Mathematical Modeling of *Arabidopsis thaliana* with Focus on Network Decomposition and Reduction

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften
im Fach Bioinformatik

vorgelegt beim Fachbereich Mathematik und Informatik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Hans Joachim Karl Nöthen
aus Bonn

Frankfurt (2014)
(D 30)

vom Fachbereich Mathematik und Informatik der

Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan:

Gutachter:

Datum der Disputation:

Dedicated to my wife, my daughter, and my godfather.

Zusammenfassung

Die Forschung auf dem Gebiet der Systembiologie gewinnt in den letzten Jahren zunehmend an Bedeutung. Sie beschäftigt sich mit dem System 'hinter' den gewonnenen biologischen Daten. Ein Teilbereich umfasst die Modellierung von Netzwerkstrukturen zur Beschreibung von verknüpften biologischen Elementen. Hierzu gehören insbesondere metabolische Netzwerke zur Analyse der Daten über die Verbindung zwischen Metaboliten wie auch Regulationsnetzwerke zur Untersuchung der Daten über die Regulation von Genexpression oder von Reaktionswegen.

Generell gibt es zwei verschiedene Herangehensweisen an die Netzwerkmodellierung. Die eine ist die quantitative Modellierung basierend auf Differentialgleichungen. Die andere ist die qualitative Modellierung, die - wenn überhaupt - mit diskreten, meist ganzzahligen Werten arbeitet und das Modell mit Hilfe linearer Gleichungssysteme beschreibt.

Im Forschungsgebiet der metabolischen Netzwerke dienen quantitative Analysen der Untersuchung des 'flux', des Stoffflusses durch ein metabolisches System. Dies geschieht entweder durch Vergleiche zu gemessenen Daten (MFA), oder durch Vorhersageversuche anhand des Modells ausgehend von einem steady-state im System (FBA). Qualitative Methoden werden meist herangezogen, um die Netzwerktopologie zu untersuchen und Unstimmigkeiten im Aufbau oder mögliche Zerlegungen in Subnetzwerke mit bestimmten Eigenschaften zu finden.

Netzwerke werden mit Hilfe von Graphen dargestellt. Für metabolische Systeme bieten sich bipartite Graphen an, deren Knotenmenge in zwei disjunkte Untermengen aufgeteilt werden kann. Verbindungen existieren nur zwischen Knoten verschiedener Untermengen. Repräsentiert die eine Knotenmenge die Menge der Metabolite, die andere die Menge der Reaktionen, und die Kanten eine Beteiligung der verbundenen Metabolite an den verbundenen Reaktionen, ergibt sich ein intuitives Modell metabolischer Vorgänge.

Petrinetze sind bipartite Graphen, deren Definition diese Repräsentation erlaubt. Verschiedene solide mathematische Analysemethoden für Petrinetze wurden seit den 1960'er Jahren beschrieben und weiterentwickelt, und bieten - wie in verschiedenen Studien gezeigt - ein wertvolles Werkzeug zur Untersuchung metabolischer Netzwerke. Ergänzt wird dies durch die Möglichkeit der Darstellung eines diskreten Metabolitverteilungszustandes unter Anwendung des Instruments der Token, mit anschließender dynamischer Simulation.

Die Modellpflanze (*Arabidopsis thaliana*) wird seit vielen Jahren in der biologischen Forschung eingesetzt. Verschiedene quantitative metabolische Modelle dieser Pflanze sind in den letzten Jahren veröffentlicht worden, qualitative Modelle allerdings sind rar.

In der vorliegenden Arbeit wird ein qualitatives metabolisches Netzwerk beschrieben, das

als Petrinetz modelliert ist und den Kernmetabolismus der Pflanze *Arabidopsis thaliana* darstellt. Anhand des Modells werden verschiedene Methoden entwickelt und getestet, die eine Validierung und biologische Interpretation dieses - zuerst für eine Analyse zu komplexen - Netzwerkes erlauben. Hierbei wird besonders die Berechnung der Transitionsinvarianten (T-Invarianten) berücksichtigt. Eine besonders wichtige Eigenschaft ist die Überdecktheit mit T-Invarianten (CTI). Ein Petrinetz ist CTI, wenn jede Transition Teil mindestens einer T-Invariante ist.

Das entwickelte Netzwerk ist manuell anhand von Informationen aus diversen Veröffentlichungen aufgebaut worden. Das Modell besteht aus 134 Metaboliten, 243 Reaktionen, wobei reversible Reaktionen doppelt zählen, und 572 Verbindungskanten. 29 Reaktionen stellen Verbindungen zur Umgebung her, 8 davon stellen Stoff bereit (IN), während 21 Reaktionen Stoff abführen (OUT). Die 4 externen Metabolite sind sowohl mit jeweils einer zuführenden als auch mit einer abführenden Reaktion verbunden, wodurch die Verbindungen zur Umwelt der internen Metabolite auf 4 zuführende und 17 abführende Reaktionen reduziert werden.

Um die Komplexität des Netzwerkes so gering wie möglich zu halten, werden verschiedene Maßnahmen ergriffen. Zum einen berücksichtigt das Modell keine Kompartimente, zum anderen sind kleine Metabolite wie Wasser und Kohlenstoffdioxid, aber auch Kofaktoren wie NAD/NADH oder ATP nicht modelliert. Frühere Studien haben gezeigt, dass Metabolite, die an vielen Reaktionen beteiligt sind, weniger stark konserviert sind als solche, die Verbindungen zwischen verschiedenen Pathways herstellen. Dies deutet an, dass eine Größenreduktion über Entfernen von Hub-Metaboliten vorzuziehen ist gegenüber einer solchen durch Entfernen anderer Stoffe, die möglicherweise verschiedene Pathways verbinden.

Besondere Aufmerksamkeit bei der Modellierung wird auf die Einbindung des Stärkemetabolismus gelegt. Bis heute ist die Initiation der Stärkesynthese nicht vollständig verstanden, und die Modellierung des Auf- und Abbaus einer Polymerstruktur in einem Petrinetz birgt Risiken für die spätere Analyse. Der Stärkemetabolismus wird in der Literatur zum Teil als Reaktionskaskade beschrieben, ohne speziellen Fokus auf die Polymerstruktur von Stärke zu legen. Diese Art der Beschreibung wird im vorgestellten Netzwerk genutzt, wobei besonderer Wert auf eine thermodynamische Durchführbarkeit gelegt wird. Hierfür werden - wie schon in anderen Studien zuvor - die stöchiometrischen Parameter aller Reaktionen im Stärkemetabolismus so angepasst, dass Stoff weder entsteht noch verschwindet. Über den Nachweis einer Invariante wird diese thermodynamische Machbarkeit der Modellierung nachgewiesen.

Da die Komplexität des Modells zu hoch ist, um die T-Invarianten zu berechnen, werden verschiedene Methoden zur Verringerung der Berechnungskomplexität entwickelt und angewendet. Hierbei kommen sowohl eine Netzwerkteilung als auch eine Netzwerkreduktion zur Anwendung.

Die Teilung des Netzwerkes in zwei Subnetze wird mittels zweier Herangehensweisen durchgeführt. Einmal wird das Netzwerk aufgrund biologischen Vorwissens auf eine Weise geteilt, die Rücksicht auf die - in der biologischen Wirklichkeit vorhandene - Kompartimentierung der Zelle und auf die Unversehrtheit wichtiger metabolischer Zyklen wie Zitrat- und Calvinzyklus nimmt (biologisch motivierte Netzwerkteilung). In einer zweiten Analyse wird das Netzwerk mit Hilfe einer angepassten Variante des Kernighan-Lin Algorithmus geteilt, welcher das Netzwerk in zwei gleich große Subnetze teilt, wobei die Anzahl der kreuzenden Kanten minimiert wird

(automatische Netzwerkteilung).

Die biologisch motivierte Netzwerkteilung ergibt zwei Module, das *Sucrose module*, bestehend aus 69 Metaboliten, 132 Reaktionen und 294 Verbindungskanten, und das *Citrate module*, bestehend aus 72 Metaboliten, 125 Reaktionen und 289 Verbindungskanten.

Aus der automatischen Netzwerkzerlegung resultieren zwei Module, Graph A mit 70 Metaboliten, 132 Reaktionen und 288 Kanten, und Graph B mit 72 Metaboliten, 127 Reaktionen und 304 Kanten.

Beide Teilungsstrategien erzeugen ähnlich große Module, von denen große Teile des *Sucrose module* mit Graph A und des *Citrate module* mit Graph B übereinstimmen. Hierbei bestätigt die automatische Teilung verschiedene Schnittkanten der biologisch motivierten Teilung, die sich ansonsten nicht direkt biologisch rechtfertigen lassen.

Im Laufe der weiteren Analysen zeigt sich, dass die Module der biologischen Teilung deutlich mehr T-Invarianten aufweisen, also deutlich mehr potentielle steady-state Reaktionswege ermöglichen, als die Module der automatischen Teilung. Das *Sucrose module* erlaubt 4602 T-Invarianten im Vergleich zu Graph A mit 2286 T-Invarianten, und für das *Citrate module* ergeben sich 3214 T-Invarianten im Vergleich zu Graph B mit 1966.

Es zeigt sich eine zusätzliche signifikante Verringerung des prozentualen Anteils aller potentiellen steady-state Reaktionswege, die eine von Rubisco katalysierte Reaktion enthalten, im automatisch geteilten Netzwerk. So nutzen ca. 45% aller Reaktionswege des reduzierten Netzes, und ca. 75% des *Sucrose module*, aber nur ca. 30% des Graph A die Rubiscokatalysierten Reaktionen. Zusätzlich können im Module der automatischen Teilung verschiedene Signalstoffe nicht aus den Produkten des Calvinzyklus, unter Annahme des steady-state, synthetisiert werden.

Diese Gründe weisen eher auf die biologisch motiverte als die geeignetere Teilung hin, obwohl vom Minimalitätskriterium ausgehend die automatische Teilung mit neun geschnittenen Kanten der biologisch motivierten mit zehn geschnittenen Kanten vorzuziehen wäre. Alle Module sowohl der automatischen als auch der biologisch motivierten Teilung sind CTI.

Es kann Fälle geben, in denen ein Netzwerk nicht CTI ist, aber alle Module einer Teilung sind CTI. Dies betrifft dann in der Folge auch die Interpretierbarkeit der T-Invarianten dieser Module. Um sicher zu gehen, eine CTI-konservierende Komplexitätsreduktion zu erreichen, wird eine Netzwerkreduktionsmethode vorgestellt, für die die Eigenschaft der CTI-Konserviertheit mathematisch nachgewiesen wird. Es wird ebenso gezeigt, dass durch die CTI Eigenschaft eines reduzierten Netzwerkes die CTI Eigenschaft des ursprünglichen Netzwerkes impliziert wird.

Zusätzlich wird gezeigt, dass eine enge Beziehung zwischen den T-Invarianten des reduzierten Netzwerkes und den T-Invarianten des ursprünglichen Netzwerkes besteht. Diese erlaubt einen Rückschluss von den T-Invarianten des reduzierten Netzwerkes auf die T-Invarianten des ursprünglichen Netzwerkes. Daraus ergibt sich die Aussagekraft der biologischen Interpretation der T-Invarianten des reduzierten Netzwerkes.

Die angewendete Reduktionsmethode definiert Substrukturen, das *invariant transition pair* (ITP) und das *common transition pair* (CTP). Ein ITP stellt eine triviale Invariante dar, die durch eine reversible Reaktion erzeugt wird. Eine unverzweigte Reaktionskette bildet ein CTP. Ähnlich wie schon in anderen wissenschaftliche Arbeiten dargestellt, reduziert der vorgestellte Algorith-

mus außerdem parallele Reaktionen, also Reaktionen mit gleichem Edukt und gleichem Produkt. Eine parallele Reaktion mit mehreren Edukten und/oder Produkten wird hierbei nicht reduziert.

Der Reduktionsprozess reduziert 86 CTP, 62 ITP und 2 paralelle Strukturen. Das resultierende Netz besteht aus 60 Metaboliten (45% von ehemals 134 Metaboliten des ursprünglichen Netzes), 131 Reaktionen (54% von ehemals 243 Reaktionen) und 329 Kanten (58% von ehemals 572 Kanten). Eine Berechnung der T-Invarianten für das reduzierte Netzwerk ergibt 27646 T-Invarianten, die das gesamte Netz überdecken.

Obwohl im Allgemeinen die Struktur eines reduzierten Netzwerkes nicht umfassend biologisch interpretierbar ist, finden sich doch für das reduzierte Netz ohne weitere händische Kuratierung einige Substrukturen, die biologisch nachvollziehbar sind. So werden im Reduktionsprozess in mehreren Schritten große Teile des Calvinzyklus, der Glykolyse und des Zitratzyklus in ein einzelnes Metabolit verschmolzen. In früheren Studien konnte gezeigt werden, dass die Enzyme der verschmolzenen Reaktionen von Calvinzyklus und Glykolyse gemeinsam benutzt werden, was im Gegenzug einen gemeinsamen Enzympool vorraussetzt.

Aufbauend auf den Ergebnissen der beiden Methoden der Komplexitätsreduktion, der Netzwerkteilung und der Netzwerkreduktion, werden weitere Analysen und Tests mit dem ursprünglichen Modell, seinen Modulen und dem abgeleiteten reduzierten Netz durchgeführt.

Eine besonders wichtige Eigenschaft für metabolische Netze ist die Lebendigkeit. Lebendigkeit wird mittels der Erreichbarkeitsgraphen beziehungsweise der Überdeckungsgraphen bestimmt, deren Berechnung EXSPACE-hart ist.

Um diese Problematik zu umgehen, wird ein neuer, auf metabolische Netzwerke zugeschnittener Lebendigkeitstest vorgestellt. Hierfür wird der Tokenzustand jedes Metaboliten auf einen Bool'schen Wert (Token vorhanden, Token nicht vorhanden) reduziert, und sämtliche Zwischenzustände ignoriert. Der resultierende Algorithmus hat eine Laufzeit von $O(|P| + |P| \cdot |T|^2)$.

Der Algorithmus ignoriert die anfängliche Tokenbelegung des Netzwerkes und findet in der Folge alle strukturell lebendigen Transitionen; das sind solche Transitionen, die unabhängig von der Anfangsbelegung lebendig sind. Der Test ergibt, dass das entwickelte Petrinetzmodell strukturell lebendig ist.

Das Netzwerk weist zusätzlich die Eigenschaft der Reinheit auf. Diese ist für eine sinnvolle Interpretation der T-Invarianten essentiell.

Die Interpretation der T-Invarianten unterstreicht ein biologisch angemessenes Netzwerkverhalten. So sind in der Logik des Netzwerkes alle thermodynamisch unmöglichen T-Invarianten erklärbar.

Eine neu entwickelte Organisationsstruktur für T-Invarianten wird vorgestellt, die *shared invariant sets* (SIS), und als Spezialform, die *maximal shared invariant sets* (maxSIS). SIS fassen alle T-Invarianten zusammen, denen ausgewählte Transitionen (Transitionen von Interesse) gemeinsam sind. Wenn in einem SIS die Menge der Transitionen von Interesse nicht erweiterbar ist, es also keine weiteren Transitionen mehr gibt, die ebenfalls in allen T-Invarianten des SIS vorkommen, wird von einem maxSIS gesprochen.

Mit Hilfe dieser neuen SIS-Struktur kann gezeigt werden, dass verschiedene Reaktionswege und Reaktionszyklen sich wie erwartet verhalten. So existieren viele Reaktionswege, die fixiertes

Kohlenstoffdioxyd aus dem Calvinzyklus großen Teilen des Netzes zugänglich machen. Hierbei wird unter anderem der Zitratzyklus versorgt, übereinstimmend zu früheren biologischen Studien.

Der Calvinzyklus verhält sich erwartungsgemäß. Werden verschiedene Reaktionen der Phase der Kohlenstoffdioxydfixierung und der Regenerationsphase des Calvinzyklus als Transitionen von Interesse ausgewählt, ergibt sich ein maxSIS, dass als erweiterte Transitionen von Interesse den gesamten Zyklus umfasst. Dies zeigt eine starke Abhängigkeit der beiden Phasen voneinander, vorausgesetzt das System befindet sich im steady-state.

Auswertungen einer beispielhaften T-Invariante ergeben, dass die RNA Synthese unter Verwendung von UTP ein mit der biologischen Wirklichkeit übereinstimmendes Verhalten aufweist. Die SIS-Analyse ergibt erwartungsgemäß eine zwingende Erfordernis verschiedener für die UTP Synthese essentieller Metabolite.

Die Analyse des Stärkemetabolismus führt zu einem *maximum common transition sets* (MCT-S), das den Stärkeaufbau und Teile des Stärkeabbaus überdeckt. MCT-S wurden in früheren Studien vorgestellt und bilden die kleinste biologisch aussagekräftige Zerlegung eines Netzwerkes. Mit Hilfe der SIS/maxSIS-Methodik, wobei als Transition von Interesse alle Transitionen des MCT-S ausgewählt werden, zeigt sich ein erweiterter Zusammenhang zwischen der Stärkesynthese und dem Stärkeabbau, der durch die MCT-S Analyse alleine nicht gefunden wird. Dieser Zusammenhang ist essentiell für den Nachweis der thermodynamischen Realisierbarkeit der Modellierung des Stärkemetabolismus.

Auch andere wichtige Reaktionswege zeigen eine starke Konsistenz mit der Biologie. Ein Großteil der *de novo* Synthese von Uridinemonophosphat (UMP) bildet sowohl im *Sucrose module* als auch in Graph A ein MCT-S. In der reduzierten Netzversion ist die Synthese über mehrere Schritte zu einer einzigen Transition zusammengefaßt. Da die Anfangsreaktionen der UMP-Synthese durch die Netzwerkteilung in die einzelnen Module abgetrennt ist, lassen die Ergebnisse der Reduktion darauf schliessen, dass ein MCT-S den gesamten Reaktionsweg überdeckt.

Der GABA-shunt Reaktionsweg bildet ein MCT-S. GABA ist ein wichtiges Metabolit in der Stressantwort und als alleinige Stickstoffquelle ausreichend.

Der Harnstoffzyklus und der Harnstoffabbau bilden ebenso ein MCT-S, wodurch ein enges Zusammenspiel dieser Reaktionswege angedeutet wird. Harnstoff ist eine wichtige Stickstoffquelle in Pflanzen und wird vorwiegend im Harnstoffzyklus synthetisiert.

Die Analyseergebnisse, dass wichtige Reaktionswege MCT-S darstellen, belegen die Plausibilität der Netzwerkmodellierung.

Die vorgestellten neuen Algorithmen und Strukturen erweitern die bisherigen Analysemethoden für eine aussagefähige Interpretation qualitativer biologischer Netzwerkmodelle. Insbesondere ermöglichen sie, in Kombination benutzt, eine Validation und Analyse ursprünglich zu komplexer Netzwerke.

Abstract

Systems biology has become an important research field during the last decade. The focus is on the understanding of the structures which emit the measured data. An important part of systems biology, the network analysis, investigates metabolic networks.

In general there are two different strategies to model biological systems: The quantitative modeling, which is based on real values and ordinary differential equations, and the qualitative modeling, which mostly uses discrete values and linear equation systems.

The quantitative modeling of a metabolic system is mostly interested in the determination of the 'flux', the rates at which different reactions operate, which defines the change of the concentration of the different compounds over time. The qualitative approach focuses more on the network topology and possible network decompositions into smaller subnetworks with certain properties.

Both strategies utilize graphs to illustrate the investigated system. For the qualitative modeling Petri nets have shown their very usefulness. These bipartite graphs inherit an intuitive representation of biological pathways in addition to several sound mathematical analysis methods developed during recent years.

This work describes a qualitative modeling of the core metabolism of *Arabidopsis thaliana*. The PN consists of 134 metabolites, 243 reactions, and 572 connecting edges between metabolites and reactions. The connection to the environment is handled by 29 different reactions, 21 remove substances from the network (OUT), and 8 provide compounds (IN) to the network. There exists 4 external metabolites which are each connected to one of both, IN and OUT. Compartments and small metabolites like water and co-factors like NAD/NADH and ATP are not considered in the model.

As the model in its original form is too complex to calculate the t-invariants several methods for complexity-reduction in the form of network decomposition and network reduction are developed and applied. The application of these newly developed methods enable a determination and interpretation of the t-invariants.

The network decomposition is conducted by following two different approaches. First the network is manually split into two submodules by a biologically driven decomposition, which especially considers the integrity of several important metabolic cycles like the Calvin cycle, the citric acid cycle, and the urea cycle. Second the network is automatically decomposed into two submodules using an adapted Kernighan-Lin algorithm. This is a heuristic approach to split a network into two submodules and at the same time minimizing the number of crossing edges between the modules.

As the decomposition possibly does not conserve the CTI property, a network reduction procedure is presented, which is mathematically proven to conserve this essential property.

The analysis of the network decompositions and of the reduced network focuses especially on determination of the liveness property as well as on the investigation of the steady-state pathways represented by the t-invariants of the network.

Liveness is, in general, determined using the reachability graph or the coverability graph. The computation of both is EXPSPACE-hard. For the liveness-test of the network an algorithm is developed which resembles an adapted boolean coverability analysis and has a running time of $O(|P| + |P| \cdot |T|^2)$. This test is especially designed for metabolic networks, and detects structural liveness of transitions. Structural live means live independent from the initial marking. The PN model is structural live.

A general analysis of the t-invariants demonstrates that the t-invariants support the thermodynamic feasibility of the network, and an investigation of an exemplary t-invariant underlines its biological meaningfulness. The network is pure, which renders the t-invariants feasible.

Next to the established organizational structure for t-invariants, the *maximum common transition sets* (MCT-S) a new structure is introduced. MCT-S represent the smallest biologically meaningful entities, into which a network can be decomposed. The new structure is called *shared invariant set* (SIS) and its special form *maximal shared invariant set* (maxSIS). All of these organizational structures are successfully applied in the analysis process.

Various pathways are investigated regarding their consistency with biological data. It is demonstrated that the carbon fixation phase and the regeneration phase of the modelled Calvin cycle strongly depends on each other. Additionally potential steady-state pathways exist, which provide this fixed carbon to nearly all parts of the network, especially to the citric acid cycle.

An analysis of the starch metabolism demonstrates, that a combined application of MCT-S and SIS/maxSIS discloses certain dependencies that are not detected by the sole application of the MCT-S method.

The MCT-S analysis reveals close cooperation of important pathways, e.g., the *de novo* synthesis of UMP, the GABA-shunt, and the urea cycle in combination with urea degradation.

Altogether, the new and adapted algorithms and structures extend the established instruments utilized for a feasible interpretation of qualitative biological network models. Used in combination, they provide a possibility to validate and interpret previously over-complex models.

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Aim of the Work . . . 2

1.3 Course of the Work . . . 2

2 Studies of Biological Networks **4**

2.1 Modeling Approaches . . . 4

2.1.1 Quantitative Modeling . . . 5

2.1.2 Qualitative Modeling . . . 7

2.1.3 Models of *Arabidopsis thaliana* . . . 8

2.2 Decomposition of Networks . . . 10

2.3 Reduction of Networks . . . 11

2.4 Verification of Biological Network Models . . . 12

3 Materials and Methods **14**

3.1 Petri Nets . . . 14

3.2 Petri Net Properties . . . 16

3.2.1 Liveness . . . 16

3.2.2 Structural Properties . . . 17

3.2.3 Invariants . . . 17

3.3 Construction Principles of Metabolic Petri Nets . . . 19

3.4 Node-Degree and Power-Law Distribution . . . 22

3.5 Databases and Software . . . 22

3.6 Kernighan-Lin Algorithm . . . 23

4 Results and Discussion **25**

4.1 The *Arabidopsis thaliana* Petri Net Model . . . 25

4.1.1 General Properties of the Model . . . 26

4.1.1.1 Environmental Connection . . . 26

4.1.1.2 Small Metabolites . . . 28

4.1.2 Sucrose Subnetwork . . . 31

4.1.2.1 Starch Synthesis . . . 33

4.1.2.2 Starch degradation . . . 35
4.1.3 Citrate Subnetwork . . . 37
4.1.4 Shikimate Subnetwork . . . 37
4.1.5 UTP Subnetwork . . . 40
4.1.6 Comparison to Models in the Literature . . . 41
4.2 Network Decomposition . . . 42
4.2.1 Biologically Driven Decomposition . . . 43
4.2.2 Automatic Decomposition . . . 51
4.2.3 Comparison of the two Decomposed Modules . . . 58
4.3 Reduction of the Network . . . 59
4.3.1 Definitions . . . 60
4.3.2 CTP and ITP Reduction Rules . . . 61
4.3.3 Extension of the CTP Reduction . . . 67
4.3.4 Comparison to Other Reduction Methods . . . 69
4.3.5 Complexity Aspects of the Reduction Method . . . 70
4.3.6 Application to the *Arabidopsis thaliana* Model . . . 70
4.4 Network Verification . . . 75
4.4.1 Dynamical and Structural Properties . . . 75
4.4.1.1 Liveness . . . 76
4.4.1.2 Structural Properties . . . 82
4.4.2 Invariant Analysis . . . 82
4.4.2.1 General Invariant Analysis . . . 84
4.4.2.2 Exemplary Inspection of a Single T-Invariant . . . 86
4.4.2.3 T-Cluster . . . 88
4.4.2.4 MCT-Sets . . . 88
4.4.2.5 Shared Invariant Sets . . . 90

5 Summary and Conclusions **96**

Bibliography **103**

Appendix

A General Definitions **116**

B Tables **117**
B.1 Table of Metabolites . . . 117
B.2 Table of Reactions . . . 121

C Programm Outputs **132**
C.1 Reduction of the *Arabidopsis thaliana* Network . . . 132
C.2 Static Conflicts . . . 145
C.3 Deleted Metabolites . . . 149

C.4 Decomposition Comparison 151

List of Figures

3.1 Firing process of an example PN. 15
3.2 Possible representations of biochemical networks 20
3.3 Stepwise creation of a Petri net 21

4.1 PN model of the metabolism of *Arabidopsis thaliana* 27
4.2 The distribution of node-degrees of compounds in *Arabidopsis thaliana* 30
4.3 The distribution of node-degrees of reactions 32
4.4 *Sucrose subnet* 34
4.5 Simplified starch biosynthesis/degradation modeled by a PN 35
4.6 Starch synthesis and degradation in the *Arabidopsis thaliana* PN. 36
4.7 *Citrate subnet* 38
4.8 *Shikimate subnet* 39
4.9 *UTP subnet* 40
4.10 Biologically driven decomposition of the *Arabidopsis thaliana* network. 47
4.11 Examples of linking areas in the biological decomposition 48
4.12 Examples of linking areas in the biological decomposition,continued 49
4.13 Automatic decomposition of the *Arabidopsis thaliana* network using the KL-algorithm. 53
4.14 Examples of linking areas in the automatic decomposition 56
4.15 Examples of linking areas in the automatic decomposition, continued 57
4.16 Problematic nature of the CTI determination of decomposed networks 60
4.17 Example for a CTP and its reduced form. 62
4.18 Example for an ITP and its reduced form. 64
4.19 Reduced Network 73
4.20 Liveness-test of our *Arabidopsis thaliana* model. 81
4.21 Example of a cyclic t-invariant. 85
4.22 SIS of the Calvin cycle in the reduced network. 91

List of Tables

4.1 Omitted metabolites 29
4.2 Properties of different metabolic networks 41
4.3 Table of edges cut by biologically driven decomposition 44
4.4 Table of size of modules: Biological decomposition 46
4.5 Table of size of modules for automatic decomposition 52
4.6 Table of edges cut by automatic decomposition 54
4.7 Comparison of the two decompositions 59
4.8 Table of reduction efficiency. 71
4.9 Number of invariants per module and reduced net 83
4.10 Exemplary t-invariant calculated on the reduced network. 87

B.1 Metabolites of the model 117
B.2 Reactions of the model 121

C.1 Table of reductions. 132
C.2 Table of static conflicts of the network 145
C.3 Table of deleted secondary metabolites and their possible connections to the network 149
C.4 Comparison between automatic and biological decomposition 151

List of Abbreviations

13C	Carbon isotope of atomic weight 13
14C	Carbon isotope of atomic weight 14
ACoM	Agglomeration around common motifs
ADP	Adenosine-5'-diphosphate
AMP	Adenosine-5'-monohosphate
ATP	Adenosine-5'-triphosphate
C	Carbon
CDP	Cytidine-5'-diphosphate
CMP	Cytidine-5'-monophosphate
CTL	Computation tree logic
CTI	covered by t-invariants
CoA	coenzyme A
CTP	Common Transition Pair
EC (number)	Enzyme Commission (number)
EFM	elementary flux mode
EM	elementary mode
FBA	flux balance analysis
gcd(a,b)	greatest common divisor of a and b
GDP	Guanosine-5'-diphosphate
GMP	Guanosine-5'-monophosphate
GRSN	generalized reducible subnets
GTP	Guanosine-5'-triphosphate
HFPN	hybrid functional Petri nets

IN	input transition(s)
ITP	Invariant Transition Pair
KEGG	Kyoto Encyclopedia of Genes and Genomes
KL-algorithm	Kernighan-Lin algorithm
lcm(a,b)	least common multiple of a and b
maxSIS	maximal shared invariant set
MCT-S	Maximum Common Transition Sets
MFA	metabolic flux analysis
ML	MONALISA
NAD(P)/NAD(P)H	Nicotinamide adenine dinucleotide (phosphate) oxidised/reduced form
NP	the complexity class NP
ODE	ordinary differential equation
OUT	output transition(s)
PN	Petri net(s)
PNFL	Petri nets with fuzzy logic
P/T PN	place-/transition Petri net
PTA	Polynomial time algorithms
SUBA	Arabidopsis Subcellular Database
SIS	shared invariant set
t-invariant(s)	transition invariant(s)
UDP	Uridine-5'-diphosphate
UMP	Uridine-5'-monophosphate
UTP	Uridine-5'-triphosphate

Chapter 1

Introduction

1.1 Motivation

The study of systems behind biological processes is condensed in the field of systems biology (Kitano, 2002b; Ideker et al., 2001). With the strongly increasing amount of available biological data, the development of new computational methods to investigate the system behind the data becomes more and more important (Kitano, 2002a).

Several quantitative models have been developed for the model organism *Arabidopsis thaliana* (Poolman et al., 2009; Gomes de Oliveira Dal'Molin et al., 2010; Radrich et al., 2010; Mintz-Oron et al., 2012). These models provide insights into the arrangement of the network fluxes. However as the same quantitative data fit to different network topologies with the same level of confidence, it is not possible to provide a final mathematical verification of such a network (Wiechert, 2002; Crampin et al., 2004; Craciun et al., 2009; Masakapalli et al., 2010; Sweetlove and Ratcliffe, 2011). There is further research required to prove whether an appropriate qualitative model can generate the base for a successful network validation.

Graphs are used to describe relations between objects. For the modeling of metabolic network purposes bipartite graphs with its two different node types, which can encode metabolites and reactions, seem to be the most suitable way of representation. Petri nets, a well-known sort of bipartite graphs, are intuitively recognized as a biochemical network and comprise an established set of sound mathematical model checking methods (Heiner and Koch, 2004; Koch and Heiner, 2007; Koch et al., 2011). Overall, Petri nets grant a powerful tool for qualitative metabolic network modeling, which is easy to use.

In general metabolic networks inhere a complex topology. Therefore the verification process, which relies especially on the determination and interpretation of t-invariants, can be unfeasible due to the computational complexity of their calculation (Acuña et al., 2009, 2010; Pérès et al., 2011). There are additional research efforts required to specify forms of suitable complexity reduction to enable an analysis of the t-invariants.

Topological and stoichiometric data are necessary for the modeling of biological networks. These data can be obtained e.g. from existing databases like AraCyc. This database was initially created by computational predictions and its curation process is still ongoing (Mueller et al., 2003;

Zhang et al., 2005, 2010; Chae et al., 2012; Caspi et al., 2012). An unaudited usage of data may lead to the specification of an incorrect network topology (Poolman et al., 2006; Ginsburg, 2009; Radrich et al., 2010). Therefore, during the modeling process it is necessary to check that all extracted data are consistent with the literature.

1.2 Aim of the Work

The aim of this work is to create a qualitative network model of the metabolism of *Arabidopsis thaliana* consistent with the existing literature and using Petri net modeling techniques. Suitable decomposition and reduction methods have to be developed and applied to verify the emerging complex network topology. A successful validation process requires the determination of several structural, behavioral and invariant properties and their interpretation, especially in a biological manner.

1.3 Course of the Work

The following Chapter 2 contains an overview of the current status of academic works in the areas of quantitative and qualitative modeling approaches for the analysis of biological networks. A focus is put on results concerning metabolic networks and *Arabidopsis thaliana*. As the verification of our developed network will require special decomposition and reduction procedures, recent advancements in these research areas are presented. Studies on the validation of Petri nets models put attention on the analysis of invariants and behavioral properties like liveness.

A detailed description of Petri nets and their properties is given in Chapter 3. We use this mathematical structure for the representation of biological networks and illustrate the construction of a metabolic Petri net by an example. The Kernighan-Lin algorithm is described which will be applied to our constructed metabolic network model for decomposition purposes.

The gained results of the network modeling and validation process and their discussion are contained in Chapter 4. First, we introduce our Petri net model of *Arabidopsis thaliana* and discuss approaches for its verification. In addition to a reflection of the global properties of the developed core metabolism model, the detailed network description takes place on the basis of several subnetworks. The modeling process is illustrated by a part of one of these subnetworks, the starch metabolism in the Sucrose subnet.

Due to the high complexity of this metabolic network, the verification of the complete model by t-invariants is not possible. Therefore, it is necessary to implement methods for complexity reduction. First, the network is decomposed following biological pathway structures. Additionally the Kernighan-Lin algorithm is applied to the network to achieve an automatic decomposition. To ensure the CTI property of the network, a suitable reduction method is developed and its applicability is proven. This new method is used to derive a reduced Petri net from the network model of *Arabidopsis thaliana*.

In Chapter 4.4 the results of network verification are outlined. For the complete network the validation is performed on the basis of a newly developed liveness analysis. The determination

and inspection of the t-invariants is conducted on the four modules emerging from the two network decompositions and on the reduced network. Hereby various methods like the calculation of MCT-sets are applied and the new methodology of shared invariant sets is introduced.

Using the described decomposition and reduction techniques, a process is demonstrated to enable the network validation by the determination and interpretation of the t-invariants for a former overly complex network.

Chapter 2

Studies of Biological Networks

The integration of the metabolic profile into the analysis of cellular networks is a rapidly growing field of great interest (Reaves and Rabinowitz, 2011; Mittler and Blumwald, 2010; Urano et al., 2010; Fukushima et al., 2009; Feist et al., 2009). New insights regarding the determination of the metabolome (Bermejo et al., 2011; Winder et al., 2011; Lei et al., 2011; Zhang et al., 2011; Harkewicz and Dennis, 2011) or the investigation of the localization of the proteome (Yates, 2013; Simm et al., 2013; Papasotiriou et al., 2013; Weis et al., 2013) motivate the creation of new network models. A number of methods for the analysis of metabolic networks has been established (Tomar and De, 2013) not only for microorganisms, but also for experimental and mathematical modeling approaches for the investigation of plant metabolic networks (Rios-Estepa and Lange, 2007).

2.1 Modeling Approaches

The range of network modeling approaches for the investigation of biological networks is vast. One field of interest is the inspection of signal transduction pathways (Bhalla and Iyengar, 1999), e.g. by a qualitative approach (Sackmann et al., 2006). A second area are the studies of gene regulatory networks (De Jong, 2002), e.g. implementing a hybrid strategy (Matsuno et al., 2000). A hybrid strategy combines a continuous model, i.e., based on ordinary differential equations (ODE), with a qualitative discrete model.

Another field of research is the investigation of metabolic networks. The strategies of metabolic network modeling can be divided into quantitative and qualitative approaches. Quantitative approaches are usually based on the determination of network-flux with the help of ODE, e.g., the investigation of the core metabolism of barley (*Hordeum vulgare*) (Grafahrend-Belau et al., 2009). Qualitative approaches often investigate structural properties of a network, e.g., the examination of the sucrose breakdown in potato tubers (*Solanum tuberosum*) (Koch et al., 2005).

Network analysis is a powerful tool in determining functions of undetectable fast biological processes (Karr et al., 2012) and is a essential prerequisite for metabolic engineering (Stephanopoulos and Stafford, 2002; Schwender, 2008; Blank and Ebert, 2013). Recently, a whole-cell computational model of *Mycoplasma genitalium* was able to predict properties as localization, count, and activity for each molecule in the cell. These predictions were consistent

with observed values of the cellular chemical composition. Additionally, the model provided insight into some biological processes, for which experimental examination is difficult or nearly impossible (Karr et al., 2012).

Networks should be interpreted as a 'potential' topology exhibiting all possible connections. Due to the different specializations of cells and the genetic regulations that come along with these specializations it is often not the case that every edge and node is present together in the network *in vivo* (Klein et al., 2012).

In the following we introduce various quantitative and qualitative modeling approaches of biological networks, and highlight important progresses made in the metabolic modeling of *Arabidopsis thaliana*. We describe existing concepts of graph decompositions and graph reductions, and inspect network verification approaches as well.

2.1.1 Quantitative Modeling

Quantitative modeling *quantifies* certain properties of a network. Of outstanding interest is the analysis of the *flux*, i.e., the flow of substance through the network. This flow is specified in the ratios of substance-conversion, identified for each reaction, and traditionally represented in an equation system of ODE. Different strategies have been developed to investigate the network flux and to utilize the gained information. Two kinds of flux analysis methods reached the main research attention (Rios-Estepa and Lange, 2007; Tomar and De, 2013; Klein et al., 2012) during the last years: the *metabolic flux analysis* (MFA) and the *flux balance analysis* (FBA).

Metabolic flux analysis is a mixture of computational and experimental methods (Ratcliffe and Shachar-Hill, 2006). Two different approaches exist: the *dynamical* approach and the *steady-state* approach. The aim of MFA is the creation of a flux map for a given network. MFA relies on experimental flux determination as main data source. Possible gaps are closed with parameter estimation methods. Dynamical MFA investigates the flux change, possibly under the effect of environmental alterations, e.g., a rearrangement of input fluxes. Steady-state MFA examines the flux distribution for cells in their metabolic steady-state (Rios-Estepa and Lange, 2007). Experimental procedures for metabolic flux analysis are based on labeled isotopes, in the overwhelming majority of cases the isotopes of carbon (C). Here, it is differentiated between radioactive labeling, using ^{14}C-isotopes, and stable labeling, utilizing ^{13}C-isotopes. In most cases, ^{13}C-isotopes are used for steady-state MFA because the positioning of the carbon atoms is of interest, and the radioactive isotope is used in the dynamical MFA due to the high sensitivity of the measurements (Ratcliffe and Shachar-Hill, 2006). The flux map is finally created by the computer aided interpretation of isotope patterns (Sauer, 2006). Here, one of the core requirements is a correct network topology, representing the metabolic interactions. MFA is still a research area of interest and methods on that are in development (Wiechert and Nöh, 2013).

During the last years several reviews concerning MFA utilizing ^{13}C approaches have been published (Wiechert, 2001; Sauer, 2006), some of them concerning network analysis methods in general (Morgan and Rhodes, 2002; Klein et al., 2012; Tomar and De, 2013). It has been confirmed that the enrichment of the ^{13}C-isotope in multiple carbon compounds like glucose does

not influence the metabolism in *Arabidopsis thaliana* (Kruger et al., 2007). Following the routes of carbon or other atoms through the network may require a complex mixture of isotopomers and subsequent an elaborated computational interpretation of the findings. A new algorithm utilizing Monte Carlo sampling was published (Schellenberger et al., 2012), which provides a method to estimate the possible degree of success of the experiment. The algorithm calculates whether there is any new insight possible from the given data. In addition the algorithm predicts an optimal labeling pattern for the labeling compound.

Flux balance analysis is a constrained-based steady-state method that maximizes a given objective function. Contrary to MFA, FBA is a solely computational method (Orth et al., 2010). It is based on a $m \times n$ matrix, containing the stoichiometric factors of any metabolite in each reaction of a network built of n reactions and m metabolites. The value of the factor is zero, if a metabolite does not participate in a reaction, is less than zero, if the metabolite is an educt of this reaction, and is greater than zero, if the metabolite is a product of this reaction. Each of the entries of the matrix formulates a constraint of the solution space of the objective function, and a linear program maximizes the function. Additional to the stoichiometric constraints, flux constraints can be set for each reaction, e.g., a minimal and/or maximal flux value, allowing to exactly manipulate the simulation to recreate the environmental conditions under investigation.

If the objective function considers, e.g., the overall growth rate of an organism and the influence of each metabolite to the overall growth rate is specified, an FBA determines the fluxes which maximizes this growth rate under the given environmental conditions (Klein et al., 2012). But, the use of FBA is limited as it cannot predict metabolite concentrations (Edwards et al., 2002) and the presented solution is possibly not a unique solution of the maximization or minimization problem (Sweetlove and Ratcliffe, 2011).

Several reviews summarize the impact of FBA in the fields of metabolic analysis of microbes (Edwards et al., 2002; Shastri and Morgan, 2005) and plants (Sweetlove and Ratcliffe, 2011), or the challenges FBA has to master in the future (Lee et al., 2006b).

Recently, FBA has been utilized for the investigation of substrate shortage in a hand-made model of barley (Grafahrend-Belau et al., 2009) and an estimation of gene importance with the help of a knock-out analysis. There are applications to bacteria for many years as well (Reed and Palsson, 2003) including the development of genome-scale networks (Schilling et al., 2002). Other strategies include the network validation with the help of FBA in genome-scale networks of plants (Poolman et al., 2009; Gomes de Oliveira Dal'Molin et al., 2010). The combination of FBA and MFA can uncover the efficiency of carbon and energy conversion in plants (Chen and Shachar-Hill, 2012).

Especially the field of metabolic engineering (Blank and Ebert, 2013) benefits from FBA. Metabolic engineering is not only the insertion of new production pathways, but also the struggle to maximize the yield of products already part of the metabolic network of the producing species (Stephanopoulos and Stafford, 2002; Libourel and Shachar-Hill, 2008). MFA has proven its value for the engineering of metabolic networks (Schwender, 2008; Kruger and Ratcliffe, 2009) as well.

2.1.2 Qualitative Modeling

Several qualitative modeling approaches exist in systems biology, e.g., boolean networks (Wang et al., 2012) to analyze several different cellular systems and applications related to boolean networks to investigate genetic or metabolic networks (Akutsu et al., 2000). Other advancements address the state space problem with the help of logical approaches (Bérenguier et al., 2013), or utilize qualitative differential equations and inductive logical programming to deduct Petri nets (PN) from biological data (Srinivasan and Bain, 2012; Srinivasan and King, 2008). Several reviews highlight the usage of PN in systems biology (Pinney et al., 2003; Chaouiya, 2007; Koch et al., 2011).

Since the development of PN by Carl Adam Petri in 1962 (Petri, 1962) PN have been utilized in a range of research fields. Developed for the handling of concurrent systems, Petri nets are a most welcome method to analyze workflows (van der Aalst, 1998) or multi-thread computer programs (Martiník, 2011). PN have as well been employed to explore the variation of plasmid number in bacteria with the help of a stochastic PN (Goss and Peccoud, 1999) and the detailed analysis of the assembly of the spliceosome (Bortfeldt et al., 2010) with the help of methods such as MCT-Sets (Sackmann et al., 2006) and t-cluster (Grafahrend-Belau et al., 2008). Other approaches automatically reconstruct Petri nets with Fuzzy Logic (PNFL), a type of PN related to Hybrid Functional Petri nets (HFPN), based on time-series data of gene regulatory data (Küffner et al., 2010).

Research fields in systems biology taking advantage of PN modeling are vast. Signal transduction pathways, gene regulatory networks as well as metabolic pathways are in the scope of *in silico* Petri net models. A Place-/Transition Petri net (P/T PN) of the mating pheromone pathway in *Saccharomyces cerevisiae* has been built (Sackmann et al., 2006), validated by new concepts of system verification. Here a structural approach for the inspection of the pathway is used (Sackmann et al., 2006). A simulation of a signal transduction network stimulated by the epidermal growth factor in human *Homo sapiens sapiens* utilizing colored PN (Lee et al., 2006a) shows a relation between the discrete simulation of the Petri net and an ODE-based model. In colored PN tokens can be of different colors, and transitions can be restricted to be only activated by tokens of a certain color and possibly effect these tokens.

Gene regulatory and signal transduction pathways can be tightly interwoven as signal transduction often influences the expression of one or several genes. There is an approach to model a combination of a signal transduction pathway and the correlated gene regulatory network in the analysis of the high osmolarity glycerol pathway in *Saccharomyces cerevisiae* as a PN (Tomar et al., 2013), generating a mathematical model of the osmoregulation in yeast. A modeling and validation process of the apoptosis signaling pathway has been published using PN as mathematical representation (Heiner et al., 2004). Several new methods to analyze gene regulatory networks with the help of PN have been proposed as well, e.g., the new class of so-called *Multi-level Regulatory Petri Nets* (Chaouiya et al., 2008). These networks contain several places for each gene representing different expression states which facilitate the detailed modeling of the current expression state. Modeling the gene regulation network responsible for the Duchenne muscular dys-

trophy leads to the conclusion that the RAP2B-calcineurin pathway seems to have a strong effect on the network behavior (Grunwald et al., 2008). In addition a tree-like structure called *mauritius maps* is introduced, which orders transitions depending on their occurrence in t-invariants. This structure allows, e.g., a computational knock-out analysis.

Petri nets have been applied to metabolic networks for about 20 years (Reddy et al., 1993; Pinney et al., 2003; Chaouiya, 2007; Koch and Heiner, 2007; Koch, 2010; Koch et al., 2011). It has been shown that the physiological steady-state can be inspected with the help of the t-invariants of a PN (Voss et al., 2003). T-invariants are related to *elementary (flux) modes* (EM or EFM) (Schuster and Hilgetag, 1994; Schuster et al., 2000, 2002a; Zevedei-Oancea and Schuster, 2003). Several other publications as well use t-invariants and their interpretation to analyze metabolic networks. To reduce the computational effort of the t-invariant determination, a decomposition of the metabolic network of *Mycoplasma pneumoniae* into smaller subnets is suggested (Schuster et al., 2002b), depending on the node degree of metabolites, i.e., the number of reactions these metabolites participate in. Another study establishes a model for the sucrose breakdown pathway in *Solanum tuberosum* and validates the model by showing a biological meaning for its t-invariants (Koch et al., 2005). The combination of regulatory and metabolic pathways to regulated metabolic pathways with the help of PN has been investigated (Simão et al., 2005).

Several extensions can be applied to P/T Petri nets, forming mixtures of qualitative and quantitative approaches and blurring the borders between the two modeling strategies and allowing PN to be applied to quantitative modeling of metabolic networks. The regulation of the urea cycle in the human liver has been investigated with a hybrid Petri net approach (Chen and Hofestädt, 2003). Here, the metabolic fluxes are described by the continuous parts and the regulation is modeled by discrete elements. Other bio-pathways have been represented as hybrid Petri nets as well (Genrich et al., 2001; Matsuno et al., 2003). A case study provides a quantitative model of the *Extracellular Signal Regulated Kinase* signaling pathway, which is deduced from a discrete model, and derives protein concentrations from the qualitative model suggesting a possibility to estimate quantitative parameters from a qualitative analysis (Gilbert and Heiner, 2006).

2.1.3 Models of *Arabidopsis thaliana*

There exist various models for *Arabidopsis thaliana*. Several approaches are concerned with gene regulatory networks (Lucas and Brady, 2013), e.g., a method which is based on a bayesian network to investigate root cell differentiation (Bruex et al., 2012), and an approach focusing on stress response investigation in leaves (Hickman et al., 2013). Utilizing a recently developed statistical linear regression technique, novel genes are detected, which are involved in the mucilage biosynthesis in *Arabidopsis thaliana* (Vasilevski et al., 2012). In the following we focus on recent research results of metabolic modeling approaches of *Arabidopsis thaliana*.

The first steady-state MFA flux maps of *Arabidopsis thaliana* were introduced in 2008 (Williams et al., 2008). A heterotrophic cell suspension, grown under two different oxygen concentrations, was used as experimental data source. The results suggest a possible alteration of metabolite abundance without changes in the balance between respiratory and biosynthetic flux

or a major rearrangement of the network. The photosynthetic reactions (Calvin cycle, photorespiration) and pathways involved in the seed germination (glyoxylate cycle) are not part of the model, because there seems to be no evidence of their presence in heterotrophic cells (Williams et al., 2008). Based on this study further investigations are performed especially on the flux in the pentose phosphate pathway (Masakapalli et al., 2010). For this purpose three new models were derived from the aforementioned model, which differ in the compartmental organization of the pentose phosphate pathway. The measured data fit to each of the three models in an acceptable manner, which necessitate further investigations in addition to the MFA. This underlines the problems of metabolic flux analysis (Masakapalli et al., 2010). Most recently a review concerning ^{13}C-MFA in *Arabidopsis thaliana* and other systems was published (Kruger et al., 2012).

Several genome-scale networks of *Arabidopsis thaliana* have been developed, based on one or both of two major metabolic databases: AraCyc (Mueller et al., 2003; Zhang et al., 2005) and the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000; Kanehisa et al., 2008).

Poolman network (Poolman et al., 2009): This network model is based on AraCyc version 4.5. The data are extracted by a self-developed tool in an automatic fashion. The model consists of 1253 metabolites and 1406 reactions. After the removal of such reactions which are not possible to maintain a steady-state, the number of reactions is reduced to 855. In addition to the standard networks a minimal solution is provided, which is able to operate at steady-state and fulfills the conditions which constrain the linear programming verification approach. The minimal network consists of 232 reactions.

AraGEM (Gomes de Oliveira Dal'Molin et al., 2010): This model is as well automatically extracted from a database, the KEGG database version 49.0. Contrary to the Poolman model, AraGEM is partitioned into compartments. There are a total of 1567 reactions and 1748 metabolites. 446 of the metabolites from a dead end, meaning they are only produced or consumed by reactions. A total of 512 reactions connect these dead-end metabolites to the network. Some of these dead ends are possible because of missing knowledge to attached reactions, others are due to the automatic annotation of the KEGG database, in which reactions known to possess a certain EC number are automatically included in the database.

Radrich network (Radrich et al., 2010): As the other two models it is automatically reconstructed, but following another idea by creating a consensus network from AraCyc and KEGG. This is achieved by a comparison of metabolites and reactions between the two databases. There are three submodels established: the *core model*, the *intermediate model*, and the *complete model*. The core model is the intersection between the two databases, meaning it contains every metabolite and reaction which is present in both databases. In the intermediate model every reaction is present, for which either all educts or all products are part of the core model. The complete model is the union of both databases. The *core model* includes 753 reactions and 914 metabolites, the *intermediate model* consists of 1388 reactions and 1248 metabolites) and the *complete model* of 2315 reactions and 2328 metabolites.

Mintz-Oron network (Mintz-Oron et al., 2012): This is a method to semi-automatically derive metabolic networks from the databases KEGG and AraCyc (Mintz-Oron et al., 2012), including compartmental and tissue-specific localization data of reactions and metabolites in the process. The constructed model consists of 1363 reactions and 1078 metabolites. AraCyc and KEGG are used for the metabolic data and the Arabidopsis Subcellular Database (SUBA, (Heazlewood et al., 2007)) for compartmental data. Tissue specific localization data were taken from the literature. Based on this approach, a total of 942 out of 1363 inspected reactions are predicted to take place in every tissue. These reactions include reactions of the primary metabolite pathways such as glycolysis, pentose phosphate pathway, and fatty acid, nucleotide, and amino acid metabolism. The treatment of dead ends is performed by a gap-filling algorithm, which adjusts reaction directions or inserts new reactions. The validation of the network model is done by comparing the predicted metabolic flux values to measured experimental data. The main focus of this work is not on the model itself, but on the development of a pipeline to construct such models. Further experiments are required to validate the predicted results.

2.2 Decomposition of Networks

Graph theory (West et al., 2001) is an important research field in computer science to investigate properties and their determination in graphs. In graph theory two identifiers are known to describe the division of a graph into edge-disjoint subgraphs. In most cases the term *graph partition* implicates a focus on the weight of cut edges, while the term *graph decomposition* indicates a concentration on finding subgraphs that fulfill certain properties. A well-known problem of the graph partitioning is the *minimal bisection problem* (Kernighan and Lin, 1970; Karpinski, 2002). A minimal bisection is a set of edges, whose sum of weights is minimal, and its removal divides the graph into two equal sized subgraphs. This problem is NP-hard (Johnson, 1982).

An established example of a graph decomposition problem is the *H-decomposition*. This problem addresses the question, whether a set of edge-disjoint subgraphs can be found, which completely covers the graph, and each subgraph is isomorph to a second graph H. These partition and cover problems are closely related and are often NP-hard (Johnson, 1982; Cohen and Tarsi, 1991). Obviously, a decomposition is a partition of the vertices and edges implicating certain properties of the subgraphs, whereas a partition is a decomposition into subgraphs focusing on the edges which are not part of any subgraph.

Network decomposition has been utilized for the analysis of different network-types during the last years (Ravasz et al., 2002; Schuster et al., 2002b; Holme et al., 2003; Ma et al., 2004; Zaitsev, 2004; Guimera and Amaral, 2005). This method has been successfully applied to metabolic networks to reduce the computational effort required to calculate certain properties like t-invariants/elementary modes (Schuster et al., 2002b; Zaitsev, 2004). Other approaches of network decomposition have been published based on a topological overlap matrix (Ravasz et al., 2002), on the betweenness centrality (Holme et al., 2003), and on the shortest path length between reaction nodes (Ma et al., 2004). With the help of a simulated annealing based algorithm a method

based on the modularity of the network has benn developed, to find modules based on the relation between inter-module-connections and intra-module-connections of vertices (Guimera and Amaral, 2005).

One approach is to decompose networks into subnetworks with reduced complexity for the later-on computation of t-invariants of these subnetworks (Schuster et al., 2002b). The method is based on the removal of nodes of a certain node-degree. Hereby, the node-degree of a node n measures the number of other nodes n is connected to. There is a comparable method which removes vertices with a certain (Holme et al., 2003). The betweenness of a node n is the number of shortest paths between all pairs of nodes, which tangent an edge connected to n.

A further approach dedicated to Petri nets divide such networks into *functional nets*. A functional PN is a subnet of a PN, whose vertices have no connection to the rest of the Petri net except given sets of linking places (Zaitsev, 2004).

2.3 Reduction of Networks

In Computer Science, the question for the correctness of algorithms, their running times and other theoretical aspects of the reduction process are of interest (Arnborg et al., 1993). Other applications seem to be economically expedient, e.g., the reduction of circuits in electronic devices (US Patent No. 5790415, 1998) which reduces resource usage, and the minimization of fuel consumption of compressor stations in steady-state natural gas pipeline networks (Ríos-Mercado et al., 2002). Here network reduction methods are utilized which conserve the properties and behavior of the network and significantly reduce its complexity.

Also in verification approaches the execution of reduction processes prove their usefulness. A reduction algorithm for the verification of workflow graphs has been introduced and analyzed which iteratively applies conflict-preserving reduction steps to remove all structures from the graph that are definitely correct (Sadiq and Orlowska, 2000). If the remaining part of the total graph is empty, the workflow graph is structurally correct.

Reduction has been employed in the analysis of complex networks in the Petri net community during the last decades. Several studies analyze structural and dynamical PN reductions.

Various methods for the structural reduction of series of transitions and places have been developed (Lee-kwang et al., 1987). This reduction process is strongly influenced by the connection of surrounding nodes and conserves liveness and boundedness.

Other techniques rely on the properties two or more nodes share in the network. Parallel transitions or places are connected to the same sets of pre-places and post-places or pre-transitions and post-transitions, respectively. These parallel structures can be merged (Murata, 1989).

Several studies make use of these predefined reduction rules. Investigations on metabolic networks (Reddy et al., 1993) utilize reductions of parallel structures for the reduction of complexity. Other methods try to integrate deadlock avoidance policies in flexible manufacturing systems (Uzam, 2004), applying rules related to serial as well as parallel reductions. Deadlock analysis can also be performed using reduction techniques in the analysis of Ada tasking (Ada, 2012) programs (Shatz et al., 1996).

Beside these structural reduction methods, there exist dynamical techniques for PN reductions as well, based on the network behavior (Berthelot, 1987). Certain places p can be omitted from the network without any impact on the behavior of the network. This requires that for all post-transitions t_p of a place p it is true that, if all pre-places of t_p (except p) activate t_p, then p activates t_p as well. Here 'activate' means, that the number of tokens on the place p is greater or equal to the weight of the edge leading from p to t_p.

There are only a few studies on the reduction of metabolic networks for analysis purposes. One approach establish a matrix-based method to study the hierarchical organization of metabolic networks (Ravasz et al., 2002). To reduce the size of the matrix, non-branching pathways in the metabolic network of *Escherichia coli* are replaced by single reactions, resulting in a decrease of complexity.

2.4 Verification of Biological Network Models

Network verification or as it is also called, network validation, is an essential part of the modeling process. A model which is not validated is of little use for behavioral predictions, as it is not possible to confirm their quality. Several strategies have been developed to verify a network.

Quantitative models are often validated using measured flux data either taken from the literature or from own experiments. The verification of the Poolman network was performed using a combination of experimental results and a linear programming approach, showing that the model is capable of reproducing the experimentally measured growth rate and biomass production given the measured glucose uptake (Poolman et al., 2009). Similarly, flux analysis data predicted by AraGEM were compared to respective values taken from the literature (Gomes de Oliveira Dal'Molin et al., 2010), showing that its flux predictions are consistent with experimental results. In the Radrich model the focus is on the model construction process. A flux oriented validation of this network is not presented. Instead *levels of confidence* are discussed, depending on the assignment of reactions to the different submodels (Radrich et al., 2010).

The problem of such continuous models is, that it is difficult to confirm the network topology in a mathematical way by flux analysis. It is possible to have two topologically different networks, and still be incapable of distinguishing between them, even if experimental data of perfect accuracy and unlimited temporal resolution are available (Crampin et al., 2004; Craciun et al., 2009; Masakapalli et al., 2010; Sweetlove and Ratcliffe, 2011). Even if a mathematical verification is not possible, the common accepted validation process is to show the model can predict all experimental results (Wiechert, 2002).

The validation of qualitative models is reached by different methods depending on the type of qualitative model. There are approaches which utilize *computation tree logic* (CTL) using qualitative simulation to show that certain measured system behaviors can be possibly resembled by a gene regulatory model (Batt et al., 2005; Monteiro et al., 2008) or using symbolic model checking techniques to validate protein-protein and protein-DNA networks (Chabrier-Rivier et al., 2004).

Biological Petri nets are often validated with the help of t-invariants and other properties, espe-

cially *liveness*, of the network (Heiner and Koch, 2004; Koch and Heiner, 2007). T-invariants are comparable to the elementary (flux) modes (EM or EFM) (Schuster and Hilgetag, 1994; Schuster et al., 2000, 2002a; Gagneur and Klamt, 2004; Trinh et al., 2009). The computation of EFM (and therefore of t-invariants as well) can be very difficult as the counting of elementary modes is #P-complete (Acuña et al., 2009). #P is the complexity class of counting problems corresponding to decision problems in NP. The task of counting all Hamiltonian cycles in a graph corresponds to the decision problem whether at least one Hamiltonian cycle exists (Leiserson et al., 2001). The question of the complexity of enumerating all EM is undecided yet (Acuña et al., 2010; Pérès et al., 2011).

It has been shown that t-invariants are suitable for model validation of metabolic networks using the example of the sucrose breakdown pathway in potato tuber (Koch et al., 2005). T-invariants support also the validation of signal transduction pathways using the example of the apoptosis pathway (Heiner et al., 2004). To prove the validity of a model by using t-invariants it is required, that each of the invariants can be assigned to a biological meaning. This biological interpretation can be very difficult, as the number of t-invariants in complex networks can rise exponentially (Klamt and Stelling, 2002). Two organizing structures are suggested to interpret large set of t-invariants: *t-cluster* (Grafahrend-Belau et al., 2008) and *maximum common transition sets* (MCT-S or MCT-Sets) (Sackmann et al., 2006). T-cluster generate a matrix containing the distances between each pair of t-invariants and further perform a distance-based clustering procedure like neighbor joining (Saitou and Nei, 1987). MCT-S are defined as sets of transitions which exclusively participate in exactly the same t-invariants. Both structures have been successfully utilized to analyze the gene regulatory network of the Duchenne muscular dystrophy (Grunwald et al., 2008).

Chapter 3

Materials and Methods

3.1 Petri Nets

Petri nets (PN) were invented by Carl Adam Petri in 1962 (Petri, 1962) and are used to model biochemical pathways since 1993 (Reddy et al., 1993). There exist several surveys on Petri nets and their properties (Murata, 1989; Chaouiya, 2007; Koch and Heiner, 2007; Koch et al., 2011). The definition of PN and their properties are hardly different in the literature.

PN consist of two types of nodes, *places* and *transitions*, represented by the sets P and T, respectively, with $P \cap T = \emptyset$. The links between the nodes are directed, weighted edges. The links are represented by the set $F = \{P \times T\} \cup \{T \times P\}$, called *flow relation*, while the function $w : f \in F \to \mathbb{N}_0$ provides the weight of the edges: $w(p,t) = k, k \in \mathbb{N}_0$ is the weight of the edge $(p,t) \in F$. A weight of 0 is considered as the absence of the edge. The topology of a Petri net N is denoted as $N = (P,T,F,w)$.

A direct successor of a node is called its post-transition, if the successor-type is *transition*, or post-place, if the successor-type is *place*. A direct predecessor of a node is analogically called pre-transition or pre-place. Given a set D of nodes, the set of successors are denoted by $\bullet D$ and the set of predecessors by $D\bullet$. We speak of ...

... pre-transitions, for $\bullet p = \{t|(t,p) \in F\}$

... post-transitions, for $p\bullet = \{t|(p,t) \in F\}$

... pre-places, for $\bullet t = \{p|(p,t) \in F\}$

... post-places, for $t\bullet = \{p|(t,p) \in F\}$

$\bullet K$ refers to all pre-places or pre-transitions of the set K, and $K\bullet$ refer to all post-places or post-transitions. $\bullet K$ and $K\bullet$ contain either places or transitions but not both, depending on the node-type of K. We call places (transitions) *parallel*, if they share the entire set of pre-transitions (pre-places) and the entire set of post-transitions (post-places). Note that places can have no post- nor pre-places, while transitions can have no post- nor pre-transitions.

The dynamic simulation of PN is done with the help of discrete units, called *tokens*. The distribution of tokens define a system-state, called the *marking* m. This is a vector $m = (b_1, ..., b_{|P|}), b_i \in$

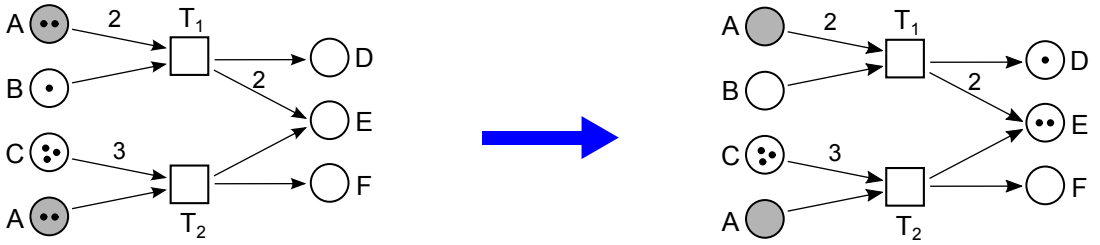
$\mathbb{N}$ (note: $\mathbb{N}$ includes the number 0), defining the number of tokens on every place. $m(p_k)$ denotes the number of tokens on place p_k, while m_k denotes a certain system-state, e.g. the state after k simulation steps. Dynamic simulation of PN is also called *playing the token game*. A Petri net is fully defined as $N = (P, T, F, w, m_0)$ with m_0 as the *initial marking*.

The simulation follows a fixed rule-set, called *firing rules*, or *firing* for short. Firing takes place at the transitions. If every pre-place p_i of a transition t_j holds at least as many tokens as the weight of the connecting edge, meaning $\forall p_i \in \{p_k | (p_k, t_j) \in F\} : m(p_i) \geq w(p_i, t_j)$, the transition t_j is called *enabled* or *activated*. Otherwise, it is called *disabled* or *not activated*. Note that transitions without pre-places are always enabled.

During the simulation enabled transitions may fire. If the firing occurs, tokens on every pre-place get destroyed, and tokens on every post-place get created accordingly to the weight of the respective edges. In the firing process of a transition t_j, we speak of...

... token destruction, meaning $\forall p_i \in \{p_k | (p_k, t_j) \in F\} : m_{k+1}(p_i) = m_k(p_i) - w(p_i, t_j)$

... token creation, meaning $\forall p_i \in \{p_k | (t_j, p_k) \in F\} : m_{k+1}(p_i) = m_k(p_i) + w(t_j, p_i)$



(a) Token distribution *before* firing (m_k)

T_1 fires

(b) Token distribution *after* firing (m_{k+1})

Figure 3.1: Firing process of an example PN. A is a logical place indicated by the gray color. (a) T_1 and T_2 are enabled. (b) T_1 has fired. Now both, T_1 and T_2 are disabled, because they share a pre-place (place A), which does not hold enough tokens to enable any of them. T_1 and T_2 are in a *static conflict* (see also Section 3.2.2).

Figure 3.1 shows an example of the firing process. The pre-places of the transition T_1 are the places A and B, the pre-places of the transition T_2 are the places A and C. For optical convenience, PN allow *logical places*. Logical places represent the same place p_i, but are drawn at different locations in the visualization of the network, helping to represent the network more clearly. In Figure 3.1, place A is a logical place. That means, if place A does not hold enough tokens to fire both transition T_1 and T_2 consecutively, both transitions are enabled, but only one of them may fire. If T_1 fires, two tokens on place A will be destroyed, leaving the place empty, which in turn disables T_2. If T_2 fires instead, one token on place A will be destroyed, leaving not enough tokens on place A to activate T_1. This sharing of pre-places is called *static conflict* of T_1 and T_2. In our example T_1 fires. This means, on every pre-place tokens get destroyed respective to the weight of the edge connecting that pre-place with T_1, and on every post-place tokens get created respective

to the connecting edge-weight. Note that, if tokens get destroyed on place A, they disappear on every of the logical nodes.

The set of all markings is M, with the initial marking is $m_0 \in M$. If a firing sequence s exists, which creates a new marking $m_k \in M$ from $m_{k-|s|}$, the marking m_k is called *reachable from* $m_{k-|s|}$. The set of all markings reachable from a given marking m_k is $R(m_k)$. Note that the number of markings in $R(m_k)$, for any m_k, can easily be infinite, especially in networks containing transitions which are always enabled (e.g. input transitions).

3.2 Petri Net Properties

This chapter is dedicated to introduce some of the Petri net properties and their determination. We define *liveness* for PN and introduce the *invariants*, especially the *t-invariants* (Lautenbach, 1973), of PN.

A PN is called *free-choice* (Desel and Esparza, 1995), if no part of the network is depending on another part. That means, if a place activates at least one of its post-transitions, it activates *all* of them (resulting in a free choice of one of them), which is the case if all transitions have only one pre-place. A weaker condition is given for *extended free-choice* nets: If two transitions share a pre-place, they have to share all pre-places.

3.2.1 Liveness

Some studies define a hierarchy of liveness levels (Murata, 1989). They are dependent on the number of possible firings of the inquired transition. We consider only one state of liveness which is comparable to: 'Every transition in the Petri net is *L4-live*' (the definition of L4-live is given in (Murata, 1989), we call it *live* as in (Koch and Heiner, 2007)).

Definition 3.2.1. *live:*
Given a Petri net $N = (P,T,F,w,m_0)$. A transition t is called live*, if there exist a firing sequence s for every marking in $R(m_0)$, that enables t. A Petri net is called* live*, if each of its transitions is* live*.*

This means, a transition and a Petri net as well can only be live relating to the initial marking m_0. If a Petri net is *live* for arbitrary markings m_0, it is called *structurally live*.

Definition 3.2.2. *dead:*
Given a Petri net $N = (P,T,F,w,m_0)$. A transition t is called dead*, if there exist no firing sequence s for every marking in $R(m_0)$, that enables t. A Petri net is called* dead*, if each of its transitions is* dead*.*

In every other case, a transition is called *non-live*. E.g. the number of firing sequences s that enables t is finite or the structure of the PN allows non-deterministic decisions affecting the liveness of t (see also *static conflict*, Section 3.2.2 and Figure 3.1).

3.2.2 Structural Properties

We use the following definitions for the respective structural properties:

Pureness: A PN is called *pure*, if no transition is connected to the same place with two edges of oppositional directions ($\forall t \in T : \nexists p : (t,p) \in F \wedge (p,t) \in F$).

free-choice: A PN is called *free-choice*, if each place either has only one post-transition, or its post-transitions have no other pre-places: $\forall p \in P : (|p\bullet| \leq 1) \vee (\bullet(p\bullet) = \{p\})$

Static conflict: Two transitions, which share an arbitrary set of pre-places, allow non-deterministic behavior of the PN. They are called to be in a *static conflict* for each pair of pre-places which the two transitions share (see also Figure 3.1). The number of static conflicts can be greater than the number of places in the net.

Structural deadlock: A *structural deadlock* is a set of places $K \subseteq P$, which can only gain tokens, if they already have any. This means, every transition t providing tokens to this set has pre-places in the same set: $\bullet K \subseteq K\bullet$.

Siphon or *trap*: The opposite structure of a structural deadlock is the *Siphon* or *trap*. A set of places $L \subseteq P$, which cannot lose previously gained tokens. This means, every transition destroying tokens on places in L as well produces tokens on (possibly other) places in L: $L\bullet \subseteq \bullet L$.

Ordinary: A PN is *ordinary*, if all its edge weights equal 1.

Homogeneous: A *homogeneous* PN contains only places, whose outgoing edges have the same weight.

Conservative: If all transitions are token-preserving, meaning they create as many tokens as they consume in one firing step, the PN is called *conservative*.

(Strongly) Connected: If a path between two nodes exist, they are called *connected*. A PN is called *connected* if all possible pairs of nodes are connected. If the path is a directed path, we speak of *strongly connected*.

3.2.3 Invariants

The definitions of invariants are widely known and illustrated in the literature (Murata, 1989; Koch et al., 2011; Lautenbach, 1973; Koch and Heiner, 2007). We define t-invariants and p-invariants analog to the literature. A detailed review concerning t-invariants including the explanation of an algorithm determining them is given by (Koch and Ackermann, 2013).

The two invariants of Petri nets are *place invariants* and *transition invariants*, called p-invariants and t-invariants for short. We focus here on t-invariants. To define these invariants, we need the concept of the *incidence matrix*. (Note that, if we multiply a matrix with a vector, we imply that either the number of rows or the number of columns is identical to the number of elements of the vector, whichever is needed to allow a meaningful multiplication of both.)

The weights w of a PN are assembled in a $m \times n$ matrix (m places, n transitions) called the *incidence matrix* C. The element $c_{i,j}$ represents the change of the number of tokens on place p_i, if transition t_j fires once:

$$c_{i,j} := \begin{cases} w(t_j,p_i) & ,if (t_j,p_i) \in F \\ -w(p_i,t_j) & ,if (p_i,t_j) \in F \\ w(t_j,p_i) - w(p_i,t_j) & ,if (t_j,p_i) \in F \wedge (p_i,t_j) \in F \\ 0 & ,otherwise. \end{cases} \tag{3.1}$$

Note that the third case in Equation 3.1 does only occur in PN which are not *pure*. If $c_{i,j} = 0$ occurs in a non-pure Petri net, it is not possible to distinguish between the case $w(t_j,p_i) = w(p_i,t_j)$ with $w(t_j,p_i), w(p_i,t_j) \neq 0$ (meaning the weights of the two edges of opposite directions between p_i and t_j are equal), and the case $w(t_j,p_i) = 0 \wedge w(p_i,t_j) = 0$ (meaning no edge between t_j and p_i).

Given a sequence of transitions $s = (t_{i_1}, t_{i_2}, ..., t_{i_q})$ and a corresponding vector $x = (x_1, ..., x_i, ..., x_n), x_i \in \mathbb{N}$, with each x_i giving the number of occurrences (firings) of t_i in s, the change of the token distribution is determined by $\triangle m = Cx$. $\triangle m$ is a vector containing at each position i the token-change of place p_i. There is no 'negative' firing of transitions, which means each x_i is greater or equal to zero.

The vector $supp(x) = (k_1, ..., k_i, ..., k_n)$, with $k_i = 1$, if $x_i > 0$, and $k_i = 0$ otherwise, is called the *support* of the vector x. The term $|supp(x)|$ is the number of entries of x which are greater than zero.

Commonly spoken t-invariants are collections of transitions that do not change the marking of the PN ($\Delta m = 0$), if they are all fired a certain number of times. How often a transition has to be fired to fulfill this condition is given by the vector x. Following this rule, solutions to the equation

$$Cx = 0 \tag{3.2}$$

with $x_i \geq 0$ are called t-invariants.

All solutions, which are not the trivial solution $\forall i : x_i = 0$, are called *true* or *semi-positive*. The set of x satisfying Equation 3.2 and the constraints for true t-invariants is called the *Invariant set* $I = \{x | x_i \geq 0 \wedge |supp(x)| > 0\}$. It is also possible to write the t-invariants in an $n \times |I|$ matrix V, each row representing a transition, and each column a t-invariant. $v_{i,j}$ represents the occurrence of transition i in t-invariant j.

The number of possible x satisfying Equation 3.2 is infinite if at least one solution x exists (because each $\alpha x, \alpha \in \mathbb{N}$ is a solution as well). This leads to the definition of *minimal semi-positive t-invariants*, which are a generating set of all t-invariants (Koch and Heiner, 2007). Minimal t-invariants are equivalent to elementary modes (Schuster et al., 2002a; Koch et al., 2005).

Definition 3.2.3. *A set I of solutions x of Equation 3.2 is called* minimal*, written I_{min}, if*

- $\nexists z \in I : \exists h \in I : supp(z) \subset supp(h)$ *(no invariant h contains another invariant z)*
- $\nexists \alpha \in \mathbb{N}_{>1} : \forall x \in I : x_i = \alpha x'_i, x'_i \in \mathbb{N}$ *(greatest common divisor of all entries of x is one)*

In this work, if not stated otherwise, the term *t-invariants* refers to the set I_{min}, described in Definition 3.2.3, and a *t-invariant* refers to an $x \in I_{min}$. We write $t_i \in I$, if there exist a t-invariant x in I, for that the x_i corresponding to t_i is greater than zero. The definition of $t_i \in x$ is analog.

If every transition of the network is part of at least one t-invariant ($\forall i : t_i \in I$), or in other words, a solution of Equation 3.2 with $\exists x \in I : \forall i : x_i > 0$ (x can here be a combination of other t-invariants $y \in I$) exists, the network is *covered by t-invariants* (CTI). This property is also called *consistency* (Murata, 1989), and a net that fulfills this property is called *consistent*. Note that the covering t-invariants do not need to be minimal.

Definition 3.2.4. CTI*:*
Let $N = (P, T, F, w, m_0)$ be a Petri net, and I its set of t-invariants. If $\forall t_i \in T : \exists x \in I : x_i > 0$, the net is covered by t-invariants (CTI). *(This is equivalent with $\exists x \in I : \forall i : x_i > 0$)*

3.3 Construction Principles of Metabolic Petri Nets

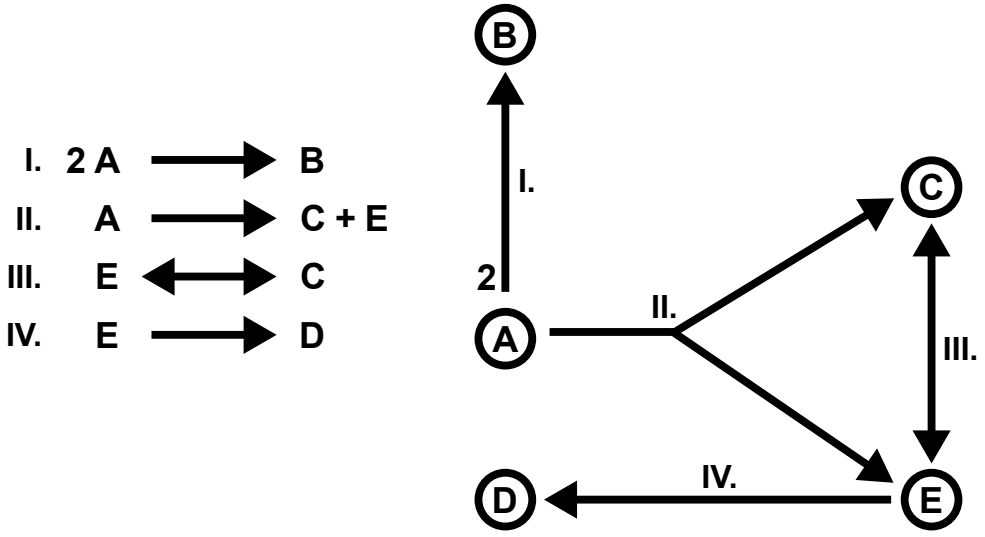
In this Chapter we briefly introduce the development of a metabolic Petri net. The knowledge source of a biochemicalPN is in most cases a hypergraph or a reaction equation system. An example for both cases is shown in Figure 3.2. It exhibits most of the possible interconnections between reactants:

- Molecules merge (2 A becomes 1 B)
- Molecules split (A becomes C and E)
- A reaction is reversible (E becomes C and vice versa)
- One molecule is converted to another molecule (E becomes D)

In general, transitions model the active parts of a system, while places model the passive parts. In a metabolic PN the places model the metabolites, which are transformed, and the transition model the reactions, which transform the metabolites. To convert a given reaction network into a PN, each metabolite has to be substituted by a place, and each reaction has to be substituted by a transition. Every edge has two connected nodes, one place and one transition, and a certain weight. The edge weight is set to the stoichiometric factor of its connected metabolite (the place) in its connected reaction (the transition). If a metabolite does not participate in a reaction, the weight is zero indicating the absence of the edge.

A token represents a discrete unit of the modeled entities. In a metabolic PN, a token represents a certain amount of a metabolite, e.g. one molecule. Figure 3.3 depicts the conversion of the given exemplary reaction system into a PN. The figure is divided into four subfigures, where each subfigure represents one of the four building steps (the ordering of the building steps is randomly chosen):

- Subfigure 3.3a illustrates the integration of the reaction I. $2A \rightarrow B$ to the PN. It should be noted that the PN needs two tokens on A to create one token on B, which is reflected by a weight of 2 of the edge from A to the reaction I.

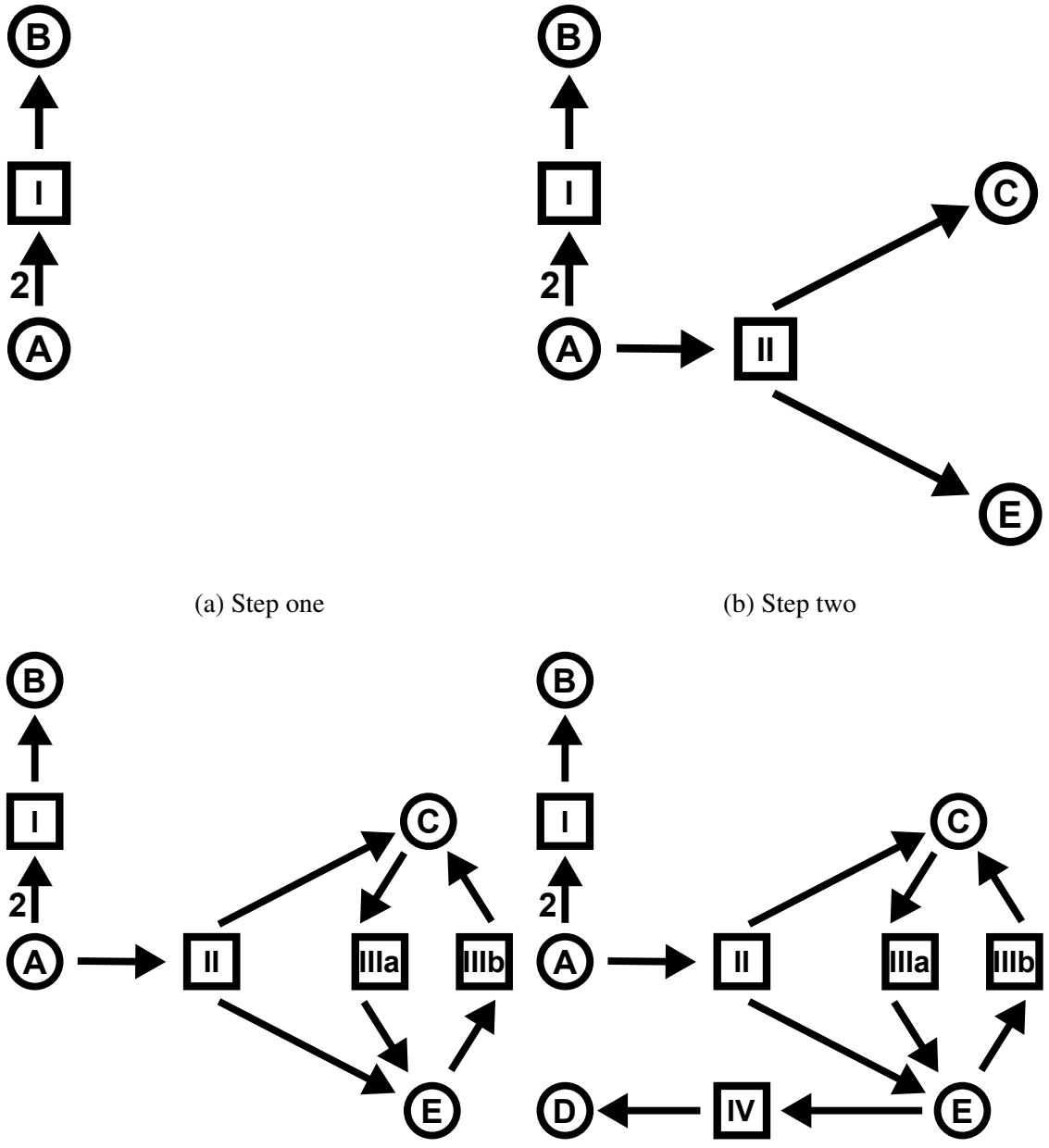


(a) Exemplary reaction system (b) Exemplary reaction hypergraph

Figure 3.2: Two possible representations of biochemical networks. Subfigure (a) shows the reaction equation system of an exemplary reaction cascade, subfigure (b) shows the hypergraph representation.

- Subfigure 3.3bb depicts how reaction $II.\ A \rightarrow C + E$ is included in the PN. It is worth mentioning that the PN needs only one token on A to create one token on each C and E.
- Subfigure 3.3c represents the integration of the reversible reaction $III.\ C \leftrightarrow E$ into the PN. As there is no possibility to insert reversible transitions in PN, hence two transitions, one for each direction of the reaction, have to be inserted. The two transitions are named $IIIa$ and $IIIb$ to ensure a unique name for each transition.
- Subfigure 3.3d shows the network after the inclusion of the reaction $IV.\ E \rightarrow D$.

All metabolic networks possess connections to the environment to model for example the uptake of food and the disposal of waste (in this context here 'waste' consists of substances which cannot be metabolized any further by the cell secreting it). To model such a consumption of a substance, *input transitions* (also called IN, or *Inputs*) are added to the Petri net. Inputs are transitions without pre-places and are therefore always enabled. Waste-disposal is modeled by *output transitions* (also called OUT, or *Outputs*), which are transitions without post-places. In this way, Inputs produce their products from no educts, providing them to the network, while Outputs produce no products from their educts, removing them from the modeled system. Note that in models which contains energy delivering agents like ATP, it is possible to model active transporters connecting the model to the environment. These Inputs can consume e.g. ATP (and/or other modeled metabolites) to produce the input substrate, providing a realistic model of active transport processes.



(a) Step one

(b) Step two

(c) Step three

(d) Step four

Figure 3.3: Stepwise creation of a Petri net from the exemplary network given in Figure 3.2. Subfigure (a) shows the inclusion of reaction I, (b) of reaction II, (c) of reaction III, and (d) of reaction IV. Note that the reversible reaction included in step III (subfigure c) needs to be split in two transitions, one for each direction. The names are adjusted to $IIIa$ and $IIIb$, to ensure a unique name for each transition.

3.4 Node-Degree and Power-Law Distribution

The *node-degree* k of a node $v \in V$ of a graph $G = (V, E)$ is the number of edges connected to v. Loops (i.e. edges for which the source and the target are the same node v) are counted twice. In directed graphs the node-degree splits into the *ingoing* (k_{in}) and the *outgoing* (k_{out}) node-degree, counting edges targeting or leaving the vertex v, respectively. The sum of k_{in} and k_{out} is the overall node-degree k of a vertex v.

We introduce the *power-law distribution* and the determination of the parameter α as mentioned in (Newman, 2005).

We speak of a *power-law distribution* of values, if the probability $p(x)$ for the appearance of a value x follows the rule

$$p(k) = Cx^{-\alpha}$$

where C and α are constants.

While the constant C is not of interest (after the determination of α, a value is assigned to C that fix the sum of the distribution to 1), the identification of the parameter α is important and is done by the formula

$$\alpha = 1 + \frac{n}{\sum_{i=1}^{n} ln(\frac{x_i}{x_{min}})}$$

Given a set of values x, n is the number of values $n = |x|$. $x_i \in x$ is the ith value and x_{min} is the minimal value of x.

3.5 Databases and Software

For the acquisition of data concerning pathway affiliation of reactions, we rely on the database *AraCyc* (Mueller et al., 2003; Zhang et al., 2005). AraCyc is a tool for the visualization of the biochemical pathways in *Arabidopsis thaliana*. It is based on *MetaCyc* (Caspi et al., 2012), part of the *Plant Metabolic Network* (Chae et al., 2012; Zhang et al., 2010), and the *BioCyc* (Caspi et al., 2012) database collection. In the beginning it was computationally predicted using the sequenced genome of *Arabidopsis thaliana* and afterwards it was manually validated. This curation process resulted in the removal of non-*arabidopsis* reactions and is a still on-going process. The latest version of AraCyc is version 8.0, released in April 2011 (http://www.arabidopsis.org/biocyc/). The latest data download for this work was performed on march 12th, 2012.

Modeling and presentation of the model is done with the software MONALISA (ML, Einloft et al., 2013) which is based on the TInA tool (Thormann et al., 2009). The ML toolkit features a powerful Petri net editor, allowing all needed interactions with objects of a PN, e.g. adding and removing places, transitions, and edges between them. Additionally, it allows coloring of nodes and edges either manually or accordingly to computed network decompositions. The ML software also includes methods for the analysis of PN like the invariant determination and their further investigation with additional methods like the formation of MCT-Sets and the determination of t-cluster.

For the programming we used the NetBeans Integrated Development Environment in

the versions 6.9.1 and 7.1 (http://www.netbeans.org) and Java in versions 1.6 and 1.7 (http://www.java.com). Some scripts are written in Python (http://www.python.org) in version 3.1. Pictures not done with ML are created with the Open Source vector graphics editor Inkscape (http://www.inkscape.org).

3.6 Kernighan-Lin Algorithm

The automatic network decomposition in Section 4.2 is performed by the *Kernighan-Lin Algorithm* (KL-algorithm) (Kernighan and Lin, 1970). The KL-algorithm is a heuristic method, which divides the set of nodes V of a graph $G = (V, E)$ into two subgraphs A and B of identical size and minimizes the edges crossing between the subgraphs. If $|V|$ is odd, the algorithm adds a dummy-node to V without any connections to the graph. This has no effect on the crossing edges.

Starting point of the algorithm is an arbitrary partition of V into A and B of equal size. For each node v the sum of connections to its own subgraph, the *inner costs* I_v, and the connections to its other subgraph, the *outer costs* O_v, are determined. For each node $a \in A$ is

$$I_a = \sum_{x \in A} c_{a,x}$$

$$O_a = \sum_{y \in B} c_{a,y}$$

where $c_{a,v}$ is the entry of the *cost matrix* C, containing the cost of the edge between the node a and the node v. This cost can be the weight of the edge between them, or just 1 or 0, depending if or not an edge is present between the two nodes. We use the presence of the edge, meaning $c_{a,v} = 1$ if $(a, v) \in E$ and $c_{a,v} = 0$ otherwise. The results of these two cost functions are used to calculate the difference between the two costs: $D_v = O_v - I_v$ for each node v.

The KL-algorithm is divided into *rounds* and *phases*. Each round contains several phases and the algorithm iterates over several rounds until its termination. In each round a set of nodes $X \subset A$ and $Y \subset B$ are identified with the help of the *gain function*. The gain function provides the possible gain of a swap of two nodes $a \in A$ and $b \in B$. It is calculated by $g = D_a + D_b - 2c_{a,b}$. A phase provides values for g by repeatedly performing four steps:

1. $p = 1, A_p = A, B_p = B$
2. select $a \in A_p$ and $b \in B_p$ with a maximal $g_p = D_a + D_b - 2c_{a,b}$.
3. set $A_{p+1} = A_p \setminus \{a\}$, $B_{p+1} = B_p \setminus \{b\}$ and $X \cup \{a\}$, $Y \cup \{b\}$
4. $p = p + 1$, update values for D_v for each node v
5. repeat 1. - 4. until $p = |V|/2$
6. return $G = \{g_1, \ldots, g_{|V|/2}\}$, X, Y

Each round of the KL-algorithm includes the starting of a phase. A phase stops after $|V|/2$ steps and a round terminates, if a certain quality condition is met. A round is composed of three repeated steps:

1. $G, X, Y =$ returnvalues of phase(A,B)
2. find k, so that $S = \sum_{l=1}^{k} g_l$ is maximal
3. if $S > 0$, move $x_1, \ldots, x_k$ to B and $y_1, \ldots, y_k$ to A
4. repeat 1. - 3. until $S \leq 0$
5. return A and B

After the termination of the algorithm the two subsets of nodes A and B are connected via a minimal count of edges. Note that this minimum in the solution space can be a local minimum. To provide a more reliable solution the KL-algorithm can be started with different starting assignments of nodes to A and B to sample a larger part of the solution space.

Chapter 4

Results and Discussion

4.1 The *Arabidopsis thaliana* Petri Net Model

Petri nets have been shown as a valuable method to represent biochemical networks e.g. for the analysis of the sucrose breakdown pathway in potato (*Solanum tuberosum*) tuber (Koch et al., 2005) and the stochastic modeling of the yeast (*Saccharomyces cerevisiae*) metabolism (Mura and Csikász-Nagy, 2008).

PN are favored for their sound analysis methods and their intuitive network representation. In this section we illustrate the creation of our Petri net model, representing the main metabolism of *Arabidopsis thaliana*. The modeled reactions are inspired by our experimental coworkers (personal communication, Schleiff, 2010). The complete model originates from extensive literature search.

A comparable network was developed before (Nöthen, 2009). This network was around 20% smaller than the PN presented here and consisted of 113 metabolites, 208 reactions, and 490 edges. The complete network was checked again in the literature and various reactions were changed or deleted. Combined with several new insights into the metabolism of *Arabidopsis thaliana* in the last years, which entail the possibility for considerable extensions, this renders a reconstruction of the network suitable.

The presented network consists of 134 metabolites and 243 reactions which are connected via 572 edges. Because long names are difficult to place inside of a dense network model, the metabolites are numbered. The assignment of the numbers of the metabolites to their names is listed in Table B.1. The reactions are named a combination of letters and numbers. A reaction name is represented by 'E_x', where x stands for a number. The number is random and does not incorporate any information about the connected metabolites.

Some of the enzymes modeled by the reactions catalyze more than one reaction. In these cases the number of one unique metabolite these reactions connect is added to the name of the transition, e.g. 'E_1_44' is the transition of 'E_1' which is connected to the metabolite '44' (β-D-fructose 6-phosphate). This rule is not applied for the reaction 'E129' (ribulose-1,5-biphosphate carboxylase oxygenase), which is split into the two transitions 'E129_1' and 'E129_2'. The names of the reactions of the starch metabolism follow another rule described in Section 4.1.2. Table B.2

lists the reactions, their pathway affiliations, the enzymes they are encoding, and the respective literature.

4.1.1 General Properties of the Model

For convenience, we optically divide the PN into four subnetworks: The *Sucrose subnet*, the *Citrate subnet*, the *Shikimate subnet*, and the *UTP subnet*, which are introduced in detail in the following subsections. Note that partitions of topologically connected parts, modulated with the help of logical nodes, are called *subnets* or *subnetworks*, while a partition by cutting edges results in different *modules*.

Figure 4.1 gives an overview of the complete network. Each of the four subnetworks is indicated by a different color, and the names of the subnets are chosen according to metabolites playing a key role in it. The four subnetworks are shown in more detail in Figures 4.4, 4.7, 4.8, and 4.9.

4.1.1.1 Environmental Connection

We describe the connection of the model to the environment by 29 different reactions. Eight of these connections are attached to the four external metabolites of the network, whereas the remaining 21 supply or remove internal metabolites. The four external metabolites are coenzyme A (CoA, metabolite '93'), acetyl-CoA (metabolite '92'), ammonia (metabolite '29'), and citrulline (metabolite '94'), which are all part of the citrate module (see Figure 4.7).

While CoA, acetyl-CoA and ammonia are known as co-factors with many connections inside the metabolism, citrulline is proved to be part of a dead-lock in early modeling stages. Pantothenate, which is a precursor to CoA (Raman and Rathinasabapathi, 2004), is not modeled in the network. Due to the multiple pathways which need CoA and as we do not model the biosynthesis of CoA, it has to be treated as external metabolite. The acetylated form of CoA, acetyl-CoA, is found as product of the β-oxidation of fatty acids in *Arabidopsis thaliana* (Fulda et al., 2002) which justifies the external setting. Finally, ammonia is supposed to be imported at the mycorrhizal interface (Chalot et al., 2006), suggesting its external setting. This symbiotic association between roots and fungi is assumed to exist for most plants under natural conditions.

Citrulline is part of a dead-lock, made from metabolites '94' (citrulline), '27' (L-argininosuccinate), '28' (arginine), and '1' (ornithine). It has been demonstrated that citrulline is sufficient as the sole source of nitrogen for the effective growth of plants (Ludwig, 1993). This behavior suggests the possibility to transport citrulline. The supply of citrulline proved to be sufficient to dissolve the dead-lock.

In PN external metabolites can be treated in different ways (Section 3.1 and Zevedei-Oancea and Schuster, 2003). We choose a method which connects each external metabolite with an input transition supplying that substance, and an output transition removing it (Starke, 1990). A total number of 3 IN and 18 OUT connects the internal metabolites to the environment.

Contrary to all other reactions, the numbers in the names of the Inputs and Outputs yield the information about the connected metabolite. In our network the supplying transitions are called

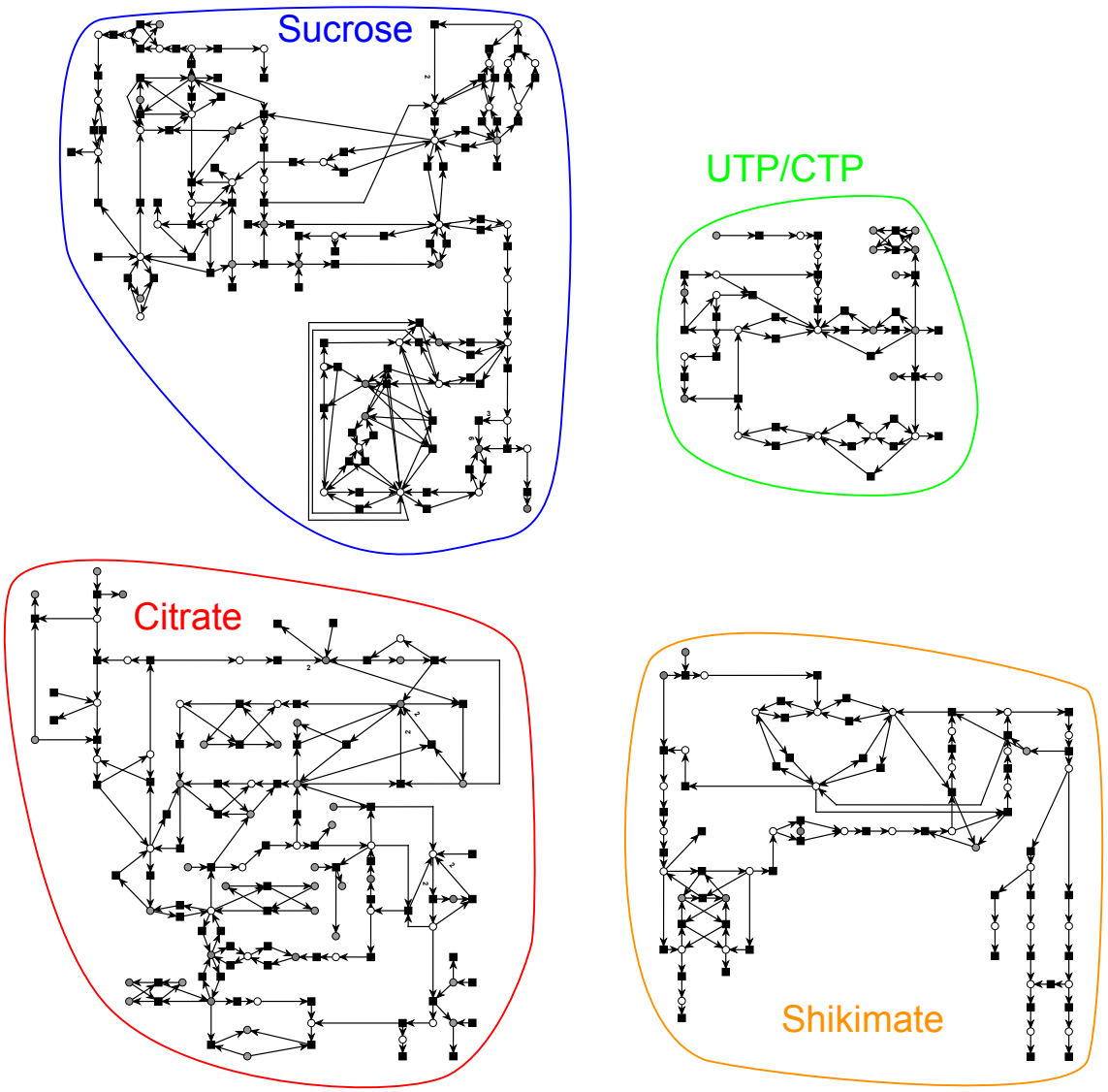


Figure 4.1: PN model of the metabolism of *Arabidopsis thaliana*. Circles are places encoding metabolites and squares are transitions encoding reactions. Grey circles denote logical places (some places are of a darker gray than others, this is caused by internal treatment of logical places in ML, and has no effects on the meaning of the place). The network consists of 134 metabolites and 243 reactions which are connected via 572 edges. Each of the subnetworks are displayed in separate figures in more detail (see Figures 4.4, 4.7, 4.8, 4.9).

OUT_xx, where xx is the number of the respective supplied metabolites. The removing reactions are named OUT_xx, where xx stands for the number of the respective removed metabolite. A list of the metabolite numbers can be found in Table B.1 and the IN and OUT can be found at the end of Table B.2.

Beside the four IN supplying the external metabolites, there are three IN added to the network model providing internal metabolites:

- *Sucrose subnet*: D-fructose (metabolite '63') and D-galactose (metabolite '73')
- *Citrate subnet*: glycine (metabolite '18').

In addition to the four OUT of the external metabolites, there are 22 OUT removing internal metabolites, which are distributed over all four subnets:

- *Citrate subnet*: UDP-glucose (metabolite '82') and cysteine (metabolite '113')
- *Sucrose subnet*: eight OUT eliminate α-D-glucose 1-phosphate (metabolite '50'), β-D-glucose (metabolite '51'), sorbitol (metabolite '62'), sucrose (metabolitece '66'), stachyose (metabolite '71'), D-*myo*-inositol (metabolite '74'), L-galactono-1,4-lactone (metabolite '78'), and UDP-xylose (metabolite '88').
- *Shikimate subnet*: two output reactions for lignin (metabolites '30a' and '30b'), and one each for homogentisate (metabolite '99'), chorismate (metabolite '103'), and 4-hydroxyphenylpyrovate (metabolite '106').
- *UTP subnet*: uridine diphosphate (UDP, metabolite '115'), cytidine triphosphate (metabolite '122'), and uridine triphosphate (UTP, metabolite '129').

Summarizing, the network consumes 7 substrates from the environment to produce 22 metabolites. Hereby the system increases the diversity of metabolites in the environment.

4.1.1.2 Small Metabolites

Catalytic metabolites, cofactors like NAD/NADH, and metabolites which provide energy to reactions like ATP are often treated as external (Zevedei-Oancea and Schuster, 2003). In our network the great majority of these metabolites is omitted, which is a common treatment of these substances (Baldan et al., 2010). For a comparable network the removal of metabolites, e.g., carbon dioxide, has a considerable impact on the number of t-invariants without changing the CTI-property (Nöthen, 2009). The metabolites omitted from our model are presented in Table 4.1. A complete list of the possible connections of these metabolites to the network is given in Table C.3.

Table 4.1: Omitted metabolites of the network. For each of these metabolites the number of possibly connected reactions is given in the second column.

metabolite	connected reactions
H^+	65
H_2O	44
NAD(P)H	27
NAD(P)	27
ATP	24
phosphate	20
ADP	17
CO_2	16
diphosphate	9
AMP	6
O_2	5
H_2O_2	2
bicarbonate	2
S-adenosyl-L-methionine	2
S-adenosyl-L-homocysteine	2
Tetrahydrofolate	2
5,10-methylene-tetrahydrofolate	2
e^-	1
hydrogen sulfide	1

To justify the procedure of metabolite removal, we perform a node-degree analysis of our network compared to the complete AraCyc database. AraCyc contains 1884 reactions and 1797 compounds (12th march 2012). The leading opinion is that the distribution of the node-degree of metabolites in biological networks follows a power-law (Jeong et al., 2000), albeit this issue is still under discussion (Lima-Mendez and van Helden, 2009; Laherrère and Sornette, 1998).

Figure 4.2 shows the distribution of node-degrees of the metabolites in AraCyc. The vertical axis lists the absolute node-degrees, while the horizontal axis indicates the percentage either of the fraction of how many metabolites in the database have this node-degree (gray bar), or in how many reactions they take part in (white bar). E.g. Hydrogen (or its ion, the proton), having the highest node-degree of over 900 (which resembles the number of reactions it takes place in), is part of nearly 50% of the reactions.

An evaluation of the node-degree of the metabolites listed in AraCyc as shown in Figure 4.2 demonstrates that most of the omitted metabolites of our network have a high node-degree in AraCyc. This is consistent with the observation presented in Table 4.1, which suggests a high node-degree of the omitted metabolites as well. A removal of these hub-metabolites (Jeong et al., 2000) renders an analysis of the node-degree distribution, in order to detect a power-law distribu-

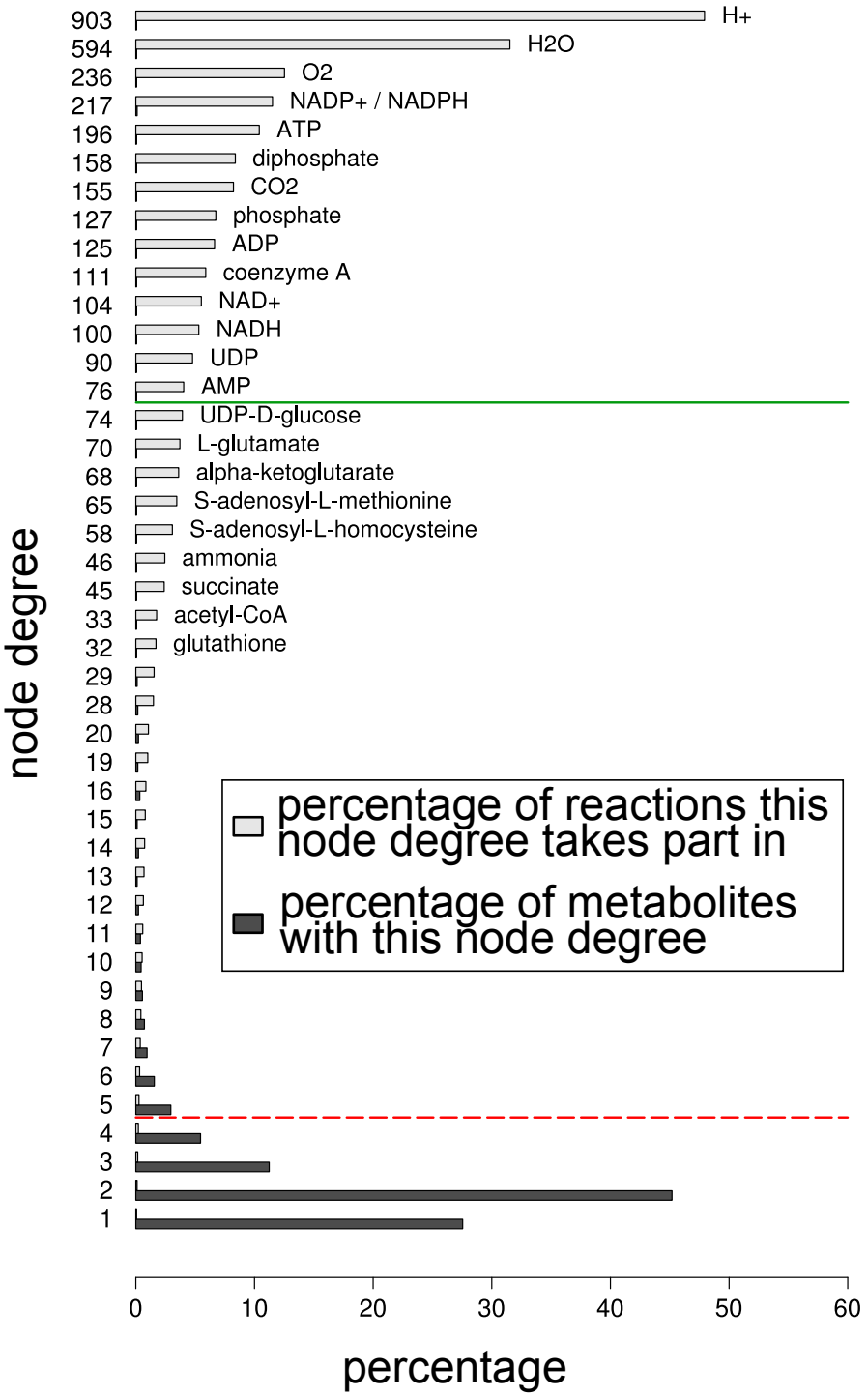


Figure 4.2: The distribution of node-degrees of compounds in *Arabidopsis thaliana* as given in AraCyc. The vertical axis displays the absolute node-degree, the horizontal axis indicates a percentage either for the fraction of reactions this compound take place in (white bar), or the fraction of metabolites exhibiting this node-degree (gray bar). The green line indicates a node-degree of 75, the red dashed line indicates the average node-degree of 4.7 in the AraCyc database.

tion, useless. The ordering of a list of metabolites accordingly to their node-degree can be seen as an ordering of importance of these metabolites (Wagner and Fell, 2001). Investigations of the metabolic network of *Escherichia coli* suggest an important role of several substances which seem, following the node-degree-importance paradigm, to play an important role in the PN as well. Examples of these compounds are glutamate, succinate, acetyl-CoA, and α-ketoglutarate (Wagner and Fell, 2001).

The green line in Figure 4.2 indicates the node-degree of 75 representing our cut-off value. To limit the overall complexity of the model, most of the metabolites with a higher node-degree than this threshold are omitted from the network. The value of 75 removes cofactors like AMP, whereas ammonia-shuttles like glutamate remain in the network.

Different to this cut-off treatment, we include and remove a few compounds into/from the network:

- UDP and coenzyme A remain in the network.
- S-adenosyl-L-methionine and S-adenosyl-L-homocysteine are removed from the network.

These exceptions are inspired by measurements of our coworkers suggesting a modeling of these substances (personal communication Schleiff, E.). The two compounds remaining in the network are treated differently. The cofactor coenzyme A is modeled as external, i.e. it is connected to an IN and an OUT. UDP is only connected to an OUT, but the *de-novo* synthesis of UMP is part of the model and can provide UDP.

It has been shown that the conservation of metabolites depends more on the connection they provide between modules than on their hub-character (Guimera and Amaral, 2005), suggesting that the importance determined with the help of the node-degree of metabolites is less accurate as considered before.

To further justify the removal of these metabolites, we investigate the node-degree of the reactions of our network compared to the node-degree of reactions in AraCyc. Figure 4.3 presents the distribution of reaction node-degrees. It displays the percentage of the reactions with a certain node-degree of our network model (circles), of the AraCyc database (squares), and of the downscaled AraCyc database (triangles). Downscaled means the deletion of all metabolites with a node-degree of greater then or equal to 75. While the distributions of the reaction degrees of the PN model and the AraCyc data are clearly not related, this changes significantly if we remove all secondary metabolites from the data obtained from AraCyc, suggesting a comparable network structure.

Summarizing, an investigation of the node-degrees of the metabolites and of the reactions is combined and seem to justify the removal of several metabolites from our network.

4.1.2 Sucrose Subnetwork

The *Sucrose subnet* is presented in Figure 4.4. The picture highlights four major pathways:

- The blue path marks the Calvin cycle, which is the main carbon fixation pathway in plants.

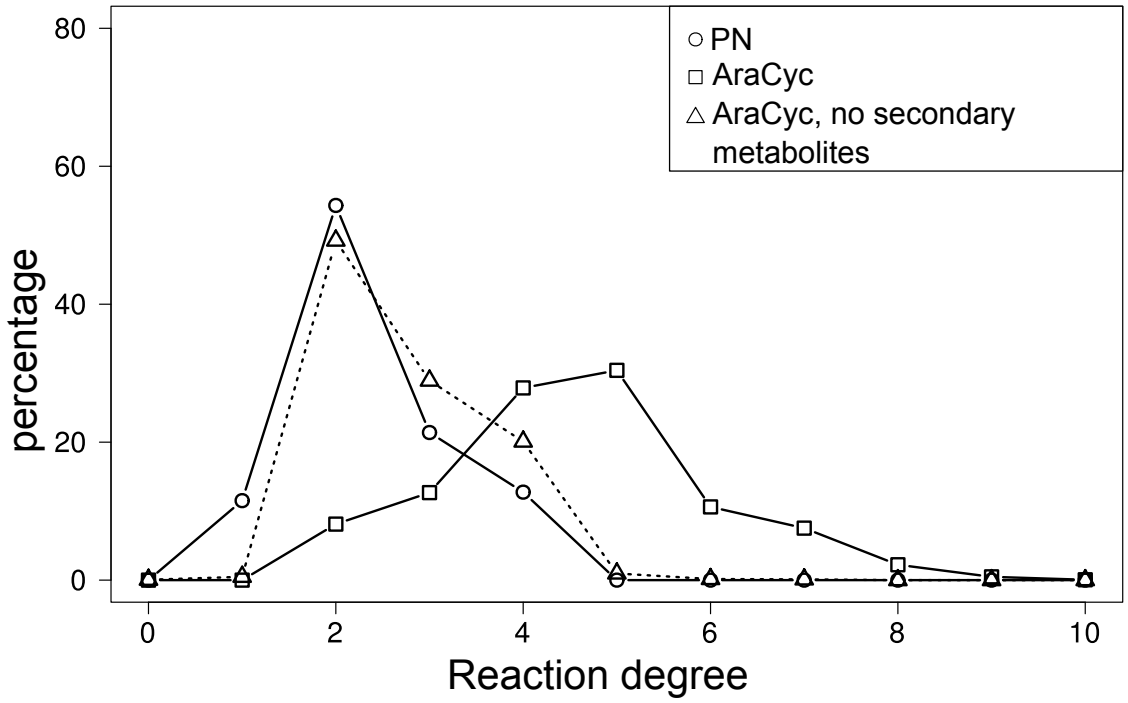


Figure 4.3: The distribution of node-degrees of reactions in different networks. The vertical axis displays the percentage of all reactions exhibiting this reaction degree, and the horizontal axis displays the corresponding reaction degree. Circles indicate the distribution of node-degrees of reactions in our PN model, while squares and triangles depict the node-degrees of the reaction listed in the AraCyc database. We removed all metabolites with a node-degree of 75 or more from the data obtained from AraCyc resulting in a shifted distribution (triangles).

- The sugar metabolism is indicated by the green box. It includes the synthesis and degradation of sucrose and UDP-glucose.

- The orange path indicates the glycolysis. In the picture of the network, the glycolysis takes place at several locations. This is a consequence of logical places as a number of metabolites, e.g., β-D-fructose 6-phosphate (metabolite '44'), occur multiple times in the image of the subnet (compare Section 3.1). In the underlying network topology the glycolysis is a connected pathway. As in general the glycolysis is the reaction cascade leading from β-D-glucose (metabolite '51') to pyruvate (metabolite '20'), only the part of the reaction chain leading from β-D-glucose (metabolite '51') to glycerate 3-phosphate (metabolite '40') is modeled in the *Sucrose subnet*. The remaining reactions of the glycolysis are placed in the *Citrate subnet*.

- The starch (metabolite '59') synthesis and degradation is highlighted by the red circle.

We put focus on a special part of the *Sucrose subnet*, the starch metabolism, to exemplarily show how we use the literature to model our network. The starch metabolism is indicated in red in Figure 4.4, and shown in detail in Figure 4.6. Starch is a polymer of sugar molecules. For modeling purposes utilizing Petri nets this means starch has to be present to be produced. In this context 'synthesis' and 'degradation' means the addition or removal of single sugar molecules to or from this polymer (Buchanan et al., 2000). As also the initiation of starch is still under discussion (Szydlowski et al., 2009), a suitable modeling of starch biosynthesis and degradation is still an issue.

As the start of starch synthesis is still unknown, a model as shown in Figure 4.5a is not suitable, due to the following problem. To provide a constant availability of starch, the model must not run out of starch. If starch loses all of its tokens in this model, the system will be stuck in a deadlock (see Section 3.1). That is, because the transition *Starch synthesis*, which is the only source of starch, cannot be enabled again. Additionally, the transition *Starch synthesis* would be connected to starch with two edges of oppositional directions, resulting in a non-pure PN.

Starch gets special attention in our model due to the mentioned reasons, and the reactions in the starch pathway follow a different nomenclature. The reaction names begin with *R* instead of *E*.

4.1.2.1 Starch Synthesis

Our detailed model of starch and its surroundings is presented in Figure 4.6. We decide to model starch and its synthesis analog to (Kossmann and Lloyd, 2000) in the following steps:

1. Starch synthesis starts from α-D-glucose 1-phosphate (metabolite '50') which is converted (reaction 'E46') to ADP-Glucose (metabolite '58').

2. ADP-Glucose is converted to amylose (metabolite '86') and amylopectin (metabolite '81'), by the reactions 'R5_1' and 'R5_2', respectively.

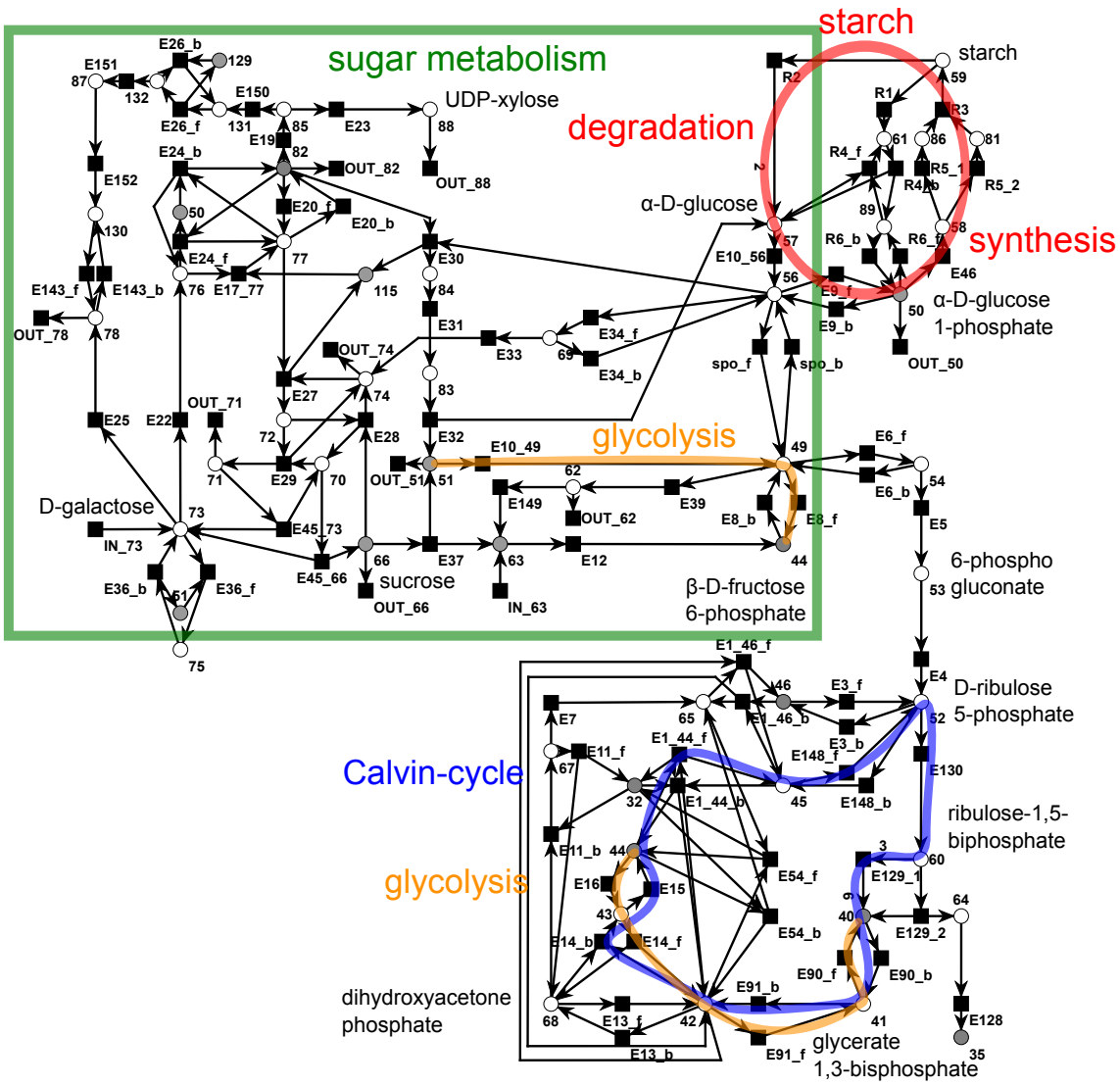
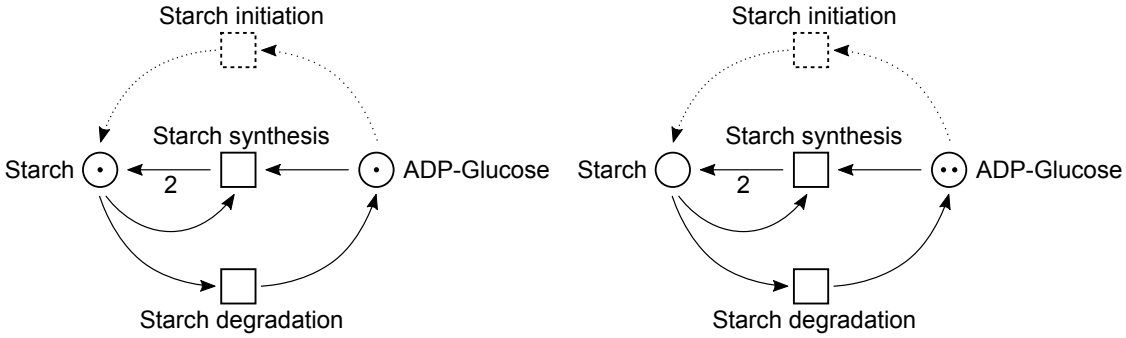


Figure 4.4: *Sucrose subnet*. Part of the network are the Calvin cycle (blue), the beginning part of glycolysis (orange, β-D-glucose (metabolite '51') $\rightarrow$ ... $\rightarrow$ glycerate 3-phosphate (metabolite '40')), the sugar metabolism (green box), and starch (metabolite '59') synthesis/degradation (red circle).



(a) transition "Starch synthesis" is live (b) transition "Starch synthesis" is dead

Figure 4.5: Simplified starch biosynthesis/degradation modeled by a PN. Starch is a multi-sugar molecule needing itself for its synthesis. The net is not pure because the pre-place for starch synthesis is also its post-place. Without the dashed parts, the place *Starch* is a structural deadlock. In Subfigure 4.5a the transition *Starch synthesis* is active. The situation after firing of the transition *Starch degradation* is shown in Subfigure 4.5b. The transition *Starch synthesis* is dead, and subsequently is the transition *Starch degradation*. The question of initiation of starch synthesis is still under discussion (Szydlowski et al., 2009). Therefore it is not suitable to model starch synthesis in this way. We decide to model the starch biosynthesis straightforward, as described in Section 4.1.2 and Figure 4.6. This results in a pure network, i.e., there are no backward edges as in Subfigure (a) between the transition *Starch synthesis* and the place *Starch*.

3. Amylose and amylopectin are combined to one starch (metabolite '59') molecule by the reaction 'R3'.

4.1.2.2 Starch degradation

The degradation of starch requires some additional fine-tuning, to ensure its thermodynamic feasibility. That means there must not be any substance creation, i.e., there must be no starch molecule which needs one sugar molecule for its synthesis, but produces two of this sugar molecules when it is degraded. Comparable adaptions in the starch metabolism have been made in the Poolman model (Poolman et al., 2009) providing a thermodynamically feasible model. In our PN, starch degradation is modeled in two different ways. First via the Glucan, water dikinase (modeled in reaction 'R2') and second via the Phosphoglucan, water dikinase (modeled in reaction 'R1'). The starch degradation using reaction 'R2' results in the following steps:

1. Starch breaks apart into two molecules of α-D-glucose (metabolite '57') via reaction 'R2' (Fettke et al., 2009). The amount of α-D-glucose produced by this reaction is adjusted in such a way that a molecule of starch needs as many sugar molecules (it needs two of them, one for amylose and one for amylopectin, which combine to one starch) to be built as it produces if it falls apart.

2. α-D-glucose can be phosphorylated via hexokinase (reaction 'E10_56') to α-D-glucose 6-phosphate (metabolite '56') (Lu and Sharkey, 2006; Guy et al., 2008).

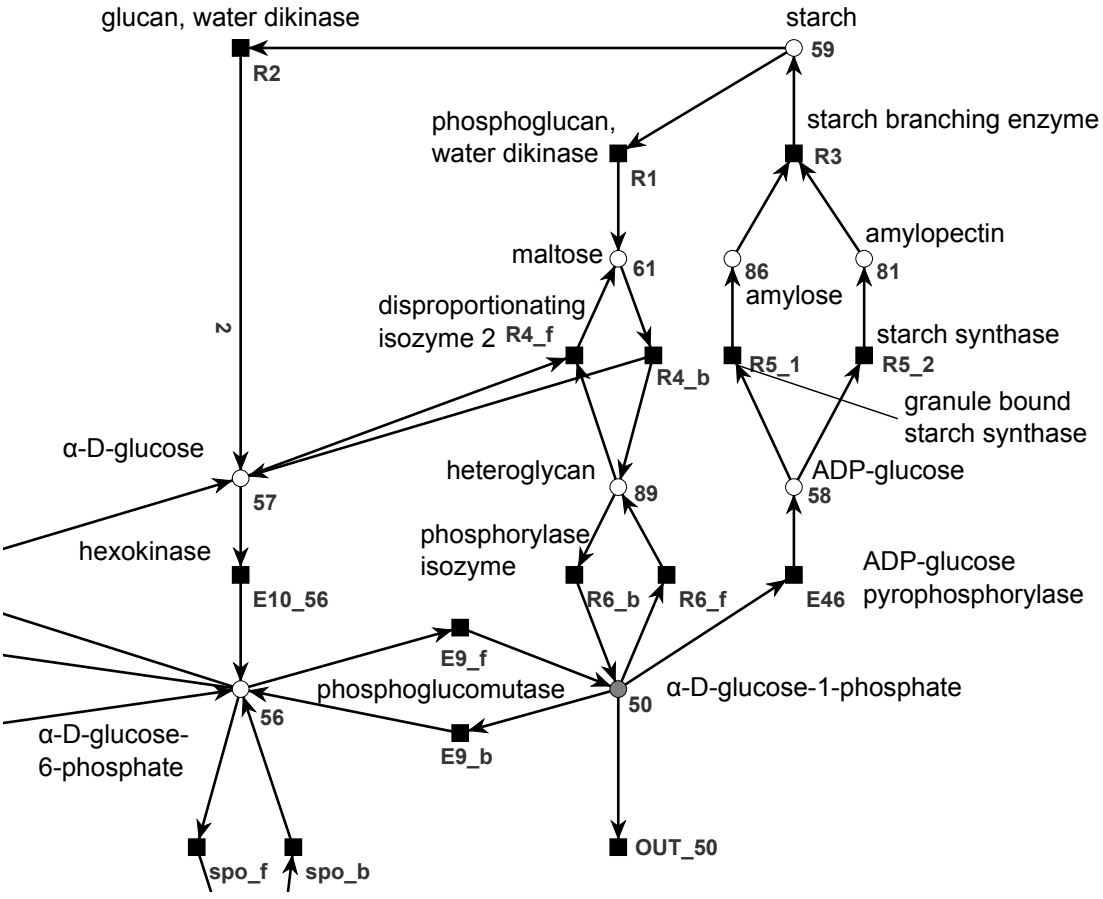


Figure 4.6: Starch synthesis and degradation in the *Arabidopsis thaliana* PN. Starch is synthesized from amylopectin and amylose (in reaction 'R3') and degraded to maltose (in reaction 'R1') or α-D-glucose (in reaction 'R2').

3. α-D-glucose 6-phosphate can be converted to α-D-glucose 1-phosphate (metabolite '50') by reaction 'E9' (Reiter, 2008), and by this conversion restore the substrate of the starch synthesis as mentioned in Section 4.1.2.1 step 1.

The second pathway of starch degradation via the enzyme Phosphoglucan, water dikinase (reaction 'R1') is modeled as follows:

I. Starch is degraded into maltose (metabolite '61') by reaction 'R1' (Guy et al., 2008).

II. Maltose is further split into heteroglycan (metabolite '89') and α-D-glucose by the reaction 'R4' (Fettke et al., 2009). The further reactions concerning α-D-glucose can be found in step 2 of the explanation of reaction 'R2' above.

III. Heteroglycan can be converted to α-D-glucose 1-phosphate by the reaction 'R6' (Fettke et al., 2009). This conversion restores the substrate of the starch synthesis mentioned in Section 4.1.2.1 step 1.

Note that in both degradation pathways the total number of possibly produced α-D-glucose 1-phosphate molecules per starch is two, and by that the number of α-D-glucose 1-phosphate molecules needed to produce one starch molecules has also to be two (see Section 4.1.2.1). The first degradation pathway produces two α-D-glucose molecules from one starch molecule in step 1, which can be subsequently converted to α-D-glucose 1-phosphate. The second degradation pathway produces one molecule heteroglycan and one molecule α-D-glucose from one molecule maltose in step 2. Both of them can further react to α-D-glucose 1-phosphate.

4.1.3 Citrate Subnetwork

Figure 4.7 portrays the *Citrate subnet*. In the *Citrate subnet* five major pathways are marked:

- The biosyntheses of glutamine (metabolite '33') and glutamate (metabolite '2') are illustrated in the blue box.
- The red line marks the citric acid cycle, which is a main part of the energy metabolism in aerobic species.
- The green path highlights the glyoxylate cycle (glyoxylate: metabolite '13') which is involved in the synthesis of carbohydrates.
- The pink box indicates the uridine 5'-phosphate biosynthesis. The *Citrate subnet* incorporates only a part of this synthesis, the reaction chain from bicarbonate to carbamoyl aspartate (metabolite '117') whereas the remaining reactions are part of the *UTP subnet*. Bicarbonate is a small molecule and is not explicitly modeled in the network.

4.1.4 Shikimate Subnetwork

The *Shikimate subnet* is displayed in Figure 4.8. It consists of three main pathways:

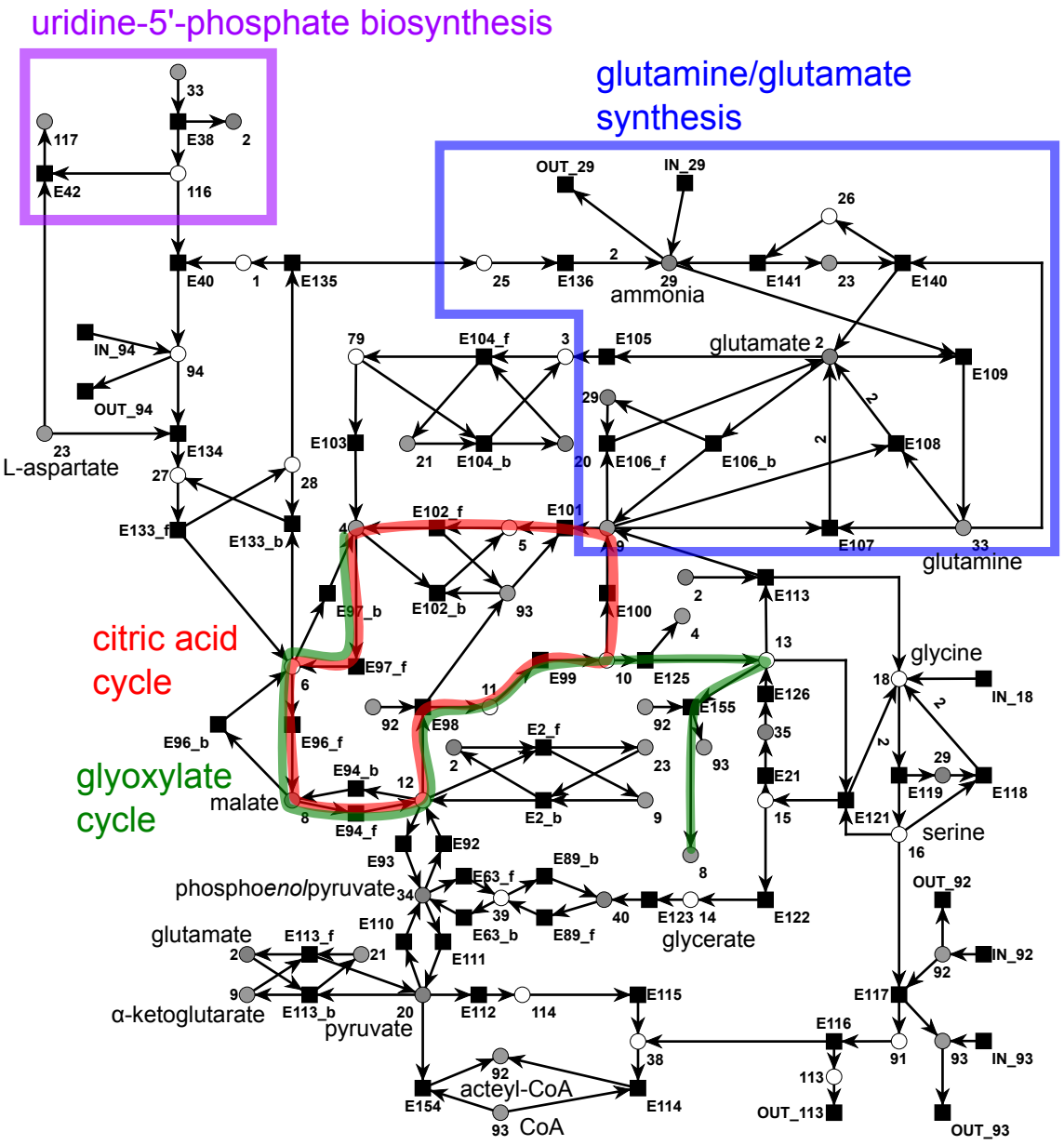


Figure 4.7: *Citrate subnet*. Part of the network is the citric acid cycle (red), the glyoxylate cycle (green), the start of uridine 5'-phosphate biosynthesis (purple box), glutamate/glutamine biosynthesis (blue box), and the final part of glycolysis (orange, glycerate 3-phosphate → ... → acetyl-CoA).

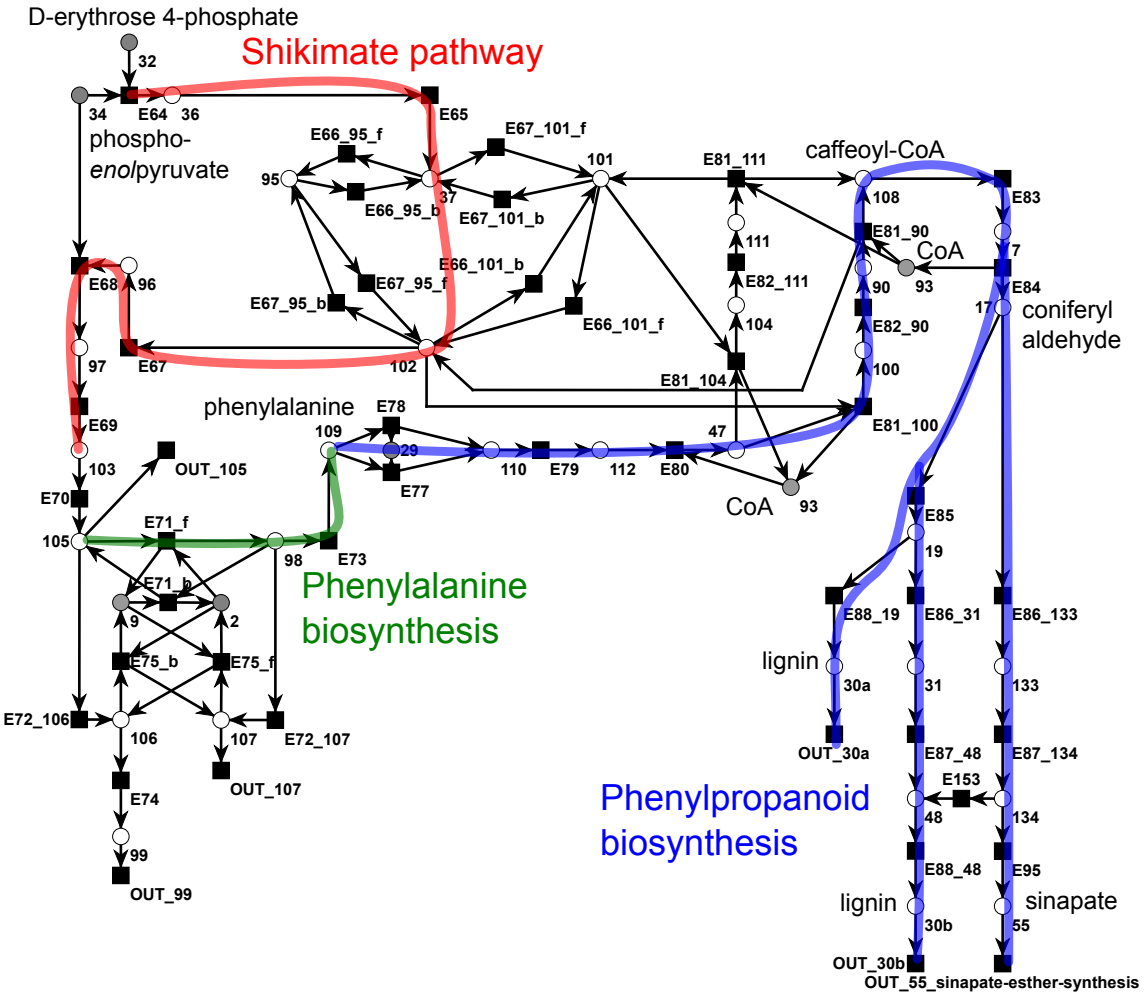


Figure 4.8: *Shikimate subnet*. Part of the network is the shikimate pathway (red), the phenylalanine biosynthesis (green) and the phenylpropanoid biosynthesis (blue).

- The red path highlights the shikimate (metabolite '102') synthesis. Shikimate is an important precursor to a group of aromatic amino acids, including phenylalanine (metabolite '109') (Herrmann and Weaver, 1999).
- The synthesis of phenylalanine is indicated by the green color.
- The blue marked phenylpropanoid pathway starts from phenylalanine. Phenylpropanoids are precursor to lignin (metabolite '30a' & '30b') synthesis.

The synthesis of lignin is e.g. explicitly included as a part of the biomass function in a genome scale flux balance model of maize (*Zea mays*) (Saha et al., 2011). We divide lignin into two compounds '30a' and '30b', representing guaiacyl lignin and syringyl lignin, respectively. Guaiacyl lignin is believed to be synthesized from coniferyl alcohol (metabolite '19') while syringyl lignin is produced from sinapyl alcohol (metabolite '48') (Humphreys and Chapple, 2002).

4.1.5 UTP Subnetwork

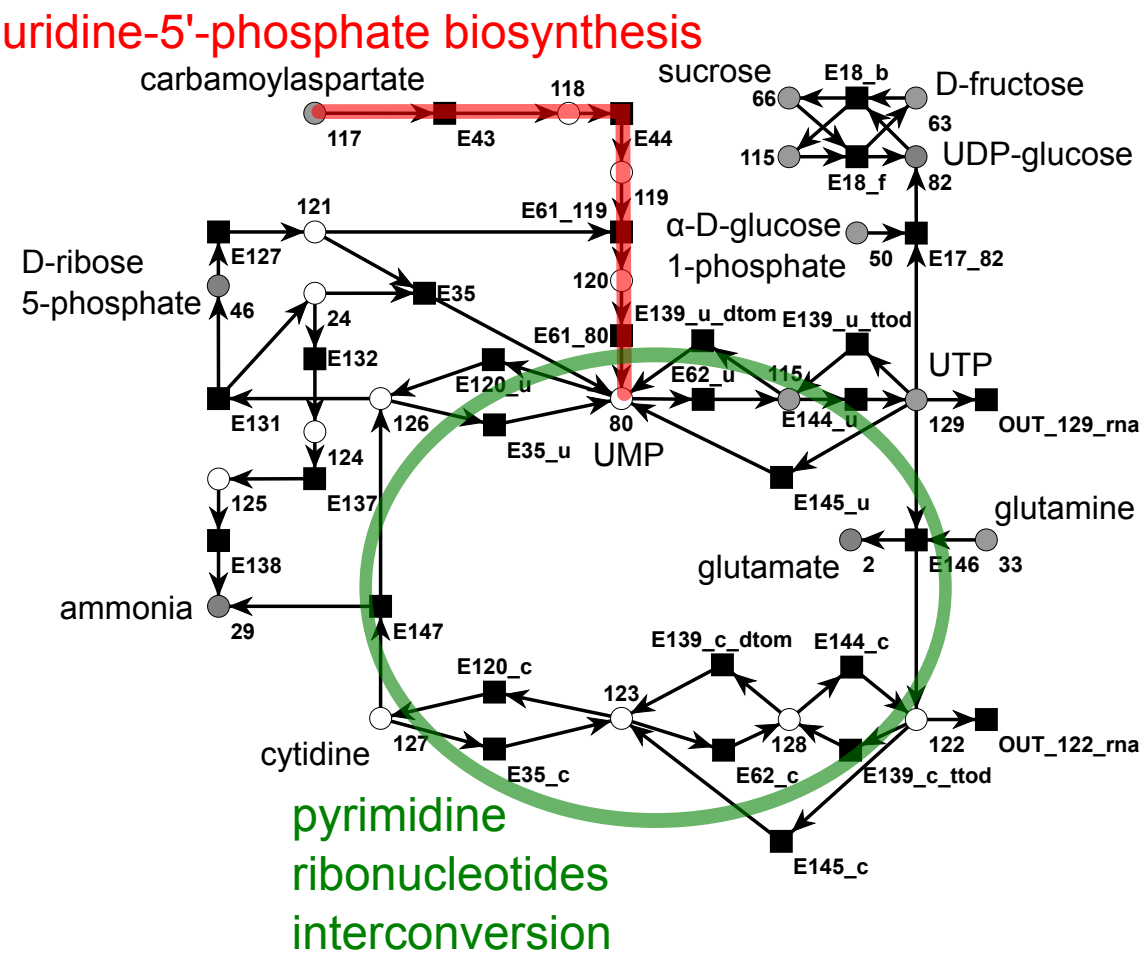


Figure 4.9: *UTP subnet*. Part of the network are the final reactions of the uridine 5'-phosphate (metabolite 80') synthesis (red) and the interconversion of uridine mono-, di- and triphosphate to cytidine mono-, di- and triphosphate (green cirlce) and vice versa.

Figure 4.9 shows the *UTP subnet* with two main pathways:

- The reaction cascade leading to uridine 5'-phosphate is marked in red. In the *UTP subnet* only the part of uridine 5'-phosphate synthesis is modeled which is missing in the *Citrate subnet*. The cascade in the *UTP subnet* spans the reactions from N-carbamoyl-L-phosphate (metabolite '117') to uridine 5'-phosphate (UMP, metabolite '80'). N-carbamoyl-L-phosphate is synthesized in the *Citrate subnet*, which contains the synthesis pathway of N-carbamoyl-L-phosphate starting from bicarbonate.

- The pathway of possible interconversions of pyrimidine ribonucleotides is highlighted in green.

4.1.6 Comparison to Models in the Literature

Due to the lack of usage of standardized formats a comparison of network models is complicated (Radrich et al., 2010).

Although it is difficult to compare our network to other networks of a considerably larger size, we want to stick to networks modulated for *Arabidopsis thaliana*. Other networks of comparable size to our PN which are built on the literature as well exist for barley (Grafahrend-Belau et al., 2009). This network consists of 234 metabolites and 257 reactions, of which 65 are transport reactions.

Table 4.2: Properties of different metabolic networks. All submodels of the Radrich network are listed: (a) core model, (b) intermediate model, (c) complete model.

	our PN model	Poolman model	AraGEM	Radrich model	Mintz-Oron model
data source	literature	AraCyc	KEGG	KEGG & AraCyc	KEGG & AraCyc
metabolites	134	1253	1748	914 (a) 1248 (b) 2328 (c)	1078
reactions	243	1406	1567	753 (a) 1388 (b) 2315 (c)	1363
compartments	no	no	yes	no	yes

Several quantitative models concerning the metabolic network of *Arabidopsis thaliana* have been published during the last years (Poolman et al., 2009; Gomes de Oliveira Dal'Molin et al., 2010; Radrich et al., 2010; Mintz-Oron et al., 2012). These models differ from each other in various properties (compare Section 2.1.3 for an introduction to these networks). The Poolman network sets H^+ and water to external, and input reactions provide NO_3, ammonia, SO_4^{2-}, glucose and phosphate. In the Radrich model H^+ is removed because of stoichiometric inconsistencies and AraGEM sets all external and input metabolites of the Poolman model as well as CO_2, O_2,

hydrogen sulfide, PO_4^{3-}, sucrose, fructose, maltose, glutamine, glutamate, asparagine, alanine, and serine to external metabolites. Our model contains Inputs for fructose, glycine, and galactose, additional to the four external metabolites CoA, acetyl-CoA, ammonia, and citrulline. In the PN model all the small compounds are removed.

Table 4.2 provides an overview about the sizes, data sources, and compartmentations of the different network models. Our qualitative PN model is significantly smaller than the other four quantitative models, and it is completely built on literature. The Poolman model and AraGEM are manually curated with the help of literature after an automatic extraction from the AraCyc and KEGG database, respectively. In contrast the Radrich model is based upon the idea of confidence levels reached by a comparison of different databases. The Mintz-Oron model follows the idea of automatic network reconstruction supported by a gap-filling algorithm, as partly done for AraGEM as well.

Compartments are included by AraGEM and the Mintz-Oron model. AraGEM relies on manually curated compartmentation data, whereas the Mintz-Oron model uses the SUBA database, in which they found 49% of the compartmentation of their reactions proven by experimental results. All other models including our PN model do not take compartments into account.

Validation of the models is mostly provided by the comparison of predicted flux data with measured data (Mintz-Oron, Poolman) or with data from the literature (AraGEM). As mentioned before the Radrich model discusses different levels of confidence emerging from the comparison of two databases. Our model is completely manually curated and literature covers every reaction. Additionally, we provide a pathway prediction and compare these predicted pathways to the literature in Section 4.4. This procedure is comparable to the flux predictions and their comparisons to the literature/measured data which is used to validate the quantitative models. Summarizing, our modeling approach seems to be well comparable to the established models.

4.2 Network Decomposition

Although our approach is more related to the graph partitioning, we use the terminus decomposition in our work, because it was used before in the analysis of (metabolic) networks (Ravasz et al., 2002; Holme et al., 2003; Ma et al., 2004; Guimera and Amaral, 2005; Schuster et al., 2002b; Zaitsev, 2004; Berthelot, 1987). Decomposition is a more widely used term, known as well for methods on discrete structures like linear equation systems (Mohring and Radermacher, 1984). Even though definitions of *graph splits* and *graph decompositions* (Charbit et al., 2012) exist, which describe certain connectivity properties concerning the generated subgraphs, we will not apply the meaning of these definitions, if we speak of splits or decompositions. A split, decomposition, or partition here means a distribution of the nodes of the network over two subnetworks.

We follow the idea that a division of the network in two subnets of identical size is the best opportunity to keep the number of sub-models as small as possible, while keeping the models as meaningful as possible. The network partition of our PN into two subnets is performed in two different ways.

First, we split our network into two subnetworks manually following the human intuition, the

definitions of certain biological pathways, e.g. Calvin cycle and citric acid cycle (Buchanan et al., 2000), and knowledge about compartmentation.

Second, we create a new automatic partition of the network to eradicate a possible bias caused by erroneous intuition in the biologically driven partition. Therefor we implement the *Kernighan-Lin algorithm* (KL-algorithm) (Kernighan and Lin, 1970). There are other algorithms for network partitioning, e.g., algorithms following the *Max Flow-Min Cut theorem*, which has been developed by two different groups in parallel (Ford and Fulkerson, 1956; Elias et al., 1956), methods based on the node-degree (Schuster et al., 2002b), and algorithms developed especially for PN partitioning following property-conserving strategies (Zaitsev, 2004). We choose the KL-algorithm for the possibility to directly constrain the size and number of the subgraphs, which is not possible by applying the other mentioned methods.

4.2.1 Biologically Driven Decomposition

We divide our network into two logically divided modules (for the definition of the terms *module* and *subnet* compare Section 4.1). The complete network (see Figure 4.1) is split into a module containing the *Citrate subnet* and the *Shikimate subnet* (see Figures 4.7 and 4.8, respectively), called *Citrate module*, and a module containing the *Sucrose subnet* and the *UTP subnet* (see Figures 4.4 and 4.9, respectively), called *Sucrose module*.

This biological split features the undivided modeling of the three cycles: Citric acid cycle, glyoxylate cycle (both in Figure 4.7), and Calvin cycle (see Figure 4.4). The split of the network divides the glycolysis in two parts, distributing the pathway over the two modules as shown in the Figures 4.7 and 4.4. The part of the glycolysis taking place in the *Sucrose module* is the reaction chain from β-D-glucose (metabolite '51') to glycerate 3-phosphate (metabolite '40'), while the remaining reactions of the glycolysis from glycerate 3-phosphate (metabolite '40') to acetyl-CoA (metabolite '92') taking place in the *Citrate module*.

The Calvin cycle and the citric acid cycle taking place in different compartments (for a review concerning both cycles (Sweetlove and Fernie, 2013)), which suggest a separation of both if required. A division of the glycolysis seems to be a minor problem compared to a cut of one of the two cycles, as the glycolysis is a linear pathway which takes place in both, the cytosol and the plastid compartments (Plaxton, 1996). Additionally the connections between the glycolysis in the cytosol and the plastid are plentiful (Plaxton, 1996). That suggests that a cut through the glycolysis, which results in two different glycolysis-parts connected by Inputs and Outputs, is more feasible than a cut through an one-compartment cycle.

The glyoxylate cycle and the citric acid cycle share some enzymes and investigations concerning their evolutionary history show similarity to their eubacterial homologues (Schnarrenberger and Martin, 2002), which could suggest a long term co-evolution. Therefore, a division of both cycles into different modules seems not suitable. Following these ideas, we model the Calvin cycle in the *Sucrose module* and the citric acid cycle in the *Citrate module*.

We try to keep the number of crossing edges as small as possible. A cut in a cycle necessitate a second cut in the cycle in order to divide it into two parts, while one cut in a linear pathway is

sufficient for a complete separation of the two emerging parts. This suggests, that a decomposition constructed by the cut of one or more cycles has more crossing edges than a decomposition without the cutting of these cycles. Here, the glycolysis is cut at a reversible reaction, which does not result in less cut edges than the cut through a cycle. Nevertheless, a cut of the same reaction still seems more suitable than a cut through a cycle at two different reactions.

Table 4.3: Table of edges cut by biologically driven decomposition. The first column gives the number of the cut edge (the sort sequence is random). The entries in the columns *Citrate module* and *Sucrose module* present the reactions occurring in the respective modules. The connection direction is shown in the column *con.*. Reversible reactions are split in a forward and a backward reaction (see line No. '3' and '4') to provide a correct count of the cut edges. To decompose the network a *main module* is chosen for each metabolite to which it belongs to. If this metabolite is connected to one or more edges which are cut in the decomposition process, it is split into a *link* in its main module and a *co-link* in the other module. The model-names of each of this connecting metabolites get a short character sequence attached, '_suc' or '_cit', depending whether they are part of the *Sucrose module* or the *Citrate module*, respectively. If the metabolite is a link, its attached character sequence start with a capital letter.

No.	*Citrate module*	con.	*Sucrose module*
1	carbamoyl aspartate (117_Cit) ↑ aspartate transcarbamoylase (E42)	→	carbamoyl aspartate (117_suc) ↓ dihydroorotase (E43)
2	glycolate (35_Cit) ↓ glycolate oxidase (E126)	←	glycolate (35_suc) ↑ phosphoglycolate phosphatase (E128)
3	glycerate 3-phosphate (40_cit) ↓ phosphoglyceromutase (E89_f)	←	glycerate 3-phosphate (40_Suc) ↑ I. 3-PGA kinase (E90_f) II. ribulose 1,5-bisphosphate carboxlyase/oxygenase (Rubisco, E129_1) III. ribulose 1,5-bisphosphate carboxlyase/oxygenase (Rubisco, E129_2)
4	glycerate 3-phosphate (40_cit) ↑ phosphoglyceromutase (E89_b)	→	glycerate 3-phosphate (40_Suc) ↓ 3-PGA kinase (E90_b)
5	glycerate 3-phosphate (40_cit) ↑ glycerate kinase (E123)	→	glycerate 3-phosphate (40_Suc) ↓ 3-PGA kinase (E90_b)

Table 4.3: (continued)

No.	*Citrate module*	con.	*Sucrose module*
6	D-erythrose 4-phosphate (32_cit) ↓ 3-deoxy-arabinoheptulonate 7-phosphate synthase (E64)	←	D-erythrose 4-phosphate (32_Suc) ↑ I. transketolase (E1_44_f) II. aldolase (E11_f) III. transaldolase (E54_f)
7	ammonia (29_Cit) ↓ I. ammonia output (OUT_29) II. asparaginase (E141) III. glutamate dehydrogenase (E106_f) IV. glutamine synthetase (E109) V. serine hydroxymethyltransferase (E118)	←	ammonia (29_suc) ↑ β-ureidopropionase (E138)
8	ammonia (29_Cit) ↓ I. ammonia output (OUT_29) II. asparaginase (E141) III. glutamate dehydrogenase (E106_f) IV. glutamine synthetase (E109) V. serine hydroxymethyltransferase (E118)	←	ammonia (29_suc) ↑ cytidine deaminase (E147)
9	glutamate (2_Cit) ↓ I. aspartate transaminase (E2_b) II. glutamate decarboxylase (E105) III. glutamate dehydrogenase (E106_b) IV. glutamine synthetase (E109) V. alanine aminotransferase (E113_b) VI. glyoxylate aminotransferase (E113)	←	glutamate (2_suc) ↑ CTP synthase (E146)
10	glutamine (33_Cit) ↑ glutamine synthetase (E109)	→	glutamine (33_suc) ↓ CTP synthase (E146)

The complete biologically driven decomposed network is presented in Figure 4.10. Each module (*Sucrose module* and *Citrate module*) has transitions without pre-places which unconditionally provide tokens to the system. The sets of this transitions for each module are the combinations of the Inputs (connections of the network to the environment) and the Imports (connections be-

tween the modules, a detailed explanation of Imports and Exports is given later on). The set of this token-providing transitions of the *Sucrose module* is colored in orange, while the rest of the module is colored in red. The set of Inputs and Imports of the *Citrate module* is colored in blue, the remaining *Citrate module* is highlighted in green. The sizes of each of the two modules are listed in Table 4.4.

Table 4.4: Table of size of modules: Biological decomposition.

	Sucrose module	*Citrate module*
Metabolites	69	72
Reactions	132	125
Edges	295	295

It takes several steps to form a consistent partition of the network. For each metabolite a main module is chosen, in which it is modeled. If the metabolite is connected to a cut edge it is called *link*. For each link a *co-link* is inserted in the non-main module. A list of all cut edges and the links and co-links they are connected to is given in Table 4.3. Each link gets '_Cit' or '_Suc' added to its name, depending on the module it is part of, whereas the name the co-link gets '_cit' or '_suc' appended. E.g., link '117_Cit' belongs to the *Citrate module* (see Table 4.3, line '1'). The corresponding *co-link* in the *Sucrose module* is called '117_suc'. Each edge of the link to transitions in the module of the co-link is shifted to the co-link.

To provide a realistic model each of the links and co-links gets input and output transitions attached accordingly to the direction of the respective connection (see Table 4.3, column *con.*). This process seems to be related to the treatment of external metabolites which as well have two occurrences, one inside the net and one outside of it. From this point of view the link is the external metabolite and its co-link the corresponding occurrence outside of the net (or the other way around for the other module). The difference here is, that the direction of the connection matters for the insertion of the transitions modeling the transport process from one module to the other. If the connecting edge originates in the link the inserted transition is an Output for the link in the link's main module and an Input for the co-link in the other module. Otherwise, if the connecting edge targets the link, this link gets the Input and the co-link gets the Output. If two edges are cut, one targeting the link and one originating from the link, each of the link and the co-link gets an input transition and an output transition.

To separate these new Inputs and Outputs, we name the connecting input transitions Import_XXX_fromYYY, where *YYY* denotes the module from which the metabolite is imported from (*cit* for the *Citrate module* and *suc* for the *Sucrose module*), and *XXX* is to be substituted by the number of the imported metabolite (see Table B.1). The new output transitions are accordingly named Export_XXX_toYYY, where *YYY* denotes the module to which the metabolite is exported to, and *XXX* stands for the number of the exported metabolite (see Table B.1).

Examples for this export and import transitions and their usage are shown in Figures 4.11 and 4.12. Figures 4.11 (a) and (b) show a basic export/import connection using the example of metabolite '117' (carbamoyl aspartate). Basic export/import connections like this are listed in Ta-

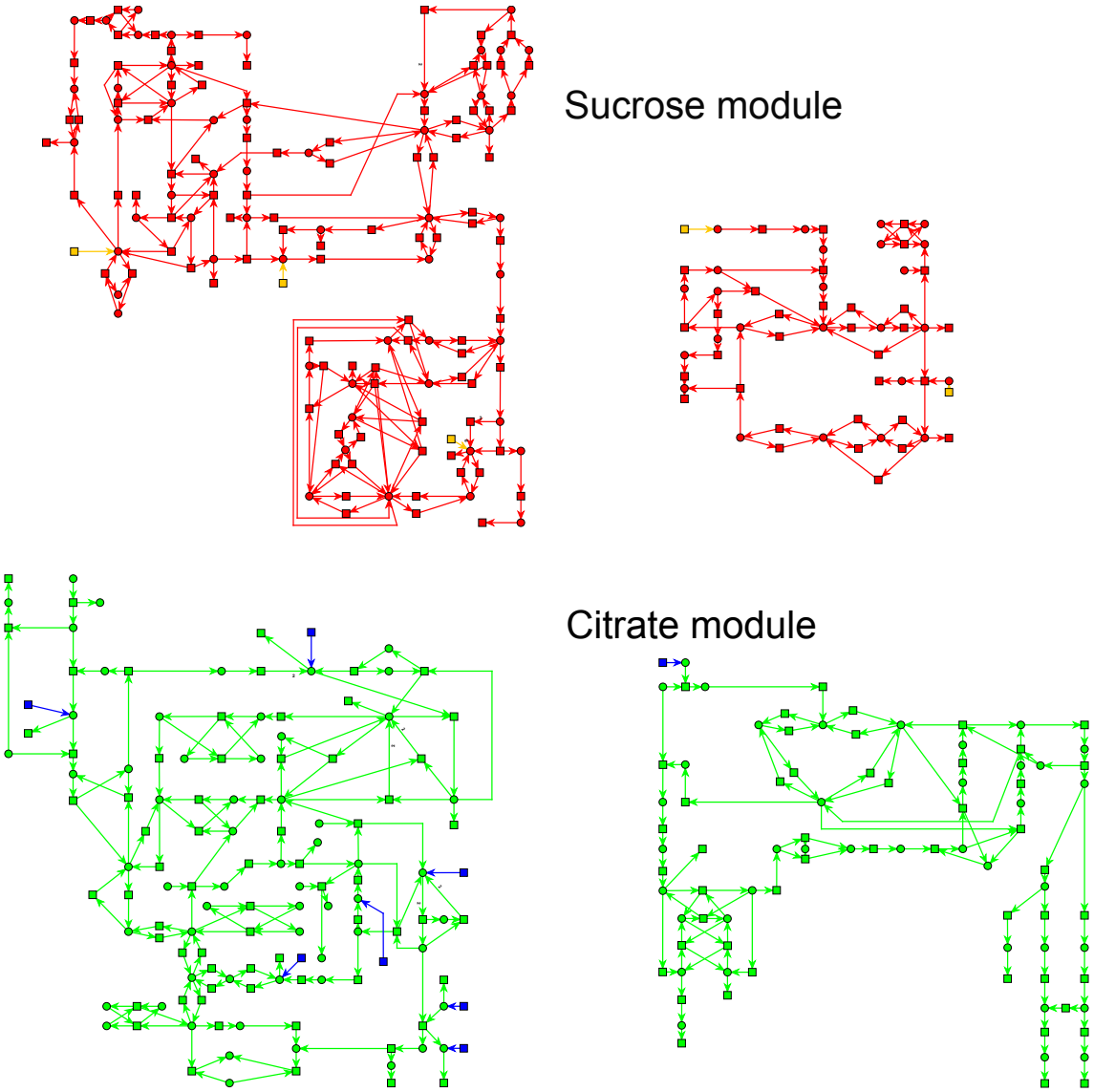
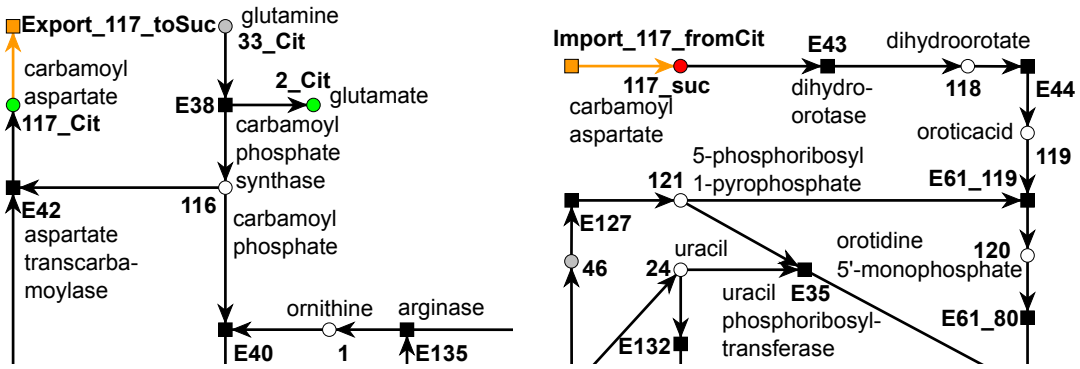


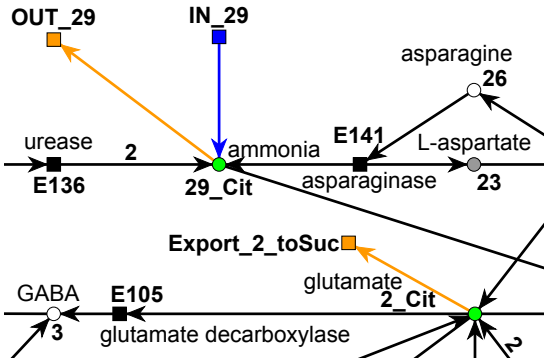
Figure 4.10: Biologically driven decomposition of the *Arabidopsis thaliana* network. The two modules are the *Sucrose module* consisting of the *Sucrose subnet* and the *UTP subnet* (see Figures 4.4 and 4.9) and the *Citrate module* consisting of the *Citrate subnet* and the *Shikimate subnet* (see Figures 4.7 and 4.8). The *Sucrose module* is colored in red, while all transitions providing tokens to the *Sucrose module* without preconditions, which are the Imports (connections between modules) and the Inputs (connections to the environment), are colored in orange. For the *Citrate module* all token-providing transitions are blue, while the *Citrate module* itself is highlighted in green.



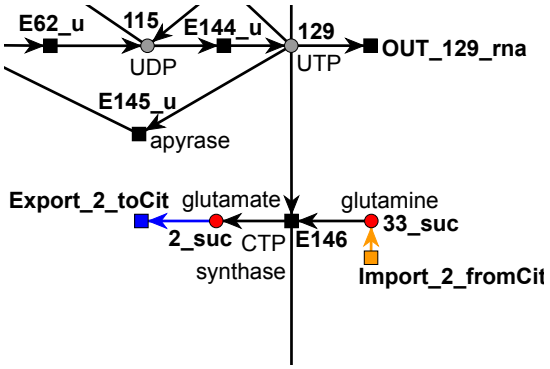
(a) Metabolite '117' (carbamoyl aspartate, Table 4.3 line No. '1') in the *Citrate module*

(b) Metabolite '117' (carbamoyl aspartate, Table 4.3 line No. '1') in the *Sucrose module*

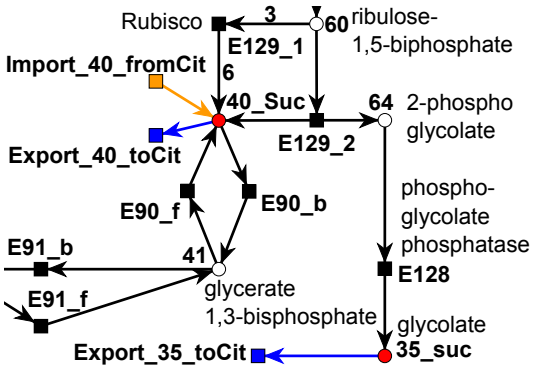
Figure 4.11: Examples of linking areas in the biological decomposition of the *Arabidopsis thaliana* network. To the linking metabolites (the ones that are needed in both network modules) a 'cit' or 'suc' is added depending if the link is part of the *Citrate module* or the *Sucrose module*, respectively. The importing linking transitions are named Import_XXX_fromYYY, where XXX denotes the number of the imported metabolite (for the corresponding name see Table B.1) and YYY is 'cit' or 'suc' depending on the module, from which the metabolite is imported. The exporting transitions are named Export_XXX_toYYY, where XXX stands for the number of the exported metabolite and YYY is 'cit' or 'suc' depending on to which module the metabolite is exported to ('cit' for export to the *Citrate module* and 'suc' for export to the *Sucrose module*). Contrary to the coloring in Figure 4.10, in which all output transitions (Exports and Outpus) are colored in red (*Sucrose module*) or green (*Citrate module*), the Exports are here colored in the same color as the Imports of the other module (Exports of the *Sucrose module* are orange, Exports of the *Citrate module* are blue). Transferred metabolites of the *Sucrose module* are red and transferred metabolites of the *Citrate module* are green, while every other metabolite remains white (or gray if it is depicted by a logical place). Identifiers used in the model are printed in boldface.



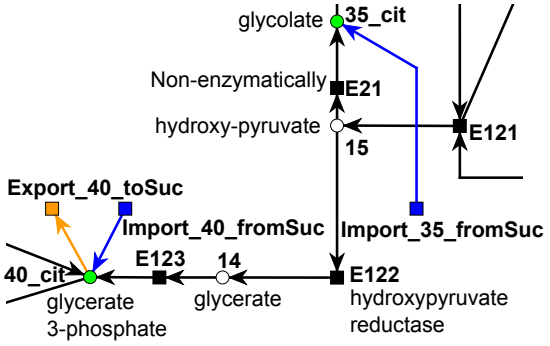
(a) Metabolites '29' (ammonia, Table 4.3 line No. '7' and '8) and '2' (glutamate, Table 4.3 line No. '9') in the *Citrate module*



(b) Metabolites '2' (glutamate, Table 4.3 line No. '9') and '33' (glutamine, Table 4.3 line No. '10') in the *Sucrose module*



(c) Metabolites '35' (glycolate, Table 4.3 line '2') and '40' (glycerate 3-phosphate, Table 4.3 line No. '3', '4', and '5') in the *Sucrose module*



(d) Metabolites '35' (glycolate, Table 4.3 line '2') and '40' (glycerate 3-phosphate, Table 4.3 line No. '3', '4', and '5') in the *Citrate module*

Figure 4.12: Examples of linking areas in the biological decomposition of the *Arabidopsis thaliana* network, continued. For description see Figure 4.11.

ble 4.3 line No. '1', '2', '5', and '6'. Some of these import or export reactions can be combined, following reduction rules for Petri nets presented in earlier studies (Murata, 1989; Reddy et al., 1993) which allow a merging of parallel transitions. Following these rules, the two transitions 'IN_29' and 'Import_29_fromSuc' which provide link 29_Cit (metabolite '29' correspond to ammonia) to the *Citrate module* (see Table 4.3 line No. '7' and '8', and Figure 4.12a) can be merged. Figure 4.12b shows the situation of Table 4.3 line No. '9' and '10', where metabolite '2' (glutamate) is imported and metabolite '33' (glutamine) is exported. Both are consumed (glutamine) and produced (glutamate) by the same transition 'E146'.

Metabolite '40' (glycerate 3-phosphate) is reversible produced and consumed in both network modules, which requires a reversible connection between the *Citrate module* and the *Sucrose module* (see Table 4.3 line No. '3' and '4'). This situation is presented in the Figures 4.12c (glycerate 3-phosphate in the *Sucrose module*) and 4.12d (glycerate 3-phosphate in the *Citrate module*).

The PN model does not include compartments, and none of the decompositions is created to insert compartments to the model. Nevertheless compartments are part of a living cell. We use this property to justify our network decomposition.

For glycolate (metabolite '35', Table 4.3 line No. '2', Figures 4.12c and 4.12d) we find transport in C_3-plants from the chloroplast to the cytosol (Flugge and Heldt, 1991), which could support a division of glycolate over two network modules.

The decision to export D-erythrose 4-phosphate (metabolite '32', Table 4.3 line No. '6') from the *Sucrose module* is driven by the choice to model the Calvin cycle and the citric acid cycle in different modules. D-erythrose 4-phosphate is part of the Calvin cycle as well as of the shikimate pathway. The other compound needed by reaction 'E64' (Herrmann, 1995), which starts the shikimate synthesis, is phospho*enol*pyruvate (metabolite '34'). Phospho*enol*pyruvate is connected to the citric acid cycle. This leads to two options, to put the shikimate pathway either to the *Sucrose module* or to the *Citrate module*, which means either D-erythrose 4-phosphate (metabolite '32') or phospho*enol*pyruvate (metabolite '34') has to be distributed over the two modules. A localization of the shikimate pathway in the *Sucrose module* would render this module significantly bigger than the *Citrate module*, which is contrary to our intention of providing modules of comparable size.

The reactions of the shikimate pathway which synthesize chorismate (metabolite '103') from D-erythrose 4-phosphate (metabolite '32') and phospho*enol*pyruvate(metabolite '34') are suggested to take place in the plastid (Herrmann and Weaver, 1999). On the other side, there exist theories of a dual localization of the chorismate synthesis in both, plastid and cytosol, driven by the observation that the phenylpropanoid pathway takes place in the cytosol (Weaver and Herrmann, 1997). Additionally the chorismate mutase, catalyzing the reaction of chorismate (metabolite '103') to prephenate (metabolite '105'), has isoenzymes located in the cytosol (Schmid and Amrhein, 1995), which either as well suggests a dual pathway location of chorismate synthesis or an export of chorismate from the plastid to the cytosol.

If we put the chorismate synthesis to the *Sucrose module* an import of phospho*enol*pyruvate would be necessary, but phospho*enol*pyruvate does interact badly with the C_3 transporter which

transports glycerate 3-phosphate (metabolite '40', Table 4.3 line No. '3-5') (Flugge and Heldt, 1991). Phospho*enol*pyruvate is part of the glycolysis, meaning its import should not be independent of the export of glycerate 3-phosphate, which is part of the glycolysis as well. To model the complete glycolysis in the *Sucrose module* would render the module too complex for an analysis.

We decompose the network in such a way that D-erythrose 4-phosphate (metabolite '32', Table 4.3 line No. '6') is exported from the *Sucrose module*, because (i) a strict division is not needed as we do not model compartments, and (ii) we do not model the cytosolic and the plastidic glycolosis as different pathways. Otherwise a less feasible export of phospho*enol*pyruvate from the *Citrate module* would be required. Together this put the shikimate pathway in the *Citrate module*.

The distribution of Glutamine (metabolite '33', Table 4.3 line No. '10' over the two models is justified by the fact that glutamine plays a central role in the nitrogen economy of conifers (Cánovas et al., 2007). Additionally, α-ketoglutarate is known as a important compound for nitrogen transportation in plants (Temple et al., 1998). As glutamate is a compound which is convertible into both, α-ketoglutarate and glutamine (Aubert et al., 2001; Forde and Lea, 2007), a transport of glutamate (metabolite '2', Table 4.3 line No. '9') between several network modules seems feasible.

The distribution of metabolite '117' (carbamoyl aspartate, Table 4.3 line No. '1', Figures 4.11a and 4.11b) over the two modules is driven by practical intentions. It is a direct product from the urea cycle with one producing (transition 'E42', aspartate *trans*carbamoylase, (Zrenner et al., 2006)) and one consuming reaction (transition 'E43', dihydroorotase, (Zrenner et al., 2006)). Therefore it is a good candidate for a cut as only one edge is cut, which is concordant with the intention to minimize the crossing edges between the modules. Furthermore we need a cut in the neighborhood of metabolite '117' to ensure a more or less equal size of the two modules, and we decide not to cut through the urea cycle.

The decision on the distribution of metabolite '29' (ammonia, Table 4.3 line No. '7' and '8') over the two modules is driven by the fact that ammonia is set to external in our network, which solves the problem of the correct transfer. External metabolites are attached to an input and an output transition each, and import and export do not have impact on the availability of external metabolites.

4.2.2 Automatic Decomposition

To avoid a partition biased by human intuition we additionally split the network by an automatic method. The problem of graph partition in equal-sized subgraphs with minimum crossing edges is known to be NP-hard (Johnson, 1982). We choose the well-known *Kernighan-Lin Algorithm* (KL-algorithm, Kernighan and Lin, 1970), which provides a heuristic approach for graph partitioning. It starts from a given distribution of the nodes into two subgraphs, followed by a node swapping between the two subgraphs until a given quality criteria is met (compare Section 3.6).

Heuristic methods can find non-optimal solutions for the problem they are working on, and the optimality of solutions can strongly depend on the starting configuration. The quality of the solution presented by the KL-algorithm is based on the initial partition used as input for the algorithm.

To address this issue of quality dependency, we extend the KL-algorithm by starting it one million times, each time with a different random initial graph partition as input. This approach results in one million *runs*, each generating one solution for the partitioning problem.

Each of the one million solutions of our implementation of the KL-algorithm is composed of two modules, which we call *Graph A* and *Graph B*, and the crossing edges. We identify all solutions with a minimal count of crossing edges compared to all other solutions. Every solution can occur multiple times. We count the occurrence for each of these *minimal solutions* and use the one with the most occurrences for further analysis.

The chosen minimal solution is refined in the same way as the biological driven partition. We choose a belonging module for each metabolite called link indicated by an '_A' or '_B' appended to the metabolite name depending on the module it is belonging to. To the name of the corresponding co-link we attach an '_a' or '_b'. Note that lignin (metabolite '30') is split into two submetabolites which are named '30a' and '30b' (see Section 4.1). This naming is not motivated by the automatic network decomposition, and both lignins are not distributed over different modules.

Table 4.6 presents every edge spanning from Graph A to Graph B in our chosen minimal solution. Note that some of the crossing edges are connected to reversible reactions, which results in a reversible connection between the two modules (Table 4.6 line No. '6' and '7'). Further in the refinement of the partition, transitions and places are added to the modules to provide a realistic connection between them (Figure 4.14a and 4.14b). Metabolite '117' is part of both modules, and gets a new transition in each of them, one consuming the substance, one producing it.

Table 4.5: Table of size of modules for automatic decomposition.

	Graph A	Graph B
Metabolites	70	72
Reactions	132	127
Edges	288	304

Figure 4.13 illustrates the two modules of the automatic graph decomposition. Graph A is shown in red and Graph B is highlighted in green. All token-providing transitions (Inputs and Imports) of Graph A are orange and of Graph B these transitions are colored in blue. The sizes of Graph A and Graph B are presented in Table 4.5. The mismatch of the number of metabolites and reactions in Graph A and B is arisen from the partition method and the following refinement. Our implementation of the KL-algorithm does not distinguish between metabolites and reactions, both are treated equally.

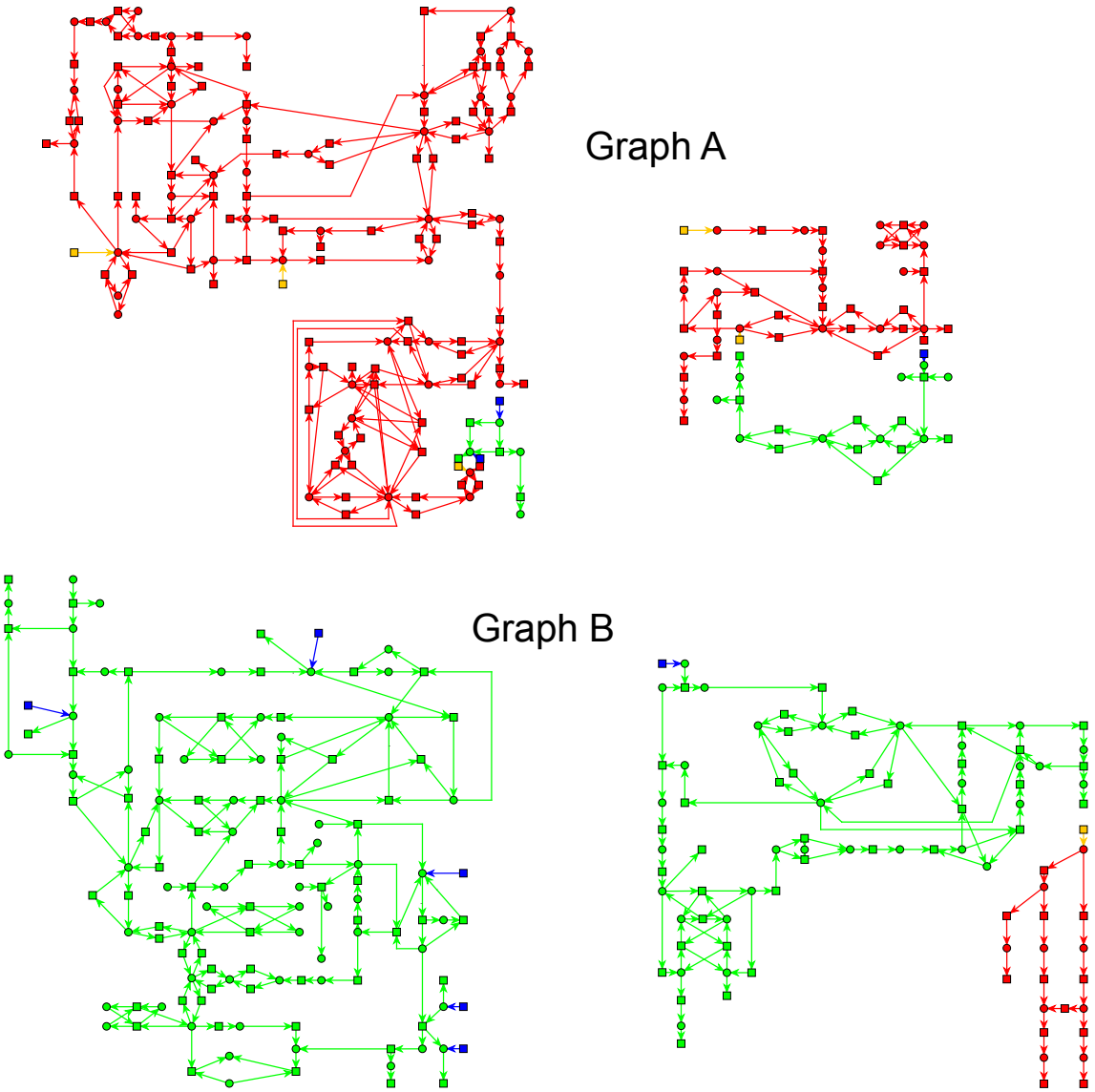


Figure 4.13: Automatic decomposition of the *Arabidopsis thaliana* network using the KL-algorithm. Graph A is colored in red, Graph B in green. Input transitions (including Import and IN) of Graph A are colored orange, input transitions of Graph B are colored in blue.

Table 4.6: Table of edges cut by automatic decomposition. The first column gives the number of the cut edge (the sort sequence is random). The entries in the columns *Graph A* and *Graph B* show the reactions occurring in the respective modules. The connection direction is shown in the column *con.*. Reversible reactions are split in a forward and a backward reaction (see line No. '6' and '7', column *con.*) to provide a correct count of the cut edges. *Links* get an '_A' or '_B' added to their names depending on the module they belong to, *co-links* get an '_a' or '_b'.

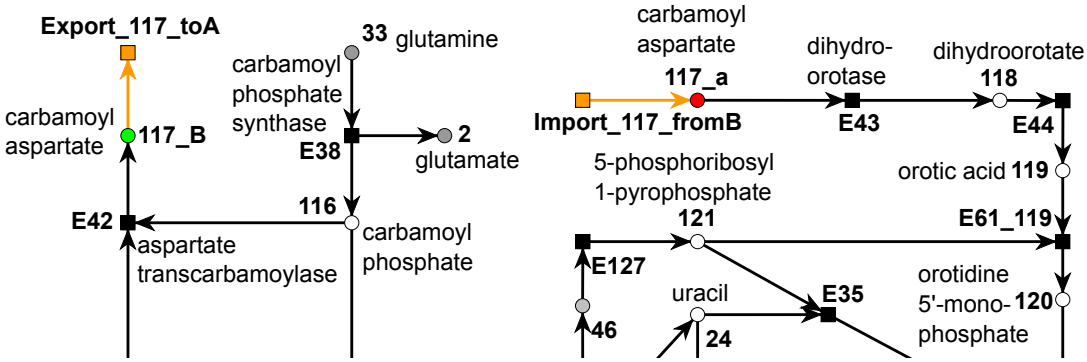
No.	Graph A	con.	Graph B
1	carbamoyl aspartate (117_a) ↓ aspartate transcarbamoylase (E42)	← ←	carbamoyl aspartate (117_B) ↑ dihydroorotase (E43)
2	ribulose-1,5-biphosphate (60_a) ↑ phosphoribulokinase (E130)	→	ribulose-1,5-biphosphate (60_B) ↓ I. ribulose 1,5-bisphosphate carboxlyase/oxygenase (Rubisco, E129_1) II. ribulose 1,5-bisphosphate carboxlyase/oxygenase (Rubisco, E129_2)
3	coniferyl aldehyde (17_A) ↓ I. cinnamyl alcohol dehydrogenase/ sinapyl alcohol dehydrogenase (E85) II. ferulate 5-hydroxylase (E86_133)	←	coniferyl aldehyde (17_b) ↑ cinnamoyl CoA reductase (E84)
4	D-erythrose 4-phosphate (32_A) ↑ I. transketolase (E1_44_f) II. aldolase (E11_f) III. transaldolase (E54_f)	→	D-erythrose 4-phosphate (32_b) ↓ 3-deoxy-arabinoheptulonate 7-phosphate synthase (E64)
5	ammonia (29_a) ↑ β-ureidopropionase (E138)	→	ammonia (29_B) ↓ I. ammonia output (OUT_29) II. glutamate dehydrogenase (E106_f) III. glutamine synthetase (E109) IV. serine hydroxymethyltransferase (E118) V. asparaginase (E141)

Table 4.6: (continued)

No.	Graph A	con.	Graph B
6	glycerate 3-phosphate (40_A) ↑ I. transketolase (E1_44_b) II. transketolase (E1_48_b) III. triose-phosphate isomerase (E13_f) IV. fructose-bisphosphate aldolase (E14_f) V. transaldolase (E54_b)	→	glycerate 3-phosphate (40_b) ↓ 3-PGA kinase (E90_f)
7	glycerate 3-phosphate (40_A) ↓ I. transketolase (E1_44_f) II. transketolase (E1_48_f) III. triose-phosphate isomerase (E13_b) IV. fructose-bisphosphate aldolase (E14_b) V. transaldolase (E54_f)	←	glycerate 3-phosphate (40_b) ↑ 3-PGA kinase (E90_b)
8	uridine diphosphate (129_A) ↑ I. UDP-galacturonate pyrophosphatase (E26_f) II. nucleoside diphosphate kinase (E144_u)	→	uridine diphosphate (129_b) ↓ CTP synthase (E146)
9	uridine (126_A) ↓ I. uracil phosphoribosyltransferase (E35_u) II. uridine nucleosidase (E131)	←	uridine (126_b) ↑ cytidine deaminase (E147)

Figures 4.14 and 4.15 show several examples of split edges suggested by the KL-algorithm. All distributed places, i.e., the links and co-links that are part of Graph A are colored in red and all distributed places part of Graph B in green. The colors of Imports in Graph A are orange. To allow an easy identification of the corresponding Exports, all of the Exports in Graph B (which are the ones that correspond to Imports in Graph A) are colored in orange as well. The other way around is true for Graph B. Here, all Imports are blue and therefore all Exports in Graph A are blue as well.

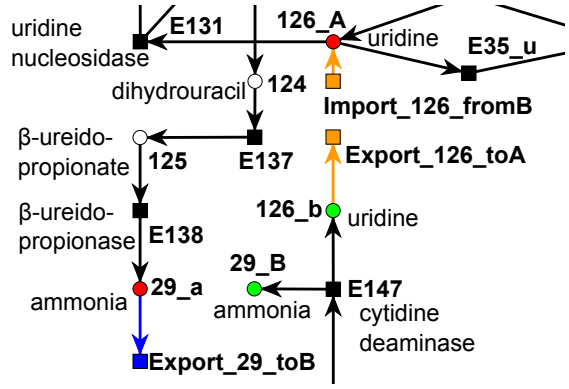
Figures 4.14a and 4.14b present the integration of metabolite '117' (carbamoyl aspartate) in the two modules Graph A and Graph B. The import and export of metabolite '117' is described in



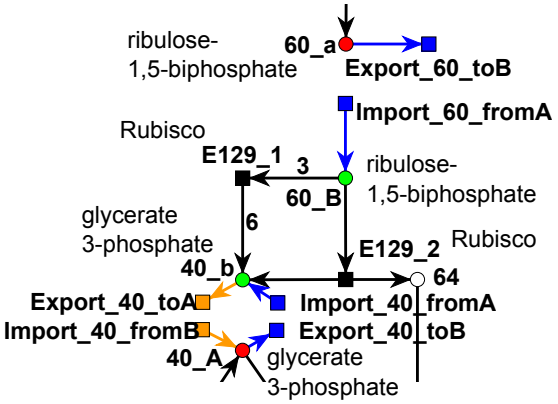
(a) Metabolite '117' (carbamoyl aspartate, Table 4.6 line No. '1') in Graph B

(b) Metabolites '117' (carbamoyl aspartate, Table 4.6 line No. '1') in Graph A

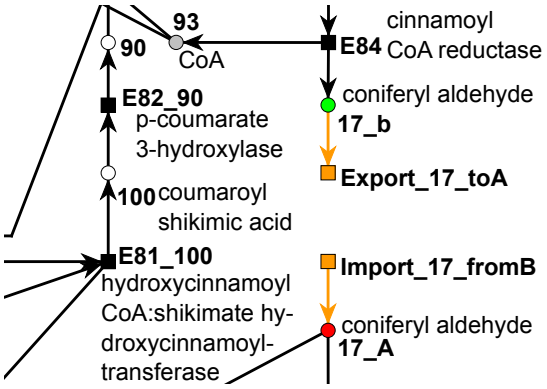
Figure 4.14: Examples of linking areas in the automatic decomposition of the *Arabidopsis thaliana* network. To the linking metabolites (the ones that are needed in both network modules) an '_a' or '_b' is added, depending whether the link is part of Graph A or Graph B, respectively. The importing linking transitions are named Import_XXX_fromYYY, where XXX denotes the number of the imported metabolite (for the corresponding name see Table B.1) and YYY is 'A' or 'B' depending on the module from which the metabolite is imported from. The exporting transitions are named Export_XXX_toYYY, where XXX stands for the number of the exported metabolite, and YYY is 'A' or 'B' depending on to which module the metabolite is exported to ('A' for export to Graph A and 'B' for export to the Graph B). Contrary to the coloring in Figure 4.13, in which all output transitions (Exports and Outpus) are colored in red (Graph A) or green (Graph B), here the Exports are colored in the same color as the Imports of the other module (Exports of Graph B are orange, Exports of Graph A are blue). Transferred metabolites of Graph A are red and transferred metabolites of Graph B are green, while every other metabolite remains white (or gray if it is depicted by a logical place). Identifiers used in the model are printed in boldface.



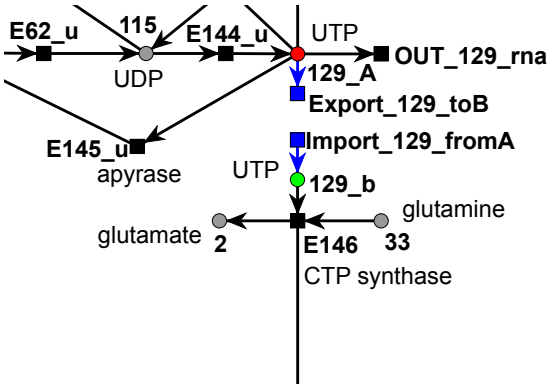
(a) Metabolites '29' (ammonia, Table 4.6 line No. '5') and '126' (uridine, Table 4.6 line No. '9') in Graph A (red) and Graph B (green)



(b) Metabolites '40' (glycerate 3-phosphate, Table 4.6 line No. '6' and '7') and '60' (ribulose-1,5-biphosphate, Table 4.6 line No. '2') in Graph A (red) and B (green)



(c) Metabolite '17' (coniferyl aldehyde, Table 4.6 line No. '2') in Graph A (red) and Graph B (green)



(d) Metabolite '129' (UTP, Table 4.6 line No. '8') in Graph A (red) and Graph B (green)

Figure 4.15: Examples of linking areas in the automatic decomposition of the *Arabidopsis thaliana* network, continued. For description see Figure 4.14.

Table 4.6 line No. '1'.

The connection of the two modules by metabolites '29' (ammonia) and '126' (uridine) is depicted in Figure 4.15a. It corresponds to Table 4.6 line No. '5' (ammonia) and '9' (uridine).

Metabolite '40' (glycerate 3-phosphate) is reversibly connected in both modules, necessitating a reversible connection to both modules after the cut. Figure 4.15b demonstrates this reversible connection and refers to Table 4.6 line No. '6' (Graph A $\rightarrow$ Graph B) and '7' (Graph B $\rightarrow$ Graph A). Additionally, the figure presents a cut edge connected to metabolite '60' (ribulose-1,5-biphosphate) addressed in Table 4.6 line No. '2'. These two cuts split the Calvin cycle over the two modules.

The cut of the edge between UTP (metabolite '129') and cytidine triphosphate synthase (reaction 'E146') is represented in Figure 4.15d and Table 4.6 line No. '8'. Together with the distribution of metabolite '126' (uridine) shown in Figure 4.15a and Table 4.6 line No. '9' the cut causes a break of the interconversion cycle of pyrimidine phosphates.

The connection of metabolite '17' (coniferyl aldehyde, Table 4.6 line No. '3', Figure 4.15c) to a cut edge is most likely an artifact of the weighting function. The algorithm tries to distribute the nodes equally over the two modules minimizing the crossing edges. Cutting an edge connected to coniferyl aldehyde allows the algorithm to put the nodes following metabolite '17' to count to one Graph without adding any more cut edges besides the first, because every node following '17' has no connection to any other node upstream of metabolite '17'(Figure 4.8).

4.2.3 Comparison of the two Decomposed Modules

The KL-algorithm cuts several edges analog to the biologically driven decomposition. Both decompositions have the distribution of carbamoyl aspartate (metabolite '117'), ammonia (metabolite '29'), D-erythrose 4-phosphate (metabolite '32'), and glycerate 3-phosphate (metabolite '40') over the respective submodules in common.

Table 4.7 shows a numerical comparison of the modules of the two decompositions. Graph A and the *Sucrose module* have a lot of metabolites and reactions in common. The same holds true for Graph B and the *Citrate module*. As links and co-links are counted in each module, the similarity between Graph B and the *Sucrose module* indicates 15 common metabolites, of which 10 are distributed over two modules in either one or both different decompositions. Similar to this, 5 of 12 the common metabolites between the *Citrate module* and Graph A are distributed in one or both decompositions. This indicates a lower similarity between Graph A and the *Citrate module* and between the *Sucrose module* and Graph B as implied in Table 4.7. A detailed comparison of metabolites (places), reactions (transitions), and edges between all pairs of modules of the decompositions is given in Table C.4.

Table 4.7: Comparison of the two decompositions. For each combination of modules the number of shared metabolites, reactions and edges is given. Identical links and co-links and their Imports and Exports are counted. For a detailed comparison, see Table C.4.

	Graph A - *Sucrose module*	Graph B - *Sucrose module*	Graph A - *Citrate module*	Graph B - *Citrate module*
metabolites	61	15	12	64
reactions	115	15	14	109
edges	264	31	23	274

Contrary to the biologically driven decomposition the automatic decomposition cuts the Calvin cycle, as metabolite '60' (ribulose-1,5-biphosphate, Table 4.6 line No. '2', Figure 4.15b) and metabolite '40' (glycerate 3-phosphate, Table 4.6 line No. '6' and '7', Figure 4.15b) are connected to cut edges. This has severe effects on the supplied pathways as discussed later in Section 4.4.2.

The cut of edges connected to uridine (metabolite '126', Table 4.6 line No. '9', Figure 4.15a) and cytidine triphosphate (metabolite '129', Table 4.6 line No. '8', Figure 4.15d) splits the circular pyrimidine ribonucleotides interconversion pathway. It is not clear, what impact this split of the pyrimidine interconversion cycle has on the aforementioned pathway reduction found in Section 4.4.2. Modulated by this split, the transition 'E146' is now part of Graph B (see Figure 4.15d) which is contrary to the *Citrate module* of the biological decomposition (see Figure 4.12b). As a consequence glutamate (metabolite '2') and glutamine (metabolite '33') are not distributed over both modules in the automatic decomposition.

Only one edge connected to ammonia (metabolite '29') is cut in the automatic decomposition (Figure 4.15a and Table 4.6 line No. '5') due to the fact that the other transition producing ammonia in the *Sucrose module* ('E147', cytidine deaminase (Zrenner et al., 2006)) is now part of Graph B.

The distribution of metabolite '117' (carbamoyl aspartate) in the biological driven decomposition (Table 4.3 line No. '1' and Figures 4.11a and 4.11b) was affected by the idea of minimizing the crossing edges, which seems to be a step reproducible by the automatic decomposition (Table 4.6 line No. '1' and Figure 4.14a). The same holds true for the distribution of ammonia (biologically driven decomposition: Table 4.3 line No. '7' and '8', automatic decomposition: Figure 4.15a and Table 4.6 line No. '5').

Taken together, uncertain cuts of the biologically driven decomposition are justified by the findings of the automatic decomposition. Cuts through cycles and their following effects make clear that an uncurated use of automatic decompositions are not advisable.

4.3 Reduction of the Network

An analysis of the t-invariants of submodules emerging from a decomposition as performed in Section 4.2 may result in the finding that both submodules are CTI. It is a dangerous conclusion to assume that the CTI property of the submodules implicates the CTI property of the original

network. Figure 4.16 shows an example of a network decomposition which does not conserve the CTI property between the original network and its decompositions.

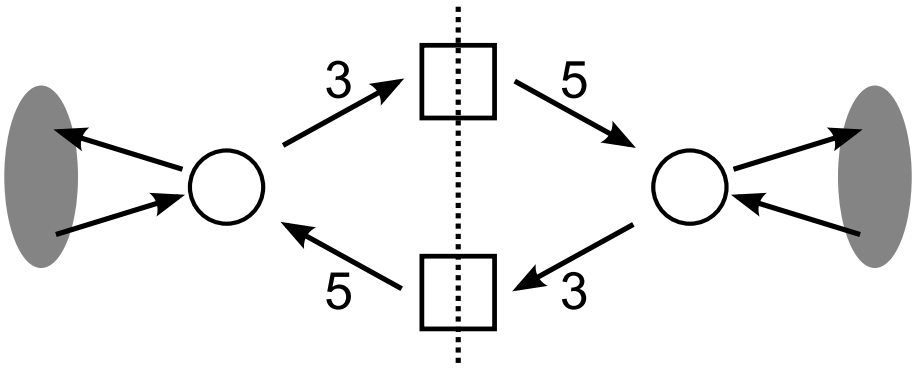


Figure 4.16: Problematic nature of the CTI determination of decomposed networks. The presented total network is not CTI. The gray ovals indicate the structure of the remaining parts of the network. If it is split in a way (dashed line) that both transitions are substituted by input and output transitions in the two submodules, the decomposed modules on the left and on the right are CTI.

In an earlier study a comparable network has been reduced following biological intuition, e.g., by the merging of α-D-glucose and β-D-glucose (Nöthen, 2009). The method is not comparable to the reduction process discussed here as no strict set of reduction-rules is given. It can not be guaranteed that the CTI property is conserved throughout the reduction process. On the other hand the method suggests that a reduction by merging metabolites may have a considerable effect on the number of t-invariants, which motivates a deeper inspection of this possibility.

Our approach to reduce a PN is determined to conserve especially the CTI property. The method has been in parts presented before (Ackermann et al., 2012). We introduce the complete approach and extend the part of the *Reduction by Common Transition Pair* (see Definitions 4.3.2, 4.3.4, and 4.3.6) from the trivial to the general case, and provide extended proofs for the CTI-conservativeness of the CTP, the ITP, and the extended CTP reduction. Given a network which is covered by t-invariants, this reduction methods preserve the covered status.

4.3.1 Definitions

For the development of our method we make use of the fact, that the set of all possible finite Petri nets $\mathbb{P}$ divides into two disjoint subsets, the Petri nets which are CTI ($\mathbb{P}_{cti}$), and the ones that are not ($\mathbb{P}_{ncti}$).

Definition 4.3.1. CTI-conserving*:*
A relation $r : N \to N'$ is called CTI*-conserving, if* $\forall N \in \mathbb{P} : N \in \mathbb{P}_{cti} \Leftrightarrow N' \in \mathbb{P}_{cti}$.

Inspired by the idea of *maximum common transition sets* (MCT-S, (Sackmann et al., 2006)) and trivial t-invariants, we introduce two network structures common to biochemical networks.

First, the *Common Transition Pair*, which is commonly spoken an unbranched reaction chain of a reaction t_i and t_j, connected via the place p_c (see Figure 4.17a). Unbranched means, p_c does

not participate in any further reaction, while the reactions t_i and t_j may have further metabolites attached. Longer unbranched reaction chains are CTPs in series.

Definition 4.3.2. *Common Transition Pair (CTP):*
A pair of transitions $(t_i,t_j);t_i,t_j \in T$ *of a* PN $N=(P,T,F,w,m_0)$ *is called* Common Transition Pair*, if* $\exists p_c \in P$ *with:*

- $(t_i,p_c) \in F \wedge (p_c,t_j) \in F$ *(p_c is connected to both, t_i and t_j)*
- $\forall t_l \in T \backslash \{t_i,t_j\} : (p_c,t_l) \notin F \wedge (t_l,p_c) \notin F$ *(p_c is not connected to any other transition t_l)*

p_c *is called* Connector *of* t_i *and* t_j.

Second, another network structure is obviously abundant in metabolic networks: The reversible reaction. In PN terminology a reversible reaction is split into a *forward* and a *backward* reaction (see Section 3.3). We focus on reversible reactions with one educt and one product only.

Definition 4.3.3. *Invariant Transition Pair (ITP):*
A pair of transitions $(t_i,t_j);t_i,t_j \in T$ *of a* PN $N=(P,T,F,w,m_0)$ *is called* Invariant Transition Pair*, if* $\exists p_k, p_l$ *with:*

- $\bullet t_i = t_j \bullet = p_l$ *(pre-place (t_i) = post-place (t_j))*
- $\bullet t_j = t_i \bullet = p_q$ *(pre-place (t_j) = post-place (t_i))*
- $w(p_l,t_i) = w(t_j,p_l) = w(p_q,t_j) = w(t_i,p_q) = 1$ *(each weight w from and to t_i and t_j is 1)*

p_k and p_l are called *shared places* of t_i and t_j.

4.3.2 CTP and ITP Reduction Rules

We introduce two types of reductions which conserve the CTI property: The *CTP reduction* and the *ITP reduction*. For the development of the CTP reduction we follow the idea of MCT-S which are sets of transitions not participating independently in t-invariants. All transitions of an MCT-S take part in the same t-invariants and are absent in all others. Intuitively, tokens on p_c are generated by t_i only, and to clear the place of tokens, they must be removed by t_j. If t_i is covered by at least one of all t-invariants, t_j has to be covered by the same invariant as well (i.e. t_i and t_j constitute as MCT-S). As the two transitions are always covered or not covered together, a reduction seems feasible.

Definition 4.3.4. *CTP reduction:*
Given a Petri net $N=(P,T,F,w,m_0)$*, a CTP* $(t_i,t_j),t_i,t_j \in T$ *with the Connector* $p_c \in P$*, and weights* $w(t_i,p_c) = w(p_c,t_j) = a, a \in \mathbb{N}, \{(t_i,p_c),(p_c,t_j) \in F\}$*, the relation* $r : N \to N'$ *is called* CTP reduction*. The reduced net* $N'=(P',T',F',w',m'_0)$ *is defined by*

1. $P' \leftarrow P \backslash \{p_c\}$ *(delete p_c)*
2. $T' \leftarrow T \backslash \{t_j\}$ *(t_j is merged to t_i)*

3. $F' \leftarrow F \setminus \{(p,t_j),(t_j,p) | p \in P\} \cup \{(t_i,p) | p \in t_j\bullet\} \cup \{(p,t_i) | p \in \bullet t_j\}$ *(all edges leading from and to t_j are removed, for each of them a new edge is added, leading from or to t_i, respectively)*

4. $w' : \begin{cases} (t_i,p) \rightarrow w(t_i,p) + w(t_j,p) & \forall p \in t_j\bullet \\ (p,t_i) \rightarrow w(p,t_i) + w(p,t_j) & \forall p \in \bullet t_j \\ e \rightarrow w(e) & otherwise \end{cases}$

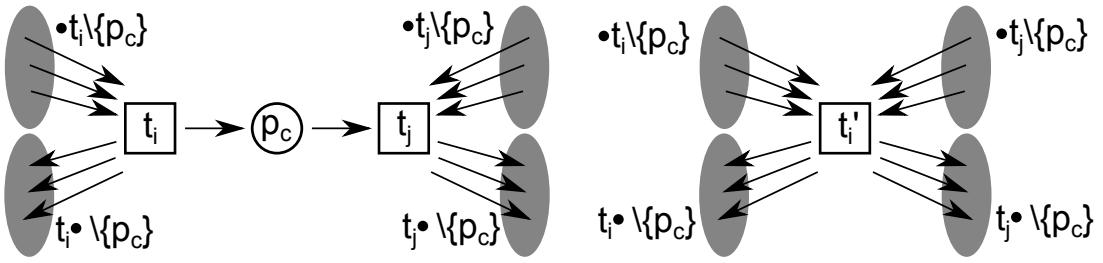
5. $\forall p \in P' : m'_0(p) \leftarrow m_0(p)$

Note that any of the weights $w(t_i,p)$, $w(t_j,p)$, $w(p,t_i)$, and $w(p,t_j)$ can be zero, if the corresponding edge does not exist. The cases

$$w' : \begin{cases} (t_i,p) \rightarrow w(t_j,p) & \forall p \in t_j\bullet \\ (p,t_i) \rightarrow w(p,t_j) & \forall p \in \bullet t_j \end{cases}$$

are hence implicitly covered by Definition 4.3.4.

Figure 4.17 demonstrates the CTP reduction. Place p_c is only connected to two transitions t_i and t_j. The weights of the edges are identical. Subfigure 4.17a depicts the unreduced state (i.e. a CTP), while Subfigure 4.17b shows the reduced state after the removal of the Connector and transition t_j. All edges of t_j apart from that connected to p_c are transferred to t_i.



a) CTP b) reduced CTP

Figure 4.17: Example for a CTP (a) and its reduced form (b). A CTP is defined in Definition 4.3.2, the reduction process in Definition 4.3.5. The gray ellipses represent the rest of the network. The naming of the transitions (squares) and the places (circles) are corresponding to the names in the definitions of a CTP, while the removed transition and place are conform with the definition of the CTP reduction.

Lemma 4.3.1. *The relation $r : N \rightarrow N'$ is* CTI *conserving, if N' is the net N reduced by one CTP reduction.*

To proof this lemma, we have to show the equivalence of $N \in \mathbb{P}_{cti} \Leftrightarrow N' \in \mathbb{P}_{cti}$.

Proof of Lemma 4.3.1. Given a Petri net $N = (P,T,F,w,m_0)$ and its reduction $N' = (P',T',F',w',m'_0)$, N is CTI, and contains a CTP (t_i,t_j) and its Connector p_c ($|T| = n, |P| = m$).

The topology of the net then implies a row k in the incidence matrix, corresponding to the place p_c, where $c_{k,i} \neq 0$, $c_{k,j} \neq 0$ and $c_{k,l} = 0, \forall l \neq i, j$. Because N is CTI, a vector x exists in the invariant set I of N, fulfilling the condition $Cx = 0$. As $c_{k,1}x_1 + \cdots + c_{k,i}x_i + \cdots + c_{k,j}x_j + \cdots + c_{k,n}x_n = 0$, and with $c_{k,l} = 0, \forall l \neq i, j$, row k implies

$$\begin{aligned} c_{k,i}x_i + c_{k,j}x_j + &= 0 \\ ax_i - ax_j &= 0 \\ ax_i &= ax_j \\ x_i &= x_j \end{aligned}$$

It applies that

$$\forall q \in \{1 \ldots m\} : \sum_{l \neq i,j} (c_{q,l}x_l) + c_{q,i}x_i + c_{q,j}x_j = 0 \tag{4.1}$$

The reduction rules which map C to C' specify $c'_{q,i} = c_{q,i} + c_{q,j}, \forall 1 \leq q \leq m$, and leave the other elements of the incidence matrix untouched. The invariant set I' of the reduced network contains the invariant vectors x'. This induces $\forall q \in \{1 \ldots m-1\}$:

$$\begin{aligned} \sum_{l \neq i,j} (c_{q,l}x_l) + c'_{q,i}x'_i &= 0 \\ \sum_{l \neq i,j} (c_{q,l}x_l) + (c_{q,i} + c_{q,j})x'_i &= 0 \\ \sum_{l \neq i,j} (c_{q,l}x_l) + c_{q,i}x'_i + c_{q,j}x'_i &= 0 \end{aligned} \tag{4.2}$$

If $x'_i = x_i$ (which induces $x'_i = x_j$), Equation 4.2 is identical to Equation 4.1. For a new row m corresponding to place p_c in the unreduced network, it holds true that $c_{m,i} = -c_{m,j}$ and $\forall k : c_{m,k} = 0$. With $x_i = x_j$ it immediately follows that

$$\sum_k c_{m,k}x_k = 0$$

This implies that a mapping from x to x' and from x' to x exists induced by the reduction of $C \rightarrow C'$. □

In the analysis of the t-invariants a reversible reaction forms a *trivial t-invariant*. A reversible reaction is split into a forward and a backward reaction. Given the definition of *minimal semi-positive t-invariants*, no other t-invariant will include both transitions. Each reversible reaction forms its own trivial *minimal* t-invariant. This property gives rise to the idea of a reduction rule of this network structure which conserve the CTI property.

Definition 4.3.5. *ITP reduction:*
Given a Petri net $N = (P,T,F,w,m_0)$, an ITP $(t_i,t_j),t_i,t_j \in T$ connecting the shared places $p_l \in P$ and $p_q \in P$, and weights $w(t_i,p_q) = w(p_l,t_i) = w(t_j,p_l) = w(p_q,t_j) = 1, \{(t_i,p_q),(p_l,t_i),(t_j,p_l),(p_q,t_j)\} \in F$, the relation $r : N \rightarrow N'$ is called ITP reduction. *The reduced net $N' = (P',T',F',w',m'_0)$ is defined by*

1. $P' \leftarrow P \backslash \{p_q\}$ *(p_l and p_q are merged)*

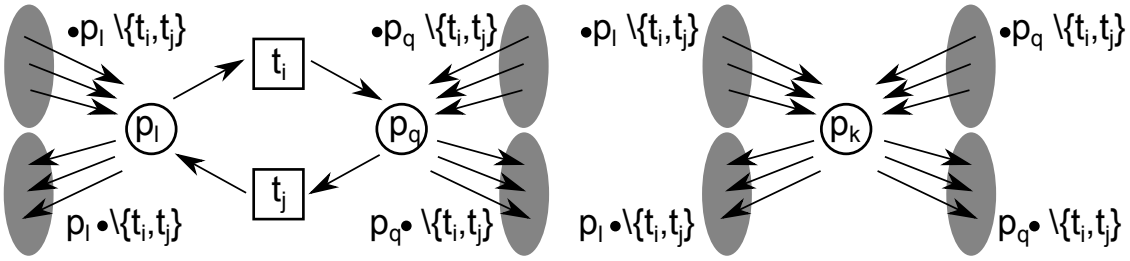
2. $T' \leftarrow T \backslash \{t_i, t_j\}$ *(t_j and t_i are deleted)*

3. $F' \leftarrow F \backslash \{(t_i,p_q),(p_q,t_j),(t_j,p_l),(p_l,t_i)\} \cup \{(t,p_l) | t \in {\bullet}p_q\} \cup \{(p_l,t) | t \in p_q{\bullet}\}$ *(all edges leading from and to p_q are removed, for each of them a new edge is added, leading from or to p_l, respectively)*

4. $w' : \begin{cases} (t,p_l) \leftarrow w(t,p_l) + w(t,p_q) & \forall t \in p_q{\bullet} \\ (p_l,t) \leftarrow w(p_l,t) + w(p_q,t) & \forall t \in {\bullet}p_q \\ e \leftarrow w(e) & otherwise \end{cases}$

5. $m_0 : \begin{cases} p \leftarrow m_0(p_l) + m_0(p_q) & p = p_l \\ p \leftarrow m_0(p) & otherwise \end{cases}$

In Figure 4.18, the ITP reduction is presented. Subfigure 4.18a unveils the structure of an ITP, while Subfigure 4.18b exhibits the network structure after the reduction of the ITP. The two transitions t_i and t_j are connected to p_l and p_q only, and all edges have a weight of 1. In the reduction step t_i, t_j, and p_q are removed from the network (as defined in Definition 4.3.5) and all edges connected to p_q are transferred to p_l.



a) ITP b) reduced ITP

Figure 4.18: Example for an ITP (a) and its reduced form (b). The grey elipses represents the rest of the network. The naming of the transitions (squares) and the places (circles) are corresponding to the names in the definition of the ITP (Definition 4.3.2). The reduction follows the rules given by Definition 4.3.5.

Lemma 4.3.2. *The relation $r : N \rightarrow N'$ is* CTI *conserving, if N' corresponds to the net N reduced by one ITP reduction.*

Analog to lemma 4.3.1 we prove this lemma by showing the equivalence of $N \in \mathbb{P}_{cti} \Leftrightarrow N' \in \mathbb{P}_{cti}$.

Proof of Lemma 4.3.2

$N \in \mathbb{P}_{cti} \Rightarrow N' \in \mathbb{P}_{cti}$**:** Given a Petri net $N = (P,T,F,w,m_0) \in \mathbb{P}_{cti}$, containing an ITP t_i, t_j and the shared places p_l, p_q, and a Petri net $N' = (T',P',F',w',m'_0)$ which is identical to the ITP reduced N.

Because N is CTI, the following must be true:

$$\exists x : \sum_k c_{l,k} x_k = 0 \wedge \sum_k c_{q,k} x_k = 0.$$

As an addition of 0 in a sum is always possible, we sum this up to

$$\exists x : \sum_k c_{l,k} x_k + \sum_k c_{q,k} x_k = 0,$$

which can be converted to

$$\exists x : \sum_{k \neq i,j} c_{l,k} x_k + c_{l,i} x_i + c_{l,j} x_j + \sum_{k \neq i,j} c_{q,k} x_k + c_{q,i} x_i + c_{q,j} x_j = 0,$$

and subsequently reformed to

$$\exists x : \sum_{k \neq i,j} (c_{l,k} + c_{q,k}) x_k + c_{l,i} x_i + c_{l,j} x_j + c_{q,i} x_i + c_{q,j} x_j = 0. \tag{4.3}$$

From the definition of the ITP follows

$$c'_{l,k} = c_{l,k} + c_{q,k}, \forall k \neq i,j \tag{4.4}$$

and

$$\begin{aligned} c_{l,i} &= -c_{l,j} \\ c_{l,j} &= -c_{q,j} \\ c_{q,j} &= -c_{q,i} \\ c_{q,i} &= -c_{l,i} \\ c_{q,i} &= c_{l,j} \\ c_{q,j} &= c_{l,i} \end{aligned} \tag{4.5}$$

The Equations 4.4 and 4.5 can be used to reform Equation 4.3 to

$$\exists x : \sum_{k \neq i,j} c'_{l,k} x_k + c_{l,i} x_i + c_{l,j} x_j - c_{l,i} x_i - c_{l,j} x_j = 0,$$

which simplifies to

$$\exists x : \sum_{k \neq i,j} c'_{l,k} x_k = 0.$$

If we interpret x without the elements x_i and x_j as the invariant vector x' of the reduced net, we have shown that $C'x' = 0$, and given the CTI inducing nature of x and can conclude that N' is CTI, if N is CTI.

$N' \in \mathbb{P}_{cti} \Rightarrow N \in \mathbb{P}_{cti}$**:** Given a Petri net $N' \in \mathbb{P}_{cti}$, which is the ITP (t_i, t_j and p_l, p_q) reduced net of N.

Given the CTI property of N', the equation

$$\exists x' : \sum_{k \neq i,j} c'_{l,k} x'_k = 0 \tag{4.6}$$

holds true. To undo the reduction and conserve the CTI property, we have to add the transitions t_i and t_j and the place p_q accordingly to the definitions of the ITP (Definition 4.3.3) and its reduction (Definition 4.3.5) in a way that $Cx = 0$ and Equations 4.4 and 4.5 hold true. This means two values x'_i and x'_j have to be added to the invariant vector x' of the reduced net and Equation 4.6 has to be transformed into two equations

$$\sum_{k \neq i,j} c_{l,k} x'_k + c_{l,i} x'_i + c_{l,j} x'_j = 0 \tag{4.7}$$

and

$$\sum_{k \neq i,j} c_{q,k} x'_k + c_{q,i} x'_i + c_{q,j} x'_j = 0 \tag{4.8}$$

with the condition (following from Equation 4.4) that

$$\exists x' : \sum_{k \neq i,j} (c_{l,k} + c_{q,k}) x'_k = 0.$$

As we cannot guarantee that

$$\sum_{k \neq i,j} c_{l,k} x'_k = 0 \wedge \sum_{k \neq i,j} c_{q,k} x'_k = 0$$

we set

$$\sum_{k \neq i,j} c_{q,k} x'_k = d.$$

With Equation 4.7 and

$$\sum_{k \neq i,j} c_{l,k} x'_k = - \sum_{k \neq i,j} c_{q,k} x'_k$$

(which follows from Equations 4.4 and 4.6) we get

$$c_{l,i} x'_i + c_{l,j} x'_j = d.$$

and

$$c_{q,i} x'_i + c_{q,j} x'_j = -d.$$

Given the equalities in Equation 4.5, both equations are identical and can be reformed to

$$\begin{array}{rcll} c_{l,i} x'_i + c_{l,j} x'_j & = & d & | c_{l,j} = -c_{l,i} \\ c_{l,i} x'_i - c_{l,i} x'_j & = & d & | + c_{l,i} x'_j \\ x'_i - x'_j & = & \frac{d}{c_{l,i}} & \end{array} \tag{4.9}$$

The extension of x' by x'_i and x'_j to x forms a mapping of $x' \to x$, if we set $x'_i = x_i$ and $x'_j = x_j$. As we can now choose arbitrary x_i and x_j, which fulfill Equation 4.9 and depend on the sum $\sum_{k \neq i,j} c_{q,k} x'_k$, we make sure that, no matter how we divide $c'_{l,k}$ to $c_{l,k}$ and $c_{q,k}$, there exists an extension of x' to x. For the case $d = 0$, $x'_i = 1$, $x'_j = 1$, and $x_k = 0, \forall k \neq i, j$ apply, forming the minimal t-invariant. □

As we have proven the CTI conserving nature of the CTP and ITP reductions, we can now conclude that a combination of the two methods, if carried out on a network step by step, also leads to a CTI-conserved reduced network.

Additionally, we show that the invariants of the reduced net are related to the invariants in the unreduced network allowing an interpretation of the invariants of the reduced network in the unreduced environment, i.e. the interpretation of t-invariants of reduced structures like e.g. the Calvin cycle becomes meaningful.

Theorem 4.3.1. CTI-*conserving reduction:*
Given a Petri net $N = (P,T,F,w,m_0) \in \mathbb{P}_{cti}$. A reduction $N' = (P',T',F',w',m'_0)$ of network N is a CTI-*conserving reduction, if only CTP reductions and ITP reductions are applied, one step at a time.*

Proof of Theorem 4.3.1. The proof follows directly from Lemma 4.3.1 and Lemma 4.3.2, which prove the CTI-conserving behavior of each reduction step with each method. □

4.3.3 Extension of the CTP Reduction

We extend the CTP reduction from the trivial to the general case allowing weights $w(t_i, p_c) \neq w(p_c, t_j)$ for each of the edges connected to the Connector. Therefore, we extend the reduction rules given in Definition 4.3.4 by an adaption of the weights of $t_i \in T'$.

Definition 4.3.6. *CTP reduction-extended:*
Let $N = (P,T,F,w,m_0)$ be a Petri net, t_i and t_j, $t_i, t_j \in T$ a CTP, and p_c the Connector of t_i and t_j. Let further be the weights $w(t_i, p_c) = a$ and $w(p_c, t_j) = b$. The PN *$N' = (P',T',F',w',m'_0)$ is the Petri net N reduced by the CTP, and defined by*

1. $P' = P \backslash \{p_c\}$ *(delete p_c)*
2. $T' = T \backslash \{t_j\}$ *(t_j is merged to t_i)*
3. $F' = F \backslash \{(p,t_j),(t_j,p) | p \in P\} \cup \{(t_i,p) | p \in t_j\bullet\} \cup \{(p,t_i) | p \in \bullet t_j\}$ *(all edges leading from and to t_j are removed, for each of them a new edge is added, leading from or to t_i, respectively)*
4. $w' : \begin{cases} (t_i,p) = \frac{lcm(a,b)}{a} w(t_i,p) + \frac{lcm(a,b)}{b} w(t_j,p) & \forall p \in t_i\bullet \cup t_j\bullet \\ (p,t_i) = \frac{lcm(a,b)}{a} w(p,t_i) + \frac{lcm(a,b)}{b} w(p,t_j) & \forall p \in \bullet t_i \cup \bullet t_j \\ e = w(e) & otherwise \end{cases}$
5. $\forall p \in P' : m'_0(p) = m_0(p)$

Lemma 4.3.3. *The relation $r : N \to N'$ is* CTI*-conserving, if N' is a CTP reduction of N.*

Proof of Lemma 4.3.3: Given a Petri net $N = (P,T,F,w,m_0) \in \mathbb{P}_{cti}$ containing an extended CTP t_i, t_j and its Connector p_c. The number of places is m and the number of transitions is n. Then it exists a row q in the incidence matrix corresponding to p_c with $c_{q,i} = a$, $c_{q,j} = -b$, and $c_{q,l} = 0, \forall l \neq i, j$. As $N \in \mathbb{P}_{cti}$, the equation

$$\exists x : \forall l \in \{1,\ldots,m\} : \sum_{h \neq i,j} (c_{l,h} x_h) + c_{l,i} x_i + c_{l,j} x_j = 0$$

holds true. Because $c_{q,l} = 0, \forall l \neq i, j$ it follows

$$\begin{array}{rcll}
\exists x : c_{q,i} x_i - c_{q,j} x_j & = & 0 & \\
a x_i - b x_j & = & 0 & \\
a x_i & = & b x_j & \\
\frac{x_i}{x_j} & = & \frac{b}{a} & \\
\frac{x_i}{x_j} & = & \frac{b' gcd(a,b)}{a' gcd(a,b)} & \\
\frac{x_i}{x_j} & = & \frac{b'}{a'} & \Rightarrow gcd(a',b') = 1 \\
a' x_i & = & b' x_j &
\end{array}$$

An obvious solution for x_i and x_j would be $x_i = b'$ and $x_j = a'$, which additionally implies that $gcd(x_i, x_j) = 1$. From the correlation $a \cdot b = lcm(a,b) \cdot gcd(a,b)$ (for derivation and proof see (Tattersall, 2005)) follows:

$$\begin{array}{rcll}
a \cdot b & = & lcm(a,b) \cdot gcd(a,b) & | : b \\
a & = & \frac{lcm(a,b) \cdot gcd(a,b)}{b} & \\
a' \cdot gcd(a,b) & = & \frac{lcm(a,b) \cdot gcd(a,b)}{b} & | : gcd(a,b) \\
a' & = & \frac{lcm(a,b)}{b} & \\
b' & = & \frac{lcm(a,b)}{a} & analog \\
x_i & = & \frac{lcm(a,b)}{a} & with\ x_i = b' \\
x_j & = & \frac{lcm(a,b)}{b} & with\ x_j = a'
\end{array} \tag{4.10}$$

The minimal possible value of x_i and x_j in the vector x are then $x_i = \frac{lcm(a,b)}{a}$ and $x_j = \frac{lcm(a,b)}{b}$. That means it exists a $k \in \mathbb{N}$ for each x in the set of the minimal invariants for that the equation

$$k \cdot \frac{lcm(a,b)}{a} = x_i \wedge k \cdot \frac{lcm(a,b)}{b} = x_j \tag{4.11}$$

is always true. k is based on the connection of t_i and t_j to p_c. Of course, it is possible that other paths leading from t_i, t_j, or both, restricting this k to subsets of $\mathbb{N}$. This does not change the overall behavior of the relation between x' and x.

We now have to show that $Cx = 0 \Leftrightarrow C'x' = 0$. The reduction of the incidence matrix can be summed up to

$$\forall l \in \{1,\ldots,m\} \setminus \{q\} : c'_{l,i} = c_{l,i} \frac{lcm(a,b)}{a} + c_{l,j} \frac{lcm(a,b)}{b} \tag{4.12}$$

If the reduced network is CTI, the equation

$$\forall l \in \{1,\ldots,m\} \setminus \{q\} : \sum_{g \neq i,j} (c_{l,g} x_g) + c'_{l,i} x'_i = 0$$

has to be true, with $x'_i \neq 0$ (each $x_g, \forall g \neq i, j$ is not 0 by definition). With Equation 4.12 it reforms to

$$\forall l \in \{1,\ldots,m\} \setminus \{q\} : \sum_{g \neq i,j} (c_{l,g} x_g) + (c_{l,i} \frac{lcm(a,b)}{a} + c_{l,j} \frac{lcm(a,b)}{b}) x'_i = 0$$

and with $x'_i = k$ results in

$$\forall l \in \{1,\ldots,m\} \setminus \{q\} : \sum_{g \neq i,j} (c_{l,g} x_g) + (c_{l,i} k \frac{lcm(a,b)}{a} + c_{l,j} k \frac{lcm(a,b)}{b}) = 0 \tag{4.13}$$

With Equation 4.11 follows from Equation 4.13

$$\forall l \in \{1,\ldots,m\} \setminus \{q\} : \sum_{g \neq i,j} (c_{l,g} x_g) + c_{l,i} x_i + c_{l,j} x_j = 0.$$

This shows a strong relation between x' and x with all elements $x' \in I'$ can be found in $x \in I$. The row q is the source of a and b and therefore always sums up to 0, if we reduce the incidence matrix in the way defined by Definition 4.3.6. □

This extended reduction allows the condensation of reaction chains merging metabolites from (or splitting them into) two or more identical compounds. Such reactions appear to be present in most biochemical networks.

4.3.4 Comparison to Other Reduction Methods

We compare our methods to three other approaches in the literature (Lee-kwang et al., 1987; Shatz et al., 1996; Ravasz et al., 2002):

- Lee-kwang et al.: This study defines several *generalized reducible subnets* (GRSN) and provides respective reduction rules. Our ITP reduction, CTP reduction and extended CTP reduction are extensions of the reduction of a GRSN. The so-called GRSN-1T is related to the CTP reduction, and the GRSN-2P is comparable to the ITP reduction.

  GRSN-1T applies stricter rules concerning the connections of t_i and t_j to either p_c and the rest of the network, allowing a reduction only if the weights are equal or a common multiple of a integer k. Our CTP reduction does not require any rules concerning the rest of the network and the extended CTP does not require restrictions on the involved edge weights.

  GRSN-2P enables a reduction only in cases where all edges leading to and coming from the two involved places have the same weights. Our ITP reduction does not restrict edge weights except the weights of the edges to and from the connecting transitions.

- Shatz et al.: This work proposes another approach related to the CTP reduction. Two transitions may be combined into one which gets all edges of these two if certain conditions concerning a connecting place p are met:

  1. there exists only one place p connecting the two transitions,
  2. p has no other connections,
  3. p is the only pre-place to the first transition and the only post-place to the second transition.

  Our CTP reduction does not restrict the number of pre-places and post-places of the two transitions or the number of connecting places.

- Ravasz et al.: A reduction of metabolic networks is proposed which reduces unbranched reaction chains. This is the basic principle of our CTP reduction as well.

4.3.5 Complexity Aspects of the Reduction Method

A reduced network has a significantly shorter computation time for complex tasks like the determination of the t-invariants. While the computation of the t-invariants of the complete network runs for months without any result, the determination of the t-invariants of the reduced network is completed in the timespan of a few seconds. Unfortunately, this reduction of computation time does not mean a reduction of the computational complexity of the t-invariant calculation. Considering the reduction process as a folding of reactions onto themselves, the unfolding process of a formerly reduced network creates the original unreduced network. Similarly, a partial unfolding of a reduced network leads to an unreduced subnet, nestling between reduced parts of the reduced network. Due to other affected transitions in the reduction process (see Section 4.3.6), each of these unreduced subnets can form a significantly more complex structure than the reduced part. This means circles of several consecutive transitions can be destroyed, rendering a reverse mapping of the reduced invariants to the unreduced invariants incomplete, and necessitating a recalculation of the t-invariants in unfolded regions of the network. Therefore the determination of the t-invariants of the unreduced network from the t-invariants of the reduced network remains a challenging task.

4.3.6 Application to the *Arabidopsis thaliana* Model

We apply the reduction method described in Section 4.3.2 to our Petri net model of *Arabidopsis thaliana*. We shortly inspect the similarities and differences between the algorithm used here (further called work-algorithm) and the algorithm published earlier (further called paper-algorithm, Ackermann et al., 2012). The paper-algorithm implementation and the tests published in the paper are done by our coworkers.

Both algorithms base on the same ITP and CTP reduction rules, the work-algorithm additionally affect parallel structures. As the chronological order of the ITP and CTP identification and reduction can affect the reduction result, it may become difficult to compare different reduction

conductions of the same network. Both algorithms have a similar search order of identifying reducible structures. The search order is: CTP, ITP, and parallel (work-algorithm only). If any of these structures are encountered, it is reduced and the next search step starts with the search for a CTP. If no parallel (ITP for the paper-algorithm) reduction is found, the algorithm terminates.

A special case of an CTP is a structure that as well forms an ITP. This is the fact, if a place is connected to the rest of the network only via an ITP-forming transition pair. In that case, the work-algorithm reduces the ITP first.

A special case of ITP is given, if a certain metabolite is connected to an Input and an Output. Contrary to the paper-algorithm the work-algorithm does not reduce such structures, while the paper-algorithm removes these metabolites and all the edges connected to them. In any case that does not change the CTI-conservativeness. The network described here contains several external metabolites which fulfill these special ITP rules. The work-algorithm does not remove these external metabolites, as this treatment possibly has a negative effect on the biological interpretability of results obtained for the reduced network.

To perform the reduction of the PN described here the work-algorithm is used. Table 4.8 shows the efficiency of the reduction by presenting the number of metabolites, reactions, and edges before and after the reduction. We analyze the difference between the reduced network and the unreduced by comparing the number of metabolites, reactions, and edges. Additionally we measure the efficiency of the reduction by the relative amount of removed metabolites, reactions, and edges.

Table 4.8: Table of reduction efficiency. The number of metabolites, reactions and edges are presented, for the unreduced and for the reduced network. Difference is $value_{\text{unreduced}} - value_{\text{reduced}}$, and reduction efficiency is calculated by $eff = \frac{difference}{value_{\text{unreduced}}}$.

	Number of metabolites	Number of reactions	Number of edges
unreduced net	134	243	572
reduced net	60	131	329
difference	74	112	243
efficiency	≈ 0.55	≈ 0.46	≈ 0.42

86 CTP, 62 ITP, and 2 parallel reduction steps are performed. The reduction has an efficiency of around 50%. Hereby, the efficiency of the metabolite reduction is greater than the one of reaction and edge reduction.

All reductions of our algorithm are presented in Table C.1. To perform a traceable reduction, each reduction step changes the naming of reactions and metabolites in the network. If two reactions are merged in a CTP reduction, the newly merged transition gets 'ctp(name$_1$+name$_2$)' as a new name. 'name$_1$' and 'name$_2$' are the names of the transitions t_i and t_j, respectively. If, e.g., the two reactions 'E151' and 'E152' connected by metabolite '87' are identified as a potentially reducible CTP, the algorithm performs multiple steps:

1. connect all edges from reaction 'E152' to reaction 'E151'
2. remove metabolite '87'

3. remove reaction 'E152'

4. rename reaction 'E151' to 'ctp(E151+E152)'

If the reduction is an ITP reduction, the merged metabolites are renamed. As for the CTP reduction the names of both metabolites are connected by a '+', embraced by round brackets, and the emerging name-construct is preceded by an 'itp'. The reaction or metabolite chosen for renaming (and thereby the reaction or metabolite which gets deleted) depends on an internal identification number and seems totally random from the outside. This has no effect on the functionality of the process. If, e.g., the reactions 'E24_f' and 'E24_b', which connect the metabolites '76' and '50', are reduced by an ITP reduction, the algorithm will carry out four steps:

1. connect all arcs from metabolite '50' to metabolite '76'

2. remove reaction 'E24_f' and 'E24_b'

3. remove metabolite '50'

4. rename metabolite '76' to 'itp(76+50)'

Additionally parallel reactions are reduced, i.e. if two transitions have the same input place and the same output place, and the same weights on corresponding edges both transitions can be combined into one without changing the net behavior (Murata, 1989). This rule is adopted for the analysis of biochemical networks (Reddy et al., 1993). We use this technique to further reduce the complexity of our network by reducing parallel pairs of transitions. We constrain this parallel reduction to pairs of transitions with one pre-place and one post-place only, and all four edges having a weight of 1.

The reduction procedure is recursive. It is possible that, e.g., the place 'itp(76+50)' can be identified as an ITP with another place in the next step. Each step of the reduction is followable by this naming practice and produce a timescale of reductions.

Our algorithm can remove edges from other transitions than the chosen ITP in an ITP reduction, and from other places than the chosen CTP in a CTP reduction. Both of this third-party transitions and places are not directly involved in the reduction process. This is possible, if the transitions of the CTP are connected by more than one place. In such a situation one of these places, which is connected to t_i and t_j with weight 1, is randomly chosen to be the Connector. After the reduction all of the other potential Connectors would be connected by an ingoing and an outgoing edge to the newly reduced transition t_i'. To conserve the pureness property of the network we adjust the weight of the edges connecting these other potential Connectors to t_i and t_j instead of adding an edge of the opposite direction:

$$w'(p,t_i') \leftarrow \begin{cases} w(t_i,p) - w(p,t_j) & ,if\, p \in t_i\bullet \cap \bullet t_j \\ w(t_j,p) - w(p,t_i) & ,if\, p \in t_j\bullet \cap \bullet t_i \end{cases}$$

An edge of weight 0 is removed from the network. If $w'(p,t_i) < 0$, the direction of the edge is inverted. The treatment of transitions connected to p_i and p_j in an ITP is analog. There is no case

of an inverted edge-direction in the reduction process of the PN.

A place or transition that has no connections left is removed from the network. Note that it is possible to remove the just reduced CTP (i.e. the transition $t'_i \in T'$), if the transition pair t_i, t_j is additionally a reversible reaction for all other places connected to it. If this situation appears, each ingoing edge for each connected place would cancel an outgoing edge of the same place, and therefore cancel all edges from and to $t'_i \in T'$. After an ITP reduction it is possible that another ITP is removed, if it connects the same p_i and p_j. The reduction of the first ITP results, for every other potential ITP, in two transitions connected to the reduced place $p'_i \in P'$ with an ingoing and an outgoing edge. These transitions are then removed as they are no longer connected to the network. Comments on removed isolated places or transitions are given in Table C.1.

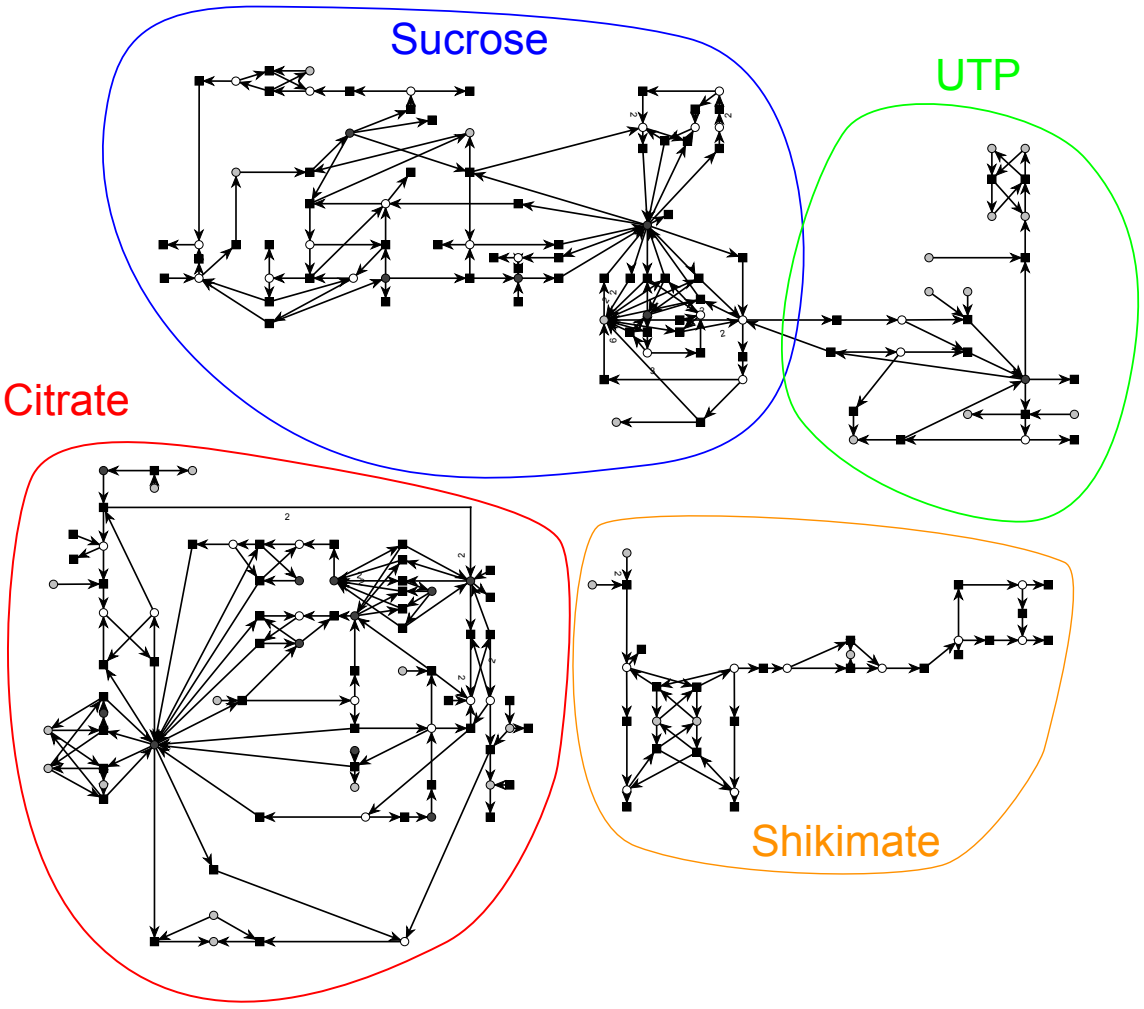


Figure 4.19: Reduced Network. The regions corresponding to the subnets shown in Figure 4.1 are highlighted in the same color and marked with the name of the subnet.

Figure 4.19 depicts the network after performing the reduction steps listed in Table C.1. The regions of the reduced net are marked by color and by name corresponding to the subnets emphasized in Figure 4.1.

We exemplary demonstrate the interpretation of the results of the reduction procedure, choosing a place following from a series of ITP reductions. The place represents a conglomeration

of substances involved in the Calvin cycle, the glycolysis, and the citric acid cycle. It combines the compounds succinate (metabolite '4'), fumarate (metabolite '6'), malate (metabolite '8'), oxalacetate(metabolite '12'), pyruvate (metabolite '20'), phospho*enol*pyruvate (metabolite '34'), glycerate 2-phosphate (metabolite '39'), glycerate 3-phosphate (metabolite '40'), glycerate 1,3-bisphosphate (metabolite '41'), glyceraldehyde 3-phosphate (metabolite '42'), and dihydroxyacetone phosphate (metabolite '68') into the same place. This reduction steps additionally removes the transitions 'E13', 'E14', 'E15', 'E90', and 'E91', while the transitions 'E11' and 'E14' remain in the reduced network.

The first four metabolites are part of the citric acid cycle. Out of the remaining seven metabolites which are part of the glycolysis, there are five, namely glycerate 2-phosphate (metabolite '39'), glycerate 3-phosphate (metabolite '40'), glycerate 1,3-bisphosphate (metabolite '41'), glyceraldehyde 3-phosphate (metabolite '42'), and dihydroxyacetone phosphate (metabolite '68'), which are additionally part of the Calvin cycle.

This result biologically hard to interpret. In plastids of *Arabidopsis thaliana* the glycolysis and the Calvin cycle share several enzymes (Peltier et al., 2006): phosphogylcerate kinase (transition 'E90', reversible), glyceraldehyde 3-phosphatedehydrogenase (transition 'E91', reversible), triosephosphate isomerase (transition 'E13', reversible), seduheptulosebisphosphate aldolase (transitions 'E11' and 'E14', both reversible), and fructose bisphosphatase (transition E15, reversible by transition E16). The sharing of these reactions suggests that the participating compounds are shared as well. In this case the shared metabolites would be: glycerate 3-phosphate (metabolite '40'), glycerate 1,3-bisphosphate (metabolite '41'), glyceraldehyde 3-phosphate (metabolite '42'), D-fructose 1,6-bisphosphate (metabolite '43'), D-fructose 6-phosphate (metabolite '44'), and dihydroxyacetone phosphate (metabolite '68'). Besides D-fructose 1,6-bisphosphate (metabolite '43') and D-fructose 6-phosphate (metabolite '44'). The absence of D-fructose 1,6-bisphosphate (metabolite '43') and D-fructose 6-phosphate (metabolite '44') in the reduced place is caused by restrictions in the reduction process. A merging of places by an ITP reduction is only allowed if all involved edges have the weight 1. This is, due to the added weights of edges throughout the reduction process, not the case for the reactions connecting metabolite '43' with '42' and '68' (transition 'E14').

The merging of parts of the citric acid cycle, namely succinate (metabolite '4'), fumarate (metabolite '6'), malate (metabolite '8'), and oxalacetate (metabolite '12'), is inspired by the reversible reactions between the compounds which is the main idea behind an ITP reduction. The combination of this part of the citric acid cycle and the last steps of the glycolysis, producing phospho*enol*pyruvate and pyruvate, is induced by the synthesis pathway of oxalacetate from phospho*enol*pyruvate (Dey and Harborne, 1997).

Considering these biological aspects, the reduction process does not seem so unfeasible.

In this connection it is worth to mention that the reduction process is a theoretical approach, whose main function is to conserve the CTI property. In general it is difficult to interpret the reduction results in a biological sense. We have shown that a mapping of reduced invariants to invariants of the unreduced net is possible, but we are far from providing a method to unfold these reduced invariants to their unreduced versions. Additionally to the exemplary interpretation of

parts of the reduced network, in Section 4.4.2 a discussion of the invariants of the reduced and the unreduced networks is performed.

4.4 Network Verification

Network verification or validation is a crucial part of the process of model construction. Certain structural and dynamical properties are supposed to be important to validate a model, especially for biological Petri nets structural and behavioral properties like *liveness* and *connectedness* are of importance (Heiner and Koch, 2004; Koch et al., 2011). A *live* network cannot reach a system state in which a part of the network is not longer able to produce or consume metabolites due to missing substances required for the reactions in this *dead* part of the network.

Another important property of a valid network is the biological interpretability of the t-invariants (Koch et al., 2005; Heiner et al., 2004; Grunwald et al., 2008), or their clusters (Grafahrend-Belau et al., 2008) and MCT-Sets (Sackmann et al., 2006). A result of the t-invariant analysis is the determination of the CTI property. A network is CTI, if all its transitions are covered by at least one t-invariant. Such networks are also called *consistent*. It is of particular importance, that a biological network is CTI. This property ensures that each reaction may contribute to the basic system behavior (Koch and Heiner, 2007) while the steady-state of the system is preserved.

It was not possible to verify the aforementioned previously constructed network (Nöthen, 2009). The liveness analysis used in that study depends on the *Integrated Net Analyzer* (INA, Starke, 2003). This tool utilizes the reachability graph if the reachability graph is available and otherwise perform a test investigating the reactions which are not dead in the initial marking. The calculation of the t-invariants manner was impossible, and results obtained for a biologically reduced form of the net, which was not CTI-conserving, were obviously erroneous.

Taking this into account we provide a network verification in the following subsections. The verification approach is based on a newly developed algorithm for liveness testing and on a inspection of the t-invariants. The t-invariants are calculated for the provenly CTI-conserving network reduction and for the network decompositions, and the obtained results are used in combination to provide a meaningful analysis.

4.4.1 Dynamical and Structural Properties

The three behavioral properties *liveness*, *boundedness* and *reversibility* are independent of each other (Murata, 1989), and all of them are computationally NP-hard (Esparza, 1998) to determine, as their calculation depends on the computation of the reachability graph. The reachability graph grows exponentially in the number of enabled transitions and possible markings (Murata, 1989). Furthermore, it has been demonstrated that the computation of the reachability graph is EXPSPACE-hard for general Petri nets (Esparza, 1998). If the network has transitions which are always enabled (i.e., input transitions), the size of the reachability graph is infinite. To circumvent this problem the *coverability graph* is introduced (Murata, 1989) which can deal with infinite tokens on a place, but has the same computational complexity as the reachability graph.

As the computation of the reachability graph is unfeasible for our network, and therefore the computation of boundedness and reversibility is unfeasible as well, we focus on the liveness (for its definition see Section 3.2.1) property only, which is proposed to be essential for biochemical networks (Heiner and Koch, 2004). To prove the liveness of our network we follow a new approach demonstrated in Section 4.4.1.1. A network is live if every transition can be activated after some firing sequence, starting with all input transitions activated. This method can only lead to a positive ('the Petri net is *live*') result, if the network has input transitions. This positive result is the only reliable one. There is no information about liveness if the algorithm does not return the *live* result.

Networks with Inputs are in general unbounded (Koch and Heiner, 2007), which indicates that our model is not bounded.

Several structural properties are defined for PN (Koch and Heiner, 2007) like *pureness*, *homogeneity*, and the number of *static conflicts*. We test and inspect these properties and present the results in Section 4.4.1.2.

4.4.1.1 Liveness

The liveness problem is recursively equivalent to the reachability problem (the question whether a certain marking is reachable from a given initial marking), and its complexity is still unknown (Esparza and Nielsen, 1994).

There exist polynomial-time algorithms (PTA) for liveness checking. They are based on the precondition of boundedness in free-choice Petri nets (Esparza and Silva, 1992; Kemper and Bause, 1992), and depend on the question 'are the siphons of the net controlled?' (Barkaoui and Pradat-Peyre, 1996). Another PTA for the liveness test of bounded PN is based on the determination of t-invariants (Lautenbach and Ridder, 1994). As our net is unbounded (Koch and Heiner, 2007) and not free-choice (meaning each place has either only one post-transition or is the only pre-place to its post-transitions), we perform a strongly modified coverability analysis to show the liveness of our network.

The main idea of our approach to test the liveness property of the PN is the partial creation of a boolean coverability graph. A coverability graph is a version of the reachability graph that can deal with infinite token values (Murata, 1989). Our boolean reachability graph has two possible states for every place, *active* (*has tokens*) and *inactive/not active* (*does not have tokens*). The actual amount of tokens on a place is not considered.

Description of the Boolean Reachability Algorithm: We refer to each call of the `while` loop as a *step*. In the initialization phase ahead of the `while` loop, the three variables of the algorithm get their initial load (Algorithm 4.4.1, line 2 - 4):

- `visitedT` is the set of *visited transitions* and initialized as empty set (line 2)
- `activeT` is the set of all *active transitions* of the network, and gets the set of all input transitions (transitions with no pre-places, line 3) as initial value

Algorithm 4.4.1 Boolean Reachability Algorithm. The algorithm tests whether there exists a sequence of firing transitions that facilitate a marking with $m(p_i) > 0, p_{\in}H$ of a set of places H. For our test H corresponds to the set of all places P of the PN. It might be restricted to a smaller set and then tests, if a certain set can be reached by a substance flow from the input reactions. The algorithm returns the set of places for which a marked state in the reachability graph exists, and a set of transitions, which are activated if each of the determined places possess tokens. If this set of returned places is P, the net is live. The algorithm does not require the weights and markings of the PN.

Input: Petri net (P,T,F).
Output: A list of all reachable transitions and places.

procedure POSTREACH(P,T,F)
  $visitedT \leftarrow \emptyset$ ▷ initialize the Set of visited transitions
  $activeT \leftarrow \{t \in T | (p,t) \cap F = \emptyset\}$ ▷ init of active transitions
  $activeP \leftarrow \{p \in P | (t,p) \in F \wedge t \in activeT\}$ ▷ init of active places
  while $activeT \neq \emptyset$ **do**
    $t \leftarrow$ arbitrary element from $activeT$
    $activeT \leftarrow activeT \setminus t$ ▷ remove visited from active transitions
    $visitedT \leftarrow visitedT \cup t$ ▷ update visited transitions
    $activeP \leftarrow activeP \cup \{p \in P | (t,p) \in F\}$ ▷ update active places
    for all $p \in activeP$ **do**
      for all $\{t_x | (p,t_x) \in F\}$ **do**
        if $t_x \notin visitedT$ & $t_x \notin activeT$ **then**
          if $\forall p1 \in \{p1 \in P | (p1,t_x) \in F\} : p1 \in activeP$ **then**
            $activeT \leftarrow activeT \cup t_x$ ▷ if all preplaces are active, add transition to active transitions
          end if
        end if
      end for
    end for
  end while
  if $visitedT = T \wedge activeP = P$ **then**
    print: The PN is live.
  end if
end procedure

- `activeP` is the set of all *active places* and gets initialized with all places which receive tokens from transitions in `activeT` (line 4)

Each step of the the algorithm chooses one transition t to perform its test (line 6), remove t from the set of active transitions (`activeT`, line 7), and add it to the set of visited transitions (`visitedT`, line 7). Afterwards, the algorithm updates the set of active places (`activeP`) in such a way that it now considers the places that are activated by the chosen transition t (line 9). Now the update of the set of active transitions starts. It tests for all active places (line 10), whether all of their post-transitions (t_x, line 11) are now active (line 13), if they were not already visited or active (line 12). If this test is positive, t_x is added to the set of active transitions (line 14). We assume that a transition t_i cannot be part of any set (`activeT`, `visitedT`) more than once. The same holds true for places and their set (`activeP`).

Proof of correctness: We prove that the algorithm always terminates on a finite Petri net and that, if the result is *the network is live*, the result is correct. We use a simplified formulation saying 'a place p cannot be reached by tokens' which means 'it is not known whether a sequence of firing transitions exists, that can create a marking in which the number of tokens on p is greater than zero'. The meaning of 'p can be reached by tokens' is analog to 'it is known that a sequence of firing transitions exists, that can create a marking in which the number of tokens on p is greater than zero'.

In each step each transition t has three possible states:

1. t is not active,
2. t is active ($t \in$ `activeT`),
3. t was active in an earlier step ($t \in$ `visitedT`).

Each place p has two possible states:

1. p is not activating (cannot be reached by tokens yet),
2. t is activating ($p \in$ `activeP`, can be reached by tokens).

Termination: Every place, once activated (by any transition in any step of the algorithm (line 9)), will remain active and will never be removed from `activeP`.

The three different states of the transitions can only be traversed in increasing order (from state '1' to state '3'). Every transition is in state '1' at the beginning of the algorithm. Then the input transitions are added to the set of active transitions, and the places activated by them are added to the set of active places (lines 3 and 4). If after the initialization step no transition is part of `activeT`, the algorithm terminates (immediately).

As long as the set of active transitions contains elements one of them will be removed in each step (lines 6 and 7). If no transitions are added in any step of the algorithm, it will run one step for every transition added in the initialization phase and terminate afterwards.

The situation is more complicated if transitions are added throughout the steps of the algorithm. A transition can only be added to `activeT`, if it is not in `visitedT` (line 12) and if all its pre-places are included in `activeP` (line 13). Each transition can only remain in the set of active transitions (`activeT`) until it is chosen to activate its post-places (line 6). If a transition is chosen, it is added to the set of visited transitions (`visistedT`, line 8). This means a transition can only be added once to the set of active transitions and the same transition t can only occur once at a time in the set. Taken together, the set of active transitions can grow, but its size is limited by $|T|$ as upper bound. Due to the fact that one transition is removed from `activeT` in each step, and `activeT` has a maximal size, the algorithm will terminate after maximal $|T|$ steps. □

Correct result: As the algorithm always terminates on a finite Petri net, we argue that the answer 'PN is live' is correct.

Every active transition marks every post-place as an *activating place* (line 9) in the step it is chosen and is added to the set of visited transitions. Next, the algorithm updates the active transitions. It cycles through every active place (line 10) and tests every of its post-transitions t_p (line 11). This test checks whether all pre-places of an arbitrary t_p are activating (line 13). If this is the case, t_p is added to the set of active transitions (line 14).Taken together the algorithm tests whether a transition can be activated by an arbitrary firing sequence of a certain subset of the transitions in `visitedT`. As *live* means that such an arbitrary firing sequence of transitions exists which enables the tested transition, the result 'PN is live' is correct. □

Note that, if the algorithm does not answer 'PN is live', the liveness of the PN is undecided and the network can still be live.

Running time: The worst case network for the algorithm is a completely connected PN containing one input transition. Each place is everything, a pre-place and a post-place of every transition, and a post-place of the single input transition. This means, in the very first update step of the algorithm every place is added to the set `activeP` in $O(|P|)$. Now for each transition t the update process is started, requiring $O(|T|)$ steps. Each step includes two loops running through post-transitions of all active places in $O(|P| \cdot |T|)$, together resulting in an overall worst-case running time of $O(|P| + |P| \cdot |T|^2)$.

Comparison to other methods: In general algorithms determining the reachability graph consider the weights of the edges (Murata, 1989). Our algorithm does not take the weights of the edges into account, because if a token can reach a place once, the same firing sequence can provide an arbitrary number of tokens to this place. It ignores the initial marking as well, because the algorithm is not able to handle a fixed amount of tokens on a place and therefore could falsely assume transitions to be active.

The algorithm is comparable to a partial boolean coverability graph determination. A coverability graph enumerates the different possible states of token distributions (each possible marking equals a state) like the reachability graph. Unlike the reachability graph, the coverability graph can handle an infinite amount of tokens on a place (Murata, 1989). The coverability graph algorithm

performs a state transition, meaning it fires a transition. This is comparable to the choosing of a transition and the following update of the current state in Algorithm 4.4.1. If a place which already contains tokens gains one or more new tokens in the coverability graph calculating algorithm, the amount of tokens on this place is set to infinite. Compared to this our algorithm ignores the state of 'has a fixed amount of tokens' for a place completely.

Furthermore, our algorithm does not enumerate the complete state space. If we consider two different places p_1 and p_2 we have four different possibilities in our state space:

1. p_1 and p_2 have no tokens,
2. p_1 has tokens and p_2 has no tokens,
3. p_1 has no tokens and p_2 has tokens,
4. p_1 and p_2 have tokens.

Contrary to our algorithm the reachability graph algorithm as well as the coverability graph algorithm enumerate and buffer each different state. Compared to the four states mentioned above state '2', '3', and '4' would need to be modified in such a way that the amount of tokens are considered. In the following each state can lead to every other state, e.g., different state '2' situations (different amounts of tokens on p_1) can lead to every other state ('1', '2', '3', '4'), which is not the case using our algorithm. By this behavior of the reachability graph algorithm and the coverability graph algorithm their state space is exponentially increased, as each possible state imply a set of new states, which in turn have to be saved and investigated.

Our algorithm considers only the final state, which can be any of the four states, and ignores all intermediate states required to reach this state. This is done in raising order. The algorithm starts with state '1' for the exemplary set of places. From this it can only reach the states '2', '3', or '4'. If state '2' or '3' is reached, the only possible state change is the change to state '4'. A certain amount of tokens is not considered.

Only if state '4' is hit at the termination point, the algorithm can possibly give the result 'PN is live'. This is entirely sufficient to test the liveness property, but it is not comparable to a full line reachability or coverability analysis.

The given method is a special case which can test a network with input transitions on liveness, and its result is only reliable if the algorithm result is 'PN is live'. Otherwise it is not possible to decide whether the network is live or not. Furthermore, the algorithm detects structural liveness as its liveness test does not consider any certain initial marking.

Test of the *Arabidopsis thaliana* Petri net: Our network is completely active, if the set of input transitions is considered as starting point of the algorithm. This means the algorithm marks every place and transition as reachable by a finite sequence of transitions firing. Figure 4.20 shows the results of the algorithm. Every transition of our network can be activated from a certain firing sequence of transitions and all places are activating their post-transitions. The initial assignment of `activeT` are all input transitions, marked in blue, while all possible active transitions (which are the *live* transitions), as well as the activating places, are marked in green.

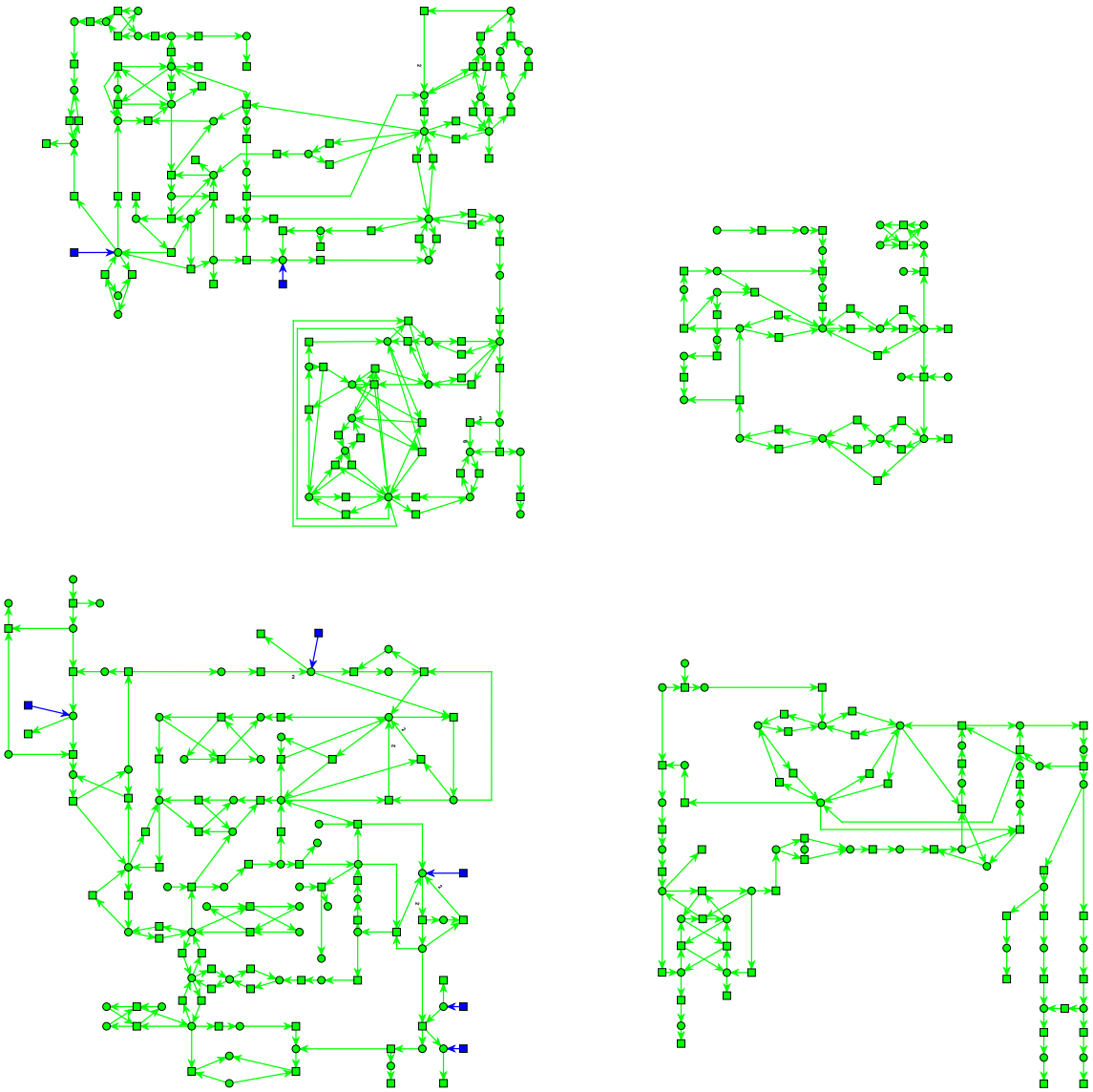


Figure 4.20: Liveness-test of our *Arabidopsis thaliana* model. The topology is the same as in Figure 4.1. All reachable places and transitions are colored in *green*, all activated input transitions are colored in *blue*. The algorithm we use is presented in Algorithm 4.4.1. Before the first step of the computation, only the input transitions are activated.

4.4.1.2 Structural Properties

Table C.2 lists all static conflicts of the network. Metabolic Petri nets are typically not free of static conflicts (Koch et al., 2005). In our network we find a total of 630 static conflicts. This number seems quite high compared with the number of overall transitions (243). We have to take into account that the real number of transition pairs sharing pre-places is 315, as for each transition A, which is in conflict with a transition B, transition B is in conflict with transition A as well. Nevertheless, we have more static conflicts than reactions which suggests a well-interwoven network structure.

The network we constructed is not *ordinary*, not *homogeneous*, and not *conservative*. As a matter of fact, it is difficult to create a metabolic petri net that fulfills these three properties, because

- an *ordinary* network would not be able to combine two molecules of the same compound, or dissolve a polymer, and in general no substance would be allowed to have a stoichiometry in any reaction greater than one.
- an *homogeneous* network cannot contain metabolites which are part of reactions of different complexity. An involvement of a metabolite in two reactions with a different stoichiometry in each of the two reactions is not possible.
- a *conservative* network has the same problem as the *ordinary*. If a molecule dissolves into two different molecules, the network is not *conservative* any more.

On the other hand, our network is *connected*, but not *strongly connected*. The latter is not possible as our net contains output transitions, because there exist no edges leading from an output transition to a place and therefore no directed path can be found from an OUT to the rest of the net. A metabolic network should be connected, but must not be strongly connected (Koch et al., 2005).

A PN is *pure*, if there do not exist any two nodes, which are connected by two edges, one in each direction (Sifakis, 1978). This structure is normally referred to as *read-arcs*. Read-arcs can influence a further interpretation of t-invariants (Sackmann et al., 2006), because t-invariants are computed from the incidence matrix, and read-arcs cannot be detected from the incidence matrix. To prevent this problem the term of *feasible* t-invariants is introduced (Sackmann et al., 2006). These invariants exclude t-invariants with read-arcs to places which get no tokens in the run of the t-invariant. In our case, the analyzed network is pure. This means all computed t-invariants of our network are feasible, and therefore can be investigated concerning their biological meaning (Sackmann et al., 2006).

4.4.2 Invariant Analysis

In the process of analyzing and validating a model with the help of t-invariants the result can be elusive after all and a complete analysis can be impossible. The problem of enumerating all minimal t-invariants can not be solved in *polynomial total time*. The overall complexity of this task

Table 4.9: Number of invariants per module and reduced net. The t-invariants are grouped according to the environmental connection they inherit. *IN + OUT* invariants contain input and output transitions, *IN only* contain only input, *OUT only* contain only output transitions. *Cyclic* t-invariants contain neither IN nor OUT. *Trivial* invariants are reversible reactions which are split into a forward and a backward transition in Petri nets. This structure forms a minimal t-invariant.

invariant	Decomposition				Reduction
type	biological		automatic		automatic
	Sucrose module	*Citrate module*	Graph A	Graph B	reduced network
all	4602	3214	2286	1966	27646
trivial	31	25	28	28	22
IN + OUT	4473	3140	2226	1851	26095
IN only	18	41	18	78	1298
OUT only	63	0	0	0	132
cyclic	17	8	14	9	99
CTI	yes	yes	yes	yes	yes

remains unknown and the decision problem, whether two transitions occur in the same t-invariant, is NP-complete (Acuña et al., 2010).

Our complete model seems to be too complex for a computation of the t-invariants. Several weeks of computation time on an AMD® Opteron™ 2.2 GHz with 32 GB RAM do not lead to any result.

To provide sets of t-invariants for further analysis, we decompose the network in two different ways, and introduce and extend a reduction technique for Petri nets, which conserves the CTI property, as described in Sections 4.2 and 4.3, respectively.

Table 4.9 presents the results of the calculation of the t-invariants. The t-invariants are computed separately for each of the decompositions and the reduction of the net, and subdivided into different types. The row 'all' states the complete number of invariants, which is the sum of all different types of t-invariants. Each module obtained by the different decompositions and the reduced network are covered by t-invariants. Note that the *Sucrose module* of the biologically driven decomposition and Graph A of the automatic decomposition share a lot of transitions, and the *Citrate module* shares a lot of transitions with Graph B (compare Section 4.2.3).

In the following, we provide an invariant analysis of the t-invariants of our network. We conduct some general analysis completed by a biological interpretation of the MCT-Sets. Additionally we present a new method, the *shared invariant sets*, to

- explain groups of calculated steady-state pathways with regard to their biological interpretation,
- extend the analysis of MCT-sets to a further biological meaning.

4.4.2.1 General Invariant Analysis

The types of t-invariants are motivated by input and output transitions. For the decompositions no difference is made between Import and Input as well as Export and Output. The different types of t-invariants are:

- trivial: A reversible reaction, which is split into a forward and a backward transition in PN, forms a *trivial* t-invariant.
- IN + OUT: t-invariants which contain at least one input transition and one output transition.
- IN only: t-invariants containing no output transition.
- OUT only: t-invariants containing no input transition.
- cyclic: t-invariants containing neither input nor output transitions.

The explanations for some of the different types of invariants are intuitive. In a metabolic Petri net containing reversible reactions it is not possible to avoid trivial t-invariants. The condition of minimality ensures that no other invariant contains both of the transitions of the reversible reaction.

Invariants containing IN + OUT can basically be seen as pathways through the network, a succession of consecutive biochemical reactions transforming given educts (metabolites produced by IN) to the corresponding products (metabolites consumed by OUT).

Invariants containing IN only or OUT only, or no IN and OUT at all (cyclic), seem to be thermodynamically impossible. These invariants are artifacts emerging from the deletion of secondary metabolites, which implies these invariants include hidden exports and imports.

All t-invariants containing OUT only, which exist only in the *Sucrose module*, can be explained easily. Our model does not include secondary metabolites. In this case an important, not modeled substance is carbon dioxide. In the Calvin cycle (Bassham et al., 1954; Calvin, 1956) carbon dioxide is bound to ribulose-1,5-biphosphate by ribulose-1,5-biphosphate carboxylase oxygenase (Raines, 2003; Roy and Andrews, 2004; Parry et al., 2003). This process produces two trioses from one pentose, thereby raising the number of carbons in the system. Due to deleted carbon dioxide the Calvin cycle is modeled in our network without an explicit IN for carbon dioxide. All OUT only t-invariants contain one of the two transitions modeling Rubisco ('E129_1' and 'E129_2') and thereby an implicit IN of carbon dioxide. This implicit modeling reduces all OUT only invariants to IN + OUT pathways. 'E129_1' models the carboxylation reaction (Calvin cycle) of Rubisco, 'E129_2' models the oxygenation reaction (photorespiration) (Eckardt, 2005). All other modules do not possess OUT only t-invariants. The OUT only t-invariants of the reduced network are comparable to those of the *Sucrose module*.

All cyclic t-invariants are as well artifacts resulting from missing metabolites and cofactors. Figure 4.21 demonstrates an example of a cyclic t-invariant. The modeled starch metabolism requires ATP and α-D-glucose 1-phosphate (metabolite '50') to form ADP-Glucose (metabolite '58') and pyrophosphate in transition 'E46' (Kossmann and Lloyd, 2000; Streb and Zeeman, 2012). ATP and pyrophosphate are not modeled in our network. Otherwise this reaction would require ATP and produce pyrophosphate. This leads to two options:

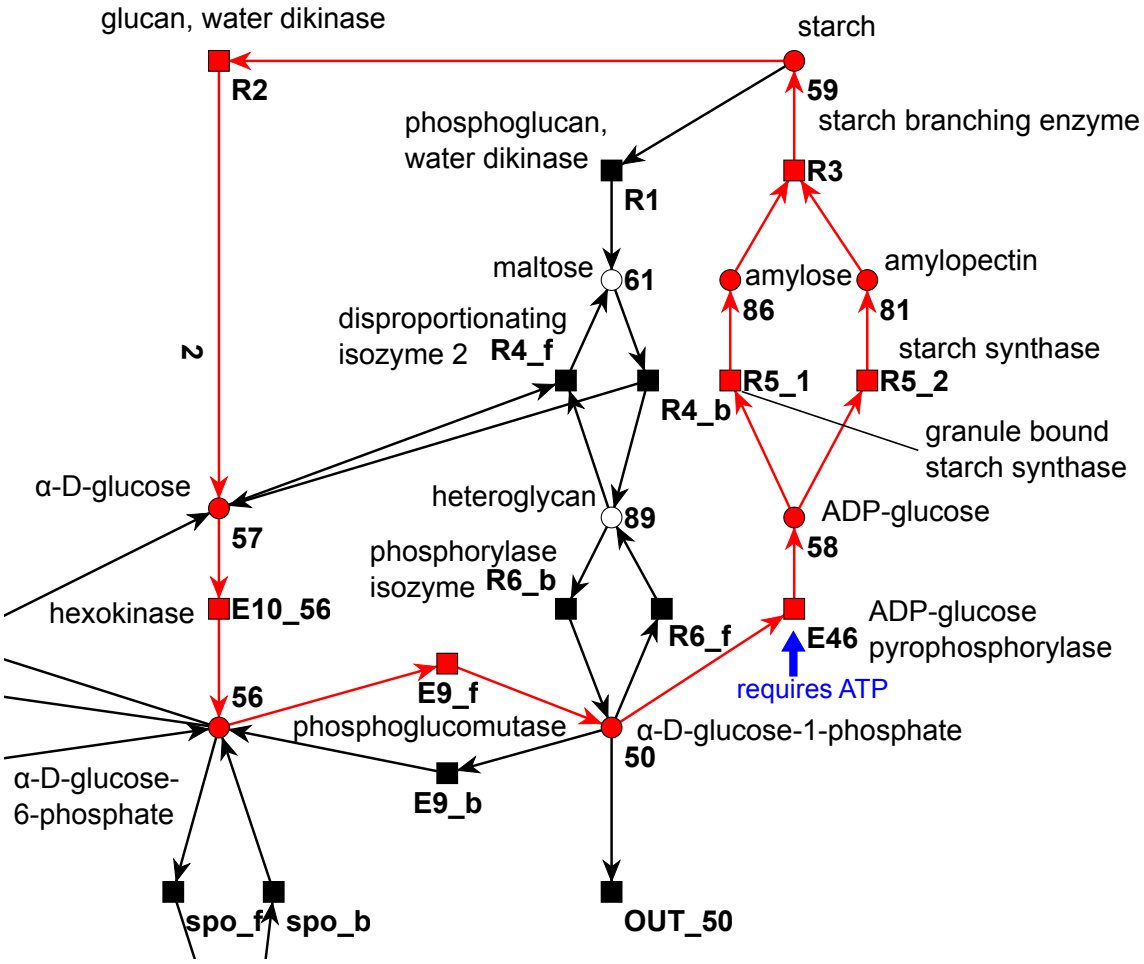


Figure 4.21: Example of a cyclic t-invariant. The cycle evolves out of a silent IN. The underlying reaction of transition 'E46' produces ADP-Glucose and pyrophosphate from ATP and α-D-glucose 1-phosphate (Streb and Zeeman, 2012). ATP is not modeled in our network, otherwise this t-invariant would be an IN only. The additional modeling of pyrophosphate would lead to a standard IN + OUT t-invariant.

- ATP and pyrophosphate are directly provided and removed,
- ATP and pyrophosphate are produced and consumed throughout the network.

Both cases result directly or indirectly in the involvement of input and output transitions leading to an invariant of type IN + OUT.

IN only t-invariants are as well caused by substances which are not modeled. In the *Sucrose module* and Graph A all IN only t-invariants are artifacts of the modeling of the pentose phosphate pathway. transition 'E4' produces a carbon dioxide (Kruger and von Schaewen, 2003) in the pentose phosphate pathway from metabolite '53' (6-phospho gluconate). As carbon dioxide is not modeled in the network, this has the function of a hidden export. This behavior corresponds to the OUT only t-invariants, which are artifacts of the missing modeling of carbon dioxide as well.

4.4.2.2 Exemplary Inspection of a Single T-Invariant

Additionally to the general t-invariant analysis an exemplary inspection of a single t-invariant consisting of twelve reactions of the reduced network is performed. As the reduction process merges reactions, the number of traversed reactions in the original net can be higher (compare Section 4.3).

Table 4.10 illustrates the considered exemplary t-invariant. Additionally syntheses of substrates a reaction is involved in are listed. It has been shown before that several biological pathways can be combined to form a single t-invariant (Koch et al., 2005). The considered t-invariant is producing UTP for the synthesis of RNA from D-fructose (metabolite '63') and ammonia (metabolite '29'). To biologically verify this pathway combination, it has to be demonstrated that each part of the t-invariant can be biologically explained.

The product of the t-invariant is two UTP, which are removed from the network by reaction 'OUT_129_rna'. The used compounds for this product are three D-fructose provided by 'IN_63' and four ammonia provided by 'IN_29'. In plants one UMP is synthesized from four different compounds: one bicarbonate, one glutamine, one 5-phosphoribosyl 1-pyrophosphate, and one L-aspartate (Zrenner et al., 2006). UTP is then synthesized from UMP by adding phosphor-groups. The metabolite bicarbonate is not modeled in the PN. It is demonstrated in the following, that each of the three remaining modeled compounds is synthesized and used in a biological explainable way.

Two L-aspartate are synthesized by 2*'E2_f' (aspartate aminotransferase). The substrates of this reaction are oxalacetate and glutamate and the products are L-aspartate and α-ketoglutarate (Wilkie and Warren, 1998; Graindorge et al., 2010). In the reduction process several pathways leading to oxalacetate are affected, resulting in a merged metabolite which represents various compounds. The affected pathways are: glycerate 3-phosphate synthesis, oxalacetate synthesis, and glycolysis. It seems as if a biologically unreasonable behavior occurs in the t-invariant, as the multi-compound-metabolite, which beside others represents oxalacetate, is synthesized in two steps from D-fructose. In the knowledge of the reduction effects and the following conclusion that the multi-compound-metabolite is an abstraction of the merged pathways, the contradiction of biological unreasonableness vanishes.

Table 4.10: Exemplary t-invariant calculated on the reduced network. Each reaction is listed, including the number of its occurrences in the t-invariant. Additional information is given about the syntheses the reaction is involved in: either the synthesis of one of the substrates L-aspartate, 5-phosphoribosyl 1-pyrophosphate, glutamine, and UMP, or to the final synthesis of RNA. If different subpathways are assigned to the same reaction, the respective multiplicity is mentioned (see 'IN_63' and 'E12')

reaction	involved in synthesis of
2*E2_f	L-aspartate
E14_f	L-aspartate
3*IN_63	L-aspartate, 2*5-phosphoribosyl 1-pyrophosphate
3*E12	L-aspartate, 2*5-phosphoribosyl 1-pyrophosphate
2*ctp(E5+E4)	5-phosphoribosyl 1-pyrophosphate
2*E127	5-phosphoribosyl 1-pyrophosphate
4*IN_29	2*glutamate, 2*glutamine
2*E106_f	glutamate
2*E109	glutamine
2*E38	uridine 5'-phosphate (UMP)
2*ctp(ctp(E42+E43)+ ctp(E44+ctp(E61_119+E61_80)))	uridine 5'-phosphate (UMP)
2*OUT_129_rna	RNA

5-phosphoribosyl 1-pyrophosphate can be synthesized from D-fructose, via D-fructose 6-phosphate and D-ribose 5-phosphate (Dey and Harborne, 1997; Buchanan et al., 2000; Berg et al., 2002; Zrenner et al., 2006). This pathway structure is represented in the t-invariant by the reactions 'E12' (fructokinase, synthesis of D-fructose 6-phosphate), 'ctp(E5+E4)' (6-phosphogluconolactonase and phosphogluconate dehydrogenase, synthesis of D-ribose 5-phosphate), and 'E127' (5-phosphoribosyl 1-pyrophosphatesynthase, synthesis of 5-phosphoribosyl 1-pyrophosphate). Again, a number of reactions in the overall synthesis of 5-phosphoribosyl 1-pyrophosphate is affected by the reduction process, and the precursor of 'E127' combines several compounds.

The remaining reactions 'E106_f' and 'E109' are required to regenerate two glutamate and two glutamine, thereby using four ammonia. Glutamine is an important part of the nitrogen metabolism of conifers (Cánovas et al., 2007). α-ketoglutarate is a known nitrogen transporter in plants (Temple et al., 1998). As glutamate is convertible in both, glutamine and α-ketoglutarate (Aubert et al., 2001; Forde and Lea, 2007), it seems to share these nitrogen-transportation duties. In the t-invariant the regeneration of glutamate and glutamine, respectively from α-ketoglutarate and glutamate and in the process consuming ammonia, resembles this biological interpretation.

These findings prove this t-invariant to be a combination of parts of the nitrogen economy, the oxalacetate synthesis, the synthesis of 5-phosphoribosyl 1-pyrophosphate, and the synthesis of pyrimidines. Altogether that are the important steps to synthesize UTP (Zrenner et al., 2006). Each part of the inspected t-invariant can be biologically interpreted, which proves the possibility

of the network to model these syntheses in a steady-state sustaining manner.

4.4.2.3 T-Cluster

T-cluster analysis proved to be helpful for the interpretation of amounts of t-invariants which are too big for manual review. It was not possible to provide a t-cluster analysis for our network model. An inherent part of the computation of the t-clusters is the determination of the distance matrix (Grafahrend-Belau et al., 2008). Each pair of t-invariants gets its own distance, which turns out to be 27646^2 elements in the matrix considering our reduced network. This exceeds the scope of a practicable computation because of the necessary memory allocation.

4.4.2.4 MCT-Sets

MCT-Sets have been shown to constitute the smallest biological meaningful entities in which a network can be decomposed (Sackmann et al., 2006). We give several examples of MCT-Sets of our network and their biological counterparts. This additionally provides a possibility of comparison between the different decomposition methods and the reduced network.

MCT-S 1, *Sucrose module/* Graph A: The MCT-S of the *Sucrose module* consists of the transitions 'Import_117_fromB/fromCit' ('fromCit' and 'fromB' depends on the decomposition method), 'E43', 'E44', 'E61_119', and 'E61_80'. All of these transitions form the reaction chain leading to uridine 5'-phosphate (Zrenner et al., 2006). The Import transition adds carbamoyl aspartate to the modules, which is produced by transition 'E42' in the corresponding other module, the *Citrate module* in the biological decomposition, and Graph B in the automatic decomposition. *De novo* synthesis of uridine 5'-phosphate is highly energy consuming and in some tissues partly replaced by the recycling of already build compounds. Nevertheless, *de novo* synthesis of UMP is still needed to replenish the nucleotide stock (Moffatt and Ashihara, 2002).

In the reduced network, this complete synthesis pathway merges into one transition, called 'ctp(ctp(E42+E43)+ctp(E44+ctp(E61_119+E61_80)))'. In the recursive reduction process all of the reactions forming the *de novo* synthesis of uridine 5'-phosphate are found to fulfill the conditions needed for a CTP reduction. As transition 'E42' is part of the *Citrate module* it cannot be part of this MCT-S. Nevertheless the possible CTP reduction of the complete reaction chain including 'E42' suggests a strong connection between the reactions of the uridine 5'-phosphate *de novo* synthesis and could indicate an MCT-S in the PN including all of them.

MCT-S 2, *Citrate module/* Graph B: The next MCT-S we discuss is part of the *Citrate module* and Graph B. The set consists of the transitions 'E103', 'E104_f', 'E105', and 'E113_f'. This configuration of the MCT-S occurs in the *Citrate module* of the biologically driven decomposition and Graph B of the automatic decomposition as well as in the reduced network. Compared to the decompositions and the original network, the connections of the transitions have changed during the reduction process. In the decompositions the transitions form the reactions:

- 'E103': $79 \xrightarrow{\text{E103}} 4$

- 'E104_f': $3+20 \xrightarrow{\text{E104\_f}} 21+79$
- 'E105': $2 \xrightarrow{\text{E105}} 3$
- 'E113_f': $9+21 \xrightarrow{\text{E113\_f}} 20+2$

with 2 = glutamate, 3 = γ-aminobutyric acid (GABA), 4 = succinate, 9 = α-ketoglutarate, 20 = pyruvate, 21 = phospho*enol*pyruvate, and 79 = succinate semialdehyde. In the reduced network, the metabolites '4' and '20' are combined by an ITP reduction, forming a new place. The connections of this new place are the combined connections of the merged original places, i.e., the new place connects to 'E103' (as metabolite '4'), 'E104_f' (as metabolite '20'), and 'E113_f' (as metabolite '20'), leading to the reduced reaction system

- 'E103': $79 \xrightarrow{\text{E103}} X$
- 'E104_f': $3+X \xrightarrow{\text{E104\_f}} 21+79$
- 'E105': $2 \xrightarrow{\text{E105}} 3$
- 'E113_f': $9+21 \xrightarrow{\text{E113\_f}} X+2$

with 2 = glutamate, 3 = γ-aminobutyric acid (GABA), 9 = α-ketoglutarate, 21 = phospho*enol*pyruvate, 79 = succinate semialdehyde, and X = the merged place (4 + 20).

While the pathways for these transitions mentioned in the AraCyc database are *glutamate degradation* ('E103', 'E104_f', and 'E105') and *alanine degradation* ('E113_f'), other literature declare them as the *GABA-shunt* (Bouché and Fromm, 2004). Finding an MCT-S of these reactions strongly suggests a close collaboration between them indicating a possible network behavior consistent with the literature, as GABA is sufficient as sole nitrogen source for effective growth of *Arabidopsis thaliana* (Breitkreuz et al., 1999) and the GABA-shunt seems to play an important role in the reaction to oxidative stress (Bouché et al., 2003; Bouché and Fromm, 2004).

MCT-S 3, *Citrate module*/ Graph B: The last MCT-S presented is constituted by the transitions 'E40', 'E133_f', 'E134', 'E135', and 'E136' in the *Citrate module* and in Graph B. The transitions 'E40', 'E133_f', 'E134', and 'E135' form the urea cycle (Tischner et al., 2007), and transition 'E136' models the degradation of urea (metabolite '25', Sirko and Brodzik, 2000). This MCT-S is a collection of the transitions forming and degrading urea.

In the reduced network two CTP reductions take place in this cycle. Initially 'E135' and 'E40' are condensed to 'ctp(E135+E40)' which is further flattened to 'ctp(ctp(E135+E40)+E136)' by the inclusion of 'E136'. Together with 'E134' and 'E133_f' this new reduced transition forms an MCT-S in the reduced network, providing proof of the MCT-S of the decomposed network.

Urea is an important nitrogen source for plants (Polacco and Holland, 1993) and mainly believed to be predominantly synthesized by the urea cycle (Reinbothe and Mothes, 1962). The finding of this MCT-S suggests the importance of the urea metabolism in our model of the core metabolism of *Arabidopsis thaliana*.

4.4.2.5 Shared Invariant Sets

To further analyze the t-invariants of our network we define a new organizational structure called *shared invariant sets* (SIS). Commonly spoken SIS are sets of t-invariants which share a given set of transitions. They are related to the *common motifs* used by the *Agglomeration around Common Motifs* (ACoM) method (Pérès et al., 2006). In that study an algorithm is introduced which automatically finds common motifs of reactions in elementary modes depending on a predefined range of motif sizes. While the ACoM method is more related to a clustering technique, our SIS provides a possibility to perform mathematical set operations on the set of t-invariants, allowing a detailed inspection of subsets and intersections of subsets. Each of the subsets is inspired by the transitions which are part of the t-invariants of this subset.

Definition 4.4.1. *shared invariant sets (SIS):*
Given a Petri net $N = (P,T,F,w,m_0)$*, a subset* $A \subseteq T$*, and a set of t-invariants* $I = \{i_1,...,i_l\}$ *computed from N. The set* $S(A) = \{i_k | \forall t_i \in A : t_i \in i_k\}$ *is called* shared invariant sets (SIS)*. If there exists no non-empty set* $B \subseteq T$*,* $B \cap A = \emptyset$*, with* $\{i_k | \forall t_i \in A : t_i \in i_k\} = \{i_k | \forall t_i \in A \cup B : t_i \in i_k\}$*, the SIS is called* maximal*. Transitions in set A are called* transitions of interest*, transitions of set* $A \cup B$ *are called* extended transitions of interest*.*

An $SIS(A)$ is a set of t-invariants which all have a given set $A \subseteq T$ of transitions in common. A *maximal shared invariant sets* (maxSIS) describes an $SIS(A)$ which does not have another set $B \subseteq T, B \cap A = \emptyset$ of transitions in common. If there exists such a non-empty set B, then $SIS(A)$ would be equal to $SIS(A \cup B)$, and $SIS(A \cup B)$ describes the maxSIS. A maxSIS identifies reactions which always occur together with the transitions of interest in the t-invariants unified in the SIS. These reactions in set B can, e.g., provide raw material for the reactions of A, dispose their products, or both. By this a considerable extended information is given by the determination of set B.

To further demonstrate the usefulness of our new methodology we provide a few examples of their usage. First, we show how the Calvin cycle and the citric acid cycle are integrated into our metabolic model. Second, we extend the exemplary analysis of the single t-invariant (compare Section 4.4.2.2) to a moer general level. Afterwards we contrast an SIS and a corresponding MCT-S, and at last, we compare an SIS found in the reduced network with the corresponding SIS of the decomposed networks.

SIS of the Calvin cycle: We discuss the SIS which is generated by the set ot transitions forming the Calvin cycle. Therefor we show the connection of the Calvin cycle to the rest of the net with the help of the newly defined SIS. Figure 4.22 demonstrates an SIS of the reduced net: (E129_1, E14_b, E1_44_f, E7, E130) + [E11_b, E1_46_f]. Transitions of interest are surrounded by round brackets, the extended transitions of interest, which are required to transform this SIS into a maxSIS, are surrounded by squared brackets. The complete maxSIS contains 277 t-invariants.

The resulting set of transitions {E129_1, E14_b, E1_44_f, E7, E130, E11_b, E1_46_f} forms the complete Calvin cycle in the reduced network. This set of transitions, if provided as transitions of interest for SIS-calculation in the *Sucrose module*, causes a maxSIS made of 215 t-invariants

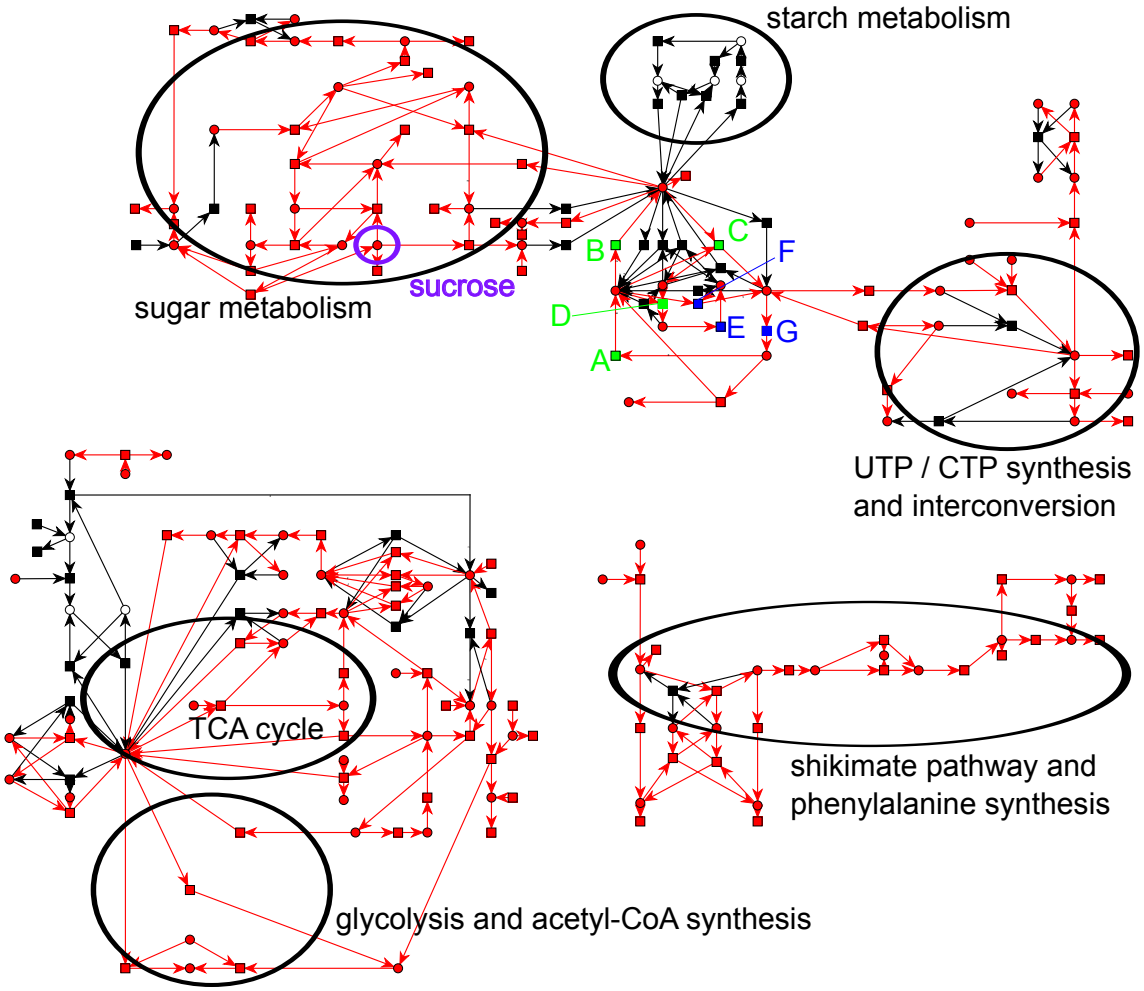


Figure 4.22: SIS of the Calvin cycle in the reduced network. The complete SIS contains 277 t-invariants. Transitions of interest are colored in green, the transitions needed to extend the SIS to a maxSIS are colored in blue, all parts of the net covered by the SIS are colored in red, and parts not covered are black. A: E129_1, B: E14, C: E1_44, D: E11, E: E7, F: E1_46, G: E130. Important areas of the network are indicated by black circles. Sucrose is highlighted in pink.

covering the complete Calvin cycle in the module. This behavior suggests a possible invariant-conserving reduction of the Calvin cycle.

The lower number of invariants in the module can be explained by the decomposition of the network. The Calvin cycle is connected to the complete network in the reduced model, but it is only connected to the *Sucrose module* in the decomposed model. This removes all invariants occurring from a possible connection to the *Citrate module*.

Figure 4.22 shows the SIS of the Calvin cycle in the reduced network. Transitions in green are transitions of interest, transitions in blue are needed to extend the SIS to a maxSIS. All parts (places/metaoblites and transitions/reactions) of the network covered by the SIS are red.

Due to the rules of the reduction process several reactions are combined and to some extend the metabolites participating in the Calvin cycle are merged, e.g. glyceraldehyde 3-phosphate and dihydroxyacetone phosphate are combined into the same place by an ITP reduction.

Nevertheless the maxSIS demonstrates, that the reactions restoring ribulose-1,5-biphosphate, which are in the reduced net 'E7' (sedoheptulose-bisphosphatase), 'E1_46' (transketolase), and 'E130' (phosphoribulokinase), are always combined with the reactions of the carboxylase function of Rubisco ('E129_1') if a steady-state is assumed. This behavior is consistent with the literature (Buchanan et al., 2000) as the restoration of ribulose-1,5-biphosphate is essential for the perpetuation of the Calvin cycle.

The part of the network covered by the invariants which are contained in the SIS includes the sugar metabolism, especially the production of sucrose (metabolite '66', highlighted in pink, see Figure 4.22), and the shikimate pathway. In nature the carbon compounds which are produced in the Calvin cycle and not used to regenerate ribulose-1,5-biphosphate are utilized for the synthesis of sucrose and starch and provide substances to the shikimate pathway as well (Raines, 2003). All three pathways are covered by the SIS indicating that reaction chains can provide substance to these pathways while maintaining a steady-state.

The starch metabolism is not covered by this SIS, because all invariants used for the assembly of the SIS are minimal t-invariants. Minimal t-invariants do not overlap and the starch metabolism itself forms a minimal t-invariant. This means no other minimal t-invariant can cover the starch synthesis and its degradation. That behavior is a consequence of the reduction process, which combines the start metabolite of the starch metabolism ('50', α-D-glucose 1-phosphate) with its end metabolite ('57', α-D-glucose). The starch metabolism is covered by the SIS of the Calvin cycle in the *Sucrose module*, thereby suggesting a behavior consistent with the literature.

The citric acid cycle is covered by the Calvin cycle SIS in the reduced network as well. It has been shown before, that there are tight connections between the citric acid cycle and photosynthesis and carbon fixation in plants (Raghavendra and Padmasree, 2003; Plaxton and Podestá, 2006; Noguchi and Yoshida, 2008). This suggests a proper behavior of the network.

By this we show that the Calvin cycle, as it is modeled and connected in our network, maintains connections to all parts of the network. Demonstrating the comparability of measured and predicted flux data has been used before to validate network models (Poolman et al., 2009; Gomes de Oliveira Dal'Molin et al., 2010). As t-invariants constitute possible steady-state fluxes, the coverage of the network by an SIS of the Calvin cycle suggests that compounds coming from the carbon fixation are able to reach the fundamental pathways.

Application of the SIS methodology to the exemplary single t-invariant: An investigation of the reaction 'OUT_129_rna', which encodes the usage of UTP (metabolite '129') for the RNA synthesis (compare Section 4.4.2.2), is performed. The SIS (OUT_129_rna) extends to the maxSIS (OUT_129_rna) + [ctp(ctp(E42+E43)+ctp(E44+ctp(E61_119+E61_80))), E38, E109, E127]. This means, that all steady-state reaction cascades ending in reaction 'OUT_129_rna' include the synthesis of 5-phosphoribosyl 1-pyrophosphate, the regeneration of glutamine, and the *de novo* synthesis of UMP (compare Table 4.10).

This demonstrates that the synthesis of UTP for the creation of RNA mandatorily requires several basic compounds in the PN model. Especially the regeneration of glutamine from glutamate (Reaction 'E109'), the synthesis of 5-phosphoribosyl 1-pyrophosphate (Reaction 'E127'), the syn-

thesis of L-aspartate (Reaction 'E2_f'), and the *de novo* synthesis of UMP (Reactions 'E38' and 'ctp(ctp(E42+E43)+ ctp(E44+ctp(E61_119+E61_80)))') seem to be of special importance.

The source of UMP in the model is its *de novo* synthesis which consumes 5-phosphoribosyl 1-pyrophosphate, L-aspartate, and glutamine (Zrenner et al., 2006). It is remarkable that no reaction restoring glutamate is part of the maxSIS, but the production of the amino acid L-aspartate depends on ammonia provided by glutamate (reaction 'E2_f', aspartate transaminase, (Wilkie and Warren, 1998; Graindorge et al., 2010)).

Further investigation of this issue reveals three reactions restoring glutamate from α-ketoglutarate (reaction 'E106_f', glutamate dehydrogenase (Aubert et al., 2001); reaction 'E107' and reaction E'108', glutamate synthase (Forde and Lea, 2007)). Each of these is included in exactly one-third of the t-invariants contained in the maxSIS. Considering the three respectively adapted SIS ('OUT_129_rna', 'E106_f'), ('OUT_129_rna', 'E108'), and ('OUT_129_rna', 'E109') it is revealed that all reactions which extend the SIS ('OUT_129_rna') as well extend each of the three adapted SIS. This is an expected behavior as these three reactions are parallel and therefore each gives rise to its own set of t-invariants. Biologically, in the context of the described PN model, the presence of glutamate, and therefore its regeneration, is mandatory for the production of L-aspartate (Wilkie and Warren, 1998; Graindorge et al., 2010).

Overall, these findings confirm the coherence of the model, within its limitations, with the biology.

Comparison of SIS and MCT-S: We discuss an MCT-Set made from the transitions 'E46', 'R5_1', 'R5_2', 'R3' occurring in the *Sucrose module* and in Graph A of the automatic reduction. The MCT-S describes essential transitions of the starch metabolism (compare Figure 4.6). 'E46' is the entrance to this pathway forming ADP-Glucose (metabolite '58') from α-D-glucose 1-phosphate (metabolite '50') followed by 'R5_1', 'R5_2', and 'R3'. 'R5_1' and 'R5_2' form amylose (metabolite '86') and amylopectin (metabolite '81'), respectively. Both compounds are then combined to starch (metabolite '59') by 'R3'.

The transition 'E10_56', which takes part in both degradation pathways of starch, is not part of the MCT-S. This modeled reaction transforms α-D-glucose (metabolite '57') to α-D-glucose 6-phosphate (metabolite '56'). α-D-glucose 6-phosphate itself is in equilibrium with α-D-glucose 1-phosphate (metabolite '50') (Atkinson et al., 1961), which again is the starting agent of the starch (metabolite '59') synthesis (Kossmann and Lloyd, 2000; Streb and Zeeman, 2012). That is an unexpected behavior, as this reaction is obviously essential for the maintenance of a steady-state of the starch metabolism (compare, e.g., the t-invariant shown in Figure 4.21).

Further investigations of the transition 'E10_56' illustrates that the reaction extends the SIS ('E46', 'R5_1', 'R5_2', 'R3') to the maxSIS ('E46', 'ctp(R5_1+ctp(R5_2+R3))') + ['E10_56']. Examinations of the SIS ('E10_56') show connections to the sugar metabolism (see Figure 4.4) which the maxSIS lacks. These connections force the exclusion of transition 'E10_56' from the MCT-S in defiance of its tight connection to the starch metabolism. The procedure of SIS extension finds this connection.

We detect the MCT-S and the SIS in both decompositions suggesting stability of this path-

way, even in systems differently connected to the environment. Additionally, we search for this MCT-S and the SIS in the reduced network. There is an MCT-S consisting of 'E46' and 'ctp(R5_1+ctp(R5_2+R3))'. The latter transition is a double CTP reduction, first of 'R5_2' and 'R3' to 'ctp(R5_2+R3)', and further of the new transition and 'R_5' to 'ctp(R5_1+ctp(R5_2+R3))'. The corresponding SIS ('E46', 'ctp(R5_1+ctp(R5_2+R3))') again extends to the maxSIS ('E46', 'ctp(R5_1+ctp(R5_2+R3))') + ['E10_56'], which suggests a tighter connection of transition 'E10_56' to the starch metabolism as proposed by the examination of MCT-S.

As the starch metabolism in our model is disputable (see Section 4.1.2.1 and 4.1.2.2), the fact that starch biosynthesis and starch degradation forms a maxSIS suggests a reasonable modeling of these two pathways. The findings of the maxSIS covering the degradation pathway ensure a thermodynamically feasible modeling of the starch metabolism. This means, the starch metabolism is able to maintain a steady-state for every possible involvement in the reaction-routes of the network.

Comparison of SIS between decomposed network modules and reduced network. The fact that the automatic decompositions have much less t-invariants than the biological decompositions is hard to explain. In general the number of t-invariants cannot be estimated from the number of transitions in the network.

We find that SIS(E130) contains 3383 t-invariants in the *Sucrose module*, and SIS(E130) + [Export_60_toB] contains 675 t-invariants in Graph A. 'E130' is the only pre-transition to ribulose-1,5-biphosphate ('60'), meaning every t-invariant which includes 'E130' includes one of the transitions encoding Rubisco ('E129_1' and 'E129_2') as well, as they are the only post-transitions of ribulose-1,5-biphosphate. Both transitions are not part of Graph A, i.e. the Calvin cycle is not completely modeled in this module anymore.

12291 t-invariants are part of the SIS(E130) in the reduced network, which means approximately 45% of all steady-state pathways include one of the two reactions modulated by Rubisco. In the *Sucrose module* nearly 75% of the possible steady-state pathways utilize one of these Rubisco-reactions. Compared to Graph A, where about 30% of the steady-state pathways uses Rubisco, these numbers seem quite high. The reaction catalyzed by Rubisco is the only source of fixed carbon, and it seems that the percentages determined for the reduced network and for the biologically driven decompositions resembles this fact a lot better than values calculated for the automatic decomposition.

Together this suggests that an intact Calvin cycle is essential for many pathways throughout the network. Indeed, the stachyose biosynthesis and the D-*myo*-inositol biosynthesis are not contained by any t-invariants that also contain 'E130' in Graph A. On the other hand, in the *Sucrose module* as well as in the reduced network there exist t-invariants which cover 'E130' and the biosyntheses of stachyose and D-*myo*-inositol.

Stachyose is suggested to be a important compound in the stress response in plants (ElSayed et al., 2014). D-*myo*-inositol is an essential substrate required for the synthesis of stachyose, and is argued to play additional roles in cellular signaling (Nelson et al., 1999). This suggest a possibility to synthesize these compounds from the main carbon-source, the fixed carbon of the Calvin cycle, suitable.

The reason for this loss of function in Graph A remains elusive, but suggests the possibility of some serious problems resulting from a not curated automatic decomposition.

Chapter 5

Summary and Conclusions

Our inspection of the literature about the recent academic research in the area of modeling biological networks using quantitative and qualitative approaches has revealed various progress in the understanding of more and more complex systems, supported by the growing amount of measured data and the increasing computational power. Although several quantitative models of the metabolism of *Arabidopsis thaliana* exist, there is a lack of qualitative models in this area of interest.

During our studies, we describe a self-developed qualitative network model of the metabolism of *Arabidopsis thaliana* as a Petri net with the following properties:

- source of model data: Similar to a network of barley (Grafahrend-Belau et al., 2009), and contrary to the existing quantitative models of *Arabidopsis thaliana* which are an automatic read-out of databases (Poolman et al., 2009; Gomes de Oliveira Dal'Molin et al., 2010; Radrich et al., 2010; Mintz-Oron et al., 2012), our network is created by stepwise integrating information from scientific publications.

- connection to the environment: 29 input and output transitions. Of these, 21 are outgoing connections and 8 are incoming. The 4 external metabolites are each both supplied and removed by these connections, which reduce the environmental connections of the internal metabolites to 4 incoming and 17 outgoing junctions.

- compartments: As done in other studies (Poolman et al., 2009; Radrich et al., 2010) we refrain from considering compartments in the model due to network complexity reasons.

- removal of metabolites: To further limit the network complexity as far as possible, all small metabolites like water and carbon dioxide, catalytic substances like ions, and cofactors like NAD/NADH and the energy providing compounds like ATP are omitted from the network. Hereby the network model is reduced by 19 metabolites with 274 edges. Studies prove that hubs are less conserved than compounds which connect different modules (Guimera and Amaral, 2005), suggesting a model without cofactors and small metabolites is more suitable than a model which lacks essential connections between the different biological modules. A

complexity reduction by removing such smaller metabolites is preferred to a restriction of the overall network size.

Several of the small metabolites, namely CoA and UDP, remain in the network due to suggestions of our coworkers (personal communication, Schleiff, 2010). We check our procedure by performing a node-degree analysis on the total AraCyc database. After excluding all metabolites with a node-degree greater than 74, we show that the node-degree distribution of the AraCyc network is very similar to the one of our model.

- starch metabolism: Due to the polymer character of the starch molecule it is difficult to model its synthesis and degradation in detail. There is literature (Kossmann and Lloyd, 2000; Fettke et al., 2009; Lu and Sharkey, 2006; Guy et al., 2008; Reiter, 2008) which describes the reaction cascade of the starch metabolism without special focus on the polymeric structure of starch. We put special efforts into the curation of the thermodynamic feasibility of the starch pathway by an adaption of the stoichiometric parameters in the cascade in such a way that no substance is created or consumed (comparable to the procedure in (Poolman et al., 2009)). We prove this thermodynamic feasibility as we found two cyclic t-invariants, one for each degradation pathway, which together cover the starch metabolism.

- final size of the network: 134 metabolites, 243 reactions, and 572 edges.

As the complexity of this network is too high for the determination of its t-invariants, we decide to decompose the PN model in two partitions. The decomposition is conducted following two different strategies:

- biologically driven decomposition: The partitioning process is based on the literature and academic intuition. We cut 10 edges, most of them possibly representing transport processes from one compartment to another. The resulting two submodules are of similar size:

  → *Sucrose module* (*Sucrose subnet* and *UTP subnet*) with 69 metabolites, 132 reactions, and 294 edges, and

  → *Citrate module* (*Citrate subnet* and *Shikimate subnet*) with 72 metabolites, 125 reactions, and 289 edges.

  We put special efforts into the treatment of important cyclic pathways, especially the Calvin cycle, the glyoxylate cycle, the urea cycle, and the citric acid cycle. In our decomposition we keep these cycles completely uncut. The partitioning affects the glycolysis, which is less problematic, as this pathway is linear. Therefore a cut does not impact the parts upstream of the cut, as long as connections between the modules are introduced in a way that conserves the direction of the cut edge.

- automatic decomposition: The biological partitioning proves the possibility of a bisection of the network into submodules with similar sizes. As any split of the network may negatively influence the t-invariants, we decide to use the KL-algorithm, which is constructed to partition a graph in two subsets of similar size while minimizing the cut edges. The application of the algorithm leads to 9 cut edges and the result is a partitioning in

→ Graph A with 70 metabolites, 132 reactions, and 288 edges, and

→ Graph B with 72 metabolites, 127 reactions, and 304 edges.

Contrary to the biologically driven partitioning, the automatic decomposition destroys the integrity of the Calvin cycle and of the pyrimidine phosphate interconversion cycle.

As a result of the automatic approach, glutamine and glutamate are no longer distributed over the two modules. As both are well-known ammonia-shuttles in plants, this is not likely a major advantage. A cut through important cyclic pathways seems to be a more severe decision than the distribution over several modules of metabolites which are already known to take part in a number of pathways.

Both decomposition procedures lead to submodules with similar sizes and the respective submodules have many metabolites and reactions in common.

The automatic decomposition using the KL-algorithm confirms that the biological decomposition is not affected by possible faulty human intuitions, as the cutting of edges which can not be argued by biological means, especially those connected to metabolites '32' (D-erythrose 4-phosphate) and '117' (carbamoyl aspartate), is performed by both approaches without any difference.

The biologically driven decomposition leads to a slightly higher number of cut edges (biological 10 vs. automatic 9), which is less desirable according to the minimal-criteria in computer science.

The analysis of t-invariants shows a significantly lower number of t-invariants derived from the automatic decomposition; e.g., for the *Sucrose module* and Graph A only about 50 per cent (biological 4602 vs. automatic 2286) and for the *Citrate module* and Graph B only about 70 per cent (biological 3214 vs. automatic 1966). This reveals that the biological driven decomposition enables a significantly higher number of possible reaction chains to operate at steady-state. We can assume that the more diverse set of pathways found in the biological decomposition is able to cope with a wider range of environmental conditions.

This suggestion is reinforced by the fact that the automatic decomposition has a considerably lower percentage of all steady-state pathways that utilize a reaction catalyzed by Rubisco than both, the biologically driven decomposition as well as the reduced network. So in Graph A of the automated decomposition approximately 30% utilize a Rubisco-catalyzed reaction, compared to approximately 45% in the reduced network and around 75% in the *Sucrose module* of the biologically driven decomposition. In addition some important signaling compounds cannot be synthesized from substrates coming from the Calvin cycle in the automatic decomposition, if a steady-state is assumed.

Thereby a comparison of these two decomposition procedures suggests a certain superiority of the biologically driven decomposition compared to the automatic approach.

All submodules emerging from both decomposition methods are CTI. But, as the decomposition method might not be CTI conserving, the result on the submodules is insufficient to conclude the CTI property for the original network. This affects the interpretability of the determined t-invariants as well, necessitating a CTI conserving method of complexity reduction.

As the original PN is too complex to calculate the set of t-invariants and the desired CTI property can not be definitely derived through network decomposition, we decide to implement a suitable mathematical approach to reduce the network complexity. A suitable reduction procedure needs to conserve the CTI property which is essential (Heiner and Koch, 2004) for biological networks.

The developed network reduction algorithm is proved to fulfill the requirement of CTI conservation. We show that a reduced network fulfilling the CTI property implicates a CTI original network.

Additionally, we prove that there is a further relationship between the reduced and original network in such a way, that we are able to conclude from the t-invariants of the reduced network to the t-invariants of the original network. Each invariant x' including a set of reduced metabolites and reactions from the reduced network has at least one corresponding invariant x in the original network. Such a corresponding invariant x traverses a part of the original metabolites and reactions which are merged by the algorithm into the set of metabolites and reactions included by x'. This confirms that the biological interpretation of the reduced set of t-invariants is meaningful.

Our algorithm uses two structures common to biological networks:

- the common transition pair (CTP)
  $\rightarrow$ biological interpretation: unbranched reaction chain
- the invariant transition pair (ITP)
  $\rightarrow$ biological interpretation: reversible reaction.

Additionally parallel structures of connections between two metabolites are merged as described in other Petri net analyses (Murata, 1989; Reddy et al., 1993).

The algorithm reduces the original network recursively using the aforementioned structures. After 86 CTP, 62 ITP, and 2 parallel reductions the algorithm stops as no further reduction capability exists. The resulting reduced network consists of 60 metabolites (45% of original 134), 131 reactions (54% of original 243), and 329 edges (58% of original 572) with a total of 27646 t-invariants which completely cover the reduced network.

Usually the structure of a reduced network is not completely biologically interpretable. For our uncurated reduced network we are able to find at least for certain substructures a possible biological explanation, e.g., the reduction procedure merges several metabolites and reactions of the Calvin cycle, the glycolysis and the citric acid cycle into a single metabolite. It has been shown that enzymes are shared between the glycolysis and Calvin cycle (Peltier et al., 2006). This requires a shared pool of the metabolites which take part in the reactions catalyzed by these enzymes.

We also introduce an extended version of the CTP reduction procedure to a structure with arbitrary edge-weights. This enables the reduction of structures of any number of identical molecules of type A into an arbitrary number of identical molecules of type B. It is proved that such a reduction also conserves the CTI property, and the aforementioned relationship between invariants of the original and invariants of the reduced network, which is important for biological interpretations, is concluded as well.

As outlined before the developed qualitative metabolic network is CTI. During our studies we check further behavioral and invariant properties of our biological network. Therefor we use the derived decomposed networks, the reduced network as well as additional newly developed algorithms for network verification:

- Our network is live.

  To prove this important behavioral property, usually the computation of the reachability graph or its derivation, the coverability graph, is mandatory. This computation is an EXPSPACE-hard problem (Esparza, 1998).

  We develop a method to limit the token-state per place of the reachability graph to a boolean structure, additionally ignoring all intermediate states. The running time of the algorithm is $O(|P| + |P| \cdot |T|^2)$.

  Applying our algorithm we determine the network liveness by calculating a boolean reachability graph depending on the input transitions of the network. Our algorithm determines the reactions which are structurally live, meaning live independent from the initial marking.

  The developed method is especially suitable for liveness tests of metabolic networks, as this kind of networks always contains input reactions.

- Our network is pure.

  This structural property is essential for a feasible interpretation of the t-invariants (Sackmann et al., 2006).

- The interpretation of t-invariants exhibits a suitable behavior of the network.

  Within the logic of the model the feasibility of the thermodynamically unreasonable t-invariants (IN only, OUT only, and cyclic) is demonstrated.

- Fixed carbon dioxide reaches all important parts of the network.

  Our newly developed structure of the shared invariant sets (SIS) determines all steady-state pathways that traverse a given set of reactions of interest. Applied to the Calvin cycle, the emerging SIS can be interpreted as possible steady-state connections of the reactions of the Calvin cycle to the rest of the network. The SIS of the Calvin cycle shows connections to all parts of the network, especially to the citric acid cycle. Tight connections between these two cycles have been described before (Plaxton and Podestá, 2006; Raghavendra and Padmasree, 2003; Noguchi and Yoshida, 2008).

- Calvin cycle shows a suitable behavior.

  We introduce an extension to SIS, the maxSIS. A maxSIS incorporates all reactions next to the reactions of interest, which are part of each pathway in the SIS as well. The SIS of several reactions of the regeneration and carboxylation phase of the Calvin cycle extends to the maxSIS which includes the complete cycle, suggesting a close correlation of these two phases.

- RNA synthesis is in line with the biological reality.

  The exemplary analysis of the RNA synthesis using UTP shows a behavior which is in line with the biological reality. An extension of this analysis by the SIS methodology reveals the expected behavior: A mandatory requirement of ammonia, 5-phosphoribosyl 1-pyrophosphate, L-aspartate, glutamine regeneration, and in parts glutamate regeneration in all possible steady-state pathways that contribute to the UTP production for RNA synthesis.

- Starch metabolism is feasible.

  The analysis of t-invariants gives rise to an MCT-Set (Sackmann et al., 2006) including the starch synthesis reactions, but not the starch degradation.

  An inspection of the SIS using the MCT-S as transitions of interest shows that both degradation pathways are part of the SIS. Additionally a maxSIS incorporating transition 'E10_56' is revealed. This reaction regenerates a precursor to α-D-glucose 1-phosphate, from which the starch synthesis starts. 'E10_56' takes part in both starch degradation pathways, but is not part of the MCT-S. The occurrence of this reaction in the t-invariants covering the starch synthesis is required for the thermodynamic feasibility of this pathway.

- Several important pathways show a strong consistency.

  $\rightarrow$ The pathway of the *de novo* synthesis of UMP is an MCT-S in Graph A (automatic decomposition) and the *Sucrose module* (biologically driven decomposition). Additionally, it is completely reduced by several CTP reduction steps in the reduced network into a single reaction. UMP synthesis is split by the decompositions, leaving a part of this synthesis uncovered by the MCT-S in the submodules. But, as CTP reduction is based on the method of MCT-S, the finding in the reduced model suggests an MCT-S covering the complete pathway.

  $\rightarrow$ The GABA shunt forms an MCT-S. GABA is sufficient as sole nitrogen source (Breitkreuz et al., 1999) and seems to play an important role in the response to oxidative stress (Bouché et al., 2003; Bouché and Fromm, 2004).

  $\rightarrow$ Urea cycle and urea degradation form an MCT-S in the reduced and decomposed networks. Urea is shown to be an important nitrogen source for plants (Polacco and Holland, 1993) and is believed to be predominantly synthesized in the urea cycle (Reinbothe and Mothes, 1962).

  As MCT-S constitute the smallest biologically meaningful functional units (Sackmann et al., 2006), the finding that important pathways constitute MCT-S of the network strongly suggests a feasible modeling.

In this work a qualitative biological network model of *Arabidopsis thaliana* is created, using Petri nets as the mathematical representation form. The modeled metabolic system is completely composed of biological structures described in the literature. To analyze the behavior of this

initially overly complex network new methods for network decomposition and reduction are developed and successfully applied. The network reduction procedure conserves the CTI property. A liveness test customized for metabolic networks is developed and our model is positively checked. We calculate the t-invariants of all the decomposed and reduced networks and analyze them applying the established MCT-S approach. This analysis is extended by the new SIS method, which enables the biological interpretation of precisely defined groups of steady-state pathways.

The mathematical modeling of *Arabidopsis thaliana* and the developed methods establish new dimensions for a prosperous interpretation of qualitative biological network models. Further studies may focus on the application of the SIS technique for the determination of reaction and pathway importance, and on its combination with approved methods like MCT-S. Another field of interest is the application of the developed reduction procedures, especially the extended CTP reduction method, to other biological network models.

Bibliography

Ackermann, J., Einloft, J., Nöthen, J., and Koch, I. (2012). Reduction techniques for network validation in systems biology. *Journal of Theoretical Biology*, 315(0):71 – 80.

Acuña, V., Chierichetti, F., Lacroix, V., Marchetti-Spaccamela, A., Sagot, M.-F., and Stougie, L. (2009). Modes and cuts in metabolic networks: Complexity and algorithms. *Biosystems*, 95(1):51 – 60.

Acuña, V., Marchetti-Spaccamela, A., Sagot, M.-F., and Stougie, L. (2010). A note on the complexity of finding and enumerating elementary modes. *Biosystems*, 99(3):210 – 214.

Ada (2012). ISO/IEC 8652:2012 Information technology – Programming languages – Ada.

Akutsu, T., Miyano, S., and Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734.

Andre, C., Froehlich, J. E., Moll, M. R., and Benning, C. (2007). A heteromeric plastidic pyruvate kinase complex involved in seed oil biosynthesis in arabidopsis. *The Plant Cell Online*, 19(6):2006–2022.

AraCyc (2012). www.arabidopsis.org/biocyc/.

Arnborg, S., Courcelle, B., Proskurowski, A., and Seese, D. (1993). An algebraic theory of graph reduction. *J. ACM*, 40(5):1134–1164.

Atkinson, M., Johnson, E., and Morton, R. (1961). Equilibrium constants of phosphoryl transfer from c (1) to c (6) of α-d-glucose 1-phosphate and from glucose 6-phosphate to water. *Biochemical Journal*, 79(1):12.

Aubert, S., Bligny, R., Douce, R., Gout, E., Ratcliffe, R., and Roberts, J. (2001). Contribution of glutamate dehydrogenase to mitochondrial glutamate metabolism studied by (13)c and (31)p nuclear magnetic resonance. *Journal of Experimental Botany*, 52(354):37–45.

Baldan, P., Cocco, N., Marin, A., and Simeoni, M. (2010). Petri nets for modelling metabolic pathways: a survey. *Natural Computing*, 9:955–989.

Barkaoui, K. and Pradat-Peyre, J.-F. (1996). On liveness and controlled siphons in petri nets. In Billington, J. and Reisig, W., editors, *Application and Theory of Petri Nets 1996*, volume 1091 of *Lecture Notes in Computer Science*, pages 57–72. Springer Berlin Heidelberg.

Bassham, J. A., Benson, A. A., Kay, L. D., Harris, A. Z., Wilson, A. T., and Calvin, M. (1954). The path of carbon in photosynthesis. xxi. the cyclic regeneration of carbon dioxide acceptor1. *Journal of the American Chemical Society*, 76(7):1760–1770.

Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., and Schneider, D. (2005). Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in escherichia coli. *Bioinformatics*, 21(suppl 1):i19–i28.

Bérenguier, D., Chaouiya, C., Monteiro, P. T., Naldi, A., Remy, E., Thieffry, D., and Tichit, L. (2013). Dynamical modeling and analysis of large cellular regulatory networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(2):025114.

Berg, J. M., Tymoczko, J. L., and Stryer, L. (2002). *Biochemistry*. W. H. Freeman, New York, 5th edition.

Bermejo, C., Ewald, J. C., Lanquar, V., Jones, A. M., and Frommer, W. B. (2011). In vivo biochemistry: quantifying ion and metabolite levels in individual cells or cultures of yeast. *Biochemical Journal*, 438(1):1–10.

Berthelot, G. (1987). Transformations and decompositions of nets. In *Petri Nets: Central models and their properties*, pages 359–376. Springer.

Bhalla, U. S. and Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387.

Blank, L. M. and Ebert, B. E. (2013). From measurement to implementation of metabolic fluxes. *Current Opinion in Biotechnology*, 24(1):13 – 21.

Bortfeldt, R. H., Schuster, S., and Koch, I. (2010). Exhaustive analysis of the modular structure of the spliceosomal assembly network: A petri net approach. *In Silico Biology*, 10(1):89–123.

Bouché, N., Fait, A., Bouchez, D., Møller, S. G., and Fromm, H. (2003). Mitochondrial succinic-semialdehyde dehydrogenase of the γ-aminobutyrate shunt is required to restrict levels of reactive oxygen intermediates in plants. *Proceedings of the National Academy of Sciences*, 100(11):6843–6848.

Bouché, N. and Fromm, H. (2004). Gaba in plants: just a metabolite? *Trends in Plant Science*, 9(3):110 – 115.

Breitkreuz, K. E., Shelp, B. J., Fischer, W. N., Schwacke, R., and Rentsch, D. (1999). Identification and characterization of gaba, proline and quaternary ammonium compound transporters from arabidopsis thaliana. *FEBS Letters*, 450(3):280 – 284.

Bruex, A., Kainkaryam, R. M., Wieckowski, Y., Kang, Y. H., Bernhardt, C., Xia, Y., Zheng, X., Wang, J. Y., Lee, M. M., Benfey, P., Woolf, P. J., and Schiefelbein, J. (2012). A gene regulatory network for root epidermis cell differentiation in arabidopsis. *PLoS Genet*, 8(1):e1002446.

Bruneau, L., Chapman, R., and Marsolais, F. (2006). Co-occurrence of both l-asparaginase subtypes in arabidopsis: At3g16150 encodes a K^+-dependent l-asparaginase. *Planta*, 224:668–679.

Buchanan, B. B., Gruissem, W., and Jones, R. L. (2000). *Biochemistry & molecular biology of plants*. American Soc. of Plant Physiologists, Rockville, Md.

Calvin, M. (1956). The photosynthetic carbon cycle. *Journal of the Chemical Society (Resumed)*, pages 1895–1915.

Cánovas, F. M., Avila, C., Cantón, F. R., Cañas, R. A., and de la Torre, F. (2007). Ammonium assimilation and amino acid metabolism in conifers. *Journal of Experimental Botany*, 58(9):2307–2318.

Caspi, R., Altman, T., Dreher, K., Fulcher, C. A., Subhraveti, P., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Ong, Q., Paley, S., Pujar, A., Shearer, A. G., Travers, M., Weerasinghe, D., Zhang, P., and Karp, P. D. (2012). The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Research*, 40(D1):D742–D753.

Chabrier-Rivier, N., Chiaverini, M., Danos, V., Fages, F., and Schächter, V. (2004). Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25 – 44.

Chae, L., Lee, I., Shin, J., and Rhee, S. Y. (2012). Towards understanding how molecular networks evolve in plants. *Current Opinion in Plant Biology*, 15(2):177 – 184.

Chalot, M., Blaudez, D., and Brun, A. (2006). Ammonia: a candidate for nitrogen transfer at the mycorrhizal interface. *Trends in Plant Science*, 11(6):263 – 266.

Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210–219.

Chaouiya, C., Remy, E., and Thieffry, D. (2008). Petri net modelling of biological regulatory networks. *Journal of Discrete Algorithms*, 6(2):165 – 177.

Charbit, P., de Montgolfier, F., and Raffinot, M. (2012). Linear time split decomposition revisited. *SIAM Journal on Discrete Mathematics*, 26(2):499–514.

Chen, M. and Hofestädt, R. (2003). Quantitative petri net model of gene regulated metabolic networks in the cell. *In Silico Biology*, 3:347 – 365.

Chen, X. and Shachar-Hill, Y. (2012). Insights into metabolic efficiency from flux analysis. *Journal of Experimental Botany*, 63(6):2343–2351.

Cho, M.-H., Corea, O. R. A., Yang, H., Bedgar, D. L., Laskar, D. D., Anterola, A. M., Moog-Anterola, F. A., Hood, R. L., Kohalmi, S. E., Bernards, M. A., Kang, C., Davin, L. B., and Lewis, N. G. (2007). Phenylalanine biosynthesis in arabidopsis thaliana. identification and characterization of arogenate dehydratases. *The Journal of Biological Chemistry*, 282:30827–30835.

Cohen, E. and Tarsi, M. (1991). Np-completeness of graph decomposition problems. *Journal of Complexity*, 7(2):200 – 212.

Collakova, E. and DellaPenna, D. (2001). Isolation and functional analysis of homogentisate phytyltransferase from synechocystis sp. pcc 6803 and arabidopsis. *Plant Physiology*, 127(3):1113–1124.

Cornah, J. E., Germain, V., Ward, J. L., Beale, M. H., and Smith, S. M. (2004). Lipid utilization, gluconeogenesis, and seedling growth in arabidopsis mutants lacking the glyoxylate cycle enzyme malate synthase. *The Journal of Biological Chemistry*, 279:42916–42923.

Cousins, A., Pracharoenwattana, I., Zhou, W., Smith, S., and Badger, M. (2008). Peroxisomal malate dehydrogenase is not essential for photorespiration in arabidopsis but its absence causes an increase in the stoichiometry of photorespiratory co2 release. *Plant Physiology*, 148(2):786–95.

Craciun, G., Pantea, C., and Rempala, G. A. (2009). Algebraic methods for inferring biochemical networks: A maximum likelihood approach. *Computational Biology and Chemistry*, 33(5):361 – 367.

Crampin, E., Schnell, S., and McSharry, P. (2004). Mathematical and computational techniques to deduce complex biochemical reaction mechanisms. *Progress in Biophysics and Molecular Biology*, 86(1):77 – 112.

Davey, M., Gilot, C., Persiau, G., Ostergaard, J., Han, Y., Bauw, G., and Van Montagu, M. (1999). Peroxisomal malate dehydrogenase is not essential for photorespiration in arabidopsis but its absence causes an increase in the stoichiometry of photorespiratory co2 release. *Plant Physiology*, 121(2):535–43.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103.

Dean, J. V. and Delaney, S. P. (2008). Metabolism of salicylic acid in wild-type, ugt74f1 and ugt74f2 glucosyltransferase mutants of arabidopsis thaliana. *Physiologia Plantarum*, 132(4):417–425.

Desel, J. and Esparza, J. (1995). *Free choice Petri nets*, volume 40 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, Cambridge, UK.

Dey, P. and Harborne, J., editors (1997). *Plant Biochemistry*. Academic Press, San Diego, USA.

Eckardt, N. A. (2005). Photorespiration revisited. *The Plant Cell Online*, 17(8):2139–2141.

Edwards, J. S., Covert, M., and Palsson, B. (2002). Metabolic modelling of microbes: the flux-balance approach. *Environmental Microbiology*, 4(3):133–140.

Einloft, J., Ackermann, J., Nöthen, J., and Koch, I. (2013). Monalisa - visualization and analysis of functional modules in biochemical networks. *Bioinformatics*, 29(11):1469–1470.

Elias, P., Feinstein, A., and Shannon, C. (1956). A note on the maximum flow through a network. *Information Theory, IRE Transactions on*, 2(4):117–119.

ElSayed, A. I., Rafudeen, M. S., and Golldack, D. (2014). Physiological aspects of raffinose family oligosaccharides in plants: protection against abiotic stress. *Plant Biology*, 16(1):1–8.

Esparza, J. (1998). Decidability and complexity of petri net problems—an introduction. In *Lectures on Petri Nets I: Basic Models*, pages 374–428. Springer.

Esparza, J. and Nielsen, M. (1994). Decidability issues for petri nets. *Petri nets newsletter*, 94:5–23.

Esparza, J. and Silva, M. (1992). A polynomial-time algorithm to decide liveness of bounded free choice nets. *Theoretical Computer Science*, 102(1):185 – 205.

Feist, A. M., Herrgård, M. J., Thiele, I., Reed, J. L., and Palsson, B. O. (2009). Reconstruction of biochemical networks in microorganisms. *Nature Reviews Microbiology*, 7:129–143.

Fettke, J., Hejazi, M., Smirnova, J., Höchel, E., Stage, M., and Steup, M. (2009). Eukaryotic starch degradation: integration of plastidial and cytosolic pathways. *Journal of Experimental Botany*, 60(10):2907–2922.

Flugge, U.-l. and Heldt, H. W. (1991). Metabolite translocators of the chloroplast envelope. *Annual review of plant biology*, 42(1):129–144.

Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404.

Forde, B. G. and Lea, P. J. (2007). Glutamate in plants: metabolism, regulation, and signalling. *Journal of Experimental Botany*, 58(9):2339–2358.

Fraser, C. M. and Chapple, C. (2011). The phenylpropanoid pathway in arabidopsis. In *The Arabidopsis Book*, volume 9. American Society of Plant Biologists. Epub ahead of print: Dec 6, 2011.

Fukushima, A., Kusano, M., Redestig, H., Arita, M., and Saito, K. (2009). Integrated omics approaches in plant systems biology. *Current Opinion in Chemical Biology*, 13(5-6):532–538.

Fulda, M., Shockey, J., Werber, M., Wolter, F. P., and Heinz, E. (2002). Two long-chain acyl-coa synthetases from arabidopsis thaliana involved in peroxisomal fatty acid β-oxidation. *The Plant Journal*, 32(1):93–103.

Gagneur, J. and Klamt, S. (2004). Computation of elementary modes: a unifying framework and the new binary approach. *BMC Bioinformatics*, 5(1):175.

Garcia, I., Rodgers, M., Pepin, R., Hsieh, T.-F., and Matringe, M. (1999). Characterization and subcellular compartmentation of recombinant 4-hydroxyphenylpyruvate dioxygenase from arabidopsis in transgenic tobacco. *Plant Physiology*, 119(4):1507–1516.

Genrich, H., Küffner, R., and Voss, K. (2001). Executable petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technology Transfer*, 3(4):394–404.

Gilbert, D. and Heiner, M. (2006). From petri nets to differential equations – an integrative approach for biochemical network analysis. In Donatelli, S. and Thiagarajan, P., editors, *Petri Nets and Other Models of Concurrency - ICATPN 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 181–200. Springer Berlin / Heidelberg.

Ginsburg, H. (2009). Caveat emptor: limitations of the automated reconstruction of metabolic pathways in plasmodium. *Trends in Parasitology*, 25(1):37 – 43.

Gomes de Oliveira Dal'Molin, C., Quek, L.-E., Palfreyman, R. W., Brumbley, S. M., and Nielsen, L. K. (February 2010). Aragem, a genome-scale reconstruction of the primary metabolic network in arabidopsis. *Plant Physiology*, 152(2):579–589.

Goss, P. and Peccoud, J. (1999). Analysis of the stabilizing effect of rom on the genetic network controlling plasmid replication. In *Pacific symposium on biocomputing*, volume 4, pages 65–76.

Goyer, A., Collakova, E., de la Garza, R. D., Quinlivan, E. P., Williamson, J., Gregory, III, J. F., Shachar-Hill, Y., and Hanson, A. D. (2005). 5-formyltetrahydrofolate is an inhibitory but well tolerated metabolite in arabidopsis leaves. *The Journal of Biological Chemistry*, 280:26137–26142.

Grafahrend-Belau, E., Schreiber, F., Heiner, M., Sackmann, A., Junker, B., Grunwald, S., Speer, A., Winder, K., and Koch, I. (2008). Modularization of biochemical networks based on classification of petri net t-invariants. *BMC Bioinformatics*, 9(1):90.

Grafahrend-Belau, E., Schreiber, F., Koschützki, D., and Junker, B. H. (2009). Flux balance analysis of barley seeds: A computational approach to study systemic properties of central metabolism. *Plant Physiology*, 149(1):585–598.

Graindorge, M., Giustini, C., Jacomin, A. C., Kraut, A., Curien, G., and Matringe, M. (2010). Identification of a plant gene encoding glutamate/aspartate-prephenate aminotransferase: The last homeless enzyme of aromatic amino acids biosynthesis. *FEBS Letters*, 584(20):4357 – 4360.

Gross, K. C. and Pharr, D. M. (1982). A potential pathway for galactose metabolism in cucumis sativus l., a stachyose transporting species. *Plant Physiology*, 69(1):117–121.

Grunwald, S., Speer, A., Ackermann, J., and Koch, I. (2008). Petri net modelling of gene regulation of the duchenne muscular dystrophy. *Biosystems*, 92(2):189 – 205.

Guimera, R. and Amaral, L. A. N. (2005). Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900.

Guy, C., Kaplan, F., Kopka, J., Selbig, J., and Hincha, D. K. (2008). Metabolomics of temperature stress. *Physiologia Plantarum*, 132(2):220–235.

Harkewicz, R. and Dennis, E. A. (2011). Applications of mass spectrometry to lipids and membranes. *Annual Review of Biochemistry*, 80:301–325.

Heazlewood, J. L., Verboom, R. E., Tonti-Filippini, J., Small, I., and Millar, A. H. (2007). Suba: the arabidopsis subcellular database. *Nucleic Acids Research*, 35(suppl 1):D213–D218.

Heiner, M. and Koch, I. (2004). Petri net based model validation in systems biology. In Cortadella, J. and Reisig, W., editors, *Applications and Theory of Petri Nets 2004*, volume 3099 of *Lecture Notes in Computer Science*, pages 216–237. Springer Berlin Heidelberg.

Heiner, M., Koch, I., and Will, J. (2004). Model validation of biological pathways using petri nets—demonstrated for apoptosis. *Biosystems*, 75(1–3):15 – 28.

Herrmann, K. M. (1995). The shikimate pathway: Early steps in the biosynthesis of aromatic compounds. *The Plant Cell Online*, 7(7):907–919.

Herrmann, K. M. and Weaver, L. M. (1999). The shikimate pathway. *Annual review of plant biology*, 50(1):473–503.

Hickman, R., Hill, C., Penfold, C. A., Breeze, E., Bowden, L., Moore, J. D., Zhang, P., Jackson, A., Cooke, E., Bewicke-Copley, F., Mead, A., Beynon, J., Wild, D. L., Denby, K. J., Ott, S., and Buchanan-Wollaston, V. (2013). A local regulatory network around three nac transcription factors in stress responses and senescence in arabidopsis leaves. *The Plant Journal*, 75(1):26–39.

Holländer-Czytko, H., Grabowski, J., Sandorf, I., Weckermann, K., and Weiler, E. W. (2005). Tocopherol content and activities of tyrosine aminotransferase and cystine lyase in arabidopsis under stress conditions. *Journal of Plant Physiology*, 162(7):767 – 770.

Holme, P., Huss, M., and Jeong, H. (2003). Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538.

Humphreys, J. M. and Chapple, C. (2002). Rewriting the lignin roadmap. *Current Opinion in Plant Biology*, 5(3):224 – 229.

Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: Systems biology. *Annual Review of Genomics and Human Genetics*, 2(1):343–372. PMID: 11701654.

Igarashi, D., Miwa, T., Seki, M., Kobayashi, M., Kato, T., Tabata, S., Shinozaki, K., and Ohsumi, C. (2003). Identification of photorespiratory glutamate: glyoxylate aminotransferase (ggat) gene in arabidopsis. *The Plant Journal*, 33(6):975–987.

ISHIKAWA, T. and SHIGEOKA, S. (2008). Recent advances in ascorbate biosynthesis and the physiological significance of ascorbate peroxidase in photosynthesizing organisms. *Bioscience, Biotechnology, and Biochemistry*, 72(5):1143–1154.

Ismond, K. P., Dolferus, R., De Pauw, M., Dennis, E. S., and Good, A. G. (2003). Enhanced low oxygen survival in arabidopsis through increased metabolic flux in the fermentative pathway. *Plant Physiology*, 132(3):1292–1302.

Jander, G., Norris, S. R., Joshi, V., Fraga, M., Rugg, A., Yu, S., Li, L., and Last, R. L. (2004). Application of a high-throughput hplc-ms/ms assay to arabidopsis mutant screening; evidence that threonine aldolase plays a role in seed nutritional quality. *The Plant Journal*, 39(3):465–475.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabasi, A.-L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654.

Johnson, D. S. (1982). The np-completeness column: An ongoing guide. *Journal of Algorithms*, 3(2):182 – 195.

Joshi, V., Laubengayer, K. M., Schauer, N., Fernie, A. R., and Jander, G. (December 2006). Two arabidopsis threonine aldolases are nonredundant and compete with threonine deaminase for a common substrate pool. *The Plant Cell Online*, 18(12):3564–3575.

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., and Yamanishi, Y. (2008). Kegg for linking genomes to life and the environment. *Nucleic Acids Research*, 36(suppl 1):D480–D484.

Kanehisa, M. and Goto, S. (2000). Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30.

Karpinski, M. (2002). Approximability of the minimum bisection problem: An algorithmic challenge. In Diks, K. and Rytter, W., editors, *Mathematical Foundations of Computer Science 2002*, volume 2420 of *Lecture Notes in Computer Science*, pages 59–67. Springer Berlin Heidelberg.

Karr, J. R., Sanghvi, J. C., Macklin, D. N., Gutschow, M. V., Jacobs, J. M., Jr., B. B., Assad-Garcia, N., Glass, J. I., and Covert, M. W. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389 – 401.

Kemper, P. and Bause, F. (1992). An efficient polynomial-time algorithm to decide liveness and boundedness of free-choice nets. In Jensen, K., editor, *Application and Theory of Petri Nets 1992*, volume 616 of *Lecture Notes in Computer Science*, pages 263–278. Springer Berlin Heidelberg.

Kernighan, B. W. and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307.

Kitano, H. (2002a). Computational systems biology. *Nature*, 420(6912):206–210.

Kitano, H. (2002b). Systems biology: a brief overview. *Science*, 295(5560):1662–1664.

Klamt, S. and Stelling, J. (2002). Combinatorial complexity of pathway analysis in metabolic networks. *Molecular Biology Reports*, 29(1-2):233–236.

Klein, C., Marino, A., Sagot, M.-F., Vieira Milreu, P., and Brilli, M. (2012). Structural and dynamical analysis of biological networks. *Briefings in Functional Genomics*.

Koch, I. (2010). Petri nets – a mathematical formalism to analyze chemical reaction networks. *Molecular Informatics*, 29(12):838–843.

Koch, I. and Ackermann, J. (2013). On functional module detection in metabolic networks. *Metabolites*, 3(3):673–700.

Koch, I. and Heiner, M. (2007). *Petri Nets*, pages 139–179. John Wiley & Sons, Inc.

Koch, I., Junker, B. H., and Heiner, M. (2005). Application of petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(7):1219–1226.

Koch, I., Reisig, W., and Schreiber, F., editors (2011). *Modeling in Systems Biology: The Petri Net Approach*, volume 16 of *Computational Biology*. Springer London, 1 edition.

Kossmann, J. and Lloyd, J. (2000). Understanding and influencing starch biochemistry. *Critical Reviews in Plant Sciences*, 19(3):171–226.

Kruger, N. J., Huddleston, J. E., Lay, P. L., Brown, N. D., and Ratcliffe, R. G. (2007). Network flux analysis: Impact of 13c-substrates on metabolism in arabidopsis thaliana cell suspension cultures. *Phytochemistry*, 68(16–18):2176 – 2188.

Kruger, N. J., Masakapalli, S. K., and Ratcliffe, R. G. (2012). Strategies for investigating the plant metabolic network with steady-state metabolic flux analysis: lessons from an arabidopsis cell culture and other systems. *Journal of Experimental Botany*, 63(6):2309–2323.

Kruger, N. J. and Ratcliffe, R. G. (2009). Insights into plant metabolic networks from steady-state metabolic flux analysis. *Biochimie*, 91(6):697 – 702.

Kruger, N. J. and von Schaewen, A. (2003). The oxidative pentose phosphate pathway: structure and organisation. *Current Opinion in Plant Biology*, 6(3):236 – 246.

Küffner, R., Petri, T., Windhager, L., and Zimmer, R. (2010). Petri nets with fuzzy logic (pnfl): Reverse engineering and parametrization. *PLoS ONE*, 5(9):e12807.

Kupke, T., Hernández-Acosta, P., Steinbacher, S., and Culiáñez-Macià, F. A. (2001). Arabidopsis thaliana flavoprotein athal3a catalyzes the decarboxylation of 4'-phosphopantothenoylcysteine to 4'-phosphopantetheine, a key step in coenzyme a biosynthesis. *Journal of Biological Chemistry*, 276(22):19190–19196.

Laherrère, J. and Sornette, D. (1998). Stretched exponential distributions in nature and economy: "fat tails" with characteristic scales. *The European Physical Journal B - Condensed Matter and Complex Systems*, 2(4):525–539.

Lam, H. M., Peng, S., and Coruzzi, G. M. (1994). Metabolic regulation of the gene encoding glutamine-dependent asparagine synthetase in arabidopsis thaliana. *Plant Physiology*, 106(4):1347–1357.

Lautenbach, K. (1973). *Exakte Bedingungen der Lebendigkeit für eine Klasse von Petri-Netzen*. Number 82. Gesellschaft für Mathematik und Datenverarbeitung. In German.

Lautenbach, K. and Ridder, H. (1994). *Liveness in bounded Petri nets which are covered by T-invariants*. Springer.

Lee, D.-Y., Zimmer, R., Lee, S. Y., and Park, S. (2006a). Colored petri net modeling and simulation of signal transduction pathways. *Metabolic Engineering*, 8(2):112 – 122.

Lee, J. M., Gianchandani, E. P., and Papin, J. A. (2006b). Flux balance analysis in the era of metabolomics. *Briefings in Bioinformatics*, 7(2):140–150.

Lee-kwang, H., Favrel, J., and Baptiste, P. (1987). Generalized petri net reduction method. *Systems, Man and Cybernetics, IEEE Transactions on*, 17(2):297–303.

Lefebvre, S., Lawson, T., Zakhleniuk, O., Lloyd, J., Raines, C., and Fryer, M. (2005). Increased sedoheptulose-1,7-bisphosphatase activity in transgenic tobacco plants stimulates photosynthesis and growth from an early stage in development. *Plant Physiology*, 138(1):451–60.

Lei, Z., Huhman, D. V., and Sumner, L. W. (2011). Mass spectrometry strategies in metabolomics. *The Journal of Biological Chemistry*, 286:25435–25442.

Leiserson, C. E., Rivest, R. L., Stein, C., and Cormen, T. H. (2001). *Introduction to algorithms*. The MIT press.

Libourel, I. G. and Shachar-Hill, Y. (2008). Metabolic flux analysis in plants: From intelligent design to rational engineering. *Annual Review of Plant Biology*, 59(1):625–650.

Liepman, A. H. and Olsen, L. J. (2001). Peroxisomal alanine: glyoxylate aminotransferase (agt1) is a photorespiratory enzyme with multiple substrates in arabidopsis thaliana. *The Plant Journal*, 25(5):487–498.

Liepman, A. H. and Olsen, L. J. (2003). Alanine aminotransferase homologs catalyze the glutamate: Glyoxylate aminotransferase reaction in peroxisomes of arabidopsis. *Plant Physiology*, 131(1):215–227.

Lima-Mendez, G. and van Helden, J. (2009). The powerful law of the power law and other myths in network biology. *Molecular BioSystems*, 5(12):1482–1493.

Lin, M. and Oliver, D. J. (2008). The role of acetyl-coenzyme a synthetase in arabidopsis. *Plant Physiology*, 147(4):1822–1829.

Linka, M. and Weber, A. P. (2005). Shuffling ammonia between mitochondria and plastids during photorespiration. *Trends in Plant Science*, 10(10):461 – 465.

Litterer, L., Schnurr, J., Plaisance, K., Storey, K., Gronwald, J., and Somers, D. (2006). Characterization and expression of arabidopsis udp-sugar pyrophosphorylase. *Plant Physiology and Biochemistry*, 44(4):171 – 180.

Lu, Y. and Sharkey, T. (2006). The importance of maltose in transitory starch breakdown. *Plant, Cell & Environment*, 29(3):353–66.

Lucas, M. D. and Brady, S. M. (2013). Gene regulatory networks in the arabidopsis root. *Current Opinion in Plant Biology*, 16(1):50 – 55.

Ludwig, R. A. (1993). Arabidopsis chloroplasts dissimilate l-arginine and l-citrulline for use as n source. *Plant Physiology*, 101(2):429–434.

Ma, H.-W., Zhao, X.-M., Yuan, Y.-J., and Zeng, A.-P. (2004). Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics*, 20(12):1870–1876.

Martiník, I. (2011). Bi-relational p/t petri nets and the modeling of multithreading object-oriented programming systems. In Snasel, V., Platos, J., and El-Qawasmeh, E., editors, *Digital Information Processing and Communications*, volume 188 of *Communications in Computer and Information Science*, pages 222–236. Springer Berlin Heidelberg.

Masakapalli, S. K., Le Lay, P., Huddleston, J. E., Pollock, N. L., Kruger, N. J., and Ratcliffe, R. G. (February 2010). Subcellular flux analysis of central metabolism in a heterotrophic arabidopsis cell suspension using steady-state stable isotope labeling. *Plant Physiology*, 152(2):602–619.

Matsuno, H., Doi, A., Nagasaki, M., and Miyano, S. (2000). Hybrid petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing*, volume 5, page 87. World Scientific Press Singapore.

Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., and Miyano, S. (2003). Biopathways representation and simulation on hybrid functional petri net. *In silico biology*, 3(3):389–404.

Matsuoka, M. (1995). The gene for pyruvate,orthophosphate dikinase in c4 plants: Structure, regulation and evolution. *Plant and Cell Physiology*, 36(6):937–943.

Mintz-Oron, S., Meir, S., Malitsky, S., Ruppin, E., Aharoni, A., and Shlomi, T. (2012). Reconstruction of arabidopsis metabolic network models accounting for subcellular compartmentalization and tissue-specificity. *Proceedings of the National Academy of Sciences*, 109(1):339–344.

Mittler, R. and Blumwald, E. (2010). Genetic engineering for modern agriculture: Challenges and perspectives. *Annual Review of Plant Biology*, 61:443–462.

Moffatt, B. A. and Ashihara, H. (2002). Purine and pyrimidine nucleotide synthesis and metabolism. *The Arabidopsis Book/American Society of Plant Biologists*, 1.

Mohring, R. and Radermacher, F. (1984). Substitution decomposition for discrete structures and connections nith. *Algebraic and combinatorial methods in operations research*, 19:257.

Monteiro, P. T., Ropers, D., Mateescu, R., Freitas, A. T., and de Jong, H. (2008). Temporal logic patterns for querying dynamic models of cellular interaction networks. *Bioinformatics*, 24(16):i227–i233.

Morgan, J. A. and Rhodes, D. (2002). Mathematical modeling of plant metabolic pathways. *Metabolic Engineering*, 4(1):80 – 89.

Morgan, M. A. and Jackson, W. A. (1988). Inward and outward movement of ammonium in root systems: Transient responses during recovery from nitrogen deprivation in presence of ammonium. *Journal of Experimental Botany*, 39(2):179–191.

Mueller, L. A., Zhang, P., and Rhee, S. Y. (2003). Aracyc: A biochemical pathway database for arabidopsis. *Plant Physiology*, 132(2):453–460.

Mura, I. and Csikász-Nagy, A. (2008). Stochastic petri net extension of a yeast cell cycle model. *Journal of Theoretical Biology*, 254(4):850 – 860.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Negm, F. B. and Loescher, W. H. (1981). Characterization and partial purification of aldose-6-phosphate reductase (alditol-6-phosphate:nadp 1-oxidoreductase) from apple leaves. *Plant Physiology*, 67(1):139–142.

Nelson, D. E., Koukoumanos, M., and Bohnert, H. J. (1999). Myo-inositol-dependent sodium uptake in ice plant. *Plant Physiology*, 119(1):165–172.

Nelson, D. L. and Cox, M. M. (2000). *Lehninger principles of biochemistry*. W. H. Freeman, New York, 3rd edition.

Newman, M. E. (2005). Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

Noguchi, K. and Yoshida, K. (2008). Interaction between photosynthesis and respiration in illuminated leaves. *Mitochondrion*, 8(1):87 – 99.

Noiraud, N., Maurousset, L., and Lemoine, R. (2001). Transport of polyols in higher plants. *Plant Physiology and Biochemistry*, 39(9):717 – 728.

Nosarzewski, M., Downie, A. B., Wu, B., and Archbold, D. D. (2012). The role of sorbitol dehydrogenase in arabidopsis thaliana. *Functional Plant Biology*, 39:462–470.

Nöthen, J. (2009). Metabolische netzwerke in pflanzen. *Diploma-thesis, Johann Wolfgang Goethe-University, Frankfurt am Main, Germany*. In German.

Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature biotechnology*, 28(3):245–248.

Papasotiriou, D. G., Markoutsa, S., Gorka, J., Schleiff, E., Karas, M., and Meyer, B. (2013). Maldi analysis of proteins after extraction from dissolvable ethylene glycol diacrylate cross-linked polyacrylamide gels. *ELECTROPHORESIS*, 34(17):2484–2494.

Parry, M. A. J., Andralojc, P. J., Mitchell, R. A. C., Madgwick, P. J., and Keys, A. J. (2003). Manipulation of rubisco: the amount, activity, function and regulation. *Journal of Experimental Botany*, 54(386):1321–1333.

Paul, M. J., Primavesi, L. F., Jhurreea, D., and Zhang, Y. (2008). Trehalose metabolism and signaling. *Annual Review of Plant Biology*, 59(1):417–441.

Peltier, J.-B., Cai, Y., Sun, Q., Zabrouskov, V., Giacomelli, L., Rudella, A., Ytterberg, A. J., Rutschow, H., and van Wijk, K. J. (2006). The oligomeric stromal proteome of arabidopsis thaliana chloroplasts. *Molecular and Cellular Proteomics*, 5(1):114–133.

Pérès, S., Beurton-Aimar, M., and Mazat, J. P. (2006). Pathway classification of TCA cycle. *IEE Proceedings - Systems Biology*, 153:369–371(2).

Pérès, S., Vallée, F., Beurton-Aimar, M., and Mazat, J. (2011). ACoM: A classification method for elementary flux modes based on motif finding. *Biosystems*, 103(3):410 – 419.

Petri, C. A. (1962). Kommunikation mit automaten. *Ph.D. Thesis*. In German.

Pinney, J. W., Westhead, D. R., and McConkey, G. A. (2003). Petri net representations in systems biology. *Biochemical Society Transactions*, 31(6):1513–1515.

Plaxton, W. C. (1996). The organization and regulation of plant glycolysis. *Annual Review of Plant Physiology and Plant Molecular Biology*, 47(1):185–214.

Plaxton, W. C. and Podestá, F. E. (2006). The functional organization and control of plant respiration. *Critical Reviews in Plant Sciences*, 25(2):159–198.

Polacco, J. C. and Holland, M. A. (1993). Roles of urease in plant cells. *International review of cytology*, pages 65–65.

Poolman, M., Bonde, B., Gevorgyan, A., Patel, H., and Fell, D. (2006). Challenges to be faced in the reconstruction of metabolic networks from public databases. *IEE Proceedings-Systems Biology*, 153(5):379–384.

Poolman, M. G., Miguet, L., Sweetlove, L. J., and Fell, D. A. (2009). A genome-scale metabolic model of arabidopsis and some of its properties. *Plant Physiology*, 151(3):1570–1581.

Radrich, K., Tsuruoka, Y., Dobson, P., Gevorgyan, A., Swainston, N., Baart, G., and Schwartz, J.-M. (2010). Integration of metabolic databases for the reconstruction of genome-scale metabolic networks. *BMC Systems Biology*, 4:114.

Raghavendra, A. S. and Padmasree, K. (2003). Beneficial interactions of mitochondrial metabolism with photosynthetic carbon assimilation. *Trends in Plant Science*, 8(11):546 – 553.

Raines, C. (2003). The calvin cycle revisited. *Photosynthesis Research*, 75(1):1–10.

Raman, S. B. and Rathinasabapathi, B. (2004). Pantothenate synthesis in plants. *Plant Science*, 167(5):961 – 968.

Ratcliffe, R. and Shachar-Hill, Y. (2006). Measuring multiple fluxes through plant metabolic networks. *The Plant Journal*, 45(4):490–511.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555.

Reaves, M. L. and Rabinowitz, J. D. (2011). Metabolomics in systems microbiology. *Current Opinion in Biotechnology*, 22(1):17 – 25.

Reddy, V. N., Mavrovouniotis, M. L., and Liebman, M. N. (1993). Petri net representations in metabolic pathways. In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, pages 328–336. AAAI Press.

Reed, J. L. and Palsson, B. O. (2003). Thirteen years of building constraint-based in silico models of escherichia coli. *Journal of Bacteriology*, 185(9):2692–2699.

Reinbothe, H. and Mothes, K. (1962). Urea, ureides, and guanidines in plants. *Annual Review of Plant Physiology*, 13(1):129–149.

Reiter, W. (2008). Biochemical genetics of nucleotide sugar interconversion reactions. *Current Opinion in Plant Biology*, 11:236–43.

Rios-Estepa, R. and Lange, B. M. (2007). Experimental and mathematical approaches to modeling plant metabolic networks. *Phytochemistry*, 68(16–18):2351 – 2374.

Ríos-Mercado, R., Wu, S., Scott, L., and Boyd, E. (2002). A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research*, 117(1-4):217–234.

Rippert, P. and Matringe, M. (2002). Molecular and biochemical characterization of an arabidopsis thaliana arogenate dehydrogenase with two highly similar and active protein domains. *Plant Molecular Biology*, 48:361–368.

Roy, H. and Andrews, T. (2004). Rubisco: Assembly and mechanism. In Leegood, R., Sharkey, T., and Caemmerer, S., editors, *Photosynthesis*, volume 9 of *Advances in Photosynthesis and Respiration*, pages 53–83. Springer Netherlands.

Sackmann, A., Heiner, M., and Koch, I. (2006). Application of petri net based analysis techniques to signal transduction pathways. *BMC bioinformatics*, 7(1):482.

Sadiq, W. and Orlowska, M. E. (2000). Analyzing process models using graph reduction techniques. *Information Systems*, 25(2):117 – 134.

Saha, R., Suthers, P. F., and Maranas, C. D. (2011). Zea mays irs1563: A comprehensive genome-scale metabolic reconstruction of maize metabolism. *PLoS ONE*, 6(7):e21784.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Sauer, N., Friedländer, K., and Gräml-Wicke, U. (1990). Primary structure, genomic organization and heterologous expression of a glucose transporter from arabidopsis thaliana. *The EMBO Journal*, 9(10):3045.

Sauer, N. and Stolz, J. (1994). Suc1 and suc2: two sucrose transporters from arabidopsis thaliana; expression and characterization in baker's yeast and identification of the histidine-tagged protein. *The Plant Journal*, 6(1):67–77.

Sauer, U. (2006). Metabolic networks in motion: 13c-based flux analysis. *Molecular systems biology*, 2(1).

Schellenberger, J., Zielinski, D., Choi, W., Madireddi, S., Portnoy, V., Scott, D., Reed, J., Osterman, A., and Palsson, B. (2012). Predicting outcomes of steady-state 13c isotope tracing experiments using monte carlo sampling. *BMC Systems Biology*, 6(1):9.

Schilling, C. H., Covert, M. W., Famili, I., Church, G. M., Edwards, J. S., and Palsson, B. O. (2002). Genome-scale metabolic model of helicobacter pylori 26695. *Journal of Bacteriology*, 184(16):4582–4593.

Schleiff, E. (2010). personal communication. Professor Enrico Schleiff, Institute for Bioscience, Goethe-University, Frankfurt, Germany.

Schmid, J. and Amrhein, N. (1995). Molecular organization of the shikimate pathway in higher plants. *Phytochemistry*, 39(4):737 – 749.

Schnarrenberger, C. and Martin, W. (2002). Evolution of the enzymes of the citric acid cycle and the glyoxylate cycle of higher plants. *European Journal of Biochemistry*, 269(3):868–883.

Schuster, S., Fell, D. A., and Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature biotechnology*, 18(3):326–332.

Schuster, S. and Hilgetag, C. (1994). On elementary flux modes in biochemical reaction systems at steady state. *Journal of Biological Systems*, 02(02):165–182.

Schuster, S., Hilgetag, C., Woods, J., and Fell, D. (2002a). Reaction routes in biochemical reaction systems: Algebraic properties, validated calculation procedure and example from nucleotide metabolism. *Journal of Mathematical Biology*, 45(2):153–181.

Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. (2002b). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to mycoplasma pneumoniae. *Bioinformatics*, 18(2):351–361.

Schwender, J. (2008). Metabolic flux analysis as a tool in metabolic engineering of plants. *Current Opinion in Biotechnology*, 19(2):131 – 137.

Shastri, A. A. and Morgan, J. A. (2005). Flux balance analysis of photoautotrophic metabolism. *Biotechnology Progress*, 21(6):1617–1626.

Shatz, S., Tu, S., Murata, T., and Duri, S. (1996). An application of petri net reduction for ada tasking deadlock analysis. *Parallel and Distributed Systems, IEEE Transactions on*, 7(12):1307–1322.

Sifakis, J. (1978). Structural properties of petri nets. In Winkowski, J., editor, *Mathematical Foundations of Computer Science 1978*, volume 64 of *Lecture Notes in Computer Science*, pages 474–483. Springer Berlin Heidelberg.

Simão, E., Remy, E., Thieffry, D., and Chaouiya, C. (2005). Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in e.coli. *Bioinformatics*, 21(suppl 2):ii190–ii196.

Simm, S., Papasotiriou, D. G., Ibrahim, M., Leisegang, M. S., Müller, B., Schorge, T., Karas, M., Mirus, O., Sommer, M. S., and Schleiff, E. (2013). Defining the core proteome of the chloroplast envelope membranes. *Frontiers in plant science*, 4.

Sirko, A. and Brodzik, R. (2000). Plant ureases: roles and regulation. *Acta Biochimica Polonica*, 47(4):1189–95.

Srinivasan, A. and Bain, M. (2012). Knowledge-guided identification of petri net models of large biological systems. In Muggleton, S., Tamaddoni-Nezhad, A., and Lisi, F., editors, *Inductive Logic Programming*, volume 7207 of *Lecture Notes in Computer Science*, pages 317–331. Springer Berlin Heidelberg.

Srinivasan, A. and King, R. D. (2008). Incremental identification of qualitative models of biological systems using inductive logic programming. *J. Mach. Learn. Res.*, 9:1475–1533.

Starke, P. H. (1990). *Analyse von Petri-Netz-Modellen*. BG Teubner, Stuttgart. In German.

Starke, P. H. (2003). Integrated net analyzer. http://www2.informatik.hu-berlin.de/lehrstuehle/automaten/ina/.

Stephanopoulos, G. and Stafford, D. E. (2002). Metabolic engineering: a new frontier of chemical reaction engineering. *Chemical Engineering Science*, 57(14):2595 – 2602.

Strawn, M. A., Marr, S. K., Inoue, K., Inada, N., Zubieta, C., and Wildermuth, M. C. (2006). Arabidopsis isochorismate synthase functional in pathogen-induced salicylate biosynthesis exhibits properties consistent with a role in diverse stress responses. *The Journal of Biological Chemistry*, 282:5919–5933.

Streb, S. and Zeeman, S. C. (2012). Starch metabolism in arabidopsis. *The Arabidopsis book/American Society of Plant Biologists*, 10.

Sweetlove, L. J. and Fernie, A. R. (2013). The spatial organization of metabolism within the plant cell. *Annual Review of Plant Biology*, 64(1):723–746.

Sweetlove, L. J. and Ratcliffe, R. G. (2011). Flux-balance modeling of plant metabolism. *Frontiers in plant science*, 2.

Szydlowski, N., Ragel, P., Raynaud, S., Lucas, M. M., Roldán, I., Montero, M., Muñoz, F. J., Ovecka, M., Bahaji, A., Planchot, V., Pozueta-Romero, J., D'Hulst, C., and Mérida, A. (2009). Starch granule initiation in arabidopsis requires the presence of either class iv or class iii starch synthases. *The Plant Cell Online*, 21(8):2443–2457.

Tao, Y., Ferrer, J.-L., Ljung, K., Pojer, F., Hong, F., Long, J. A., Li, L., Moreno, J. E., Bowman, M. E., Ivans, L. J., Cheng, Y., Lim, J., Zhao, Y., Ballaré, C. L., Sandberg, G., Noel, J. P., and Chory, J. (2008). Rapid synthesis of auxin via a new tryptophan-dependent pathway is required for shade avoidance in plants. *Cell*, 133(1):164 – 176.

Tattersall, J. J. (2005). *Elementary number theory in nine chapters*. Cambridge University Press.

Temple, S. J., Vance, C. P., and Gantt, J. S. (1998). Glutamate synthase and nitrogen assimilation. *Trends in Plant Science*, 3(2):51 – 56.

Thormann, A., Rudolph, K., Ackermann, J., and Koch, I. (2009). TInA - (T-Invariant Analysis): a tool box for exploring pathways in biochemical systems at steady state. In *Abstract Book of GCB 2009*, pages 157–158.

Tischner, R., Galli, M., Heimer, Y. M., Bielefeld, S., Okamoto, M., Mack, A., and Crawford, N. M. (2007). Interference with the citrulline-based nitric oxide synthase assay by argininosuccinate lyase activity in arabidopsis extracts. *FEBS Journal*, 274(16):4238–4245.

Tomar, N., Choudhury, O., Chakrabarty, A., and De, R. (2013). An integrated pathway system modeling of saccharomyces cerevisiae hog pathway: a petri net based approach. *Molecular Biology Reports*, 40(2):1103–1125.

Tomar, N. and De, R. K. (2013). Comparing methods for metabolic network analysis and an application to metabolic engineering. *Gene*, 521(1):1 – 14.

Trinh, C. T., Wlaschin, A., and Srienc, F. (2009). Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Applied Microbiology and Biotechnology*, 81(5):813–826.

Tymowska-Lalanne, Z. and Kreis, M. (1998). Expression of the arabidopsis thaliana invertase gene family. *Planta*, 207(2):259–265.

Urano, K., Kurihara, Y., Seki, M., and Shinozaki, K. (2010). 'omics' analyses of regulatory networks in plant abiotic stress responses. *Current Opinion in Plant Biology*, 13(2):132 – 138.

US Patent No. 5790415 (1998).

Uzam, M. (2004). The use of the petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 23(3-4):204–219.

van der Aalst, W. M. P. (1998). The application of petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 08(01):21–66.

Vasilevski, A., Giorgi, F. M., Bertinetti, L., and Usadel, B. (2012). Lasso modeling of the arabidopsis thaliana seed/seedling transcriptome: a model case for detection of novel mucilage and pectin metabolism genes. *Molecular BioSystems*, 8(10):2566–2574.

von Wirén, N., Gazzarrini, S., Gojont, A., and Frommer, W. B. (2000). The molecular physiology of ammonium uptake and retrieval. *Current Opinion in Plant Biology*, 3(3):254 – 261.

Voss, K., Heiner, M., and Koch, I. (2003). Steady state analysis of metabolic pathways using petri nets. *In Silico Biology*, 3(3):367–387.

Wagner, A. and Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1478):1803–1810.

Wang, R.-S., Saadatpour, A., and Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001.

Weaver, L. M. and Herrmann, K. M. (1997). Dynamics of the shikimate pathway in plants. *Trends in plant science*, 2(9):346–351.

Wei, Y., Lin, M., Oliver, D., and Schnable, P. (2007). The roles of aldehyde dehydrogenases (aldhs) in the pdh bypass of arabidopsis. *BMC Biochemistry*, 10(7).

Weis, B. L., Schleiff, E., and Zerges, W. (2013). Protein targeting to subcellular organelles via mrna localization. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1833(2):260 – 273.

West, D. B. et al. (2001). *Introduction to graph theory*, volume 2. Prentice hall Englewood Cliffs.

Wiechert, W. (2001). 13c metabolic flux analysis. *Metabolic Engineering*, 3(3):195 – 206.

Wiechert, W. (2002). Modeling and simulation: tools for metabolic engineering. *Journal of Biotechnology*, 94(1):37 – 63.

Wiechert, W. and Nöh, K. (2013). Isotopically non-stationary metabolic flux analysis: complex yet highly informative. *Current Opinion in Biotechnology*, 24(6):979–986.

Wilkie, S. E. and Warren, M. J. (1998). Recombinant expression, purification, and characterization of three isoenzymes of aspartate aminotransferase from arabidopsis thaliana. *Protein Expression and Purification*, 12(3):381 – 389.

Williams, T. C., Miguet, L., Masakapalli, S. K., Kruger, N. J., Sweetlove, L. J., and Ratcliffe, R. G. (2008). Metabolic network fluxes in heterotrophic arabidopsis cells: Stability of the flux distribution under different oxygenation conditions. *Plant Physiology*, 148(2):704–718.

Winder, C. L., Dunn, W. B., and Goodacre, R. (2011). Tardis-based microbial metabolomics: time and relative differences in systems. *Trends in Microbiology*, 19(7):315 – 322.

Wirtz, M. and Hell, R. (2006). Functional analysis of the cysteine synthase protein complex from plants: Structural, biochemical and regulatory properties. *Journal of Plant Physiology*, 163(3):273 – 286.

Wolucka, B. A. and Montagu, M. V. (2007). The {VTC2} cycle and the de novo biosynthesis pathways for vitamin c in plants: An opinion. *Phytochemistry*, 68(21):2602 – 2613.

Yates, J. R. (2013). The revolution and evolution of shotgun proteomics for large-scale proteome analysis. *Journal of the American Chemical Society*, 135(5):1629–1640.

Zaitsev, D. A. (2004). Decomposition-based calculation of petri net invariants. In *Proceedings of Token based computing Workshop of the 25-th International conference on application and theory of Petri nets, Bologna, Italy*, pages 79–83.

Zevedei-Oancea, I. and Schuster, S. (2003). Topological analysis of metabolic networks based on petri net theory. *In Silico Biology*, 3:323–345.

Zhang, G.-F., Sadhukhan, S., Tochtrop, G. P., and Brunengraber, H. (2011). Metabolomics, pathway regulation, and pathway discovery. *The Journal of Biological Chemistry*, 286:23631–23635.

Zhang, P., Dreher, K., Karthikeyan, A., Chi, A., Pujar, A., Caspi, R., Karp, P., Kirkup, V., Latendresse, M., Lee, C., Mueller, L. A., Muller, R., and Rhee, S. Y. (2010). Creation of a genome-wide metabolic pathway database for populus trichocarpa using a new approach for reconstruction and curation of metabolic pathways for plants. *Plant Physiology*, 153(4):1479–1491.

Zhang, P., Foerster, H., Tissier, C. P., Mueller, L., Paley, S., Karp, P. D., and Rhee, S. Y. (2005). Metacyc and aracyc. metabolic pathway databases for plant research. *Plant Physiology*, 138(1):27–37.

Zrenner, R., Stitt, M., Sonnewald, U., and Boldt, R. (2006). Pyrimidine and purine biosynthesis and degradation in plants. *Annual Review of Plant Biology*, 57(1):805–836.

Appendix A

General Definitions

Mathematical Foundations The *greatest common divisor* of a and b is denoted by $gcd(a,b)$, the *least common multiple* by $lcm(a,b)$.
The set of the natural numbers, $\mathbb{N}$, contains 0. The set $\mathbb{N}_{>k}$ represents all natural numbers greater than k.
If x is a vector $x = 0$ means every position x_i of the vector is zero. $x > 0$ means every x_i is greater than zero.

Graphs A *graph* is a representation of a set of objects, called *vertices* or *nodes*. In the further subsections we will refer to them as vertices. This objects are connected by links, called *edges* or *arcs*. In a formal way, a graph is the set $G = (V,E)$, where V are the vertices and $E = \{(x,y)|x,y \in V\}$ are the edges.

Weighted Graph The *weighted graph* is built from a set of vertices V and a set of edges E, where the function $f : e \in E \to \mathbb{R}$ assign a number to each edge e, representing for example costs or distances.

Directed Graph In the *directed graph*, the links between the vertices are called *arcs*. $A = \{(x,y)|x,y \in V\}$ is a set of ordered pairs, each denoting an arc from a vertex x to a vertex y. x is called *direct predecessor* of y, and y is a *direct successor* of x.

Bipartite Graph In a *bipartite graph* the vertices can be divided into two disjoint set U and V in such way that no edge links two vertices of U or two vertices of V. A bipartite graph can be written as $G = (U,V,E)$, where $E = \{(x,y)|x \in V \wedge y \in U \vee x \in U \wedge y \in V\}$ is the set of edges.

Appendix B

Tables

B.1 Table of Metabolites

Table B.1: Metabolites of the model

No.	Name
1	ornithine
2	glutamate
3	γ-aminobutyric acid (GABA)
4	succinate
5	succinyl-CoA
6	fumarate
7	feruloyl-CoA
8	malate
9	α-ketoglutarate
10	isocitrate
11	citrate
12	oxalacetate
13	glyoxylate
14	glycerate
15	hydroxy-pyruvate
16	serine
17	coniferyl aldehyde
18	glycine
19	coniferyl alcohol
20	pyruvate
21	L-alanine
22	Not modeled
23	L-aspartate
24	uracil
25	urea
26	asparagine
27	L-arginino-succinate
28	arginine
29	ammonia

Table B.1: (continued)

No.	Name
30a + 30b	lignin
31	hydroxy-coniferyl alcohol
32	D-erythrose 4-phosphate
33	glutamine
34	phospho*enol*pyruvate
35	glycolate
36	7-phospho-2-dehydro-3-deoxy-D-arabinoheptonate
37	3-dehydroquinate
38	acetate
39	glycerate 2-phosphate
40	glycerate 3-phosphate
41	glycerate 1,3-bisphosphate
42	glyceraldehyde 3-phosphate
43	β-D-fructose 1,6-phosphate
44	β-D-fructose 6-phosphate
45	D-xylulose 5-phosphate
46	D-ribose 5-phosphate
47	coumaroyl-CoA
48	sinapyl alcohol
49	β-D-glucose 6-phosphate
50	α-D-glucose 1-phosphate
51	β-D-glucose
52	D-ribulose 5-phosphate
53	6-phospho gluconate
54	D-glucono-1,5-lactone 6-phosphate
55	sinapate
56	α-D-glucose 6-phosphate
57	α-D-glucose
58	ADP-Glucose
59	starch
60	ribulose-1,5-biphosphate
61	maltose
62	sorbitol
63	D-fructose
64	2-phospho glycolate
65	sedoheptulose 7-phosphate
66	sucrose
67	sedoheptulose 1,7-biphosphate
68	dihydroxyacetone phosphate
69	1-L-*myo*-inositol 1-phosphate
70	raffinose
71	stachyose
72	galactinol
73	D-galactose
74	D-*myo*-inositol
75	galactane

Table B.1: (continued)

No.	Name
76	α-D-galactose 1-phosphate
77	UDP-galactose
78	L-galactono-1,4-lactone
79	succinate semialdehyde
80	uridine 5'-phosphate (UMP)
81	amylopectin
82	UDP-glucose
83	trehalose
84	trehalose 6-phosphate
85	UDP-D-glucuronate
86	amylose
87	D-galacturonate
88	UDP-xylose
89	heteroglycan
90	caffeoyl shikimic acid
91	acetyl-L-lserine
92	acetyl-CoA
93	coenzyme A (CoA)
94	citrulline
95	3-dehydroshikimate
96	shikimate 3-phosphate
97	5-enolpyruvylshikimate 3-phosphate
98	L-arogenate
99	homogentisate
100	coumaroyl shikimic acid
101	quinate
102	shikimate
103	chorismate
104	coumaroyl quinic acid
105	prephenate
106	4-hydroxy-phenylpyrovate
107	tyrosine
108	caffeoyl-CoA
109	phenylalanine
110	trans-cinnamate
111	caffeoyl quinic acid
112	p-coumaric acid
113	cysteine
114	acetaldehyde
115	uridine diphosphate (UDP)
116	carbamoyl phosphate
117	carbamoyl aspartate
118	dihydroorotate
119	orotic acid
120	orotidine 5'-monophosphate
121	5-phosphoribosyl 1-pyrophosphate

Table B.1: (continued)

No.	Name
122	cytidine triphosphate
123	cytidine monophosphate (CMP)
124	dihydrouracil
125	β-ureidopropionate
126	uridine
127	cytidine
128	cytidine diphosphate (CDP)
129	uridine triphosphate (UTP)
130	L-galactonate
131	UDP-galacturonate
132	D-galacturonate 1-phosphate
133	hydroxy-coniferyl aldehyde

B.2 Table of Reactions

Table B.2: Transitions of the model. The column Transitions give the name used in the network. Biochemical Pathway denotes the biochemical pathways, in which the reaction occurs. Enzyme name gives the name of the enzyme catalyzing the reaction. Literature gives the publications in which the reaction can be found, and Comments and reactions gives the catalyzed reactions and comments for special treatments. The number in the comments and reactions column are the metabolite encoding, which can be found in Table B.1.

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E1	1. ribose degradation 2. pentose phosphate pathway (non-oxidative branch) 3. Calvin-Benson-Bassham cycle 4. Rubisco shunt	transketolase	(Buchanan et al., 2000)	1. 45+46 $\rightarrow$ 42+65 2. 45+46 $\rightarrow$ 42+65 & 32+45 $\rightarrow$ 42+44 3. 42+65 $\rightarrow$ 45+46 4. 42+44 $\rightarrow$ 32+45 & 42+65 $\rightarrow$ 45+46
E2	aspartate biosynthesis	aspartate transaminase	(Wilkie and Warren, 1998; Graindorge et al., 2010)	2+12 $\leftrightarrow$ 23+9
E3	1. ribose degradation 2. pentose phosphate pathway (non-oxidative branch) 3. Calvin-Benson-Bassham cycle 4. Rubisco shunt	ribose-5-phosphate isomerase	(Berg et al., 2002; Buchanan et al., 2000)	1. 46 $\rightarrow$ 52 2. 52 $\rightarrow$ 46 3. 46 $\rightarrow$ 52 4. 46 $\rightarrow$ 52
E4	pentose phosphate pathway (oxidative branch)	phosphogluconate dehydrogenase	(Dey and Harborne, 1997)	53 $\rightarrow$ 52
E5	pentose phosphate pathway (oxidative branch)	6-phosphogluconolactonase	(Dey and Harborne, 1997)	54 $\rightarrow$ 53
E6	pentose phosphate pathway (oxidative branch)	glucose-6-phosphate 1-dehydrogenase	(Dey and Harborne, 1997)	49 $\leftrightarrow$ 54
E7	Calvin-Benson-Bassham cycle	sedoheptulose-bisphosphatase	(Lefebvre et al., 2005)	67 $\rightarrow$ 65
E8	1. starch biosynthesis 2. gluconeogenesis 3. sucrose degradation III 4. glycolysis I (plastidic)	spontaneous	(Nelson and Cox, 2000)	1. 49 $\rightarrow$ 56 2. 56 $\rightarrow$ 49 3. 49 $\leftrightarrow$ 56 4. 49 $\leftrightarrow$ 56

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E9	1. starch degradation 2. starch biosynthesis 3. sucrose biosynthesis 4. UDP-glucose biosynthesis (from glucose 6-phosphate) 5. sucrose degradation III 6. superpathway of sucrose and starch metabolism I (non-photosynthetic tissue)	phosphoglucomutase	(Reiter, 2008)	1. 50 → 56 2. 56 → 50 3. 56 → 50 4. 56 → 50 5. 50 → 56 6. 50 → 56
E10	1. starch degradation 2. sucrose degradation III	hexokinase/glucokinase	(Lu and Sharkey, 2006; Guy et al., 2008)	57 → 56
E11	Calvin-Benson-Bassham cycle	aldolase	(Buchanan et al., 2000)	32 + 68 ↔ 67
E12	sucrose degradation III	fructokinase	(Guy et al., 2008)	63 → 44
E13	1. glycolysis IV (plant cytosol) 2. Calvin-Benson-Bassham cycle 3. glycolysis I (plastidic)	triose-phosphate isomerase	(Berg et al., 2002)	42 ↔ 68
E14	1. Gluconeogenesis 2. Calvin-Benson-Bassham cycle 3. glycolysis I (plastidic)	fructose-bisphosphate aldolase	(Guy et al., 2008)	43 ↔ 42+68
E15	1. Gluconeogenesis 2. Calvin-Benson-Bassham cycle 3. glycolysis I (plastidic)	fructose-1,6-bisphosphatase	(Guy et al., 2008)	43 → 44
E16	1. glycolysis I (plastidic) 2. glycolysis IV (plant cytosol)	6-phosphofructokinase	(Dey and Harborne, 1997)	44 → 43

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E17	1. galactose degradation II 2. galactose degradation III 3. sucrose biosynthesis 4. UDP-glucose biosynthesis (from glucose 6-phosphate) 5. sucrose degradation III 6. superpathway of sucrose and starch metabolism I (non-photosynthetic tissue) 7. superpathway of sucrose and starch metabolism II (photosynthetic tissue)	(Guy et al., 2008): UDP-glucose pyrophosphorylase (Reiter, 2008): UDP-D-galactose pyrophosphorylase	(Guy et al., 2008; Reiter, 2008)	(Guy et al., 2008): 50+129 → 82 (Reiter, 2008): 76 + 129 → 77
E18	1. galactose degradation III 2. sucrose biosynthesis 3. UDP-glucose biosynthesis (from sucrose) 4. sucrose degradation III	sucrose synthase	(Guy et al., 2008)	66+115 ↔ 63+82
E19	1. galactose degradation III 2. UDP-D-xylose biosynthesis	UDP-D-glucose dehydrogenasae	(Reiter, 2008)	82 → 85
E20	1. galactose degradation II 2. galactose degradation III 3. UDP-galactose biosynthesis 4. galactose degradation I (Leloir pathway)	UDP-glucose 4-epimerase	(Guy et al., 2008)	1. 77 → 82 2. 77 → 82 3. 77 ↔ 82 4. 77 → 82
E21	Non-enzymatically	Non-enzymatically H2O2 interaction	(Cousins et al., 2008)	15 → 35
E22	1. galactose degradation II , 2. galactose degradation III , 3. galactose degradation I (Leloir pathway)	D-galactokinase	(Reiter, 2008)	73 → 76
E23	UDP-D-xylose biosynthesis	UDP-D-glucuronate decarboxylase	(Reiter, 2008)	85 → 88
E24	galactose degradation I (Leloir pathway)	ADP:glucose-1-phosphate adenylyltransferase	(Berg et al., 2002)	76+82 ↔50+77
E25	ascorbate biosynthesis I (L-galactose pathway)	L-galactose dehydrogenase	(Davey et al., 1999)	73 → 78
E26	No pathway assigned in AraCyc	UDP-galacturonate pyrophosphatase	(Litterer et al., 2006)	131 ↔ 132 + 129
E27	stachyose biosynthesis	galactinol synthase	(Guy et al., 2008)	74+77 → 72+115
E28	stachyose biosynthesis	raffinose synthase	(Guy et al., 2008)	72+66 → 74+70

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E29	stachyose biosynthesis	stachyose synthase	(Guy et al., 2008)	70+72 → 71+74
E30	trehalose biosynthesis I	trehalose phosphate synthatase	(Paul et al., 2008)	56+82 → 84+115
E31	trehalose biosynthesis I	trehalose phosphate phosphatase	(Paul et al., 2008)	84 → 83
E32	trehalose biosynthesis I	trehalase	(Paul et al., 2008)	83 → 51+57
E33	myo-inositol biosynthesis	myo-inositol 1-phosphate monophosphatase	(Guy et al., 2008)	69 → 74
E34	1D-myo-inositol hexakisphosphate biosynthesis III (Spirodela polyrrhiza)	myo-inositol-1-phosphate synthase	(Guy et al., 2008)	56 ↔ 69
E35	pyrimidine salvage pathway	uracil phosphoribosyltransferase	(Zrenner et al., 2006)	24+121 → 80
E36	No pathway assigned	unknown	(Berg et al., 2002)	51+73 ↔ 75
E37	sucrose degradation III	vacuolar invertase	(Tymowska-Lalanne and Kreis, 1998)	66 → 51+63
E38	uridine-5'-phosphate biosynthesis	carbamoylphosphate synthase	(Zrenner et al., 2006)	33 → 2+116
E39	No pathway assigned	aldose 6-phosphate reductase	Hypothetical reaction, mentioned in: (Nosarzewski et al., 2012) enzyme name and reaction: (Negm and Loescher, 1981)	49 → 62
E40	citrulline biosynthesis arginine biosynthesis I arginine biosynthesis II (acetyl cycle)	ornithine transcarbamoylase	(Tischner et al., 2007)	1+116 → 94
E41	Not assigned			
E42	uridine-5'-phosphate biosynthesis	aspartate transcarbamoylase	(Zrenner et al., 2006)	23+116 → 117
E43	uridine-5'-phosphate biosynthesis	dihydroorotase	(Zrenner et al., 2006)	117 → 118
E44	uridine-5'-phosphate biosynthesis	dihydroorotate dehydrogenase	(Zrenner et al., 2006)	118 → 119
E45	No pathway assigned	alpha-galactosidase	(Gross and Pharr, 1982)	71 → 70+73 & 70 → 66+73
E46	starch biosynthesis	ADP glucose pyrophophorylase	(Kossmann and Lloyd, 2000)	50 → 58
E47	Not assigned			
E48	Not assigned			
E49	Not assigned			
E50	Not assigned			
E51	Not assigned			
E52	Not assigned			
E53	Not assigned			

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E54	Connection of glycolysis and pentose phophate pathway (Stryer)	transaldolase	(Berg et al., 2002)	32+44 $\leftrightarrow$ 42+65
E55	Not assigned			
E56	Not assigned			
E57	Not assigned			
E58	Not assigned			
E59	Not assigned			
E60	Not assigned			
E61	uridine-5'-phosphate biosynthesis	UMP synthase (orotate phosphoribosyl-transferase)	(Zrenner et al., 2006)	119+121 $\rightarrow$ 120 & 120 $\rightarrow$ 80
E62	pyrimidine ribonucleotides interconversion	uridylate/cytidylate kinase	(Zrenner et al., 2006)	80 $\rightarrow$ 115 & 123 $\rightarrow$ 128
E63	glycolysis	Enolase	(Plaxton, 1996)	39 $\leftrightarrow$ 34
E64	chorismate biosynthesis	3-deoxy-arabinoheptulonate 7-phosphate (DAHP) synthase	(Herrmann, 1995)	32+34 $\rightarrow$ 36
E65	chorismate biosynthesis	3-dehydroquinate synthase	(Herrmann, 1995)	36 $\rightarrow$ 37
E66	chorismate biosynthesis	1. 3-dehydroquinatedehydratase 2. quinate hydrolyase	(Herrmann, 1995)	1. 37 $\leftrightarrow$ 95 2. 101 $\leftrightarrow$ 102
E67	chorismate biosynthesis	1. quinate dehydrogenase 2. shikimate dehydrogenase 3. shikimate kinase	(Herrmann, 1995)	1. 37 $\leftrightarrow$ 101 2. 95 $\leftrightarrow$ 102 3. 102 $\rightarrow$ 96
E68	chorismate biosynthesis	5-enolpyruvylshikimate 3-phosphate (EPSP) synthase	(Herrmann, 1995)	34+96 $\rightarrow$ 97
E69	chorismate biosynthesis	chorismate synthase	(Herrmann, 1995)	97 $\rightarrow$ 103
E70	chorismate biosynthesis	chorismate mutase	(Strawn et al., 2006)	103 $\rightarrow$ 105
E71	phenylalanine biosynthesis II tyrosine biosynthesis II	prephenate aminotransferase	(Graindorge et al., 2010)	2+105 $\leftrightarrow$ 9+98
E72	tyrosine biosynthesis II	1. arogenate dehydrogenase 2. prephenate dehydrogenase	1. (Rippert and Matringe, 2002) 2. hypothetical, mentioned in (Rippert and Matringe, 2002)	1. 98 $\rightarrow$ 107 2. 105 $\rightarrow$ 106
E73	phenylalanine biosynthesis II	arogenate dehydratase	(Cho et al., 2007)	98 $\rightarrow$ 109
E74	tyrosine degradation I	4-hydroxyphenylpyruvate dioxygenase	(Garcia et al., 1999)	106 $\rightarrow$ 99

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E75	tyrosine biosynthesis I	tyrosine aminotransferase	(Tao et al., 2008; Holländer-Czytko et al., 2005)	9 + 107 ↔ 2 + 106
E76	Not assigned			
E77	phenylpropanoid biosynthesis	phenylalanine ammonialyase	(Humphreys and Chapple, 2002)	109 → 110 + 29
E78	phenylpropanoid biosynthesis	phenylalanine ammonialyase	(Humphreys and Chapple, 2002)	109 → 110 + 29
E79	phenylpropanoid biosynthesis	cinnamate 4-hydroxylase	(Humphreys and Chapple, 2002)	110 → 112
E80	phenylpropanoid biosynthesis	4-(hydroxy)cinnamoyl CoA ligase	(Humphreys and Chapple, 2002)	112 + 93 → 47
E81	phenylpropanoid biosynthesis	1. hydroxycinnamoyl CoA:quinate hydroxycinnamoyltransferase 2. hydroxycinnamoyl CoA:shikimate hydroxycinnamoyltransferase	(Humphreys and Chapple, 2002)	1. 93+111 → 101+108 & 47+101 → 93+104 2. 90+93 → 102+108 & 47+102 → 93+100
E82	phenylpropanoid biosynthesis	p-coumarate 3-hydroxylase	(Humphreys and Chapple, 2002)	100 → 90 & 104 → 111
E83	phenylpropanoid biosynthesis	caffeoyl CoA O-methyltransferase	(Humphreys and Chapple, 2002)	108 → 7
E84	phenylpropanoid biosynthesis	cinnamoyl CoA reductase	(Humphreys and Chapple, 2002)	7 → 17+93
E85	phenylpropanoid biosynthesis	cinnamyl alcohol dehydrogenase/ sinapyl alcohol dehydrogenase	(Humphreys and Chapple, 2002)	17 → 19
E86	phenylpropanoid biosynthesis	ferulate 5-hydroxylase	(Humphreys and Chapple, 2002)	19 → 31 & 17 → 133
E87	phenylpropanoid biosynthesis	caffeic acid/5-hydroxyferulic acid O-methyltransferase	(Humphreys and Chapple, 2002)	31 → 48 & 133 → 134
E88	phenylpropanoid biosynthesis		(Humphreys and Chapple, 2002)	19 → 30a & 48 → 30b (two different kinds of lignin)
E89	1. glycolysis IV (plant cytosol) 2. gluconeogenesis 3. glycolysis I (plastidic) 4. Rubisco shunt	phosphoglyceromutase	(Plaxton, 1996)	1. 40 → 39 2. 39 → 40 3. 39 ↔ 40 4. 40 → 39
E90	1. glycolysis IV (plant cytosol) 2. gluconeogenesis 3. Calvin-Benson-Bassham cycle 4. glycolysis I (plastidic)	3-PGA kinase	(Plaxton, 1996)	1. 41 → 40 2. 40 → 41 3. 40 → 41 4. 40 ↔ 41
E91	1. Calvin-Benson-Bassham cycle 2. gluconeogenesis	NAD-dependent GAPDH (phosphorylating)	(Plaxton, 1996)	1. 41 → 42 2. 41 ↔ 42

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E92	CO2 fixation into oxaloacetate glutamine biosynthesis III	phosphoenolpyruvate carboxylase	(Dey and Harborne, 1997)	34 → 12
E93	Gluconeogenesis superpathway of glyoxylate cycle	phosphoenolpyruvate carboxylase	(Cornah et al., 2004)	12 → 34
E94	TCA –cycle glyoxylate cycle	malate dehydrogenase	(Berg et al., 2002; Cornah et al., 2004)	8 ↔ 12
E95	sinapate ester biosynthesis	hydroxycinnamaldehyde dehydrogenase	(Fraser and Chapple, 2011)	134 → 55
E96	TCA –cycle	fumarate hydratase	(Berg et al., 2002; Cornah et al., 2004)	6 ↔ 8
E97	TCA –cycle	succinate dehydrogenase	(Berg et al., 2002; Cornah et al., 2004)	4 ↔ 6
E98	TCA –cycle glyoxylate cycle	citrate synthase	(Berg et al., 2002; Cornah et al., 2004)	12+92 → 11+93
E99	TCA –cycle glyoxylate cycle	aconitase	(Berg et al., 2002; Cornah et al., 2004)	11 → 10
E100	TCA –cycle	isocitrate dehydrogenase	(Berg et al., 2002; Cornah et al., 2004)	10 → 9
E101	TCA –cycle	α-ketoglutarate dehydrogenase	(Berg et al., 2002; Aubert et al., 2001)	9+93 → 5
E102	TCA –cycle	succinate-CoA ligase	(Berg et al., 2002; Cornah et al., 2004)	5 → 4+93 & 4+93 → 5
E103	glutamate degradation IV 4-aminobutyrate degradation IV	succinic semialdehyde dehydrogenase	(Bouché and Fromm, 2004)	79 → 4
E104	glutamate degradation IV 4-aminobutyrate degradation IV	gamma-aminobutyric acid transaminase	(Bouché and Fromm, 2004)	3+20 ↔ 21+79
E105	glutamate degradation IV 4-aminobutyrate degradation IV	glutamate decarboxylase	(Bouché and Fromm, 2004)	2 → 3
E106	glutamate degradation I	glutamate dehydrogenase	(Aubert et al., 2001)	9+29 ↔ 2
E107	glutamate biosynthesis V ammonia assimilation cycle II glutamine biosynthesis III	glutamate synthase	(Forde and Lea, 2007)	9+33 → 2x2

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E108	glutamate biosynthesis V ammonia assimilation cycle II glutamine biosynthesis III	glutamate synthase	(Forde and Lea, 2007)	9+33 → 2x2
E109	ammonia assimilation cycle II glutamine biosynthesis I glutamine biosynthesis III	glutamine synthetase	(Forde and Lea, 2007)	2+29 → 33
E110	glutamine biosynthesis III	pyruvate, orthophosphate dikinase	(Matsuoka, 1995)	20 → 34
E111	glycolysis IV (plant cytosol) glycolysis I (plastidic) Rubisco shunt	pyruvate kinase	(Andre et al., 2007)	34 → 20
E112	sucrose degradation to ethanol and lactate (anaerobic) pyruvate fermentation to ethanol II acetaldehyde biosynthesis I	pyruvate decarboxylase	(Ismond et al., 2003)	20 → 114
E113	1. alanine degradation III 1. alanine biosynthesis II 2. glycine biosynthesis 2. photorespiration	1. alanine aminotransferase 2. glyoxylate aminotransferase	(Igarashi et al., 2003)	1. 2+20 ↔ 9+21 2. 2+13 → 9+18
E114	ethanol degradation II (cytosol) ethanol degradation IV (peroxisomal)	acetate-CoA ligase	(Lin and Oliver, 2008)	38+93 → 92
E115	oxidative ethanol degradation III oxidative ethanol degradation I	aldehyde dehydrogenase	(Wei et al., 2007)	114 → 38
E116	cysteine biosynthesis I	O-acetylserine (thiol) lyase	(Wirtz and Hell, 2006)	91 → 38+113
E117	cysteine biosynthesis I	serine acetyltransferase	(Wirtz and Hell, 2006)	16+92 → 91+93
E118	Glycine biosynthesis	serine hydroxymethyltransferase	(Berg et al., 2002)	16+29 → 2x18
E119	glycine biosynthesis photorespiration	serine hydroxymethyltransferase	(Goyer et al., 2005)	2x18 → 16+29
E120	pyrimidine salvage pathway	pyrimidine specific 5'-nucleotidase	(Zrenner et al., 2006)	80 → 126 & 123 → 127
E121	glycine biosynthesis photorespiration	glyoxylate aminotransferase	(Liepman and Olsen, 2001)	13+16 → 15+18
E122	photorespiration	hydroxypyruvate reductase	(Cousins et al., 2008)	15 → 14
E123	photorespiration	glycerate kinase	(Liepman and Olsen, 2003)	14 → 40
E124	Not assigned			
E125	glyoxylate cycle	isocitrate lyase	(Cornah et al., 2004)	10 → 4+13
E126	glycolate and glyoxylate degradation II	glycolate oxidase	(Liepman and Olsen, 2003)	35 → 13

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E127	uridine-5'-phosphate biosynthesis	5-phosphoribosyl-1-pyrophosphate synthase	(Zrenner et al., 2006)	46 → 121
E128	photorespiration	phosphoglycolate phosphatase	(Linka and Weber, 2005)	64 → 35
E129	photorespiration Rubisco shunt/Calvin-Benson-Bassham cycle	ribulose 1,5-bisphosphate carboxlyase/oxygenase (Rubisco)	(Linka and Weber, 2005)	60 → 40+64 3x60 → 6x40
E130	Calvin-Benson-Bassham cycle Rubisco shunt	phosphoribulokinase	(Berg et al., 2002)	52 → 60
E131	pyrimidine salvage pathway pyrimidine ribonucleosides degradation II	uridine nucleosidase (+ ribokinase for ribose → 46)	(Zrenner et al., 2006)	126 → 24+46
E132	uracil degradation (reductive)	dihydrouracil dehydrogenase	(Zrenner et al., 2006)	24 → 124
E133	citrulline biosynthesis arginine biosynthesis I arginine biosynthesis II (acetyl cycle) urea cycle	argininosuccinate lyase	(Tischner et al., 2007)	27 ↔ 6+28
E134	citrulline biosynthesis arginine biosynthesis I arginine biosynthesis II (acetyl cycle) urea cycle	argininosuccinate synthetase	(Tischner et al., 2007)	23+94 → 27
E135	citrulline biosynthesis arginine biosynthesis I arginine biosynthesis II (acetyl cycle) urea cycle arginine degradation VI (arginase 2 pathway)	arginase	(Tischner et al., 2007)	28 → 1+25
E136	urea degradation II	urease	(Sirko and Brodzik, 2000)	25 → 2x29
E137	uracil degradation (reductive)	dihydropyrimidinase	(Zrenner et al., 2006)	124 → 125
E138	uracil degradation (reductive)	β-ureidopropionase	(Zrenner et al., 2006)	125 → 29 Beta-alanine not modeled, reaction only recycles bound NH3, which would be set free in the reaction to beta-alanine
E139	pyrimidine ribonucleotides interconversion	1. nucleoside diphosphate phosphatase 2. nucleoside triphosphate phosphatase	(Zrenner et al., 2006)	1. 115 → 80 & 128 → 123 2. 129 → 115 & 122 → 128
E140	asparagine biosynthesis I	asparaginesynthetase	(Lam et al., 1994)	23+33 → 2+26
E141	asparagine degradation I	asparaginase	(Bruneau et al., 2006)	26 → 23+29

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
E142	Not assigned			
E143	No pathway assigned	aldonolactonase	(ISHIKAWA and SHIGEOKA, 2008)	130 ↔ 78
E144	pyrimidine ribonucleotides interconversion	nucleoside diphosphate kinase	(Zrenner et al., 2006)	115 → 129 & 128 → 122
E145	pyrimidine ribonucleotides interconversion	apyrase	(Zrenner et al., 2006)	129 → 80 & 122 → 123
E146	pyrimidine ribonucleotides interconversion	CTP synthase	(Zrenner et al., 2006)	33+129 → 2+122
E147	pyrimidine ribonucleotides interconversion	cytidine deaminase	(Zrenner et al., 2006)	127 → 29+126
E148	1. pentose phosphate pathway (non-oxidative branch) 2. Calvin-Benson-Bassham cycle 3. Rubisco shunt	ribulose-phosphate 3-epimerase	(Berg et al., 2002; Dey and Harborne, 1997)	1. 52 → 45 2. 45 → 52 3. 45 → 52
E149	No pathway assigned	sorbitol dehydrogenase	(Nosarzewski et al., 2012)	62 → 63
E150	No pathway assigned	UDP-glucuronate 4-epimerase	(ISHIKAWA and SHIGEOKA, 2008)	85 → 131
E151	No pathway assigned	UDP-galacturonate pyrophosphatase	(ISHIKAWA and SHIGEOKA, 2008)	132 → 87
E152	No pathway assigned	D-galacturonate reductase	(ISHIKAWA and SHIGEOKA, 2008)	87 → 130
E153	phenylpropanoid biosynthesis	hydroxycinnamaldehyde dehydrogenase	(Fraser and Chapple, 2011)	134 → 48
E154	glycolysis	lipoate acetyltransferase	(Buchanan et al., 2000)	20+93 → 92
E155	Glyoxylate cycle	malate synthase	(Buchanan et al., 2000)	13+92 → 8+93
IN_18	glycine		(Jander et al., 2004; Joshi et al., 2006)	→ 18
IN_29	ammonia		(von Wirén et al., 2000; Morgan and Jackson, 1988)	→ 29
IN_63	D-fructose		(Sauer et al., 1990)	→ 63
IN_73	D-galactose		(Sauer et al., 1990)	→ 73
IN_92	acetyl-CoA		(Fulda et al., 2002)	→ 92
IN_93	CoA		(Kupke et al., 2001)	→ 93
IN_94	citrulline		(Ludwig, 1993)	→ 94
OUT_29	ammonia		(Morgan and Jackson, 1988)	29 →
OUT_30a	lignin		(Buchanan et al., 2000)	30a →
OUT_30b	lignin		(Buchanan et al., 2000)	30b →
OUT_50	Alpha-D-glucose 1-phosphate		(Wolucka and Montagu, 2007)	50 →
OUT_51	Beta-D-glucose		(AraCyc, 2012)	51 →
OUT_62	sorbitol		(Noiraud et al., 2001)	62 →
OUT_66	sucrose		(Sauer and Stolz, 1994)	66 →
OUT_71	stachyose		(AraCyc, 2012)	71 →
OUT_74	D-myo-inositol		(Reiter, 2008)	74 →

Table B.2: (continued)

Transition	Biochemical Pathway	Enzyme Name	Literature	Comments or reactions
OUT_78	L-galactono-1,4-lactone		(Davey et al., 1999)	78 →
OUT_82	UDP-glucose		(Dean and Delaney, 2008)	82 →
OUT_88	UDP-xylose		(Buchanan et al., 2000)	88 →
OUT_92	acetyl-CoA		External metabolite (compare Section 4.1.1.1)	92 →
OUT_93	CoA		External metabolite (compare Section 4.1.1.1)	93 →
OUT_94	citrulline		External metabolite (compare Section 4.1.1.1)	94 →
OUT_99	homogentisate		(Collakova and DellaPenna, 2001)	99 →
OUT_103	chorismate		(Strawn et al., 2006)	103 →
OUT_106	4-Hydroxy-Phenylpyrovate		(AraCyc, 2012)	106 →
OUT_113	cysteine		(Buchanan et al., 2000)	113 →
OUT_115	Uridine diphosphate (UDP)		(Buchanan et al., 2000)	115 →
OUT_122	CTP		(Zrenner et al., 2006)	122 → (RNA synthesis)
OUT_129	UTP		(Zrenner et al., 2006)	129 → (RNA synthesis)
R1	Starch degradation	Phosphoglucan, water dikinase	(Guy et al., 2008)	59 → 61
R2	Starch degradation	Glucan, water dikinase	(Fettke et al., 2009)	59 → 2x57
R3	Starch synthesis	Starch branching enzyme	(Kossmann and Lloyd, 2000)	81 & 88 → 59
R4	Starch degradation	disproportionating isozyme 2	(Fettke et al., 2009)	61 ↔ 57 & 89
R5	Starch synthesis	1. R5_1: Granule bound starch synthase 2. R5_2: Starch synthase	(Kossmann and Lloyd, 2000)	1. 58 → 86 2. 58 → 81
R6	Starch degradation	phosphorylase isozyme	(Fettke et al., 2009)	89 ↔ 50

Appendix C

Programm Outputs

C.1 Reduction of the *Arabidopsis thaliana* Network

Table C.1: Table of reductions. RT = reduction type (CTP = Common Transition Pair, ITP = Invariant Transition Pair, Parallel = reduction of parallel transitions, see Reddy et al. (1993)), reduced node = the name of the node after the reduction. The columns t_i / p_i, p_c / t_i,t_j, and t_j / p_j show t_i, p_c, and t_j, if the reduction type is 'CTP', and p_i, t_i and t_j, and p_j, if the reduction type is 'ITP' or 'Parallel'. Note that the column *reduced node* holds a transition, if the reduction type is 'CTP', and a place, if the reduction type is 'ITP'.

RT	t_i/p_i		$p_c/t_i,t_j$		t_j/p_j	reduced node
CTP	E151	→	87	→	E152	ctp(E151 + E152)
CTP	E23	→	88	→	E41	ctp(E23 + E41)
CTP	E30	→	84	→	E31	ctp(E30 + E31)
CTP	ctp(E30 + E31)	→	83	→	E32	ctp(ctp(E30 + E31) + E32)
CTP	R5_2	→	81	→	R3	ctp(R5_2 + R3)
CTP	R5_1	→	86	→	ctp(R5_2 + R3)	ctp(R5_1 + ctp(R5_2 + R3))
CTP	E5	→	53	→	E4	ctp(E5 + E4)
CTP	E129_2	→	64	→	E128	ctp(E129_2 + E128)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
CTP	E122	→	14	→	E123	ctp(E122 + E123)
CTP	E42	→	117	→	E43	ctp(E42 + E43)
CTP	E61_119	→	120	→	E61_80	ctp(E61_119 + E61_80)
CTP	E44	→	119	→	ctp(E61_119 + E61_80)	ctp(E44 + ctp(E61_119 + E61_80))
CTP	ctp(E42 + E43)	→	118	→	ctp(E44 + ctp(E61_119 + E61_80))	ctp(ctp(E42 + E43) + ctp(E44 + ctp(E61_119 + E61_80)))
CTP	E132	→	124	→	E137	ctp(E132 + E137)
CTP	ctp(E132 + E137)	→	125	→	E138	ctp(ctp(E132 + E137) + E138)
CTP	E98	→	11	→	E99	ctp(E98 + E99)
CTP	E140	→	26	→	E141	ctp(E140 + E141)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 23 → E140, weight: 1 and arc2: E141 → 23, weight: 1					
CTP	E135	→	1	→	E40	ctp(E135 + E40)
CTP	E117	→	91	→	E116	ctp(E117 + E116)
CTP	E112	→	114	→	E115	ctp(E112 + E115)
CTP	ctp(E117 + E116)	→	113	→	OUT_113	ctp(ctp(E117 + E116) + OUT_113)
CTP	E64	→	36	→	E65	ctp(E64 + E65)
CTP	ctp(E135 + E40)	→	25	→	E136	ctp(ctp(E135 + E40) + E136)
CTP	E69	→	103	→	E70	ctp(E69 + E70)
CTP	E67	→	96	→	E68	ctp(E67 + E68)
CTP	ctp(E67 + E68)	→	97	→	ctp(E69 + E70)	ctp(ctp(E67 + E68) + ctp(E69 + E70))
CTP	E83	→	7	→	E84	ctp(E83 + E84)
CTP	E82_90	→	90	→	E81_90	ctp(E82_90 + E81_90)
CTP	E82_111	→	111	→	E81_111	ctp(E82_111 + E81_111)
CTP	E81_100	→	100	→	ctp(E82_90 + E81_90)	ctp(E81_100 + ctp(E82_90 + E81_90))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E81_100 → 93, weight: 1 and arc2: 93 → ctp(E82_90+E81_90), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 102 → E81_100, weight: 1 and arc2: ctp(E82_90+E81_90) → 102, weight: 1					

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
CTP	E81_104	→	104	→	ctp(E82_111 + E81_111)	ctp(E81_104 + ctp(E82_111 + E81_111))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E81_104 → 93, weight: 1 and arc2: 93 → ctp(E82_111+E81_111), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 101 → E81_104, weight: 1 and arc2: ctp(E82_111+E81_111) → 101, weight: 1					
CTP	E79	→	112	→	E80	ctp(E79 + E80)
CTP	E36_f	→	75	→	E36_b	ctp(E36_f + E36_b)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 73 → E36_f, weight: 1 and arc2: E36_b → 73, weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 51 → E36_f, weight: 1 and arc2: E36_b → 51, weight: 1					
	Note: after CTP-reduction, the transition ctp(E36_f+E36_b) has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
CTP	E95	→	55	→	OUT_55_sinapate − esther − synthesis	ctp(E95 + OUT_55_sinapate − esther − synthesis)
CTP	E86_31	→	31	→	E87_48	ctp(E86_31 + E87_48)
CTP	E88_19	→	30a	→	OUT_30a	ctp(E88_19 + OUT_30a)
CTP	ctp(E83 + E84)	→	17	→	E85	ctp(ctp(E83 + E84) + E85)
CTP	E86_133	→	133	→	E87_134	ctp(E86_133 + E87_134)
CTP	E88_48	→	30b	→	OUT_30b	ctp(E88_48 + OUT_30b)
CTP	E74	→	99	→	OUT_99	ctp(E74 + OUT_99)
ITP	130	→	E143_f	→	78	itp(130 + 78)
	130	←	E143_b	←	78	itp(130 + 78)
ITP	82	→	E20_f	→	77	itp(82 + 77)
	82	←	E20_b	←	77	itp(82 + 77)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 82 → E24_f, weight: 1 and arc2: E24_f → 77, weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E24_b → 82, weight: 1 and arc2: 77 → E24_b, weight: 1					
ITP	76	→	E24_f	→	50	itp(76 + 50)
	76	←	E24_b	←	50	itp(76 + 50)
ITP	69	→	E34_b	→	56	itp(69 + 56)
	69	←	E34_f	←	56	itp(69 + 56)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
ITP	49	$\rightarrow$	spo_b	$\rightarrow$	itp(69 + 56)	itp(49 + itp(69 + 56))
	49	$\leftarrow$	spo_f	$\leftarrow$	itp(69 + 56)	itp(49 + itp(69 + 56))
ITP	itp(49 + itp(69 + 56))	$\rightarrow$	E8_f	$\rightarrow$	44	itp(itp(49 + itp(69 + 56)) + 44)
	itp(49 + itp(69 + 56))	$\leftarrow$	E8_b	$\leftarrow$	44	itp(itp(49 + itp(69 + 56)) + 44)
ITP	itp(itp(49 + itp(69 + 56)) + 44)	$\rightarrow$	E9_f	$\rightarrow$	itp(76 + 50)	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))
	itp(itp(49 + itp(69 + 56)) + 44)	$\leftarrow$	E9_b	$\leftarrow$	itp(76 + 50)	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))
ITP	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))	$\rightarrow$	R6_f	$\rightarrow$	89	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)
	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))	$\leftarrow$	R6_b	$\leftarrow$	89	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)
ITP	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)	$\rightarrow$	E16	$\rightarrow$	43	itp(itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89) + 43)
	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)	$\leftarrow$	E15	$\leftarrow$	43	itp(itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89) + 43)
ITP	68	$\rightarrow$	E13_f	$\rightarrow$	42	itp(68 + 42)
	68	$\leftarrow$	E13_b	$\leftarrow$	42	itp(68 + 42)
ITP	itp(68 + 42)	$\rightarrow$	E91_f	$\rightarrow$	41	itp(itp(68 + 42) + 41)
	itp(68 + 42)	$\leftarrow$	E91_b	$\leftarrow$	41	itp(itp(68 + 42) + 41)
ITP	itp(itp(68 + 42) + 41)	$\rightarrow$	E90_f	$\rightarrow$	40	itp(itp(itp(68 + 42) + 41) + 40)
	itp(itp(68 + 42) + 41)	$\leftarrow$	E90_b	$\leftarrow$	40	itp(itp(itp(68 + 42) + 41) + 40)
ITP	45	$\rightarrow$	E148_f	$\rightarrow$	52	itp(45 + 52)
	45	$\leftarrow$	E148_b	$\leftarrow$	52	itp(45 + 52)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
ITP	$itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)$	$\rightarrow$	E6_f	$\rightarrow$	54	$itp(itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)+54)$
	$itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)$	$\leftarrow$	E6_b	$\leftarrow$	54	$itp(itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)+54)$
ITP	$itp(45+52)$	$\rightarrow$	E3_b	$\rightarrow$	46	$itp(itp(45+52)+46)$
	$itp(45+52)$	$\leftarrow$	E3_f	$\leftarrow$	46	$itp(itp(45+52)+46)$
ITP	$itp(itp(itp(68+42)+41)+40)$	$\rightarrow$	E89_f	$\rightarrow$	39	$itp(itp(itp(itp(68+42)+41)+40)+39)$
	$itp(itp(itp(68+42)+41)+40)$	$\leftarrow$	E89_b	$\leftarrow$	39	$itp(itp(itp(itp(68+42)+41)+40)+39)$
ITP	129	$\rightarrow$	E139_u_ttod	$\rightarrow$	115	$itp(129+115)$
	129	$\leftarrow$	E144_u	$\leftarrow$	115	$itp(129+115)$
ITP	$itp(129+115)$	$\rightarrow$	E139_u_dtom	$\rightarrow$	80	$itp(itp(129+115)+80)$
	$itp(129+115)$	$\leftarrow$	E62_u	$\leftarrow$	80	$itp(itp(129+115)+80)$
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: itp(129+115) $\rightarrow$ E145_u, weight: 1 and arc2: E145_u $\rightarrow$ 80, weight: 1					
	Note: after ITP-reduction, the transition E145_u has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
ITP	126	$\rightarrow$	E35_u	$\rightarrow$	$itp(itp(129+115)+80)$	$itp(126+itp(itp(129+115)+80))$
	126	$\leftarrow$	E120_u	$\leftarrow$	$itp(itp(129+115)+80)$	$itp(126+itp(itp(129+115)+80))$
ITP	128	$\rightarrow$	E139_c_dtom	$\rightarrow$	123	$itp(128+123)$
	128	$\leftarrow$	E62_c	$\leftarrow$	123	$itp(128+123)$
ITP	122	$\rightarrow$	E139_c_ttod	$\rightarrow$	$itp(128+123)$	$itp(122+itp(128+123))$
	122	$\leftarrow$	E144_c	$\leftarrow$	$itp(128+123)$	$itp(122+itp(128+123))$
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 122 $\rightarrow$ E145_c, weight: 1 and arc2: E145_c $\rightarrow$ itp(128+123), weight: 1					
	Note: after ITP-reduction, the transition E145_c has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
ITP	$itp(122+itp(128+123))$	$\rightarrow$	E120_c	$\rightarrow$	127	$itp(itp(122+itp(128+123))+127)$
	$itp(122+itp(128+123))$	$\leftarrow$	E35_c	$\leftarrow$	127	$itp(itp(122+itp(128+123))+127)$

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i,t_j$		t_j/p_j	reduced node
ITP	12	$\rightarrow$	E94_b	$\rightarrow$	8	itp(12+8)
	12	$\leftarrow$	E94_f	$\leftarrow$	8	itp(12+8)
ITP	itp(12+8)	$\rightarrow$	E93	$\rightarrow$	34	itp(itp(12+8)+34)
	itp(12+8)	$\leftarrow$	E92	$\leftarrow$	34	itp(itp(12+8)+34)
ITP	itp(itp(12+8)+34)	$\rightarrow$	E111	$\rightarrow$	20	itp(itp(itp(12+8)+34)+20)
	itp(itp(12+8)+34)	$\leftarrow$	E110	$\leftarrow$	20	itp(itp(itp(12+8)+34)+20)
ITP	6	$\rightarrow$	E97_b	$\rightarrow$	4	itp(6+4)
	6	$\leftarrow$	E97_f	$\leftarrow$	4	itp(6+4)
ITP	itp(itp(itp(12+8)+34)+20)	$\rightarrow$	E96_b	$\rightarrow$	itp(6+4)	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))
	itp(itp(itp(12+8)+34)+20)	$\leftarrow$	E96_f	$\leftarrow$	itp(6+4)	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))
ITP	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))	$\rightarrow$	E63_f	$\rightarrow$	itp(itp(itp(itp(68+42)+41)+40)+39)	itp(itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))+itp(itp(itp(itp(68+42)+41)+40)+39))
	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))	$\leftarrow$	E63_b	$\leftarrow$	itp(itp(itp(itp(68+42)+41)+40)+39)	itp(itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))+itp(itp(itp(itp(68+42)+41)+40)+39))
ITP	102	$\rightarrow$	E66_101_b	$\rightarrow$	101	itp(102+101)
	102	$\leftarrow$	E66_101_f	$\leftarrow$	101	itp(102+101)
ITP	itp(102+101)	$\rightarrow$	E67_95_b	$\rightarrow$	95	itp(itp(102+101)+95)
	itp(102+101)	$\leftarrow$	E67_95_f	$\leftarrow$	95	itp(itp(102+101)+95)
ITP	37	$\rightarrow$	E66_95_f	$\rightarrow$	itp(itp(102+101)+95)	itp(37+itp(itp(102+101)+95))
	37	$\leftarrow$	E67_101_b	$\leftarrow$	itp(itp(102+101)+95)	itp(37+itp(itp(102+101)+95))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 37 $\rightarrow$ E67_101_f, weight: 1 and arc2: E67_101_f $\rightarrow$ itp(itp(102+101)+95), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E66_95_b $\rightarrow$ 37, weight: 1 and arc2: itp(itp(102+101)+95) $\rightarrow$ E66_95_b, weight: 1					
	Note: after ITP-reduction, the transition E66_95_b has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
	Note: after ITP-reduction, the transition E67_101_f has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
CTP	ctp(E64 + E65)	→	itp(37 + itp(itp(102 + 101) + 95))	→	ctp(ctp(E67 + E68) + ctp(E69 + E70))	ctp(ctp(E64 + E65) + ctp(ctp(E67 + E68) + ctp(E69 + E70)))
Parallel	47	→	ctp(E81_104 + ctp(E82_111 + E81_111))	→	108	parallel(ctp(E81_104 + ctp(E82_111 + E81_111))ctp(E81_100 + ctp(E82_90 + E81_90)))
	47	→	ctp(E81_100 + ctp(E82_90 + E81_90))	→	108	parallel(ctp(E81_104 + ctp(E82_111 + E81_111))ctp(E81_100 + ctp(E82_90 + E81_90)))
CTP	parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90)))	→	108	→	ctp(ctp(E83 + E84) + E85)	ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85))
CTP	ctp(E79 + E80)	→	47	→	ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85))	ctp(ctp(E79 + E80) + ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85)))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 93 → ctp(E79+E80), weight: 1 and arc2: ctp(parallel(ctp(E81_104+ctp(E82_111+E81_111))+ ctp(E81_100+ctp(E82_90+E81_90)))+ctp(ctp(E83+E84)+E85)) → 93, weight: 1					
CTP	E151	→	87	→	E152	ctp(E151 + E152)
CTP	E23	→	88	→	E41	ctp(E23 + E41)
CTP	E30	→	84	→	E31	ctp(E30 + E31)
CTP	ctp(E30 + E31)	→	83	→	E32	ctp(ctp(E30 + E31) + E32)
CTP	R5_2	→	81	→	R3	ctp(R5_2 + R3)
CTP	R5_1	→	86	→	ctp(R5_2 + R3)	ctp(R5_1 + ctp(R5_2 + R3))
CTP	E5	→	53	→	E4	ctp(E5 + E4)
CTP	E129_2	→	64	→	E128	ctp(E129_2 + E128)
CTP	E122	→	14	→	E123	ctp(E122 + E123)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
CTP	E42	→	117	→	E43	ctp(E42 + E43)
CTP	E61_119	→	120	→	E61_80	ctp(E61_119 + E61_80)
CTP	E44	→	119	→	ctp(E61_119 + E61_80)	ctp(E44 + ctp(E61_119 + E61_80))
CTP	ctp(E42 + E43)	→	118	→	ctp(E44 + ctp(E61_119 + E61_80))	ctp(ctp(E42 + E43) + ctp(E44 + ctp(E61_119 + E61_80)))
CTP	E132	→	124	→	E137	ctp(E132 + E137)
CTP	ctp(E132 + E137)	→	125	→	E138	ctp(ctp(E132 + E137) + E138)
CTP	E98	→	11	→	E99	ctp(E98 + E99)
CTP	E140	→	26	→	E141	ctp(E140 + E141)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 23 → E140, weight: 1 and arc2: E141 → 23, weight: 1					
CTP	E135	→	1	→	E40	ctp(E135 + E40)
CTP	E117	→	91	→	E116	ctp(E117 + E116)
CTP	E112	→	114	→	E115	ctp(E112 + E115)
CTP	ctp(E117 + E116)	→	113	→	OUT_113	ctp(ctp(E117 + E116) + OUT_113)
CTP	E64	→	36	→	E65	ctp(E64 + E65)
CTP	ctp(E135 + E40)	→	25	→	E136	ctp(ctp(E135 + E40) + E136)
CTP	E69	→	103	→	E70	ctp(E69 + E70)
CTP	E67	→	96	→	E68	ctp(E67 + E68)
CTP	ctp(E67 + E68)	→	97	→	ctp(E69 + E70)	ctp(ctp(E67 + E68) + ctp(E69 + E70))
CTP	E83	→	7	→	E84	ctp(E83 + E84)
CTP	E82_90	→	90	→	E81_90	ctp(E82_90 + E81_90)
CTP	E82_111	→	111	→	E81_111	ctp(E82_111 + E81_111)
CTP	E81_100	→	100	→	ctp(E82_90 + E81_90)	ctp(E81_100 + ctp(E82_90 + E81_90))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E81_100 → 93, weight: 1 and arc2: 93 → ctp(E82_90+E81_90), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 102 → E81_100, weight: 1 and arc2: ctp(E82_90+E81_90) → 102, weight: 1					
CTP	E81_104	→	104	→	ctp(E82_111 + E81_111)	ctp(E81_104 + ctp(E82_111 + E81_111))

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E81_104 → 93, weight: 1 and arc2: 93 → ctp(E82_111+E81_111), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 101 → E81_104, weight: 1 and arc2: ctp(E82_111+E81_111) → 101, weight: 1					
CTP	E79	→	112	→	E80	ctp(E79 + E80)
CTP	E36_f	→	75	→	E36_b	ctp(E36_f + E36_b)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 73 → E36_f, weight: 1 and arc2: E36_b → 73, weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 51 → E36_f, weight: 1 and arc2: E36_b → 51, weight: 1					
	Note: after CTP-reduction, the transition ctp(E36_f+E36_b) has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
CTP	E95	→	55	→	OUT_55_sinapate − esther − synthesis	ctp(E95 + OUT_55_sinapate − esther − synthesis)
CTP	E86_31	→	31	→	E87_48	ctp(E86_31 + E87_48)
CTP	E88_19	→	30a	→	OUT_30a	ctp(E88_19 + OUT_30a)
CTP	ctp(E83 + E84)	→	17	→	E85	ctp(ctp(E83 + E84) + E85)
CTP	E86_133	→	133	→	E87_134	ctp(E86_133 + E87_134)
CTP	E88_48	→	30b	→	OUT_30b	ctp(E88_48 + OUT_30b)
CTP	E74	→	99	→	OUT_99	ctp(E74 + OUT_99)
ITP	130	→	E143_f	→	78	itp(130 + 78)
	130	←	E143_b	←	78	itp(130 + 78)
ITP	82	→	E20_f	→	77	itp(82 + 77)
	82	←	E20_b	←	77	itp(82 + 77)
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 82 → E24_f, weight: 1 and arc2: E24_f → 77, weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E24_b → 82, weight: 1 and arc2: 77 → E24_b, weight: 1					
ITP	76	→	E24_f	→	50	itp(76 + 50)
	76	←	E24_b	←	50	itp(76 + 50)
ITP	69	→	E34_b	→	56	itp(69 + 56)
	69	←	E34_f	←	56	itp(69 + 56)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
ITP	49	$\rightarrow$	spo_b	$\rightarrow$	itp(69 + 56)	itp(49 + itp(69 + 56))
	49	$\leftarrow$	spo_f	$\leftarrow$	itp(69 + 56)	itp(49 + itp(69 + 56))
ITP	itp(49 + itp(69 + 56))	$\rightarrow$	E8_f	$\rightarrow$	44	itp(itp(49 + itp(69 + 56)) + 44)
	itp(49 + itp(69 + 56))	$\leftarrow$	E8_b	$\leftarrow$	44	itp(itp(49 + itp(69 + 56)) + 44)
ITP	itp(itp(49 + itp(69 + 56)) + 44)	$\rightarrow$	E9_f	$\rightarrow$	itp(76 + 50)	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))
	itp(itp(49 + itp(69 + 56)) + 44)	$\leftarrow$	E9_b	$\leftarrow$	itp(76 + 50)	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))
ITP	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))	$\rightarrow$	R6_f	$\rightarrow$	89	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)
	itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50))	$\leftarrow$	R6_b	$\leftarrow$	89	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)
ITP	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)	$\rightarrow$	E16	$\rightarrow$	43	itp(itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89) + 43)
	itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89)	$\leftarrow$	E15	$\leftarrow$	43	itp(itp(itp(itp(itp(49 + itp(69 + 56)) + 44) + itp(76 + 50)) + 89) + 43)
ITP	68	$\rightarrow$	E13_f	$\rightarrow$	42	itp(68 + 42)
	68	$\leftarrow$	E13_b	$\leftarrow$	42	itp(68 + 42)
ITP	itp(68 + 42)	$\rightarrow$	E91_f	$\rightarrow$	41	itp(itp(68 + 42) + 41)
	itp(68 + 42)	$\leftarrow$	E91_b	$\leftarrow$	41	itp(itp(68 + 42) + 41)
ITP	itp(itp(68 + 42) + 41)	$\rightarrow$	E90_f	$\rightarrow$	40	itp(itp(itp(68 + 42) + 41) + 40)
	itp(itp(68 + 42) + 41)	$\leftarrow$	E90_b	$\leftarrow$	40	itp(itp(itp(68 + 42) + 41) + 40)
ITP	45	$\rightarrow$	E148_f	$\rightarrow$	52	itp(45 + 52)
	45	$\leftarrow$	E148_b	$\leftarrow$	52	itp(45 + 52)

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
ITP	$itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)$	$\rightarrow$	E6_f	$\rightarrow$	54	$itp(itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)+54)$
	$itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)$	$\leftarrow$	E6_b	$\leftarrow$	54	$itp(itp(itp(itp(itp(itp(49+itp(69+56))+44)+itp(76+50))+89)+43)+54)$
ITP	$itp(45+52)$	$\rightarrow$	E3_b	$\rightarrow$	46	$itp(itp(45+52)+46)$
	$itp(45+52)$	$\leftarrow$	E3_f	$\leftarrow$	46	$itp(itp(45+52)+46)$
ITP	$itp(itp(itp(68+42)+41)+40)$	$\rightarrow$	E89_f	$\rightarrow$	39	$itp(itp(itp(itp(68+42)+41)+40)+39)$
	$itp(itp(itp(68+42)+41)+40)$	$\leftarrow$	E89_b	$\leftarrow$	39	$itp(itp(itp(itp(68+42)+41)+40)+39)$
ITP	129	$\rightarrow$	E139_u_ttod	$\rightarrow$	115	$itp(129+115)$
	129	$\leftarrow$	E144_u	$\leftarrow$	115	$itp(129+115)$
ITP	$itp(129+115)$	$\rightarrow$	E139_u_dtom	$\rightarrow$	80	$itp(itp(129+115)+80)$
	$itp(129+115)$	$\leftarrow$	E62_u	$\leftarrow$	80	$itp(itp(129+115)+80)$
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: itp(129+115) $\rightarrow$ E145_u, weight: 1 and arc2: E145_u $\rightarrow$ 80, weight: 1					
	Note: after ITP-reduction, the transition E145_u has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
ITP	126	$\rightarrow$	E35_u	$\rightarrow$	$itp(itp(129+115)+80)$	$itp(126+itp(itp(129+115)+80))$
	126	$\leftarrow$	E120_u	$\leftarrow$	$itp(itp(129+115)+80)$	$itp(126+itp(itp(129+115)+80))$
ITP	128	$\rightarrow$	E139_c_dtom	$\rightarrow$	123	$itp(128+123)$
	128	$\leftarrow$	E62_c	$\leftarrow$	123	$itp(128+123)$
ITP	122	$\rightarrow$	E139_c_ttod	$\rightarrow$	$itp(128+123)$	$itp(122+itp(128+123))$
	122	$\leftarrow$	E144_c	$\leftarrow$	$itp(128+123)$	$itp(122+itp(128+123))$
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 122 $\rightarrow$ E145_c, weight: 1 and arc2: E145_c $\rightarrow$ itp(128+123), weight: 1					
	Note: after ITP-reduction, the transition E145_c has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
ITP	$itp(122+itp(128+123))$	$\rightarrow$	E120_c	$\rightarrow$	127	$itp(itp(122+itp(128+123))+127)$
	$itp(122+itp(128+123))$	$\leftarrow$	E35_c	$\leftarrow$	127	$itp(itp(122+itp(128+123))+127)$

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i,t_j$		t_j/p_j	reduced node
ITP	12	$\rightarrow$	E94_b	$\rightarrow$	8	itp(12+8)
	12	$\leftarrow$	E94_f	$\leftarrow$	8	itp(12+8)
ITP	itp(12+8)	$\rightarrow$	E93	$\rightarrow$	34	itp(itp(12+8)+34)
	itp(12+8)	$\leftarrow$	E92	$\leftarrow$	34	itp(itp(12+8)+34)
ITP	itp(itp(12+8)+34)	$\rightarrow$	E111	$\rightarrow$	20	itp(itp(itp(12+8)+34)+20)
	itp(itp(12+8)+34)	$\leftarrow$	E110	$\leftarrow$	20	itp(itp(itp(12+8)+34)+20)
ITP	6	$\rightarrow$	E97_b	$\rightarrow$	4	itp(6+4)
	6	$\leftarrow$	E97_f	$\leftarrow$	4	itp(6+4)
ITP	itp(itp(itp(12+8)+34)+20)	$\rightarrow$	E96_b	$\rightarrow$	itp(6+4)	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))
	itp(itp(itp(12+8)+34)+20)	$\leftarrow$	E96_f	$\leftarrow$	itp(6+4)	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))
ITP	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))	$\rightarrow$	E63_f	$\rightarrow$	itp(itp(itp(itp(68+42)+41)+40)+39)	itp(itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))+itp(itp(itp(itp(68+42)+41)+40)+39))
	itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))	$\leftarrow$	E63_b	$\leftarrow$	itp(itp(itp(itp(68+42)+41)+40)+39)	itp(itp(itp(itp(itp(12+8)+34)+20)+itp(6+4))+itp(itp(itp(itp(68+42)+41)+40)+39))
ITP	102	$\rightarrow$	E66_101_b	$\rightarrow$	101	itp(102+101)
	102	$\leftarrow$	E66_101_f	$\leftarrow$	101	itp(102+101)
ITP	itp(102+101)	$\rightarrow$	E67_95_b	$\rightarrow$	95	itp(itp(102+101)+95)
	itp(102+101)	$\leftarrow$	E67_95_f	$\leftarrow$	95	itp(itp(102+101)+95)
ITP	37	$\rightarrow$	E66_95_f	$\rightarrow$	itp(itp(102+101)+95)	itp(37+itp(itp(102+101)+95))
	37	$\leftarrow$	E67_101_b	$\leftarrow$	itp(itp(102+101)+95)	itp(37+itp(itp(102+101)+95))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 37 $\rightarrow$ E67_101_f, weight: 1 and arc2: E67_101_f $\rightarrow$ itp(itp(102+101)+95), weight: 1					
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: E66_95_b $\rightarrow$ 37, weight: 1 and arc2: itp(itp(102+101)+95) $\rightarrow$ E66_95_b, weight: 1					
	Note: after ITP-reduction, the transition E66_95_b has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					

Table C.1: (continued)

RT	t_i/p_i		$p_c/t_i, t_j$		t_j/p_j	reduced node
	Note: after ITP-reduction, the transition E67_101_f has no inputs and outputs left, because of removed arcs of weight 0, and will be removed					
CTP	ctp(E64 + E65)	→	itp(37 + itp(itp(102 + 101) + 95))	→	ctp(ctp(E67 + E68) + ctp(E69 + E70))	ctp(ctp(E64 + E65) + ctp(ctp(E67 + E68) + ctp(E69 + E70)))
Parallel	47	→	ctp(E81_104 + ctp(E82_111 + E81_111))	→	108	parallel(ctp(E81_104 + ctp(E82_111 + E81_111))ctp(E81_100 + ctp(E82_90 + E81_90)))
	47	→	ctp(E81_100 + ctp(E82_90 + E81_90))	→	108	parallel(ctp(E81_104 + ctp(E82_111 + E81_111))ctp(E81_100 + ctp(E82_90 + E81_90)))
CTP	parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90)))	→	108	→	ctp(ctp(E83 + E84) + E85)	ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85))
CTP	ctp(E79 + E80)	→	47	→	ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85))	ctp(ctp(E79 + E80) + ctp(parallel(ctp(E81_104 + ctp(E82_111 + E81_111)) + ctp(E81_100 + ctp(E82_90 + E81_90))) + ctp(ctp(E83 + E84) + E85)))
	Note: the following arcs cancel each other to a weight of zero after reduction, and will be removed: arc1: 93 → ctp(E79+E80), weight: 1 and arc2: ctp(parallel(ctp(E81_104+ctp(E82_111+E81_111))+ ctp(E81_100+ctp(E82_90+E81_90)))+ctp(ctp(E83+E84)+E85)) → 93, weight: 1					

C.2 Static Conflicts

Table C.2: Table of static conflicts of the network. *Conflicts* are the transitions which are in a static conflict with *Transition*. The complete number of conflicts is 630. Note that each conflict is counted twice, because if a transition A is in conflict with a transition B, also Transition B is in conflict with Transition A.

Transition	Conflicts
E86_31	E88_19, E86_133
E88_19	E86_31, E86_133
E95	E153
OUT_93	E101, E102_b, E114, E154
OUT_92	E98, E117, E155
OUT_107	E75_f
E111	E92, E63_f, E68, E64
E100	E125
E93	E94_b, E98, E2_f
E98	E94_b, E93, E117, E2_f, OUT_92, E155
E101	E106_f, E107, E108, E102_b, E113_f, E114, E2_b, E75_f, OUT_93, E71_b, E154
E103	E104_b
E92	E111, E63_f, E68, E64
E97_b	E96_f, E133_b
E102_b	E101, E97_f, E114, OUT_93, E154
E151	E26_b
E34_b	E33
E34_f	E30, spo_f, E9_f
OUT_71	E45_73
OUT_66	E28, E37, E18_f
E45_66	E29
E33	E34_b
OUT_51	E10_49, E36_f
OUT_94	E134
E149	OUT_62
E24_f	E19, E20_f, E17_77, E30, OUT_82, E18_b
E17_77	E24_f, E139_u_dtom, E144_u, E18_f
E133_b	E97_b, E96_f, E135
E143_b	OUT_78
E106_b	E105, E109, E113, E113_b, E2_f, E71_f, E75_b
E106_f	E107, E108, E109, E101, E118, E113_f, E141, E2_b, E75_f, OUT_29, E71_b
E107	E38, E146, E106_f, E108, E101, E113_f, E140, E2_b, E75_f, E71_b
E108	E38, E146, E106_f, E107, E101, E113_f, E140, E2_b, E75_f, E71_b
E109	E105, E106_b, E106_f, E113, E118, E113_b, E2_f, E141, E71_f, OUT_29, E75_b
E104_b	E103, E113_f
E104_f	E112, E110, E113_b, E154
E94_b	E98, E93, E2_f
E94_f	E96_b
E36_f	E25, OUT_51, E10_49, E22
E54_b	E8_b, E16, E1_44_b, E1_44_f, E11_b, E64
E14_f	E15

Table C.2: (continued)

Transition	Conflicts
E15	E14_f
E1_44_f	E8_b, E16, E54_f, E14_b, E54_b, E13_b, E91_f, E1_46_f
E14_b	E54_f, E1_44_f, E13_f, E11_b, E13_b, E91_f, E1_46_f
E13_f	E14_b, E11_b
R1	R2
E1_44_b	E54_b, E11_b, E148_f, E1_46_b, E64
E54_f	E1_44_f, E14_b, E13_b, E91_f, E1_46_f
OUT_105	E71_f, E72_106
E68	E111, E92, E63_f, E64
E72_106	E71_f, OUT_105
E81_100	E66_101_b, E67, E67_95_b, E81_104
E75_f	E106_f, E107, E108, E101, E113_f, E2_b, OUT_107, E71_b
E71_f	E105, E106_b, E109, E113, E113_b, E2_f, E72_106, OUT_105, E75_b
E132	E35
E127	E1_46_b, E3_f
E61_119	E35
E145_c	E139_c_ttod, OUT_122_rna
E144_c	E139_c_dtom
E120_c	E62_c
E38	E146, E107, E108, E140
E64	E1_44_b, E54_b, E11_b, E111, E92, E63_f, E68
E122	E21
E21	E122
E105	E106_b, E109, E113, E113_b, E2_f, E71_f, E75_b
E35_c	E147
E129_2	E129_1
E129_1	E129_2
E81_104	E66_101_f, E67_101_b, E81_100
E26_b	E139_u_ttod, OUT_129_rna, E17_82, E146, E145_u, E151
E19	E24_f, E20_f, E30, OUT_82, E18_b
E20_b	E24_b, E27
E62_c	E120_c
E145_u	E139_u_ttod, OUT_129_rna, E17_82, E146, E26_b
E86_133	E88_19, E86_31
E153	E95
E18_f	E17_77, E28, E37, OUT_66, E139_u_dtom, E144_u
E35_u	E131
E62_u	E120_u
E18_b	E19, E24_f, E20_f, E30, OUT_82, E12
E146	E38, E139_u_ttod, OUT_129_rna, E17_82, E145_u, E107, E108, E140, E26_b
E139_c_ttod	E145_c, OUT_122_rna
E120_u	E62_u
E139_c_dtom	E144_c
E10_56	R4_f
E8_b	E16, E1_44_f, E54_b
E8_f	E39, spo_b, E6_f

Table C.2: (continued)

Transition	Conflicts
spo_b	E39, E8_f, E6_f
E39	spo_b, E8_f, E6_f
E12	E18_b
E9_b	E24_b, R6_f, E46, OUT_50, E17_82
E9_f	E30, E34_f, spo_f
spo_f	E30, E34_f, E9_f
E10_49	OUT_51, E36_f
OUT_29	E106_f, E109, E118, E141
E66_101_f	E67_101_b, E81_104
E7	E11_f
E11_f	E7
E11_b	E1_44_b, E13_f, E14_b, E54_b, E64
E13_b	E54_f, E1_44_f, E14_b, E91_f, E1_46_f
E91_f	E54_f, E1_44_f, E14_b, E13_b, E1_46_f
E91_b	E90_f
E90_f	E91_b
E148_f	E1_44_b, E1_46_b
E148_b	E3_b, E130
E67_95_b	E66_101_b, E67, E81_100
E67	E66_101_b, E67_95_b, E81_100
E66_101_b	E67, E67_95_b, E81_100
E2_b	E42, E106_f, E107, E108, E101, E113_f, E134, E140, E75_f, E71_b
E140	E38, E42, E146, E107, E108, E134, E2_b
E141	E106_f, E109, E118, OUT_29
E66_95_f	E67_101_f
OUT_122_rna	E139_c_ttod, E145_c
E23	E150
E67_95_f	E66_95_b
E66_95_b	E67_95_f
E20_f	E19, E24_f, E30, OUT_82, E18_b
E67_101_f	E66_95_f
E67_101_b	E66_101_f, E81_104
E17_82	E24_b, E9_b, R6_f, E46, OUT_50, E139_u_ttod, OUT_129_rna, E146, E145_u, E26_b
E144_u	E17_77, E139_u_dtom, E18_f
E114	E101, E102_b, OUT_93, E154
E35	E61_119, E132
E135	E133_b
E2_f	E105, E106_b, E109, E94_b, E98, E93, E113, E113_b, E71_f, E75_b
E42	E40, E134, E140, E2_b
E139_u_ttod	OUT_129_rna, E17_82, E146, E145_u, E26_b
OUT_62	E149
E139_u_dtom	E17_77, E144_u, E18_f
OUT_129_rna	E139_u_ttod, E17_82, E146, E145_u, E26_b
E25	E22, E36_f
OUT_78	E143_b

Table C.2: (continued)

Transition	Conflicts
OUT_82	E19, E24_f, E20_f, E30, E18_b
E27	E24_b, E20_b, OUT_74
E29	E28, E45_66
E28	E29, E37, OUT_66, E18_f
E45_73	OUT_71
E37	E28, OUT_66, E18_f
E147	E35_c
E131	E35_u
E110	E104_f, E112, E113_b, E154
E112	E104_f, E110, E113_b, E154
E63_b	E89_b
E63_f	E111, E92, E68, E64
E113_b	E105, E106_b, E109, E104_f, E113, E112, E110, E2_f, E71_f, E75_b, E154
E113_f	E106_f, E107, E108, E104_b, E101, E2_b, E75_f, E71_b
E121	E113, E118, E117, E155
E125	E100
E134	E42, OUT_94, E140, E2_b
E40	E42
E75_b	E105, E106_b, E109, E113, E113_b, E2_f, E71_f, E74
E154	E104_f, E101, E102_b, E112, E110, E113_b, E114, OUT_93
E22	E25, E36_f
E71_b	E106_f, E107, E108, E101, E113_f, E2_b, E75_f, E73, E72_107
E74	E75_b
E155	E98, E113, E117, E121, OUT_92
E150	E23
E6_b	E5
OUT_74	E27
E16	E8_b, E1_44_f, E54_b
R2	R1
R4_f	E10_56, R6_b
R5_1	R5_2
R5_2	R5_1
OUT_50	E24_b, E9_b, R6_f, E46, E17_82
R6_b	R4_f
R6_f	E24_b, E9_b, E46, OUT_50, E17_82
E46	E24_b, E9_b, R6_f, OUT_50, E17_82
E24_b	E20_b, E27, E9_b, R6_f, E46, OUT_50, E17_82
E30	E19, E24_f, E20_f, OUT_82, E34_f, spo_f, E9_f, E18_b
E118	E106_f, E109, E117, E121, E141, OUT_29
E117	E98, E118, E121, OUT_92, E155
E72_107	E73, E71_b
E73	E72_107, E71_b
E96_f	E97_b, E133_b
E113	E105, E106_b, E109, E113_b, E121, E2_f, E71_f, E75_b, E155
E97_f	E102_b
E96_b	E94_f

Table C.2: (continued)

Transition	Conflicts
E78	E77
E77	E78
E89_f	E90_b
E90_b	E89_f
E130	E148_b, E3_b
E1_46_b	E1_44_b, E148_f, E3_f, E127
E1_46_f	E54_f, E1_44_f, E14_b, E13_b, E91_f
E3_b	E148_b, E130
E5	E6_b
E6_f	E39, spo_b, E8_f
E3_f	E1_46_b, E127
E89_b	E63_b

C.3 Deleted Metabolites

Table C.3: Table of deleted secondary metabolites and their possible connections to the network. *Possible connections* is the number of reactions this metabolite would be connected to if it is modeled. *Connected reactions* lists each of these reactions connected to the removed metabolites.

Possible connections	Metabolite	Connected reactions
65	H^+	E155, E154, E149, E141, E140, E146, E145, E139, E138, E137, E134, E132, E131, E127, E123, E122, E116, E115, E112, E111, E110, E108, E107, E106, E105, E103, E98, E95, E94, E87, E153, E86, E85, E84, E83, E82, E79, E78, E77, E73, E67, E46, E61, E43, E42, E40, E39, E38, E30, E27, E26, E25, E23, E91, E129, E130, E6, E5, E10, E12, E16, E17, E18, E19, E22
44	H_2O	E155, E147, E141, E140, E146, E145, E139, E138, E137, E136, E135, E131, E128, E120, E119, E118, E115, E110, E106, E103, E99, E98, E96, E95, E86, E82, E79, E73, E66, E64, E63, E129, E15, E7, E5, E19, E31, E32, E33, E36, E37, E38, E43, E45
27	NAD(P)H	E154, E149, E132, E122, E115, E108, E107, E106, E103, E101, E100, E95, E94, E153, E86, E85, E84, E82, E6, E4, E91, E19, E25, E39, E67, E72, E79
27	NAD(P)	E154, E149, E132, E122, E115, E108, E107, E106, E103, E101, E100, E95, E94, E153, E86, E85, E84, E82, E6, E4, E91, E19, E25, E39, E67, E72, E79

Table C.3: (continued)

Possible connections	Metabolite	Connected reactions
24	ATP	E140, E146, E144, E134, E131, E127, E123, E114, E111, E110, E90, E130, E10, E12, E16, E22, E38, E46, E62, E67, E80, E93, E102, E109
20	phosphate	E145, E139, E128, E120, E110, E109, E102, E91, E7, E15, E31, E33, E38, E40, E42, E92, E64, E65, E68, E69
17	ADP	E146, E144, E131, E123, E111, E90, E130, E10, E12, E16, E22, E38, E62, E67, E93, E102, E109
16	CO_2	E154, E138, E129, E4, E21, E23, E61, E72, E73, E74, E93, E100, E101, E105, E112, E136
9	diphosphate	E17, E26, E35, E61, E46, E80, E114, E134, E140
6	AMP	E80, E110, E114, E127, E134, E140
5	O_2	E126, E74, E79, E82, E86
2	H_2O_2	E21, E126
2	bicarbonate	E38, E92
2	S-adenosyl-L-methionine	E83, E87
2	S-adenosyl-L-homocysteine	E83, E87
2	Tetrahydrofolate	E118, E119
2	5,10-methylene-tetrahydrofolate	E118, E119
1	e^-	E97
1	hydrogen sulfide	E116

C.4 Decomposition Comparison

Table C.4: Comparison between automatic and biological decomposition. For each combination of modules between the two decompositions the sets of places, transitions, and edges in common are listed.

Module combination	number of similarities	list of similarities
PLACES (see Table B.1)		
Graph A - *Sucrose module*	61	29, 58, 59, 24, 49, 66, 65, 63, 62, 61, 60, 41, 40, 43, 42, 45, 44, 69, 46, 130, 131, 132, 115, 52, 118, 119, 68, 32, 51, 74, 75, 76, 77, 70, 71, 72, 73, 56, 57, 54, 78, 53, 50, 117, 67, 129, 121, 120, 126, 125, 124, 85, 84, 87, 86, 81, 80, 83, 82, 89, 88
Graph B - *Sucrose module*	15	128, 29, 129, 117, 2, 122, 32, 33, 127, 35, 126, 123, 64, 60, 40
Graph A - *Citrate module*	12	134, 29, 55, 117, 133, 30, 31, 32, 48, 17, 40, 19
Graph B - *Citrate module*	64	29, 28, 23, 8, 21, 20, 27, 26, 25, 40, 47, 112, 113, 110, 111, 116, 117, 114, 98, 99, 92, 93, 90, 91, 96, 97, 94, 95, 38, 39, 32, 33, 34, 35, 36, 37, 12, 13, 10, 9, 16, 17, 14, 15, 18, 79, 101, 100, 103, 102, 105, 104, 107, 106, 109, 108, 1, 6, 7, 4, 5, 2, 11, 3
TRANSITIONS (see Table B.2)		
Graph A - *Sucrose module*	115	E139_u_d, E143_b, E34_b, Import_40, E44, Import_117, R4, E6_b, E1_44, E36_b, E10_49, E17_82, E148_b, E62, E17_77, E10_56, E61_80, E26_b, Export_29, E1_46_b, E20_b, E1_44_b, R6_b, E1_46, E90_b, E43, E30, spo_b, spo_f, E90, E91, OUT_129_rna, E139_u_ttod, Export_40, E16, E14, E15, E12, E13, E11, E18, E19, E130, E131, E132, E137, E138, IN_73, E9_b, OUT_62, OUT_66, R4_b, E28, E27, E26, E25, E24, E23, E22, E20, OUT_82, E29, E8, E9, E6, E7, E4, E14_b, E3, E127, E120, E91_b, IN_63, OUT_78, OUT_71, OUT_74, E39, E54_b, E3_b, E31, E32, E33, E34, E35, E36, E37, E152, E150, E151, E13_b, E61_119, E8_b, E11_b, E18_b, E41, R2, E45_66, R6, E46, Export_32, E145, E144, E5, E143, E149, E148, E24_b, R5_1, R5_2, E45_73, E54, R3, OUT_51, OUT_50, R1
Graph B - *Sucrose module*	15	OUT_122_rna, E144, E147, E146, E139_c_d, Export_40, Import_40, E62, E139_c_ttod, E145, E120, E129_2, E129_1, E35, E128
Graph A - *Citrate module*	14	E88_19, OUT_30b, E153, E95, OUT_30a, E86_133, Export_40, Import_40, E87_134, OUT_55_sinapate_esther_synthesis, E88_48, E85, E86_31, E87_48

Table C.4: (continued)

Module combination	number of similarities	list of similarities
Graph B - *Citrate module*	109	Import_40, E67_95_b, E63, E67, E65, E66_95, E69, E68, E82_111, Import_32, E78, E79, E74, IN_29, E77, E70, E71, E73, E75, E66_101_b, E135, E118, E119, E112, E113, E110, E111, E116, E117, E114, E115, E89_b, IN_93, IN_92, E89, IN_94, E81_111, E84, E80, E83, IN_18, E94_b, E81_100, E96_b, E100, E109, E108, E101, E105, E103, E102, E81_104, E104, E107, E106, E98, E99, E96, E97, E94, E92, E93, Export_40, OUT_99, E102_b, OUT_92, OUT_94, OUT_105, OUT_107, E106_b, OUT_29, E133_b, E133, E134, E81_90, E136, Export_117, E66_101, E67_101_b, E104_b, E21, E64, E2, E71_b, E126, E125, E123, E122, E121, E38, E75_b, E72_106, E72_107, E154, E155, E66_95_b, E82_90, E63_b, E40, E42, OUT_113, E141, E140, E97_b, E67_101, E113_b, E67_95, OUT_93, E2_b
		ARCS (source, target, weight)
Graph A - *Sucrose module*	264	(89, R6_b, 1), (E1_46_b, 65, 1), (44, E1_44, 1), (76, E24, 1), (85, E150, 1), (E1_46, 45, 1), (E54, 32, 1), (E10_56, 56, 1), (50, E24_b, 1), (66, E37, 1), (129, E145, 1), (41, E91_b, 1), (49, E6, 1), (E14_b, 43, 1), (E1_44, 32, 1), (Import_40, 40, 1), (R6_b, 50, 1), (125, E138, 1), (124, E137, 1), (71, OUT_71, 1), (E131, 46, 1), (E18_b, 66, 1), (E34, 69, 1), (E27, 115, 1), (131, E26, 1), (115, E144, 1), (82, E18_b, 1), (E3_b, 46, 1), (spo_b, 56, 1), (82, OUT_82, 1), (E151, 87, 1), (E14, 42, 1), (R4_b, 57, 1), (42, E13_b, 1), (E29, 74, 1), (E36_b, 51, 1), (24, E35, 1), (E39, 62, 1), (89, R4, 1), (80, E120, 1), (E132, 124, 1), (53, E4, 1), (E24_b, 76, 1), (24, E132, 1), (E20_b, 82, 1), (68, E14_b, 1), (E7, 65, 1), (E25, 78, 1), (E8_b, 49, 1), (42, E1_44, 1), (62, E149, 1), (85, E23, 1), (E30, 115, 1), (130, E143, 1), (R3, 59, 1), (68, E13, 1), (E37, 63, 1), (E36_b, 73, 1), (E32, 57, 1), (R5_1, 86, 1), (E143, 78, 1), (74, OUT_74, 1), (73, E22, 1), (E45_66, 73, 1), (67, E11, 1), (E28, 74, 1), (58, R5_1, 1), (132, E151, 1), (E5, 53, 1), (57, E32, 1), (70, E29, 1), (50, OUT_50, 1), (R1, 61, 1), (E24_b, 82, 1), (72, E29, 1), (120, E61_80, 1), (E34_b, 56, 1), (32, E1_44_b, 1), (E20, 77, 1), (E1_46_b, 42, 1), (49, E39, 1), (45, E1_44_b, 1), (65, E54, 1), (E26, 132, 1), (E61_119, 120, 1), (E32, 51, 1), (E139_u_d, 80, 1), (E24, 77, 1), (40, Export_40, 1), (50, E46, 1)

Table C.4: (continued)

Module combination	number of similarities	list of similarities
		(82, E19, 1), (88, E41, 1), (E9, 50, 1), (49, spo_b, 1), (E30, 84, 1), (E6, 54, 1), (44, E16, 1), (IN_73, 73, 1), (43, E14, 1), (42, E1_46, 1), (115, E17_77, 1), (73, E36, 1), (57, R4, 1), (45, E148, 1), (E45_73, 70, 1), (E45_73, 73, 1), (E148_b, 45, 1), (73, E25, 1), (65, E1_46, 1)(52, E130, 1), (E90, 40, 1), (E91, 41, 1), (R2, 57, 2), (E144, 129, 1), (E18, 82, 1), (E16, 43, 1), (51, OUT_51, 1), (75, E36_b, 1), (E91_b, 42, 1), (40, E90_b, 1), (82, E20, 1), (E11, 32, 1), (66, OUT_66, 1), (52, E148_b, 1), (E54, 44, 1), (46, E1_46_b, 1), (E17_77, 77, 1), (52, E3_b, 1), (E1_44, 45, 1), (E61_80, 80, 1), (66, E28, 1), (IN_63, 63, 1), (E10_49, 49, 1), (117, E43, 1), (80, E62, 1), (E139_u_ttod, 115, 1), (61, R4_b, 1), (129, E139_u_ttod, 1), (42, E91, 1), (42, E14_b, 1), (E62, 115, 1), (spo_f, 49, 1), (42, E54, 1), (57, E10_56, 1), (Import_117, 117, 1), (E131, 24, 1), (E90_b, 41, 1), (84, E31, 1), (E6_b, 49, 1), (E152, 130, 1), (E19, 85, 1), (49, E8, 1), (E14, 68, 1), (E130, 60, 1), (E22, 76, 1), (E1_46, 46, 1), (E1_44_b, 42, 1), (29, Export_29, 1), (E45_66, 66, 1), (74, E27, 1), (E23, 57, 1), (62, OUT_62, 1), (76, E17_77, 1), (56, E34, 1), (67, E7, 1), (56, E9, 1), (82, E30, 1), (126, E35, 1), (E13_b, 68, 1), (32, E11_b, 1), (81, R3, 1), (E26, 129, 1), (E37, 51, 1), (R6, 89, 1), (56, spo_f, 1), (E27, 72, 1), (78, E143_b, 1), (51, E10_49, 1), (E17_82, 82, 1), (E13, 42, 1), (E26_b, 131, 1), (129, OUT_129_rna, 1), (E18, 63, 1) (129, E17_82, 1), (E46, 58, 1), (E127, 121, 1), (E54_b, 42, 1), (121, E35, 1), (77, E27, 1), (E8, 44, 1), (43, E15, 1), (70, E45_66, 1), (E3, 52, 1), (E43, 118, 1), (E23, 88, 1), (E150, 131, 1), (87, E152, 1), (69, E34_b, 1), (E44, 119, 1), (E4, 52, 1), (E18_b, 115, 1), (E137, 125, 1), (82, E24, 1), (71, E45_73, 1), (E24, 50, 1), (E31, 83, 1), (63, E18_b, 1), (50, E17_82, 1), (63, E12, 1), (E9_b, 56, 1), (E29, 71, 1), (44, E54_b, 1), (66, E18, 1), (69, E33, 1), (129, E26_b, 1), (77, E20_b, 1), (83, E32, 1), (46, E127, 1), (41, E90, 1), (68, E11_b, 1), (115, E139_u_d, 1), (56, E30, 1), (E149, 63, 1), (E145, 80, 1), (59, R2, 1), (E12, 44, 1), (59, R1, 1), (126, E131, 1), (E36, 75, 1), (77, E24_b, 1), (44, E8_b, 1), (121, E61_119, 1), (54, E6_b, 1), (45, E1_46_b, 1), (119, E61_119, 1), (78, OUT_78, 1), (86, R3, 1), (E35, 80, 1), (E138, 29, 1), (72, E28, 1), (E143_b, 130, 1), (E11, 68, 1), (E120, 126, 1), (R4_b, 89, 1), (115, E18, 1), (E15, 44, 1), (E54_b, 65, 1), (E1_44_b, 44, 1), (54, E5, 1), (118, E44, 1), (E33, 74, 1), (50, E9_b, 1), (R5_2, 81, 1), (32, E54_b, 1), (R4, 61, 1), (132, E26_b, 1), (51, E36, 1), (E28, 70, 1), (58, R5_2, 1), (32, Export_32, 1), (E148, 52, 1), (E11_b, 67, 1), (50, R6, 1), (46, E3, 1)
Graph B - *Sucrose module*	31	(Import_40, 40, 1), (E120, 127, 1), (E146, 122, 1), (128, E144, 1), (127, E147, 1), (E147, 29, 1), (123, E62, 1), (40, Export_40, 1), (E139_c_ttod, 128, 1), (60, E129_1, 3), (60, E129_2, 1), (123, E120, 1), (E145, 123, 1), (E147, 126, 1), (E128, 35, 1), (E144, 122, 1), (128, E139_c_d, 1), (64, E128, 1), (122, E145, 1), (122, E139_c_ttod, 1), (E129_1, 40, 6), (E146, 2, 1), (E62, 128, 1), (122, OUT_122_rna, 1), (E129_2, 40, 1), (127, E35, 1), (33, E146, 1), (E139_c_d, 123, 1), (E35, 123, 1), (129, E146, 1), (E129_2, 64, 1)
Graph A - *Citrate module*	23	(Import_40, 40, 1), (E88_19, 30, 1), (19, E86_31, 1), (48, E88_48, 1), (55, OUT_55_sinapate_esther_synthesis, 1), (133, E87_134, 1), (134, E95, 1), (40, Export_40, 1), (19, E88_19, 1), (31, E87_48, 1), (E153, 48, 1), (E85, 19, 1), (E86_133, 133, 1), (E88_48, 30, 1), (E95, 55, 1), (E86_31, 31, 1), (E87_134, 134, 1), (30, OUT_30b, 1), (30, OUT_30a, 1), (E87_48, 48, 1), (17, E86_133, 1), (17, E85, 1), (134, E153, 1)

Table C.4: (continued)

Module combination	number of similarities	list of similarities
Graph B - *Citrate module*	274	(33, E107, 1), (E113, 9, 1), (Import_40, 40, 1), (E108, 2, 2), (32, E64, 1), (39, E89_b, 1), (E97, 6, 1), (34, E64, 1), (102, E67_95_b, 1), (9, E2_b, 1), (E66_95_b, 37, 1), (35, E126, 1), (E93, 34, 1), (20, E112, 1), (E2_b, 12, 1), (9, E113, 1), (4, E97, 1), (E121, 18, 1), (E82_90, 90, 1), (E155, 8, 1), (10, E125, 1), (E107, 2, 2), (E114, 92, 1), (E66_101_b, 101, 1), (2, E71, 1), (E94, 12, 1), (6, E96, 1), (2, E75_b, 1), (2, E38, 1), (92, E155, 1), (E116, 38, 1), (E81_100, 100, 1), (79, E104_b, 1), (E103, 4, 1), (20, E113_b, 1), (E117, 93, 1), (114, E115, 1), (109, E78, 1), (E81_100, 93, 1), (E96, 8, 1), (29, E106, 1), (E81_90, 108, 1), (E98, 93, 1), (E97_b, 4, 1), (111, E81_111, 1), (E105, 3, 1), (5, E102, 1), (28, E133_b, 1), (E2, 9, 1), (E67, 96, 1), (E89_b, 40, 1), (100, E82_90, 1), (79, E103, 1), (107, OUT_107, 1), (9, E71_b, 1), (E110, 34, 1), (E75_b, 9, 1), (39, E63_b, 1), (E72_107, 107, 1), (E81_104, 104, 1), (12, E98, 1), (9, E107, 1), (13, E155, 1), (93, E114, 1), (21, E104_b, 1), (E66_101, 102, 1), (8, E94, 1), (E102_b, 5, 1), (34, E111, 1), (6, E97_b, 1), (23, E2_b, 1), (47, E81_100, 1), (E104, 79, 1), (E78, 29, 1), (E102, 4, 1), (101, E81_104, 1), (9, E101, 1), (E133_b, 27, 1), (E102, 93, 1), (E113_b, 21, 1), (E141, 23, 1), (101, E67_101_b, 1), (E126, 13, 1), (116, E42, 1), (33, E108, 1), (E133, 6, 1), (25, E136, 1), (96, E68, 1), (9, E108, 1), (102, E67, 1), (90, E81_90, 1), (34, E63, 1), (E113, 20, 1) (E84, 93, 1), (40, Export_40, 1), (E121, 15, 1), (E65, 37, 1), (E111, 20, 1), (23, E140, 1), (29, OUT_29, 1), (E140, 2, 1), (101, E66_101, 1), (95, E67_95, 1), (109, E77, 1), (105, E72_106, 1), (E81_111, 108, 1), (29, E109, 1), (E136, 29, 2), (93, E81_90, 1), (23, E134, 1), (E106_b, 9, 1), (92, E98, 1), (E104_b, 3, 1), (2, E105, 1), (29, E141, 1), (E101, 5, 1), (E70, 105, 1), (E67_101, 101, 1), (37, E67_101, 1), (E135, 25, 1), (10, E100, 1), (E115, 38, 1), (3, E104, 1), (95, E66_95_b, 1), (102, E66_101_b, 1), (92, E117, 1), (29, E118, 1), (E78, 110, 1), (E106, 2, 1), (28, E135, 1), (93, E102_b, 1), (37, E66_95, 1), (105, OUT_105, 1), (E67_95, 102, 1), (E75, 2, 1), (E83, 7, 1), (4, E102_b, 1), (E117, 91, 1), (E2, 23, 1), (36, E65, 1), (E125, 13, 1), (2, E109, 1), (15, E21, 1), (E140, 26, 1), (27, E133, 1), (E116, 113, 1), (102, E81_100, 1), (IN_93, 93, 1), (97, E69, 1), (E109, 33, 1), (E38, 116, 1), (E38, 2, 1), (E71, 98, 1), (108, E83, 1), (94, E134, 1), (2, E113_b, 1), (103, E70, 1), (40, E89, 1), (1, E40, 1), (8, E96_b, 1), (47, E81_104, 1), (E104_b, 20, 1), (E155, 93, 1), (94, OUT_94, 1), (E81_90, 102, 1), (2, E113, 1), (E113_b, 9, 1), (34, E92, 1), (112, E80, 1), (E84, 17, 1), (E71, 9, 1), (IN_29, 29, 1), (E42, 117, 1), (E67_101_b, 37, 1), (9, E75, 1), (38, E114, 1), (23, E42, 1), (E122, 14, 1), (E68, 97, 1), (E100, 9, 1), (E89, 39, 1), (104, E82_111, 1), (107, E75, 1), (116, E40, 1), (110, E79, 1)

Table C.4: (continued)

Module combination	number of similarities	list of similarities
		(E63, 39, 1), (21, E113, 1), (IN_92, 92, 1), (E80, 47, 1), (2, E2, 1), (99, OUT_99, 1), (E79, 112, 1), (E125, 4, 1), (E119, 16, 1), (20, E110, 1), (E106_b, 29, 1), (18, E113, 1), (E134, 27, 1), (E123, 40, 1), (E112, 114, 1), (E135, 1, 1), (34, E68, 1), (E67_95_b, 95, 1), (6, E133_b, 1), (E74, 99, 1), (98, E73, 1), (E77, 29, 1), (E40, 94, 1), (E118, 18, 2), (E75, 106, 1), (E82_111, 111, 1), (93, E80, 1), (33, E38, 1), (117, Export_117, 1), (E81_104, 93, 1), (E98, 11, 1), (13, E113, 1), (29, IN_29, 1), (98, E71_b, 1), (E64, 36, 1), (E92, 12, 1), (IN_18, 18, 1), (E75_b, 107, 1), (93, E101, 1), (15, E122, 1), (E73, 109, 1), (20, E154, 1), (E113, 18, 1), (14, E123, 1), (12, E94_b, 1), (93, OUT_93, 1), (93, E154, 1), (E119, 29, 1), (E2_b, 2, 1), (E96_b, 6, 1), (2, E106_b, 1), (113, OUT_113, 1), (105, E71, 1), (E66_95, 95, 1), (93, E81_111, 1), (18, E119, 2), (33, E140, 1), (E71_b, 105, 1), (E99, 10, 1), (E69, 103, 1), (E113, 2, 1), (20, E104, 1), (E133, 28, 1), (91, E116, 1), (9, E106, 1), (E94_b, 8, 1), (106, E75_b, 1), (E72_106, 106, 1), (11, E99, 1), (13, E121, 1), (106, E74, 1), (26, E141, 1), (E63_b, 34, 1), (E81_111, 101, 1), (E154, 92, 1), (E21, 35, 1), (16, E118, 1), (12, E93, 1), (12, E2, 1), (16, E117, 1), (98, E72_107, 1), (E71_b, 2, 1), (Import_32, 32, 1), (IN_94, 94, 1), (E77, 110, 1), (16, E121, 1), (E104, 21, 1), (7, E84, 1), (92, OUT_92, 1)