübertragung vermutlich groß genug, um schon bei kleinem Stichprobenumfang dieses Inputelement korrekt zu identifizieren.
Die Computersimulationen dieser Arbeit, die Boolesche Funktionen mit ausschließlich *canalyzing* Inputelementen untersuchen, liefern den Verlauf der Wahrscheinlichkeit P_{Input} (Abb. 4.9). Aufgrund der Irrtumswahrscheinlichkeit des Signifikanztestes können nicht alle Elternmengen korrekt gelernt werden, denn es wird immer einige Mengen geben, die falsch positive Eltern enthalten ($P_{positive}(k)$ ist ca. 98% für alle k bei einer $P_{identified}(k)$ von 100% für alle k). Im allgemeinen ist es aber kein Problem Eltern, die *canalyzing* sind, bei genügend großer Stichprobengröße, korrekt zu lernen. Wie verhält sich jedoch der Reverse Engineering Algorithmus, falls die Abhängigkeit zwischen Input- und Outputelement nicht so einfach zu identifizieren ist?
Die meisten Booleschen Funktionen besitzen Inputelemente, die keinen Zustand des Outputs alleine garantieren. Solche Inputelemente sind in der Lage, den Zustand des Outputs in Verbindung mit weiteren Inputelementen vollständig festzulegen. Trotzdem können sie separat für sich eine kleine Informationsquelle bilden, die wenig Information auf das Outputelement weitergibt. Die Abhängigkeit in solchen Fällen zwischen einem einzelnen Inputelement und dem Output ist meistens gering und daher wird die Mutual Information nur wenig größer 0 sein. Die Nullhypothese (4.10) wird erst bei viel größerem Stichprobenumfang abgelehnt, denn nur so können kleine Unterschiede anhand von statistischen Tests erkannt werden. Wieviel Zustandsübergangspaare notwendig sind, ist in Abbildung 4.10 zu sehen. Während bei Regeln mit *canalyzing* Inputelementen ca. 300 Übergänge benötigt wurden, sind ca. 2500 notwendig, um eine Eltermenge mit 4 Elementen zu identifizieren. Dieser sprunghafte Anstieg für $k_1 = 4$-Regeln ist nicht bei $k_1 = 3$-Regeln zu beobachten. Die korrekte

[11] Siehe Definition 3.4.

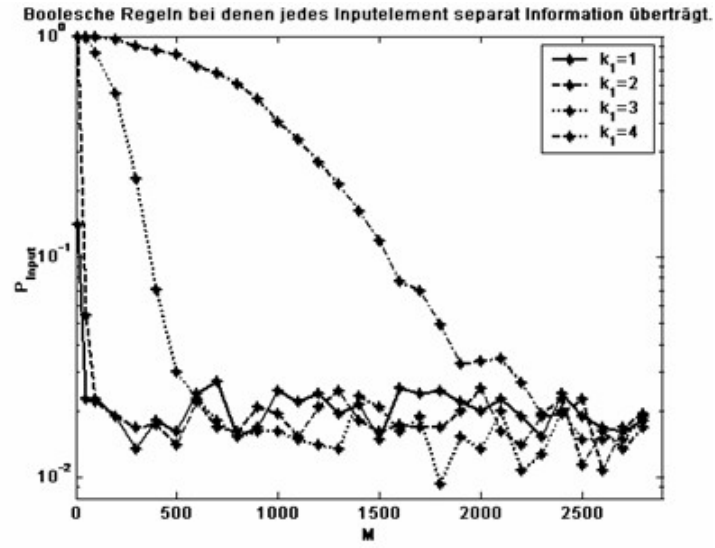


Abbildung 4.10: Reverse Engineering mit Booleschen Regeln bei denen jedes Inputelement eine unabhängige Quelle für die Informationsübertragung ist. Als Abbruchbedingung wurde das Kriterium 1 gewählt ($N = 20$, $K = 4$, $l = 1$). Aufgrund von Fehlern des χ^2-Unabhängigkeitstests konvergieren die Kurven nicht gegen Null.

Identifizierung ihrer Inputelemente erfolgt schon bei ca. 700 Übergängen.

Diese beiden Simulationsstudien sind die Basis für Aussagen über k_{eff}-Boolesche Regeln. Ihr Resultat ist, dass ausschließlich Eltern, die eine einzelne Informationsquelle bilden, unabhängig von allen anderen gelernt werden können. Nur bei einer Teilmenge von Booleschen Regeln ist jedoch jedes Inputelement eine separate Quelle (Tabelle 4.4). Ihr Anteil bei den Booleschen Regeln mit effektivem Input beträgt 100% für $k_{eff} = 1$, 80% für $k_{eff} = 2$, 63% für $k_{eff} = 3$ und 57% für $k_{eff} = 4$. Bei allen anderen Regeln kann der Reverse Engineering Algorithmus die Inputelemente nicht identifizieren. Die Computersimulationen verdeutlichen diese Aussagen. Abbildung 4.11 zeigt die Wahrscheinlichkeiten, die Eltern falsch zu bestimmen, abhängig von k_{eff}. Sie pendeln sich jeweils auf feste Werte ein, welche genau den theoretisch hergeleiteten Wahrscheinlichkeiten entsprechen. Diese Kurven zeigen, dass die Unsicherheit bei den Ergebnissen des Algorithmus, außer bei $k_{eff} = 1$, zu groß ist, um in realen Anwendungen bestehen zu können. Wäre es jedoch möglich, gleichzeitig die Anfangszustände von zwei Inputelementen zu charakterisieren ($l = 2$), dann fände der Reverse Engineering Algorithmus mit hoher Wahrscheinlichkeit die optimale Konstellation aller Eltern. Denn bei den meisten Booleschen Funktionen ist der Input in unabhängige Informationsquellen mit einem oder zwei Eltern unterteilt (Tab. 4.4).

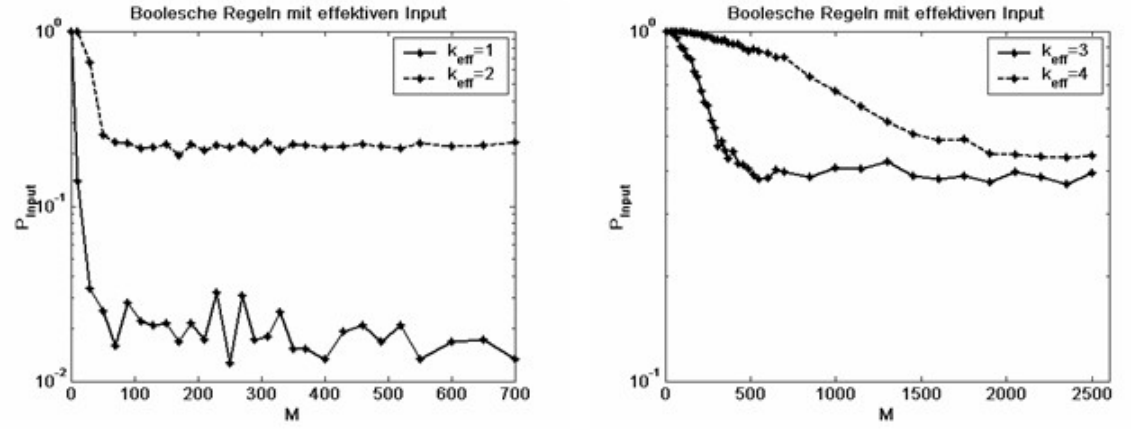


Abbildung 4.11: Reverse Engineering mit Mutual Information und Booleschen Regeln mit effektiven Inputelementen. Als Abbruchbedingung wurde das Kriterium 1 gewählt ($N = 20$, $K = 4$, $l = 1$). Die Konvergenz für $k_{eff} = 1$-Regeln ergibt sich aus Fehlern des χ^2-Unabhägigkeitstests. Die Konvergenz der anderen Kurven hängt dagegen von dem Anteil an k_1-Regeln ab. Sie konvergieren jeweils bei ca. $1 - \frac{\#k_1}{\#k_{eff}}$ (vgl. Tab. 4.4).

Für biologische Systeme können die zu betrachteten Booleschen Regeln auf die *Canalyzing Funktionen* beschränkt werden. Im folgenden bezeichnet k_{bio} die Anzahl der Inputelemente einer *Canalyzing Funktion*. Bei ihnen ist der Informationsfluss vom Input zum Output ebenfalls in mehrere Quellen aufgeteilt. Deshalb soll auch für *Canalyzing Funktionen* analysiert werden, welcher Anteil von regulierenden Einflüssen mit dem Abhängigkeitsmaß Mutual Information korrekt erkannt werden kann, falls die Anfangszustände unvollständig gegeben sind (s. Tabelle 4.5). Als erstes wurde der Anteil der *Canalyzing Funktionen* bei den k_{eff}-Booleschen Regeln bestimmt. So gibt es für Regeln mit 3 Inputelementen 88 Boolesche Regeln, die nach Stuart Kauffman biologisch relevant sind. Eine obere Grenze wurde in [24] angegeben. Sie beträgt

$$4K * 2^{(2^{K-1})}. \tag{4.15}$$

Für die Analyse des hier vorgestellten Reverse Engineering Algorithmus sind die Schnittmengen mit den Booleschen Regeln interessant, bei denen jedes Inputelement für sich eine Quelle zur Informationsübertragung ist und mit denen, bei denen die Inputelemente maximal zu zweit eine Quelle bilden. Kann im Anfangszustand eines Zustandsübergangspaares nur der Wert eines Elementes gemessen werden, sind die Inputelemente von Regeln mit $k_{bio} = 1$ und $k_{bio} = 2$ lernbar (vgl. Tab. 4.5). Für die anderen k_{bio}-Regeln sind die Anteile der identifizierbaren Elternmengen größer,

k	k_{bio} (Canalyzing Funktionen)	k_1	$k_{1,2}$
1	2	2 (100%)	2 (100%)
2	8	8 (100%)	8 (100%)
3	88	64 (73%)	88 (100%)
4	3104	1888 (61%)	3072 (99%)

Tabelle 4.5: Analyse der *Canalyzing Funktionen*. Es wurde untersucht wieviel Regeln es in Abhängigkeit von k gibt, die für biologische Systeme relevant sind. Außerdem wurden ihre Schnittmengen mit den k_1 und $k_{1,2}$ Regeln bestimmt. Spalte $\mathbf{k_{bio}}$ enthält die Anzahl der Regeln, die nach Kauffman ([24]) biologisch relevant sind. Spalte $\mathbf{k_1}$ enthält die Anzahl der k_{bio}-Regeln bei denen jedes Inputelement eine separate Auswirkung auf den Output hat (vgl. Def. 4.5). Spalte $\mathbf{k_{1,2}}$ enthält die Anzahl der k_{bio}-Regeln bei denen jedes Inputelement entweder eine separate Auswirkung hat oder eine Auswirkung in Verbindung mit einem weiteren Inputelement auf den Output hat (vgl. Def. 4.6).

als bei den nicht biologisch relevanten Funktionen. Sie betragen 72%, wenn das Outputelement 3 Eltern hat und ca. 61%, für 4 Eltern (vgl. Tab. 4.5). Der Verlauf der Wahrscheinlichkeit, die Eltern falsch gelernt zu haben, hängt von der Zahl der zur Verfügung stehenden Stichproben ab, siehe Abbildung 4.12.
Da *Canalyzing Funktionen* mindestens ein Inputelement haben, das *canalyzing* ist, wird dieses immer gefunden. Für Regeln mit nur einem weiteren Inputelement ($k_{bio} = 2$) ist das zweite Element, unabhängig von dem *canalyzing* Inputelement, eine unabhängige Informationsquelle. Deshalb ist es möglich, die Struktur von *Canalyzing Funktionen*, die 2 Eltern haben, mit einer hohen Wahrscheinlichkeit zu identifizieren. Hängt das Outputelement jedoch von drei oder vier Inputelementen ab, werden diese nicht immer korrekt erkannt. Die Wahrscheinlichkeit, ihre Eltern korrekt zu lernen, überschreitet 0.27 bzw. 0.39 nicht.
Bessere Ergebnisse sind möglich, wenn zwei Anfangszustände gleichzeitig bekannt sind (Abb. 4.13). Obwohl $k_{bio} = 3$-Regeln sich immer in Quellen der Größe eins oder zwei aufteilen lassen, erreicht P_{Input} zwar einen niedrigen, aber dafür dass alle Eltern gefunden werden müssten, einen relativ hohen Wert. Dieser Umstand hängt mit möglichen Testfehlern bei (4.13) zusammen. Das Signifikanzniveau $\alpha_T = 0.0001$ ist sehr niedrig gewählt, trotzdem treten noch falsch positive Eltern auf (vgl. Tabelle 4.6). Der Grund hierfür ergibt sich aus der Notwendigkeit, jede Kombination von $l = 1$ bis $l = 2$ bekannten regulierenden Elementen mit dem Outputelement analysieren zu müssen. Jedes Inputelement wird daher mehrfach auf signifikanten Einfluss getestet und die Wahrscheinlichkeit, dass es irrtümlich identifiziert wird, ist relativ hoch.
In der Abbildung 4.14 ist für $k_{bio} = 4$ aufgeschlüsselt, wieviel Zustandsübergänge notwendig sind, um eine Teilmenge der Eltern korrekt zu finden, ohne falsch positive

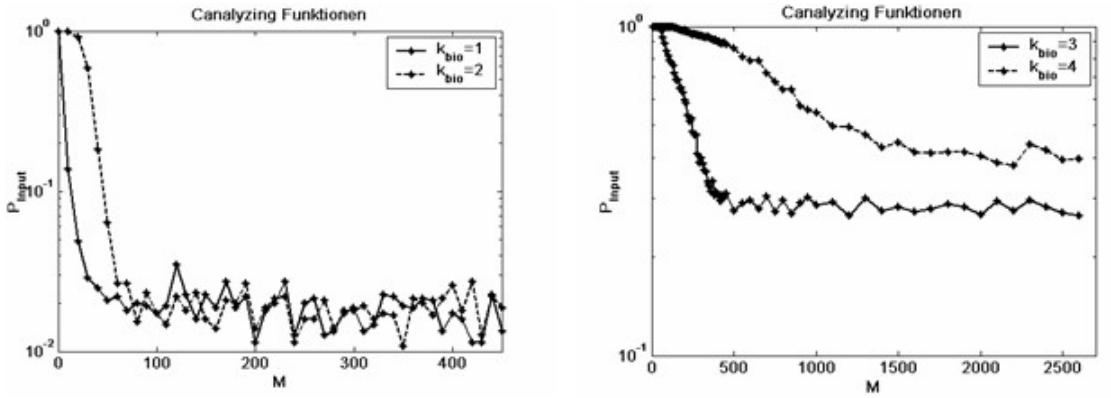


Abbildung 4.12: Reverse Engineering mit *Canalyzing Funktionen*. Als Abbruchbedingung wurde das Kriterium 1 gewählt ($N = 20$, $K = 4$, $l = 1$). Die Konvergenz für $k_{bio} = 1$-Regeln und $k_{bio} = 2$-Regeln ergibt sich aus Fehlern des χ^2-Unabhägigkeitstests. Die Konvergenz für $k_{bio} = 3$-Regeln und $k_{bio} = 4$-Regeln hängt dagegen von dem Anteil an k_1-Regeln ab. Sie konvergieren jeweils bei ca. $1 - \frac{\#k_1}{\#k_{bio}}$ (vgl. Tab. 4.5 und 4.6).

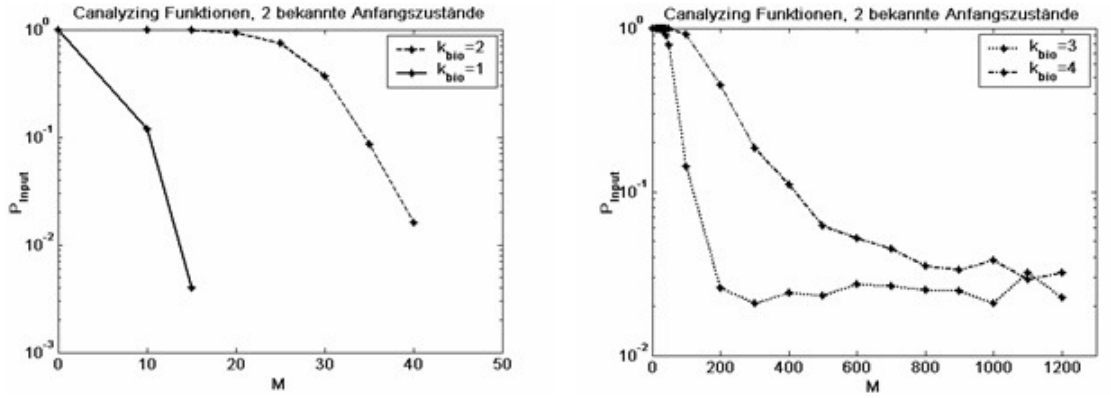


Abbildung 4.13: Reverse Engineering mit *Canalyzing Funktionen*. Als Abbruchbedingung wurde das Kriterium 2 gewählt ($N = 20$, $K = 4$, $l = 2$). Die Kurven für $k_{bio} = 1$ und $k_{bio} = 2$ konvergieren gegen Null, da für sie das Kriterium 2 verwendet wurde (vgl. Abb. 4.1). Die Konvergenz für $k_{bio} = 3$-Regeln ergibt sich aus Fehlern des χ^2-Unabhägigkeitstests. Die Konvergenz für $k_{bio} = 4$-Regeln hängt dagegen von dem Anteil an $k_{1,2}$-Regeln (vgl. Tab. 4.5 und 4.6) und von Fehlern des statistischen Tests ab. Da ein Inputelement in mehreren Kombinationen mit zwei Inputelementen auftritt, wird es mehrfach auf signifikanter Informationsübertragung getestet und die Wahrscheinlichkeit, dass es irrtümlicherweise identifiziert wird, ist daher relativ hoch.

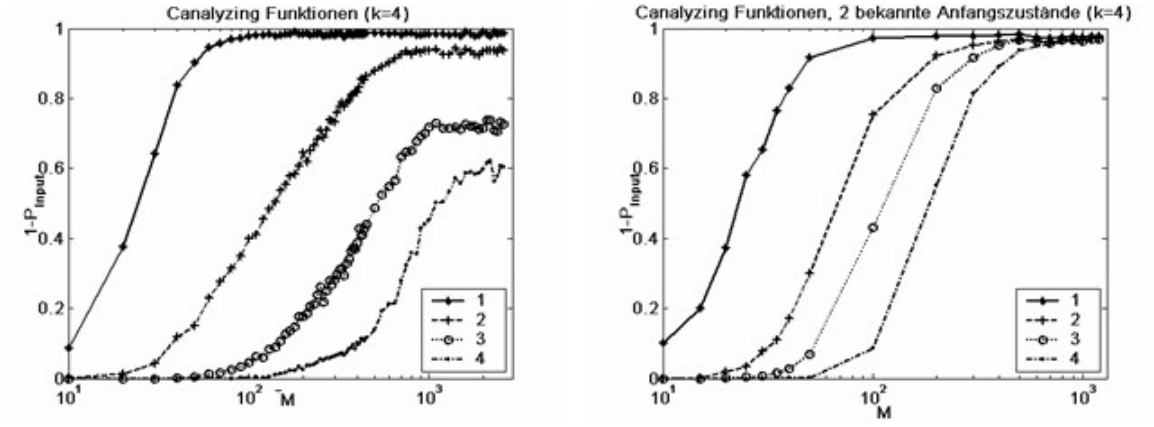


Abbildung 4.14: Reverse Engineering mit *Canalyzing Funktionen*. Es sind die Wahrscheinlichkeiten, unterschiedlich große Teilmengen von Elternmengen korrekt zu identifizieren, für $k = 4$ zu sehen. Die linke Abbildung ist für $l = 1$ und die rechte für $l = 2$. Die Kurven mit **1** gekennzeichnet, bezeichnen die Wahrscheinlichkeit, dass ein Elternelement von insgesamt vier Elternelementen korrekt gefunden wurde und keine falsch positiven Elemente enthalten waren. Die Kurven mit **2** gekennzeichnet, bezeichnen die Wahrscheinlichkeit, dass zwei Elternelemente von insgesamt vier Elternelementen korrekt gefunden wurden und keine falsch positiven Elemente enthalten waren, u.s.w..

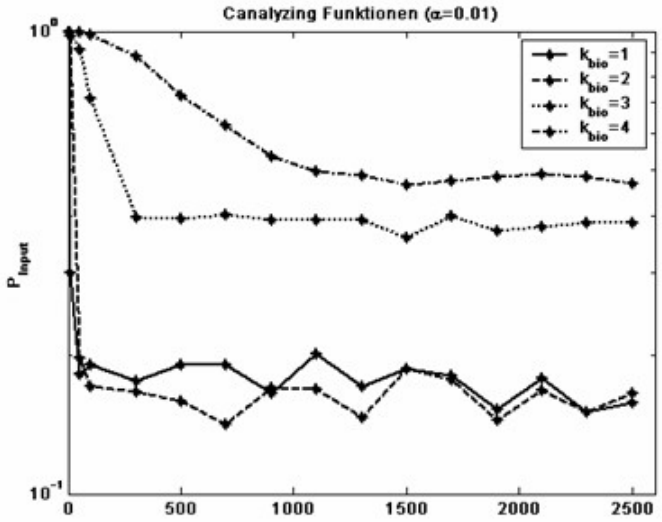


Abbildung 4.15: Reverse Engineering mit *Canalyzing Funtionen*. Es wurde ein höheres Signifikanzniveau ($\alpha = 0.01$) gewählt und die Abbruchbedingung war das Kriterium 1 ($l = 1$). Die Konvergenzwerte der Kurven ergeben sich aus dem Anteil an k_1-Regeln und aus Fehlern des χ^2-Unabhängigkeitstests.

	l=1		l=2	
k_{bio}	$P_{identified}(k_{bio})$	$P_{positive}(k_{bio})$	$P_{identified}(k_{bio})$	$P_{positive}(k_{bio})$
1	$\approx 100\%$	$\approx 98\%$	$\approx 100\%$	$\approx 100\%$
2	$\approx 100\%$	$\approx 99\%$	$\approx 100\%$	$\approx 100\%$
3	$\approx 82\%$	$\approx 99\%$	$\approx 100\%$	$\approx 99\%$
4	$\approx 82\%$	$\approx 99\%$	$\approx 99\%$	$\approx 99\%$

Tabelle 4.6: $P_{identified}(k_{bio})$ und $P_{positive}(k_{bio})$ für *Canalyzing Funktionen*. Die Werte gelten jeweils ab M, bei dem Konvergenz eintritt. Für $l = 1$ wurde das Kriterium 1 und für $l = 2$ wurde das Kriterium 2 als Abbruchbedingung gewählt.

Eltern zuzulassen. Für die korrekte Identifizierung von einem Elternelement reichen schon weniger als 100 verschiedene Zustandsübergänge für $l = 1$. Diese Elemente sind *canalyzing* Elemente, denn nur sie übertragen genügend Information. Elemente mit starkem Einfluss auf den Output werden schon bei einer kleinen Stichprobengröße mit hoher Wahrscheinlichkeit erkannt. Für die Anwendung auf Genexpressionsdaten bedeutet dieser Sachverhalt, dass, wenn auch bei kleinen Stichprobengrößen nicht alle regulierenden Gene gefunden werden, so werden doch mit einer hohen Wahrscheinlichkeit diese mit größtem Einfluss identifiziert! Stehen für ein biologisches System nur wenige Stichproben zur Verfügung, können trotzdem die „Hauptspieler“ gefunden werden, denn hohe Werte für $P_{positive}(k)$ zeigen, dass es kaum falsch positive Inputelemente gibt.

Wie die regulierenden Gene identifizierbar sind, hängt einzig davon ab, ob ihr Einfluß unabhängig von den anderen regulierenden Genen groß genug ist, um allein bzw. in Kombination mit einem zweiten Gen erkannt zu werden. Der wesentliche Nachteil ist die hohe Zahl an benötigten verschiedenen Zustandsübergangspaaren. Sie stellt eine klare Beschränkung für die Anwendung auf biologische Probleme dar, denn es scheint unwahrscheinlich, die geforderte Vielfalt von Anfangszuständen in biologischen Systemen messen zu können. Ein Minimum an sich unterscheidenden Trainingsvektoren muss aber gegeben sein, denn nur verschiedene Zustandsübergangspaare enthalten die Information, um Korrelationen zu erkennen. Deshalb liegt es nahe Methoden zu erarbeiten, die biologisches Vorwissen über das System einbeziehen. Diese können den Umfang der notwendigen Trainingsdaten verringern.

Eine Aussage darüber, ob alle Eltern gefunden wurden oder ob die identifizierte Elternmenge $\mathbf{Pa}(X_i)$ die Variable X_i nur unvollständig beschreibt, ist mit dem partiellen Lernen eingeschränkt möglich. Dafür sei angenommen, dass in $\mathbf{Pa}(X_i)$ ausschließlich korrekte Eltern vorkommen, also keine falsch positiven Eltern enthal-

ten sind. X_i wird von den Elementen in $\mathbf{Pa}(X_i)$ vollständig beschrieben, wenn

$$MI(X_i, \mathbf{Pa}(X_i)) = H(X_i)$$

gilt (vgl. Def. 2.2). Für Schätzungen muß die Beziehung

$$|\hat{MI}(X_i, \mathbf{Pa}(X_i)) - \hat{H}(X_i)| \leq \epsilon \quad (0 < \epsilon << 1) \tag{4.16}$$

herangezogen werden (vgl. Abb. 2.2). Ist sie für genügend kleines ϵ wahr, werden keine weiteren Elemente, als die in $\mathbf{Pa}(X_i)$, benötigt, um die Zustände von X_i zu erklären. Ansonsten ist eine Aussage nur dann möglich, wenn die Anzahl der Eltern in $\mathbf{Pa}(X_i)$ die Anzahl der Elemente mit bekannten Anfangszuständen l nicht überschreitet. Denn die Mutual Information aller identifizierten Eltern ist nur in diesen Fällen bestimmbar und eine Distanz größer als ϵ weist auf weitere noch unbekannte Eltern hin. In diesen Fällen sollte in das Modell eine *Hidden Variable* eingefügt werden. Je kleiner der Unterschied zwischen $\hat{MI}(X_i, \mathbf{Pa}(X_i))$ und $\hat{H}(X_i)$ ist, desto genauer beschreiben die Eltern die Zustände von X_i. In den Fällen bei denen die Anzahl der Eltern größer als l ist, ist nicht feststellbar, ob es noch weitere unbekannte Inputelemente gibt. Die Mutual Information aller Eltern kann nicht berechnet werden.

4.5 Lernen der Parameter bei bekannter Struktur

Das Aufstellen eines Modells für Genregulationsnetzwerke erfordert, neben der Suche nach einer optimalen Struktur G, das Schätzen der Modellparameter $\mathbf{\Theta}$, gegeben G. Ein geeigneter Optimierungsalgorithmus für unvollständige Trainingsdaten ist EM. Gute Schätzer für Systeme, für die kein Vorwissen vorliegt, sind MLE Schätzer. Anhand von EM soll nach $\mathbf{\Theta}^{MLE}$ gesucht werden, wenn die Modellstruktur korrekt gegeben ist.
Die Simulationsstudien werden für $N = 150$ zufällig generierte Boolesche Netze durchgeführt. Sie haben jeweils 12 Elemente. Die maximale Anzahl K von Inputelementen kann in einem Netz 4 betragen. Wieviel Inputelemente ein Outputelement zugewiesen bekommt, wird zufällig ermittelt. Die Anzahl der Inputelemente ist nicht mehr zu gleichen Teilen verteilt, denn es soll eine Analyse für das gesamte Netz und keine nach Anzahl der Eltern getrennt erfolgen. Alle Kombinationen von l bekannten Anfangszuständen werden gleichzeitig betrachtet. Deshalb ergibt sich eine Zustandsübergangstabelle, wie Tabelle 4.2. Die ausgewählten Booleschen Funktionen sind die *Canalyzing Funktionen*, die nach [24] in biologischen Systemen auftreten können. Für durchgeführte Simulationen wurde die EM Implementierung [36] verwendet.

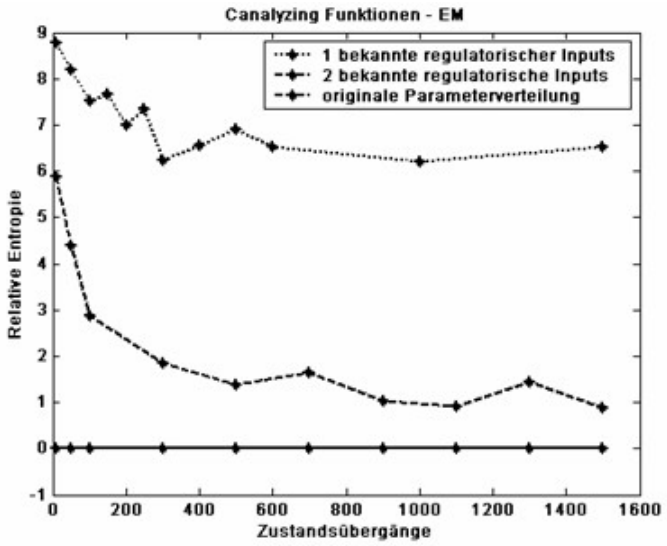


Abbildung 4.16: EM - Schätzen der Parameter bei bekannter Struktur und *Canalyzing Funktionen*. Es ist die relative Entropie abgebildet.

Anhand von EM werden optimale Schätzer für die Modellparameter gesucht. EM startet mit einem zufällig erzeugten Parameter, da a-priori Wissen über ihre Verteilung nicht zur Verfügung steht. Für die Evaluierung wird das gelernte Modell getestet, wie es Daten, die von dem originalen Modell erzeugt wurden, generieren kann. Dies geschieht anhand einer Testmenge mit 1000 Trainingsvektoren und der *relativen Entropie* (vgl. Def. 4.2.). Die Ergebnisse der Simulationen sind in Abb. 4.16 dargestellt. Sie zeigen, dass ein erfolgreiches Lernen von Parametern nur möglich ist, wenn die Expressionszustände von mindestens zwei regulierenden Genen gleichzeitig charakterisierbar sind. Ansonsten erklären die gelernten Parameter die Daten sehr schlecht. Die Begründung ist wiederum die Eigenschaft der meisten Booleschen Funktionen, mit $k \leq 4$, dass sich ihr Input in unabhängige Informationsquellen mit einem oder zwei Elementen unterteilen läßt. Insbesondere garantiert bei *Canalyzing Funktionen* mindestens ein Inputelement mit einer Zuweisung einen Zustand des Outputs. Die Wahrscheinlichkeit, dass der Output diesen Zustand bei einer bestimmten Konstellation von Zuweisungen der Inputelemente annimmt, kann daher gut geschätzt werden. Sie wird wesentlich, aber nicht vollständig, von dem *canalyzing Inputelementes* bestimmt. Zum Beispiel ist $X_1[t]$ der *Canalyzing Input* bei der Booleschen Regel in Tabelle 4.7. Der Zustand 0 von $X_2[t+1]$ wird wesentlich von $X_1[t]$ bestimmt und die Parameter $\theta^{\rightarrow}_{X_2[t+1],[011],1}$, $\theta^{\rightarrow}_{X_2[t+1],[010],1}$, $\theta^{\rightarrow}_{X_2[t+1],[001],1}$ und $\theta^{\rightarrow}_{X_2[t+1],[000],1}$ können deshalb auch nur mit dem Wissen von $X_1[t]$ gut geschätzt wer-

$X_1[t]$	$X_2[t]$	$X_3[t]$	$X_2[t+1]$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	0

Tabelle 4.7: Beispiel einer Booleschen Regel

den[12]. Sie sind ungefähr gleich der Wahrscheinlichkeit $P(X_2[t+1] = 1|X_1[t] = 0)$. Für genauere Schätzungen und für die Bestimmung der restlichen Parameter ist dagegen zusätzliche Information notwendig, die sich bei dem Großteil von *Canalyzing Funktionen* über mehrere Inputelemente verteilt. Kann der Zustand nur eines Inputelementes bei *Canalyzing Funktionen* charakterisiert werden, erklärt die gelernte Parameterverteilung nur einen geringen Teil der Parameter mit gewisser Wahrscheinlichkeit korrekt. Daraus folgt die schlechte relative Entropie für Simulationsstudien, bei denen der Zustand nur eines Inputelementes bekannt war. Können dagegen zwei Inputelemente gleichzeitig charakterisiert werden, erfolgt auch für die restlichen Parameter eine gute Schätzung.

4.6 Stammzellen befinden sich im Attraktor

Eine Diskussion über Attraktoren (vgl. Definition 3.3) im Kontext dieser Arbeit ist erforderlich, da sich das Expressionsmuster von multipotenten Blutstammzellen in einem Attraktor befindet. Denn unabhängig davon welcher Expressionszustand eine Zelle annimmt, wird eine Folge von Aktualisierungen ihres Genregulationsnetzwerkes zu einem Expressionszustand in einem Attraktor führen, wenn das Genregulationsnetzwerk nicht gestört wird [38]. Multipotente Blutstammzellen besitzen wahrscheinlich ein sich wiederholendes Genexpressionsmuster (vgl. Abb. 1.2), d.h. sie befinden sich wahrscheinlich in einem zyklischen Attraktor.
Attraktoren sind gegenüber kleineren vorübergehenden Störungen relativ stabil. Eine solche Störung kann die Veränderung eines Bits eines Attraktorzustandes auf den

[12] $\overrightarrow{\theta}_{X_i[t+1],[j_i],k_i} = P(X_i[t+1] = k_i|\mathbf{Pa}(X_i[t+1]) = j_i)$

entgegengesetzten Wert, z.B.

$$(010110) \quad \longrightarrow \quad (010111),$$

sein. Zellen, die ein stabiles Genexpressionsmuster aufweisen, verändern bei der Mehrzahl von kleineren Umweltstörungen dieses auch nicht.
Die Stabilität von Attraktoren ist gegeben, da mehrere Trajektorien in einen Attraktor laufen.

Definition 4.7 (Attraktorbecken) *Die Menge der Trajektorien, die in den gleichen Attraktor führen, bilden das Attraktorbecken.*

Im Fall einer Störung, die nicht aus dem Attraktorbecken des gestörten Attraktors führt, läuft die Trajektorie wieder in den gleichen Attraktor. Das Langzeitverhalten des Systems wird deshalb nicht verändert und der Attraktor bleibt erhalten, ist also stabil. Je größer das Attraktorbecken, desto stabiler ist der Attraktor. Denn je mehr Zustände ein Attraktorbecken besitzt, je wahrscheinlicher ist es, dass eine Störung zu einem dieser Zustände führt.
Die Expressionszustände von Stammzellen befinden sich in einem Attraktor und deshalb sind in einem biologischen Experiment zur Charakterisierung der Genexpression nicht alle Zustände, die die Zellen annehmen können, verfügbar. Transiente oder persistente Störungen können in einem neuen Zustand führen, der entlang auf einer Trajektorie in einen neuen Attraktor läuft [38]. Es ist zu beachten, dass persistente Störungen eine Veränderung der Regeln des Systems verursachen können. Das ursprüngliche Genregulationsnetzwerk ist damit nicht mehr rekonstruierbar, zumindestens können die veränderten Regeln nicht mehr korrekt identifiziert werden. Es sei nun angenommen, dass eine Population von Zellen vorliegt, die sich jeweils in einem beliebigem Zustand des Attraktors, der diese Stammzellen charakterisiert, befinden. Die wesentliche Aussage der Abschnitte 4.4 und 4.3 war, dass eine große Stichprobe mit unterschiedlichen Anfangszuständen notwendig ist, damit der Reverse Engineering Algorithmus eine gute Übergangsstruktur des Genregulationsnetzwerkes findet. Nun stellt sich die Frage, ob die Zustände eines Attraktors diesen Forderungen gerecht werden.
Im allgemeinen kommt nur ein Bruchteil aller Zustände in einem Attraktor vor. Eine Möglichkeit, die Anzahl der zur Verfügung stehenden Zustände zu erhöhen, ist durch Störungen der Zellzustände zu versuchen, einen neuen Attraktor zu erreichen. Die Hoffnung besteht darin, durch eine Folge von Störungen mehrere verschiedene Attraktoren zu erhalten. Die beste Strategie wäre, jedes Bit in jedem Attraktorzustand einmal auf den entgegengesetzen Wert zu setzen. Eine Wiederholung der Störungen für jedes Gen im interessierenden Genregulationsnetzwerk ergibt die größtmögliche Anzahl von unterschiedlichen Zellzuständen, die für die Experimente zur Verfügung

stehen. Der Zustand in dem sich eine Zelle zur Zeit der Störung befindet, ist jedoch nicht bekannt. Daher ist es nicht möglich zu unterscheiden, ob gerade eine Zelle mit einem schon gesehenen oder mit einem neuen Zustand der Population zur Störung entnommen wurde. Es kann aber die Wahrscheinlichkeit $P(z)$, jeden Zellzustand mindestens einmal aus der Population zu ziehen, abhängig von der Anzahl der gezogenen Zellen z, bestimmt werden. Sie gibt Auskunft darüber, wieviel Zellen aus der Zellpopulation entnommen werden müssen, um mit hoher Wahrscheinlichkeit jeden möglichen Attraktorzustand mindestens einmal in der Teilpopulation repräsentiert zu haben:

$$P(z) = \sum_{i=0}^{S} (-1)^i \binom{S}{i} (1 - \frac{i}{S})^z \quad . \tag{4.17}$$

z ist die Anzahl der aus der Population gezogenen Zellen und S die Anzahl der Attraktorzustände in einem Attraktor[13]. Für 4 Attraktorzustände müssen z.B. 29 Zellen entnommen werden, um mit einer Wahrscheinlichkeit von 0.999 jeden Zustand mindestens einmal gezogen zu haben. Für 20 Zustände sind es dann schon 194.
Für das Lernen eines Modells dürfen nur Zustandsübergangspaare von Zellen mit dem gleichen Genregulationsnetzwerk genommen werden. Ist dies nicht der Fall, würde man versuchen, aus verschiedenen Genregulationsnetzwerken Gemeinsamkeiten zu finden. Es dürfen also die Regeln des Genregulationsnetzwerkes nicht verändert werden, um das Ziel, die maximal mögliche Anzahl von verschiedenen Trainingsdaten, zu erreichen.

Es sei die Zeit zwischen zwei Aktualisierungen des Genregulationsnetzwerkes mit τ, und mit $\hat{T}$ die durchschnittliche Zeit, für eine Folge von Netzwerkaktualisierungen, die wieder in einem stabilen Expressionsmuster enden, angegeben. Im Modell ist $\hat{T}$ die Zeit, die eine Trajektorie benötigt, um einen Attraktor zu erreichen. Außerdem definiert δT_S die Zeit für die Anwendung einer Störung und δT_C die Zeit bis das Expressionsmuster einer erfolgreich gestörten Zelle vollständig charakterisiert ist, nachdem sich ihr genetisches Netzwerk einmal aktualisierte. Daraus ergeben sich folgende Überlegungen:

1: *$\delta T_S < \tau$: Es können mehrere Störungen durchgeführt werden. Es sind insgesamt $\frac{\tau}{\delta T_S}$ und die gleiche Anzahl von Genen ist im ersten Zustand eines Zustandsübergangspaares bekannt.*

[13]Für die Herleitung dieser Wahrscheinlichkeit siehe Anhang B.

N	Mittelwert	Median	25% Perzentil	75% Perzentil	EW(Anz. Attraktoren)
12	2.8	2	1	4	3.5
20	3.7	2	1	4	4.5

Tabelle 4.8: Anzahl der Attraktoren, die von einem Attraktor ausgehend, durch eine Sequenz von Störungen erreicht wurden ($\mathcal{A}_S$). Es sind der Mittelwert und die Perzentile im Vergleich zu dem Erwartungswert $\sqrt{N}$ für die Anzahl aller Attraktoren eines Booleschen Netzes aus [24], aufgeführt. Es wurden insgesamt 3000 Netze mit $N = 20$ und 1011 für Netze mit $N = 12$ betrachtet. Für die statistischen Analysen wurde jeweils eine Stichprobe mit einer extrem hohen Anzahl an verschiedenen Attraktoren entfernt.

2: *$\tau < \delta T_S < \hat{T}$: Resultiert eine erste Störung in einem Zustand auf einer Trajektorie, wird eine zweite Störung auf einem Zustand auf einer Trajektorie wirken. In einer Folge von Störungen werden u.a. auch transiente Zellen (Zustand auf Trajektorie) gestört.*

3: *$\delta T_S > \hat{T}$: Resultiert eine erste Störung in einem Zustand auf einer Trajektorie, wird eine zweite Störung auf einem Zustand in einem Attraktor wirken. In einer Folge von Störungen werden immer stabile Zellen (Zustand im Attraktor) gestört.*

4: *$\delta T_C < \tau$: Nach Anwendung einer Störung hat die Zelle nur einmalig ihr Genregulationsnetzwerk aktualisiert und der unmittelbare Folgezustand wird charakterisiert. In dieser Arbeit wird vorausgesetzt, dass $\delta T_C < \tau$ und nur dann sind die gewonnenen Ergebnisse anwendbar.*

5: *$\delta T_C > \tau$: Nach Anwendung einer Störung hat sich das Genregulationsnetzwerk der Zelle mehrmals aktualisiert bis das Expressionsmuster vollständig gemessen wird. Es wird nicht die direkte Folge der Störung charakterisiert. Die unbekannten vorangegangenen Expressionszustände der Zelle müssen mit Hidden Variablen modelliert werden. Die Ergebnisse dieser Arbeit sind nicht anwendbar.*

In den folgenden Analysen wird davon ausgegangen, dass $\delta T_S > \hat{T}$ ist und die Störung somit immer in Attraktorzuständen wirkt. Dieser Grund macht es erforderlich, Attraktoren von Booleschen Netzen näher zu betrachten, um Schlussfolgerungen für reale biologische Experimente zu erhalten. Außerdem wird angenommen, dass die Störung transient ist. Nachdem sich das Netzwerk einmal neu aktualisierte, ist sie aufgehoben.

Umfangreiche Analysen über Attraktoren von Booleschen Netzen mit *Canalyzing Funktionen* wurden in [24] durchgeführt. Sie ergaben, dass Netze mit *Canalyzing Funktionen*, genau wie Netze mit $K = 2$, Ordnung aufzeigen. Ordnung bedeutet,

N	Mittelwert	Median	25% Perzentil	75% Perzentil
12	8.3	6	3	12
20	14.4	9	4	19

Tabelle 4.9: Anzahl der Attraktorzustände, die von einem Attraktor ausgehend, durch eine Sequenz von Störungen erreicht wurden. Der Mittelwert und die Perzentile sind aufgeführt. Es wurden insgesamt 3000 Netze mit $N = 20$ und 1011 für Netze mit $N = 12$ betrachtet. Für die statistischen Analysen wurde jeweils eine Stichprobe mit einer extrem hohen Anzahl an verschiedenen Attraktoren entfernt.

dass es viele Elemente im System gibt, die einen fixen Zustand besitzen. Im Gegensatz dazu stehen einige andere Elemente, deren Zustände auf eine komplexe Art und Weise schwanken. Sie sind in Regionen unterteilt, die durch Elemente mit fixen Zuständen getrennt sind. Dadurch entziehen sie sich einer gegenseitigen Beeinflussung [24]. Boolesche Netze, die diese Ordnung aufzeigen, haben wichtige Eigenschaften. Numerische Simulationen in [24] ergaben, dass ein *Erwartungswert* (EW) für die Anzahl von Attraktoren in diesen Netzen direkt aus der Zahl der Elemente im Netz abgeleitet werden kann:

$$\text{EW(Anzahl Attraktoren)} = \sqrt{N}. \tag{4.18}$$

Die gleiche Abschätzung gilt für den Erwartungswert der Anzahl von Zuständen in einem Attraktor:

$$\text{EW(Anzahl Zustände im Attraktor)} = \sqrt{N}. \tag{4.19}$$

Diese Schätzer lassen vermuten, dass es in den Systemen sehr wenige Attraktoren gibt, die zudem einige wenige Zustände enthalten. Die Mehrzahl der Booleschen Netze mit *Canalyzing Funktionen* haben Attraktoren mit wenigen Zuständen, wenn auch einige Netze wenige Attraktoren mit vielen Zuständen besitzen [24].
Anhand von Simulationen in der vorliegenden Arbeit wurde die Idee analysiert, den Attraktor in dem sich die Zellpopulation befindet, zu stören, um wenn möglich in einen neuen Attraktor zu springen. Die Simulation erfolgte für zufällig erzeugte Boolesche Netze mit 12 bzw. 20 Elementen. Die maximale Zahl von Eltern war jeweils auf $K = 4$ festgelegt. Die Zuordnung, wieviel Eltern ein Element hat, erfolgte zufällig und war nicht an der Vorgabe, gleiche Häufigkeiten für $k = 1, \ldots, 4$ (vgl. Abschnitte 4.2, 4.3 und 4.4), gebunden. Es wurde eine Zelle generiert und gewartet, bis sich ihr genetisches Netzwerk in einem Attraktor befand. Dieser ist einer der möglichen Attraktoren für das analysierte Boolesche Netz. Daraufhin wurde in allen Zuständen dieses Attraktors jedes Bit einmal, ohne dabei die anderen Bits zu verändern, auf den entgegengesetzten Wert gesetzt. War das Resultat einer Störung ein Zustand

auf einer Trajektorie, die in einen neuen Attraktor führte, wurde sich der noch nicht gesehene Attraktor gemerkt. Nach [24] ist zu erwarten, dass nur ungefähr 10% bis 20% der Störungen zu einem neuen Attraktor führen. In allen gemerkten Attraktoren wurden wiederum alle Zustände in jedem Bit gestört und getestet, ob sie in einen noch nicht gesehenen Attraktor resultieren. Das Ergebnis war eine Menge von Attraktoren, die durch Störungen ineinander übergehen können. Es sei diese Menge mit $\mathcal{A}_S$ bezeichnet:

Definition 4.8 ($\mathcal{A}_S$) *Die Menge $\mathcal{A}_S$ enthält alle Attraktoren, die von einem Attraktor $\mathcal{A}$ ausgehend durch eine Sequenz von 1-Bit Störungen erreicht werden.*

Es ist in [24] aufgeführt, dass es keinen Attraktor gibt, von dem aus jeder andere direkt durch Störungen erreichbar ist, auch wenn eine Menge von Attraktoren existiert, die durch eine Sequenz von Störungen ineinander übergehen können. Wieviel Attraktoren in den Simulationen im Mittel in $\mathcal{A}_S$ enthalten waren, ist in Tabelle 4.8 aufgelistet. Für Netze mit 20 Elementen ist zu erwarten, dass durch 1-Bit Störungen in 3 bis 4 (genau 3.7) verschiedene Attraktoren gesprungen wird. Der Wert 3.7 entspricht annähernd dem Erwartungswert $\sqrt{20}$ für die Anzahl der Attraktoren eines Booleschen Netzes. Es läßt sich daraus schließen, dass durch eine Sequenz von 1-Bit Störungen ein großer Teil der Attraktoren des Netzes erreicht werden kann. Die Erwartung für die Zustände in einem Attraktor multipliziert mit 3.7, ergibt die zu erwartende Anzahl von verschiedenen Attraktorzuständen, also $\sqrt{20} * 3.7 = 16.5$. Die Simulationen ergaben dafür einen Wert von 14.4.

Die Zellzustände in $\mathcal{A}_S$ bilden die Ausgangszustände für die biologischen Experimente, bei denen die Anfangszustände von l Genen festgelegt werden. Die Gene werden entweder überexprimiert oder unterexprimiert. Für einen Attraktorzustand gibt es $\binom{N}{l} * 2^l$ Möglichkeiten l Bits zu manipulieren. In jedem Attraktorzustand sind l Bits variabel und es ist existieren $\binom{N}{l}$ Möglichkeiten eine Kombination von l Bits gemeinsam festzulegen. Außerdem kann jede dieser Kombinationen auf insgesamt 2^l verschiedene Bitzustände gesetzt werden. Von ihnen werden sich nur $2^l - 1$ von dem originalen Attraktorzustand unterscheiden. Daher können die Manipulierungen für einen Attraktorzustand maximal zu $\binom{N}{l} * (2^l - 1)$ Zuständen führen, die nicht in $\mathcal{A}_S$ enthalten sind.
Numerische Analysen geben eine Vorstellung darüber, wieviel neue unterschiedliche Anfangszustände resultieren können (vgl. Tab. 4.10). Für jedes Boolesche Netz wurde $\mathcal{A}_S$ für einen beliebigen Attrakor $\mathcal{A}$ dieses Netzes bestimmt. Alle Zustände in $\mathcal{A}_S$ wurden in l Bits gestört. Die restlichen Bits blieben jeweils unverändert. Für jede mögliche Kombination von l Bits wurde jede mögliche Störung einmal angewendet. Die für den Reverse Engineering Algorithmus verfügbaren Anfangszustände ergaben

N	l	Mittelwert	Median	25% Perzentil	75% Perzentil
12	1	91	72	39	122
12	2	435.8	369.5	215	593.8
20	1	235.9	165	80	328
20	2	2226.2	1601	752	3091.8

Tabelle 4.10: Anzahl der unterschiedlichen Anfangszustände, die durch Manipulationen der Zustände in $\mathcal{A}_S$ erreicht wurden. Der Mittelwert und die Perzentile sind aufgeführt. Es wurden insgesamt 3000 Netze mit $N = 20$ und 3000 für Netze mit $N = 12$ betrachtet.

sich aus den Zuständen in $\mathcal{A}_S$ und den Zuständen, die durch die Störungen erhalten wurden und sich von den Zuständen in $\mathcal{A}_S$ unterschieden.
Falls $l = 1$ und $N = 20$ können nur für die Gene mit $k_{bio} = 1$ oder $k_{bio} = 2$ die Eltern bestimmt werden (vgl. Abbildung 4.12). Dafür sind ca. 100 bis 200 unterschiedliche globale Anfangszustände notwendig. Die Ergebnisse in Tabelle 4.10 deuten darauf hin, dass es möglich ist, diese Anforderung zu erfüllen. Im Mittel stehen ca. 240 unterschiedliche Anfangszustände zur Verfügung. Wenn dagegen zwei Gene gleichzeitig manipuliert werden können, erreicht man wegen größeren kombinatorischen Möglichkeiten eine noch größere Zahl an unterschiedlichen Anfangszuständen. Sie sind ausreichend, um ein Modell für die Topologie eines genetischen Netzwerkes mit 20 Elementen zu rekonstruieren (vgl. Abb. 4.13). Als Fazit ist zu sagen, dass die notwendige Menge an verschiedenen Zuständen theoretisch erzeugt werden kann!

Die Attraktorzustände entsprechen einer Teilmenge aus allen Zuständen für ein Boolesches Netz. Befindet sich eine Zellpopulation in einem Attraktor, könnte dieser gegebenenfalls durch eine vollständige Genexpressionsanalyse charakterisiert werden. Eine zweite Population mit dem gleichen Genregulationsnetzwerk und im gleichen Attraktor dient zur Gewinnung der Zustandsübergangsdaten. Das ergibt eine prior-Verteilung, nicht für die Übergangswahrscheinlichkeiten, aber für den Anfangszustand der Zustandsübergangspaare. Würden bestimmte Gene mit einer hohen Wahrscheinlichkeit immer hoch bzw. immer niedrig exprimiert sein, wäre diese Wahrscheinlichkeit ein guter Schätzer für den Zustand dieses Gens im Anfangszustand. Dies bedeutet, dass für weitere Elemente ein Schätzer für den Zustand existiert, auch wenn nur l Elemente einen bekannten Anfangszustand haben. Tatsächlich ergeben die numerischen Simulationen in [24], dass ca. 70% der Elemente einen festen Zustand besitzen, der auch in allen verschiedenen Attraktoren des Netzes keine Veränderung aufzeigt. Dies gilt jedoch nur für Boolesche Netze mit *Canalyzing Funktionen* oder $K = 2$.
Die in diesem Abschnitt gemachten Aussagen relativieren sich, wenn spezielle Genregulationsnetzwerke betrachtet werden. Denn dort sind zuerst die für dieses Netzwerk

spezifischen Eigenschaften zu untersuchen, die sich stark von den allgemeinen Eigenschaften unterscheiden können. Zudem kann vorhandenes Zusatzwissen über das interessierende System zu einer verbesserten Ausgangssituation führen[14].

4.7 Gestörte Trainingsdaten

Die bisher vorgestellten Simulationsergebnisse beruhten ausschließlich auf Trainingsdaten, die keiner Störung unterlagen. Bei Datenerhebungen oder durch die experimentelle Eigenschaften treten in realen Anwendungen immer Fehler auf, die in den Simulationen berücksichtigt werden müssen. Die Fehlerrate bei der Expressionsanalyse auf Basis von Einzelzellen liegt bei ca. 5% [M. Cross, private Mitteilung]. Bei der Manipulation eines einzelnen Gens beträgt sie ca. 1% [M. Cross, private Mitteilung]. Für die Hemmung ist dafür aber eine gut gewählte inhibitorische RNA Sequenz notwendig.
Simulationen für die Rekonstruktion der Modellstruktur wurden für Boolesche Netze mit 20 Genen durchgeführt und ihre Ergebnisse denen der letzten Abschnitte gegenübergestellt (Abb. 4.17 und 4.18). Die Übergangsstruktur wurde mit der Reverse Engineering Strategie aus Abschnitt 4.4 gelernt. Die verwendeten Booleschen Regeln waren ausschließlich *Canalyzing Funktionen*. Die Anfangszustände der Gene wurden zu 99% korrekt aus den in-silico Zellen abgelesen. 1% der Gene erhielten den entgegengesetzten Zustand zugewiesen. Die Folgezustände hatten in 95% der Fälle den richtigen Wert.
Diese Fehlerraten führen zu einer Verzögerung der Identifizierung der Eltern.
Die Wahrscheinlichkeit, die Eltern falsch zu bestimmen, konvergiert langsamer. Die Sättigungswahrscheinlichkeiten werden trotz der Störungen erreicht und $P_{identified}(k)$ und $P_{positive}(k)$ verändern sich nicht (vgl. Tab. 4.6 und 4.11). Bei einem genügend großen Stichprobenumfang ist es dem Reverse Engineering Algorithmus trotz der Störungen möglich, Ergebnisse, wie bei ungestörten Daten, zu liefern. Einzig die Entscheidung aufgrund der Gleichheitsbeziehung $\hat{MI}(X_i, \mathbf{Pa}(X_i)) = \hat{H}(X_i)$ ist nicht mehr zulässig (Siehe Abb. 4.18). Durch die auftretenden Störungen sind die Beziehungen zwischen den Variablen nicht mehr deterministischer sondern stochastischer Art und eine maximale Abhängigkeit zwischen Variablen ist nicht mehr erkennbar. In Kapitel 2 wurde kurz begründet, wann sich maximale Kovarianz zwischen Zufallsvariablen ergibt. Demnach müssen die Wahrscheinlichkeiten p_{ij} für das Auftreten der Belegungen $X_i = i$ und $\mathbf{Pa}(X_i) = j$ gleich den Wahrscheinlichkeiten p_i und p_j oder gleich 0 sein. Bei gestörten Trainingsdaten werden jedoch Beobachtungen gemacht, die so in Wirklichkeit nicht existieren, was zu $\hat{p}_{ij} \neq \hat{p}_i \neq \hat{p}_j$ führt. Dadurch wird es unmöglich die Gleichung (2.2) herzuleiten und die Abhängigkeiten zwischen Zu-

[14] Siehe Kapitel 5

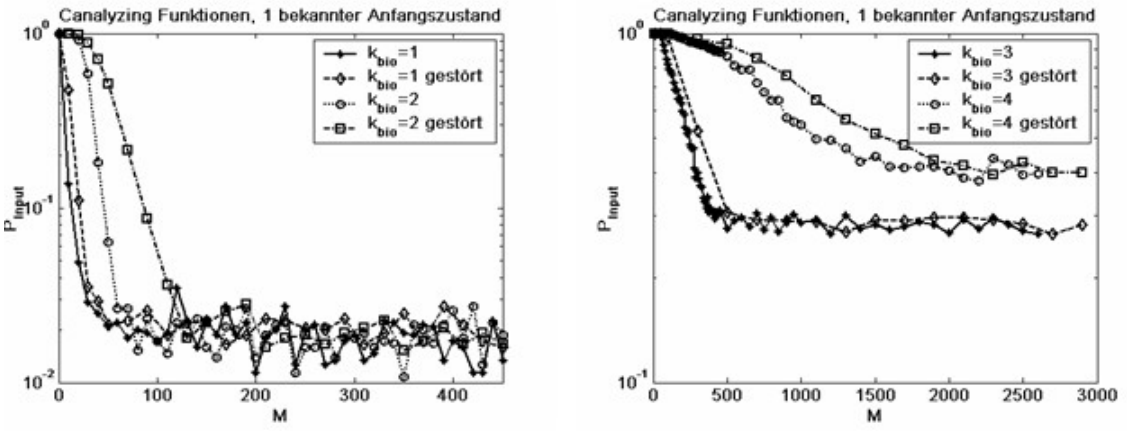


Abbildung 4.17: Reverse Engineering mit *Canalyzing Funktionen* und verrauschten Daten. Als Abbruchbedingung wurde das Kriterium 1 gewählt ($N = 20$, $K = 4$, $l = 1$). Es sind die Kurven für verrauschte Daten, im Vergleich mit ungestörten Daten zu sehen.

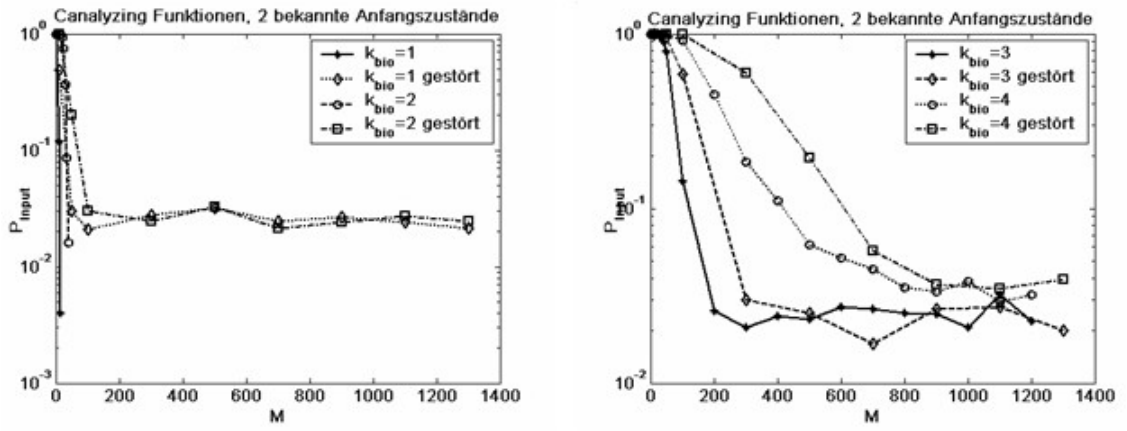


Abbildung 4.18: Reverse Engineering mit *Canalyzing Funktionen* und verrauschten Daten. Als Abbruchbedingung wurde das Kriterium 2 gewählt ($N = 20$, $K = 4$, $l = 2$). Es sind die Kurven für verrauschte Daten, im Vergleich mit ungestörten Daten zu sehen.

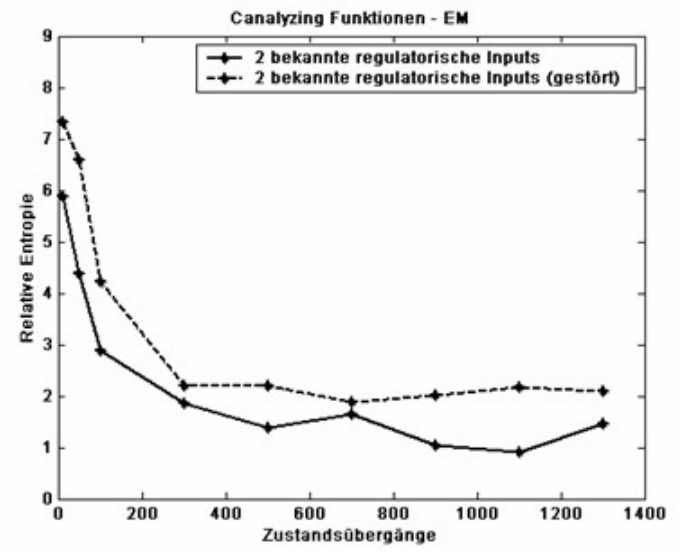


Abbildung 4.19: EM - Schätzen der Parameter bei bekannter Struktur, *Canalyzing Funktionen* und verrauschten Daten. Es ist die relative Entropie abgebildet (EM, $N = 12$, $K = 4$, $l = 2$). Es sind die Kurven für verrauschte Daten, im Vergleich mit ungestörten Daten zu sehen.

	l=1		l=2	
k	$P_{identified}(k)$	$P_{positive}(k)$	$P_{identified}(k)$	$P_{positive}(k)$
1	$\approx 100\%$	$\approx 98\%$	$\approx 100\%$	$\approx 98\%$
2	$\approx 100\%$	$\approx 98\%$	$\approx 100\%$	$\approx 98\%$
3	$\approx 82\%$	$\approx 99\%$	$\approx 100\%$	$\approx 99\%$
4	$\approx 82\%$	$\approx 99\%$	$\approx 99\%$	$\approx 99\%$

Tabelle 4.11: $P_{identified}(k)$ und $P_{positive}(k)$ für *Canalyzing Funktionen* bei verrausschten Daten (vgl. Abb. 4.6). Die Werte gelten jeweils ab M, bei dem Konvergenz eintritt. Sie haben sich im Vergleich mit Tabelle 4.6 nicht verändert. Die Veränderung in $P_{positive}(1)$ und $P_{positive}(2)$ für $l = 2$ folgt daraus, dass die Abbruchbedingung $\hat{MI}(X_i, \mathbf{Pa}(X_i)) = \hat{H}(X_i)$ nicht mehr anwendbar ist.

fallsvariablen können nur noch durch statistische Tests bewertet werden. REVEAL in seiner originalen Konfiguration und das Kriterium 2 in Abschnitt 4.4 sind deshalb nicht mehr anwendbar!
Das Lernen der Parameter mit gestörten Trainingsdaten erfolgte unter der Annahme, dass zwei Anfangszustände gemeinsam bekannt sind. Gute Schätzer für Parameter sind nicht identifizierbar, wenn die Anfangszustände der Gene nicht gemeinsam gemessen werden können (Abschnitt 4.5). Wie die gelernten Parameter die Daten erklären, wurde an einem ungestörten Datensatz untersucht. Das Ergebnis ist in Abbildung 4.19 zu sehen. Es werden auch mit gestörten Trainingsdaten noch relativ gute Parameterschätzer identifiziert. Die relative Entropie erreicht nicht den gleichen Wert wie bei ungestörten Trainingsdaten, aber der Unterschied bleibt gering.
Die Fehler, die bei der Manipulierung von Genen und der Expressionsanalyse bei Einzelzellen auftreten, bewegen sich in einer Größenordnung, die keinen wesentlichen Einfluß hat, bei genügend vielen Trainingsdaten. Folglich ist es auch mit realen Daten möglich ein gutes Modell zu lernen, vorausgesetzt $l = 2$.

4.8 Zusammenfassung

Die in diesem Kapitel beschriebenen Analysen ergaben, dass die Reverse Engineering Strategie des partiellen Lernens der Struktur für die Anwendung auf unvollständige Trainingsdaten nach Definition 1.4 geeignet ist. Die Topologie des Modells wird gelernt, indem der Einfluss jedes regulierenden Gens auf die Zielgene unabhängig voneinander charakterisiert wird. Diese Strategie hat den Vorteil, dass auf keine fehlenden Werte mehr geachtet werden muss, und den Nachteil, dass nur die Eltern identifizierbar sind, die allein einen signifikanten Einfluss auf das regulierte Gen haben. Es wurde gezeigt, dass nur ein Bruchteil der Booleschen Funktionen diese Eigenschaft aufzeigt. Deshalb ist die Modellstruktur nicht rekonstruierbar ist, wenn die Anfangszustände der Gene nicht gemeinsam bekannt sind. Aber Elemente mit höchster Informationsübertragung lassen sich schon bei einem geringen Stichprobenumfang identifizieren. Obwohl nicht alle Eltern lernbar sind, können doch die „Hauptspieler“ schon mit geringem Aufwand festgestellt werden. Durch die Wahl einer sehr niedrigen Irrtumswahrscheinlichkeit sind die identifizierten Elemente sehr sicher auch korrekte Eltern. Es ist unmöglich, gute Schätzer für die Modellparameter zu lernen, falls der Anfangszustand nur eines Gens beobachtet wird.
Andere Schlüsse sind zu ziehen, falls in einem biologischen Experiment zwei Anfangszustände gemeinsam bekannt sind. Denn die Inputelemente der meisten Booleschen Funktionen teilen sich in Informationsquellen mit maximal zwei Elementen auf. Ihr Einfluss auf die regulierten Gene ist unabhängig voneinander bestimmbar und die Modellstruktur kann rekonstruiert werden. Dafür sind jedoch sehr viele Zu-

standsübergangsdaten notwendig.
Der große Stichprobenumfang stellt einen Nachteil für die Anwendung auf biologische Systeme dar. Es wird angenommen, dass die Zellzustände während eines Experimentes Zustände in Attraktoren sind. Die mittlere Anzahl der Attraktoren in einem System mit *Canalyzing Funktionen* und die mittlere Anzahl der Attraktorzustände lassen erwarten, dass in realen biologischen Anwendungen nicht der notwendige Stichprobenumfang zur Verfügung steht. Methoden, die Zellen aus einem Attraktor in einen anderen zwingen, um die Menge der Ausgangszustände für die genetischen Manipulationen zu vergrößern, führen zu einer ausreichenden Erhöhung des Stichprobenumfangs. Sind genügend Zellen mit diesen Ausgangszuständen verfügbar, damit jede Kombination von zwei Genen auf ihre vier möglichen Expressionszustände einmal manipuliert werden kann, enthalten die Zustandsübergangstabellen ausreichend sich unterscheidende Anfangszustände. Wenn nur ein Gen festlegbar ist, enthalten sie genügend unterschiedliche Anfangszustände, um die wichtigsten regulierenden Gene zu finden.
Zusammenfassend ist zu sagen, dass ein gutes Modell für ein genetisches Netzwerk aus unvollständigen Trainingsdaten nach Definition 1.4 gelernt werden kann, wenn die Anfangszustände von zwei Genen gleichzeitig manipuliert werden.
Es ist notwendig, spezifische Eigenschaften des biologischen Systems, das untersucht werden soll, zu analysieren. Die Integration von Vorwissen kann zu einer wesentlichen Verringerung des notwendigen Stichprobenumfangs führen.

Kapitel 5

Das genetische Netzwerk in Blutstammzellen

Für eine aktuelle biologische Fragestellung soll untersucht werden, ob ein Modell für das genetische Netzwerk rekonstruierbar ist und Vorwissen unterstützend wirkt. Zuerst werden Annahmen über das biologische System erläutert. Der zweite Abschnitt erklärt wie Vorwissen über die strukturellen Beziehungen in den Reverse Engineering Algorithmus integriert werden kann und stellt die Ergebnisse der Simulationsstudien vor. Im letzten Abschnitt wird auf das dynamische Verhalten des biologischen Systems eingegangen, falls es mit einem Booleschen Netz modelliert wird.

5.1 Hypothese

Das Genregulationsnetzwerk in multipotenten Blutstammzellen enthält nach Meinung von M. Cross [private Mitteilung] sechs wichtige Gene. Literaturrecherchen von M. Cross ergaben eine Hypothese für ihre Abhängigkeiten (Abb. 5.1). Die Kanten stellen Ergebnisse aus separaten Experimenten von verschiedenen Forschungsgruppen dar. Die gestrichelte Kante von SCL nach $PU.1$ bedeutet, dass diese Relation noch nicht analysiert wurde, aber von Wichtigkeit sein könnte. Projekte, die die gemeinsame Auswirkung aller Input-Gene auf ein Zielgen untersuchen, wurden bisher noch nicht durchgeführt. Dieses Kapitel soll eine Aussage darüber treffen, ob es möglich ist, ein Modell für das genetische Netzwerk in multipotenten Blutstammzellen aufzustellen, wenn das Expressionsmuster von einzelnen Zellen gemessen wird.
Die einzelnen Kanten und ihre Qualität der Hypothese wurden von M. Cross bewertet, je nachdem ob er den Daten eines Experimentes und den daraus gezogenen Schlüssen vertraut. Folgende Interpretationen für den qualitativen und quantitativen Einfluss ergeben sich für die Bewertungen:

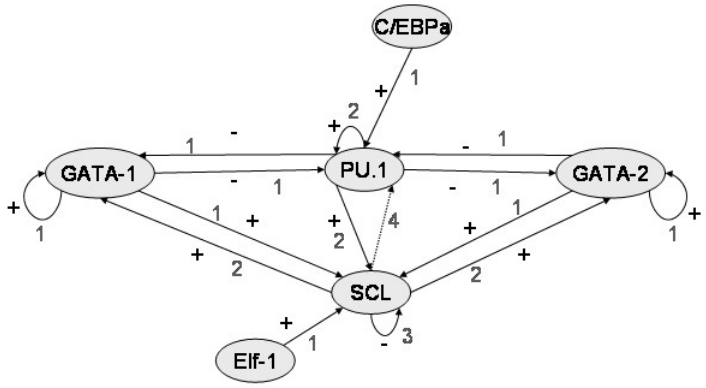


Abbildung 5.1: Das Kernnetzwerk mit bewerteten Kanten. Die Bewertungen geben gleichermaßen Auskunft über die Unsicherheit der Kanten und über die Art ihrer Wirkung.

1: *In Literatur überzeugend gezeigt, aber noch nicht an FDCPmix Zellen untersucht* [1].

2: *In Literatur weniger überzeugend gezeigt, aber diese Beziehung liegt nahe.*

3: *In Literatur nicht überzeugend gezeigt.*

4: *Qualitative Analysen wurden noch nicht durchgeführt, wären aber sinnvoll. Quantitative Beziehung ist nicht bekannt.*

Für Computersimulationen, aus denen Aussagen über das reale System hergeleitet werden können, werden Boolesche Netze erzeugt, die dieser Hypothese ähneln.

Definition 5.1 *Boolesche Netze sind* ***ähnlich*** *zu der Hypothese in Abb. 5.1, wenn eine Kante mit einer Wahrscheinlichkeit, die von ihrer Bewertung abhängt, in den Graphen eines Booleschen Netzes hinzugefügt wird. Kanten mit Bewertung* 1 *treten sehr wahrscheinlich im realen System auf und sind in* 90% *der Netze enthalten. Weniger sicher ist man sich über Kanten mit Bewertung* 2*. Sie kommen in den Modellen mit einer Wahrscheinlichkeit von* 66% *vor. Über alle anderen Kanten ist kein genaueres Wissen vorhanden und sie werden zu* 50% *erzeugt. Die Qualität der Relationen ergibt sich ebenfalls aus der Hypothese. Ist die Qualität einer Kante mit* $-$ *bezeichnet, ist das regulierende Gen ein Repressor. Ein* $+$ *kennzeichnet aktivierende Eigenschaften.*

[1] FDCPmix Zellen sollen aber für die Expressionsanalyse auf Basis von Einzelzellen verwendet werden.

k	Anz. Regeln in Hypothese	Schnittmenge mit k_{bio}	Schnittmenge mit k_1
1	2	100%	100%
2	4	100%	100%
3	8	100%	100%
4	15	100%	100%

Tabelle 5.1: Analyse der Regeln, die in der Hypothese auftreten für $k = 1$ bis $k = 4$.

Boolesche Netze nach dieser Definition generieren künstliche Trainingsdaten, aus denen ein Dynamisch Bayessches Netz rekonstruiert wird.
Da über den kombinatorischen Einfluss der regulierenden Gene nichts weiteres bekannt ist, motivieren ihn folgende Aussagen:

- *Repressoren wirken in den meisten biologischen Systemen dominant [31] und stehen deshalb mit jedem anderen regulierenden Gen in einer AND Relation. Dies stellt sicher, das das Zielgen immer inaktiv ist, sobald ein Repressor einwirkt.*
- *Aktivatoren wirken entweder synergistisch [31] oder kompetetiv [M. Cross, private Mitteilung]. Ihre Relation zueinander wird deshalb zufällig zwischen AND und OR ausgewählt.*

Die resultierenden Booleschen Funktionen sind eine Teilmenge der *Canalyzing Funktionen*. Dies ergab eine Analyse des Modellraumes, bei der 10.000 in-silico Zellen generiert wurden (Tab. 5.1). Nach diesen Booleschen Funktionen zu urteilen, können die Eltern für alle Relationen mit bis zu vier regulierenden Genen korrekt identifiziert werden, auch wenn die Anfangszustände nicht gemeinsam bekannt sind. Wieviel Zustandsübergangspaare notwendig sind, wird im folgenden Abschnitt ohne und mit der Integration von Vorwissen über die Topologie bestimmt. Für fünf regulierende Gene sind aufgrund der hohen Anzahl von möglichen Booleschen Funktionen keine theoretischen Analysen möglich und es kann sich nur auf Simulationen bezogen werden.

5.2 Reverse Engineering

Die Bewertungen in Abbildung 5.1 repräsentieren Vorwissen über die strukturellen und quantitativen Zusammenhänge in dem Genregulationsnetzwerk von multipotenten hämatopoetischen Stammzellen. In diesem Abschnitt wird untersucht, ob die Integration der qualitativen Hypothesen in den Reverse Engineering Algorithmus, zu einer Verringerung der notwendigen Trainingsvektoren führt. Neben der Struktur

sollen die Parameter geschätzt werden, um das wahrscheinlichste Modell, gegeben der Daten, zu erhalten.
Das partielle Lernen einer Struktur erfordert einen χ^2-Unabhängigkeitstest für jede Kombination von bekannten Inputelementen mit jedem Outputelement. Eine andere Methode, *Likelihood Ratios*, bewertet Beobachtungen als Evidenz für oder gegen die Hypothese H_1, dass zwei Zufallsvariablen abhängig sind, gegenüber der Hypothese H_0, dass sie unabhängig sind:

$$H_0 : p_{ij} = p_i * p_j.$$

Die Likelihood Ratio oder der Bayes Faktor für ein Outputelement X_i und die aktuellen Kandidaten für seine Eltern $\mathbf{Pa}(X_i)$ ist [34]:

$$LR = \frac{P(D|H_0)}{P(D|H_1)} = \frac{\prod_{k_i=1}^{r_i} P(X_i = k_i) * \prod_{j_i=1}^{q_i} P(\mathbf{Pa}(X_i) = j_i)}{\prod_{k_i j_i} P(X_i = k_i, \mathbf{Pa}(X_i) = j_i)} \quad , \tag{5.1}$$

denn die Wahrscheinlichkeiten für das Auftreten der Hypothesen sind jeweils mit Multinomialverteilungen unter den jeweiligen Annahmen beschreibbar[2]. Die Multinomialkoeffizienten kürzen sich heraus.
Sind die Beobachtungen unter Annahme von H_1 wahrscheinlicher als unter Annahme von H_0, dann ist dies eine Evidenz für H_1 gegenüber H_0. Die Likelihood Ratio spiegelt die Stärke dieser Evidenz wieder. Sie ergibt sich daraus, um wie vieles größer die Wahrscheinlichkeit der Beobachtungen unter Annahme von H_1 ist.
Vorwissen über die Modellstruktur läßt sich mit dieser Strategie in den Reverse Engineering Algorithmus integrieren. Denn Vermutungen können als zusätzlich gesehene Beobachtungen verstanden werden und definieren prior-Verteilungen für die beiden Hypothesen. Die prior-Verteilungen $P(H)$ werden anhand der eigentlichen Beobachtungen aktualisiert und das Ergebnis sind posterior-Verteilungen $P(H|D)$:

Definition 5.2 (Theorem von Bayes) $P(H|D) = \frac{P(D|H)P(H)}{P(D)}$

Ihr Unterschied ist wiederum als Likelihood Ratio darstellbar:

$$LR = \frac{P(H_0|D)}{P(H_1|D)} = \frac{P(D|H_0)}{P(D|H_1)} * \frac{P(H_0)}{P(H_1)} \quad . \tag{5.2}$$

Die Wahrscheinlichkeiten $P(D)$ kürzen sich heraus.
Auf Basis der posterior-Wahrscheinlichkeiten kann eine Entscheidung für eine der beiden Hypothesen getroffen werden. Eine Likelihood Ratio kleiner als 1, wird als

[2]In dieser Arbeit handelt es sich um Binomialverteilungen, da eine Zufallsvariable X_i nur zwei Zustände annimmt.

Evidenz für H_1 gegenüber H_0 bewertet. Denn je wahrscheinlicher die Hypothese H_1 gegenüber H_0 ist, desto kleiner wird die Likelihood Ratio. Ist die Likelihood Ratio kleiner als $\frac{1}{r}$, ist H_1 r-mal wahrscheinlicher als H_0. Ist sie dagegen größer als 1, entspricht das einer Evidenz für H_0 gegenüber H_1. Eine Likelihood Ratio größer als $r \quad (r > 1)$ besagt, dass H_0 r-mal wahrscheinlicher als H_1 ist. Damit sich für oder gegen eine Hypothese entschieden werden kann, muss ein Wert für r festgelegt oder ein statistischer Test gemacht werden. Nach [34] gilt:

$$-2 * \ln LR \sim \chi^2_{r_i-1,q_i-1;1-\alpha} \tag{5.3}$$

Je kleiner die Likelihood Ratio ist, desto größer wird $-2 * \ln LR$. Auf Basis der tabellierten Signifikanzschranken für die χ^2-Verteilung kann die Hypothese H_0, dass X_i unabhängig von $\mathbf{Pa}(X_i)$ ist, getestet werden. Außerdem wird in [34] gezeigt, dass:

$$-2 * \ln LR(X, \mathbf{Pa}(X_i)) = \ln 4 * M * MI(X_i, \mathbf{Pa}(X_i)). \tag{5.4}$$

Deshalb ist die Verwendung von Likelihood Ratios, um Vorwissen einzubinden, analog zu dem Vorgehen für das partielle Lernen der Struktur ohne Vorwissen.
Likelihood Ratios sind ein Ansatz, der in statistischen *Likelihood Methoden* verwendet wird. Durch die Festlegung von r erfolgt die Entscheidung für eine Hypothese gegenüber einer zweiten Hypothese. Indem die Entscheidung mit einem statistischen Signifikanztest getroffen wird, überführt man den Likelihood Ratio Ansatz in die *Frequentistische Statistik*[3].
Aus den Bewertungen der Kanten in Abbildung 5.1 ergeben sich folgende prior-Verteilungen für H_0 und H_1:

- *1:* $P(H_0) = 0.1$, $P(H_1) = 0.9$
- *2:* $P(H_0) = 0.34$, $P(H_1) = 0.66$
- *3,4:* $P(H_0) = 0.5$, $P(H_1) = 0.5$

Wenn die Möglichkeit besteht, die Anfangszustände von zwei Genen gemeinsam zu messen, ist es auch bei dieser Strategie notwendig zu bestimmen, ob die beobachtete Informationsübertragung von beiden Kandidaten oder eigentlich nur von einem ausgeht. Für jede Teilmenge $\mathbf{Pa}_T(X_i)$ von $\mathbf{Pa}(X_i)$ muss wieder getestet werden, ob die Abhängigkeit zwischen $\mathbf{Pa}(X_i)$ und X_i signifikant größer ist, als die Abhängigkeit

[3]Im allgemeinen ist es üblich, zwischen zwei Hypothesen anhand eines Signifikanztests zu entscheiden. Die Wahrscheinlichkeiten einen Fehler 1. Art oder einen Fehler 2. Art zu machen, lassen sich über die Wahl der Irrtumswahrscheinlichkeit α und des Stichprobenumfangs kontrollieren. Likelihood Ratios sind zu bevorzugen, falls die *Evidenz* einer Hypothese gegenüber einer zweiten Hypothese bewertet werden soll.

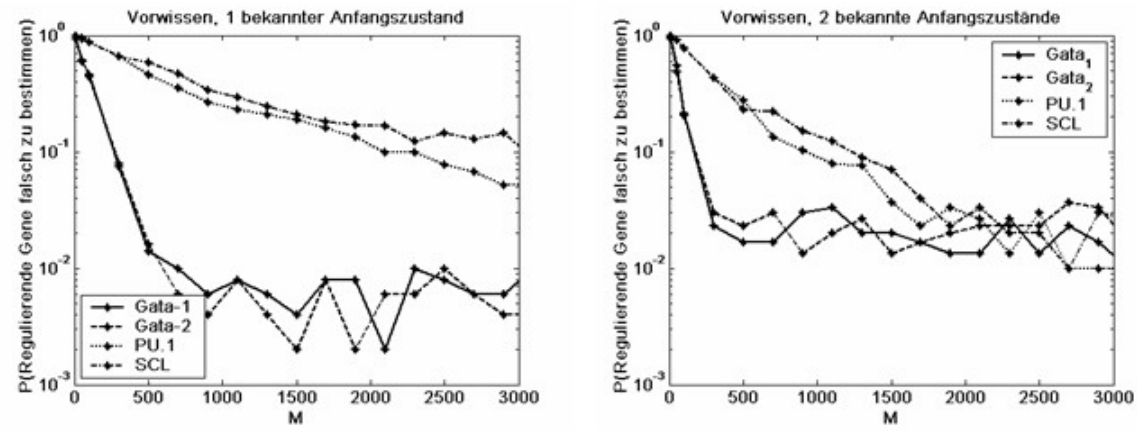


Abbildung 5.2: Wahrscheinlichkeiten, die Input-Gene für $GATA-1$, $GATA-2$, $PU.1$ und SCL falsch zu bestimmen, falls Vorwissen berücksichtigt wird. Wenn zwei Anfangszustände bekannt sind, treten geringfügig mehr falsch positive Input-Gene auf (vgl. Tab. 5.2).

zwischen $\mathbf{Pa}_T(X_i)$ und X_i. $\mathbf{Pa}_T(X_i)$ enthält einen Kandidaten weniger als $\mathbf{Pa}(X_i)$. Wegen der Beziehungen 4.12 und 5.4 kann die Nullhypothese

$$H_0: \quad -2 * \ln(LR(X_i, \mathbf{Pa}(X_i)) - LR(X_i, \mathbf{Pa}_T(X_i))) = 0 \tag{5.5}$$

unter der χ^2-Verteilung getestet werden. Die Freiheitsgrade sind

$$df(X_i, \mathbf{Pa}(X_i)) - df(X_i, \mathbf{Pa}_T(X_i)).$$

Mit dieser Methode läßt sich feststellen, ob das Element, welches nicht in $\mathbf{Pa}_T(X_i)$ enthalten ist, ein Elternelement von X_i ist. Die Likelihood Ratios sind, wie oben, die Quotienten der posterior-Wahrscheinlichkeiten. Die prior-Wahrscheinlichkeiten für eine mehrelementige Menge $\mathbf{Pa}(X_i)$ richtet sich nach dem Inputelement, dessen Kante eine höhere Bewertung hat. Denn es definiert die Erwartung des Beobachters, dass diese Menge Inputelemente enthält.
Somit ergibt sich ein Reverse Engineering Algorithmus, der strukturelles Vorwissen berücksichtigt. Die Prozedur LEARN_STRUCTURE_PARTLY verändert sich wie folgt:

FOR EACH output element o

IF $-2 * \ln LR(\mathbf{Pa}_j, o) > \chi^2_{(q_i-1)*(r_i-1);1-\alpha}$ *AND no. elements in* $\mathbf{Pa}_j = 1$

parents[o] $\leftarrow \mathbf{Pa}_j$

IF $-2 * \ln LR(\mathbf{Pa}_j, o) > \chi^2_{(q_i-1)*(r_i-1);1-\alpha}$ *AND no. elements in* $\mathbf{Pa}_j > 1$

FOR EACH element X *in* $\mathbf{Pa}_j$
$\mathbf{Pa}_{j_T} = \mathbf{Pa}_j \setminus X$
IF $-2 * ln(LR(\mathbf{Pa}_j, o) - LR(\mathbf{Pa}_{j_T}, o)) > \chi^2_{df-df_T;1-\alpha_T}$
parents[o] $\leftarrow X$
ELSE
*RETURN /*Other subsets has been checked already*/*
RETURN G

Für alle Simulationsstudien in diesem Abschnitt wurden 300 Boolesche Netze zufällig erzeugt, die ähnlich zu der Hypothese in Abbildung 5.1 waren (vgl. Def. 5.1). Die Irrtumswahrscheinlichkeiten betrugen jeweils $\alpha = 0.001$ und $\alpha_T = 0.0001$. Die niedrige Wahl der Irrtumswahrscheinlichkeiten läßt sich damit begründen, dass die in das Modell aufgenommenen Eltern mit einer hohen Wahrscheinlichkeit einen realen Einfluß auf das Outputelement haben sollen. Für die Nullhypothese (5.5) fiel die Wahl auf eine noch kleinere Irrtumswahrscheinlichkeit, denn ein Inputelement tritt in mehreren Mengen $\mathbf{Pa}(X_i)$ auf und wird dadurch mehrmals auf Unabhängigkeit mit dem gleichen Outputelement getestet (vgl. Abschnitt 4.4). Die Trainingsdaten wurden wie in Abschnitt 4.7 gestört. Außerdem musste die Bedingung, dass alle M Zellen eines Experimentes keine redundanten globalen Anfangszustände haben, nicht mehr erfüllt sein. Dadurch entsprechen die simulierten Trainingsdaten sehr gut realen Trainingsdaten. Redundanz in den globalen Anfangszuständen bewirkt jedoch bei gleichem Stichprobenumfang M einen geringen Informationsverlust. Es ist weniger Information in den M Anfangszuständen enthalten, als unter der Bedingung, dass keine redundanten Zustände gegeben sind. Denn das mehrmalige Auftreten des gleichen Zustandsübergangs resultiert in keiner neuen Information. Der Informationsverlust ist minimal, da jeder globale Anfangszustand mit gleicher Wahrscheinlichkeit gezogen wird und daher die Entropie der Grundmenge für ein Inputelement maximal bleibt.
Vorwissen kann dazu führen, dass schon bei wenigen Zustandsübergangsdaten die Korrelation zwischen einem Element X_i und seinen Eltern $\mathbf{Pa}(X_i)$ korrekt erkannt wird. Nimmt man aber fälschlicherweise eine Abhängigkeit an, werden sehr viele Daten benötigt, um diese Hypothese zu relativieren. Im Mittel können sich daher die Wirkungen aufheben und geringfügig mehr falsch positive Eltern beobachtet werden. Wie gut der Reverse Engineering Algorithmus im Schnitt die Eltern korrekt identifiziert, wenn die Anfangszustände nicht gemeinsam bekannt sind, zeigt Abbildung 5.3. Die Berücksichtigung von Vorwissen bringt ein etwas besseres Ergebnis, aber keine wesentliche Verringerung des Stichprobenumfangs. Ein anderes Ergebnis liefern Simulationsstudien, wenn zwei Anfangszustände beobachtet werden (Abbildung 5.4). Qualitatives Vorwissen bewirkt in diesem Fall eine umfangreiche Verringerung der

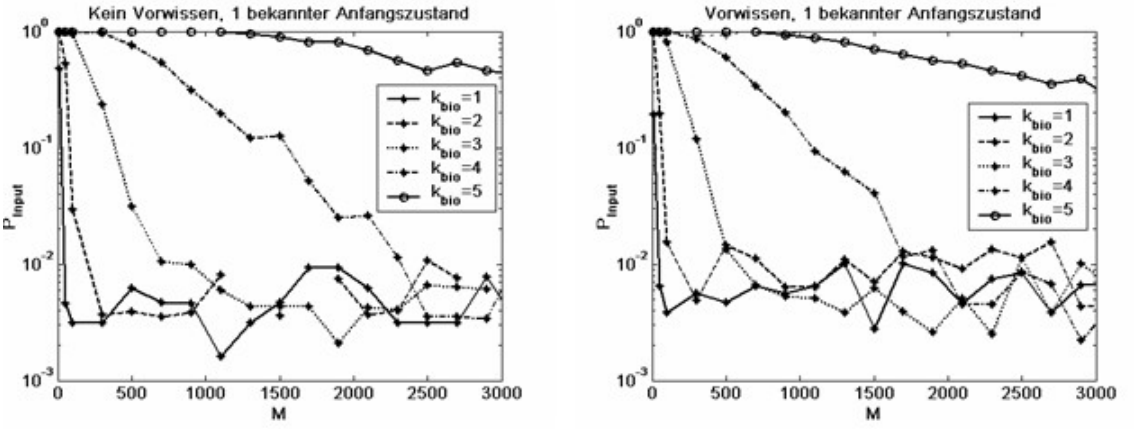


Abbildung 5.3: Reverse Engineering mit und ohne Vorwissen ($l = 1$). Die Konvergenzwerte für $k_{bio} = 1$ bis $k_{bio} = 4$ ergeben sich aus Fehlern des χ^2-Unabhängigkeitstests. Die Inputelemente für $k_{bio} = 5$ sind dagegen nicht immer korrekt identifizierbar. Der kleinste erkennbare Fehler für $P_{Input}(k_{bio})$ ist $\frac{1}{1800}$. Er tritt ein, falls in allen 300 Booleschen Netze alle Elemente die gleiche Anzahl k_{bio} von Eltern haben. Für einige Stichproben wurden alle Inputelemente für alle k_{bio}-Regeln korrekt gefunden.

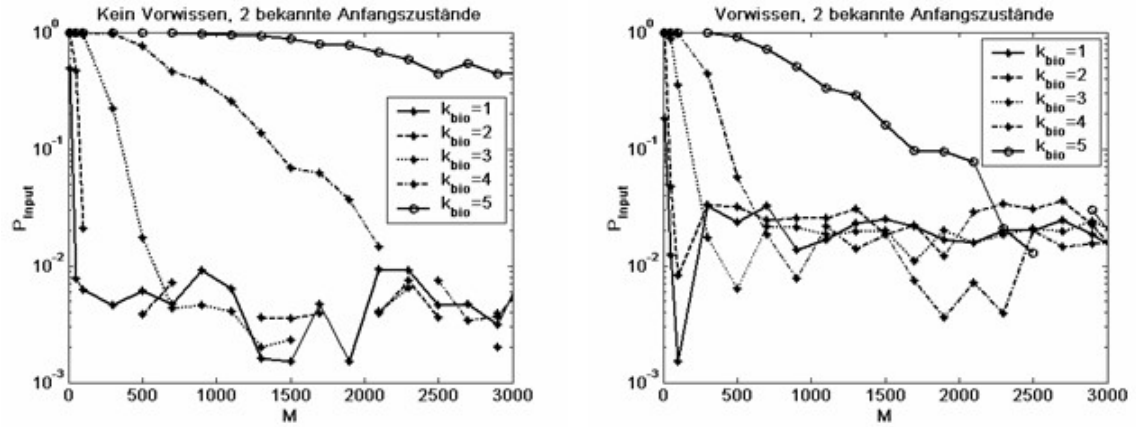


Abbildung 5.4: Reverse Engineering mit und ohne Vorwissen ($l = 2$). Die Konvergenzwerte für $k_{bio} = 1$ bis $k_{bio} = 4$ ergeben sich aus Fehlern des χ^2-Unabhängigkeitstests. Die Inputelemente für $k_{bio} = 5$ sind dagegen nicht immer korrekt identifizierbar, falls kein Vorwissen vorhanden ist. Der kleinste erkennbare Fehler für $P_{Input}(k_{bio})$ ist $\frac{1}{1800}$. Er tritt ein, falls in allen 300 Booleschen Netze alle Elemente die gleiche Anzahl k_{bio} von Eltern haben. Für einige Stichproben wurden alle Inputelemente für alle k_{bio}-Regeln korrekt gefunden.

	l=1		l=2	
k	$P_{identified}(k)$	$P_{positive}(k)$	$P_{identified}(k)$	$P_{positive}(k)$
1	$\approx 100\%$	$\approx 99\%$	$\approx 100\%$	$\approx 98\%$
2	$\approx 100\%$	$\approx 99\%$	$\approx 100\%$	$\approx 98\%$
3	$\approx 100\%$	$\approx 99\%$	$\approx 100\%$	$\approx 99\%$
4	$\approx 100\%$	$\approx 99\%$	$\approx 100\%$	$\approx 99\%$
5	$\approx 90\%$	$\approx 100\%$	$\approx 99\%$	$\approx 99\%$

Tabelle 5.2: $P_{identified}(k)$ und $P_{positive}(k)$, falls Vorwissen berücksichtigt wurde. Die angegebenen Werte wurden bei einem entsprechend großen M erhalten.

notwendigen Trainingsdaten. Zudem werden nun auch die Inputelemente für Boolesche Funktionen mit $k = 5$ richtig identifiziert. Die prior-Wahrscheinlichkeiten, sind als zusätzliche Beobachtungen für die jeweiligen Hypothesen zu verstehen. Aus den Simulationsstudien läßt sich schließen, dass ihre Wirkung bei einem bekannten Anfangszustand viel geringer als bei zwei bekannten Anfangszuständen ausfällt. Außerdem ist zu beobachten, dass Experimente mit zwei bekannten Anfangszuständen gegenüber Experimenten mit einem bekannten Anfangszustand kein besseres Ergebnis liefern, falls kein Vorwissen vorhanden ist. Der Gewinn an Information, den man durch die Kenntnis von zwei Anfangszuständen gegenüber der Kenntnis eines Anfangszustandes erhält, ist bei den Booleschen Funktionen der Hypothese nicht sehr groß. Die Kombination mit Vorwissen bringt jedoch eine wesentliche Verbesserung. Abbildung 5.2 enthält den Verlauf der Wahrscheinlichkeiten, die regulierenden Gene für $GATA-1$, $GATA-2$, SCL und $PU.1$ falsch zu bestimmen, wenn Vorwissen berücksichtigt wird. Für $GATA-1$ und $GATA-2$ gibt es nach Abbildung 5.1 maximal nur drei Input-Gene. Daher werden ihre Eltern schon bei wenigen Trainingsdaten korrekt gefunden. $PU.1$ und SCL können dagegen fünf Input-Gene haben. Folglich werden ihre regulierenden Gene im Mittel erst bei einem viel größeren Stichprobenumfang identifiziert.
Das eigentliche Ziel war nicht die Identifizierung der originalen Struktur, sondern des wahrscheinlichsten Modells gegeben der Daten. Unter Umständen kann schon eine schlechtere Struktur mit guten Schätzungen für die Parameter die Daten D sehr gut erklären. Der *Hamming-Abstand* definiert den Unterschied zwischen der gelernten und der originalen Modellstruktur.

Definition 5.3 (Hamming-Abstand) *Der Hamming-Abstand gibt an, in wie vielen Kanten sich die originale und die gelernte Modellstruktur unterscheiden. Ein Hamming-Abstand von* 1 *besagt, dass die gelernte Struktur entweder eine Kante mehr oder weniger als die originale Struktur besitzt.*

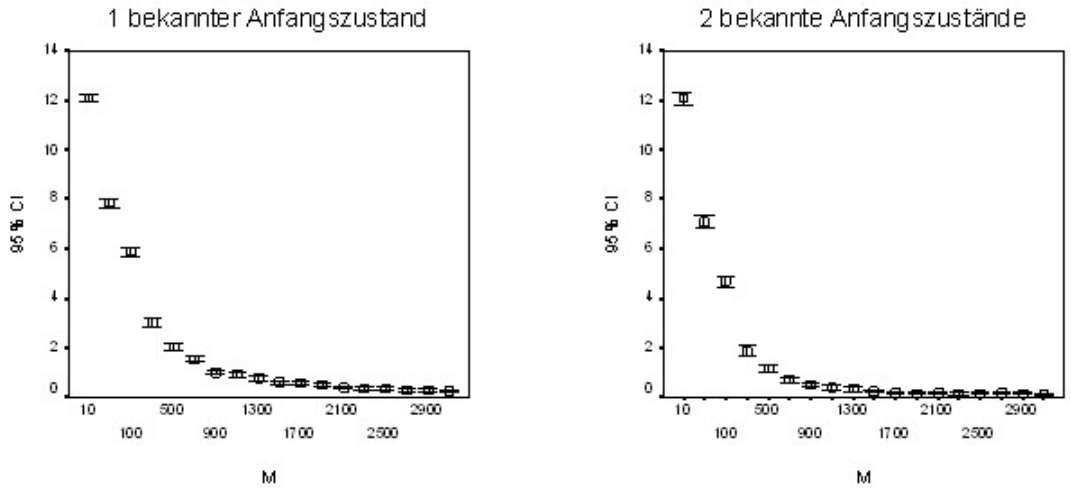


Abbildung 5.5: Reverse Engineering mit Vorwissen. Es sind die Hamming-Abstände für $l = 1$ und $l = 2$ zu sehen.

Abbildung 5.5 zeigt die 95% Konfidenzintervalle für die Mittelwerte der Hamming-Abstände, als Vorwissen berücksichtigt wurde. Zu erkennen ist, dass der Unterschied in der Struktur schon bei $M = 500$ jeweils nur zwei bis drei Kanten, für $l = 1$, und ein bis zwei Kanten, für $l = 2$, entspricht. Dieser Unterschied ist durch einige falsch positive Eltern für Boolesche Regeln bis zu vier Inputelemente und durch nicht identifizierbare Eltern für fünf Inputelemente erklärt (s. Tabelle 5.2). Bei einem realistischen Stichprobenumfang enthält die gelernte Struktur schon sehr genau die korrekten Abhängigkeiten, wenn $l = 2$.
Abbildung 5.6 zeigt die relative Entropie, falls für die gelernte Struktur die Parameter anhand von EM geschätzt wurden. Wenn für ein Element keine Eltern identifizierbar waren, bedeutet dies, dass seine Inputelemente nicht unter den bekannten Elementen sind und es daher bisher unbekannte Inputelemente gibt. Damit die Parameter für das DBN geschätzt werden können, müssen sie durch eine *Hidden Variable* in der Übergangsstruktur modelliert werden. Die Hidden Variable tritt in beiden Zeitscheiben auf, hat sich selbst als Inputelement und ihre Zuweisungen sind zu keinem Zeitpunkt bekannt.
Die relative Entropie erreicht ihren Konvergenzwert bei einem Stichprobenumfang von 500 Zustandsübergängen. Wenn zwei Gene manipulierbar sind, unterscheidet sich die gelernte Struktur ab dieser Stichprobengröße in ein bis zwei Kanten mit der originalen Struktur (vgl. Ab.. 5.4). Diese Strukturen und ihre geschätzten Parameter beschreiben die Abhängigkeiten in der Domäne gut. Bei einer Erhöhung des Stichprobenumfangs werden bessere Modellstrukturen gelernt, aber die gelern-

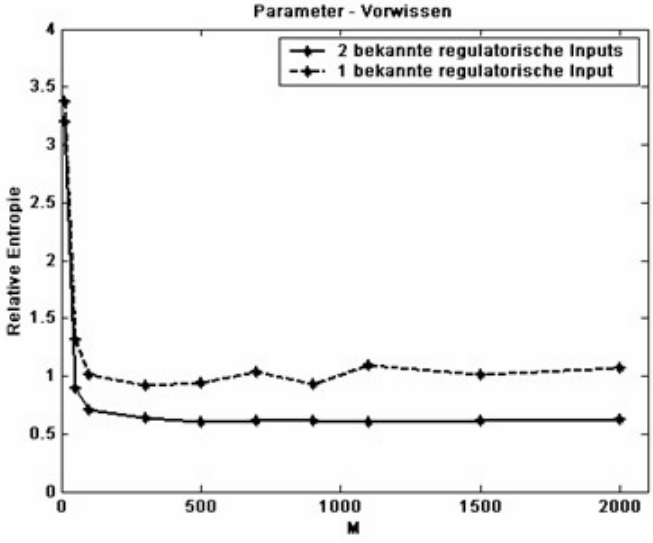


Abbildung 5.6: Reverse Engineering mit Vorwissen und Lernen der Parameter mit EM. Es ist die mittlere relative Entropie von 150 Netzen abgebildet. Die Modellstruktur wurde mit Vorwissen und partiellem Lernen der Struktur rekonstruiert. Danach wurde auf Basis der gelernten Struktur versucht, optimale Parameter mit EM zu lernen (vgl. Abschnitt 4.5). Die Zustandsübergangstabellen für das Lernen der Struktur ergaben sich aus den Kombinationen von manipulierbaren Inputelementen (vgl. Tab. 4.3). Aus jeder dieser Kombinationen wurden jeweils soviel Zustandsübergangspaare ausgewählt, so dass sich eine Zustandsübergangstabelle mit M Trainingsvektoren ergab (vgl. Tab. 4.2). Die relative Entropie wurde mit 1000 neuen Trainingsvektoren, die vom dem originalen Modell erzeugt wurden, berechnet.

ten Schätzer für die Parameter führen zu keiner besseren Erklärung der originalen Daten. Deshalb wird das bestmögliche Modell, was aus Trainingsdaten nach Definition 1.4 rekonstruiert werden kann, wenn zwei Gene gemeinsam festlegbar sind, schon bei $M = 500$ erhalten.
Wenn nur ein Gen manipulierbar ist, sind die originalen Daten nicht so gut erklärbar, denn die gelernte Modellstruktur unterscheidet sich stärker von der originalen Struktur. Außerdem ist es besonders schwierig Parameter zu schätzen, wenn nur ein Gen manipuliert wird (vgl. Abschnitt 4.5).

5.3 Dynamik

Es wird angenommen, dass sich die Zellen zum Zeitpunkt der genetischen Manipulation in einem Attraktor befinden. Aussagen über die zu erwartende Anzahl der Attraktoren, Attraktorzustände und durch 1-Bit Störungen zu erreichende At-

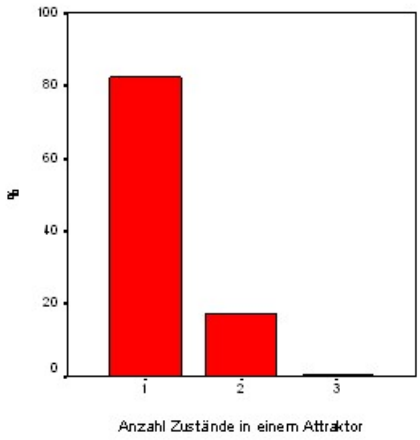


Abbildung 5.7: Anzahl der Zustände in einem Attraktor bei Booleschen Netzen, die ähnlich (vgl. Def. 5.1) zu der Hypothese sind.

traktoren, sind wichtig, um eine erfolgreiche Anwendung eines Reverse Engineering Algorithmus abzuschätzen.
Es wurden 1000 Boolesche Netze zufällig erzeugt, die ähnliche Strukturen und Regeln zu der Hypothese in Abbildung 5.1 hatten. Zuerst wurden für jedes Boolesche Netz alle Attraktoren und die Anzahl der Attraktorzustände ermittelt (Abb. 5.7). Demnach konvergieren die Zellzustände vorwiegend in Punktattraktoren. Es treten auch einige zyklische Attraktoren auf (ca. 20%). Sie besitzen meist zwei Zustände und nur in seltenen Fällen drei. Diese Tatsache motiviert eine vollständige Expressionsanalyse der Zellpopulation, die Auskunft über die Verteilung der Expressionszustände eines jeden Gens liefert. Es ist sehr wahrscheinlich, dass sich die Zellen in einem Punktattraktor befanden und somit auch die Zustände der Gene bekannt sind, die während eines Experimentes nicht manipuliert werden. Befinden sich die Zellen dagegen in einem zyklischen Attraktor, können wegen der wenigen Zustände im Attraktor, ein Großteil der Gene einen fixen Expressionszustand haben. Die geringe Anzahl an Attraktorzuständen hat den Vorteil, dass zum Zeitpunkt der genetischen Manipulation der Expressionszustand vieler Gene bekannt sein kann aber auch den Nachteil, dass die Zustandsübergangstabelle nur sehr wenige unterschiedliche Stichproben enthält. Denn durch die fixen Expressionszustände verringern sich die kombinatorischen Möglichkeiten für verschiedene Zellzustände. Aber nur unterschiedliche Zustandsübergangspaare enthalten die Information, die für die Identifizierung der Modellstruktur notwendig ist. Die Zellen müssen folglich durch gezielte Störungen in weitere Attraktoren ihres Genregulationsnetzwerkes überführt werden. Die Analysen in Kapitel 4.6 zeigten, dass durch 1-Bit Störungen fast die gesamten Attraktoren

	Mittelwert	Median	25% Perzentil	75% Perzentil
Attraktoren in $\mathcal{A}_\mathcal{S}$	11.7	11	8	16
Attraktorzustände in $\mathcal{A}_\mathcal{S}$	13.9	14	10	18

Tabelle 5.3: Auswertung der Attraktoren, die von einem Attraktor ausgehend, durch eine Sequenz von Störungen erreicht wurden. Sie bilden die Menge $\mathcal{A}_S$ (vgl. Def. 4.8) und es interessiert die Anzahl der Attraktoren in dieser Menge. Sie sind das Ergebnis von numerischen Analysen. Es wurden 1000 Netze simuliert.

eines Systems erreicht werden.
Die gleichen Analysen wurden für Boolesche Netze durchgeführt, die das vermutete Genregulationsnetzwerk in multipotenten hämatopoetischen Stammzellen modellieren. Aus ihnen geht hervor, dass es überdurchschnittlich viele Attraktoren mit einigen wenigen Attraktorzuständen in diesen Netzen gibt. Nach [24] würde man nur $\sqrt{N} = \sqrt{6} \approx 2.4$ Attraktoren erwarten. Dieses besondere dynamische Verhalten hat den Vorteil, dass überdurchschnittlich viele verschiedene Zellzustände für die biologischen Experimente verfügbar sind (vgl. Abschnitt 4.6).
Ein Experiment beinhaltet die Manipulation des Genexpressionszustandes von l Genen, um den globalen Anfangszustand teilweise festzulegen und die vollständige Charakterisierung des globalen Folgezustandes. Wenn der Zustand nur eines Elementes manipuliert wird und jedes Bit jedes Zellzustandes in $\mathcal{A}_S$ einmal auf 0 und einmal auf 1 gesetzt wird, können im Mittel 44 unterschiedliche Zellzustände erzeugt werden. Für reale Anwendungen erfordert dies mehrere Zellpopulationen, deren Zellen Expressionszustände aus $\mathcal{A}_S$ besitzen. Damit sind nur 69% der 2^6 möglichen Zellzustände für den Reverse Engineering Algorithmus verfügbar. Die Simulationsstudien ergaben bei ca. 500 Zustandsübergängen das wahrscheinlichste Modell, welches aus Trainingsdaten nach Definition 1.4 lernbar ist. Die initialen Zellzustände wurden gleichwahrscheinlich aus allen möglichen globalen Zuständen gezogen. Bei 500 Ziehungen kommen alle 64 Zellzustände zu 98% mindestens einmal vor[4]. Dies läßt vermuten, dass 44 unterschiedliche globale Anfangszustände zu wenig sind. Falls dagegen der Zustand von zwei Elementen gleichzeitig festlegbar ist, werden im Mittel 60 verschiedene Zellzustände erreicht und ein Großteil der möglichen globalen Anfangszustände sind für den Reverse Engineering Algorithmus verfügbar. Das ergibt sich aus den größeren kombinatorischen Möglichkeiten, Manipulationen anzuwenden. Demzufolge besteht die Möglichkeit, ein gutes Modell für das Genregulationsnetzwerk von multipotenten Blutstammzellen zu lernen, wenn zwei Gene gleichzeitig manipuliert werden.

[4]Dies ergibt sich aus der Wahrscheinlichkeit $P(z)$, jeder Zellzustand wird mindestens einmal aus der Population nach z Ziehungen gezogen, die in Abschnitt 4.6 eingeführt wurde.

l	Mittelwert	Median	25% Perzentil	75% Perzentil
1	43.9	47	36	52
2	60.4	64	58	64

Tabelle 5.4: Anzahl der unterschiedlichen Anfangszustände, die durch Manipulationen der Zustände in $\mathcal{A}_S$ (vgl. Def. 4.8) erreicht wurden. Sie sind das Ergebnis von numerischen Analysen. Der Mittelwert und die Perzentile sind aufgeführt. Es wurden insgesamt 1000 Boolesche Netze betrachtet.

5.4 Zusammenfassung

Die Analyse eines konkreten biologischen Systems ergab, dass qualitatives Vorwissen den notwendigen Stichprobenumfang verringert. Das Vorwissen kann man aus Literaturrecherchen erhalten, indem Ergebnisse von unabhängigen Experimenten zusammengetragen werden.
Das wahrscheinlichste Modell, welches aus Trainingsdaten nach Definition 1.4 gelernt werden kann, wurde schon bei einem Stichprobenumfang von 500 Zustandsübergangspaaren erhalten. Dies gilt jeweils für $l = 1$ und $l = 2$.
Es ist möglich, durch Störungen eine Anzahl von sich unterscheidenden globalen Anfangszuständen zu erhalten, die der Anforderung des Reverse Engineering Algorithmus genügt. Voraussetzung dafür ist jedoch, dass der Zustand von zwei Genen gemeinsam und gezielt festlegbar ist.
Für das Genregulationsnetzwerk in multipotenten Blutstammzellen ist ein gutes Modell lernbar, wenn von mindestens zwei Genen die Anfangszustände gemeinsam festgelegt werden können.

Kapitel 6

Diskussion

6.1 Zusammenfassung der Ergebnisse

Das Ziel dieser Arbeit war es, eine Reverse Engineering Strategie zu entwickeln, die aus unvollständigen Genexpressionsdaten nach Def. 1.4 ein Modell für ein genetisches Netzwerk rekonstruiert. Diese Art von Genexpressionsdaten resultieren aus Expressionsanalysen einzelner Zellen. Die theoretischen Untersuchungen in dieser Arbeit erfolgten mit künstlich erzeugten Trainingsdaten, indem genetische Netzwerke durch Boolesche Netze modelliert wurden. Das gelernte Modell sollte das wahrscheinlichste Dynamische Bayessche Netz gegeben der Daten sein. Es wurden drei verschiedene Methoden analysiert.

Die erste Strategie, REVEAL, lernt Boolesche Netze und hat keinen Ansatz für fehlende Beobachtungen. Die Inputelemente und die Booleschen Regeln für Elemente, deren Anzahl an Eltern kleiner oder gleich der manipulierbaren Anfangszustände ist, können korrekt rekonstruiert werden. Dafür sind mehrere Experimente notwendig, die sich jeweils in den manipulierten Anfangszuständen unterscheiden. Denn sobald in mindestens einem Experiment alle regulierenden Gene eines Gens manipuliert werden können, läßt sich ihr gemeinsamer Einfluss bestimmen. REVEAL erkennt die Inputelemente nur, falls ihre Mutual Information mit dem Outputelement maximal wird, also gleich der Entropie des Outputelementes ist. Es werden entweder alle Eltern für ein Outputelement oder die leere Menge zurückgegeben. Die Identifizierung von Teilmengen ist nicht möglich. Unter Umständen erklären auch schon weniger Inputelemente den Zustand des Outputelementes fast vollständig. Ihre Identifizierung wäre ein großer Erkenntnisgewinn. Außerdem haben gestörte Daten keine maximale Mutual Information. Deshalb ist REVEAL in seiner originalen Konfiguration für reale Anwendungen ungeeignet.

Eine zweite Strategie, Strukturelle Erwartungswert Maximierung, basiert auf

allgemeineren Prinzipien. Das wahrscheinlichste Modell, gegeben der Daten, wird durch lokale Strukturänderungen ausgehend von einem initialen Modell, gelernt. Für die Verarbeitung von fehlenden Beobachtungen existieren gut fundierte Ansätze. Es wurde festgestellt, dass der Umfang an fehlenden Daten zu groß ist und es daher unmöglich ist, das korrekte Modell mit hinreichend hoher Wahrscheinlichkeit zu rekonstruieren. Beobachtungen für Kombinationen, die mehr Elemente enthalten, als in einem Experiment manipulierbar sind, werden in der Zustandsübergangstabelle nicht gesehen. Die Wahrscheinlichkeit für ihr Auftreten läßt sich demzufolge nicht korrekt bestimmen.

Die dritte Strategie, partielles Lernen der Struktur, umgeht aus diesem Grunde, die fehlenden Daten. Jedes Experiment wird unabhängig von den anderen Experimenten analysiert. Die Vereinigung der getrennt voneinander identifizierten regulierenden Gene bilden die Eltern für ein Zielgen. Mit dieser Methode werden aber nur die Gene identifiziert, deren Einfluss unabhängig von den anderen regulierenden Genen groß genug ist.
Die Analysen ergaben, dass die Manipulation eines einzelnen Gens nicht ausreicht. Die Inputelemente einer Booleschen Funktion übertragen ungleichmäßig Information auf den Output. Nicht alle Funktionen besitzen Inputelemente, deren Informationsübertragung unabhängig von den restlichen signifikant ist. Aber nur solche Inputelemente werden mit dieser Strategie erkannt. Bei einem Großteil der Booleschen Funktionen mit $k \leq 4$ übertragen maximal zwei Inputelemente signifikante Information auf den Output. Wenn die Anfangszustände von mindestens zwei Genen gleichzeitig manipulierbar sind, kann ein gutes Modell rekonstruiert werden. Ist dagegen nur der Anfangszustand von einem Gen festlegbar, werden mit einer hohen Wahrscheinlichkeit die Inputelemente mit größtem Einfluss identifiziert.
Ist eine gute Modellstruktur verfügbar, werden anhand der Methode Erwartungswert Maximierung die Modellparameter geschätzt. Wenn die Anfangszustände zweier Gene festlegbar sind, lassen sich gute Parameter lernen.
Simulationsstudien gaben Aufschluß über die benötigte Menge an Trainingsdaten. Da die Inputelemente nur in getrennten Informationsquellen untersucht werden können, ist ein sehr hoher Stichprobenumfang notwendig, um Korrelationen zu erkennen. Es wird davon ausgegangen, dass sich die Zellen zum Zeitpunkt der Manipulation in Attraktoren aufhalten. Die Zellen nehmen daher nur einen Bruchteil der möglichen Expressionszustände an. Aber nur sich voneinander unterscheidende Zustandsübergangspaare liefern Erkenntnisse über die Korrelationen. Indem die insilico Zellen in verschiedene Attraktoren gezwungen wurden, wurde nachgewiesen, dass die Anzahl an verschiedenen Attraktorzuständen ausreichend erhöht werden kann. Sie fungieren als Ausgangszustände für die gezielten Manipulationen. Stehen genügend Zellen mit diesen Ausgangszuständen zur Verfügung, so dass jede Kombination mit zwei Genen einmal auf ihre vier möglichen Expressionszustände festgelegt

werden kann, enthalten die Zustandsübergangstabellen ausreichend sich unterscheidende Anfangszustände. Falls nur ein Gen gezielt manipulierbar ist, enthalten sie genügend unterschiedliche Anfangszustände, um die wichtigsten regulierenden Gene zu finden.
Da nicht alle Anfangszustände gleichzeitig bekannt sind, läßt sich mit der letzten Strategie nur im begrenzten Umfang entscheiden, ob die gefundenen Input-Gene das Zielgen vollständig beschreiben oder ob es noch weitere unbekannte regulierende Gene gibt.

Der letzte Teil der Arbeit untersuchte, ob es möglich ist, das Genregulationsnetzwerk von multipotenten Blutstammzellen zu rekonstruieren. Die Modellstruktur wurde mit der dritten Reverse Engineering Strategie gelernt und es wurde qualitatives Vorwissen integriert.
Es konnte gezeigt werden, dass ein gutes Modell rekonstruierbar ist, wenn zwei Gene gemeinsam festgelegt werden. Die gelernte Modellstruktur unterscheidet sich im Mittel nur in einer Kante von der originalen Struktur. Gemeinsam mit den geschätzten Parametern erklärt es die originalen Daten gut.
Wenn die Gene nicht gemeinsam manipulierbar sind, wird ebenfalls eine gute Modellstruktur gelernt. Aber es ist zu erwarten, dass die dafür notwendige Zahl an sich unterscheidenden Anfangszuständen nicht erreicht werden kann.
Qualitatives Vorwissen bewirkt eine wesentliche Verringerung des Stichprobenumfangs, wenn zwei Gene gleichzeitig festlegbar sind.

Folgendes Fazit läßt sich aus dieser Arbeit ziehen. Die Konstruktion von Modellen für genetische Netzwerke aus Expressionsdaten, die von Einzelzellanalysen gewonnen wurden, ist möglich, wenn der Anfangszustand von zwei Genen gemeinsam festgelegt werden kann. Sind die Anfangszustände hingegen nicht gemeinsam manipulierbar, können nur die wichtigsten regulierenden Gene identifiziert werden.

6.2 Limitationen

Da Boolesche Netze die künstlichen Trainingsdaten erzeugten, wurden einige Annahmen gemacht, die für genetische Netzwerke nicht zutreffen. Die erste Annahme betrifft die Zeit zwischen der Manipulation und der Charakterisierung des Folgezustandes. Sie beträgt einen diskreten Zeitschritt, was jedoch nicht der Realität entspricht. Es ist wahrscheinlich, dass die Zellen zwischen den Expressionsanalysen ihr Genregulationsnetzwerk mehrmals aktualisierten und daher nicht direkte Einflüsse gemessen werden. Außerdem kann die Aktivierung und Deaktivierung von Genen in unterschiedlichen Zeitskalen geschehen. Diese Punkte müssen in der Modellierung berücksichtigt werden, indem *Hidden Variablen* die nicht wahrgenommenen Zeitschritte darstellen.

Eine zweite Annahme ist, dass die Konzentrationen der Genprodukte nur diskrete Werte annehmen. In der Realität trifft man jedoch auf stetige Werte. Es gibt die Möglichkeit, die kontinuierlichen Zustände zu quantifizieren, wodurch das Modell aber an Genauigkeit verliert, auch wenn in sehr kleinen Intervallen quantifiziert wird. Das zugehörige diskrete Modell besitzt in diesem Fall sehr viele Parameter. Daher ist es angebracht, in Dynamische Bayessche Netze stetige Zufallsvariablen zu integrieren.
Die letzte Annahme betrifft die Wahl des Modells. Es sollte das beste Modell, gegeben der Daten, gelernt werden. Unter Umständen ist es unmöglich das beste Modell zu finden oder es können mehrere Modelle einen sehr ähnlichen Score haben, also gleich wahrscheinlich gegeben der Daten sein. Demzufolge ist es angebracht, mehrere Modelle als Ergebnis zuzulassen. Das gewichtete Mittel $\sum_G P(X_{n+1}|G,D)P(G|D)$ definiert dann die Wahrscheinlichkeit der nächsten Beobachtung $P(X_{n+1}|D)$ [25]. Diese Herangehensweise wird *Bayesian Averaging* genannt und ist sehr eng mit Bayesschen Lernmethoden verknüpft.

6.3 Ausblick

Eine Erhöhung der festlegbaren Expressionslevel führt zu mehr bekannten Anfangszuständen und bringt schon bei kleineren Stichprobenumfang ein gutes Ergebnis, auch für schwache Korrelationen. Es erhöhen sich die kombinatorischen Möglichkeiten die Anfangszustände festzulegen und deshalb auch die verfügbaren sich unterscheidende Zellzustände. Es ist jedoch unwahrscheinlich, dass in einem biologischen System mit seinen vielen Einflussfaktoren die Expressionslevel vieler Gene gleichzeitig gezielt kontrollierbar sind.
Wie im letzten Teil der Arbeit gezeigt, kann Vorwissen zu einer wesentlichen Verringerung des notwendigen Stichprobenumfangs führen. Ein initiales Modell ist durch vorhandenes Expertenwissen und durch die Identifizierung der Gene mit größten Einfluss anhand der dritten Reverse Engineering Strategie aufstellbar. Es beinhaltet genaues Wissen über ein Teilnetzwerk und Vermutungen über die restlichen Abhängigkeiten, die über Likelihood Ratios in den Reverse Engineering Algorithmus berücksichtigt werden.
Steht dagegen Vorwissen in Form von äquivalenten Stichprobengrößen zur Verfügung bieten sich Bayessche Lernmethoden an. Sie ermöglichen die Integration von qualitativen und quantitativen Vorwissen. Beispiele sind der *BDe* Score [21] für die Modellstruktur und die *Maximum A Posterior* Schätzer für Modellparameter [20].
Bei einer Expressionsanalyse auf Basis von Populationsdaten ist das Problem der unvollständig bekannten Anfangszustände nicht gegeben. Da die gemessene Expressionsstärke für ein Gen jedoch das Mittel über die Expression in allen Zellen ist,

kann das dynamische Verhalten nicht mehr zuverlässig charakterisiert werden, falls sich die Genregulationsnetzwerke in den Zellen asynchron aktualisieren.
Modelle von genetischen Netzwerken sollen den Biologen ein Werkzeug zur Verfügung stellen mit dem sie u.a. Vorhersagen über das dynamische Verhalten bei gezielten Manipulationen des Expressionszustandes der Zellen, ermöglichen. Die gleichen Vorhersagen sind möglich, wenn das dynamische Verhalten auf einer höheren Ebene modelliert wird, z.B. mit einem Modell über die Beziehungen der Attraktoren bei gezielten Störungen. Kann man z.B. mit einer gewissen Wahrscheinlichkeit voraussagen, dass Zellen, die sich in Attraktor $\mathcal{A}$ befinden, bei Anwendung einer bestimmten Störung in Attraktor $\mathcal{B}$ springen und die Attraktoren Zelllinien zugeordnet sind, läßt sich der mögliche Differenzierungspfad von multipotenten hämatopoetischen Stammzellen modellieren.
Letztendlich bieten sich Dynamische Bayessche Netze als Modelle für Genregulationsnetzwerke an. Denn sie definieren die existierende Unsicherheit über die Relationen. Aufgrund der vielen unbekannten Einflussfaktoren ist es angebracht, die Beziehungen mit Wahrscheinlichkeiten zu bewerten. Anstatt zu modellieren, dass zwei Gene korreliert sind, sollte besser festgehalten werden, wie sicher diese Korrelation ist. Für realistischere Modelle, ist es angebracht stetige Zufallsvariablen zu verwenden.

Literaturverzeichnis

[1] Akutsu, T., Kuhara, S.. Identification of Gene Regulatory Networks by strategic gene disruptions and gene overexpressions. *Proceedings of the Pacific Symposium on Biocomputing, 1998.*

[2] Akutsu, T., Miyano, S., Kuhara, S.. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *Proceedings of the Pacific Symposium on Biocomputing, 1999.*

[3] Akutsu, T., Miyano, S., Kuhara, S.. Algorithms for inferring qualitative models of biological networks. *Proceedings of the Pacific Symposium on Biocomputing, 2000.*

[4] Brady, G., et al.. Analysis of gene expression in a complex differentiation hierarchy by global amplification of cDNA from single cells. *Current Biology 5: 909-922, 1995.*

[5] Castillo, E., Gutièrrez, J., Hadi, A.S.. Expert Systems and Probabilistic Network Models. *Monographs in Computer Science, Springer-Verlag, 1997.*

[6] Conant, R. C.. Extended Dependency Analysis Of Large Systems Part I: Dynamic Analysis. *International Journal of General Systems, 14:97-123, 1988.*

[7] Cross, M.A., Enver, T.. The lineage commitment of haemopoietic progenitor cells. *Current Opinion in Genetics and Development, 7:609-613, 1997.*

[8] Cross, M.A., Dradso, D., Löffler, M.. Mapping the Transcriptional Networks controlling haemopoietic commitment and differentiation. *Antrag auf Gewährung einer Sachbeihilfe, 2002.*

[9] D'Haeseler, P., Liang, S., Somogyi, R.. Genetic Network Inference: From coexpression clustering to reverse engineering. *Bioinformatics, 16:707-726, 2000.*

[10] Durbin, R., Eddy, S., Krogh, A., Mitchison, G.. Biological sequence analysis - Probabilistic models of proteins and nucleic acids. *Cambridge University Press, 1998.*

[11] Enver, T., Heyworth, C.M., Dexter, T.M.. Do stem cells play dice? *Blood 92, 348-51; discussion 352, 1998.*

[12] Friedman, N.. Learning belief networks in the presence of missing values and hidden variables. *Proc. of the 14th International Conference on Machine Learning, 125-133, 1997.*

[13] Friedman, N., Murphy, K., Russell, S.. Learning the Structure of Dynamic Probabilistic Networks. *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98), 139-147, 1998.*

[14] Friedman, N., et al.. Using Bayesian Networks to Analyze Expression Data. *RECOMB, 127-135, 2000.*

[15] Friedman, N.. The Bayesian Structural EM Algorithm. *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98), 129-138, 1998.*

[16] Friedman, N.. LibB for Windows and Linux Programs. *http://www.cs.huji.ac.il/labs/compbio/LibB*

[17] Gilbert, S.F.. Developmental Biology Fifth Edition. *Sinauer Associates, Inc. Publishers, Sunderland, Massachusetts, 1997.*

[18] Glass, L., Kauffman, S.A.. Co-operative components, spatial localization and oscillatory cellular dynamics. *Journal Theoretical Biology 34, 219-237, 1972.*

[19] Hannon, G.J.. RNA interference. *Nature 2002 Jul 11; 418 (6894): 244-51.*

[20] Heckerman, D.. A Tutorial on Learning with Bayesian Networks. *Technical Report, 1995.*

[21] Heckerman, D., Geiger, D., Chickering, D.. Learning Bayesian Networks: The combination of Knowledge and Statistical Data. *Machine Learning, 20, 197-243, 1995.*

[22] Jong, H.D.. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology 9:67-103, 2002.*

[23] Ideker, T.E., Thorsson, V., Karp, R.M.. Discovery of regulatory interactions through perturbation: inference and experimental design. *Proceedings of the Pacific Symposium on Biocomputing, 2000.*

[24] Kaufman, A.S.. The Origins of Order: Self organization and selection in evolution. *Oxford University Press, Oxford, 1993.*

[25] Koller, D.. Lecture Notes and Readings. *http://robotics.stanford.edu/∼koller*

[26] Lauritzen, S.L.. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis, 19:191-201, 1995.*

[27] Li, Y. et al. (2001). Regulation of the PU.1 gene by distal elements. *BLOOD, 15 November 2001, Vol. 98, Number 10.*

[28] Liang, S., Fuhrman, S., Somogyi, R.. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Proceedings of the Pacific Symposium on Biocomputing, 1998.*

[29] Loeffler, M., Roeder, I.. Tissue Stem Cells: Definition, Plasticity, Heterogenity, Self-Organization and Models - A Conceptual Approach. *Cells Tissues Organs, 171:8-26, 2002.*

[30] McIvor, M., Hein, S., Fiegler, H., Schroeder, T., Stocking, C., Just, U. and Cross, M.A.. The transient expression of PU.1 commits multipotent progenitors to a myeloid fate, while continued expression favours macrophage over granulocyte differentiation. *Experimental Hematology, Jan 2003.*

[31] Merika, M., Thanos, D.. Enhanceosomes. *Current Opinion in Genetics and Development, 11:205-208, 2001.*

[32] Mestl., T., Plahte, E. and Omholt, S.W.. *A mathematical framework for describing and analysing gene regulatory networks.* Journal Theoretical Biology 176, 291-300, 1995.

[33] Metcalf, D.. Lineage commitment and maturation in hematopoietic cells: The case of extrinsic regulation. *Blood 92, 345-7, discussion 352, 1998.*

[34] Miller, G.. Note on the bias of information estimates. *H. Quastler, editor, Information theory in psychology. The Free Press, 1955.*

[35] Murphy, K., Mian, S.. Modelling Gene Expression data using Dynamic Bayesian Networks *Technical Report, 1999.*

[36] Murphy, K.. The Bayes Net Toolbox for Matlab (BNT). *http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html.*

[37] Ong, I.M., Glasner, J.D., Page, D.. Modelling regulatory pathways in E.coli from time series expression profiles. *Bioinformatics, 18:241S-248S, 2002.*

[38] Preisler, H.D., Kauffman, S.. A proposal regarding the mechanism which underlies lineage choice during hematopoietic differentiation. *Leukemia Research 23,685-694, 1999.*

[39] Russell, S., Norvig, P.. Artificial Intelligence: A modern approach. *Prentice Hall, New Jersey, 1995.*

[40] Sachs, L.. Angewandte Statistik: Anwendung statistischer Methoden. *Springer-Verlag Berlin Heidelberg New York, 9. Auflage, 1999.*

[41] Shannon, C.. A mathematical theory of communication. *Bell System Technical Journal, 27:379-423, 623-656, 1948.*

[42] Somogyi, R., Fuhrman, S.. Distributivity, a general information theoretic network measure, or why the whole is more than the sum of its parts. *Proc. International Workshop on Information Processing in Cells and Tissues (IPCAT), 1997.*

[43] Szallasi, Z.. Tutorial: Genetic Network analysis - from the bench to computers and back. 2^{nd} *International Conference on Systems Biology, 2001.*

[44] Weaver, D.C., Workman, C.T., and Stormo, G.D.. Modeling regulatory networks with weight matrices. *Proceedings of the Pacific Symposium on Biocomputing, 1999.*

[45] Wolpert, L.. Principles of Development *Current Biology Ltd. Oxford University Press, 1998.*

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Äquivalenz Likelihood Score und Mutual Information

Es wird von dem normalisierten Log Likelihood Score ausgegangen, der wie folgt umgeschrieben werden kann [35]:

$$
\begin{aligned}
&\log L(G : D|\hat{\Theta}_G) \\
&= \frac{1}{M}\sum_{i=1}^{N}\sum_{j_i=1}^{q_i}\sum_{k_i=1}^{r_i} N_{ij_ik_i}\log P(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i) \\
&\overset{(1)}{\approx} \frac{1}{M}\sum_{i=1}^{N}\sum_{k_ij_i} M\hat{P}(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i)\log P(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i) \\
&\overset{(2)}{=} \sum_{i=1}^{N}\sum_{k_ij_i}\Big[\hat{P}(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i)\hat{P}(\boldsymbol{Pa}(X_i) = j_i) * \log P(X_i = k_i|\boldsymbol{Pa}(X_i) = j_i)\Big] \\
&\overset{(3)}{\approx} \sum_{i=1}^{N} -H(X_i|\boldsymbol{Pa}(X_i)) \\
&\overset{(4)}{=} \sum_{i=1}^{N} MI(X_i, \boldsymbol{Pa}(X_i)) - H(X_i)
\end{aligned}
$$

Denn die empirische Schätzung einer Wahrscheinlichkeit ist:

$$\hat{P}(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i) = \frac{N_{ij_ik_i}}{N_{ij_i}} * \frac{N_{ij_i}}{M} = \frac{N_{ij_ik_i}}{M}.$$

Daraus folgt

$$(1) \quad N_{ij_ik_i} = M * \hat{P}(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i).$$

Es gilt:

$$(2) \quad P(X_i = k_i, \boldsymbol{Pa}(X_i) = j_i) = P(X_i = k_i | \boldsymbol{Pa}(X_i) = j_i) P(\boldsymbol{Pa}(X_i) = j_i).$$

Außerdem ist

$$(3) \quad H(X|Y) = -\sum_{X,Y} P(X|Y) P(Y) \log P(X|Y)$$

und

$$(4) \quad MI(X,Y) = H(X) - H(X|Y).$$

Anhang B

Ziehen verschiedener Expressionszustände

Es sind S unterschiedliche Zellzustände in einer Zellpopulation gegeben. Es wird danach gefragt, wie groß die Wahrscheinlichkeit ist, nach z Ziehungen jeden der unterschiedlichen Zellzustände mindestens einmal gezogen zu haben. Sie wird über das Gegenereignis $\bar{A}$, keinen der Zustände jemals gezogen zu haben, berechnet. Es sei A_k das Ereignis, dass der Zustand k in z Zügen nie gezogen wird, dann ist:

$$P(A_k) = (1 - \frac{1}{S})^z. \tag{B.1}$$

Analog dazu hat das Ereignis $A_{k_1} \cap A_{k_2} \cap \ldots \cap A_{k_i}$ die Zustände $k_1, k_2, \ldots, k_i$ nie zu ziehen, die Wahrscheinlichkeit:

$$P(A_{k_1} \cap A_{k_2} \ldots \cap A_{k_i}) = (1 - \frac{i}{S})^z. \tag{B.2}$$

Das Ereignis, keinen der Zustände in z Ziehungen zu sehen, ist die Vereinigung über die Ereignisse, den Zustand 1 nie zu ziehen, den Zustand 2 nie zu ziehen u.s.w., also:

$$\begin{aligned}
P(\bar{A}) &= P(\bigcup_{i=1}^{S} A_k) \\
&\overset{(Siebformel)}{=} \sum_{1 \le i \le S} P(A_k) - \sum_{1 \le i < j \le S} P(A_i \cap A_j) + \ldots \pm P(A_1 \cap \ldots \cap A_S) \\
&\overset{(B.2)}{=} \binom{S}{1}(1 - \frac{1}{S})^z - \binom{S}{2}(1 - \frac{2}{S})^z + \ldots \pm \binom{S}{S}(1 - \frac{S}{S})^z \\
&= \sum_{i=1}^{S} (-1)^{i-1} \binom{S}{i} (1 - \frac{i}{S})^z
\end{aligned} \tag{B.3}$$

Daraus folgt nun die Wahrscheinlichkeit für das Ereignis, jeden Zustand nach z Zügen mindestens einmal gezogen zu haben:

$$P(A) = 1 - P(\bar{A}) = \sum_{i=0}^{S} (-1)^i \binom{S}{i} (1 - \frac{i}{S})^z. \tag{B.4}$$

Anhang C

Literaturangabe für das hypothetische Netzwerk

1: *Nerlov, C., et al. (2000) Blood 95, 2543-51*

2: *Chen, H., et al. (1995) Oncogene 11, 1549-60*

3: *Bellon, T., et al. (1997) Blood 90, 1828-39*

4: *Bockamp, E.O., et al. (1998) J. Biol. Chem. 273, 29032-42*

5: *Lecointe, N., et al. (1994) Oncogene 9, 2623-32*

6: *Nishimura, S., et al. (2000) Mol. Cell Biol. 20, 713-23*

7: *Anderson, K.P., et al. (2000) Blood 95, 1652-5*

8: *Vyas, P., et al. (1999) Development 126, 2799-811*

9: *Toki, T., et al. (2000() Exp. Hematol. 28, 1113-1119*

10: *Katsuoka, F., et al. (2000) Embo. J. 19, 2980-91*

11: *Barbeau, B., et al. (1999) Oncogene 18, 5535-45*

12: *Starck, J., et al. (1999) Mol. Cell Biol. 19, 121-35*

13: *Sanchez, M., et al. (1999) Development 126, 3891-904*

14: *Wang, X., et al. (1999) Blood 94, 560-71*

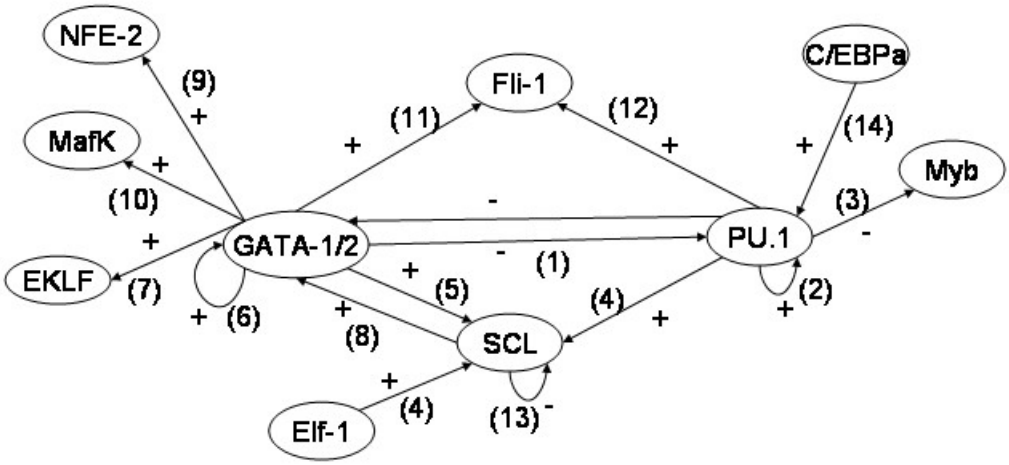


Abbildung C.1: Literaturangaben für das hypothetische Genregulationsnetzwerk in multipotenten Blutstammzellen.

Anhang D

Notation

Symbol	Erklärung
$\mathcal{A}_S$	Vgl. Def. 4.8
α, α_T	Signifikanzniveaus
B	Bayessches Netz $B(G, \mathbf{\Theta})$
BIC	Engl.: Bayesian Information Criterion
BN	Boolesches Netz
B_0	Startnetz eines DBN $B_0(G_0, \mathbf{\Theta^0})$
$B_{\rightarrow}$	Übergangsnetz eines DBN $B_{\rightarrow}(G_{\rightarrow}, \mathbf{\Theta}^{\rightarrow})$
DBN	Dynamisch Bayessches Netz
DGL	Differentialgleichungen
EM	Engl.: Expectation Maximization
F	Menge von booleschen Funktionen in G
f_i	Boolesche Funktion i
G	Graph bzw. Struktur eines Booleschen oder Dynamisch Bayesschen Netzes
$G^{(0)}$	Startstruktur eines DBN.
$G^{\rightarrow}$	Übergangsstruktur eines DBN
$H(X)$	Entropie der Zufallsvariablen X
$H(X\|Y)$	Bedingte Entropie der Zufallsvariablen X und Y
j_i	Zuweisung j für die Eltern einer Zufallsvariablen X_i
K	Maximal mögliche Anzahl von Eltern oder Inputelementen in einem BN oder DBN
k	Anzahl der Eltern eines Elementes
k_{bio}	Bezeichnet die Anzahl der Inputelemente für *Canalyzing Funktionen*
k_{can}	Bezeichnet die Anzahl der Inputelemente für Booleschen Funktionen, deren Inputelemente alle *canalyzing* sind
k_{eff}	Vgl. Def. 3.6

k_1	Vgl. Def. 4.5
$k_{1,2}$	Vgl. Def. 4.6
k_i	Zuweisung k für eine Zufallsvariable X_i
l	Anzahl der Gene, deren Anfangszustand gemeinsam manipuliert werden kann
LR	Abkürzung für Likelihood Ratio
M	Anzahl Beobachtungen einer Kombination von Genen mit bekannten Anfangszuständen
MLE	Engl.: Maximum Likelihood Estimation
M	Anzahl Zustandsübergangspaare, die in einer Stichprobe enthalten sind
MI	Abkürzung für *Mutual Information*
$MI(X, Y)$	Mutual Information der Zufallsvariablen X und Y.
N	Anzahl Variablen in einem Booleschen Netz oder einem Dynamisch Bayessches Netz
N_R	Anzahl Trajektorien
$N_{ij_ik_i}$	Anzahl Beobachtungen bei denen $X_i = k_i$ und $\mathbf{Pa}(X_i) = j_i$
N_{ij_i}	Anzahl Beobachtungen bei denen $\mathbf{Pa}(X_i) = j_i$, $N_{ij_i} = \sum_{k_i=1}^{r_i} N_{ij_ik_i}$
$P_{identified}(k)$	Vgl. Def. 4.2
$P_{positive}(k)$	Vgl. Def. 4.3
$P_{Input}(k)$	Vgl. Def. 4.1
$P(A)$	Wahrscheinlichkeit das Ereignis A eintritt
$P(A\|B)$	Bedingte Wahrscheinlichkeit von A gegeben B
$\mathbf{Pa}(\mathbf{X_i})$	Menge der Zufallsvariablen in einem DBN, die Eltern von X_i sind
p_i	Wahrscheinlichkeit des Ereignisses i
$\mathbf{pa}_i$	Belegungen für die Eltern der Zufallsvariablen X_i
q_i	Anzahl Belegungen für die Eltern der Zufallsvariablen X_i
REVEAL	*Reverse Engineering Algorithmus*, vgl. Abschnitt 3.2 und [28]
R_i	Bezeichnung für eine Trajektorie i
r_i	Anzahl Belegungen für die Zufallsvariable X_i
S	Zustandsraum von N booleschen Variablen; enthält 2^N Zustände
SEM	Engl.: Structural Expectation Maximization
T	Länge einer Trajektorie
δT_C	Zeit bis das Expressionsmuster einer erfolgreich gestörten Zelle vollständig charakterisiert ist, nachdem sich ihr genetisches Netzwerk einmal aktualisierte
δT_S	Zeit für die Anwendung einer Störung
$\hat{T}$	Durchschnittliche Zeit für eine Folge von Aktualisierungen des Genegulationsnetzwerkes, die wieder in einem stabilen Expressionsmuster enden
t	Zeit

τ	Zeit zwischen zwei Aktualisierungen eines Genregulationsnetzwerkes
u	Anzahl unbekannter Gene in einem Booleschen Netz
u_{max}	Maximal mögliche Anzahl von unbekannten Inputelementen für eine Boolesche Variable
V	Menge von booleschen Variablen in G
v_i	Boolesche Variable i
X, Y	Großbuchstaben bezeichnen Zufallsvariablen
x, y	Kleinbuchstaben bezeichnen Belegungen der Zufallsvariablen X, Y
X'	Outputelement in einem Booleschen Netz
$\hat{X}$	Schätzer der Größe X
$\bar{x}$	Vektor
X_i	Zufallsvariable i
$X_i[t]$	Zufallsvariable eines Dynamisch Bayesschen Netzes, die den Wert der Variablen X_i zur Zeit t bestimmt
$\mathbf{\Theta}$	Menge von Parametern $\theta_{ij_ik_i}$
$\mathbf{\Theta^{(0)}}$	Menge von Parametern $\theta^0_{ij_ik_i}$
$\mathbf{\Theta^{\rightarrow}}$	Menge von Parametern $\theta^{\rightarrow}_{ij_ik_i}$
$\theta_{ij_ik_i}$	Parameter in einem Bayesschen Netz. $\theta_{ij_ik_i} = P(X_i = k_i \vert Pa(X_i) = j_i)$
$\theta^{(0)}_{ij_ik_i}$	Parameter der Startstruktur in einem Dynamisch Bayesschen Netz
$\theta^{\rightarrow}_{ij_ik_i}$	Parameter der Übergangsstruktur in einem Dynamisch Bayesschen Netz
Z	Anzahl Zustandsübergangspaare in einer Zustandsübergangstabelle

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig 31. März 2003