Integration, Kombination und Visualisierung multimodaler biologischer Experimentdaten

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.) vorgelegt der

Naturwissenschaftlichen Fakultät III der
Martin-Luther-Universität Halle-Wittenberg

von Herrn
Diplom-Bioinformatiker Hendrik Rohn,
geboren am 26.01.1983 in Pößneck

1. Gutachter Prof. Dr. Falk Schreiber

2. Gutachter Prof. Dr. Oliver Kohlbacher

Verteidigung am: 31. Mai 2012

Quedlinburg, den 05. Juni 2012

Danksagung

Obwohl ich Danksagungen nicht mag (wie übrigens auch Weihnachten und Apple), sollen hier doch einige Personen erwähnt werden, welche auf die Erstellung dieser Arbeit einen großen Einfluss hatten:

Als wichtigste Person ist sicherlich mein Chef Falk Schreiber zu nennen, der als Betreuer und geistiger Vater mit Rat, Motivation und Vorschlägen dazu beigetragen hat, dass mein Schaffen innerhalb der letzten Jahre in einer veröffentlichungswürdigen Arbeit mündete. Ich möchte trotz allem auch meinem Extreme-Programming-Kollegen Christian Klukas danken, mit dem ich mich jahrelang durch die Untiefen der Softwareentwicklung gestritten habe. Dank gebührt weiterhin Astrid Junker für ihre Kreativität und dem ausdauernden Engagement mit mir vernünftige Anwendungsfälle für diese Arbeit zu entwickeln.

Für sehr viele fachliche und auch nicht-fachliche Diskussionen bedanke ich mich vor allem bei Björn Junker und Rainer Pielot, die wohl als die geistigen Mütter dieser Arbeit gelten können (was auch immer das bedeuten mag). Der AG Pflanzenbioinformatik möchte ich für die Zusammenarbeit während der Arbeit und besonders auch für das tolle Klima, die gelungenen Abende nach der Arbeit und die Hilfsbereitschaft bedanken, meine ständige Panik, Sorge und Verzweiflung in Produktivität umzulenken. Genereller Dank auch allen Gruppenmitgliedern der IPKschen Bioinformatik für die gute Zusammenarbeit und offenen Ohren bei alltäglichen fachlichen Problem. Eine Dankeschön für das Korrekturlesen geht an Claudia Herbsleb, Astrid Junker, Tobias Czauderna, Matthias Klapperstück, Carsten Rohn, sowie Jan Hüge.

Ich danke meinen Töchtern Lilith und Ella für die zuverlässig nächtlichen (aber nicht morgendlichen!) Ruhephasen und Ausfüllung des täglichen Lebens. Ein großer Dank gebührt den Junkers und den Köhn-Borcherts, welche maßgeblich das Leben in Quedlinburg in den vergangenen Jahren bereichert und geprägt haben.

Inhaltsverzeichnis

Glossar

$\mathbb{K}$	Menge der Knoten eines Graphen G	18
$\mathbb{E}$	Menge der Kanten eines Graphen G	18
$\mathbb{U}$	Menge der *numerischen Werte*	48
$\mathbb{B}$	Menge der *Bilder*	48
$\mathbb{V}$	Menge der *Volumen*	48
$\mathbb{N}$	Menge der *Netzwerke*	48
$\mathbb{D}$	Menge aller Datenwerte, die biologische Information repräsentieren, $\mathbb{D} = \mathbb{U} \cup \mathbb{B} \cup \mathbb{V} \cup \mathbb{N}$	48
$\mathbb{A}$	Menge aller Metadaten, welche experimentelle Bedingungen beschreiben, $\mathbb{A} = \mathbb{A}_{Experiment} \cup \mathbb{A}_{Lebewesen} \cup \mathbb{A}_{Messung} \cup \mathbb{A}_{Messgroesse}$	48
$\mathbb{M}$	Menge aller Mappings m	54
$\mathbb{T}$	Menge zusätzlicher Attribute eines Mappings, z. B. Positionierung von Datenwerten oder Visualisierungsparameter	54
$md(d)$	Funktion, welche die vier Metadaten eines Datenwertes d liefert	49
$rpr(v)$	Funktion, welche das Objekt o eines Objektknotens v liefert	50
map	Mappingfunktion, welche von einer nicht-leeren Menge Mappings auf ein Mapping abbildet	54
vis	Visualisierungsfunktion, welche Mappings in den euklidischen Raum $\mathbb{R}^i$ projiziert	59
Experiment	Methodisch angelegte Versuchsanordnung in einer kontrollierten Umgebung, um Aussagen über die Struktur und Funktionsweise von Lebewesen treffen zu können	44
Lebewesen	Organisierte genetische Einheit, die zu Stoffwechsel, Fortpflanzung und Evolution befähigt ist	44
Messung	Ausführen von geplanten Tätigkeiten zu einer quantitativen Aussage über eine Messgröße durch Vergleich mit einer Einheit	44
Messgröße	quantitativ bestimmbare Eigenschaft physikalischer Objekte, der eine Messung gilt	44
Metadaten	Beschreibung des Experimentaufbaus, welcher zur Erhebung von Datenwerten geführt hat	45
Datenwert	Wert einer Messgröße, der von einem Messgerät geliefert wird, Datenwerte können auch durch Anwenden von Mappingfunktionen erzeugt werden	45
Mapping	Kombination einer Menge von Datenwerten und $\mathbb{T}$	54
DWI-Knoten	Datenwertimportknoten, repräsentieren alle importierten Datenwerte eines Typs im MappingGraph	51
Objektknoten	Knoten in einem Graphen G, der ein Objekt o repräsentiert	49
Integrationssicht	Sicht auf biologische Experimentdaten, welche durch die Visualisierung eines Mapping mittels einer Visualisierungsfunktion erzeugt wird	61

1

Einführung

Fortschritte in der biologischen Grundlagenforschung und die rasante Entwicklung bioanalytischer Methoden erfordert immer größeren Aufwand bei der Erhebung, Analyse und dem Verständnis anfallender experimenteller Daten. Moderne systembiologische Ansätze fokussieren, unterstützt durch Hochdurchsatz-Methoden, nicht mehr nur auf interessante Einzelphänomene, sondern erheben enorme Datensammlungen, um vielfältige Ursache-Wirkungsbeziehungen in lebenden Wesen erfassen zu können. Aber nicht nur die Quantität der Daten steigt exponentiell, auch die Verfügbarkeit und Qualität der Daten nimmt exponentiell zu. Somit stehen zunehmend Daten unterschiedlichster Typen, Auflösungsebenen und Herkunft zur Verfügung (so genannte multimodale Daten), die jeweils individuelle Sichten auf das biologische System repräsentieren. Wichtige Datentypen sind beispielsweise strukturelle oder funktionelle 2D- und 3D-Bilddaten in Form mikroskopischer Schnittbilder, NMR-Volumendatensätze und Fotografien. Netzwerk-basierte Ansätze unterstützen das Verstehen biologischer Prozesse, etwa in Form metabolischer und genregulatorischer Netzwerke. Numerische Werte wie Metabolitkonzentrationen und Genexpressionsraten werden zunehmend mittels Hochdurchsatz-Methoden generiert und beschreiben lebende Wesen auf verschiedenen Ebenen. Methoden der Informatik werden angewandt, um beispielsweise Systeme zu modellieren und simulieren, Datenhaltung und -austausch zu realisieren, öffentliche Datenrepositories anzulegen und zu verbinden, sowie um Daten zu bearbeiten, visualisieren und analysieren zu können. Die Aufklärung komplexer Zusammenhänge setzt die übergreifende Betrachtung und Integration aller jener verschiedenen Datendomänen und Sichten voraus. Bisher existieren allerdings kaum Ansätze zur Integration, Kombination und Visualisierung multimodaler biologischer Daten in einer Methodik, auch wenn solche Ansätze zunehmend in den Fokus der Forschung rücken [149].

Das Ziel dieser Arbeit ist die Entwicklung einer leicht zugänglichen Anwendung, die biologische Experimentdaten verschiedenen Typs, unterschiedlicher Auflösungsebenen und verteilter Herkunft integriert. Die Daten sollen einfach und flexibel kombiniert werden, um

verschiedene Sichten auf das biologische System miteinander verbinden zu können. Durch geeignete Visualisierungs- und Interaktionstechniken können auf intuitive Weise neue Erkenntnisse erlangt und ansprechende Visualisierungen erzeugt werden.

Kapitel 2 beschreibt die für diese Arbeit notwendigen Grundlagen der Biologie und Informatik. Dazu gehören generelle Konzepte der Biologie und Eigenschaften multimodaler Experimentdaten, theoretische Grundlagen der Informatik, sowie Datenintegration und Visualisierungsmöglichkeiten multimodaler Experimentdaten. In Kapitel 3 folgt die Beschreibung und Analyse der Aufgabenstellung und eine Untersuchung bestehender Ansätze, vorbereitend auf Kapitel 4. In diesem wird die Methodik als Visualisierungspipeline beschrieben, welche aus drei Schritten besteht: *Datenintegration* ermöglicht die Integration multimodaler Daten in zwei Graphstrukturen, welche die Metadaten- respektive Datenwertebene der Experimentdaten repräsentieren. *Datenkombination* erlaubt es, diese integrierten Daten unabhängig von Typ und Herkunft flexibel und iterativ in verschiedene Kontexte zu setzen. *Datenvisualisierung* schließlich ermöglicht die intuitive Darstellung und visuelle Analyse der kombinierten Daten, um umfassenderes Verständnis der Daten erlangen zu können. Kapitel 5 beleuchtet Implementierungsaspekte der Methodik in Form der Anwendung Hive. Kapitel 6 beschreibt beispielhafte Anwendungen der vorgestellten Methodik für verschiedene biologische Datensätze und Fragestellungen. Schließlich fasst Kapitel 7 die Ziele und Erkenntnisse dieser Arbeit zusammen, diskutiert die Vor- und Nachteile der Methodik und gibt einen Ausblick auf zukünftige Entwicklungen.

2

Grundlagen

2.1 Einführung in die Biologie

Biologische Systeme sind sehr komplex, da sie verschiedene Organisationsebenen aufweisen. Neben der Vorstellung des Aufbaus dieser Systeme werden einige für diese Arbeit relevante Fachwörter stichpunktartig erläutert.

2.1.1 Aufbau und Organisation von Lebewesen

Alle Lebewesen bestehen aus einer oder mehreren Zellen, die als grundlegende strukturelle und funktionelle Einheit mehr oder weniger komplexe Organismen bilden. Eine Zelle kann unterschiedliche Formen aufweisen und Funktionen ausüben und besteht aus einer Vielzahl (bio-)chemischer Stoffe. Die Stoffe sind in verschiedene Kategorien einteilbar, wobei insbesondere die (Desoxy-)Ribonukleinsäuren, Proteine und Metabolite wie kleine Moleküle, Zucker und Lipide wichtig sind. Diese elementaren Bauteile organisieren sich zu Mustern, den Netzwerken. Diese Netzwerke bilden Gruppen, welche diskrete Zellfunktionen ermöglichen. Die Zelle selbst besteht aus einer Organisation solcher Netzwerke, räumlich unterteilt in Form von Kompartimenten wie Zellkern, Mitochondrium und Chloroplast. Zellen selbst können in Verbund mit anderen Zellen Gewebe formen, welche wiederum Teil von Organen sind, die den Organismus bilden. Somit weisen Lebewesen verschiedene Organisationsebenen auf, die in Abbildung 2.1 dargestellt werden. Darüber hinaus sind Organismen in einem Gefüge vieler anderer Lebewesen eingebunden. Durch Reproduktion geben diese Informationen an nachfolgende Generationen weiter, wodurch ein Entwicklungsprozess des Lebens über die Zeit erfolgt, der als Evolution beschrieben wird.

Eine der wichtigsten Strukturen von Lebewesen ist die *Desoxyribonukleinsäure* (DNA), welche bei den meisten Organismen die Weitergabe genetischer Information ermöglicht. Die DNA ist ein monomerisches Makromolekül, bestehend aus einem Zucker-Phosphat-Rückgrat und den Nukleotiden Adenin, Cytosin, Guanin und Thymin. Die DNA besteht

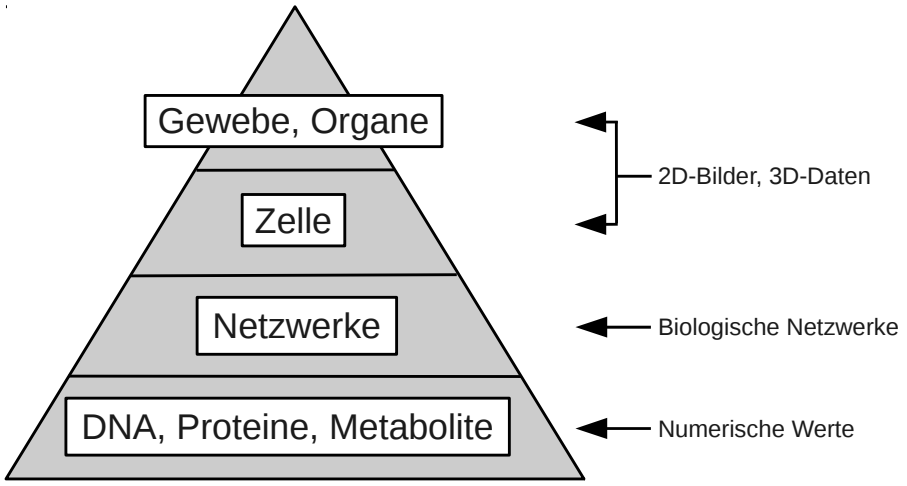


Abbildung 2.1: Schema der Organisation von Lebewesen. Dargestellt sind verschiedene Komplexitätsebenen in der Organisation biologischer Systeme mit von oben nach unten zunehmender Organismenspezifizität und abnehmender Komplexität. Rechts angedeutet sind verschiedene Datentypen, die üblicherweise auf den jeweiligen Auflösungsebenen gemessen werden (vergleiche Abschnitt 2.2, Seite 8). Adaptiert von [103].

aus zwei dieser Stränge, die über Verbindungen zwischen den Nukleotiden Adenin und Thymin, bzw. Guanin und Cytosin zusammengehalten werden und in Form einer Doppelhelix ineinander verdreht sind. Die Reihenfolge der Nukleotide kodiert Informationen zur direkten oder indirekten Bildung verschiedenster Moleküle, welche vielfältige Aufgaben in Zellen übernehmen. Der Bildungsprozess erfolgt durch den Vorgang der *Transkription*, bei der die Reihenfolge der Nukleotide abgelesen und ein neuer, zu der Nukleotidreihenfolge komplementärer Ribonukleinsäurestrang synthetisiert wird. Diese Ribonukleinsäure (RNA) kann durch den Vorgang der *Translation* wiederum in ein monomerisches Makromolekül umgewandelt werden, das aus einer sequentiellen Aneinanderreihung von Aminosäuren besteht. Diese so genannten *Proteine* besitzen meist enzymatische Eigenschaften und wirken als Katalysatoren für bestimmte biochemische Reaktionen. Der Informationstransfer von der DNA über RNA hin zu Proteinen ist ein zentrales Dogma in der Biologie, welches ausschließlich in dieser Richtung abläuft [34]. Biochemische Reaktionen ermöglichen schließlich die Umwandlung von Stoffen in andere Stoffe (so genannte *Metabolite*). Ziel dieser Reaktionen ist die Bildung lebensnotwendiger Stoffe, Energieumwandlung und -speicherung, sowie Giftstoffabbau. Alle Schritte des Dogmas und viele Detailaspekte unterliegen einer äußerst komplexen und feingranularen Regulation. Tiefergehende Informationen über wichtige Moleküle und biochemische Mechanismen können beispielsweise in dem Buch von Stryer *et al.* [143] nachgeschlagen werden.

2.1.2 Biologische Fachwörter

Die wichtigsten Begriffe der Biologie, die in dieser Arbeit verwendet werden, sollen im Folgenden stichpunktartig erläutert werden.

Ribonukleinsäuren sind Moleküle, welche primär zur Informationsspeicherung genutzt werden, aber auch strukturelle und funktionelle Aufgaben besitzen (rRNA, tRNA, mRNA). Sie bestehen aus einem (Ribonukleinsäuren, RNA) bzw. zwei (Desoxyribonukleinsäuren, DNA) Strängen von Nukleotiden. Ein Nukleotid besteht aus einem Phosphat-Rest, einem Zucker (Des-)oxyribose und einer von vier organischen Basen Adenin, Thymin (bzw. Uracil für RNA), Guanin und Cytosin. Die Abfolge der Basen kodiert Information, die z. B. die Erzeugung verschiedener Proteine ermöglicht.

Gene sind Abschnitte auf einer Nukleotidsequenz, welche einen messbaren Phänotyp hervorrufen ([9], Seite 36). Genetische Information kann vererbt werden.

Genotyp ist die Gesamtheit aller vererbbaren Merkmale eines Organismus.

Phänotyp ist die Gesamtheit aller beobachtbaren Merkmale eines Organismus. Einzelne Individuen können den gleichen Genotyp aufweisen, aber einen abweichenden, durch Umwelteinflüsse geprägten, Phänotypen aufweisen.

Genexpression beschreibt die Transkription eines Genes, wenn eine DNA-Polymerase die Nuleotidreihenfolge abliest und in RNA transkribiert. Diese RNA kann wiederum durch Ribosome in Proteine translatiert werden, welche vielfältige Aufgaben in der Zelle ausführen. Die Genexpression beschreibt also den Informationstransfer von DNA zu Zellstrukturen und Zellfunktionen. Die Umwandlung wird durch komplexe Interaktionen von Proteinen mit DNA bei den Ableseprozessen oder bei Transportprozessen feingranular reguliert.

Proteine sind Makromoleküle, bestehend aus einer oder mehrerer unverzweigter Sequenzen von Aminosäuren. Es existieren 20 verschiedene Aminosäuren, deren Rückgrate aneinander gekettet sind und sich insbesondere in ihren Resten unterscheiden. Diese Reste variieren sehr stark in ihren physikochemischen Eigenschaften wie Ladung und pH-Wert und rufen eine Faltung der Aminosäuresequenz hervor. Diese dreidimensionale Struktur und insbesondere die daraus resultierende Anordnung der Reste sind verantwortlich für die Funktionalität von Proteinen. Diese reichen von strukturellen Funktionen (Zellstruktur und -bewegung), über Transportaufgaben, Katalysatoren für biochemische Reaktionen bis hin zu regulatorischen Aufgaben. Üblicherweise erreichen Proteine ihre Funktionalität erst durch Interaktion mit anderen Proteinen oder Molekülen.

Metabolite sind alle im Zellstoffwechsel (*Metabolismus*) als eingehende Reaktanten, Zwischenprodukte oder Produkte vorkommenden Moleküle. Diese sind üblicherweise kleine Moleküle, wie zum Beispiel ATP und Aminosäuren, aber auch große Moleküle wie Kohlenhydrate und Lipide spielen wichtige Rollen. *Kohlenhydrate* sind vor allem als Energiespeicher und -transporter, sowie metabolische Zwischenprodukte relevant. *Lipide* werden sowohl als Energiespeicher, als auch als Signalmoleküle verwandt. Die wichtigste Aufgabe ist aber die Bildung von Membranen, welche die Kompartimen-

tierung biologischer Systeme realisieren. Durch Auf- und Abbau dieser beiden Stoffe kann Energie temporär gespeichert und transportiert werden.

transgene Organismen/Mutanten sind Organismen, deren Erbgut gezielt verändert wurde, um bestimmte Phänotypen zu erhalten. Gegenüber klassischer Züchtung, bei der relativ unkontrolliert phänotypische Merkmale geändert werden, können gezielt Gene ausgeschaltet (*Knockout*), sowie die Genexpressionsrate erhöht werden (*Induktion*).

2.1.3 Biologie als Wissenschaft

Der Aufbau, die Funktionsweise und Veränderung von Lebewesen ist bis heute nur zu einem kleinen Teil verstanden. Nichtsdestotrotz wurden schon mit Beginn der Domestizierung vor 8000 Jahren Lebewesen von Menschen genutzt, um das alltägliche Leben zu vereinfachen, beispielsweise zur Produktion von Essen, Medikamenten und Bekleidung. Traditionell geschieht dies durch gezieltes Kreuzen von Pflanzen und Tieren zur Hervorhebung positiver Merkmale (Züchtung). Durch Entwicklungen neuartiger molekularer Methoden ist es möglich, Lebewesen gezielt zu manipulieren. Ein klassisches Beispiel ist die genetische Veränderung von Bakterien zur Insulinproduktion, welches vorher aufwändig aus Bauchspeicheldrüsen extrahiert werden musste. Dennoch sind aufgrund der bis heute nicht im Detail verstandenen Komplexität biologischer Systeme die vorhandenen biologische Regeln für Wissenschaftler anderer Fächer eher „Wahrscheinlichkeiten, denn Wahrheiten“ ([68], Seite 3). Dies äußert sich insbesondere in der Molekularbiologie, bei der in vielen Versuchsanordnungen erst nach einer Menge von Versuchen das Ergebnis erzielt werden kann oder mit einem Unsicherheitsfaktor belegt ist. Ein großer Teil molekularbiologischer Versuche bleibt aus unerklärlichen Gründen erfolglos [persönliche Kommunikation]. Sowohl die Entwicklung neuartiger Züchtungsansätze, als auch moderner molekularer Methoden, erfordert Wissen über die Funktionsweise biologischer Systeme. Der Prozess der Erkenntnisgewinnung (siehe Abbildung 2.2) in der Biologie ist ein Zyklus und beginnt mit einem Denkmodell, welches vorhandenes Wissen über das System beschreibt. Basierend auf diesem Modell können Hypothesen aufgestellt werden, welche durch Experimente validiert werden müssen. Die aus Experimenten resultierenden Daten müssen dazu analysiert werden und Erkenntnisse daraus können in die Generierung oder Verbesserung des Modells einfließen, wodurch sich der Zyklus schließt. Jeder Durchlauf des Zyklus resultiert somit in neuen Erkenntnissen über biologische Systeme.

Dieses Wissen wird in der Biologie bis heute größtenteils aus empirischer Forschung gewonnen. Dabei sind zwei Arten der empirischen Forschung in der Biologie interessant: Korrelierende Forschung hält alle Variablen des Experiments konstant und untersucht ausschließlich Beziehungen zwischen Mengen von Variablen. Experimentierende Forschung manipuliert explizit Variablen (zum Beispiel die Expression von Genen) und misst den resultierenden Effekt auf andere Variablen. Experimentierende Forschung ist aufwändiger, kann aber leichter kausale Zusammenhänge im System ermitteln ([53], Seite 3). Die Ergebnisse biologischer Experimente sind üblicherweise Daten, welche weiter analysiert werden

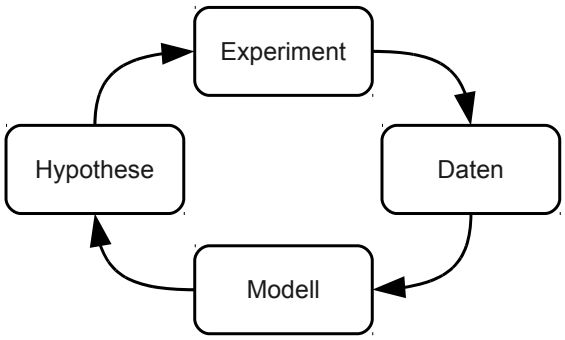


Abbildung 2.2: Zyklus der Wissensgenerierung in der biologischen Forschung. Auf Grundlage des vorhandenen Wissens werden (Denk-)Modelle entwickelt, die Hypothesen generieren können. Diese Vorhersagen werden in Form eines Experiments validiert, indem das Experiment darauf basierend geplant und durchgeführt wird. Durch Analyse der Daten können Informationen zu dem vorhandenen biologischen Wissen beitragen (adaptiert von [133], Seite 72).

müssen, um schlussendlich in Wissen transformiert werden zu können. Datenanalyse ist auf zwei verschiedene Arten möglich ([158], Seite 50): Modellgetriebene Analysen starten mit einem Modell der Funktionsweise des Systems. Dieses Modell wird durch Stichproben geprüft, bestätigt oder widerlegt und eventuell angepasst. Datengetriebene (explorative) Analyse hingegen beschreibt Muster in den vorliegenden Daten, welche als Ausgangspunkt zur Formulierung von Modellen fungieren. Insbesondere für Hochdurchsatz-Daten ist letzterer Ansatz erfolgversprechend und wird auch oft in der Systembiologie angewandt.

2.1.4 Systembiologie

Um komplexe biologische Systeme verstehen zu können, ist es hilfreich, diese als Ganzes zu betrachten (Top-Down Ansatz), statt wie früher in der Biologie üblich, einzelne Elemente zu untersuchen und diese Erkenntnisse in einem größeren Überblick zu kombinieren (Bottom-Up Ansatz). Beispielhaft sei hier der Fakt genannt, dass jedes Organ ein einzigartiges metabolisches Profil besitzt ([143], Seite 851) und dementsprechend erst durch Erhebung aller Metabolite eine umfassende Analyse möglich wird. Unterstützt wird der Top-Down-Ansatz durch die Entwicklung neuartiger Methoden, welche es erlauben, massiv-parallele Untersuchungen durchzuführen [73, 108, 124]. Die Gesamtheit aller Komponenten wird dabei üblicherweise mit der *-om* Endung versehen. Das Genom besteht beispielsweise aus der genetischer Information eines Organismus, welches sich über die Lebenszeit nicht ändert. Das Transkriptom beschreibt alle RNA der Zelle, welche durch Genexpression zu einem bestimmten Zeitpunkt gebildet und noch nicht abgebaut wurde. Das Proteom beschreibt alle Proteine, die durch Translation von mRNA entstehen. Das Metabolom ist die Menge aller Metabolite in einer Zelle zu einem bestimmten Zeitpunkt, welche durch eine große Zahl potentieller Reaktionen Änderungen erfahren und somit eine schnelle Reaktion auf Umwelteinflüsse ermöglichen. Neben diesen wichtigen und oft berücksichtigten -omen existieren viele andere, beispielsweise das Phenom oder Interac-

tom. Von den entsprechenden *Omics-Wissenschaften* existieren derzeit einige Hunderte, welche aber nur teilweise aus neuartigen Ansätzen bestehen ([68], Seite 8) und unter dem Begriff *Systembiologie* zusammengefasst sind.

Neuartige Entwicklungen in den zur Verfügung stehenden Techniken wie zum Beispiel DNA- bzw. Protein-Mikroarrays [28] und Massenspektrometrien [36] erlauben es, mit geringem zeitlichen und monetären Aufwand -ome zu erheben. Durch Wiederholung dieser Experimente für verschiedene Umweltbedingungen und Entwicklungsphasen ist es möglich, den jeweilig kompletten Zustand der Zelle zu erfassen. Der Fokus verschiebt sich also zunehmend vom Einzelphänomen hin zu einer Gesamtansicht, welche Beziehungen der Elemente mit berücksichtigt. Unterstützt werden diese Ansätze durch mathematische und informatische Methoden und Modelle, die biologische Systeme abstrahieren und Vorhersagen liefern können. Je nach Modell kann die Abstraktionsebene hoch oder niedrig sein und wiederum auf anderen Modellen basieren [61]. Diese verschiedenen Modelle gewähren schlussendlich Einblicke in den Aufbau und die Funktion des biologischen Systems auf verschiedenen Ebenen.

2.2 Typen und Erhebung multimodaler Experimentdaten

Entsprechend der komplexen Natur biologischer Systeme und verschiedener Methoden, diese Systeme zu untersuchen, existieren verschiedenste Daten unterschiedlicher Auflösungsebenen, Fokussierung und Abstraktionsebenen. Diese Daten sollen im Folgenden als *multimodale Daten* bezeichnet werden. Eine vollständige Übersicht aller biologischer Daten kann und soll hier nicht gegeben und stattdessen im Folgenden die für diese Methodik interessanten Daten behandelt werden.

In der modernen Biologie gehören Hochdurchsatz-Messungen von Molekülen zum Standardrepertoire. So sind die gleichzeitige Messung hunderter Metabolitkonzentrationen oder tausender Genexpressionsraten innerhalb weniger Stunden möglich. Zunehmend werden diese Daten in den Kontext anderer Daten gebracht und die Beziehungen zwischen den Molekülen hervorgehoben und als Netzwerke modelliert. Daneben gibt es einen hohen Output an räumlichen Informationen wie Mikroskopbilder und *in situ*-hybridisierten Gewebeschnitten. Dreidimensionale bildgebende Verfahren wie NMR und MRT erlauben es schließlich, dreidimensionale Strukturen aufzulösen. Im Folgenden sollen diese verschiedenen Datentypen, die Methoden und die Motivation zur Erhebung der Daten näher erläutert werden.

2.2.1 Numerische Werte

Ein Großteil der biologischen Forschung fokussiert derzeit auf die Charakterisierung einzelner Moleküle oder funktioneller Einheiten, welche zusammengenommen Auskunft über die Funktionsweise des Systems bieten können. Gene als die Träger vererbbarer Information und Ausgangsbasis zur Erzeugung funktionell aktiver Stoffe stehen besonders im Fokus der Forschung, da die Gesamtheit der Gene alle notwendigen Informationen für die Produktion lebenswichtiger Moleküle liefert, insbesondere der Proteine.

Genomics beschreibt das Teilgebiet der Identifizierung beziehungsweise der Messung der Expressionsaktivität von Genen. Zur Entdeckung von Genen und möglichen Mutationen werden die Techniken Southern Blotting, Polymerase-Kettenreaktion (PCR) und verschiedene DNA-Gele benutzt. Sequenzierungsmethoden wie die Sanger-Sequenzierung und 454-Sequenzierung erforschen die Abfolge von Nukleotiden auf der DNA unabhängig von genetischer Aktivität. Genexpressionsdaten werden durch Anwenden von DNA-Mikro- und Makroarrays [28], *in situ*-Hybridisierungen, Real-Time-Quantitative-PCR und Serielle Analyse der Genexpression (SAGE) erhoben. Dabei werden Genexpressionsraten selten quantitativ, sondern relativ zueinander oder binär (exprimiert oder nicht) angegeben. Durch verschiedene *in situ*-Hybridisierungsverfahren ist es möglich, räumliche und zeitliche Genexpressionsmuster bzw. RNA-Konzentration zu messen.

Transkriptomics beschreibt das Teilgebiet der Untersuchung der in der Zelle vorhandenen RNA Moleküle, wobei insbesondere die mRNA als Zwischenschritt der Produktion von Proteinen im Fokus der Experimentatoren liegt. Aber auch tRNA, rRNA und nicht-kodierende RNA können wichtige Informationen über den Zustand der Zelle geben. Northern Blotting und Real-Time-PCR erlauben die quantitative Messung von RNA Molekülen.

Proteomics beschreibt das Teilgebiet der systemweiten Untersuchung von Proteinkonzentrationen, -strukturen und -aktivitäten, wobei die Aufklärung von Proteinstrukturen und -faltungen nicht im Fokus dieser Arbeit liegt. Der Nachweis vorhandener Proteine erfolgt durch verschiedene Geltechniken (1D und 2D Gele), Zentrifugation, Säulenchromatografien (Gaschromatografie, Flüssigkeitschromatografie, Hochdruck-Flüssigkeitschromatografie) und Massenspektrometrien [36] (Elektrospray-Ionisation, MALDI-TOF). Quantitative Aussagen über Proteinkonzentrationen geben Techniken wie Western Blot. Andere Methoden wie beispielsweise Northern Blot weisen zwar nur vorhandene RNA-Moleküle nach, dies erlaubt aber oft Rückschlüsse auf die aktuelle Proteinkonzentration. Proteinaktivitäten sind vornehmlich enzymatische Funktionen (Enzymkinetiken) in biochemischen Reaktionen, welche durch Chromatografien (Gaschromatografie, Hochdruck-Flüssigkeitschromatografie, Dünnschichtchromatografie) und Spektroskopien (UV-Spektroskopie) erhoben werden.

Metabolomics schließlich beschreibt die Untersuchung des systemweiten metabolischen Zustandes. Quantitative Metabolitkonzentrationen können üblicherweise mit verschiedensten Chromatografie- und Spektrometriemethoden erhoben werden, beispielsweise Gaschromatografie kombiniert mit Massenspektrometrie, Flüssigkeitschromatografie kombiniert mit Massenspektrometrie, NMR-Spektroskopie und Ionenmobilitäts-Spektrometrie. Durchflussraten metabolischer Reaktionen (so genannte *Flussdaten*) können über radiometrische Methoden untersucht werden, welche bestimmte Atome, beispielsweise Kohlenstoff (^{13}C), radioaktiv markieren und verfolgen können.

Neben den genannten Gebieten existiert eine Vielzahl weiterer Forschungsgebiete und zusätzlicher Methoden. Als Literatur diente das Buch von Klipp *et al.* ([74], Seite 120–124). Verschiedene Visualisierungsarten numerischer Werte sind in Abbildung 2.7, Seite 24 zu sehen. Bei der Analyse solcher Daten haben in den letzten Jahren insbesondere statistische Verfahren bei der Untersuchung numerischer Hochdurchsatz-Daten aufgrund deren Quantität und Komplexität an Bedeutung gewonnen [20]. Zu solchen Verfahren gehören Korrelationsanalyse, sowie Cluster- und Dimensionsreduktionsverfahren. Sie erlauben es, komplexe Phänomene vereinfacht darzustellen und ermöglichen oder erleichtern damit Dimensionsreduktion, Klassifikation der Elemente und Fehlerreduktion. Für tiefergehende Informationen sei auf die Publikationen [43, 68, 86] verwiesen.

2.2.2 Biologische Netzwerke

Die im vorigen Abschnitt geschilderten Daten geben zwar einen Hinweis auf den Zustand der Zelle, insbesondere durch Nutzung von Hochdurchsatz-Methoden, Beziehungen zwischen den Objektmessungen werden dadurch aber nicht direkt berücksichtigt. Gerade diese sind aber wichtig, da die Interaktionen exponentiell mit der Zahl der Objekte wachsen ([68], Seite 185). So sind beispielsweise genregulatorische Netzwerke wichtiger, als die Gene an sich. Diese höheren Ebenen der Beziehungen können mit einer Netzwerk-orientierten Sicht beschrieben werden ([133], Seite 13), indem vereinfachend die Substanzen als Knoten und die Beziehungen als Kanten eines (Hyper-)Graphen modelliert werden. Biologische Netzwerke repräsentieren die vielfältigen Interaktionen in der Zelle, welche hauptverantwortlich für die Funktionalität und Komplexität von Lebewesen sind. Netzwerke sind grob in zwei Kategorien einteilbar: Masse-übermittelnde Netzwerke, welche die Umsetzung von Stoffen beinhalten und Informationen-übermittelnde Netzwerke, welche ohne Masse-Durchfluss arbeiten. Stelling *et al.* [141] schreiben deswegen, dass man auf einer sehr abstrakten Ebene eine Zelle schon durch die Kombination eines regulatorischen und eines metabolischen Netzwerkes repräsentieren kann, wenn diese sich gegenseitig beeinflussen. In Abbildung 2.3 sind die drei im Folgenden vorgestellten Netzwerktypen dargestellt.

Genregulatorische Netzwerke sind Masse-übermittelnde Netzwerke. Die Rate der Genexpression wird durch viele Prozesse reguliert, insbesondere durch am Promotor bindende Proteine. Resultat ist die Erhöhung oder Verringerung der Konzentration eines Proteines, welches dadurch wiederum die Expression von Genen verändern kann. Abstrahiert man von dem Zwischenschritt der Proteine, so interagieren Gene mit anderen Genen, wodurch sich ein genregulatorisches Netzwerk bildet. Durch zeitliche Verzögerung und wechselnde Umweltbedingungen entstehen somit komplexe räumliche und zeitliche Muster der Genexpression. Die Erhebung solcher Informationen erfolgt z. B. über die ChIP-chip-Technik. *Koexpressionsnetzwerke* hingegen bestehen aus Gene repräsentierenden Knoten, welche durch Kanten miteinander verbunden sind, falls sich die Expressionsmuster beider Gene z. B. über die Zeit ähnlich verhalten. Solche Netzwerke werden auf Basis von DNA-Mikroarrays ([74], Seite 126) erzeugt.

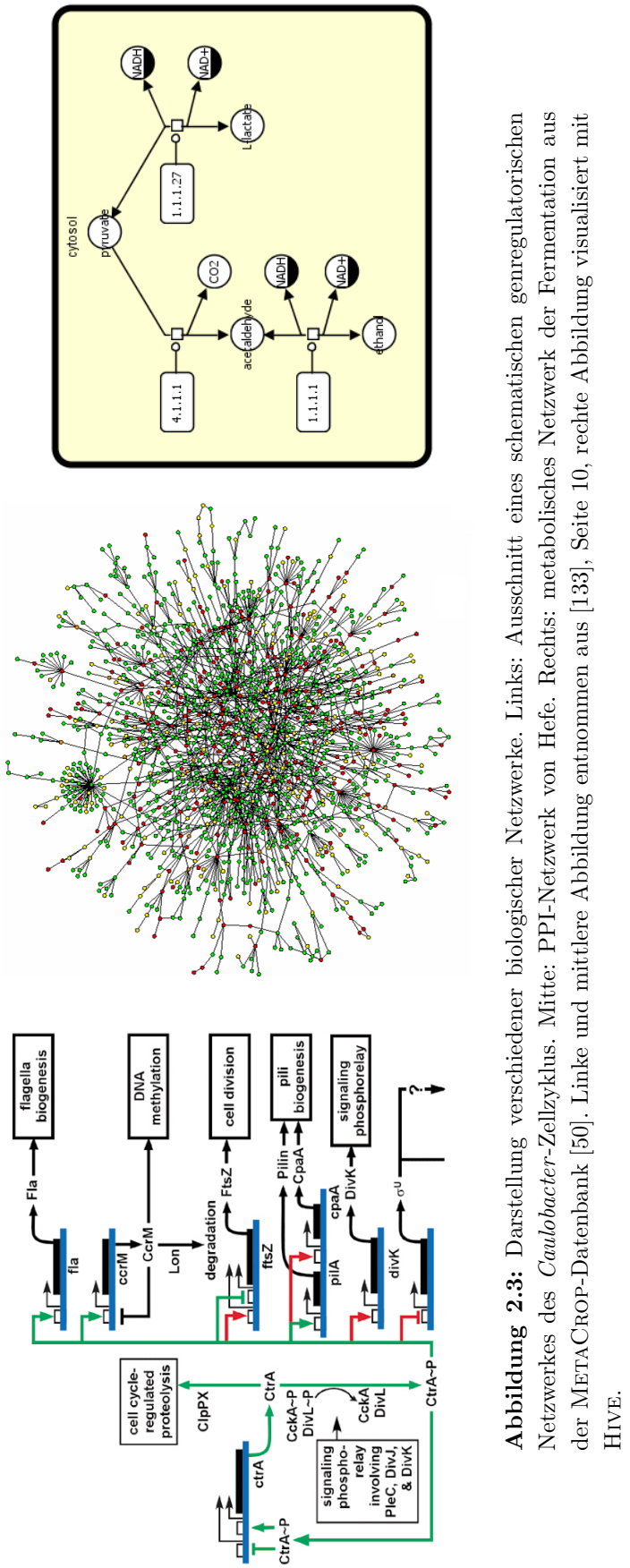


Abbildung 2.3: Darstellung verschiedener biologischer Netzwerke. Links: Ausschnitt eines schematischen genregulatorischen Netzwerkes des *Caulobacter*-Zellzyklus. Mitte: PPI-Netzwerk von Hefe. Rechts: metabolisches Netzwerk der Fermentation aus der MetaCrop-Datenbank [50]. Linke und mittlere Abbildung entnommen aus [133], Seite 10, rechte Abbildung visualisiert mit Hive.

Protein-Protein-Interaktionsnetzwerke (PPI-Netzwerk) Der Zwischenschritt der Genregulation wird durch Proteine kontrolliert, welche aber auch viele andere biologische Prozesse in Zellen kontrollieren. Sie wirken als Katalysatoren, Transporter, Strukturproteine, Signalübermittler und vieles mehr. Einzelne Proteine sind selten funktionell aktiv, sondern erst die Kombination mehrerer Teilproteine oder Interaktion mit anderen Proteinen erfüllt die erforderlichen Aufgaben. Wissen über PPI-Netzwerke sind also für das Verständnis der Zellfunktion unerlässlich. Historisch gesehen ist die wichtigste Methode zur Ermittlung von Proteininteraktionen das Yeast-Two-Hybrid System. Damit können alle potentiellen binären Bindungen (bindet/bindet nicht) von Proteinen ermittelt werden. Eine Aussage über die tatsächlich stattfindenden Interaktionen oder gar der Bindungsspezifität kann allerdings nicht getroffen werden. Analog zu DNA-Mikroarrays bieten Protein-Mikroarrays einen Hochdurchsatz-Ansatz zur Messung von Proteininteraktionen mit DNA, Proteinen und Antikörpern ([74], Seite 127).

Metabolische Netzwerke Sind Proteine enzymatisch aktiv, so regulieren sie die Umsetzung verschiedenster Metabolite, da biochemische Reaktionen überwiegend nicht spontan ablaufen. Ketten solcher Reaktionen realisieren wichtige Zellfunktionen wie Erzeugung (lebens-)notwendiger Stoffe und Abbau schädlicher Stoffe und werden als metabolische Netzwerke bezeichnet. Pro biochemischer Reaktion sind mehrere eingehende und ausgehende Metabolite möglich, insbesondere auch mehrere Moleküle desselben Typs. Metabolische Netzwerke werden folgerichtig im Gegensatz zu den bisher vorgestellten Netzwerktypen als Hypergraphen modelliert. Vielfältige Spektroskopiearten, beispielsweise Flüssigkeits-/Massenspektrometrie, NMR-Spektroskopie und Hochdruck-Flüssigkeitschromatografie erlauben die Hochdurchsatz-Erhebung metabolischer Reaktionen. Identifizierte Proteine katalysieren Reaktionen zwischen den Metaboliten und können durch Literaturrecherchen oder durch Reverse Engineering aus annotierten Genomdaten generiert werden (siehe beispielsweise [74], Seite 157–165, [136, 153]).

Metabolische Netzwerke können durch Differentialgleichungen abgebildet werden: Die Konzentrationen eines Metabolits über der Zeit hängt dabei potentiell von den Konzentrationen bzw. der Umsetzung anderer Metabolite ab. *Kinetische Modelle* erlauben die Verhaltensweisen enzymatisch-katalysierter Reaktionen im Detail zu modellieren und analysieren. Hierzu sind allerdings kinetische Daten der Enzyme notwendig, wie zum Beispiel Geschwindigkeits- und Sättigungskonstanten, aber auch genaue Konzentrationen der einzelnen Metabolite. Da schon kleine Systeme Unmengen solcher Parameter aufweisen und diese teilweise nicht oder nur mit aufwändigen Methoden gemessen werden können, sind solche Modelle meist relativ klein. Abstrahiert man diese Modelle von einzelnen Reaktionsparametern und beschreibt stattdessen die durch die Enzymgeschwindigkeit beeinflussten maximalen Umsetzungsraten, kann eine Aussage über Flüsse im Metabolismus getroffen werden (*stöchiometrische Modelle*). Abstrahiert man die Modelle weiterhin von allen dynamische Eigenschaften wie Umsetzungsraten und Flüsse, kann die Analyse potentieller Pfade durch das Netzwerk wichtige Erkenntnisse über mögliche und unmögliche

Pfade bringen. Dies realisiert die *Elementarmodenanalyse* auf Basis der Netzwerkstruktur und stöchiometrischer Koeffizienten. Ohne Stöchiometrie bleiben ausschließlich binäre Umwandlungsnetzwerke übrig, die mittels Motifsuche, Zentralitätsanalyse ([133], Seite 70) und Ähnlichem auf struktureller Ebene untersucht werden können.

2.2.3 2D-Bilder

Auch in der bildlichen Darstellung entstehen zunehmend Hochdurchsatz-Daten. Durch die fortschreitende Automatisierung in der Mikroskopsteuerung und in der Bildauswertung entwickelt sich auch die Mikroskopie zunehmend zu einer Computertätigkeit ([118], Seite 177). Die bildliche Darstellung hat in der Biologie schon immer eine große Rolle gespielt, insbesondere die Einführung des Lichtmikroskopes im 16. Jahrhundert erforderte die Dokumentation der beobachteten Strukturen in Form von Bildern. Heutzutage werden Bilder durch Fotografie oder Abscannen des Bildes direkt am Mikroskop erzeugt (wie beispielsweise von Ihlow [63] beschrieben). Die klassische Lichtmikroskopie kann Bilder mit einer Auflösung von bis zu 200nm erzeugen (das menschliche Auge erreicht eine Auflösung von 200μm). Die Auflösung des Lichtmikroskopes reicht aus, um Zellen und große Organellen zu erkennen, zum Beispiel in Gewebeschnitten (*histologische Schnitte*). In der normalen Lichtmikroskopie ist es üblicherweise nicht notwendig, Präparate speziell vorzubehandeln. Die Einführung von Lasern hat neben der ständig verbesserten Automatisierung eine entscheidende Rolle in der Entwicklung des Lichtmikroskopes gespielt. So ist es beispielsweise möglich, mittels des *Konfokalmikroskopes* (CLSM, Confocal Laser Scanning Microscope) dreidimensionale Objekte aufzunehmen (weitere Informationen hierzu siehe Abschnitt 2.2.4, Seite 15). Elektronenmikroskope nutzen Elektronenstrahlen, um eine theoretische Auflösung von bis zu 0,2nm zu erreichen. Jedoch ist die Durchdringungstiefe von Elektronen sehr gering. Im Falle des *Transelektronenmikroskopes* (TEM) muss das Material extrem dünn sein (meist nicht mehr als 100nm). Das *Rasterelektronenmikroskop* (REM) scannt die Oberfläche von (getrockneten) Präparaten mit Hilfe von Elektronenstrahlen, wobei ein charakteristischer 3D-Effekt entsteht. Die Auflösung des REM ist etwas geringer als die des TEM (1nm gegenüber 0,2nm). Da die Form des Materials aber keine Rolle spielt, können auch die Oberflächen von sehr großen Objekten studiert werden. Bilderzeugung mithilfe tomographischer Verfahren wie NMR, PET, MRT/MRI und CT werden im Abschnitt 2.2.4, Seite 15 im Rahmen der 3D-Datenerhebung genauer betrachtet.

Auch die Makroskopie spielt in Form von Fotografien in der Biologie eine große Rolle. Vor allem phänotypische Merkmale wie Größe, Farbe und Form werden immer noch vorrangig fotografisch festgelegt. Dahingegen sind Fotografien von Proteingelen, Mikroarrays und Ähnlichem eher als methodischer Zwischenschritt, denn als Ergebnisse zu verstehen. Neben den beschriebenen Verfahren existieren viele abgeleitete und verbesserte Methoden zur Bilddatenerhebung, die aber in der Biologie eine eher untergeordnete Rolle spielen.

Neben der Erhebung struktureller Information werden heutzutage immer häufiger Methoden angewandt, um funktionelle bzw. dynamische Daten zu erhalten. Bis auf wenige Ausnahmen sind diese Methoden nur mittels Lichtmikroskope realisierbar. Einige Sub-

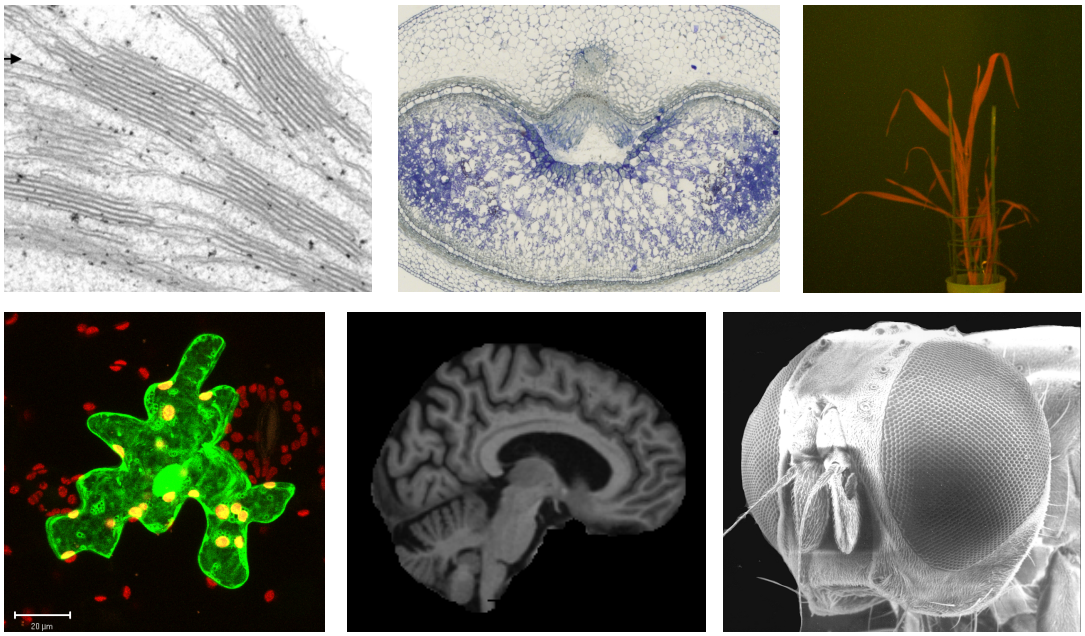


Abbildung 2.4: Auswahl typischer 2D-Bilddaten. Links oben: Detailaufnahme eines Mais-Chloroplast mit TEM. Mitte oben: Methylenblau gefärbter Querschnitt durch ein zwölf Tage altes Gerstenkorn, Lichtmikroskop. Rechts oben: fotografische Autofluoreszenz einer Gerstepflanze unter UV-Licht (Quelle: Anja Hartmann, IPK). Links unten: GFP-Expression und Chlorophyllautofluoreszenz einer Tabak-Epidermiszelle, 3D Aufnahme mit CLSM. Mitte unten: Protonen-NMR-Messung eines menschlichen Gehirnes (Quelle Dr. Rainer Pielot, IFN Magdeburg). Rechts unten: Detailaufnahme eines Fliegenkopfes mit REM. Datenquelle soweit nicht anders angegeben: Stefan Ortleb (IPK).

stanzen, wie z.B. Chlorophyll, können durch *Autofluoreszenz* in noch lebenden Systemen studiert werden. In anderen Fällen werden Stoffe wie Proteine, Stärke, Lipide, RNA, Metabolite und Hormone durch *Anfärben* hervorgehoben. *Antikörpermarkierungen* (Immunhistologie) ermöglichen den spezifischen Nachweis einzelner Komponenten, können meistens aber nur an fixiertem Material ausgeführt werden. *GFP-Anfärbungen* heben bestimmte Proteine oder Genprodukte mittels eines *green fluorescent protein* hervor, wobei die Helligkeit des Signales nur qualitative Information beschreibt (z. B. Gen ist angeschaltet oder nicht). *In situ*-Hybridisierungen erlauben die farbliche Markierung des Erbgutes oder einzelner Gene und damit die direkte Untersuchung molekularer Strukturen [147]. Spezifischere molekularbiologische Methoden markieren gezielt einzelne Gene und Genprodukte in lebenden Systemen. Alle diese Färbemethoden erlauben den Nachweis bestimmter Substanzen in zweidimensionaler Auflösung, teilweise sogar mit quantitativer Aussage. Einige der vorgestellten Bilddatentypen sind in Abbildung 2.4 dargestellt.

Die wichtigste Bilddaten-Operation ist die *Segmentierung*, welche als die Erzeugung von inhaltlich zusammenhängenden Regionen durch Zusammenfassen benachbarter Pixel entsprechend eines bestimmten Homogenitätskriteriums definiert ist [24]. Segmentierte Bilder werden im Folgenden als *Labelfields* bezeichnet und stellen jedes Segment mit einer

eindeutigen Farbe dar. Die Segmentierung ist eine der wichtigsten und schwierigsten Operationen in der Bildanalyse [15]. Für weitere Hintergründe und verschiedene Methoden der Segmentierung sei hier auf die Publikationen [15, 51, 105, 118, 157] verwiesen.

2.2.4 3D-Daten

Die Erhebung zweidimensionaler Bilddaten liefert nur ein Ausschnitt des üblicherweise dreidimensionalen Objektes. Sollen aber Strukturen in ihrer Gesamtheit untersucht werden, beispielsweise weil die äußere Form ausschlaggebend für die Funktionalität sein kann, so müssen die Methoden für zweidimensionale Daten auf drei Dimensionen erweitert werden. 3D-Daten sind dabei nicht nur auf struktureller Ebene interessant, sondern liefern auch dreidimensionale funktionelle Information wie beispielsweise *in situ*-Hybridisierungen von Zebrafischen [93]. Es ist möglich, die Form und das Volumen von Teilbereichen wie Kompartimenten, Geweben oder Organen zu erfassen und diese als Umrechnungsfaktoren in der Erhebung von beispielsweise Proteinkonzentrationen zu berücksichtigen. Der Nachteil von 3D-Daten gegenüber 2D-Bilder sind die hohen monetären und zeitlichen Kosten der Messungen, der Bedarf an aufwändigen Auswertungs- und Analysewerkzeugen und hohe Anforderungen an die Visualisierung und Interaktion (siehe dazu auch Abschnitt 2.5.2.4, Seite 27). Dreidimensionale Bilddaten liegen üblicherweise entweder als Volumenmodell oder als Oberflächenmodell vor (siehe auch Abbildung 2.8, Seite 28).

Volumenmodelle (im Folgenden *Volumen* genannt) stellen eine Menge von Werten an definierten Punkten in einem 3D-Raum dar, die untereinander keine explizite Verbindung aufweisen ([135], Seite 15). Volumen diskretisieren also das biologische Objekt (analog zu Pixeln in 2D-Bilddaten) in Würfel, den so genannten Voxeln (Volumenelement). Voxel haben eine definierte Ausdehnung, sowie einen Farb- und einen Transparenzwert, welche eine quantitative Aussage über gemessene Stoffe innerhalb dieses Bereiches geben, beispielsweise die Konzentration von Wasserstoff-Protonen. Die Erhebung dreidimensionaler Volumendaten erfolgt in den meisten Fällen durch Messen zweidimensionaler Bilddaten (so genannte Schnitte) in mehreren Ebenen des biologischen Objektes, die möglichst dickekonstant und parallel zueinander liegen sollten. Durch Alignmentalgorithmen kann die dreidimensionale Struktur ermittelt werden. Der Prozess der Aneinanderordnung von (3D-)Bilddaten wird auch Registrierung genannt. In einem solchen Stapel besitzt ein Voxel jeweils die Breite, Höhe und Position eines Pixels in einem Schnitt. Die Tiefe des Voxels wird durch die Abstände der Schnitte zueinander bestimmt. Der Prozess der *Volumenrekonstruktion* ist in Abbildung 2.5 grafisch zusammengefasst. *Kernspinresonanzspektroskopie* (NMR), beziehungsweise die methodisch eng verwandte *Magnetresonanztomografie* (MRT bzw. MRI, fMRI) bieten die Möglichkeit, verschiedenste Stoffe als volumetrische Daten zu messen ([30] Seiten 19–40 und 134–148). Üblicherweise werden Protonensignale als strukturelle Informationen betrachtet, funktionelle Information hingegen sind beispielsweise Wasser, Lipid-, Zucker- und Aminosäuresignale. Viele Stoffe können aufgrund geringer Konzentrationen oder schlechter Lösbarkeit kaum nachgewiesen werden. Weiterer Nachteil ist die relative geringe Auflösung ($> 1\mu m$) tomographischer Methoden. *Compu-*

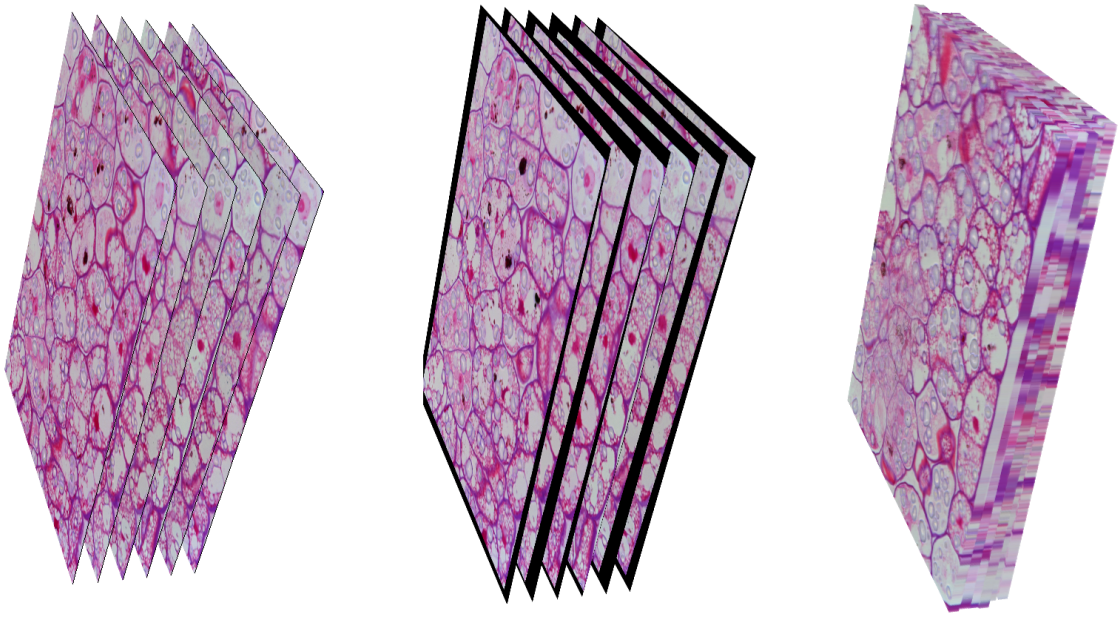


Abbildung 2.5: Erzeugung von Volumendaten eines Rapssamen-Zellverbundes aus Schnittstapeln. Links: Lichtmikroskopische Schnittbilder. Mitte: registrierte Schnittbilder. Rechts: daraus generiertes Volumen (visualisiert mit HIVE). Datenquelle: Stefan Ortleb (IPK).

tertomografie (CT) ist ein schnelles und kostengünstiges Verfahren, welches die Aufnahme von Röntgenbildern aus verschiedenen Richtungen ermöglicht, um einen dreidimensionalen Datensatz zu erzeugen. Da aber Röntgenstrahlen ausschließlich die Dichte des Voxels messen können, sind die Einsatzmöglichkeiten auf die Erhebung struktureller Datensätze beschränkt und im Gegensatz zur diagnostischen Medizin in der Biologie eher unüblich. *Positronen-Emissionstomografie* (PET) ermöglicht die Erhebung funktioneller Parameter als 3D-Volumen, indem das biologische System mit radioaktiv markierten Substanzen versetzt und deren Verteilung nach einer bestimmten Zeit gemessen wird ([30], Seite 3–18). So lassen sich beispielsweise Transportvorgänge in Wurzeln aufnehmen [65]. PET findet vor allem Anwendung in der Onkologie und Neurologie humaner Systeme, da für biologische Fragestellungen die Kosten oft zu hoch sind. Alle geschilderten Verfahren erlauben die so genannte nicht-invasive Erhebung dreidimensionaler Daten, da die biologischen Objekte bei der Messung nicht zerstört werden. Somit ist es auch möglich, die Daten am lebenden Objekt zu erheben. Mikroskopische Verfahren dagegen erfordern das Zerschneiden des Objektes in Scheiben. Diese werden fixiert, eingescannt und aligniert. Konfokalmikroskope ermöglichen die Erhebung dreidimensionaler Daten in bis zu 500facher Vergrößerung auch von etwas dickeren Scheiben. Durch Nutzung einer Lochblende können alle Objekte außerhalb der Brennebene bzw. des Fokus ausgeblendet werden. Durch Ändern der Brennebene ist es möglich, optische Schnitte mit Dicken $< 5\mu m$ zu erheben ([118], Seite 231). Ungewollte Veränderungen der Probe durch Schnitt- und Fixierungsartefakte können somit teilweise umgangen werden. Neben den geringen Kosten der Konfokalmikroskopie können Farbstoffe, Antikörper oder *in situ*-Hybridisierungen genutzt werden, um funk-

tionelle Parameter abzuleiten. Invasive Methoden erlauben also zusammenfassend hohe Auflösungen und vielfältige Anfärbemethoden, bedürfen aber eines hohen Aufwands im Sinne der Datenerhebung und Rekonstruktion [131].

Oberflächenmodelle beschreiben nur einen für die Anwendung wichtigen Teil dreidimensionaler Objekte, um beispielsweise strukturelle Eigenschaften hervorzuheben. In der Biologie repräsentieren diese Oberflächen üblicherweise konzeptuelle Eigenschaften wie Gewebe- und Organ(-ellen)grenzen. Damit ist es möglich, das Volumen (im Sinne der Ausdehnung) von Organen und Geweben zu berechnen, Formen abzuschätzen und von Strukturdetails abstrahierte Visualisierungen zu erstellen (siehe auch Abschnitt 2.5.2.4, Seite 27). Die Erzeugung von Oberflächenmodellen kann auf drei Wegen realisiert werden: Eine Möglichkeit ist die Modellierung der 3D-Struktur direkt durch Experten, beispielsweise Zellorganellen als Anschauungs- und Lehrmaterial [139]. Isoflächen (Isosurfaces) sind Flächen, die im Raum benachbarte Voxel mit gleichen Merkmalen (gleicher Farbwert) miteinander verbinden. Dies kann beispielsweise eine interessante Konzentration eines Metaboliten, die Rate der Genexpression oder Gewebegrenzen sein und wird üblicherweise durch den Marching Cube Algorithmus berechnet [88]. Neben diesem existieren weitere Verfahren wie der Cuberrille-Ansatz oder der Dividing-Cubes-Ansatz ([161], Seite 6). Letzte und verbreitetste Methode, Oberflächenmodelle zu erheben, geschieht unter Nutzung manueller, semi-automatischer, oder im besten Falle vollautomatischer Segmentierung der alignierten Volumenschnitte (vergleiche auch Abschnitt 2.2.3, Seite 13). Die Oberfläche wird dabei durch eine binäre Segmentierungsfunktion definiert, welche jedem Voxel die Werte 1 (ist Objekt bzw. Oberfläche) oder 0 (ist Hintergrund) zuweist. Danach folgt durch Anlegen eines Polygongitters an die Grenzen der Segmentierungsflächen die Rekonstruktion des dreidimensionalen Objektes (Triangulation). Dem folgen üblicherweise verschiedene Optimierungen der Oberfläche wie Glättung und Remeshing, um Datenerhebungs- oder Rekonstruktionsfehler zu eliminieren. Durch Segmentierung der Ursprungsdaten in mehrere Segmente können geschachtelte Oberflächenmodelle erzeugt werden [15].

Volumenmodelle und Oberflächenmodelle weisen verschiedene Vor- und Nachteile auf. So werden Volumen als „realitätsnähere“ Abbildung des biologischen Objektes wahrgenommen, da jede einzelne gemessene Information berücksichtigt wird. Auch erlauben sie es, diffuse Objekte mit undeutlich definierten Abgrenzungen realistisch darzustellen. Durch fehlende optische Hilfen wie Überdeckung (viele Voxel sind halbtransparent) oder Schattenwurf sind räumliche Zusammenhänge relativ schwierig abzuschätzen. Oberflächenmodelle abstrahieren die Daten auf wenige Details, wie zum Beispiel Gewebegrenzen. Dadurch kann eine intuitiv verständlichere und weniger aufwändige Visualisierung angeboten werden. Oberflächenmodelle können aber durch Fehler im Segmentierungs- und Rekonstruktionsprozess Artefakte wie Löcher oder unvollständige Strukturen aufweisen. Insbesondere aufgrund der Festlegung von scharfen Strukturgrenzen, die aus biologischer Sicht nicht zwangsweise scharf sein müssen, werden Volumenmodelle von Fall zu Fall bevorzugt.

2.3 Mathematische und informatische Grundlagen

Einige wichtige mathematische und informatische Grundlagen und Konzepte sollen im Folgenden, basierend auf den Definitionen aus Cormen *et al.* ([32], Seite 1070–1090) und Junker und Schreiber ([68], Seite 17–22), vorgestellt werden.

2.3.1 Mengen

Eine *Menge* $\mathbb{A} = \{a_1, a_2, \ldots, a_n\}$ ist eine Sammlung unabhängiger, ungeordneter und eindeutiger Objekte a_1, a_2, …, a_n. Die Objekte a_i werden *Element* der Menge $\mathbb{A}$ genannt und $a_i \in \mathbb{A}$ geschrieben. Enthält eine Menge keine Elemente, so heißt diese *leere Menge* ($\mathbb{A} = \emptyset$). Im Folgenden soll der logische Operator „und“ als $\wedge$, das logische „oder“ als $\vee$, sowie das logische „exklusiv oder“ als $\dot{\vee}$ bezeichnet werden. Mengen können auch durch Bezug auf andere Mengen mittels folgender Notation definiert werden, z. B. $\mathbb{A} = \{a : a \in \mathbb{N} \wedge \frac{a}{2} \text{ ist ganzzahlig}\}$ ($\mathbb{N}$= Menge der natürlichen Zahlen). Zwei Mengen $\mathbb{A}$ und $\mathbb{B}$ sind gleich, wenn jedes Element aus $\mathbb{A}$ auch Element der Menge $\mathbb{B}$ ist und umgekehrt. Eine Menge $\mathbb{A}$ wird *Untermenge* der Menge $\mathbb{B}$ genannt, falls alle Elemente aus $\mathbb{A}$ in $\mathbb{B}$ enthalten sind, aber Elemente aus $\mathbb{B}$ nicht zwangsläufig Elemente von $\mathbb{A}$ sind. Man schreibt $\mathbb{A} \subseteq \mathbb{B}$. Eine *Vereinigung* zweier Mengen $\mathbb{A} \cup \mathbb{B}$ ist eine Menge bestehend aus allen Elementen der Mengen $\mathbb{A}$ und $\mathbb{B}$ ($\mathbb{A} \cup \mathbb{B} = \{x : x \in \mathbb{A} \vee x \in \mathbb{B}\}$). Die *Schnittmenge* $\mathbb{A} \cap \mathbb{B}$ ist eine Menge von Elementen, welche sowohl in $\mathbb{A}$, als auch in $\mathbb{B}$ vorkommen ($\mathbb{A} \cap \mathbb{B} = \{x : x \in \mathbb{A} \wedge x \in \mathbb{B}\}$). Zwei Mengen heißen *disjunkt*, wenn sie kein gemeinsames Element besitzen und somit gilt: $\mathbb{A} \cap \mathbb{B} = \emptyset$. Die *Differenz* zweier Mengen ist die Menge aller $a_i \in \mathbb{A}$, welche nicht Teil der Menge $\mathbb{B}$ sind ($\mathbb{A} \setminus \mathbb{B} = \{x : x \in \mathbb{A} \wedge x \notin \mathbb{B}\}$). Ein *geordnetes Paar* zweier Elemente a und b ist definiert als $(a, b) = \{a, \{a, b\}\}$. Ein *Tupel* ist eine endliche, nicht-leere und geordnete Liste von nicht notwendigerweise verschiedenen Elementen. Das *kartesische Produkt* zweier Mengen ist die Menge aller geordneten Paare, deren erstes Paar-Element Teil der Menge $\mathbb{A}$ und deren zweites Paar-Element Teil der Menge $\mathbb{B}$ ist ($\mathbb{A} \times \mathbb{B} = \{(a, b) : a \in \mathbb{A}, b \in \mathbb{B}\}$). Wichtige Mengen sind beispielsweise die natürlichen Zahlen $\mathbb{N}$ und die reellen Zahlen $\mathbb{R}$. Mehrere Mengen heißen *disjunkte Mengenfamilie*, wenn ihre Elemente paarweise disjunkt sind und somit gilt: $\mathbb{A}_i \cap \mathbb{A}_j = \emptyset$ für $i \neq j$.

2.3.2 Funktionen

Eine *Funktion* f ist eine binäre Relation für $\mathbb{A} \times \mathbb{B}$, so dass für alle $a \in \mathbb{A}$ genau ein $b \in \mathbb{B}$ existiert und gilt: $(a, b) \in f$. Eine Funktion bildet also jeweils von einem Element einer Menge $\mathbb{A}$ auf ein Element einer Menge $\mathbb{B}$ ab. Man schreibt eine Funktion üblicherweise als $f(a) = b$ oder $f : \mathbb{A} \mapsto \mathbb{B}$.

2.3.3 Graphen

Ein *Graph* $G = (\mathbb{K}, \mathbb{E})$ ist ein Paar, bestehend aus der *Menge der Knoten* $\mathbb{K}$ und der *Menge der Kanten* $\mathbb{E}$. Jedes System, bestehend aus diskreten Objekten bzw. Zuständen und Beziehungen dazwischen, kann als Graph modelliert werden. Knoten und Kanten

sollen im Folgenden verallgemeinernd als *Graphelemente* bezeichnet werden. Eine Kante verbindet zwei (nicht notwendigerweise verschiedene) Knoten u und v. Der Grad eines Knotens ist die Zahl der anliegenden Kanten. Graphen werden üblicherweise durch Punkte (Knoten) und Linien zwischen Punkten (Kanten) visualisiert. Die Positionen der Kanten und Knoten werden als das *Layout* des Graphen bezeichnet ([132], Seite 31). Bei *ungerichteten* Graphen werden Kanten zwischen den Knoten u und v als ein ungeordnetes Paar $\{u, v\}$ definiert. Kanten in *gerichteten* Graphen werden hingegen als geordnetes Paar (u, v) definiert, besitzen also einen Quell- und einen Zielknoten und werden oft als Pfeil dargestellt. Gemischte Graphen enthalten entsprechend sowohl gerichtete, als auch ungerichtete Kanten. Ein Knoten x heißt *Vorgänger* respektive *Nachfolger* von y in einem gerichteten Graphen G, wenn (x, y) respektive (y, x) eine gerichtete Kante von G ist. Ein *Pfad* bezeichnet eine Sequenz von Knoten mit verbindenden Kanten, ausgehend von einem Startknoten und endend bei einem Endknoten. Pfade mit dem gleichen Start- und Endknoten werden als *Zyklen* bezeichnet. Basierend auf dem Vorhandensein von Zyklen können Graphen somit zyklisch oder azyklisch sein. Zwei Knoten werden als *verbunden* bezeichnet, falls ein Pfad zwischen beiden besteht. *Attributierte Graphen* sind Graphen, deren Kanten und/oder Knoten Attribute wie Text, numerische Werte, Farben und Koordinaten besitzen. Attribute werden als Funktion, abbildend von dem Graphelement zu einem Attributtyp, repräsentiert, beispielsweise weist die Funktion $w : \mathbb{E} \mapsto \mathbb{R}$ Kanten $e \in \mathbb{E}$ ein Gewicht $w(e)$ zu. *Gerichtete azyklische Graphen* (directed acyclic graphs, DAG) sind gerichtete Graphen, welche keine Zyklen aufweisen. Ein *Hypergraph* $G = (\mathbb{K}, \mathbb{E})$ besteht aus der Menge der Knoten $\mathbb{K}$ und der Menge der Hyperkanten $\mathbb{E}$. Eine *Hyperkante* ist mit einer nicht-leeren Menge von Knoten verbunden. Hypergraphen werden für die Modellierung metabolischer Netzwerke genutzt, da metabolische Reaktionen mehrere Eingangs- bzw. Ausgangsmetabolite besitzen können ([68], Seite 20).

2.3.4 Algorithmen

Algorithmen sind wohl-definierte Berechnungsvorschriften, welche eine Menge von Werten als Eingabe erhalten und eine Menge von Ausgabewerten produzieren ([32], Seite 5). Algorithmen repräsentieren eine Abfolge von Berechnungsschritten, welche die Eingabe in die Ausgabe umwandeln und stellen somit eine Funktion dar ($f(n)$ mit n = Eingabe). Zur Vergleichbarkeit von Algorithmen können die wichtigen Eigenschaften Laufzeit und Speicherbedarf mit der *O-Notation* theoretisch abgeschätzt werden. Für eine gegebene Funktion $g(n)$, mit n = Länge der Eingabe, existieren positive Konstanten c und $n_0 \leq n$. Dann ist die O-Notation ([32], Seite 44)

$$O(g(n)) = \{f(n) : 0 \leq f(n) \leq c\, g(n)\}. \tag{2.1}$$

Die O-Notation beschreibt damit die obere Grenze einer Funktion und somit die schlechtest mögliche Laufzeit für die Eingabe n. Aussagen über möglicherweise bessere Laufzeiten für bestimmte Eingabewerte können hingegen nicht getroffen werden. Die O-Notation ist im Gegensatz zu anderen Notation (Θ, Ω und mehr siehe [32], Seite 42–50) relativ leicht durch Analyse der Algorithmenstruktur ermittelbar und erlaubt das Rechnen mit

O-Notationen verschiedener Algorithmen, insbesondere das Addieren, Multiplizieren und Vergleichen. Ein *Overhead* ist ein im Sinne der O-Notation konstanter (und somit bei großen n vernachlässigbarer) Anteil an der Gesamtlaufzeit eines Algorithmus. Für praktische Anwendungen kann ein hoher Overhead dennoch Nachteile aufweisen, insbesondere wenn die Größe der Eingabe gering ist. Üblicherweise beschreibt der Overhead den Aufwand für Initialisierung und ähnliche technische Details.

Für eine Übersicht und Erklärung der in dieser Arbeit erwähnten und bekannten Datenstrukturen und Algorithmen kann beispielsweise in dem Buch von Cormen *et al.* [32] nachgeschlagen werden.

2.4 Datenintegration

Datenintegration bezeichnet die Lösung des Problems, Daten verschiedener Quellen zu kombinieren und dem Anwender eine vereinte Sicht auf die Daten zu ermöglichen [87]. Üblicherweise erfolgt die konzeptionelle Integration, statt einer Datenzentralisierung ([104], Seite 1). Das Problem verteilter Daten entsteht, da Anwendungen üblicherweise eigene Dateien und Datenstrukturen verwenden, Daten an verschiedenen Orten erhoben und abgespeichert werden, sowie der Zugriff aufgrund fehlender zentraler Verwaltungsinstanzen unkontrolliert abläuft. Die Übertragung von Daten zwischen Anwendungen erfordert oft aufwändige und potentiell fehleranfällige Konvertierungen. Auch die technische Entwicklung trägt zu einem erhöhten Bedarf an Datenintegration bei ([31], Seite 2). Insbesondere der Übergang von wenigen Großrechnern hin zu PCs resultiert in einer Dezentralisierung des verfügbaren Datenbestandes. Aber auch Entwicklungen in den jeweiligen Fachgebieten erhöhen die Quantität der Daten, beispielsweise hat sich das Datenaufkommen in der Biologie durch die Entwicklung neuartiger Hochdurchsatz-Verfahren in kurzer Zeit vervielfacht [73, 108, 124]. Folgerichtig verbringen biologische Wissenschaftler einen großen Teil ihrer Arbeit mit der Datenerhebung, -bereinigung und -konvertierung [142].

Obwohl Datenintegration allein keine Erklärungen wie ein mathematisches Modell bieten kann, ist Datenintegration in der Biologie auf verschiedenen Ebenen nötig ([74], Seite 12): einfachste Ebene ist die Erstellung einheitlicher Schemata für die Datenspeicherung, Datenrepräsentation und den Datentransfer. Die zweite Ebene der Datenintegration bietet Datenretrieval, Verbindung verschiedener Datentypen und Visualisierung bzw. Präsentation der Daten. Die dritte Ebene beschreibt Datenkorrelation zwischen verschiedenen Datensätzen, z. B. Korrelation von Genexpressionsdaten und metabolischer Flüsse. Die höchste Ebene ist das Kombinieren von integrierten Daten verschiedener Quellen. Wichtige Voraussetzung der Datenintegration auf den verschiedenen Ebenen ist die Modellierung der Daten, unter deren Zuhilfenahme die Realisierung der Datenintegration üblicherweise in Form von Datenbanken oder Flat Files erfolgt.

2.4.1 Datenmodellierung

Datenmodellierung bezeichnet die Erzeugung einer geeigneten abstrakten Beschreibung interessanter Teile der abzubildenden Welt. Geeignet sind Modelle, die möglichst stark ver-

einfachen, aber dennoch Antworten auf alle wichtigen Fragen geben können. Wichtig sind also Simplizität, Flexibilität und Universalität. Ausgangspunkt einer Datenmodellierung ist somit die Einschränkung der Fragestellungen an das Modell. Eine wichtige Anforderung an das Modell ist weiterhin, dass Implementierungsaspekte verborgen bleiben, um z. B. die Diskussion mit technisch nicht versiertem Personal zu ermöglichen. Datenmodelle teilt man oft in drei Klassen ein ([39], Seite 24): Konzeptionelle Modelle, schematische Modelle und physische Modelle. Konzeptionelle Modelle beschreiben dabei die Sicht eines Anwenders auf die Daten, wohingegen physische Modelle detailliert beschreiben, wie die Daten abgelegt werden. Schematische Modelle liegen zwischen beiden Extremen und erlauben es, Datenstrukturen verständlich zu halten, aber dennoch direkt implementierbar zu sein.

Es existieren verschiedene Sprachen zur Datenmodellierung. Die zwei wichtigsten sind das Entity-Relationship-Modell (ERM) und die Unified Modeling Language (UML). Für detaillierte Erklärungen und Anwendungsmöglichkeiten der Sprachen sei auf die Publikationen von Balzert [8], Chen [23] und Elmasri *et al.* [39] verwiesen.

2.4.2 Datenbanken und Flat Files

Datenbanken sind Sammlungen zueinander in Beziehung stehender Daten ([39], Seite 4) und die verbreitetste Art, Datenintegration zu realisieren. Datenbanken erlauben es laut Balzert ([8], Seite 743) mehreren Anwendern parallelen Zugriff auf den Datenbestand zu ermöglichen, Daten redundanzarm zu speichern, große Mengen an Daten vorzuhalten, bei technischen Fehlern auf einen früheren Stand zurückzuspringen, Zugriffsrechte feingranular zu vergeben und die darunterliegende Datenstruktur begrenzt anzupassen. Insbesondere durch das hohe Datenaufkommen in der Molekularbiologie besteht ein großer Bedarf an Datenbanken, der sich in der Zahl von über 500 verschiedenen spezialisierten Datenbanken widerspiegelt [17]. Trotz der großen Menge verschiedener Ansätze enthalten viele dieser Datenbanken enorme Datenmengen, die sowohl exponentiell wachsen, als auch immer komplexer werden [7, 82] und auch ausgeprägt verknüpft sind. Gründe für die Datenbank-Heterogenität in der Biologie sind einerseits technische Aspekte hinsichtlich Datensicherheit und Optimierungskriterien, aber auch ein fehlendes, weit anerkanntes einheitliches Datenbankschema, einheitliche Vokabularien, eingeschränkter Anwendungsbezug und fehlende internationale Koordination. Lösungen für Unternehmen lassen sich nur schwer auf die Biologie anwenden [5], auch wenn zunehmend Anstrengungen unternommen werden, diese mittels Technologien des Data Warehouse, Mediatoren und föderierten Datenbanken zu realisieren (wie beispielsweise [4, 84]). Einheitliche Vokabulare wie Ontologien helfen zunehmend, biologische Fachbegriffe über Systemgrenzen hinweg zu vereinheitlichen und die Datenintegration durch höhere Interoperabilität zu vereinfachen. Diese Ontologien sind formale Definitionen von Klassen und deren Beziehungen und enthalten das Vokabular einer Domäne ([158], Seite 34), so zum Beispiel die Gene Ontology [151].

Auch andere Ansätze als Datenbanken ermöglichen Datenintegration, beispielsweise *Flat Files*. Diese sind einfache Dateien im Dateisystem, welche strukturierte Daten (so genannte Records) ohne Beziehungsinformationen speichern und dementsprechend einfach realisierbar. Flat Files sind Datenbanken vorzuziehen, wenn ([39], Seite 19)

- die Vorteile von Datenbanken nicht oder nur teilweise wichtig sind
- die Kosten für Hardware, Software und Einweisung für eine Datenbank zu hoch sind
- der allgemeingültige Datenbank-Ansatz wenig Vorteile für den Anwender bringt
- der Overhead der Sicherungen zu hoch ist (z. B. Zugriffsicherung, Konsistenzwahrung, Backup)
- besondere Anforderungen an die Geschwindigkeit der Datenhaltung bestehen oder
- die Komplexität der zugreifenden Anwendungen relativ gering ist und die Anwendungen sich erwartungsgemäß kaum ändern.

2.5 Datenvisualisierung und Interaktion

Visualisierung ist ein wesentlicher Aspekt der in dieser Arbeit vorgestellten Methodik. Im Folgenden sollen generelle Visualisierungskonzepte vorgestellt und insbesondere auf die Visualisierung biologischer Experimentdaten eingegangen werden.

2.5.1 Grundlagen der Visualisierung

Der Mensch ist ein vornehmlich bildverarbeitendes Wesen, da ca. 75% seines gesamten Informationsaustausches über die visuellen Organe erfolgt. Visualisierung verstärkt die kognitive Arbeitsleistung um ein Vielfaches und ist die Transformation von Wissen oder Daten in visuelle Formen, beispielsweise Bilder und Diagramme. Da viele verschiedene Visualisierungen für ein Phänomen möglich sind, spricht man von verschiedenen Sichten, welche unterschiedliche Eigenschaften visuell hervorheben oder unterdrücken können. Visualisierung wird von Card *et al.* ([19], Seite 6) wie folgt definiert:

> „The use of computer-supported, interactive, visual representations of data to amplify cognition."

Visualisierung sollte zum Erkenntnisgewinn und zur verbesserten Einsicht in Daten genutzt werden, statt nur Bilder zu erzeugen. Rechner unterstützen den Menschen, indem große Datenmengen aufbereitet und analysiert, bevor sie visuell präsentiert werden und der Anwender etwaige Muster schneller als mit automatischen Methoden erfassen kann [86]. Die wichtigste Art der Unterstützung, die geeignete Visualisierungen bieten können, sind dabei laut Card *et al.* ([19], Seite 16) z. B. die Erweiterung der Speicher- und Berechnungsressourcen durch direkte Nutzung des visuellen Systems, Verkürzung der Suche nach Information durch Zoomen, hierarchische Suche und die Gruppierung ähnlicher Daten.

Viele Daten können direkt durch Visualisierung aufbereitet und dargestellt werden, da sie räumliche Daten (z. B. Volumendaten) oder zumindest Abstraktionen räumlicher Daten darstellen (*wissenschaftliche Visualisierung*). Sind Daten nicht-physischen Ursprunges, beispielsweise Finanzdaten, Geschäftsprozesse und konzeptuelle Informationen (z. B. metabolische Reaktionsnetzwerke), so sind auch diese durch Visualisierung in verständlichere

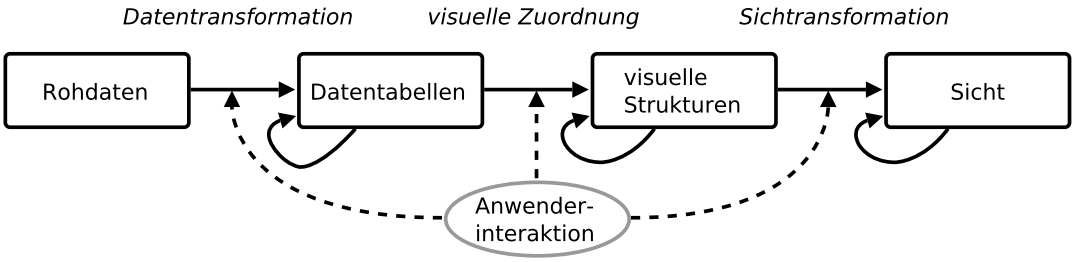


Abbildung 2.6: Visualisierungspipeline adaptiert von [19], Seite 17. Rohdaten werden in Tabellen transformiert, welche visuellen Strukturen zugeordnet und schließlich als Sicht dargestellt werden. Jeder Umwandlungsschritt kann durch Anwenderinteraktion beeinflusst und parametrisiert werden.

Formen transformierbar. Aus diesem Grund existiert das Gebiet der *Informationsvisualisierung*, welches „die Rechner-gestützte, interaktive bildliche Repräsentation abstrakter Daten zur Verbesserung der Erkenntnis“ ermöglicht ([19], Seite 7). Eine andere Kategorisierung von Visualisierungen bezieht sich auf die Art der Verwendung: Visualisierungen werden einerseits erzeugt, um Wissen effektiv kommunizieren zu können. Dies wird als *knowledge visualization* [22] bezeichnet. Andererseits können Visualisierungsmöglichkeiten insbesondere im Zusammenhang mit Interaktionstechniken genutzt werden, um Daten analysieren und somit vorher unbekanntes Wissen aus den Daten extrahieren zu können. Diese Art der Visualisierung wird als *visual analysis* bzw. *visuelle Analyse* [71] bezeichnet und ist beispielsweise in der Analyse komplexer biologischer Systeme ein wichtiger Faktor. Auch wenn es derzeit noch eine klare Trennung zwischen diesen Gebieten der Visualisierung gibt, ist nicht sicher, ob eine solche überhaupt notwendig ist. Viele Methoden in den Gebieten überschneiden sich oder werden kombiniert angewandt, um die gewünschten Ergebnisse zu erzielen (siehe hierzu beispielsweise [66, 116]).

Die Erzeugung von Visualisierungen aus Daten wird üblicherweise als *Visualisierungspipeline* strukturiert, von denen viele verschiedene existieren [27]. In Abbildung 2.6 wird eine bekannte Visualisierungspipeline dargestellt, welche Rohdaten in Sichten umwandelt. Rohdaten sind üblicherweise Tabellen, Texte, Punktwolken und Ähnliches. Der Schritt der Datentransformation wandelt Rohdaten in Datentabellen um, indem deren Struktur für spätere Transformationen in eine Menge von Relationen umgewandelt, um Metadaten erweitert und auch Datenänderungen wie Fehlerbereinigung oder Normalisierung durchgeführt werden. Solche Tabellen werden durch visuelle Zuordnungsfunktionen in visuelle Strukturen umgewandelt, die räumliche Einbettung und andere grafische Eigenschaften aufweisen. Üblicherweise sind diese Strukturen schematische Darstellungen und einfache Grafiken wie Diagramme. Sichttransformationen wandeln diese Strukturen parametrisiert in Sichten auf diese Strukturen um, welche Eigenschaften wie Kameraposition, Daten-Positionierung, Skalierung und Ähnliches enthalten. Die Sicht erlaubt es schlussendlich dem Anwender, auf dynamische Art und Weise Aufgaben zu lösen („visual sense making“ ([19], Seite 33)). Jeder der drei Umwandlungsschritte Datentransformation, visuelle

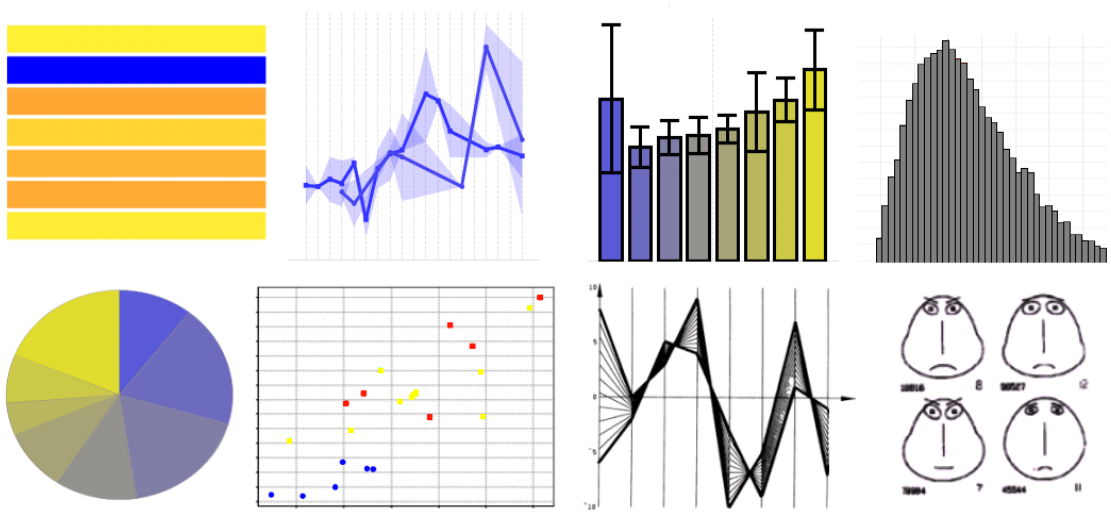


Abbildung 2.7: Verschiedene Visualisierungen numerischer Werte. Oben: Heatmap für verschiedene Umweltbedingungen, Liniendiagramm mit Standardabweichungs-Schatten, Balkendiagramm mit Fehlerbalken, Histogramm. Unten: Tortendiagramm. Streudiagramm mit verschiedenen Umweltbedingungen, Parallele Koordinaten, Chernoff-Gesichter. Visualisierungen mit HIVE, außer Parallele Koordinaten (entnommen aus [64]) und Chernoff-Gesichter (entnommen aus [26]).

Zuordnung und Sichttransformation kann durch Interaktionen des Anwenders beeinflusst werden.

2.5.2 Visualisierung und Repräsentation multimodaler Experimentdaten

Die in Abschnitt 2.2, Seite 8 vorgestellten Experimentdaten können auf verschiedene Arten visualisiert werden. Bekannte Visualisierungsmethoden sollen in diesem Abschnitt vorgestellt werden. Eine sehr gute Übersicht über verschiedene Aspekte der Visualisierung biologischer Experimentdaten geben auch Walter *et al.* [157].

2.5.2.1 Numerische Werte

Genomics-, Transkriptomics-, Proteomics- und Metabolomics-Daten sind üblicherweise durch Tausende numerische Werte der Molekülkonzentrationen oder -aktivitäten beschrieben. Zunehmend stehen somit nicht mehr einzelne Messwerte von Stoffen (=Variablen), sondern die unterschiedlichen Verhaltensweisen der Variablen zu verschiedenen Zuständen im Mittelpunkt. Dementsprechend wird versucht, üblicherweise mit statistischen Analysen (Konfidenzabschätzung, Clusterverfahren, etc.), ähnliche Verhaltensweisen aufzudecken. Aber auch (üblicherweise zweidimensionale) Visualisierungsmethoden werden zur Analyse der Daten eingesetzt (siehe Abbildung 2.7). Die einfachste Form der Datenvisualisierung ist als (grafisch komprimierte) Tabelle möglich, zum Beispiel in Form so genannter *Heatmaps* für Mikro- und Makroarraydaten, wobei die Messwerte durch eine Farbkodierung sichtbar gemacht werden. Für verschiedene Bedingungen ergeben sich somit unterschied-

liche globale Muster, welche allerdings die Zusammenhänge zwischen Daten nur schwer erkennbar machen [86]. *Liniendiagramme* stellen Sequenzen von Messwerten dar, welche eine eindimensionale Beziehung aufweisen. Die als Punkte repräsentierten Messwerte werden durch Linien entsprechend der Ordnung verbunden. Typischerweise sind diese Daten Zeitserien oder (räumliche) Gradienten. *Balken-* bzw. *Stabdiagramme* und *Histogramme* erlauben es, Häufigkeitsverteilungen von Variablen darzustellen. Die x-Achse repräsentiert die möglichen Werte, während die y-Achse die Häufigkeit dieser Werte darstellt. Es ist erforderlich, die Messwerte in Kategorien einzuteilen, um diese als eine endliche Menge von Balken darzustellen. Die Kategorisierung bzw. Diskretisierung erfolgt dabei über Wertebereiche oder logische Kategorisierungen, wie zum Beispiel verschiedene Umweltbedingungen. Balkendiagramme stellen in vielen Fällen den Mittelwert der Kategoriemesswerte dar, oft zusammen mit Fehlerbalken. Histogramme stellen nicht die Werte direkt, sondern die Zahl der Werte in der jeweiligen Kategorie dar. Sie geben visuelle Hinweise auf Mittelwert und Varianz der Daten, aber auch Ausreißer und weitere Verteilungseigenschaften. *Tortendiagramme* repräsentieren Mengenverhältnisse von Messwerten. Die Kategorien der Messwerte werden ähnlich zu den Histogrammen basierend auf der Zahl der enthaltenen Messwerte (relative Frequenz) gewichtet, normiert und auf einer Scheibe dargestellt. *Streudiagramme* visualisieren die Beziehung zwischen zwei Variablen, indem die Wertebereiche als Achsen dienen und für jedes Wertepaar ein entsprechender Punkt eingezeichnet wird. Stehen beide Variablen auf irgendeine Weise miteinander in Beziehung, kann die resultierende Messwerte-Punktwolke eine bestimmte Form annehmen, beispielsweise eine gerade Linie oder Kurve. *Parallele Koordinaten* stellen Daten als Linienzüge entlang zueinander paralleler Balken dar. Die Balken repräsentieren Variablen und der Schnittpunkt der Linie mit dem Balken entspricht dem Wert der jeweiligen Variable. Diese Darstellung wird oft zur Analyse von Genexpressionsdaten gewählt, auch wenn die Interpretation der parallelen Koordinaten schwierig ist. Sowohl Streudiagramme, als auch parallele Koordinaten können in Matrix-Form vorkommen [86]. *Piktogramme* (auch ikonisierte Darstellung genannt) repräsentieren die Werte vieler Variablen durch einfache grafische Darstellungen, wie zum Beispiel Strichmännchen, Gesichter, Farben und Formen. Die Darstellung der Messwerte ist dabei durch möglichst eindeutige, minimale Strukturänderungen realisiert, zum Beispiel die Position der Arme von Strichmännchen oder die Positionierung der Augen und Form des Kopfes bei Chernoff-Gesichtern [26]. Es sind aber auch Kreis-ähnliche Strukturen möglich, bei denen eine vom Mittelpunkt kreisförmig ausgehende Menge von Balken mit der Länge abhängig vom Variablenwert visuelle Vergleiche der Daten ermöglichen. Obwohl die Menge der Balken (und damit der Messwerte) sehr groß sein kann, sind globale und lokale Muster der Messwerteverteilung leicht für den Menschen erkennbar. Flussdaten werden üblicherweise im Kontext biologischer Netzwerke als Pfeile visualisiert, deren Dicke dem Flusswert einer Reaktion entspricht (wie beispielsweise in FBASimVis [49]).

Zusammenfassend existiert eine große Zahl von Visualisierungen raumloser biologischer Daten, die hier nicht alle genannt oder im Detail erläutert werden können (z. B. Sequenzlogos, Genomdarstellung, etc.). Der Großteil der vorgestellten Visualisierungstypen basiert auf den Ausführungen von Hill *et al.* ([53], Seite 533–554) und Lehmann *et al.* [86].

2.5.2.2 Biologische Netzwerke

Die Visualisierung von Netzwerken erlaubt es, eine komplette Übersicht über alle Beziehungen zwischen Netzwerkelementen zu erhalten oder auf spezielle Teilnetzwerke zu fokussieren. Der Fluss der Information kann verständlich dargestellt werden und ermöglicht es durch geschickte Visualisierung, Hinweise auf alternative Pfade im Netzwerk oder die Hervorhebung wichtiger oder interessanter Strukturen zu realisieren. Eine entscheidende Rolle kommt dabei dem Netzwerklayout zu. Genregulatorische Netzwerke werden üblicherweise mit Kräfte-basierten oder hierarchischen Layouts versehen ([133], Seite 33), teilweise unter Nutzung analytischer Methoden wie Netzwerkzentralitäten [78] und Motifen. PPI-Netzwerke besitzen üblicherweise eine einfache Struktur, weswegen Kräfte-basierte Layouts genügen, um die Eigenschaften der Netzwerke hervorzuheben ([133], Seite 42). Visualisierung metabolischer Netzwerke ist dagegen deutlich komplizierter. Grund ist die komplexe Struktur des Netzwerkes, die als Hypergraph modelliert wird, räumlich abgetrennte Bereiche aufweist (Kompartimente) und verschiedene Elemente, die berücksichtigt werden müssen: Ko-Substanzen, Produkte und Reaktanten besitzen beispielsweise eine spezielle Positionierung oder verschiedene Knotengrößen, die in der Visualisierung beachtet werden müssen. Kräfte-basierte Layoutmethoden, obwohl sie oftmals eingesetzt werden, erfüllen viele Visualisierungsanforderungen metabolischer Netzwerke nicht.

Biologische Netzwerke lassen sich als Graphen modellieren, wobei Substanzen als Knoten und Umwandlungsreaktionen bzw. Interaktionen als Kanten dargestellt werden (siehe Abbildung 2.3, Seite 11). Wichtige Kriterien zur Visualisierung von Netzwerkdaten stellt Schreiber ([132], Seite 39) auf:

- die zeitliche Reihenfolge von Reaktionen sollte grafisch repräsentiert werden
- Darstellung von Netzwerk-Ausschnitten sind dem Darstellen des gesamten Metabolismus vorzuziehen
- es sollten verschiedene Sichten auf die Netzwerke angeboten werden
- verschiedene Visualisierungen sollten ähnlich sein; lokal sollten alle nach dem gleichen Schema dargestellt werden, global sollten gleiche Reaktionswege in verschiedenen Visualisierungen möglichst ähnlich sein

Tiefergehende Informationen sind in den Publikationen von Schreiber [132, 133] zu finden.

2.5.2.3 2D-Bilder

Zweidimensionale Datenvisualisierung ist die am häufigsten anzutreffende Datenvisualisierung, da diese am besten verständlich und optimal für die üblicherweise zweidimensionalen Ausgabemedien wie Monitore und Projektoren ist. Die 2D-Darstellung ist besonders effektiv zur Visualisierung abstrakter Daten und zum Navigieren in den Daten [155]. Zweidimensionale Grafiken wie Bilder werden meist durch ein Raster aus Bildpunkten (so genannte Pixel) repräsentiert. Ein Pixel ist somit die kleinste Einheit einer digitalen *Rastergrafik* und stellt den Farbwert des Bildes in diesem Rasterelement dar. Die Matrix-ähnliche

Struktur kann durch Zoomen und Panning (siehe Abschnitt 2.5.3, Seite 28) interaktiv analysiert werden. Auf Basis von Pixeln werden Bildverarbeitungsmethoden angewandt, um Informationen aufzuarbeiten, beispielsweise Segmentierung und Objekt- und Mustererkennung. Wichtige Attribute des Bildes sind dabei Histogramme der Farb- und Helligkeitsverteilung, abrupte Farbwechsel und ähnlich-farbige Gebiete. Wird ein Bild statt durch Rasterung durch grafische Primitive (Linien, Kreise, Polygone, Kurven etc.) beschrieben, so spricht man von *Vektorgrafiken*. Diese haben den Vorteil, dass die Bearbeitung der Inhalte, z. B. Skalierung und Zerrung, verlustfrei möglich ist. Geeignet zur Vektordarstellung sind alle Bilder, welche durch grafische Primitive befriedigend beschrieben werden können, z. B. Logos, Diagramme, Netzwerke und Zeichnungen. Für biologische Anwendungen sind allerdings fast ausschließlich Rasterbilder von Interesse. Vektorgrafiken kommen nur in schematischen Darstellungen zum Einsatz, beispielsweise mittels Microsoft PowerPoint. Einige Darstellungen von Rastergrafik-Bilddaten sind in Abbildung 2.4, Seite 14 zu sehen.

2.5.2.4 3D-Daten

Visualisierung in drei Dimensionen erlaubt eine effektive Darstellung physikalischer 3D-Daten und einen erleichterten Überblick über ganze Datensätze [155]. Nachteile der dreidimensionalen Darstellung sind hohe Hardware-Anforderungen (Speicherbedarf und Rechenzeit), eine große Zahl veränderbarer Parameter (Beleuchtung, Texturen, mindestens sechs Freiheitsgrade in der Bewegung, etc.), Überlagerung von Objekten abhängig vom Sichtwinkel und die ungenügende Darstellung von Text, welcher insbesondere in abstrakten Daten oft vorkommt ([19], Seite 61). Zur Verbesserung des 3D-Eindruckes auf zweidimensionalen Ausgabemedien können Scheinräumlichkeiten und stereoskopische Verfahren [112], sowie Virtuelle Realitäten verwendet werden [131]. Basierend auf der Art der Daten (siehe Abschnitt 2.2.4, Seite 15) werden 3D-Daten üblicherweise mittels Volumenrendering oder Oberflächenrendering visualisiert (vergleiche Abbildung 2.8). Spezielle abstrahierende Visualisierungsarten wie Cartoon- oder Stick-And-Ball-Darstellungen sind für diese Arbeit nicht interessant.

Volumenrendering ist eine Technik, dreidimensionale Arrays zu visualisieren ([135], Seite 19). Es sind nur wenige Vorprozessierungsschritte nötig, um qualitativ hochwertige Visualisierungen zu erzeugen, insbesondere wenn wichtige Strukturen basierend auf den Voxelintensitäten unterscheidbar sind [157]. Die Technik basiert auf dem Modellieren der Daten als „durchscheinendes Gel“, weswegen vorher den Datenwerten Materialeigenschaften wie Transparenz und Farbwert zugewiesen werden müssen. Dies wird durch zwei Funktionen, der Farbtransferfunktionen und der Transparenzfunktion erreicht [16], welche Voxel mit einem bestimmten Wert eine Farbe und eine Transparenz zuweisen. Die Wahl der Funktionen ist essentiell für die Qualität der Darstellung, da insbesondere falsche Transparenz eine räumliche Abschätzung der Struktur stark erschwert. Automatische Transferfunktionen können dabei helfen, besondere Gebiete, wie z. B. Gewebegrenzen, hervorzuheben [72]. Die Darstellung von Millionen transparenter Voxel macht Volumenrendering zu einer sehr rechenaufwändigen Aufgabe, die mit verschiedenen Methoden realisiert werden kann: *Maximum-Intensity-Projection* ([161], Seite 10), *Raycasting*, *Splat-*

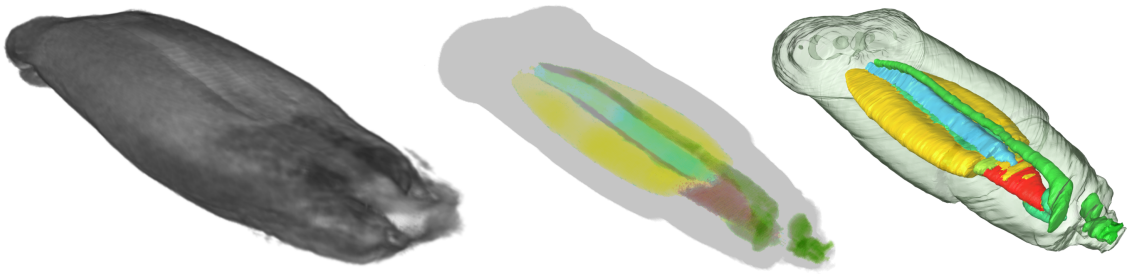


Abbildung 2.8: Verschiedene 3D-Rendermethoden eines Gerstenkorns. Links: Raycasting-Volumenrendering eines unsegmentierten NMR-Volumens. Mitte: Schnitt-basiertes Volumenrendering eines aus segmentierten Schnitten erzeugten Volumens. Rechts: Oberflächenrendering eines aus diesen Daten rekonstruiertes Oberflächenmodells. Gut zu erkennen sind die scharfen Grenzen des Oberflächenrendering gegenüber Volumenrendering und Sprünge in den Daten aufgrund fehlender Schnitte. Visualisierung mit ParaView [85], Hive und Amira [140].

ting, *Shear-Warp* ([135], Seite 21–25) und *Texture Mapping*. Letzteres nutzt, im Gegensatz zu den anderen Verfahren, direkt die Grafikhardware, um Volumen zu rendern [16, 135] und ist entsprechend performant. Dazu werden Schnitte in alle orthogonale Richtungen des Volumens erstellt, als Texturen auf Rechtecke gelegt und schließlich als Texturstapel in 3D aneinandergereiht. Die Grafikhardware kümmert sich dann um alle Blickwinkel-Interpolationen. Bei 3D-Texturen, welche von allen modernen Grafikkarten unterstützt werden, ist sogar nur eine Textur in Gebrauch, welche wiederum aus einem Stapel von Texturen besteht und schneller als einzelne Texturen dargestellt werden kann. Diese Texture Mapping Renderer werden auch als *Schnitt-basierte Volumenrenderer* [146] bezeichnet und besitzen den Nachteil, dass der vorhandene Texturspeicher der Grafikkarte die Größe des Volumens limitiert und die Darstellungsgeschwindigkeit von der Größe des Volumens stärker abhängt, als andere Verfahren.

Oberflächenrendering visualisiert ein Oberflächenmodell üblicherweise approximierend mittels geometrischer Primitiven wie einem Polygonnetz bzw. Drahtgittermodell [159]. Die Darstellung der Oberflächen kann mittels verschiedener Ansätze wie Z-Buffer-Shading und Gray-Level-Shading erreicht werden. Durch die Entwicklung hochperformanter und spezialisierter Grafikhardware in den letzten Jahrzehnten kann die Visualisierung von Oberflächendaten deutlich flüssiger angeboten werden. Die Performance ist von der Komplexität der Szene abhängig und bietet im Gegensatz zu Volumenrendering nur noch wenig Optimierungspotential ([161], Seite 11–12).

2.5.3 Interaktionstechniken

Visualisierung von Daten wäre ohne eine interaktive Komponente nur halb so effektiv. Interaktionstechniken ermöglichen die intuitive Exploration und Veränderung von Daten. Interaktion soll dabei als Mensch-Rechner-Komunikation verstanden werden, bei der eine zeitlich beschränkte Kommunikation erfolgt. Die verschiedenen Zeitebenen von Interaktionen fassen Card *et al.* ([19], Seite 231) zusammen: Ereignisse innerhalb 0,1 Sekunden

erscheinen aufgrund des psychologischen Moments als ein zusammengehöriges Ereignis und übermitteln einen kausalen Zusammenhang, beispielsweise zwischen dem Auslösen der Aktion und Beobachtung des Resultats. Ereignisse innerhalb von einer Sekunde geschehen zu schnell, um eine Reaktion zu ermöglichen, auch wenn die Pause wahrgenommen wird. Animationen und viele interaktive Aktionen befinden sich oft innerhalb dieser Zeitgrenzen. Die 10 Sekunden Grenze beschreibt die kleinste Einheit kognitiver Arbeit, beispielsweise routinenhafte Interaktionen wie Selektion. Reagiert ein System länger als 10 Sekunden nicht, so erscheint dies nicht mehr als Interaktion und irritiert Anwender. Visualisierungstechniken und Interaktionen werden oft als Werkzeuge zur visuellen Analyse komplexer Daten verwendet (vergleiche Abschnitt 2.5, Seite 22). Verschiedene Interaktionstechniken sind miteinander kombinierbar und sollen im Folgenden vorgestellt werden (siehe hierzu auch Baeza-Yates *et al.* [6]).

Brushing und Linked Views *Brushing* ist der Prozess des Auswählens von Datenpunkten, um diese hervorzuheben [91]. Dieses Hervorheben lenkt die Aufmerksamkeit des Anwenders auf bestimmte Aspekte der Daten und kann durch vielfältige Mechanismen wie Farb-, Form- und Größenänderung realisiert werden. Die Selektion wird üblicherweise direkt durch den Anwender per Maus oder Tastatur vorgenommen. Existieren mehrere Sichten auf die Daten, so wird die Selektion in allen Sichten nachvollzogen und erlaubt somit, die ausgewählten Daten unter verschiedenen Gesichtspunkten zu betrachten.

Panning und Zooming Ist die Visualisierung zu groß für die gesamte Darstellung, wird nur ein Ausschnitt gezeigt, der in alle Richtungen interaktiv verschoben werden kann (*Panning*). Durch Vergrößern und Verkleinern des Ausschnittes kann außerdem die Menge der dargestellten Daten angepasst werden, um beispielsweise den Fokus auf bestimmte Daten zu richten (*Zoomen*). Ändert sich die Zahl der visualisierten Informationen abhängig vom Zoomfaktor, so spricht man von *semantischem Zoom*, der in etwa auch der natürlichen Beobachtungsgabe des Menschen entspricht [10]. Wird Information interaktiv durch den Anwender zur Darstellung hinzugefügt, so spricht man von *Detail-on-Demand*.

Fokus+Kontext und Linsen *Fokus+Kontext* werden genutzt, um ein Problem zu lösen, welches beim Zoomen auftritt. Zoomt man an Daten heran, so sieht man weniger von den umgebenden Daten. Fokus+Kontext vergrößert hingegen die Sicht im Fokus des Anwenders und verkleinert die Darstellung der umgebenden Daten. Je weiter die Daten entfernt sind, desto kleiner werden diese an den Randbereichen, ähnlich dem Fischaugeneffekt, dargestellt. Diese Effekte können auch mittels *Magischer Linsen* (magic lenses) erreicht werden. Linsen sind Objekte, welche das Aussehen oder das Verhalten von Objekten verändern, wenn man durch sie hindurchschaut. Da Linsen verschiedene Änderungen hervorrufen können, ist die Kombination von Linsen hilfreich, um Verbindungen zwischen Daten zu aufzudecken [10].

Overview+Detail Sind Objekte in mehreren Sichten, beispielsweise einer Überblicksicht und einer Detailsicht dargestellt, so spricht man hier von *Overview+Detail*. Es ist dabei

sinnvoll, verschiedenste Interaktionstechniken zu kombinieren, wie z. B. Panning, Zoomen und Fokus+Kontext. Der Anwender kann sich einen Überblick über alle Daten in einer Sicht verschaffen und dann direkt zu einer detaillierteren Darstellung einer Auswahl von Daten wechseln [75].

Animation *Animation* ist eine Technik, um die Illusion einer Bewegung durch Darstellung individueller Schritte einer dynamischen Szene zu erzeugen [150]. Sie wird oft für die Darstellung des zeitlichen Bezuges innerhalb einer Szene benutzt. Animationen können allerdings nicht gedruckt werden und es ist schwierig, weit entfernte Zeitpunkte zu vergleichen. Animationen benötigen einen hohen Implementierungsaufwand, da sie auf den jeweiligen Anwender und dessen Interaktionen abgestimmt sein müssen ([75], Seite 31). Aus diesen Gründen werden Animationen, außer zur Exploration und Navigation, üblicherweise kaum verwendet. Eine oft genutzte Animation ist das flüssige Heranzoomen zu einem Teilgebiet, wodurch der Anwender während der Animation einen Eindruck von der Umgebung des vergrößerten Gebietes erhält.

3

Vorüberlegung

3.1 Anforderungsanalyse

Die Anforderungsanalyse erfolgte durch persönliche Gespräche mit Wissenschaftlern, die Daten erheben und Interesse an einer informationstechnischen Lösung der Datenintegration, -kombination und -visualisierung haben. Hauptsächlich waren Biowissenschaftler aus dem IPK Gatersleben und Mitglieder im Verbundprojekt „GABI-SysSEED“ involviert.

3.1.1 Datenintegration

Die Aufgabe der Anwendung ist, möglichst alle im Arbeitsfluss anfallenden Daten integrieren zu können, ungeachtet der Datentypen, Quelle der Daten, Experiment-ausführender Person und angewandter Methodik zur Erhebung der Daten. Dies bedeutet, dass die Daten aus verschiedenen Domänen stammen können, insbesondere konzeptionelle, räumliche und zeitliche Daten (vergleiche Abschnitt 4.1.1, Seite 41). Auch sollen unterschiedliche biologische Auflösungsebenen unterstützt werden, wie Organell-, Zell-, Gewebe- oder Organebene. Die Art der angewandten biologischen Protokolle sollte irrelevant sein. Zusätzlich ist der Zugriff auf ausgewählte Ressourcen, wie beispielsweise MetaCrop [50], KEGG [70] und DBE [95] gewünscht, um eigene Daten bei Bedarf durch Daten aus diesen Quellen ergänzen zu können. Auch die Integration der Daten von Kooperationspartnern sollte möglich sein.

3.1.2 Datentypen und -merkmale

Typische im Arbeitsfluss anfallende Daten kommen aus verschiedensten Datendomänen (siehe Abschnitt 2.2, Seite 8) und wurden von den Anwendern wie folgt spezifiziert: Genexpressionsdaten, Proteinaktivitäten, Metabolitkonzentrationen, Flussdaten, genregulatorische Netzwerke, PPI-Netzwerke, metabolische Netzwerke, stöchiometrische und kinetische

Modelle, Mikroskopbilder und Fotografien, Volumendaten und Oberflächenmodelle. Diese Datentypen decken einen großen Teil der in der biologischen Forschung auftretenden Daten ab.

Ein wichtige Einschränkung ist, dass vor allem vorprozessierte Daten integriert werden sollen. Dies bedeutet, dass die anfallenden Daten üblicherweise schon normalisiert und gefiltert sind. Werden Daten aus verschiedenen Quellen kombiniert, müssen vom Anwender des Systems die angewandten Vorprozessierungsschritte berücksichtigt werden. Die Experiment-Rohdaten und weitergehende Informationen über die Experimentprozedur, um beispielsweise das Experiment wiederholen zu können, sollen nicht gehandhabt werden.

3.1.3 Datensicherheit

Aufgrund sowohl zeitlicher, als auch monetär aufwändiger biologischer Experimente existiert in den Biowissenschaften ein große Vorsicht bei der Veröffentlichung von Daten. Dringen Informationen über die durchgeführten Prozeduren und Methoden an die Öffentlichkeit, können Wochen und Monate an Arbeit im Sinne der Publizierungswürdigkeit zunichte gemacht werden. In den meisten Bereichen der Biologie ist die Veröffentlichung der Primärdaten auch noch nicht Bestandteil des Publikationsprozesses [79]. Weiterhin gibt es in Zusammenarbeiten mit Industriepartnern oft Geheimhaltungsklauseln, die das Speichern von Daten auf entfernten Informationssystemen untersagen.

Aus diesen Gründen besteht der Wunsch nach einer Anwendung, welche Daten lokal integrieren und auswerten kann. Jegliche Herausgabe von Daten darf nur kontrolliert und manuell durchgeführt werden. Insbesondere das Hochladen von Daten zu entfernten Speicherorten wie Datenbankservern sollte vermieden werden. Dennoch ist es erforderlich, Teilaspekte wie Visualisierungen und Berechnungen für Webseiten und wissenschaftliche Publikationen zu exportieren. Ein weiterer Aspekt der Datensicherheit ist die mögliche Konservierung von erzeugten Resultaten: Sind befriedigende Ergebnisse der Integration, Kombination und Visualisierung erreicht, so müssen diese zu einem späteren Zeitpunkt oder auf einem fremden Rechner in exakt derselben Form wiederherstellbar sein. Auch das Wiederaufgreifen alter Ergebnisse sollte möglich sein, um Analysen beispielsweise unter Hinzunahme neuer Daten nach einigen Jahren fortführen zu können. Grund hierfür ist, dass Wissen aus Hochdurchsatz-Daten oftmals nicht schon komplett bei der ersten Publikation vollständig erfasst werden kann [7].

3.1.4 Kombination der Daten

Die erhobenen Daten können kaum einzeln die komplexen Strukturen und Zusammenhänge erklären. Durch Analyse von Datenkombinationen können mehr Erkenntnisse erlangt werden, als aus den einzelnen Daten an sich [7]. Aus diesem Grund soll die möglichst freie Kombination von Daten es erlauben, komplexe Sichten auf das biologische System zu generieren. Hervorgehobene Anforderungen sind die Kombination konzeptioneller und physischer Information (z. B. biologische Netzwerke mit funktionellen Bilddaten).

Interessant sind weiterhin Vergleiche zwischen biologischen Systemen, die sich in Details unterscheiden, beispielsweise wildtypische und genetisch veränderte Systeme. Auch der Vergleich verschiedener Entwicklungsschritte oder Daten unterschiedlicher Segmente ist hilfreich. Ergebnisse dieser Art sind oft numerischer Art (Relationen und absolute Unterschiede), aber auch Visualisierung von räumlichen und zeitlichen Gradienten zwischen diesen Experimentfaktoren (insbesondere Entwicklungsmuster) stoßen auf großes Interesse.

3.1.5 Visualisierung und Analyse der Daten

Visualisierungen kombinierter Daten können das Verständnis biologischer Zusammenhänge stark vereinfachen. Verschiedene Interaktionstechniken sollen es ermöglichen, die Darstellung zu manipulieren und damit auf individuelle Wünsche und Denkansätze einzugehen. Datenmanipulationen sind von besonderem Interesse, insbesondere Farb- und Transparenzänderungen, aber auch Eingrenzungen der Wertebereiche („Abschneiden“) und Selektion bzw. Hervorheben bestimmter Informationen. Ein oft erwähntes Beispiel ist die Auswahl segmentierter Bereiche in räumlichen Modellen und die synchrone Darstellung Segment-spezifischer Information. All diese Techniken der visuellen Analyse sollen das Verständnis komplexer Datenkombinationen erleichtern und die Hypothesengenerierung im Wissenszyklus unterstützen.

3.1.6 Anwenderfreundlichkeit

Als selbstverständliche Anforderung wird die Barrierefreiheit, insbesondere leichte Handhabbarkeit der Anwendung, betrachtet. Bezeichnungen müssen möglichst aus dem biologischen Kontext entnommen und nicht durch technische Begriffe motiviert werden. Die Eingabe von textlichen Beschreibungen sollte möglichst frei möglich sein, da jeder Anwender eigene Bezeichnungen für Substanzen, Spezies und Ähnlichem bevorzugt. Die Installation der Anwendung muss auf allen handelsüblichen PCs und Betriebssystemen auch für technisch weniger versierte Anwender möglich sein. Gewünscht wird die Entwicklung eines freien, kostenlosen Systems in einer bekannten Programmiersprache wie Java oder C++. Insbesondere die Visualisierung und Datenhaltung sollte nur wenige Anforderungen an die Hardware stellen und somit moderate Anschaffungskosten ermöglichen.

3.2 Bestehende Ansätze und Anwendungen

3.2.1 Generelle Ansätze zur Integration, Kombination und Visualisierung

Sollen Daten aus verschiedenen Datendomänen miteinander kombiniert werden, müssen auch Daten aus unterschiedlichen Quellen berücksichtigt werden, da üblicherweise einzelne Arbeitsgruppen nicht alle verschiedenen Daten erheben (können). Das Vorhandensein solcher Ressourcen, basierend auf der Veröffentlichung von Daten, ist deswegen eine

Grundvoraussetzung und in einigen Gebieten wie beispielsweise Genomics und Proteomics schon gut umgesetzt [79]. Aufgrund der geringen Quantität und hohen Komplexität von Daten anderer Gebiete, wie beispielsweise Bild- oder Neurosciencedaten, sind diese Entwicklungen dort noch nicht so weit vorangeschritten [80]. Hinzu kommen vielfältige soziale Aspekte, weswegen Daten nicht mitveröffentlicht werden. Koslow [80] zählt einige Argumente auf, weswegen Datenveröffentlichung scheinbar „Schmerzen verursachen“ [58]. Oft wird die Veröffentlichung der Daten bisher auch einfach nicht erzwungen [37]. Das liegt daran, dass im wissenschaftlichen Peer-Review-Prozess nur die Schriftstücke an sich begutachtet werden, aber die Überprüfung der Daten üblicherweise von fachunkundigem Personal übernommen wird [101].

Basierend auf diesen Fakten kann nicht davon ausgegangen werden, dass alle zur Integration und Kombination benötigten Daten frei verfügbar sind und durch (Meta-) Datenbanken integriert und kombiniert werden können. Hinzu kommt, dass zwar viele Datenbank-orientierte Integrationsansätze existieren (vergleiche [145]), der Hauptfokus hier aber üblicherweise auf dem Datenaustausch bzw. der Datenintegration in Form von Remote-Datenhaltung großer Datenmengen liegt. Laut den Anforderungen liegt der Fokus aber auf einer kleinen Menge von Daten, welche flexibel und vor allem lokal miteinander kombiniert werden sollen (vergleiche Abschnitt 3.1.3, Seite 32). Biologische Anwender sind darüber hinaus mit der enormen Zahl spezialisierter Datenbanken, den stark zunehmenden Datenmengen und unhandlichen Zugangsmethoden vieler Webinterfaces überfordert. So ist es beispielsweise immer noch sehr aufwändig, eigene Daten korrekt zu annotieren und in Datenbanken zu hochzuladen [7].

Neben Datenbank-basierten Ansätzen existieren unzählige Anwendungen, die einen Datentyp detailliert analysieren und visualisieren können, aber aufgrund des fehlenden Datenintegrations- und Datenkombinationsaspektes in dieser Arbeit nicht relevant sind. Sollen Daten aus zwei oder mehr Domänen integriert, kombiniert und visualisiert werden, so reduziert sich das Feld der verfügbaren Anwendungen stark: Es existieren einige Anwendungen zur Kombination numerischer Werte und Netzwerke (für eine Übersicht siehe [46] und [75], Seite 75). Die Kombination von 2D-Bildern oder 3D-Daten zusammen mit numerischen Werten ist größtenteils nur in der Gehirnforschung populär [54, 98, 117]. 2D- und 3D-Kombination wiederum wird in sehr vielen Anwendungen realisiert (beispielsweise [1, 11, 85, 106, 111, 140]). Die Integration von Netzwerken und 2D- oder 3D-Daten ist derzeit nur durch eine Anwendung abgedeckt [139]. Einige repräsentative Anwendungen sind in Abbildung 3.1 dargestellt. Eine detaillierte Analyse verschiedener Systeme ist im folgenden Abschnitt zu finden. Deutlich wird aus der Abbildung, dass eine Häufung von Anwendungen existiert, welche entweder 2D- und 3D-Daten (exklusiv) oder numerische Werte und Netzwerke kombinieren. Dies stützt die Ausführungen in Abschnitt 2.5.1, Seite 22, in dem die zwei Gebiete wissenschaftliche Visualisierung und Informationsvisualisierung als oft voneinander unabhängig betrachtet dargestellt werden.

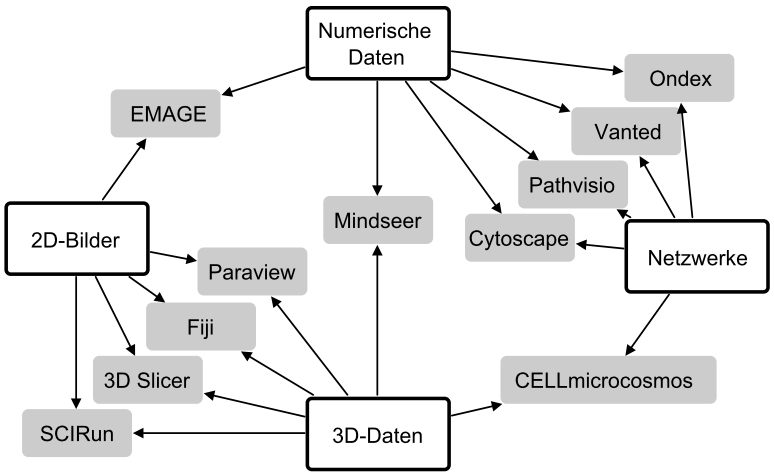


Abbildung 3.1: Darstellung der vier relevanten Datendomänen (weiße Kästen) und einiger ausgewählter Anwendungen (graue Kästen), welche mindestens zwei Datentypen integrieren können (Kanten). Nur angedeutet ist die Vielzahl der verfügbaren 2D-3D-Datenintegrationen bzw. numerische Werte-Netzwerke-Integrationen, gegenüber den anderen. In Tabelle 3.1 ist ein detaillierter Vergleich dieser und weiterer Anwendungen zu finden.

3.2.2 Vergleich von Anwendungen zur Integration, Kombination und Visualisierung

Im Folgenden sollen einige derzeit verfügbare Anwendungen verglichen werden, um die Erfüllung der Anforderungen aus Abschnitt 3.1, Seite 31 abschätzen zu können. In Tabelle 3.1 sind einige repräsentative Anwendungen aufgelistet, welche zumindest einen großen Teil der Anforderungen erfüllen.

3.2.2.1 Auswahl der Vergleichsanwendungen

Aufgrund der Vielzahl verfügbarer Anwendungen kann der Vergleich nur anhand einer kleinen Menge repräsentativer Anwendungen erfolgen. Alle Anwendungen, die nur einen Datentyp berücksichtigen, werden wegen des fehlenden Datenintegrations- und Kombinationsaspektes nicht berücksichtigt. Wie schon in Abbildung 3.1 angedeutet, konnte keine Anwendung gefunden werden, welche mehr als zwei Datentypen kombinieren kann. Weitere Ausschlusskriterien für die übrigen Anwendungen sind:

- Großteil der Anforderungen in Abschnitt 3.1, Seite 31 wird nicht erfüllt
- nicht verfügbar, veraltet oder nicht auf aktueller Hardware lauffähig
- Anwendung ist kommerziell (Ausnahme Amira, da es ungeachtet der hohen Kosten in der biologischen Forschung sehr verbreitet ist)
- Datenbank-basierte Datenintegration, da diese nicht die Lokalitätsanforderung in Abschnitt 3.1.3, Seite 32 erfüllen; Beispielanwendungen sind [4, 84, 137, 154, 158]

Anwendung	Vanted [67]	Ondex [77]	Cytoscape [138]	PathVisio [62]	Mindseer [98]	EMAGE [117]	Fiji [130]	MBAT [11]	ParaView [85]	SCIRun [106]	3D Slicer [111]	Amira [140]	CELLmicrocosmos [139]
numerische Werte	***	**	*	***	*	*	–	–	–	–	–	–	–
2D-Bilder	–	–	–	–	–	***	***	**	**	***	**	***	–
3D-Daten	–	–	–	–	***	–	***	***	***	***	***	***	***
biol. Netzwerke	***	***	***	***	–	–	–	*	–	–	–	–	**
Manipulation	***	***	***	**	*	*	***	**	***	***	***	***	*
Kombination	***	***	***	**	*	*	**	**	***	***	**	***	**
Metadatensupport	**	***	*	*	*	*	–	***	–	–	–	–	**
Ressourcen	***	***	***	**	*	**	*	***	*	*	**	–	**
Erweiterbarkeit	**	**	***	**	*	–	***	**	**	*	*	**	–
Bedienbarkeit	***	***	***	***	***	***	***	**	**	**	***	**	**
Lizenz	OS	OS[1]	OS	OS	OS	NC[2]	OS	NC	OS	OS	OS	$[3]	–
Plattform	Java[4]	Java	Java	Java	Java	Java	Java	Java	WLM[5]	WLM	WLM	WLM	Java

Tabelle 3.1: Vergleich verschiedener Anwendungen, welche einen großen Teil der Anforderungen aus Abschnitt 3.1, Seite 31 erfüllen. Je mehr Sterne, desto besser wird eine Funktionalität unterstützt (– = gar nicht, * = kaum, ** = gut, *** = sehr gut). Detaillierte Erläuterungen zu den einzelnen Punkten und Auswahl der Anwendungen sind im Text auf Seite 38 zu finden. [1] = Open Source Lizenz, [2] = nicht-kommerzielle Nutzung kostenlos, [3] = kommerziell, [4] = alle Plattformen, auf denen Java läuft, [5] = Windows, Linux und Mac.

Darüber hinaus wurde für die jeweilige Anwendung nur die Kernfunktionalität betrachtet, nicht jedoch potentiell verfügbare Erweiterungen. Dies hätte einen enormen Mehraufwand beim Testen und der Recherche bedeutet. Außerdem entspricht die Qualität der Erweiterungen oft nicht der der Hauptanwendung.

Trotz der Ausschlusskriterien existiert insbesondere für die Integration und Kombination numerischer Werte mit biologischen Netzwerken, bzw. der Integration von 2D-Bildern mit 3D-Daten eine große Menge von Anwendungen. Aus diesen musste eine möglichst repräsentative Auswahl von in der biologischen Forschung bekannten Anwendungen getroffen werden:

- basierend auf den Vergleichen von Klukas ([75], Seite 75) und Gehlenborg *et al.* [46] wurden folgende Anwendungen zur Kombination numerischer Werte und Netzwerke ausgewählt:
  - VANTED: flexible Integration komplexer numerischer Werte in den Kontext biologischer Netzwerke und Klassifikationshierachien mit vielen Statistikfunktionen
  - ONDEX: starker Fokus auf der Erstellung von Netzwerken, basierend auf der Verknüpfung verschiedener Datenbanken und mit einfachen Möglichkeiten zur Kombination der Netzwerke mit numerischen Werten
  - CYTOSCAPE: mächtige und bekannte Anwendung zur Bearbeitung biologischer Netzwerke und Graphen, Kombination mit numerischen Werten ist nur rudimentär vorhanden und kann durch Plugins erweitert werden
  - PATHVISIO: unterstützt einfache Kombination von Mikroarray- und Proteomicsdaten im Kontext biologischer Netzwerke
- MINDSEER: typischer Stellvertreter vieler Anwendungen der hauptsächlich in der Medizin relevanten Kombination numerischer Werte und 3D-Daten (z. B. strukturelle Gehirndaten)
- EMAGE: typischer Stellvertreter einiger weniger Anwendungen, die üblicherweise (*in situ*-)Bilder und numerische Werte (Genexpressionsdaten) verknüpfen
- alle 2D-Bilder und 3D-Daten kombinierenden Anwendungen, die von Walter *et al.* ([157], Tabelle 1 und 2) als wichtig gekennzeichnet sind. Ausgenommen sind kommerzielle Anwendungen, Bibliotheken oder auf wenige Plattformen beschränkte Anwendungen:
  - FIJI: sehr flexible und verbreitete 2D-Bildverarbeitungsanwendung inklusive 3D-Visualisierung basierend auf IMAGEJ, mit Fokus auf elektronenmikroskopische Bilder
  - MBAT: Erstellung von Atlanten mittels komplexer Arbeitsflüsse auf Basis verschiedener Datenbanken, inklusive Registrierung und 3D-Visualisierung

  - Paraview: Anwendung zur Analyse und Visualisierung üblicherweise technischer Daten mit Fokus auf große Datensätze
  - SCIRun: „Problemlösungs“-Anwendung zur Modellierung, Simulation und Visualisierung wissenschaftlicher Probleme mit ungewöhnlichem Graph-basierten Bedienkonzept
  - 3D Slicer: Anwendung zur Aufbereitung und Visualisierung medizinischer Daten mit Fokus auf der Erstellung von Atlanten
  - Amira: verbreitete und mächtige kommerzielle Anwendung zur Arbeit mit Volumen und Oberflächendaten, insbesondere Segmentierung und Visualisierung
  - BioImage Suite: nicht berücksichtigt wegen thematischer Überlappung mit den anderen Anwendungen

- CellMicrocosmos als einzig verfügbare Anwendung, die biologische Netzwerke und 3D-Daten kombiniert

3.2.2.2 Vergleichskriterien

Verglichen werden können die ausgewählten Anwendungen über eine Reihe von Eigenschaften, welche in den Anforderungen in Abschnitt 3.1, Seite 31 beschrieben sind. Sehr wichtige Punkte sind die Unterstützung der verschiedenen Datentypen. Weitere Kriterien sind:

Manipulation beschreibt die Quantität und Mächtigkeit der Interaktions- und Visualisierungsmöglichkeiten.

Kombination ist die Funktionalität, jeden Datenwert mit jeden anderen beliebig, flexibel und immer wieder kombinieren zu können. Mindseer und EMAGE bieten nur wenige, vorgefertigte Kombinationsmöglichkeiten.

Metadatensupport ist die Fähigkeit, Metadaten (visuell) aufzubereiten und für Suche, Sortierung und Filteroperationen nutzen zu können. * = Metadaten werden interpretiert und üblicherweise textlich dargestellt, ** = Teile der Metadaten sind durchsuchbar oder für Sortierung nutzbar, *** = Metadaten werden visuell aufbereitet oder sind flexibel nutzbar.

Ressourcen beschreibt die Verfügbarkeit von Beispieldaten und Qualität des eingebauten direkten Zugriffs auf (Beispiel-)Datenbanken.

Erweiterbarkeit beschreibt die Verfügbarkeit und Qualität von Erweiterungen (Plugins, Add-ons bzw. Module), die die Funktionalität der Hauptanwendung möglichst tiefgreifend erweitern können. * = Erweiterungen vorgesehen, aber kaum oder gar keine verfügbar, ** = wenige hochwertige Erweiterungen vorhanden, *** = sehr viele und hochwertige Erweiterungen verfügbar.

Bedienbarkeit ist die Eigenschaft, leicht installierbar zu sein und ein intuitives Bedienkonzept zu besitzen. 3D-Visualisierungen weisen aufgrund ungenügender Eingabegeräte für drei Dimensionen und hohen Hardwareanforderungen üblicherweise schlechtere Bedienbarkeit als 2D-Visualisierungen auf. SCIRun und Amira nutzen eine ungewöhnliche Graph-basierte Bedienung.

Lizenz beschreibt die Verfügbarkeit der Anwendung, OS = Open Source, NC = nichtkommerzielle Nutzung frei, aber nicht der Source Code, $ = kommerziell).

Plattform beschreibt die Betriebsysteme, für die die Anwendung verfügbar ist. W = Windows; L = Linux; M = Mac; Java = Plattformen, für die Java verfügbar ist.

Die Vergleiche wurden anhand von Publikationen, Webseiten und Benutzung der Anwendung auf Basis von Tutorials mit Beispieldaten gezogen.

3.2.2.3 Auswertung

Es existieren derzeit keine Anwendungen, welche alle Typen multimodaler Daten integriert kombinieren und visualisieren können. Die Netzwerkanwendungen zeichnen sich vor allem durch gute Bedienbarkeit und Unterstützung von Metadaten aus. Mindseer, EMAGE und CELLmicrocosmos sind primär als Betrachtungs-Anwendungen entwickelt wurden und weisen deshalb relativ wenige Kombinationsmöglichkeiten und Interaktions- bzw. Visualisierungstechniken auf. Obwohl fast alle Anwendungen Erweiterungen unterstützen, existieren für die meisten nur wenige hochwertige (und verfügbare) Erweiterungen. Erfreulich ist hingegen, dass die Quellen der meisten Anwendungen frei verfügbar und dank Java auf allen Betriebssystemen auch leicht installierbar sind.

3.3 Fazit

Resultierend aus der multi-personellen, multi-methodischen Natur der biologischen Datenakquisition wird eine Anwendung gewünscht, welche die beschriebenen multimodalen Daten lokal integrieren, kombinieren und visualisieren kann. Es wird ein Ansatz benötigt, welcher die biologische Forschung bei der Auswertung der eigenen und in Datenbanken verfügbaren Daten in der Hypothese-Phase unterstützt (vergleiche Abschnitt 2.2, Seite 7). In der Literatur finden sich keine Anwendung, welche drei oder gar vier Datentypen integrieren, flexibel kombinieren und vielfältig visualisieren kann. Aus diesem Grund wird im Folgenden eine Methodik vorgestellt, die möglichst alle genannten Kriterien erfüllen soll.

4

Methodik

Zur Erfüllung der im vorigen Kapitel beschriebenen Anforderungen wird in diesem Kapitel eine Methodik vorgestellt, welche multimodale Daten integrieren, flexibel kombinieren und vielfältig visualisieren kann. Die Methodik ist als Visualisierungspipeline strukturiert, welche in Abbildung 4.1 dargestellt ist. Die folgenden Abschnitte stellen jeweils einen Transformationsschritt vor, ausgehend von multimodalen Daten.

4.1 Multimodale Daten

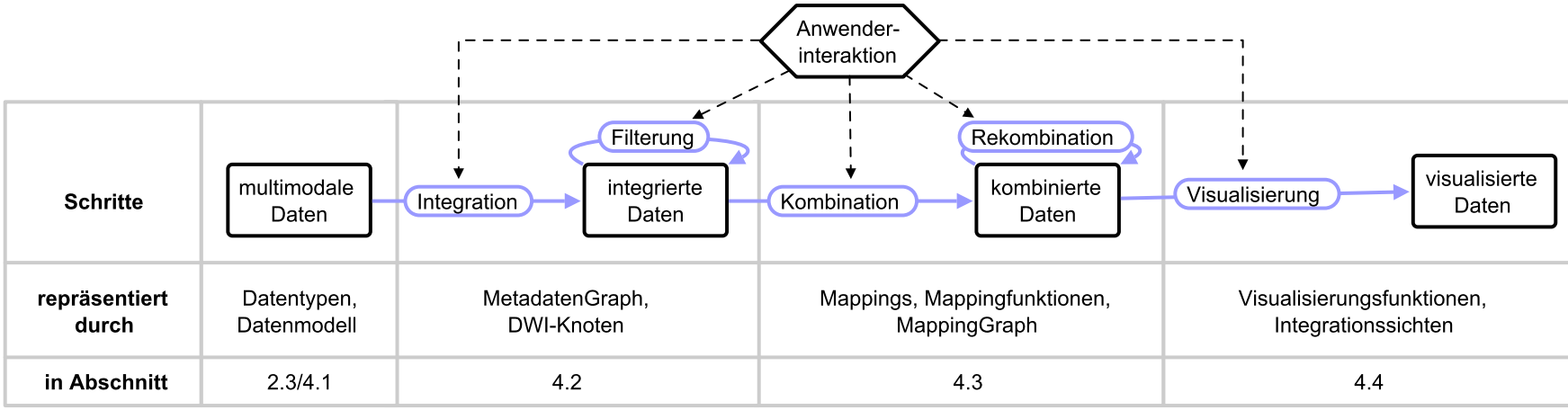
Die Erhebung sowie Eigenschaften und einige Beispiele multimodaler Daten wurden bereits ausführlich in Abschnitt 2.2, Seite 8 beschrieben. Die Struktur dieser Daten soll im Folgenden analysiert und in Form eines Datenmodells beschrieben werden.

4.1.1 Struktur biologischer Experimentdaten

Wissenschaftliche Fragen werden in der Biologie üblicherweise auf Basis von Experimenten beantwortet (vergleiche Abschnitt 2.1.3, Seite 6). Der Fokus liegt entsprechend der Anforderungsanalyse in Abschnitt 3.1, Seite 31 auf den experimentellen Ergebnissen in Form vorprozessierter Daten. Welche Eigenschaften solche Experimente aufweisen, soll im Folgenden exemplarisch anhand von drei Experimenten vorgestellt werden.

Experiment 1 *Die Frage, wie sich die Freigabe von photosynthetischem Sauerstoff auf Lipidsynthese insbesondere unter Berücksichtigung der Substanzverteilungen in den Geweben auswirkt, untersuchten Rolletschek* et al. *[122]. Dazu wurden Sojabohnen-Pflanzen unter Gewächshausbedingungen herangezogen und Teile der Pflanze verschiedenen Sauerstoffkonzentrationen ausgesetzt. Die Ernte erfolgte in verschiedenen Entwicklungsstadien. Ein Teil der Datenwerte sollen im Folgenden kurz umrissen werden:*

- *Messung der Lichtstärke in Form numerischer Werte der Strahlungsintensität*



Schritte	multimodale Daten	Integration	integrierte Daten	Kombination	kombinierte Daten	Visualisierung	visualisierte Daten
repräsentiert durch	Datentypen, Datenmodell	MetadatenGraph, DWI-Knoten		Mappings, Mappingfunktionen, MappingGraph		Visualisierungsfunktionen, Integrationssichten	
in Abschnitt	2.3/4.1	4.2		4.3		4.4	

Abbildung 4.1: Die Struktur des Methodik-Kapitels in Form einer Visualisierungspipeline. Dargestellt sind ausgehend von den multimodalen Daten drei Transformationsschritte *Integration*, *Kombination* und *Visualisierung*, welche entsprechend integrierte Daten, kombinierte Daten und visualisierte Daten erzeugen. Der Anwender kann jeden Transformationsschritt vielfältig manipulieren und beeinflussen. Die für die jeweiligen Daten und Transformationen zuständigen Strukturen und Funktionen sind genauso angegeben, wie die Nummerierung des beschreibenden Abschnittes.

- *Messung der internen Sauerstoffkonzentration in Form numerischer Werte*
- *über gesamte Sojabohne gemittelte AMP, ADP und ATP Konzentration unter verschiedenen Umweltbedingungen (Sauerstoffkonzentrationen) in Form numerischer Werte*
- *zweidimensionale Messung der ATP-Konzentration als Bild, indem jeder Bildpunkt die Konzentration von ATP an dieser Stelle des Gewebeschnittes repräsentiert*
- *Metabolitkonzentration als numerische Werte mit biologischen Replikaten*
- *biochemisches Netzwerk der Glykolyse und des Zitronensäurezyklus verknüpft mit Metabolitkonzentrationen, die als Balkendiagramme visualisiert werden*

Insgesamt stehen in dieser Studie also die verschiedenen Datentypen im Vordergrund, die erhoben oder weiterverwendet wurden, um die biologische Hypothese zu unterstützen. Die Interpretation der Daten ist somit, neben der Erhebung der Datenwerte, Herzstück der Publikation.

Experiment 2 *Ziel der Studie von Glidewell [48] ist es, mit bildgebender NMR-Technik die dreidimensionale Anatomie und Wasserverteilung des Gerstenkorns über den gesamten Entwicklungszeitraum zu messen, um die für die Brauindustrie wichtige Wasserverteilung in den Körnern zu ermitteln. Insbesondere die geringe Invasivität des NMR Verfahrens ermöglicht realitätsnahe Strukturaufklärung. Gerstepflanzen wurden unter normalen Gewächshausbedingungen herangezogen und zum entsprechenden Zeitpunkt geerntet.*

- *dreidimensionale Volumen der Wasser-, Zucker- und Lipidkonzentration im unveränderten Gerstenkorn in den Entwicklungsstadien 0, 1, 2, 4, 6, 8, 12, 16, 21, 30 und 40 Tage nach der Befruchtung*

In dieser Studie steht also die zeitliche Entwicklung des biologischen Objektes im Vordergrund, wobei der Fokus stark auf räumliche Datenwerte und die Interpretation struktureller Eigenschaften des Korns gelegt wurde.

Experiment 3 *McGrail* et al. *[93] untersuchten anhand von Zebrafischembryonen die Eigenschaften zweier Genklassen, welche wichtigen Einfluss auf die Funktionalität von Photorezeptorzellen und das zentrale Nervensystem haben. Dazu wurden drei sich in der genetischen Basis unterscheidende Zebrafischstämme herangezogen und zu verschiedenen Stunden nach der Befruchtung beobachtet. Die Messungen umfassten:*

- *Genexpression als relative numerische Werte der Fischstämme zu den Stunden 1, 4, 12, 24, 48, 96, 120 nach der Befruchtung*
- *in situ-Hybridisierungen in Form von Fotografien/Bildern zu den Stunden 2, 24, 36, 48, 72 und 120 nach der Befruchtung*
- *in situ-Hybridisierungen in Form von (auf Zellebene aufgelösten) Fotografien/Bildern adulter mutierter Zebrafische, in denen die Genexpression lila angefärbt wurde*

Basis der Studie sind somit die Messungen der relativen Genexpressionsdaten verschiedener Genklassen, die sowohl zeitlich, als auch räumlich aufgelöst sind. Die räumliche Auflösung ist in einem Fall gewebespezifisch, ansonsten Bild-basiert. Der Fokus der Studie liegt somit auf der räumlichen Verteilung der Genexpressionsdaten.

Die vorgestellten Experimente heben verschiedene Aspekte der Datenakquisition hervor, beispielsweise erhebt Experiment 1 sehr verschiedenartige Daten, Experiment 2 untersucht Eigenschaften der zeitlichen Entwicklung und Experiment 3 verschiedene Mutanten. Obwohl diese Experimente unterschiedliche Organismen beschreiben, sind solche Datensätze auch für einzelne Organismen denkbar und in Teilen vorhanden, die es in der vorgestellten Methodik zu integrieren gilt. Um diese dargestellten Ergebnisse erfassen zu können, muss ein Modell geschaffen werden, das, wie in Abschnitt 2.4.1, Seite 20 dargestellt, möglichst einfach gehalten sein soll, aber dennoch alle zur weiteren Verarbeitung nötigen Informationen beinhaltet. Zuerst sollen aber einige in den Experimentbeschreibungen genutzten Begriffe definiert werden:

Definition 1 (*Lebewesen*) *Organisierte genetische Einheit, die zu Stoffwechsel, Fortpflanzung und Evolution befähigt ist [113].*

Der Begriff *Lebewesen* ist also relativ generisch und erfasst somit unter anderem Bakterien, Pflanzen und Säugetiere. In den oben beschriebenen Experimenten sind das Sojabohne (Wachstum unter verschiedenen Sauerstoffkonzentrationen), Gerste und Zebrafisch (Wildtyp und Mutante)

Definition 2 (*Experiment*) *Methodisch angelegte Versuchsanordnung in einer kontrolliertem Umgebung, um Aussagen über die Struktur und Funktionsweise von Lebewesen treffen zu können.*

Diese Definition trifft auf alle hier vorgestellten Experimente zu.

Definition 3 (Entwicklungstadien) *Unterschiedliche Stufen in der Entwicklung eines Lebewesens, die sich durch qualitative Merkmale unterscheiden.*

Entwicklungsstadien beschreiben also den zeitlichen Aspekt des sich entwickelnden Lebens. So berücksichtigt Experiment 1 die Stadien 80mg, 240mg und 400mg, welche direkt in Zeiteinheiten umrechenbar sind. Experiment 2 dagegen umfasst ein breiteres Zeitspektrum (Tag der Befruchtung 0, 1, . . . , 40). Experiment 3 analysiert die Stunden nach der Befruchtung in relativ großen Abständen (1, 4, 12, 24, 48,. . .). Interessant sind aber nicht nur biologische Zeiteinheiten, sondern auch technisch oder logisch bedingte Zeitunterschiede in den Messungen. Beispiele sind die Dauer der Aussetzung bestimmter Umweltbedingungen (beispielsweise Sauerstofferhöhung) oder die Dauer eines Messvorganges.

Definition 4 (*Messung*) *Ausführen von geplanten Tätigkeiten zu einer quantitativen Aussage über eine Messgröße durch Vergleich mit einer Einheit [102].*

In dieser Arbeit soll eine Messung die Entnahme einer Probe vom *Lebewesen* bedeuten. Dies beinhaltet vor allem Informationen über die Entwicklungsstadien, aber auch räumliche Attribute, z. B. wo die Probenentnahme am *Lebewesen* vorgenommen wird.

Definition 5 (*Messgrößen*) *Quantitativ bestimmbare Eigenschaft physikalischer Objekte, der eine Messung gilt [102].*

Messgrößen repräsentieren das biologische Objekte, welches gemessen wird. In Experiment 1 sind die Messgrößen Strahlung, Sauerstoff und ADP/ATP. In Experiment 2 dagegen sind die Messgrößen Wasser-, Zucker- und Lipidkonzentrationen und in Experiment 3 Gene zweier Genklassen.

Definition 6 (Datenwert) *Wert einer Messgröße, der von einem Messgerät geliefert wird.*

Datenwerte sind somit die Ergebnisse von Experimenten, welche neues biologisches Wissen repräsentieren (können). In Experiment 1 sind das numerische Werte der Strahlungsintensität und Konzentration verschiedener Metabolite. Alle Datenwerte in Experiment 2 sind komplexe Volumen und in Experiment 3 Bilder der räumlichen Genexpressionsmuster im Kontext der Organismenstruktur. Datenwerte können darüber hinaus mit geeigneten Methoden aus anderen Datenwerten erzeugt werden (vergleiche Seite 54). Schlussendlich sollen im Folgenden *Experiment*, *Lebewesen* und *Messung* als Metadaten zusammengefasst werden:

Definition 7 (Metadaten) *Beschreibung des Experimentaufbaus, welcher zur Erhebung der Datenwerte geführt hat.*

Metadaten werden auch als *Daten beschreibende Information* definiert ([19], Seite 20). Eine ähnliche Strukturierung der Metadaten in *Experiment*, *Lebewesen* und *Messung* findet man beispielsweise auch in der Anwendung CORNET [35].

4.1.2 Datenmodell

Nach Analyse der Beispiele im vorigen Abschnitt ergab sich die grundlegende Eigenschaft, dass Lebewesen, die sich in bestimmten Entwicklungsstadien befinden, Proben entnommen werden. Die Untersuchungen bestehen aus Datenwerten interessanter Messgrößen, die ein Phänomen angemessen beschreiben. Die Abhängigkeiten sind dabei größtenteils hierarchisch: Experimente betrachten ein oder mehrere Lebewesen, die jeweils zu verschiedenen Zeitpunkten analysiert werden. Die Analyse besteht wiederum aus einer Menge von Datenwerten. Die betrachtete Messgröße kann allerdings durchaus in vielen Experimenten relevant sein. Das Datenmodell ist in Abbildung 4.2 in Form eines UML-Klassendiagramms dargestellt (vergleiche [75, 119]). Jede Klasse, insbesondere Datenwerte, besitzt Attribute wie z. B. Bezeichnung und Größe. Die Kardinalitäten des Klassendiagrammes implizieren, dass jedes *Experiment* nicht zwingend *Lebewesen*, *Lebewesen* nicht zwingend *Messungen* und diese wiederum nicht zwingend Datenwerte enthalten müssen, da in der Praxis oft Messungen oder gar Messreihen aus den verschiedensten Gründen nicht durchgeführt werden (können). Diese Leerstellen können in Visualisierungen beispielsweise durch Platzhalter berücksichtigt werden. Durch die $1..*$ Beziehungen fächern die Instanzen des Klassendiagrammes baumartig auf. Eine Datenschema-Normalisierung [29] zur Vermeidung von

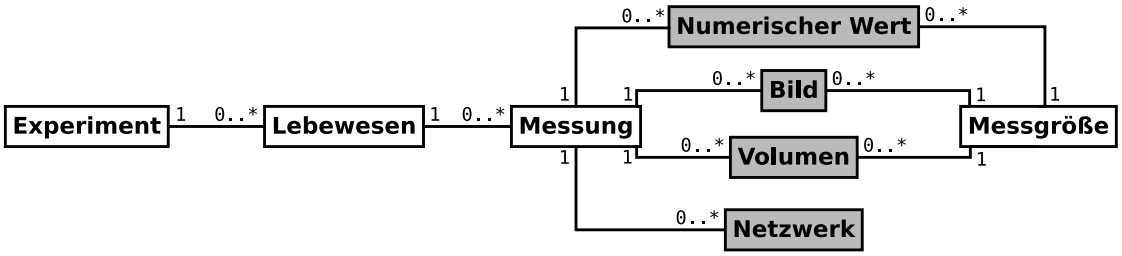


Abbildung 4.2: Klassendiagramm des Datenmodells für biologische Experimentdaten, adaptiert von [75]. Experimente untersuchen *Lebewesen* anhand von *Messungen* (zeitlich und räumlich aufgelöst) durch Erhebung einer Menge von Datenwerten interessanter *Messgrößen*. Die Datenwerte (hier grau gefärbt) repräsentieren dabei neues biologisches Wissen und sind in vier Klassen aufgeteilt.

Redundanzen, wie bei Datenbanken oft anzutreffen, erfolgt hier aus Gründen der Simplizität und Performance nicht.

Alle in Abschnitt 3.1.2, Seite 31 beschriebenen Datenwerte werden mittels vier verschiedener Datenwerttypen *Numerischer Wert*, *Bild*, *Volumen* und *Netzwerk* modelliert (vergleiche Tabelle 4.1). Diese Aufteilung basiert auf den Wünschen biologischer Anwender und entspricht sehr deren Wahrnehmung von multimodalen Daten. Weiterhin ist die Aufteilung auf vier Datentypen ein guter Kompromiss zwischen vielen Datentypen (aufwändiger zu spezifizieren, viele Datentypen werden kaum gemessen und somit selten importiert) und zu wenig Datentypen (verschiedenste Daten würden vermischt). Es ist denkbar, die Daten in zwei Typen aufzuteilen: räumlich (*numerischer Wert*, *Gradient*, *Bild*, *Volumen*) und nicht-räumlich (*Netzwerk*). Solch eine abstrakte Aufteilung legt sehr großes Gewicht auf den Typ der *Netzwerke* und entspricht auch nicht der Wahrnehmung von Biologen, da diese sich eher an der Art und Weise der Datenerhebung, -analyse und -visualisierung orientieren: auch wenn Volumen technisch gesehen Bilderstapel sind, werden Volumen z. B. mit anderen Anwendungen bearbeitet. Weiterhin könnte eine Aufteilung auf der Räumlichkeit von Daten basieren, mit entsprechend fünf Typen (*Numerischer Wert* = 0D, *Gradient* = 1D, *Bild* = 2D, *Volumen* = 3D und *Netzwerk* = nicht-räumlich). *Gradienten* sind allerdings relativ selten auftretende Datentypen. Sie sind zudem sehr leicht durch ein Positions-Attribut als Menge *numerischer Werte* modellierbar. Die Visualisierung erfolgt als einfaches (Linien-)Diagramm, weswegen auf die explizite Modellierung dieses Datenwerttyps verzichtet werden kann.

Fazit

Daten biologischer Experimente besitzen viele Modalitäten, insbesondere zeitliche und räumliche Auflösungsebene, Spezies, Genotypen, Umwelteinflüsse und Datentypen. Alle relevanten Daten werden durch vier verschiedene Datentypen modelliert, wodurch ein großer Teil biologischer Experimentdaten abgedeckt werden kann. Es wurde ein einfaches Datenmodell entwickelt, welches eine Weiterentwicklung des von Klukas [75] vorgestellten

Datenwerttyp	*Numerischer Wert*	*Bild*	*Volumen*	*Netzwerk*	Grund
Genexpressionsdaten	✓				durch Fließkommazahl modellierbar
Proteinaktivitäten	✓				durch Fließkommazahl modellierbar
Metabolitkonzentrationen	✓				durch Fließkommazahl modellierbar
Flussdaten	✓				durch Fließkommazahl modellierbar
metabolische Netzwerke				✓	durch Graphen modellierbar
genregulatorische Netzwerke				✓	durch Graphen modellierbar
PPI-Netzwerke				✓	durch Graphen modellierbar
stöchiometrische Modelle				✓	durch Netzwerke mit zusätzlichen Gesetzen modellierbar (z. B. Petri-Netze [55, 115])
kinetische Modelle				✓	sind stöchiometrische Modelle mit zusätzlichen Informationen über Geschwindigkeitsgleichungen (z. B. mittels kontinuierlicher Petri-Netze modellierbar [2, 21])
Mikroskopbilder		✓			mittels zweidimensionaler Arrays modellierbar, in denen jedes Element die Informationen jedes Bildpixels enthält
Fotografien		✓			mittels zweidimensionaler Arrays modellierbar, in denen jedes Element die Informationen jedes Bildpixels enthält
Volumendaten			✓		sind technisch gesehen dreidimensionale Bilder, benötigen aber andere Ansätze der Visualisierung, Analyse und Anwenderinteraktion; zusätzlich besteht der explizite Wunsch der Anwender, beide Datentypen getrennt zu behandeln
Oberflächenmodelle			✓		spielen oft nur zur darstellenden Visualisierung wie Bildschirmfotos eine Rolle, weniger zu analytischen Zwecken (vergleiche auch Abschnitt 2.2.4, Seite 15); können zu Volumendaten umgerechnet werden

Tabelle 4.1: Zuordnung der in Abschnitt 3.1.2, Seite 31 beschriebenen Daten zu vier modellierenden Datentypen.

Modells darstellt. Es fokussiert hauptsächlich auf vorprozessierte Daten und vernachlässigt weitestgehend Informationen, welche es erlauben würden, Experimente nachzuvollziehen und reproduzieren zu können. Dies steht im Gegensatz zu umfangreichen Datenmodellen wie dem des MIAME Standard [14] (Mikroarray-Daten), des Modells der PEDRo Datenbank [148] (Proteomics-Daten) und das des ArMet Frameworks [127] (Metabolomics-Daten). Somit erfüllt das Datenmodell die Anforderungen aus den Abschnitten 3.1.4, Seite 32 und 3.1.5, Seite 33, da vor allem Informationen zur Visualisierung und Weiterverarbeitung relevant sind. Ein solches Modell ist leichter verständlich, besser erweiterbar und erlaubt eine schnellere Spezifikation der Metadaten, als komplexere Modelle. Durch den Verzicht auf Experimentbeschreibung und Modellierung der Rohdaten sind allerdings weder die Experimente reproduzierbar, noch können Vorprozessierungsschritte korrigiert oder wiederholt werden.

4.2 Integration

Multimodale Daten sollen, basierend auf dem vorgestellten Datenmodell und einer Formalisierung, integriert werden und möglichst einfach zu handhaben sein. Dazu werden zwei Graphstrukturen vorgestellt, welche die Integration auf verschiedenen Ebenen realisieren. Interaktionen erlauben die Explorierbarkeit integrierter Experimentdaten.

4.2.1 Formalisierung multimodaler Daten

Wir definieren die Menge $\mathbb{D}$ als die *Menge aller möglichen Datenwerte*. $\mathbb{D}$ beinhaltet potentiell alle Datenwerttypen in der Biologie aus Abschnitt 2.2, Seite 8. Es existieren vier Datenwerttypen, welche verschiedene Datendomänen modellieren: *Numerische Werte* (Menge $\mathbb{U}$), *Bilder* (Menge $\mathbb{B}$), *Volumen* (Menge $\mathbb{V}$) und *Netzwerke* (Menge $\mathbb{N}$). Ein Datenwert $d \in \mathbb{D}$ ist eindeutig einer der Mengen zugeordnet, das heißt, dass z. B. ein *Netzwerk* nicht gleichzeitig ein *Volumen* sein kann. Es gilt

$$\mathbb{D} = \{d : d \in \mathbb{U} \dot{\vee} \ d \in \mathbb{B} \dot{\vee} \ d \in \mathbb{V} \dot{\vee} \ d \in \mathbb{N}\} \tag{4.1}$$

und somit ist die Menge $\mathbb{D}$ eine disjunkte Mengenfamilie. Entsprechend kann ein Datenwert $d \in \mathbb{D}$ unterschiedliche Arten von Daten modellieren: So ist ein *numerischer Wert* ein relativ einfacher Datenwert, ein *Volumen* dagegen ein dreidimensionales Array und somit deutlich komplexer. Auch abstraktere Informationen wie biologische Netzwerke sind Datenwerte, da diese basierend auf Experimentdaten erstellt werden.

In Abschnitt 4.1.1, Seite 41 sind weitere wichtige Informationen wie Eigenschaften der untersuchten *Lebewesen*, z. B. genetische Veränderungen und Umwelteinflüsse, definiert wurden, welche durch die Menge $\mathbb{A}$ beschrieben werden sollen. Die Metadaten sind dabei in vier Klassen teilbar: Informationen über das biologische *Experiment*, das untersuchte *Lebewesen*, der Zeitpunkt und Ort der *Messung* und die *Messgröße*. Entsprechend definieren wir die *Menge aller möglichen Metadaten* $\mathbb{A}$ wie folgt. Sei $I = \{Experiment, Lebewesen,$

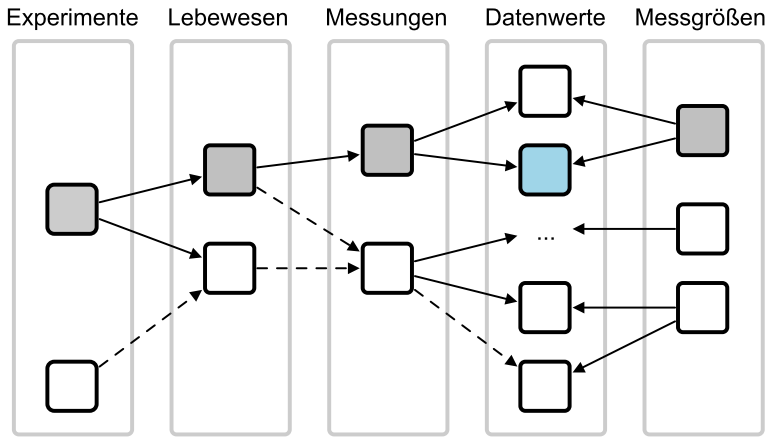


Abbildung 4.3: Beispiel eines MetadatenGraphen, welcher aus vier Schichten für die jeweiligen Metadaten $\mathbb{A}$ und einer Schicht für die Datenwerte $\mathbb{D}$ besteht. Der MetadatenGraph erlaubt es, die Beziehungen innerhalb und zwischen Experimenten auf der Metadatenebene darzustellen. Kanten repräsentieren die $1..n$-Relationen des Datenmodells („von 1 zu n“ zeigend). Gestrichelte Kanten deuten die Anwendung des Algorithmus 2, Seite 51 an. Graue Knoten stellen die Metadaten eines (hier blau gefärbten) Datenwertes d dar ($md(d)$).

Messung, *Messgröße*}. Dann gilt

$$\mathbb{A} := \bigcup_{i \in I} \mathbb{A}_i \tag{4.2}$$

mit $\mathbb{A}_{Experiment}$ als Menge der Metadaten, die ein biologisches *Experiment* beschreiben, $\mathbb{A}_{Lebewesen}$ als Menge der Metadaten, die ein *Lebewesen* beschreiben, $\mathbb{A}_{Messung}$ als Menge der Metadaten, die eine *Messung* beschreiben und $\mathbb{A}_{Messgroesse}$ als Menge der Metadaten, die eine *Messgröße* beschreiben. Es gilt ferner, dass

$$\mathbb{A}_i \cap \mathbb{A}_j = \emptyset, i \neq j.$$

$\mathbb{A}$ ist somit eine auch disjunkte Mengenfamilie. Ein *Metadatum* $a \in \mathbb{A}$ beschreibt folgerichtig entweder ein *Experiment*, ein *Lebewesen*, eine *Messung* oder eine *Messgröße*. Ferner definieren wir die Funktionen zur Verknüpfung der Datenwerte mit deren Metadaten. Sei d ein Datenwert. Dann ist die Funktion

$$md(d) = (a_1, a_2, a_3, a_4) \tag{4.3}$$

die Abbildung von einem Datenwert d auf ein Tupel der vier Metadaten des Datenwertes. Dabei gilt: $a_1 \in \mathbb{A}_{Experiment}$, $a_2 \in \mathbb{A}_{Lebewesen}$, $a_3 \in \mathbb{A}_{Messung}$ und $a_4 \in \mathbb{A}_{Messgroesse}$.

4.2.2 Integration auf der Metadatenebene

Die Integration der Daten auf der Metadatenebene soll durch einen attributierten Graphen $G = (\mathbb{K}, \mathbb{E})$, dem *MetadatenGraph*, realisiert werden (vergleiche Abbildung 4.3). Jedes Objekt $o \in \mathbb{A} \cup \mathbb{D}$ wird im Graphen durch einen *Objektknoten* repräsentiert. Objektknoten

Algorithmus 1 Erweiterung MetadatenGraph

Eingabe: $o \in \mathbb{A} \cup \mathbb{D}$, MetadatenGraph g
1: $v :=$ erzeuge Objektknoten für o in g
2: **if** $m \notin \mathbb{D}$ **then**
3: $v :=$ rufe Algorithmus 2 mit Parameter v und g auf
4: **end if**
5: **for each** m ist Nachfolger von o **do**
6: **if** $m \in \mathbb{A} \setminus \mathbb{A}_{Messgroesse}$ **then**
7: $w :=$ rufe Algorithmus 1 mit Parameter m und g rekursiv auf
8: erzeuge gerichtete Kante von v nach w in g
9: **end if**
10: **end for**
11: **if** $o \in \mathbb{D}$ **and** o besitzt Relation zu *Messgröße* m **then**
12: $w :=$ erzeuge Objektknoten für m in g
13: $w :=$ rufe Algorithmus 2 mit Parameter w und g auf
14: erzeuge gerichtete Kante von w zu v in g
15: **end if**
16: **Ende und Rückgabe:** v

besitzen diesselben Attribute wie der Datenwert. Die Zuordnung von Objektknoten v zu dem jeweiligen Objekt o erfolgt durch die Repräsentations-Funktion

$$rpr(v) = o \tag{4.4}$$

mit $v \in \mathbb{K}$ und $o \in \mathbb{A} \cup \mathbb{D}$. Um den MetadatenGraph aufzubauen, werden iterativ für jedes Objekt Knoten zum Graphen hinzugefügt (vergleiche Algorithmus 1). Für jedes Experiment (bestehend aus Metadaten und Datenwerten), das integriert werden soll, wird dazu folgende Prozedur durchlaufen: Zu Beginn wird für das *Experiment* ein Experimentknoten erstellt. Folgend werden Knoten für alle Objekte vom Typ *Lebewesen*, welche eine Relation zum *Experiment*-Objekt besitzen, hinzugefügt und durch eine gerichtete Kante vom Experimentknoten ausgehend verbunden. Die Kantenrichtung basiert im MetadatenGraph immer auf den 1..n-Kardinalitäten des Datenmodells („von 1 zu n“). Für jedes *Lebewesen*-Objekt werden im Folgenden Objektknoten der durch eine Relation verknüpften *Messung*-Objekte hinzugefügt. Analog dazu werden von diesen wiederum jeweils die verknüpften Objekte vom Typ Datenwert erstellt und wieder durch Kanten verbunden. Falls das Datenwert-Objekt eine Relation zu einem *Messgröße*-Objekt besitzt, so wird ein Knoten für das Messgröße-Objekt erzeugt und ausgehend von diesem mit dem entsprechenden Datenwertknoten verbunden (entsprechend der 1..n-Beziehung).

Für die visuelle Inspektion und Interaktion ist es von Vorteil, wenn die Größe des Graphen möglichst langsam wächst. Durch den Algorithmus 2 wird verhindert, dass mehrere Objektknoten erzeugt werden, welche identische Attributwerte aufweisen. Jedesmal, wenn ein Knoten zum Graphen hinzugefügt werden soll, erfolgt zuerst die Suche nach Objektknoten, welche das gleiche Objekt (im Sinne der Attributwerte) wie das zu impor-

Algorithmus 2 Zusammenführen

Eingabe: Objektknoten v, MetadatenGraph g

1: **for each** Objektknoten w in g **do**
2: **if** w hat gleiche Attributwerte wie v **then**
3: lösche v aus g
4: **Ende und Rückgabe:** w
5: **end if**
6: **end for**
7: **Ende und Rückgabe:** v

tierende repräsentieren. Wird ein solcher Knoten gefunden, so wird der neue Objektknoten gelöscht und der existierende Knoten so weiterverwendet, als wäre dieser der neu erzeugte Objektknoten. Dies beinhaltet auch die Erzeugung einer neuen Kante zu dem bereits bestehenden Knoten und das rekursive Bearbeiten aller Nachfolgerobjekte. Somit können Objektknoten mehrere eingehende Kanten besitzen. Die in Algorithmus 2 beschriebene Operation wird ausschließlich bei den Metadatenobjekten $o \in \mathbb{A}$ durchgeführt, aber nicht bei den Datenwerte $o \in \mathbb{D}$. Die Operation erlaubt es, Ergebnisse verschiedener Experimente in einen Kontext zu setzen, da die Metadaten-Objekte verschiedener Experimente zusammengeführt werden. Sind Datenwerte mehrfach vorhanden, beispielsweise von zusammengeführten Experimenten, so können diese als Replikatwerte behandelt werden. Es entsteht somit, basierend auf dem Datenmodell, ein kompakter, gerichteter, azyklischer Graph (vergleiche Abbildung 4.3, Seite 49).

Sei m die Zahl der bereits im MetadatenGraph integrierten Datenwerte und n die Zahl zu importierender Datenwerte. Der Import des ersten Datenwertes resultiert in der Erzeugung von fünf Objektknoten (jeweils *Experiment*, *Lebewesen*, *Messung*, Datenwert und eventuell *Messgröße*), für die jeweils die vorhandene Menge der $O(m)$ Objektknoten auf Gleichheit überprüft werden muss. Für den zweiten Datenwert müssen $O(m)+5\cdot 1$ Objektknoten durchlaufen werden usw. Der letzte (n-te) durchläuft $O(m)+5\cdot(n-1)$ Objektknoten. Somit ergibt sich eine Gesamtkomplexität für Algorithmus 1 und 2 bei n zu importierenden Datenwerten mit m integrierten Datenwerten von $n \cdot O(m) + O((n-1)\frac{(n-2)}{2})) = O(n^2 + m \cdot n)$.

Der MetadatenGraph repräsentiert zu jedem Zeitpunkt alle integrierten Daten. Wenn sich die Menge der integrierten Daten ändert, wird dies im MetadatenGraph nachvollzogen. Neue Daten können mittels Import durch den Anwender oder aber durch Anwenden einer Mappingfunktion in den MetadatenGraph gelangen (vergleiche dazu Abschnitt 4.3.1, Seite 54). Da die Graphstruktur die Beziehungen zwischen Metadatenentitäten der entsprechenden Datenwerte darstellt, erlaubt sie es, die Metadaten visuell zu analysieren und durchsuchen. Diese Möglichkeit wird in Abschnitt 4.2.4, Seite 52 näher beleuchtet. Um den Anwender nicht mit schnell wachsenden Graphen zu überfordern, werden die für die visuelle Inspektion weniger wichtigen Datenwert- und *Messgröße*-Knoten in der Visualisierung des MetadatenGraph nicht berücksichtigt.

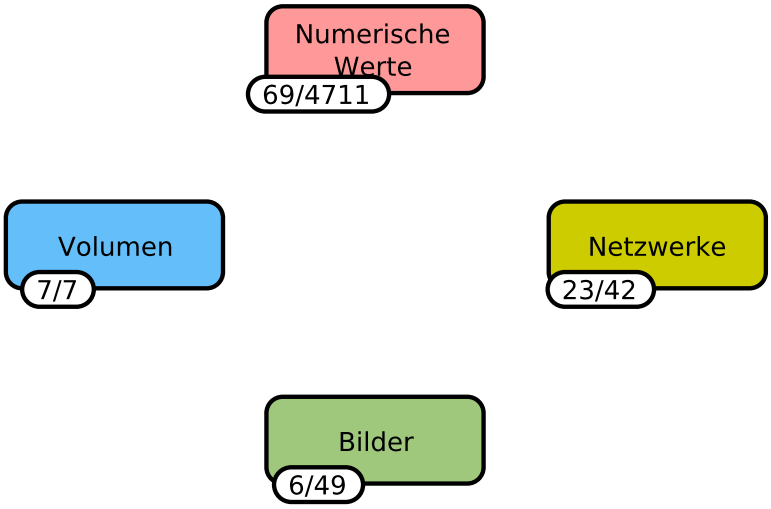


Abbildung 4.4: Integration der Daten auf der Datenwertebene in Form eines Graphen mit vier Datenwertimportknoten (DWI-Knoten). Jeder Knoten enthält eine Liste aller verfügbaren bzw. im System integrierten Datenwert-Objekte eines Typs (#verfügbare Datenwerte/#integrierte Datenwerte). In Abschnitt 4.3.1, Seite 54 wird dieser Graph um Mappings erweitert.

4.2.3 Integration auf der Datenwertebene

Die Integration der Daten auf der Datenwertebene soll durch vier Knoten eines attributierten Graphen $G = (\mathbb{K}, \mathbb{E})$, dem *MappingGraph*, realisiert werden. Da üblicherweise eine große Zahl an Datenwerten aus externen Quellen importiert werden, ist es aus Gründen der Übersichtlichkeit und Skalierbarkeit nicht sinnvoll, jeden einzelnen durch einen Knoten zu repräsentieren. Stattdessen sollen alle importierten Datenwerte entsprechend der Datentyp-Aufteilung in Tabelle 4.1, Seite 47 zusammengefasst werden. Dies wird durch *Datenwertimportknoten* (DWI-Knoten) realisiert. Ein DWI-Knoten enthält eine Liste aller importierten Datenwerte desselben Datenwerttyps (entweder *numerischer Wert*, *Bild*, *Volumen* oder *Netzwerk*). Entsprechend existieren im MappingGraph immer genau vier DWI-Knoten (siehe Abbildung 4.4). Werden Experimente in das System importiert, erfolgt die Integration in den MetadatenGraph und parallel dazu werden alle Datenwerte nach ihrem Typ aufgeteilt und der Liste des jeweiligen DWI-Knotens hinzugefügt.

4.2.4 Filterung

Der MetadatenGraph erlaubt mittels der Interaktionstechnik Filtering eine Arbeitsmenge von Datenwerten und Metadaten, an denen der Anwender temporär interessiert ist, auszuwählen. Dadurch ist es möglich, neben der Übersicht über alle integrierten Daten, auch den Fokus auf wenige Datenwerte zu legen und detaillierte Analysen zu ermöglichen. Oft sind Anwender nämlich nicht an allen, sondern einer kleinen Menge von integrierten Daten interessiert, beispielsweise denen einer bestimmten Spezies. Deswegen definieren wir die Datenwert-Arbeitsmenge als $\mathbb{D}_{work} \subseteq \mathbb{D}$, welche wie folgt erzeugt wird (vergleiche Al-

Algorithmus 3 Erzeugung Arbeitsmenge

Eingabe: $\mathbb{D}_i$ [alle integrierten Datenwerte], MetadatenGraph g

1: $\mathbb{D}_{work} := \mathbb{D}_i$
2: **repeat**
3: Anwender selektiert und versteckt v in g
4: **for each** $d \in \mathbb{D}_{work}$ **do**
5: **if** $rpr(v) \in md(d)$ **then**
6: $\mathbb{D}_{work} := \mathbb{D}_{work} \setminus \{d\}$
7: **end if**
8: **end for**
9: **for each** unversteckten Objektknoten v in g [geordnet nach Experiment-, Lebewesen-, Messung- und Datenwertknoten] **do**
10: **if** alle Vorgänger von v sind versteckt **then**
11: verstecke v
12: **end if**
13: **end for**
14: **until** Anwender beendet Interaktion oder $\mathbb{D}_{work} = \emptyset$
15: **Ende und Rückgabe:** $\mathbb{D}_{work}$

gorithmus 3): Der Anwender selektiert und versteckt im MetadatenGraph Objektknoten mit Hilfe von Suchfeldern oder durch manuelle Selektion. Ist mindestens eines der Metadatenobjekte eines Datenwertes durch den versteckten Objektknoten repräsentiert, wird der Datenwert aus der Arbeitsmenge $\mathbb{D}_{work}$ entfernt. Danach werden nacheinander alle unversteckten *Experiment*-Knoten, *Lebewesen*-Knoten, *Messung*-Knoten und Datenwert-Knoten überprüft, ob alle Vorgänger des jeweiligen Knotens versteckt sind. Ist dies der Fall, so wird auch dieser Knoten versteckt. Diese Prozedur stellt sicher, dass alle Objektknoten versteckt sind, deren Datenwerte schon aus der Menge $\mathbb{D}_{work}$ entfernt wurden. Der komplette Vorgang ist auf Wunsch wiederholbar und somit wird $\mathbb{D}_{work}$ kontinuierlich verkleinert. Der Anwender kann durch Sichtbarmachen aller Objektknoten die Arbeitsmenge wiederherstellen ($\mathbb{D}_{work} := \mathbb{D}_i$). Die Laufzeit des Algorithmus pro vom Anwender versteckten Knoten und n integrierten Datenwerten ist $O(n^2)$: Im ungünstigsten Fall versteckt der Anwender den einzigen Experimentknoten und somit werden auch automatisch alle restlichen Knoten des MetadatenGraph ($O(n)$) versteckt. Dazu wird für jeden dieser Knoten in $O(n)$ entschieden, ob er nur versteckte Vorgänger besitzt, die auch versteckt sind. Somit ergibt sich die Laufzeit $O(n) \cdot O(n) = O(n^2)$.

Der MappingGraph reagiert auf solche Änderungen der Arbeitsmenge, indem sich die verfügbaren Datenwerte der DWI-Knoten temporär verringern. *Verfügbare Datenwerte* sind somit alle Datenwerte $d \in \mathbb{D}_{work}$, die für weitere Interaktionen nutzbar sind und somit aktuell interessante Datenwerte repräsentieren. Solche Interaktionen sind im Folgenden Mappingfunktionen (siehe Abschnitt 4.3.1, Seite 54), welche diese Datenwerte miteinander kombinieren und Visualisierungsfunktionen, die einzelne Datenwerte darstellen können.

Fazit

Die Integration multimodaler Daten erfolgt auf Basis des Datenmodells und einer geeigneten Formalisierung. Der MetadatenGraph repräsentiert die Beziehungen der Metadaten verschiedener Experimente als DAG und erlaubt die visuelle Inspektion und das Filtern dieser Informationen. Die Datenwerte werden aufgeteilt und in vier DWI-Knoten des MappingGraphen als einfache Listen integriert. Die Filteroperationen im MetadatenGraph übertragen sich auch auf die DWI-Knoten. Damit kann die Komplexität großer Datenmengen reduziert und die Übersichtlichkeit erhöht werden.

4.3 Kombination

Die integrierten Datenwerte können durch Kombination in Bezug zueinander gesetzt werden. Diese kombinierten Datenwerte werden als Mappings bezeichnet, welche durch Anwenden von Mappingfunktionen erzeugt werden. Dazu nehmen die Funktionen andere Mappings als Eingabe. Somit sind Datenwerte in mehreren Mappings immer wieder kombinierbar.

4.3.1 Mappings und Mappingfunktionen

Sei $\mathcal{P}^+(\mathbb{D})$ die nicht-leere Potenzmenge der Datenwertmenge $\mathbb{D}$. Sei $\mathbb{T}$ eine (möglicherweise leere) Menge zusätzlicher Attribute. Dann ist ein *Mapping* $m \in \mathbb{M}$ (mit $\mathbb{M}$ als die *Menge aller möglichen Mappings*) definiert als ein Tupel

$$m = (\mathcal{P}^+(\mathbb{D}), \mathbb{T}). \tag{4.5}$$

Somit repräsentiert ein Mapping m die Kombination beliebig vieler Datenwerte und einer Menge zusätzlicher Attribute. $\mathbb{T}$ beschreibt zusätzliche Eigenschaften des Mappings, welche nicht als Attribute eines Datenwertes repräsentiert sind, z. B. Zuordnungen zwischen numerischen Werten und Bildsegmenten. Darüber hinaus beschreiben sie die Zuordnung des Mappings zu einer Menge möglicher Visualisierungsfunktionen und zusätzliche Visualisierungsparameter des Mappings, z. B. spezielle Darstellungsarten von Datenwerten (vergleiche hierzu Abschnitt 4.4.1, Seite 59). Da die Menge $\mathbb{T}$ beliebig groß und für jedes Mapping unabhängig bestimmt wird, soll diese nicht weiter formalisiert werden. In Abschnitt 4.4.2.2, Seite 62 sind einige Beispiele für die Menge $\mathbb{T}$ beschrieben.

Mappings werden im MappingGraph als zusätzliche Knoten repräsentiert (vergleiche Abbildung 4.5). Die in Abschnitt 4.2.4, Seite 52 beschriebene Filterung hat auch Auswirkung auf die Mappingknoten: Falls ein Datenwert eines Mappings durch die Filteroperation versteckt ist, wird auch der Mappingknoten im MappingGraph versteckt, da der Anwender temporär nicht mehr an den Daten interessiert ist.

Mappings werden durch Anwenden einer *Mappingfunktion map* erzeugt, welche definiert ist durch

$$map : \mathcal{P}^+(\mathbb{M}) \mapsto \mathbb{M}. \tag{4.6}$$

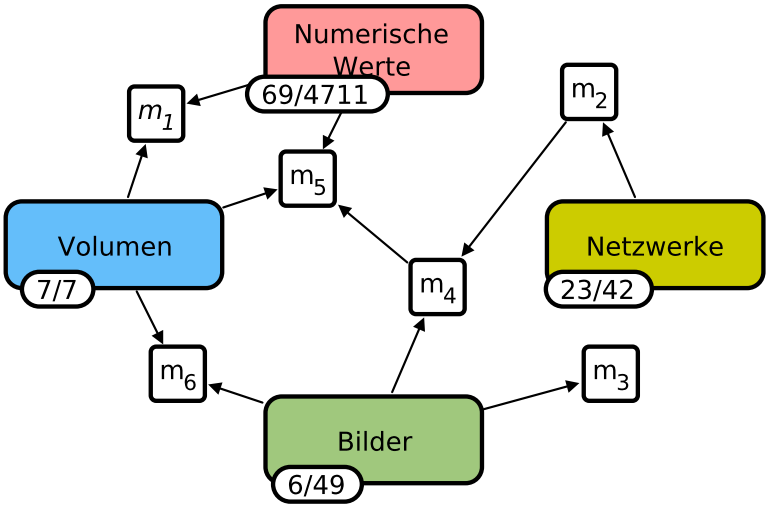


Abbildung 4.5: Beispiel eines mit Mappings gefüllten MappingGraphen. Er besteht aus einer Menge von Mappingknoten (kleine weiße Knoten), welche jeweils ein Mapping m (hier $m_1, \ldots, m_6$) repräsentieren. Zusätzlich existieren die vier Datenwertimportknoten (große bunte Knoten), welche alle integrierten Datenwert-Objekte eines Typs repräsentieren. Kanten repräsentieren das Anwenden von Mappingfunktionen und zeigen von den Quellknoten der Datenwerte zu den resultierenden Mappingknoten.

Mappingfunktionen bilden also von mindestens einem Mapping auf ein anderes Mapping ab und folgerichtig können Mappings wieder neu kombiniert werden (was in Abschnitt 4.3.2, Seite 56 näher beleuchtet wird). Zu den möglichen Eingabe-Mappings der Funktion zählen auch die verfügbaren Datenwerte der DWI-Knoten, indem diese temporär als ein Mapping gelten. Mappingfunktionen können parametrisiert werden, zum Beispiel eine Volumenrekonstruktionsfunktion (vergleiche Seite 66) mit der Größe (Voxelzahl) des zu rekonstruierenden Volumens. Mappingfunktionen können, basierend auf den Eingabe-Datenwerten, neuartige Datenwerte erzeugen. Beispielsweise kann die Volumenrekonstruktionsfunktion aus einer Menge von *Bildern* einen neuen *Volumen*-Datenwert errechnen. Das Anwenden einer Mappingfunktion wird im Folgenden auch als *kombinieren* bezeichnet und im MappingGraph durch Kanten repräsentiert (vergleiche Abbildung 4.5). Kanten zeigen jeweils ausgehend von den Knoten, die als Datenwertquelle fungieren, zu dem erzeugten Mappingknoten. Da eine Mappingfunktion mehrere Mappings als Eingabe erhalten kann, ist es möglich, dass ein Mapping mehrere eingehende Kanten besitzt. DWI-Knoten besitzen hingegen nur ausgehende Kanten, da sie die im MappingGraph integrierten Datenwerte repräsentieren und ausschließlich als Datenwertquelle zur Erstellung neuer Mappings fungieren. Kanten erlauben es somit im MappingGraph, den Fluss der Datenwerte visuell zu verfolgen, da Datenwerte wieder und wieder in Mappings kombiniert und weiterverwendet werden können.

Die Menge anwendbarer Mappingfunktionen ist abhängig von der aktuellen Selektion der Mapping- und DWI-Knoten, weil jede Mappingfunktion nur eine bestimmte Auswahl von Datenwerten verarbeiten kann. Dieser Auswahlprozess soll im Folgenden beschrieben

Algorithmus 4 Erzeugung eines Mappings

Eingabe: $\mathbb{K}_{sel}$ [Menge selektierter Knoten]
1: $\mathbb{D}_{sel} := \emptyset$ [Menge selektierter Datenwerte]
2: $\mathbb{P}_{work} := \emptyset$ [Menge ausführbarer Mappingfunktionen]
3: **for each** $v \in \mathbb{K}_{sel}$ **do**
4: **if** v ist DWI-Knoten **then**
5: $\mathbb{D}_{sel} := \mathbb{D}_{sel} \cup \{$verfügbare Datenwerte aus $v\}$
6: **else**
7: $\mathbb{D}_{sel} := \mathbb{D}_{sel} \cup \{$kombinierte Datenwerte aus $v\}$
8: **end if**
9: **end for**
10: $\mathbb{D}^+ :=$ Anwender wählt interessante, nicht-leere Teilmenge von $\mathbb{D}_{sel}$ aus
11: **for each** Mappingfunktion *map* **do**
12: **if** *map* kann Eingabe $\mathbb{D}^+$ verarbeiten **then**
13: $\mathbb{P}_{work} := \mathbb{P}_{work} \cup \{map\}$
14: **end if**
15: **end for**
16: Anwender wählt $map \in \mathbb{P}_{work}$ und führt diese mit Eingabe $\mathbb{D}^+$ aus
17: **Ende und Rückgabe:** von *map* neu erzeugter Mappingknoten w

werden (vergleiche Algorithmus 4): Der Anwender selektiert eine Zahl von Knoten im MappingGraph. Alle in den selektierten DWI-Knoten verfügbaren Datenwerte und alle kombinierten Datenwerte der selektierten Mappings werden in einer Menge $\mathbb{D}_{sel}$ zusammengefasst. Jede Mappingfunktion *map*, welche eine vom Anwender ausgewählte Menge $\mathbb{D}^+ \subseteq \mathbb{D}_{sel}$ als Eingabe verarbeiten kann, ist vom Anwender ausführbar. Die gewählte Mappingfunktion nimmt $\mathbb{D}^+$ als Eingabe und erzeugt ein neues Mapping, welches durch einen neuen Mappingknoten im MappingGraph repräsentiert wird. Zusätzlich werden Kanten von allen Knoten, die als Quelle mindestens eines Datenwertes fungierten, zu dem Mappingknoten erstellt.

4.3.2 Rekombination kombinierter Daten

Die Definition von Mappings, Mappingfunktionen und dem MappingGraphen erlaubt eine flexible Handhabung biologischer Datenwerte verschiedenen Typs: Unabhängig von der Herkunft der Datenwerte können diese zu neuen Datenwerten kombiniert werden, welche wiederum kombinierbar sind. Die dafür wesentliche Eigenschaft ist, dass Mappings Kombinationen der Datenwerte sind und sowohl als Eingabe, als auch Ausgabe einer Mappingfunktion auftreten. Dadurch befindet sich jedes Ergebnis eines Kombinationsschrittes wieder innerhalb der Menge der Mappings und kann ohne Einschränkungen rekombiniert werden.

Durch diese Abgeschlossenheit ist es möglich, Mappingfunktionen iterativ anzuwenden. Die hohe Kombinierbarkeit von Datenwerten und Mappingfunktionen erlaubt es somit,

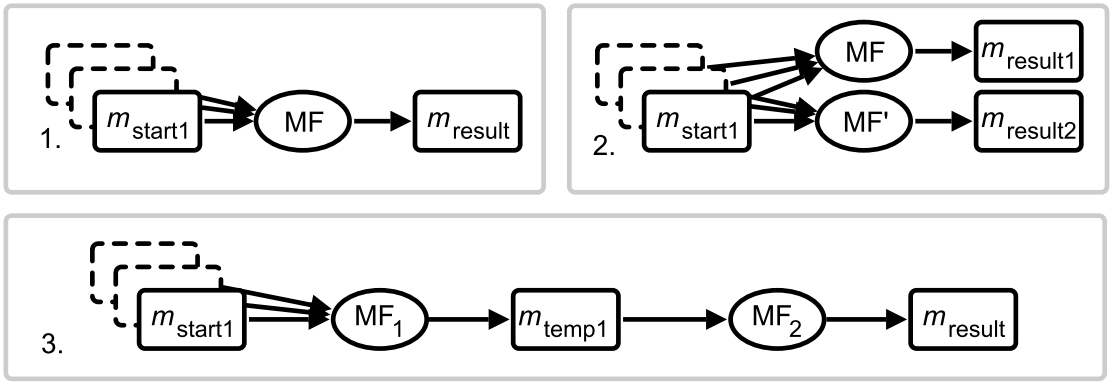


Abbildung 4.6: Darstellung verschiedener Fälle der Rekombination. Rechtecke repräsentieren Mappings, gestrichelte Rechtecke deuten potentielle weitere Eingabe-Mappings an, Ellipsen repräsentieren das Anwenden einer Mappingfunktion und Kanten den Fluss von Datenwerten. Fall 1: Kombination von Mappings zu einem neuen Mapping. Fall 2: mehrmaliges Anwenden einer Mappingfunktion mit verschiedenen Parametern. Fall 3: Anwenden einer Mappingfunktion auf das Ergebnis einer anderen Mappingfunktion.

Datenwerte beliebig in Kontext zueinander zu setzen. Es sind drei Fälle denkbar, die schematisch und beispielhaft in den Abbildungen 4.6 respektive 4.7 dargestellt sind:

1. Anwenden einer Mappingfunktion zur Erzeugung eines Mappings basierend auf den Datenwerten der Eingabe-Mappings. Hierbei steht die neuartige Kombination der vorhandenen Datenwerte und Generierung zusätzlicher Datenwerte während der Kombinationsprozedur im Mittelpunkt.

2. Mehrfaches Erzeugen neuer Mappings aus ein- und denselben Eingabe-Mappings durch wiederholtes Ausführen derselben Mappingfunktion. Im Vordergrund steht hier die Exploration verschiedener Mappingfunktionsparameter und durch Vergleichen der Ergebnisse.

3. Gerichtetes Nacheinanderschalten verschiedener Mappingfunktionen, die ausgehend von einem oder mehrerer Mappings die Datenwerte der Ergebnis-Mappings wieder kombinieren. Dies ist nützlich, wenn ein komplexer Anwendungsfall durch gezieltes Anwenden von Mappingfunktionen realisiert werden kann. Die Mappingfunktionen stellen somit funktionelle Module dar.

Visuell unterstützt und nachvollziehbar wird die flexible Rekombination durch Kanten im MappingGraphen, welche die Anwendung der Mappingfunktionen grafisch nachvollziehbar machen. Diese und weitere Hilfen werden im Abschnitt 5.1.3, Seite 73 ausführlicher diskutiert.

Sobald ein Mapping erzeugt wurde, können Datenwert-Attribute wie Farbe, Transparenz, Position, sowie die im Mapping enthaltenen Datenwerte interaktiv verändert werden. Diese Attributänderungen werden gespeichert, so dass bei einer Rekombination die Datenwerte auch verändert kombiniert werden. Ein Beispiel dafür ist das Umfärben eines

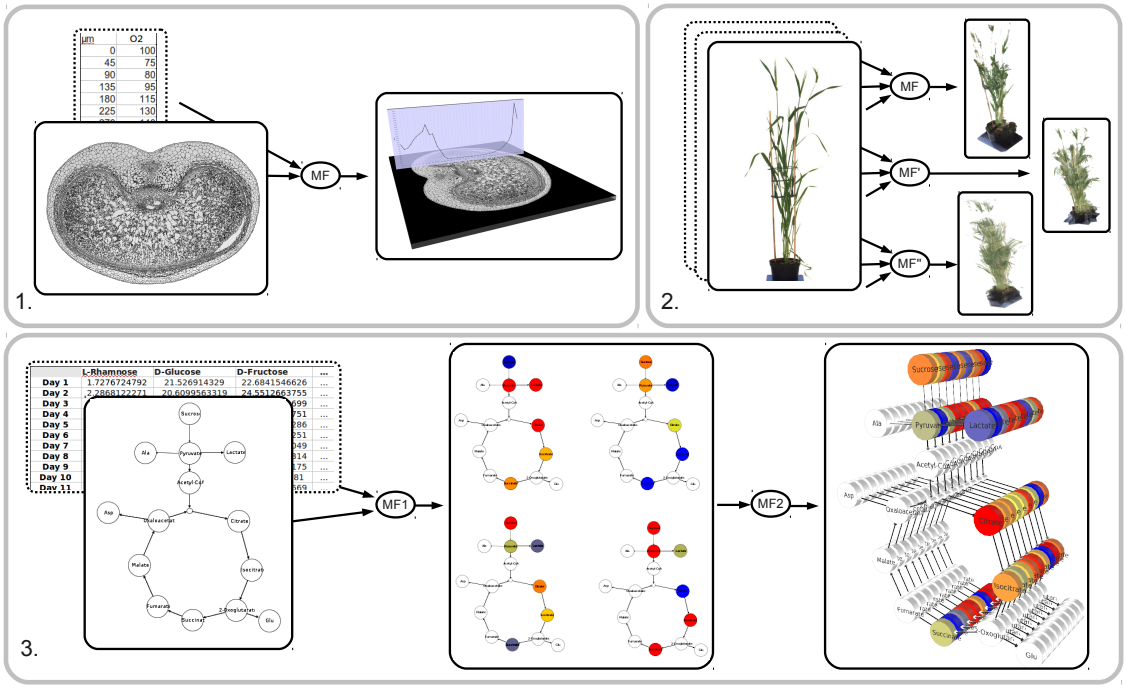


Abbildung 4.7: Beispielanwendungsfälle für die drei Fälle der Rekombination. Fall 1: Kombination eines Sauerstoffgradienten und eines Schnittbildes. Fall 2: eine Volumenrekonstruktionsfunktion (vergleiche Seite 66) wird dreimal mit jeweils 6, 20 und 60 Rotationsbildern als Eingabe ausgeführt, um die Qualität des resultierenden Volumens zu vergleichen. Fall 3: zeitabhängige Metabolitmessungen und ein Netzwerk werden zu einer Menge angereicherter Netzwerke kombiniert. Diese Netzwerke werden darauffolgend in drei Dimensionen gestapelt (vergleiche Abschnitt 6.3, Seite 85).

Volumens, welches darauffolgend in einem neuen Mapping genauso umgefärbt wiederverwendet wird.

Fazit

Der in Abschnitt 4.2.3, Seite 52 vorgestellte MappingGraph wird um Mappingknoten und Kanten zwischen diesen Knoten erweitert. Mappingknoten repräsentieren Mappings und diese wiederum eine Menge kombinierter Datenwerte. Mappings werden durch den Anwender mittels Mappingfunktionen aus Datenwerten anderer Mappings (und verfügbarer Datenwerte aus den DWI-Knoten) erzeugt. Kanten repräsentieren den Fluss von Datenwerten durch den MappingGraphen. Die Definition von Mappings und Mappingfunktionen erlaubt somit eine flexible und iterative Rekombination der Datenwerte im MappingGraph: Datenwerte können nach und nach verändert und in verschiedene Kontexte gebracht werden. Dies ist hilfreich, um komplexe Anwendungsfälle realisieren zu können (vergleiche auch Kapitel 6, Seite 79).

Schlussfolgernd stellt der MappingGraph eher ein Reservoir von Datenwerten und deren Kombinationen dar, denn ein dynamisches Konstrukt im Sinne der Interaktionsmöglichkeiten. Eine dynamischere Prozedur ist vorstellbar, in der sich zum Beispiel nachfolgende

Mappings ändern, sobald ein Datenwert in einem Quellmapping geändert wird. Dies wäre zwar sehr mächtig, aber aufgrund der sich ständig ändernden Visualisierungen und Kontextänderungen schwer nachzuvollziehen und verwirrend. Dies ist zum Beispiel der Fall für die Anwendungen SCIRun [106] und Amira [140]), welche einen Graphen als Berechnungsvorschrift nutzen, der auf Anwenderwunsch hin ausgeführt wird. Ein solcher Ansatz benötigt viel Einarbeitungzeit, um damit produktiv arbeiten zu können. Die hier vorgestellte statische Datenkombination erlaubt es hingegen, Erkenntnisse über einen längeren Zeitraum konservieren zu können und vergangene Anwenderaktion und somit den Arbeitsablauf nachvollziehbar zu machen. Somit werden die Anforderungen in Abschnitt 3.1.3, Seite 32 realisiert, da Mappings und der Rekombinationsverlauf auch bei Weitergabe der Resultate an Partner und auf anderen Rechnern unverändert und nachvollziehbar bleibt.

4.4 Visualisierung

Kombinierte multimodale Daten können durch geeignete Visualisierungen einfacher verstanden und exploriert werden. Dazu ist eine Auswahl verschiedener Visualisierungsfunktionen nötig, welche basierend auf Mappings unterschiedliche Sichten auf biologische Experimentdaten erzeugen.

4.4.1 Visualisierungsfunktionen

Visualisierungsfunktionen vis projizieren kombinierte Daten in den euklidischen Raum und sind definiert durch

$$vis : \mathbb{M} \mapsto \mathbb{R}^i, i \in \{1, 2, 3\}. \tag{4.7}$$

Welche Visualisierungsfunktion welches Mapping visualisieren kann, ist statisch in der in Mappings enthaltenen Attributmenge $\mathbb{T}$ festgelegt (vergleiche Abschnitte 4.3.1, Seite 54 und 4.4.2.1, Seite 62). $\mathbb{T}$ kann darüber hinaus zusätzliche Parameter der Visualisierungsfunktion wie Färbungen, Kameraposition und Schnittebenen beinhalten.

4.4.1.1 Beispiele für Visualisierungsfunktionen

Es sind viele verschiedene Visualisierungsfunktionen denkbar. Deswegen sollen im Folgenden drei besonders wichtige Funktionen beispielhaft im Detail beschrieben werden, welche ein großes Spektrum von Visualisierungsmöglichkeiten abdecken und flexibel auf multimodale Daten anwendbar sind. Sie können nicht nur Mappings, sondern auch einzelne Datenwerte der DWI-Knoten darstellen. Damit werden interaktive Attributänderungen integrierter Datenwerte ermöglicht, bevor diese kombiniert werden. Mögliche Änderungen sind z. B. Anpassung des Netzwerklayoutes, Hinzufügen von Clusterinformationen und Annotationen, sowie Änderung der Volumen-Farbtabelle.

3D-Visualisierung

Die *3D-Visualisierung* ermöglicht es, alle vier Datenwerttypen in 3D zu visualisieren (vergleiche z. B. Seiten 64 und 68). Die rechenaufwändigste Aufgabe ist es, typische volumetri-

sche Datensätze (< 50 Millionen Voxel) mit möglichst interaktiven Bildwiederholungsraten darzustellen, was durch einen Schnitt-basierten Volumenrenderer realisiert wird (vergleiche Abschnitt 2.5.2.4, Seite 27). Zur Darstellung des *Volumens* werden drei orthogonale Bilderstapel durch den Körper generiert, womit ein Voxel mittels drei orthogonal im 3D-Raum liegender Pixel repräsentiert wird. Die Transparenz der Stapelbilder kann durch Schieberegler verändert werden, um einen Blick in das *Volumen* zu erlauben. Neben der generellen Transparenz des *Volumens* können auch einzelne Ebenen hervorgehoben (durch Reduktion der Transparenz) und Beschneidungen (Clipping) realisiert werden (durch Unsichtbarmachen einer Menge von Ebenen). Zur Erhöhung der Bildwiederholungsfrequenz können Ebenen ausgedünnt oder ganz ausgelassen werden. Darüber hinaus sind durch Verzerrung der Texturen rechteckige (nicht-isotrophe) Voxel repräsentierbar. Im Falle von Grauwertvolumen ist es möglich, Farbtabellen anzuwenden, um Regionen hervorzuheben oder eine allgemein ansprechende Darstellungsform zu ermöglichen [97]. Segmente von *Volumen* können durch Interaktion hervorgehoben oder versteckt werden.

Neben der Darstellung von *Volumen* ist auch die Darstellung von *Bildern* möglich, indem einem flachen Quader das Bild als Textur zugewiesen wird. Im dreidimensionalen Raum bestimmt ein Dicke-Attribut das *Bildes* die Dicke des Quaders. Bilder können in der Größe und der Transparenz geändert werden. Segmentierte *Bilder* können analog zu *Volumen* teilweise versteckt oder hervorgehoben werden.

Netzwerke mit zwei- und dreidimensionalen Layouts sind auch in 3D darstellbar. Knoten sind als Kugeln, Quader oder Zylinder realisiert und Kanten können durch einen Konus und einen Zylinder, aber auch auf Anwenderwunsch als primitive Linie repräsentiert werden. Beide Graphelemente unterstützen Transparenz und Farbänderungen. Die Darstellung *numerischer Werte* ist durch Einbettung in die Knoten möglich, ähnlich zur Darstellung der Diagramme in der Graph-Visualisierung.

Alle Datenwerte können im dreidimensionalen Raum translatiert und rotiert werden. Zusätzlich ist die Kameraposition durch Panning und Zooming frei angepassbar. Weitere Details der Implementierung der 3D-Visualisierung sind in Abschnitt 5.1.4, Seite 75 zu finden.

Graph-Visualisierung

Die *Graph-Visualisierung* ermöglicht es, *Netzwerke* und *numerische Werte* in 2D zu visualisieren (vergleiche z. B. Seite 67). Eine Visualisierung von Graphen, insbesondere biologischer Netzwerke, kann die existierenden Zusammenhänge (Kanten) zwischen biologischen Objekten (Knoten) als einfache grafische Objekte, wie Rechtecke, Kreise und Pfeile, darstellen. Dabei ist es möglich, das Netzwerk frei zu editieren, also Knoten und Kanten hinzuzufügen, zu ändern (Bezeichnung, Größe, Farbe, etc.), zu verschieben und löschen zu können. Panning und Zooming der Graphdarstellung werden ebenso unterstützt. Die Darstellung *numerischer Werte* kann in Form von in den Knoten eingebetteten Diagrammen oder auf Kanten erfolgen. Die Darstellung der Diagramme kann beispielsweise durch Änderung der Farben, Achsendarstellung und Linienbreiten interaktiv angepasst werden. Für detaillierte Informationen zur Integration *numerischer Werte* in den Kontext von

Netzwerken sei auf die Publikationen [59, 76, 77] verwiesen.

Die Graph-Visualisierung unterstützt eine ähnliche Interaktionstechnik, wie von Klukas und Schreiber [76] beschrieben. Dort ist es möglich, KEGG-Pathways in einen Übersichtsknoten zusammenzuklappen. Alle Kanten zu und von den zugeklappten Knoten zeigen dann auf den Übersichtsknoten, statt auf einzelne Graphelemente. Das Aufklappen eines Übersichtsknotens resultiert in der Löschung dieses Knotens und dem Hinzufügen aller Pathway-Elemente des repräsentierten Pathways, inklusive Umsetzen der Kanten. Neu ist, dass jedes beliebige *Netzwerk* kondensiert und aufgeklappt werden kann. Um die Übersichtlichkeit zu verbessern, werden alle Kanten zwischen *Netzwerken* gebündelt (ähnlich zu [45, 56]). Dies erlaubt es, einzelne Kanten zu verfolgen und gleichzeitig eine gute Übersicht über die generellen Netzwerk-Beziehungen zu behalten.

Eine weitere Interaktionstechnik ist die Darstellung zeitlicher Zusammenhänge als Animation (vergleiche Abschnitt 2.5.3, Seite 28). Dies ist insbesondere für die Visualisierung von Flussdatenverteilungen wichtig. Diese Flussdaten sind als Kantendicken im Graphen repräsentiert und können für verschiedene Zeitpunkte oder Entwicklungsstadien variieren. Dem Anwender wird ein Schieberegler angeboten, der es erlaubt, Zeitpunkte nacheinander auszuwählen. Auf diese Ereignisse reagiert die Visualisierung durch Darstellung der Flussraten zu dem spezifizierten Zeitpunkt. Beinhalten die Messungen verschiedene *Lebewesen*, so können diese über ein zusätzliches Klappmenü ausgewählt werden.

Bild-Visualisierung

Die *Bild-Visualisierung* soll es erlauben, *Bilder* und *Volumen* darzustellen (vergleiche z. B. Seite 67). Sie stellt wie bei den üblichen Bildbetrachtern die *Bilder* dar, indem sie auf dem Bildschirm gerastert gezeichnet werden. Die Skalierung kann frei gezoomt werden, um die Ansicht auf große und kleine Ausgabemedien anpassen zu können. Segmentinformation kann durch einen Überblendeffekt zwischen dem eigentlichen Bild und dem Labelfield-Bild realisiert werden (Labelfields sind in Abschnitt 2.2.3, Seite 13 beschrieben). Der Überblendfaktor ist mittels eines Schiebereglers beliebig anpassbar, so dass visuell Segmentierungsfehler und die Zugehörigkeit einzelner Pixel zu Segmenten ermittelt werden können. Die Bild-Visualisierung erlaubt es ferner, einen ganzen sortierbaren Bilderstapel darzustellen, der mittels eines Schiebereglers durchlaufen werden kann, ähnlich zu ImageJ [1]. Dadurch ist es möglich, linear räumlich und zeitlich zusammenhängende *Bilder* während einer Animation in einen Zusammenhang zu bringen. *Volumen* werden als eine Menge von Bildern realisiert, welche mittels Traversierung des *Volumens* in z-Richtung erzeugt werden.

Die Bild-Visualisierung unterstützt die Technik des grafischen Abfragens räumlich in Bezug stehender Datenwerte basierend auf Segmentinformation, ähnlich zu EMAGE [117]: Der Anwender zeichnet Punkte auf das Bild und wählt somit Segmente aus. Alle betroffenen Segmente werden hervorgehoben und eine Abfrage nach *numerischen Werten* aus diesen Segmenten ausgeführt. Somit ist es möglich, Datenwerte basierend auf ihrem strukturellen Kontext auf intuitive Weise abzufragen und im MappingGraph weiterzuverarbeiten. Das grafische Abfragen entspricht also einer Filteroperation.

4.4.2 Integrationssichten

Integrationssichten sind durch Visualisierungsfunktionen generierte Sichten auf kombinierte Daten. Ein Anwender kann diese durch Selektieren eines Mappingknotens und Wahl einer geeigneten Visualisierungsfunktion erzeugen.

4.4.2.1 Zuordnung zwischen Mappings und Visualisierungsfunktionen

Die Erzeugung von Integrationssichten basiert auf der Wahl eines Mappings und einer Visualisierungsfunktion. Potentiell sind sehr viele Mappings und Visualisierungsfunktionen und somit verschiedenste Integrationssichten denkbar. Visualisierungsfunktionen können mehrere Mappings visualisieren, z. B. kann die 3D-Visualisierung sowohl registrierte *Volumen* (Seite 66), als auch gestapelte *Bilder* (Seite 66) darstellen. Mappings wiederum können durch mehr als eine Visualisierungsfunktion dargestellt werden, z. B. die Visualisierung eines *Netzwerkes* kombiniert mit *numerischen Werten* als zweidimensionaler Graph (Knoten als Rechtecke, Kanten als Pfeile, Seite 67) oder als dreidimensionaler Graph (Knoten als Quader, Kanten als Zylinder, Seite 67). Welche Visualisierungen für ein Mapping gewählt werden können, ist fest in der Menge $\mathbb{T}$ eines Mappings vorgegeben (vergleiche Abschnitt 4.3.1, Seite 54). In Tabelle 4.2 ist die Zuordnung $(\mathbb{M}, vis)$ von Mappings bzw. Mappingklassen (repräsentiert durch die Datenwerttypen) zu den implementierten Visualisierungsfunktionen aufgeschlüsselt. Zu erkennen ist in dieser Tabelle, dass die drei in Abschnitt 4.4.1, Seite 59 vorgestellten Visualisierungen sehr viele verschiedene Mappingklassen darstellen können. Andere implementierte Visualisierungen fokussieren hingegen auf eingegrenzte Datenwertkombination und sind somit auf bestimmte Anwendungsfälle spezialisiert. Für manche Mappingklassen, z. B. *Netzwerk*-*Volumen*-Kombination oder Kombination aller Datentypen, sind bisher noch keine Sichten entwickelt. Dies hängt stark von den Anwenderwünschen und dem Implementierungsaufwand ab.

4.4.2.2 Beispiele für Integrationssichten

Basierend auf der im vorherigen Abschnitt vorgestellten Zuordnung von Mappings bzw. Mappingklassen zu Visualisierungsfunktionen sollen hier beispielhaft resultierende Integrationssichten vorgestellt werden. Manche dieser Integrationssichten sind bereits in anderen Publikationen entwickelte Ansätze, welche in diese Methodik aufgenommen, formalisiert und erweitert oder vereinfacht wurden. Andere sind Neuentwicklungen, welche teilweise erst durch den in dieser Arbeit vorgestellten flexiblen Kombinationsansatz möglich wurden.

Aufgelistet sind im Folgenden für jede Integrationssicht aus Tabelle 4.2 der Name, die Eingabe-Datenwerte der Mappingfunktion, die im resultierenden Mapping kombinierten Datenwerte (Ausgabe), den Inhalt von $\mathbb{T}$, eine Abbildung und eine kurze Beschreibung. $\mathbb{N}^{(k)}$ repräsentiert beispielsweise eine Menge von k *Netzwerk*-Datenwerten. Für detaillierte technische Hintergründe sei auf die Implementierung der Anwendung Hive verwiesen.

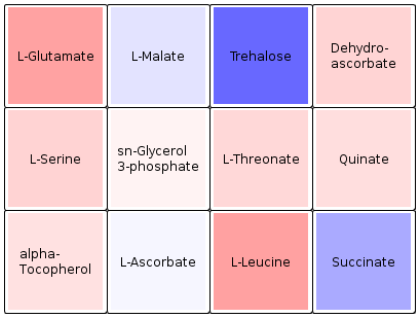
kombinierte Datentypen	3D-Vis.	Graph-Vis.	Bild-Vis.	Brushing-Vis.	Streudiagramm-Vis.	Statistik-Vis.	Netzwerknavigation-Vis.	Bildnavigations-Vis.	Omicsverteilung-Vis.
$\mathbb{U}$-$\mathbb{U}$	-	1, 2	-	-	3	4	-	-	-
$\mathbb{N}$-$\mathbb{N}$	5	6, 7	-	-	-	-	8	-	-
$\mathbb{V}$-$\mathbb{V}$	9	-	-	-	-	-	-	-	-
$\mathbb{B}$-$\mathbb{B}$	10, 11	-	12	-	-	-	-	-	-
$\mathbb{U}$-$\mathbb{N}$	14	13	-	-	-	4	-	-	-
$\mathbb{U}$-$\mathbb{V}$	-	-	-	-	-	-	-	-	-
$\mathbb{U}$-$\mathbb{B}$	15	-	16	-	-	-	-	-	-
$\mathbb{N}$-$\mathbb{V}$	-	-	-	-	-	-	-	-	-
$\mathbb{N}$-$\mathbb{B}$	-	-	-	-	-	-	-	17	-
$\mathbb{V}$-$\mathbb{B}$	18	-	-	-	-	-	-	-	-
$\mathbb{U}$-$\mathbb{N}$-$\mathbb{B}$	-	-	-	19	-	-	-	-	20

Tabelle 4.2: Übersicht über die Zuordnung $(\mathbb{M}, vis)$ von Mappings bzw. Mappingklassen in den Zeilen, repräsentiert durch die Typen der kombinierten Daten, zu möglichen Visualisierungsfunktionen (in den Spalten). Zahlen in den Tabellenzellen verweisen auf eine implementierte Integrationssicht aus Abschnitt 4.4.2.2.

1) Condition LogRatio

Eingabe: $\mathbb{U}^{(k)}, k \geq 2, \#Lebewesen = 2$
Ausgabe: $\mathbb{U}^{(j)}, j < k$
$\mathbb{T}$ **enthält:** Farbcode für die Heatmap

Erzeugt einen neuen Datensatz *numerischer Werte* durch Berechnung des logarithmierten Verhältnisses beider *Lebewesen*. Dargestellt wird der neue Datensatz als Heatmap mit der Graph-Visualisierung.

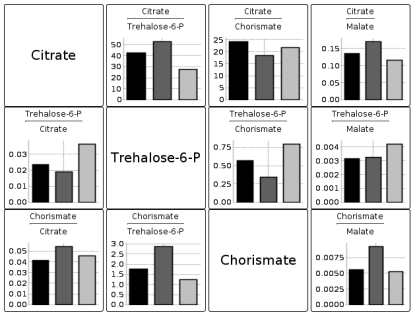


2) Substance Ratio Matrix

Eingabe: $\mathbb{U}^{(k)}, k \geq 2$

Ausgabe: $\mathbb{U}^{(j)}, j > k$

Erzeugt eine Matrix, welche die Verhältnisse aller *numerischen Werte* jedes *Messgröße*-Paares als Balkendiagramm in der Graph-Visualisierung darstellt.

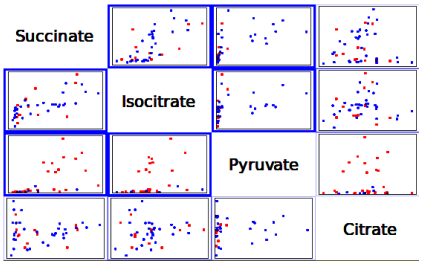


3) Substance Scatter Matrix

Eingabe: $\mathbb{U}^{(k)}, k \geq 2$

Ausgabe: $\mathbb{U}^{(k)}, k \geq 2$

Visualisiert die *numerischen Werte* mittels der Streudiagramm-Visualisierung [121] als Streudiagramm-Matrix für jedes *Messgröße*-Paar.

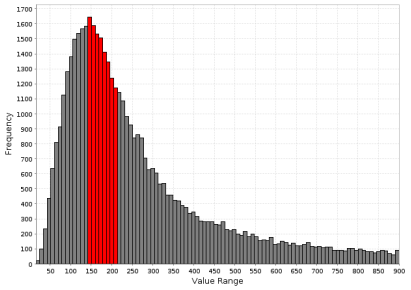


4) Network and Omics Statistics Analysis

Eingabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 0$

Ausgabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 0$

Ähnelt der Integrationssicht 14, die Visualisierung erfolgt aber in der Statistik-Visualisierung [121]. Diese stellt die Verteilung von Graphelement-Attributwerten als Histogramm dar. Dadurch können beispielsweise sehr einfach Genexpressions-Datensätze mittels der Verteilungskurve exploriert und z. B. von Ausreißern befreit werden.



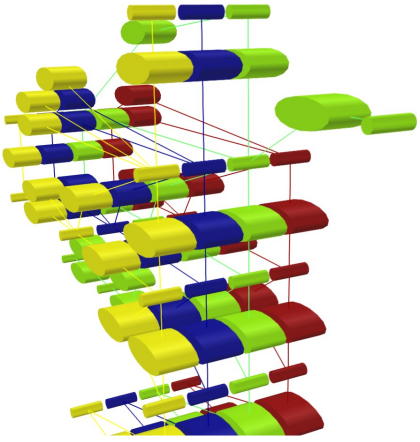
5) Network Stacking

Eingabe: $\mathbb{N}^{(k)}, k \geq 2$

Ausgabe: $\mathbb{N}^{(k)}, k \geq 2$

$\mathbb{T}$ **enthält:** Abstand der gestapelten *Netzwerke*, globale Färbung jedes einzelnen *Netzwerkes*, Sortierung der *Netzwerke* innerhalb des Stapels

Stapelt *Netzwerke* in drei Dimensionen, ähnlich zu [13] (vergleiche auch Abbildung 4.7, Seite 58, 3. Fall). Die Visualisierung erfolgt mittels der 3D-Visualisierung.



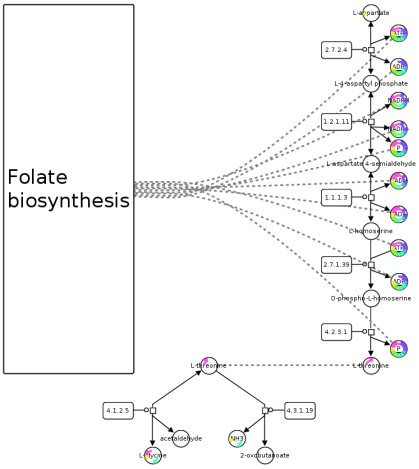
6) Linked Pathway Integration

Eingabe: $\mathbb{N}^{(k)}, k \geq 2$

Ausgabe: $\mathbb{N}^{(k)}, k \geq 2$

$\mathbb{T}$ **enthält:** Attribute zur kreisförmigen Positionierung der *Netzwerke*

Verknüpft eine Menge auf- und zuklappbarer *Netzwerke* durch Verbinden identischer *Messgröße*-Knoten in einem Übersichtsgraphen (vergleiche [76]). Durch Navigation in der Graph-Visualisierung können strukturelle Verbindungen zwischen den *Netzwerken* interaktiv aufgedeckt werden.



---

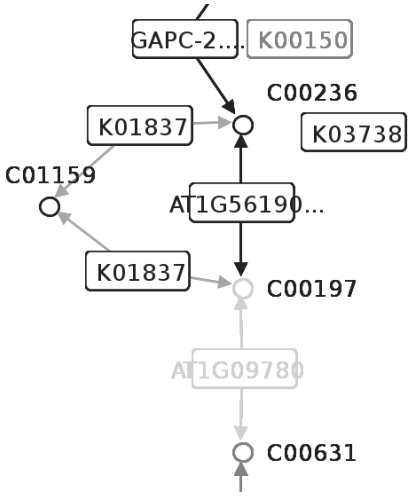
7) Network Comparison

Eingabe: $\mathbb{N}^{(k)}, k \geq 2$

Ausgabe: $\mathbb{N}^{(1)}$

$\mathbb{T}$ **enthält:** Farbcode für Hell-Dunkel-Färbung der Graphelemente

Erzeugt ein neues Netzwerk, bestehend aus den Graphelementen der Eingabe-*Netzwerke*, welche entsprechend der Häufigkeit des Vorkommens in allen Eingabe-*Netzwerken* gefärbt sind (je dunkler, desto öfter kommt das Graphelement in anderen *Netzwerken* vor). Dies ermöglicht eine Übersicht über strukturelle Übereinstimmungen verschiedener *Netzwerke*.



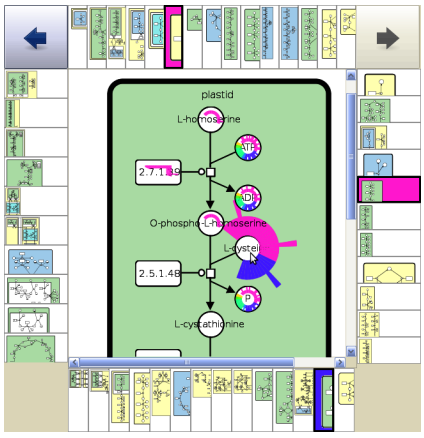
---

8) Advanced Network Navigation

Eingabe: $\mathbb{N}^{(k)}, k \geq 2$

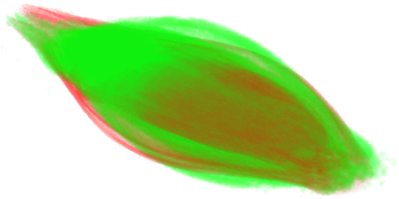
Ausgabe: $\mathbb{N}^{(k)}, k \geq 2$

Entspricht der Integrationssicht 6, wobei die Visualisierung mittels der Netzwerknavigations-Visualisierung erfolgt (vorgestellt in [69]). Per direkter Selektion kann ausgehend von einem *Netzwerk* zu einem anderen ringförmig in der Sicht angeordneten *Netzwerk* gesprungen werden. Dadurch wird die intuitive Exploration einer Menge von *Netzwerken* unterstützt.



9) Volume Registration

Eingabe: $\mathbb{V}^{(k)}, k \geq 2$
Ausgabe: $\mathbb{V}^{(k)}, k \geq 2$
$\mathbb{T}$ **enthält:** Färbung der registrierten *Volumen*

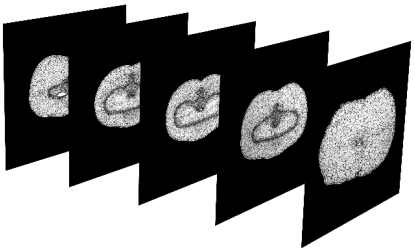


Registriert eine Menge von *Volumen* aufeinander (vergleiche Abschnitt 2.2.4, Seite 15), um verschiedene *Messgrößen* (beispielsweise strukturelle Protonenverteilung und funktionelle Lipidkonzentration) miteinander im dreidimensionalen Raum vergleichen zu können.

---

10) Image Stacking in 3D

Eingabe: $\mathbb{B}^{(k)}, k \geq 2$
Ausgabe: $\mathbb{B}^{(k)}, k \geq 2$
$\mathbb{T}$ **enthält:** Sortierung der *Bilder*



Ordnet eine Menge von *Bildern* im dreidimensionalen Raum an, um Bilderstapel eines biologischen Objektes entsprechend der tatsächlichen Position anzuordnen und durch Anpassen der Kameraposition visuell zu analysieren.

---

11) Volumetric Reconstruction

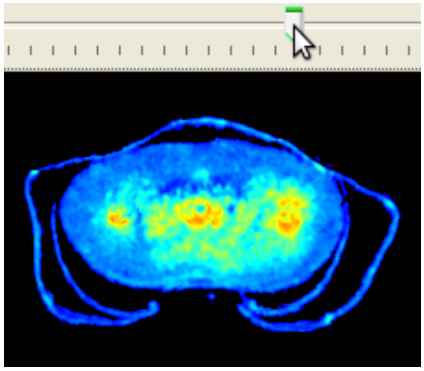
Eingabe: $\mathbb{B}^{(k)}, k \geq 1$
Ausgabe: $\mathbb{V}^{(1)}$

Rekonstruiert ein *Volumen* eines dreidimensionalen Objektes auf Basis von *Bildern* aus unterschiedlichen Perspektiven mittels eines Space-Carving-Algorithmus [81].

---

12) Image Stacking

Eingabe: $\mathbb{B}^{(k)}, k \geq 2$
Ausgabe: $\mathbb{B}^{(k)}, k \geq 2$
$\mathbb{T}$ **enthält:** Sortierung der Bilder

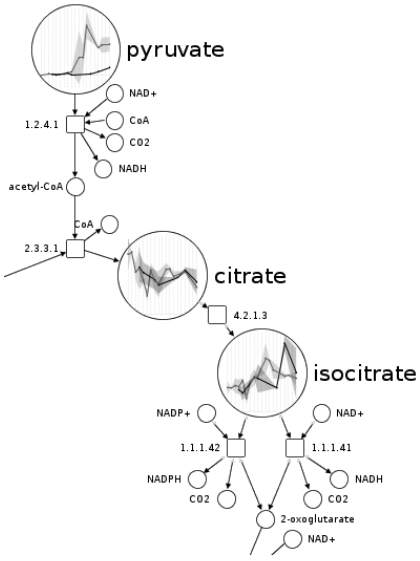


Ähnelt der Integrationssicht 10, wobei die Darstellung in der Bild-Visualisierung erfolgt. Die Eingabe besteht aus einer Menge von *Bildern*, die einen räumlichen oder zeitlichen Zusammenhang aufweisen (beispielsweise ein Warping von 2D-Bildern verschiedener *Messungen* [110]). Die Visualisierung erfolgt mit der Bild-Visualisierung, deren Schieberegler eine flüssige Animation durch die *Bilder* erlaubt.

---

13) Omics Network Context

Eingabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 1$
Ausgabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 1$
$\mathbb{T}$ **enthält:** Visualisierungsparameter für die Diagramme, wie z. B. Diagrammtyp und Farbcode der *Lebewesen*

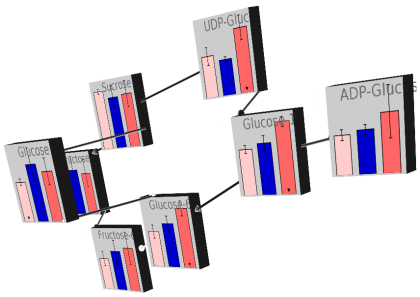


Bringt *numerische Werte* in den Kontext biologischer *Netzwerke* (vergleiche [46]). Die *numerischen Werte* werden den entsprechenden Netzwerkknoten zugewiesen und in der Graph-Visualisierung z. B. als Diagramme dargestellt. Falls Flussdaten visualisiert werden, können mittels Schieberegler und Klappmenü verschiedene *Messungen* und *Lebewesen* hervorgehoben werden.

---

14) Omics Network Context in 3D

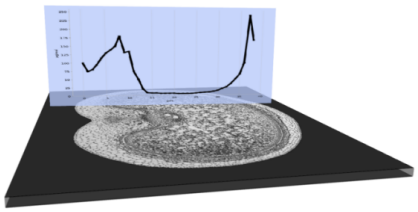
Eingabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 1$
Ausgabe: $\mathbb{N}^{(k)} \times \mathbb{U}^{(j)}, k \geq 0, j \geq 1$
$\mathbb{T}$ **enthält:** Visualisierungsparameter wie Art der Kantenzeichnung und Transparenz der Knoten



Ähnelt der Integrationssicht 13, wobei die Darstellung mit der 3D-Visualisierung erfolgt und somit dreidimensionale *Netzwerk*-Layouts berücksichtigt werden können.

---

15) Gradient on Image

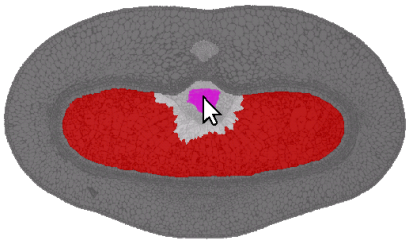


Eingabe: $\mathbb{B}^{(1)} \times \mathbb{U}^{(k)}, k \geq 2, \#Messgrößen = 1, u \in \mathbb{U}$ hat Position
Ausgabe: $\mathbb{B}^{(1)} \times \mathbb{U}^{(k)}, k \geq 2, \#Messgrößen = 1, u \in \mathbb{U}$ hat Position
$\mathbb{T}$ **enthält:** Attribute zur Visualisierung des Gradienten, z. B. Farbe und Liniendicke

Visualisiert einen Gradienten (*numerische Werte*) als Diagramm im Kontext eines Schnittbildes entlang der Linie, an der der Gradient gemessen wurde (vergleiche [123]). Die Darstellung erfolgt mit der 3D-Visualisierung.

---

16) Omics Query by Graphical Selection



Eingabe: $\mathbb{U}^{(k)} \times \mathbb{B}^{(1)}, k \geq 2, b \in \mathbb{B}$ ist segmentiert
Ausgabe: $\mathbb{U}^{(j)}, 0 \leq j \leq k$
$\mathbb{T}$ **enthält:** Zuordnung der *numerischen Werte* zu den jeweiligen Bildsegmenten

Erlaubt es dem Anwender mit grafischen Mitteln in der Bild-Visualisierung *numerische Werte* zu filtern (vergleiche [117, 121]). Dazu kann der Anwender beliebige Bereiche im Bild durch Zeichnen auswählen und alle *numerischen Werte*, die in den betroffenen Segmenten gemessen wurden, abfragen. Diese abgefragten Datenwerte dienen als Ausgabe der Mappingfunktion und können im Folgenden weiterkombiniert werden.

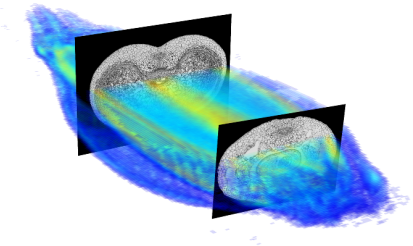
---

17) Image Browsing by Network

Eingabe: $\mathbb{N}^{(k)} \times \mathbb{B}^{(j)}, k \geq 1, b \in \mathbb{B}$ zeigt *Messgröße*
Ausgabe: $\mathbb{N}^{(k)} \times \mathbb{B}^{(j)}, k \geq 1, b \in \mathbb{B}$ zeigt *Messgröße*
Eine Beschreibung und Abbildung ist in Abschnitt 6.2, Seite 82 zu finden.

---

18) Multimodal Alignment



Eingabe: $\mathbb{V}^{(1)} \times \mathbb{B}^{(k)}, k \geq 1$
Ausgabe: $\mathbb{V}^{(1)} \times \mathbb{B}^{(k)}, k \geq 1$
Registriert *Bilder* auf ein *Volumen*, um verschieden aufgelöste oder funktionelle und strukturelle Informationen in einen Kontext zu bringen (vergleiche [129]). Die Visualisierung erfolgt mittels der 3D-Visualisierung.

---

19) Image Omics Brushing

Eingabe: $\mathbb{U}^{(k)} \times \mathbb{N}^{(j)} \times \mathbb{B}^{(1)}, k \geq 2, j \geq 0, b \in \mathbb{B}$ ist segmentiert

Ausgabe: $\mathbb{U}^{(k)} \times \mathbb{N}^{(j)} \times \mathbb{B}^{(1)}, k \geq 2, j \geq 0, b \in \mathbb{B}$ ist segmentiert

Integriert die *numerischen Werte* mit räumlicher Auflösung in den Kontext der *Netzwerke*. Die Visualisierung dieser Daten erfolgt in der Brushing-Visualisierung. Diese nutzt die Interaktionstechnik Brushing, um die *numerischen Werte* räumlich explorieren zu können [121]. Der linke Teil der Sicht visualisiert das Bild, in dem durch Mausbewegungen Segmente selektiert werden können. Die Graph-Visualisierung im rechten Teil reagiert auf dieses Ereignis durch Hervorheben der in dem Segment gemessenen *numerischen Werte* (z. B. durch Anfärben von Balken in Balkendiagrammen). Eine Abbildung ist auf Seite 88 in Abschnitt 6.3 zu finden.

---

20) Omics Distribution by Network

Eingabe: $\mathbb{U}^{(k)} \times \mathbb{N}^{(j)} \times \mathbb{B}^{(1)}, k \geq 2, j \geq 1, b \in \mathbb{B}$ ist segmentiert

Ausgabe: $\mathbb{U}^{(k)} \times \mathbb{N}^{(j)} \times \mathbb{B}^{(1)}, k \geq 2, j \geq 1, b \in \mathbb{B}$ ist segmentiert

Eine Beschreibung und Abbildung ist in Abschnitt 6.1, Seite 79 zu finden.

Fazit

Mittels Visualisierungsfunktionen können aus Mappings Integrationssichten erzeugt werden. Die Zuordnung, welche Funktion welches Mapping visualisieren kann, ist durch das Mapping festgelegt und abhängig von den kombinierten Datenwerttypen. Integrationssichten ermöglichen interaktive und explorative Arbeit mit kombinierten multimodalen biologischen Daten, um ein besseres Verständnis für die Daten und deren Bedeutung zu erlangen. Weiterhin ermöglichen sie die Erzeugung publizierungswürdiger Visualisierung und können auch selbst Teil des Publikationsprozesses sein, da Integrationssichten abgespeichert und auf anderen Rechnern in exakt dieser Form wiederhergestellt werden können. Es wird dadurch ermöglicht, diese Sichten Gutachtern und interessiertem wissenschaftlichen Personal zur Verfügung zu stellen, um interaktiv Daten zu explorieren und bei Bedarf weiterverwenden zu können. Somit realisieren Integrationssichten die geforderte Datensicherheit (siehe Abschnitt 3.1.3, Seite 32) und sind das Ergebnis der Visualisierungspipeline.

4.5 Fazit

Auf Basis einer Visualisierungspipeline konnte gezeigt werden, wie multimodale Daten integriert, kombiniert und schließlich visualisiert werden können, um das Verstehen multimodaler biologischer Daten zu unterstützen. Die Daten wurden, basierend auf einem Datenmodell, in Graphstrukturen integriert. Die integrierten Daten können flexibel miteinander kombiniert und unter Zuhilfenahme verschiedener Visualisierungs- und Interaktionstechniken dargestellt und exploriert werden.

5

Implementierung

5.1 HIVE

Die in dieser Arbeit vorgestellte Methodik wurde in Form der Anwendung Hive implementiert (siehe Abbildung 5.1), welche auf http://www.vanted.org/hive/ verfügbar ist. Hive steht für „**H**andy **I**ntegration and **V**isualisation of multimodal **E**xperimental Data“ und ist als Add-on für das Vanted-System [67, 75] realisiert. Diese am Institut für Pflanzengenetik und Kulturpflanzenforschung Gatersleben entwickelte Quellcode-freie Anwendung kann Graphen bearbeiten und numerische Werte im Kontext von Netzwerken analysieren und visualisieren (vergleiche dazu Abschnitt 3.2, Seite 33). Somit konnte bereits ein Teil der Anforderungen aus Abschnitt 3.3, Seite 39 verwirklicht werden. Hive ist in Java implementiert und ermöglicht durch Nutzung von *Java Webstart* [90] einfaches Installieren und automatische Verteilung neuer Updates, was insbesondere für biologische Anwender wichtig ist. Die Interaktion von Hive mit anderen Add-ons wie FBASimVis [49] und DBE [95] ist möglich und berücksichtigt: Sind andere Add-ons geladen, so werden weitere Mappingfunktionen bzw. Visualisierungsfunktionen aktiviert. Das Hive Add-on kann wiederum selbst durch Add-ons erweitert werden, um beispielsweise zusätzliche Mappingfunktionen und Visualisierungsalgorithmen einzubauen. Um die Anwendung zu testen und den Arbeitsablauf nachvollziehen zu können, existiert ein fünfminütiges Einstiegsvideo, ein Beispielprojekt und eine Dokumentation.

5.1.1 Der Entwicklungsprozess

Der Software-Entwicklungsprozess erfolgte nach dem evolutionären Modell ([8], Seite 56). Dieses erlaubt es, den vom Wasserfallmodell [126, 125] bekannten Softwareentwicklungsprozess mehrmals für Teilfunktionalitäten zu durchlaufen: Ausgangspunkt ist die Erhebung von Kernfunktionalitäten des Software-Systems. Diese Anforderungen wurden entsprechend des Wasserfallmodells realisiert und ausgeliefert. Basierend auf den Erfahrun-

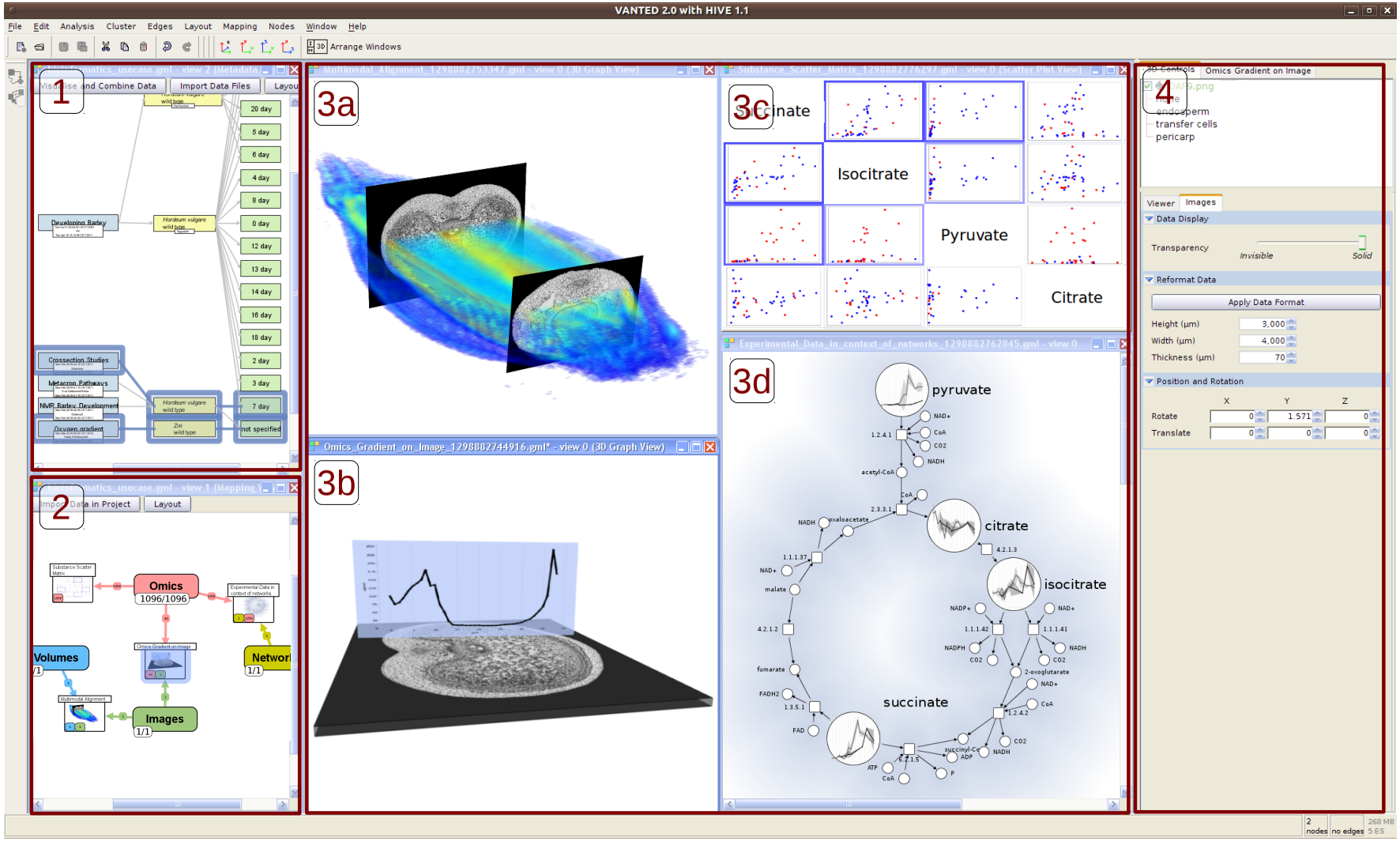


Abbildung 5.1: Bildschirmfoto der Anwendung HIVE. 1: MetadatenGraph. 2: MappingGraph (vergleiche auch Abbildung 5.2, Seite 74). 3: Vier Integrationssichten des Gerstenkorn-Datensatzes aus Abschnitt 6.3, Seite 85. 4: Bedienelemente, welche auf der aktuellen Integrationssicht arbeiten. Adaptiert von [120].

gen der Anwender mit der *Nullversion* genannten Software konnten weitere Anforderungen erhoben und als neue Funktionalität umgesetzt werden. Dieses Vorgehen erlaubt es, trotz unpräziser oder unvollständiger Anforderungen nahe an den Anwenderwünschen zu entwickeln und regelmäßig Produkte abliefern zu können. Dies ist wichtig, da sich die vollständige Erhebung konkreter Anforderungen zu Beginn der Entwicklung in Form eines Pflichtenheftes als schwierig erwies. Zielführender ist die regelmäßige Bereitstellung einer prototypischen Anwendung, die es potentiellen Anwendern erleichtert, durch Anschauen und Ausprobieren konkrete Anforderungen formulieren zu können. Durch die Entkopplung der Komponenten aufgrund der Plugin-Struktur von VANTED und HIVE konnte flexibel auf Änderungen seitens der Anwender reagiert und somit die iterativ auftretenden Anforderungen realisiert werden.

5.1.2 Datenimport und -speicherung

Der Import multimodaler Daten in das System erfolgt durch direkten Zugriff auf verschiedene Datenbanken wie METACROP [50], KEGG [70] und DBE [95], oder direkt aus dem Dateisystem, beispielsweise per Drag-and-Drop-Funktionalität. Es werden verschiedene Dateiformate unterstützt, z. B. GML, GraphML, SBML und KGML für *Netzwerke*, CSV Textdateien und Excel-Tabellen für *numerische Werte*, Analyze 7.5 und RAW-Format für *Volumen*, sowie PNG, TIFF und JPEG für *Bilder*. Die Segmentierungsinformation für *Bilder* und *Volumen* wird durch zusätzliche Grauwertbilder und -volumen beschrieben, in denen der Helligkeitswert die Zugehörigkeit der Originalpixel zu einem Segment festlegt. Alle diese Daten werden in einem vorher gewählten Projektverzeichnis abgelegt. Die Graphstrukturen speichern alle relevanten Informationen des Projektes, so dass Datenwerte, Metadaten und Mappings beim Laden wiederhergestellt werden können. *Numerische Werte* werden basierend auf den Datenintegrationsmöglichkeiten von VANTED in den Graphen direkt abgelegt, wohingegen größere Datenwerte wie *Bilder*, *Volumen* und *Netzwerke* als Verlinkung in das Dateisystem gespeichert werden. Werden Datenwerte z. B. wegen einer Visualisierung oder eines Mappings angefragt, werden diese aus dem Dateisystem in den Arbeitsspeicher geladen, die Aktion ausgeführt und anschließend alle Datenwerte wieder persistent im Dateisystem abgelegt. Versendet man das komplette Projektverzeichnis an andere Personen, so können diese das Projekt auf anderen Rechnern mit der HIVE-Anwendung im identischen Zustand wieder öffnen. Die Datenintegration erfolgt also in Form von Flat Files und realisiert damit direkt die Anforderung aus Abschnitt 3.1.3, Seite 32. Dadurch ergeben sich Vorteile wie hohe Performance, Schutz vor unbefugtem Zugriff und volle Kontrolle über die Daten. Es existieren aber auch Nachteile wie ausschließlich lokaler Zugang zum System und den Daten, sowie Konsistenzprobleme beim Abgleich mit Datenquellen.

5.1.3 Oberfläche und visuelle Hilfen

Zur Unterstützung der Navigation durch integrierte Daten werden verschiedene Hilfsmittel in der Anwendungsoberfläche genutzt, die im Folgenden erläutert werden:

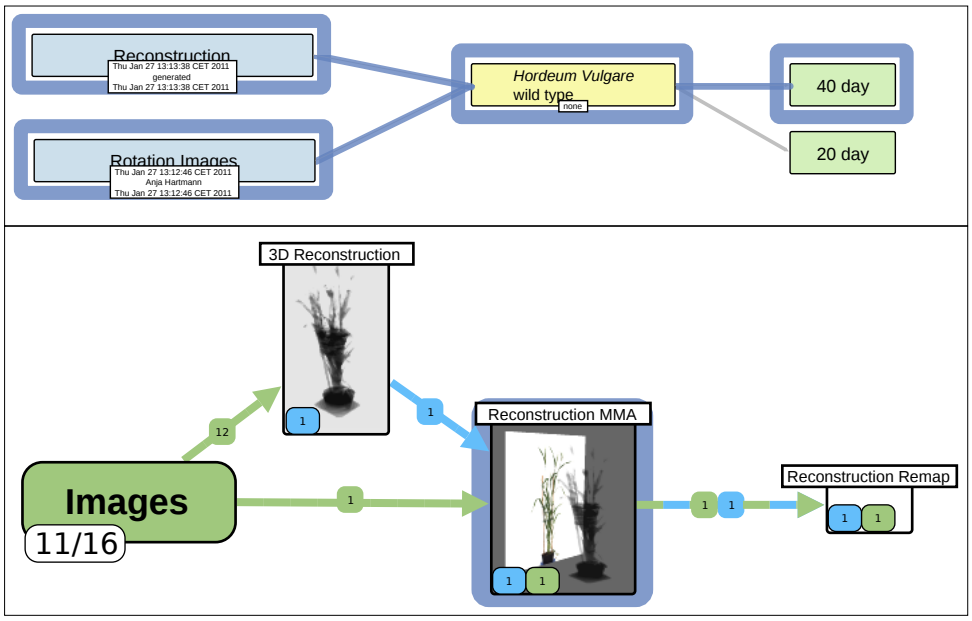


Abbildung 5.2: Übersicht über visuelle Hilfen. MetadatenGraph (oben): Metadaten aller in Integrationssichten visualisierten Datenwerte werden umrahmt. MappingGraph (unten): Alle Mappingknoten, deren Mapping in einer Integrationssicht visualisiert werden, erscheinen umrahmt. Anzahl der Datenwerte in den DWI- und Mappingknoten und Anzahl der Eingabe-Datenwerte einer Mappingfunktion werden durch Zahlen repräsentiert. Ein Farbcode lässt Rückschlüsse auf den Typ der integrierten Datenwerte zu.

Einheitlicher Farbcode für Datenwerttypen Jedem der vier Datenwerttypen wurde eine Farbe zugewiesen: *Numerische Werte* (rot), *Bilder* (grün), *Volumen* (blau) und *Netzwerke* (gelb). Viele Elemente der Oberfläche erlauben die Einfärbung nach diesem Schema, so dass die Typen der involvierten Datenwerte auf einen Blick erfassbar sind (siehe Abbildung 5.2). DWI-Knoten sind beispielsweise nach dem Typ der integrierten Datenwerte gefärbt. Dasselbe gilt für Kanten, welche den Datenwert-Fluss repräsentieren. Falls mehrere Typen involviert sind, so werden Kanten abwechselnd gestrichelt dargestellt. Auch die Auswahllisten der Mappingfunktionen und Visualisierungsfunktionen benutzen den Farbcode, um die Eingabe-Datenwerttypen anzudeuten.

Darstellung der repräsentierten Datenwerte Im MappingGraph repräsentieren alle Knoten und Kanten eine Menge von Datenwerten. Die Anzahl und Typen der Datenwerte können auf einen Blick erfasst und der Datenwertfluss nachverfolgt werden (vergleiche Abbildung 5.2): DWI-Knoten visualisieren die verfügbaren und die integrierten Datenwerte. Mappingknoten repräsentieren Mappings, welche eine Menge von Datenwerten kombinieren. Die Anzahl der Datenwerte wird, aufgeschlüsselt nach Datenwerttyp, durch umrahmte Zahlen dargestellt. Kanten stellen den Fluss der Datenwerte aufgrund der Anwendung von Mappingfunktionen dar und analog zu den Knoten werden auch hier Zahlen angezeigt.

Verstecken von Mappingknoten Neben dem in Abschnitt 4.2.4, Seite 52 definierten Verstecken von Metadaten-Objektknoten zur Reduktion der Arbeitsmengen, können im MappingGraph auch Mappingknoten versteckt werden. Dieses Verstecken dient ausschließlich der Reduktion sichtbarer Knoten und Kanten zwecks Übersichtlichkeit und hat keinen weiteren Einfluss auf Arbeitsmengen.

Herkunft visualisierter Datenwerte Navigation und Exploration kombinierter biologischer Datenwerte erfordert die parallele Arbeit mit verschiedenen Integrationssichten. Um einen Überblick über die in einer Integrationssicht dargestellten Datenwerte zu behalten, werden Markierungen im MetadatenGraph und MappingGraph genutzt: Blaue Rahmen markieren den visualisierten Mappingknoten und alle Objektnoten im MetadatenGraph, die die Metadaten der im Mapping enthaltenen Datenwerte repräsentieren (siehe links oben in Abbildung 5.1, Seite 72). Ist eine Integrationssicht geöffnet, aber gerade nicht aktiv, so ändert sich die Farbe des Rahmens zu einem Grauton.

Aufteilung des Anwendungsfenster Für den bestmöglichen Überblick über die Datenintegration und geöffnete Integrationssichten ist das Anwendungsfenster in drei Bereiche aufgeteilt (siehe Abbildung 5.1, Seite 72): MetadatenGraph bzw. MappingGraph werden im linken Drittel übereinander angeordnet, das mittlere Drittel beherbergt die geöffneten Integrationssichten und das rechte Drittel besteht aus den Bedienelementen, welche Interaktionen mit allen Sichten erlauben. Diese Anordnung ermöglicht es zu jedem Zeitpunkt, einen Überblick über alle integrierten und visualisierten Daten und den Arbeitsfluss im MappingGraph behalten zu können. Trotzdem kann auf Wunsch die Anordnung durch Maximieren, Minimieren und Verschieben der Fenster jederzeit verändert werden.

5.1.4 Implementierung der 3D-Visualisierung

In Abschnitt 4.4, Seite 59 wurden wichtige Visualisierungsfunktionen der Methodik beschrieben. Den größten Leistungsumfang realisieren dabei die Graph-Visualisierung und die 3D-Visualisierung. Da die Implementierung der Graph-Visualisierung größtenteils aus VANTED übernommen werden konnte, musste vor allem eine funktionsreiche 3D-Visualisierung implementiert werden. Volumenrendering in Java ist zwar nach wie vor aufgrund der Arbeitsspeichereffizienz nicht für sehr hoch aufgelöste Datensätze geeignet, konnte aber in den letzten Jahren diesbezüglich deutlich verbessert werden [25]. Es existieren einige Anwendungen bzw. Bibliotheken, welche Volumenrendering in Java oder Java3D anbieten (vergleiche auch Abschnitt 2.5.2.4, Seite 27):

VolRend ist ein von SUN entwickelter Volumenrenderer, welcher neben dem Schnittbasierten Modus auch 3D Texturen unterstützt. Neben dem Manipulieren der Farbpalette existieren aber keine weiteren Interaktionsmöglichkeiten [47].

BIL-Kit ist eine hauptsächlich auf Bildverarbeitung fokussierte Anwendung, welche auch Volumen rendern kann. Weder ist die Anwendung, noch der Quellcode verfügbar [60].

Fiji basiert auf ImageJ. Neben vielfältigen Bildbearbeitungsroutinen wurde ein schneller Schnitt-basierter Volumenrendering-Algorithmus entwickelt, welcher aber nur wenige Interaktionsmöglichkeiten bietet [130].

Vol<X>Rend ist ein Volumenrenderer zum Remote-Rendering von sehr großen Volumendaten. Es fehlen sowohl Testdaten als auch der Quellcode [128].

BrainImageJ ist nie aus dem Entwicklungsstadium herausgekommen und basiert auf der inzwischen veralteten Java-Bibliothek GL4Java [162].

RTVR ist eine Java-Bibliothek zum gleichzeitigen Rendern von Oberflächen- und Volumendaten mittels des Shear-Warp-Algorithmus. Die Anwendung und der Quellcode sind allerdings nicht verfügbar [99].

Spectus3D ist ein Quellcode-freier Schnitt-basierter Volumenrenderer. Die Darstellung mittelgroßer Volumen (<50 Millionen Voxel) ist flüssig und bietet vielfältige Interaktionsmöglichkeiten [94].

Neben den genannten existiert eine große Zahl von Anwendungen, welche Volumen handhaben, aber nicht in 3D rendern können. Oft werden stattdessen zweidimensionale Darstellung (Bilderstapel) oder Oberflächendarstellungen gewählt. Beispiele für diese Anwendungen sind ImageJ [1], Shiva [38], JAtlasView [40], VisAd [52] und Mindseer [98]. Daneben existieren zahlreiche Volumenrenderer, welche nicht in Java geschrieben oder kommerziell sind. Beispielhaft seien hier Paraview [85], SCIRun [106], Voxx2 [109] Covise [114], VTK [134] und openDX [156] genannt.

Zusammenfassend sprach für Spectus3D die Verfügbarkeit und Übersichtlichkeit des Quellcodes, die hohe Funktionsvielfalt und einfache Erweiterbarkeit. Der im Rahmen einer Masterarbeit entstandene Renderer wurde in Java bzw. Java3D realisiert und fügt sich deswegen nahtlos in den Rest der Anwendung ein. Neben der Verbesserung der Ladezeiten und des Speicherbedarfs wurden die Interaktionsmöglichkeiten erweitert, Unterstützung von Segmentierungsinformation hinzugefügt und stereoskopische Ausgaben implementiert (z. B. Anaglyphen und Polarisation) [112].

5.1.5 Skalierbarkeit der Anwendung

Die Integration der Daten erfolgt im Dateisystem und ist entsprechend flexibel erweiterbar und performant. Einzige Ausnahme sind die *numerischen Werte*, welche direkt im Integrationsgraphen gespeichert werden. Durch die notwendige Serialisierung des Integrationsgraphen bei Speicherung des Projektes müssen auch alle Datenwerte serialisiert werden. Das Speichern und Laden des Integrationsgraphen kann deshalb relativ lange dauern, beispielsweise benötigt ein kompletter Genexpressionsdatensatz (100.000 Werte) ca. eine Minute Ladezeit. Da dies die Grenze zur Interaktivität bereits überschreitet, sollten hier noch Verbesserungen erfolgen.

Die Skalierbarkeit der Anwendung hängt besonders stark von den Datenwerttypen und deren Visualisierungsart ab. Metabolit- bzw. Proteindaten umfassen pro Experiment üblicherweise nur wenige 100-1000 Datenwerte, welche selbst bei Integration einer

großen Zahl von Experimenten performant visualisierbar sind. Genexpressionsdaten hingegen können durchaus aus mehreren Hunderttausend Datenwerten bestehen. Die Handhabung ist mit ausreichend Speicher (> 1GB RAM) möglich, die Darstellung aller Einzelwerte aber sehr rechenintensiv und entsprechend langsam. Moderne Next-Generation-Sequencing-Daten können aufgrund ihres Umfanges nicht gehandhabt und analysiert werden. Experimente zur Erhebung von *Bildern* umfassen derzeit üblicherweise < 100 *Bilder* mit jeweils wenigen Millionen Pixel, welche relativ problemlos in 2D und auch 3D darstellbar sind. Steigt die Pixelzahl stark, wie zum Beispiel bei der Elektronenmikroskopie (> 100 Millionen Bildpunkte), kann dies zu Geschwindigkeits- und Speicherproblemen führen. Die Visualisierung von *Volumen* beansprucht besonders viele Ressourcen, insbesondere Hauptspeicher und Grafikkartenleistung. Grund ist neben der hohen Zahl an Voxeln (10-50 Millionen pro *Volumen*) die Transparenz des volumetrischen Körpers, welche dazu führt, dass alle Voxel zu jedem Zeitpunkt dargestellt werden müssen. Die Darstellung von *Netzwerken* ist bis zu 50.000 Knoten möglich. Kommen Diagrammen hinzu, reduziert sich die Zahl auf 10.000 Knoten. Die Darstellung in 3D ermöglicht die gleichzeitige Darstellung von bis zu 5.000 Knoten.

Die Größe des MetadatenGraph wächst relativ langsam, da die Zusammenführen-Operation angewandt (beschrieben in Algorithmus 2, Seite 51) und Datenwert- und *Messgröße*-Objektknoten nicht dargestellt werden. Da in der Methodik Arbeitsmengen definiert wurden, ist es den Anwendern im Arbeitszyklus möglich, temporär uninteressante Datenwerte zu verstecken. Überlappende Kanten im MappingGraph können durch Verstecken der Mappingknoten vorübergehend aus der Visualisierung entfernt werden. Dennoch ist aus Gründen der Übersichtlichkeit nicht zu empfehlen, über Monate oder gar Jahre hinweg Datenwerte und Mappings zu akkumulieren. Die Handhabung und persistente Datenhaltung von hunderten oder gar tausenden Experimenten ist nicht das Ziel dieser Arbeit, kann aber z. B. durch die Verbindung zur DBE-Datenbank [95] realisiert werden.

Der zu erwartende Anstieg der Datenquantität kann teilweise durch Optimierung der Geschwindigkeit und des Speicherbedarfes des unterliegenden Systems VANTED abgefangen werden. Aufgrund vieler verschiedener Sichten, die gleichzeitig geöffnet sein können, kann eine hohe Bildschirmauflösung die Arbeit mit der Anwendung deutlich erleichtern.

5.2 Fazit

Die Methodik konnte erfolgreich auf Basis von VANTED und SPECTUS3D als lokale Java-Anwendung HIVE umgesetzt werden. Daten können aus dem Dateisystem oder durch direkte Anbindung an Datenbanken in ein Projektverzeichnis importiert werden, wodurch hohe Performance und Datensicherheit möglich sind. Zur Unterstützung des Anwenders bei der Datenintegration und -kombination wird die Oberfläche mit Hilfsmitteln wie einem einheitlichen Farbcode versehen. Für die vorgesehenen Datenmengen und -typen ist die Skalierbarkeit der Anwendung derzeit ausreichend.

6

Anwendung

6.1 Arabidopsis Blütenentwicklung

Die Entwicklung von Blütenorganen in der Modellpflanze *Arabidopsis thaliana* ist zum großen Teil durch Expression verschiedener Genklassen determiniert. Das von Theißen und Saedler [152] vorgestellte Modell beschreibt fünf verschiedene Genklassen (A, B, C, D und E), die in bestimmten Kombinationen zur Ausprägung der einzelnen Blütenorgane führen (vergleiche linken Teil der Abbildung 6.1). Beispielsweise führt die Expression von Genen der Klassen A, B und E zur Blütenblatt-Entwicklung, C und E hingegen zur Entwicklung von Staubblättern. Die Blütenentwicklung ist ein intensiv studiertes Thema, weswegen sowohl umfangreiche und detaillierte Genexpressionsdatensätze verfügbar sind, als auch verlässliche Informationen über genregulatorische Netzwerke existieren. Auf Basis dieser Daten wird im Folgenden die Integrationssicht **Omics Distribution by Network** angewandt, um das Blütenmodell zu validieren (vergleiche auch Tabelle 6.1, Seite 90).

Die Integrationssicht ermöglicht es, Segmente von *Bildern* auf Basis räumlich aufgelöster *numerischer Werte* anzufärben, in Netzwerkknoten zu integrieren und damit die räumlichen Genexpressionsmuster im Kontext der *Netzwerke* visuell analysieren zu können. Dazu ist die Eingabe räumlich aufgelöster *numerischer Werte*, eines segmentierten *Bildes* und eines oder mehrerer *Netzwerke* erforderlich. Es wird ein Mapping erzeugt, welches alle diese Datenwerte kombiniert: Die Mappingfunktion integriert alle *numerischen Werte* in die jeweiligen Netzwerkknoten, für die die Bezeichnung des Knotens und der *Messgröße* des *numerischen Wertes* übereinstimmen. Der Anwender benennt die Segmente des *Bildes*, um eine Zuordnung der Segmente zu den *numerischen Werten* zu ermöglichen. Diese Zuordnung ist in $\mathbb{T}$ gespeichert, zusammen mit dem Farbcode. Visualisiert wird das Mapping mittels der Omicsverteilung-Visualisierung. Die resultierende Integrationssicht besteht aus zwei Teilen: im linken Teil werden angefärbte Bilder dargestellt und im rechten Teil die um die *numerischen Werte* angereicherten *Netzwerke*. Der Anwender kann nun

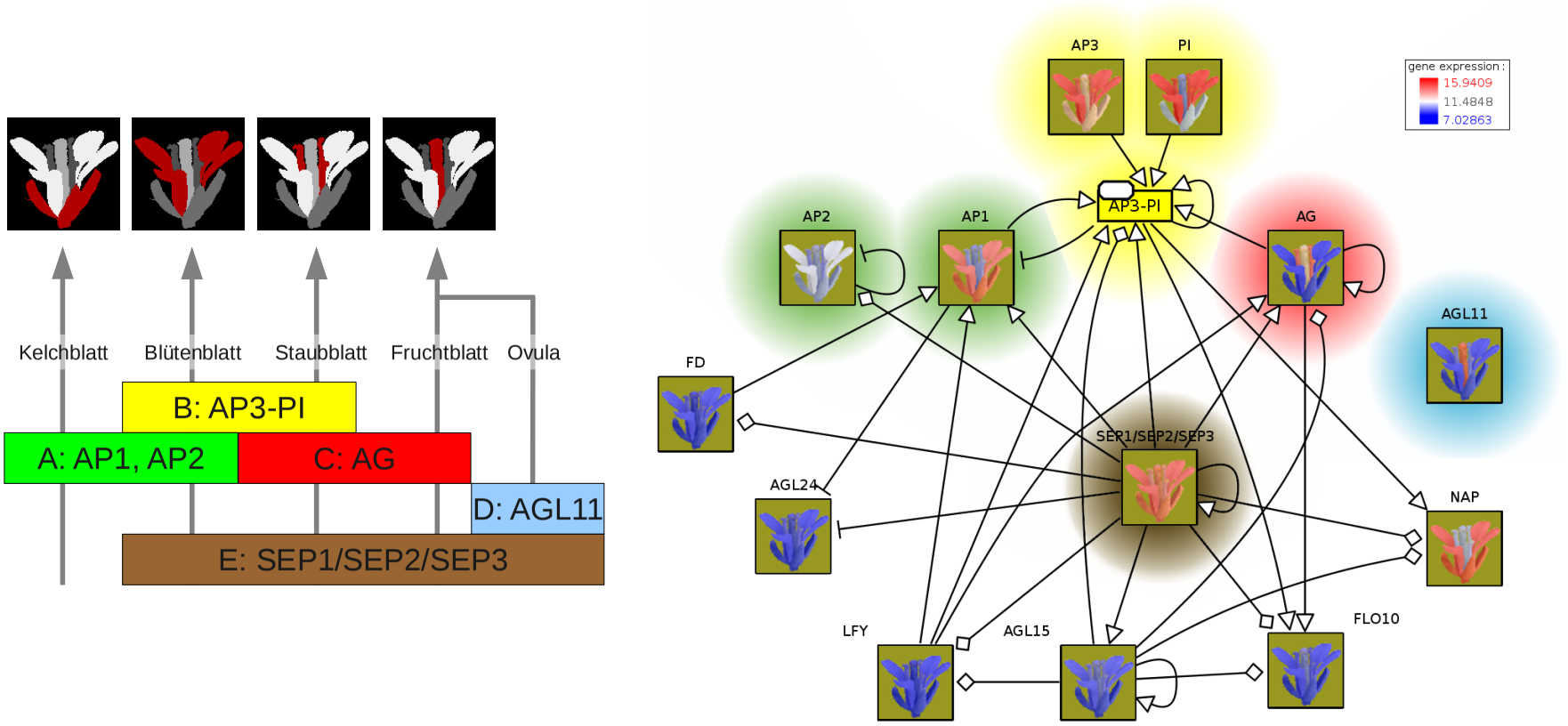


Abbildung 6.1: Validierung des ABCDE-Modells [152] der Arabidopsis Blütenentwicklung mittels der Integrationssicht **Omics Distribution by Network**. Links: Modell, welches die Entwicklung des jeweiligen Blütenorgans durch Kombination verschiedener Genklassen beschreibt. Rechts: regulatorisches Netzwerk der in der Blütenentwicklung beteiligten Gene aus AGRIS [163]. Die Segmente eines Blütenbildes wurden basierend auf räumlich aufgelösten Genexpressionsdaten aus GENEVESTIGATOR [164] farbkodiert und in die Knoten des Netzwerkes integriert, um die Genexpression im Kontext des Modells und des regulatorischen Netzwerkes zu visualisieren (Farbcode ist logarithmisch; <10 ist geringe Expression, 10-12 entspricht durchschnittlicher Expression). Die Expression der ABCDE-Gene stimmt mit den Vorhersagen des Modells überein. FD, AGL24, LFY, AGL15 und FLO10 repräsentieren übergeordnete Regulatoren, welche für die Blüteninduktion wichtig sind.

mittels der Interaktionstechnik Brushing einen Knoten im Netzwerk auswählen, wodurch im linken Teil ein Bild generiert und dargestellt wird. Das Bild wird aus dem Ursprungsbild erzeugt, indem für jedes Segment des Ursprungsbildes alle im Knoten verfügbaren Datenwerte extrahiert werden, welche diesem Segment zugeordnet sind. Alle betroffenen Werte werden verrechnet und das Segment des neuen Bildes entsprechend des globalen Farbcodes eingefärbt. Das farbkodierte Bild wird im linken Teil der Integrationssicht dargestellt. Selektiert der Anwender mehrere Knoten, werden im linken Teil der Integrationssicht entsprechend mehrere Bilder dargestellt. Der Anwender kann auf Wunsch diese generierten Bilder auch in die jeweiligen Knoten integrieren, um eine statische Visualisierung der Netzwerkstruktur und des Genexpressionsmusters zu erhalten (vergleiche rechten Teil der Abbildung 6.1). Die für diesen Anwendungsfall genutzten Daten umfassen folgende Datenwerte:

Bilder: ein repräsentatives Foto einer Arabidopsis-Blüte mit vier Segmenten: Kelchblätter, Blütenblätter, Staubblätter und Stempel (entnommen aus [152][1])

Netzwerke: ein genregulatorisches Netzwerk der Blütenentwicklung von Arabidopsis aus AGRIS [163]

numerische Werte: Genexpressionswerte einer ausgebildeten Arabidopsis-Blüte für alle im Netzwerk enthaltenen Gene und vier Blütenorgane, entnommen aus GENEVESTIGATOR [164]

Im rechten Teil der Abbildung 6.1 ist das resultierende Netzwerk dargestellt. Deutlich zu sehen ist, dass das Expressionsmuster für die fünf Genklassen (Knoten unterlegt mit der jeweiligen Klassenfarbe) größtenteils mit dem Modell übereinstimmt. AG ist, wie vorhergesagt, nur in Staubblatt und Fruchtblatt, AGL11 nur im Stempel und AP1 nur in den Kelch- und Blütenblättern exprimiert. Der Komplex bestehend aus AP3 und PI ist sehr stark in Staub- und Blütenblatt aktiv, AP3 zeigt aber auch eine etwas geringere Expression in anderen Organen. AP2 ist in Blütenblättern vorhanden, aber entgegen der Vorhersagen des Modells zeigt sich zum Zeitpunkt der Messungen keine Expression im Kelchblatt, da AP2 nur zu Beginn der Blütenentwicklung im Kelchblatt exprimiert wird [160]. SEP1, SEP2 und SEP3 sind funktionell redundante Gene [107], welche in allen Organen sehr aktiv sind. Die Expression der SEP-Gene hat großen regulatorischen Einfluss auf die Bildung aller Organe, außer auf die der Kelchblätter [41]. AGL11 akkumuliert während der Blütenentwicklung wie vorhergesagt ausschließlich in der Ovula (hier repräsentiert durch das Stempelsegment). NAP ist ein Zielgen des AP3-PI-Komplexes und weist entsprechend ein ähnliches Expressionsmuster auf.

Das ABCDE-Modell der Blütenentwicklung konnte erfolgreich mit Genexpressionsdaten unter Nutzung der Integrationssicht validiert werden. Alle weiteren Gene (FD, AGL24, LFY, AGL15 und FLO10) sind für die Validierung nicht essentiell. Sie repräsentieren verschiedene übergeordnete Regulatoren, welche für den Übergang von der vegetativen Phase zur Blütenentwicklung wichtig sind (Blüteninduktion). In der ausgebildeten Blüte

[1] Urheber Jürgen Berger, Max-Planck-Institut für Entwicklungsbiologie, Tübingen

werden diese aber kaum exprimiert. Die Visualisierung des Netzwerkes zusammen mit den räumlich aufgelösten Genexpressionsdaten ermöglicht somit eine visuelle Unterscheidung zwischen Genen der Phasen Blüteninduktion und Blüte. Es ist weiterhin möglich, die regulatorischen Interaktionen im Netzwerk mit Hilfe von Expressiondaten zu validieren. Hierzu müssen allerdings zusätzlich zu den in Abbildung 6.1, Seite 80 gezeigten Wildtyp-Expressionsdaten entsprechende Daten von Mutanten-Linien hinzugezogen werden. Ein Vergleich dieser Daten, insbesondere eine veränderte Expression der Zielgene (z. B. Blütengene AP1, AP2, AP3-PI, AG) vor dem Hintergrund eines mutierten Regulatorgenes (z. B. übergeordnete Regulatoren FD, LFY), kann die regulatorische Interaktion bestätigen. Dies ist beispielsweise bei der Regulation von AP3 durch LFY der Fall: Wird LFY ausgeknockt, senkt sich die Expressionsrate von AP3 [83]. Solche Unterschiede in den Genexpressionsmustern basierend auf Knockouts oder Überexprimierung spiegeln sich visuell sofort im Netzwerk wieder und können somit regulatorische Interaktionen bestätigen. Unerwartet auftretende Expressionsänderung weiterer Gene des Netzwerkes können Rückschlüsse auf bisher unbekannte Beziehungen ermöglichen.

Die Integrationssicht **Omics Distribution by Network** ermöglicht es folglich, Genexpressionsdaten mit räumlicher Auflösung im Kontext regulatorischer Informationen zu visualisieren, analysieren und validieren. Grundlage für weiterführende Analysen ist die Verfügbarkeit räumlich aufgelöster Expressionsdaten für Knockout-Pflanzen, die derzeit noch nicht in ausreichender Menge vorliegen.

6.2 Netzwerk-gestützte Navigation durch Drosophila-Bilddatenbanken

Große Mengen von Bilddaten werden oftmals in verteilten und sehr spezialisierten Datenbanken abgelegt, auf die externe Anwender freien oder zumindest eingeschränkten Zugriff besitzen (beispielsweise [3, 12, 44, 57, 89, 92, 117]). Dieser Zugriff erfolgt üblicherweise über ein Webinterface, welches die Suche einzelner Bilder in Listen oder aufgrund von Attributen wie Entwicklungsstadium, genetischer Veränderung oder Substanznamen ermöglicht. Es ist allerdings sehr zeitaufwändig, durch die Daten zu navigieren und diese herunterzuladen. Der Vergleich von Bildern mit verschiedenen Attributen ist nicht vorgesehen, stattdessen müssen die Bilder einzeln heruntergeladen und dann mit externen Anwendungen verglichen werden. Auch der Vergleich von Bildern aus verschiedenen Datenbanken, Berücksichtigung der Beziehungen zwischen Substanzen (z. B. Regulation) und weitergehender Informationen (z. B. Expressionsdaten) werden nicht berücksichtigt. Um Anwender bei der Exploration und Visualisierung solch großer Datenmengen zu unterstützen, sollte die Navigation der Bilder durch Navigation in biologischen Netzwerken möglich sein. Im Folgenden wird deswegen die Integrationssicht **Image Browsing by Network** angewandt, um die Inhalte zweier *Drosophila melanogaster*-Bilddatenbanken mittels eines um Genexpressionswerte angereicherten biologischen Netzwerkes explorieren, visualisieren und analysieren zu können (vergleiche auch Tabelle 6.1, Seite 90).

Die Integrationssicht ermöglicht es, *Netzwerke* mit einer Menge von *Bildern* zu verknüpfen und mittels Brushing-Interaktionstechnik explorier- und vergleichbar zu machen. Dazu ist die Eingabe eines oder mehrerer *Netzwerke* und einer (evtl. sehr großen) Menge funktioneller *Bilder* erforderlich. Funktionelle *Bilder* sind solche, auf denen die zweidimensionale Verteilung einer *Messgröße* sichtbar ist (vergleiche Abschnitt 2.2.3, Seite 13). Die Zuordnung von *Bildern* zu *Messgrößen* kann während der Integration der Daten oder beim Ausführen der Mappingfunktion durch den Anwender spezifiziert werden. Die Mappingfunktion verknüpft die *Bilder* basierend auf dieser *Messgröße* mit den Knoten in den Netzwerken, welche dieselbe *Messgröße* repräsentieren. Visualisiert wird das Mapping mittels der Bildnavigations-Visualisierung. Die daraus resultierende Integrationssicht besteht aus zwei Teilen: im linken Teil wird eine Bildermatrix und im rechten Teil die eingegebenen Netzwerke dargestellt. Der Anwender kann nun mittels der Interaktionstechnik Brushing einen oder mehrere Knoten im *Netzwerk* auswählen, wodurch im linken Teil alle *Bilder* dargestellt werden, die mit den entsprechenden *Messgrößen* verknüpft sind (vergleiche Abbildung 6.2). Dazu wird eine Bildermatrix erzeugt, die durch die *Messungen* und alle ausgewählten *Messgrößen* aufgespannt wird. Jedes Element der Matrix besteht demzufolge aus einer Menge von *Bildern*, welche eine biologische Substanz (z. B. Metabolitkonzentration, Genexpression) zu einem bestimmten Entwicklungszustand darstellen. Falls kein *Bild* existiert, bleibt das Element leer und falls mehrere *Bilder* existieren, kann der Anwender die *Bilder* nacheinander durch direkte Interaktionen anzeigen lassen. Zusätzlich ist es möglich, für jede *Messgröße* Verlinkungen hinzuzufügen, um beispielsweise direkt aus der Anwendung zu den Originaldaten in der Datenbank-Webseite oder zu informativen Einträgen anderer Datenbanken wie UniProt zu springen. $\mathbb{T}$ enthält diese URLs als Attribute. Alternativ könnten ähnlich zu dem ersten Anwendungsfall alle mit einem Knoten verknüpften *Bilder* auch in den Knoten visualisiert werden, was allerdings bei einer größeren Menge verknüpfter und evtl. hochdetaillierter *Bilder* zu einer unübersichtlichen Netzwerkvisualisierung führen würde.

Die für diesen Anwendungsfall genutzten Daten umfassen folgende Datenwerte:

Bilder: ca. 3000 GFP- oder *in situ*-Bilder des Drosophila Embryos aus der FruitFly-Datenbank [44] und der Fly-FISH-Datenbank [42]

Netzwerke: ein PPI-Netzwerk von Drosophila von Stuart *et al.* [144]

numerische Werte: Genexpressionsdaten der prä-larvalen Entwicklungsstadien 7, 12 und 17 (3h, 8h und 15h nach der Befruchtung) aus Genevestigator [164]

Ausgangspunkt ist das im DWI-Knoten *Network* enthaltene *Netzwerk*, welches in der Graph-Visualisierung manuell um Annotationen erweitert wird. Diese Annotationen weisen den Knoten des PPI-Netzwerkes kodierende Gene zu. Alle Knoten ohne Zuordnung werden aus dem *Netzwerk* entfernt, wodurch sich die Zahl der Knoten auf 800 reduziert. Dieses *Netzwerk* und die Genexpressionsdaten sind die Eingabe einer Mappingfunktion, welche die Expressionsdaten mit den Netzwerkknoten verknüpft. Die Darstellung des Mappings erfolgt mit der Graph-Visualisierung, so dass in der resultierenden Integrationssicht

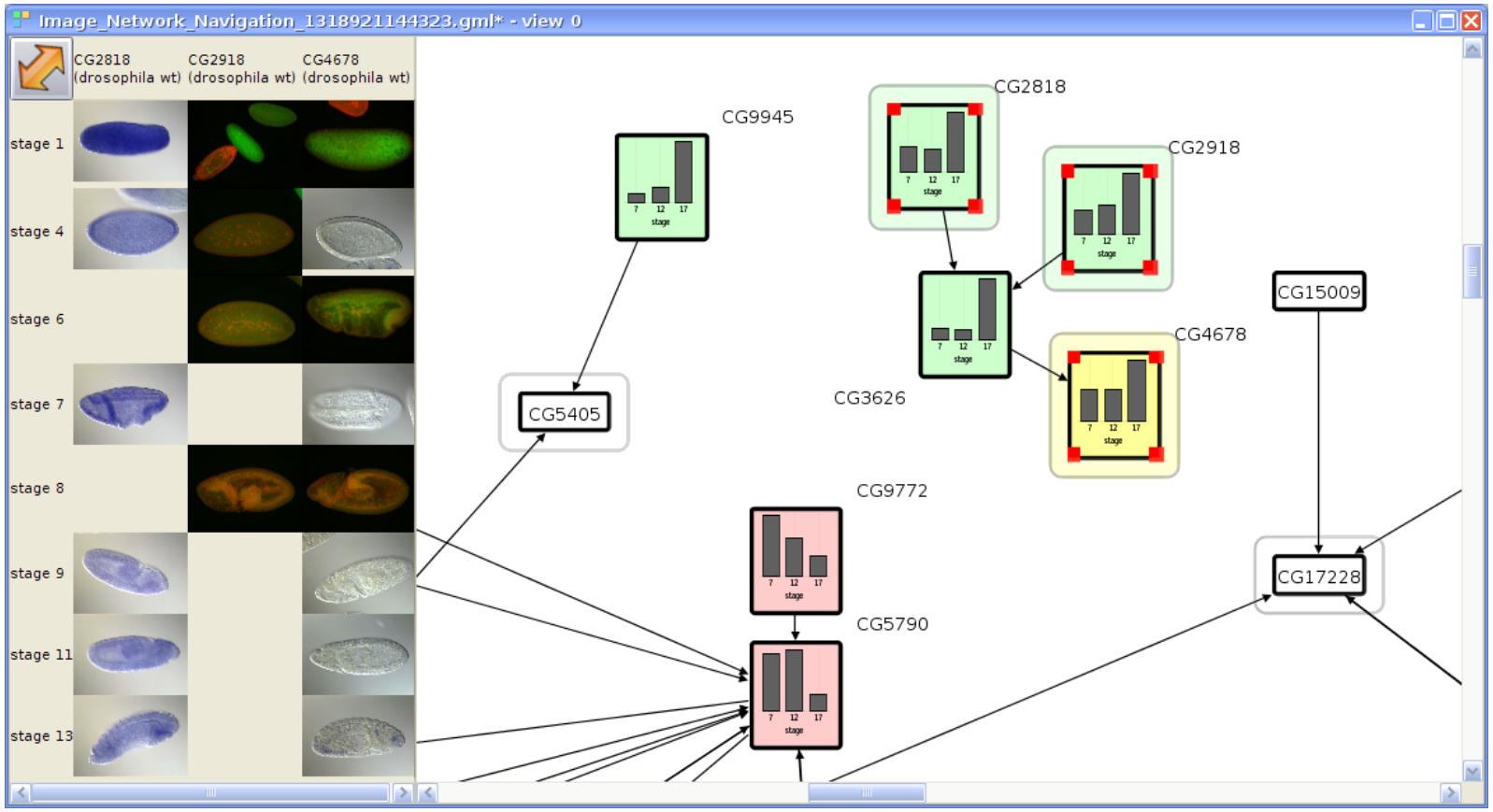


Abbildung 6.2: Bildschirmfoto der Integrationssicht **Image Browsing by Network** mit Bildern zweier Drosophila-Bilddatenbanken [42, 44] im Kontext eines PPI-Netzwerkes von Stuart *et al.* [144]. Das Netzwerk wurde um zeitlich aufgelöste Genexpressionsdaten aus GENEVESTIGATOR [164] angereichert, visualisiert durch Balkendiagramme in den Knoten („stage 7“, „stage 12“ und „stage 17“). Mittels einer Self-Organizing Map wurden drei verschiedenen Expressions-Entwicklungsmuster detektiert (repräsentiert durch die Farben rot, grün und gelb). Alle umrahmten Knoten sind mit Bilddaten verknüpft, die im linken Teil visualisiert werden können. Diese Bildermatrix wird durch die selektierten Knoten und die Entwicklungsstadien aufgespannt. Die Integrationssicht unterstützt somit die Navigation durch und Vergleichbarkeit von großen Bilddatenmengen im Kontext von Netzwerken und numerischen Werten.

Omics Network Context die zeitliche Entwicklung der entsprechenden Gene als Balkendiagramm in den Knoten dargestellt ist (von links nach rechts „stage 7“, „stage 12“ und „stage 17“). Dieses angereicherte *Netzwerk* und die 3000 *Bilder* aus beiden Datenbanken dienen als Eingabe für eine weitere Mappingfunktion, welche die Netzwerkknoten mit den entsprechenden *Bildern* verknüpft. Die Visualisierung des resultierenden Mappings erfolgt mit der oben beschriebenen Bildnavigation-Visualisierungsfunktion, resultierend in der Integrationssicht **Image Browsing by Network**. Der Anwender kann somit leicht durch die 3000 *Bilder* navigieren und einen schnellen Überblick über die räumliche Verteilung der Gene bekommen. Knoten können, wie links in Abbildung 6.2 zu sehen, auch mit *Bildern* verschiedener Quellen oder Anfärbemethoden verknüpft werden. Die Genexpressionsdaten im Netzwerk erlauben es weiterhin, *numerische Werte* und *Bilder* im Kontext des Netzwerkes zu explorieren und zu analysieren. In obiger Abbildung wurde dazu eine Self-Organizing Map [43] angewandt, um Gene mit ähnlicher zeitlicher Entwicklung zusammenzufassen. Die Analyse ergab drei Cluster, welche über die Zeit zunehmende Expression (Cluster 1, grün), abnehmende Expression (Cluster 2, rot) und anderes Expressionsverhalten (Cluster 3, gelb) beschreiben. Eine GO-Term-Enrichment-Analyse dieser Klassen mittels AmiGO [18] ergab, dass die Gene des Clusters 1 hauptsächlich im Nukleus RNA prozessieren (z. B. Splicing), während die des Clusters 2 eher in den Mitochondrien bzw. deren Membranen am Abbaumetabolismus beteiligt sind. Gene des dritten Clusters sind sehr unterschiedlichen funktionellen Kategorien zuordenbar. All diese Informationen können nun mit den Entwicklungsreihen der Bilder verglichen werden und somit die Analyse der qualitativen Bilder unterstützen.

Die Integrationssicht konnte zusätzlich auch für die EMAGE-Datenbank [117] angewandt werden, die allerdings deutlich weniger und vergleichbar diverse Bilddaten von Maus-Embryonen enthält. Derzeit ist es nicht vorgesehen, solche Bilddatenbanken direkt anzusteuern, da meist eine programmatische Abfrage der Daten via API oder Webservices nicht vorgesehen ist. Stattdessen müssen die Daten per Skript heruntergeladen und manuell importiert werden.

Bei der Interpretation der Genexpressionsdaten und funktionellen Bilder im Kontext des PPI-Netzwerkes sollte berücksichtigt werden, dass interagierende Proteine nicht zwangsläufig Schlussfolgerungen auf ähnliche Expressionsmuster der Gene erlauben. Erst die Nutzung eines Koexpressionsnetzwerkes bzw. genregulatorischen Netzwerkes erlaubt diese Analysen. Sobald ein solches Netzwerk für *Drosophila melanogaster* veröffentlicht wird, sollte eine erneute Analyse der Bilddaten im Kontext des Netzwerkes die Generierung detaillierter Erkenntnisse ermöglichen.

6.3 Multimodaler Datensatz des Gerstenkorns

Gerste (*Hordeum vulgare*) ist vor allem für die Futter- und Brauindustrie eine wichtige Getreidepflanze. Folgerichtig werden eine Vielzahl interdisziplinärer Projekte durchgeführt, welche die Untersuchung der Kulturpflanze aus einer System-basierten Sicht zur Erhöhung

des Ertrages und Stärkung der Resistenz gegen Krankheiten zum Ziel haben. Mit der vorgestellten Methodik sollen im Folgenden unterschiedliche Daten des sich entwickelnden Gerstenkorns aus verschiedenen Experimenten integriert, kombiniert und visualisiert werden (vergleiche auch Tabelle 6.1, Seite 90). Die Visualisierungen können sehr schnell mit der Anwendung erzeugt werden und ermöglichen ein intuitives Verständnis der im Kontext stehenden Daten. Der Datensatz besteht aus:

Bilder: zwei histologische Schnittbilder das Entwicklungsstadiums 9 days after fertilisation (DAF), segmentiert nach den Geweben Perikarp, Endosperm und Embryo

Netzwerke: ein Koexpressionsnetzwerk von Mochida *et al.* [96], ein metabolischen Netzwerk des Zitronensäurezyklus (TCA-Zyklus) aus MetaCrop [50]

numerische Werte: Genexpressionsdaten aufgelöst nach Geweben (Perikarp, Endosperm, Embryo) aus Genevestigator [164], Metabolit-Daten des Primärstoffwechsels (0 bis 20 DAF), ein Sauerstoff-Konzentrationsgradient von 9 DAF [123]

Volumen: ein NMR-Volumen der Wasserverteilung von 9 DAF aus [48]

Die Integrationssicht **Multimodal Alignment** visualisiert die Kombination des NMR-Volumens und der histologischen Schnittbilder (siehe Abbildung 5.1, Seite 72, Teil 3a). Die hochauflösende strukturelle Information der Schnittbilder kann im Kontext der Wasserverteilung durch Panning, Zooming und Rotation betrachtet werden. Die Hinzunahme weiterer histologischer Schnittbilder an verschiedenen Positionen des Korns erlaubt detailliertere Analysen.

Die Integrationssicht **Gradient on Image** visualisiert die Kombination des Sauerstoffgradienten und eines histologischen Schnittbildes (siehe Abbildung 5.1, Seite 72, Teil 3b). In der Publikation von Rolletschek *et al.* [123] ist eine solche Kombination in zwei Dimensionen dargestellt, welche aufwändig manuell angefertigt wurde und in der nicht leicht erkennbar ist, warum die Punkte im Schnittbild verstreut sind. Durch die Darstellung in 3D wird die Visualisierung des Gradienten und des Schnittbildes entkoppelt und auch deutlicher, entlang welchen Pfades der Gradient gemessen wurde.

Die Integrationssicht **Omics Network Context** visualisiert die Kombination der zeitlich aufgelösten Metabolit-Daten und des metabolischen TCA-Zyklus-Netzwerkes (siehe Abbildung 5.1, Seite 72, Teil 3d). Verknüpft wurden in dem Fall vier ausgewählte Intermediate des TCA-Zyklus: Pyruvat, Citrat, Isocitrat und Succinat. Die Änderung der Konzentration über die Zeit kann nun im Kontext der entsprechenden Netzwerkstruktur analysiert werden. Es ist weiterhin möglich, die Metabolit-Daten durch statistische Tests von Ausreißern zu befreien (Grubbs Test) und auf Normalverteilung zu testen.

Die Integrationssicht **Substance Scatter Matrix** visualisiert die Kombination der zeitlich aufgelösten Metabolit-Daten miteinander (siehe Abbildung 5.1, Seite 72, Teil 3c, gezeigt sind nur die vier Metabolite der vorhergehenden Integrationssicht). Die Matrix, aufgespannt durch Streudiagramme paarweiser Metabolite, erlaubt es, visuell Korrelationen zwischen den oben ausgewählten Intermediaten zu erkennen. Interaktiv können Färbungen und statistische Parameter geändert und damit eine Neuberechnung der Matrix ausgelöst

werden. Durch direkte Interaktionen sind nicht-visualisierte statistische Werte abfragbar, beispielsweise sind Succinat und Isocitrat mit einem Wert von $r = 0,94$ korreliert, Pyruvat mit Isocitrat aber nur mit $r = 0,84$. Citrat ist mit den anderen drei Intermediaten statistisch nicht korreliert ($r \simeq 0$), obwohl basierend auf der Netzwerkstruktur eine hohe Korrelation erwartet wird. Dies lässt den Schluss auf geringe Qualität der Metabolitdaten zu, da eine bisher unentdeckte biochemische Reaktion im Primärmetabolismus, welche eine solche Divergenz erklären würde, nicht wahrscheinlich ist.

Die Integrationssicht **Network Stacking** stellt mittels zeitlich aufgelöster Metabolit-Daten angefärbte Kopien des metabolischen TCA-Zyklus-Netzwerkes als Netzwerkstapel dar (siehe Abbildung 4.7, Seite 58, 3. Fall). Dazu wird das metabolische Netzwerk zuerst mittels der Graph-Visualisierung vereinfacht, indem einige Metabolitknoten und die Reaktionsknoten entfernt werden. Darauffolgend werden mehrere Mappings erzeugt, die jeweils die Metabolit-Daten eines Zeitpunktes mit einer Kopie des metabolischen Netzwerkes kombinieren. Die angereicherten metabolischen Netzwerke werden in der Integrationssicht **Omics Network Context** dargestellt und interaktiv alle Netzwerkknoten entsprechend eines Farbcodes und der Metabolit-Daten zu dem Zeitpunkt eingefärbt. Diese Menge gefärbter Netzwerke wird in einem neuen Mapping zu einem Netzwerkstapel kombiniert und in der Integrationssicht **Network Stacking** visualisiert. Mit dieser ist es möglich, die zeitliche Entwicklung der Metabolite ohne Diagramme basierend auf der Knotenfärbung in 3D zu explorieren.

Die Integrationssicht **Image Omics Brushing** stellt die Kombination der räumlich aufgelösten Genexpressionsdaten (Perikarp, Endosperm, Embryo) im Kontext des Koexpressionsnetzwerkes dar und erlaubt die Exploration der Daten auf Basis eines segmentierten histologischen Bild (siehe Abbildung 6.3). Durch Mausbewegungen über das Bild kann der Anwender mittels der Interaktionstechnik Brushing die Gewebe Perikarp, Embryo und Endosperm auswählen und damit automatisch die Netzwerkknoten auf Basis der in dem Gewebe gemessenen Expressionsdaten anfärben. Existieren in dem jeweiligen Knoten keine Expressionsdaten für das selektierte Segment, wird dieser gelb angefärbt. In Abbildung 6.3 sind dadurch sehr leicht deutliche Änderungen der Expressionsrate der Gene TC218980, TC228390, TC209890 und TC196930 in beiden Geweben zu beobachten (Rotfärbung). Diese Gene besitzen laut AmiGO [18] Funktionen in der Pathogenese und sind z. B. an der Reaktion auf biotischen und abiotischen Stress beteiligt.

Die beschriebenen Integrationssichten repräsentieren sehr unterschiedliche Sichten auf den Datensatz des sich entwickelnden Gerstenkorns. Verschiedenartige Visualisierungen ermöglichen schnelles Verständnis und Interaktionstechniken unterstützen Exploration und Manipulation kombinierter Datenwerte. Viele weitere Integrationssichten und Anwendungsfälle sind möglich, z. B. könnten die einzelnen *numerischen Werte* des Gradienten mittels der Integrationssicht **Image Omics Brushing** durch den Anwender interaktiv ausgewählt werden. Diese Werte dienen als Eingabe für die Simulation eines stöchiometrischen Modells, deren Resultat als Flussverteilung in der Graph-Visualisierung dargestellt wird (vergleiche [119]). Segmentierte Volumen können als Rückgrat für die

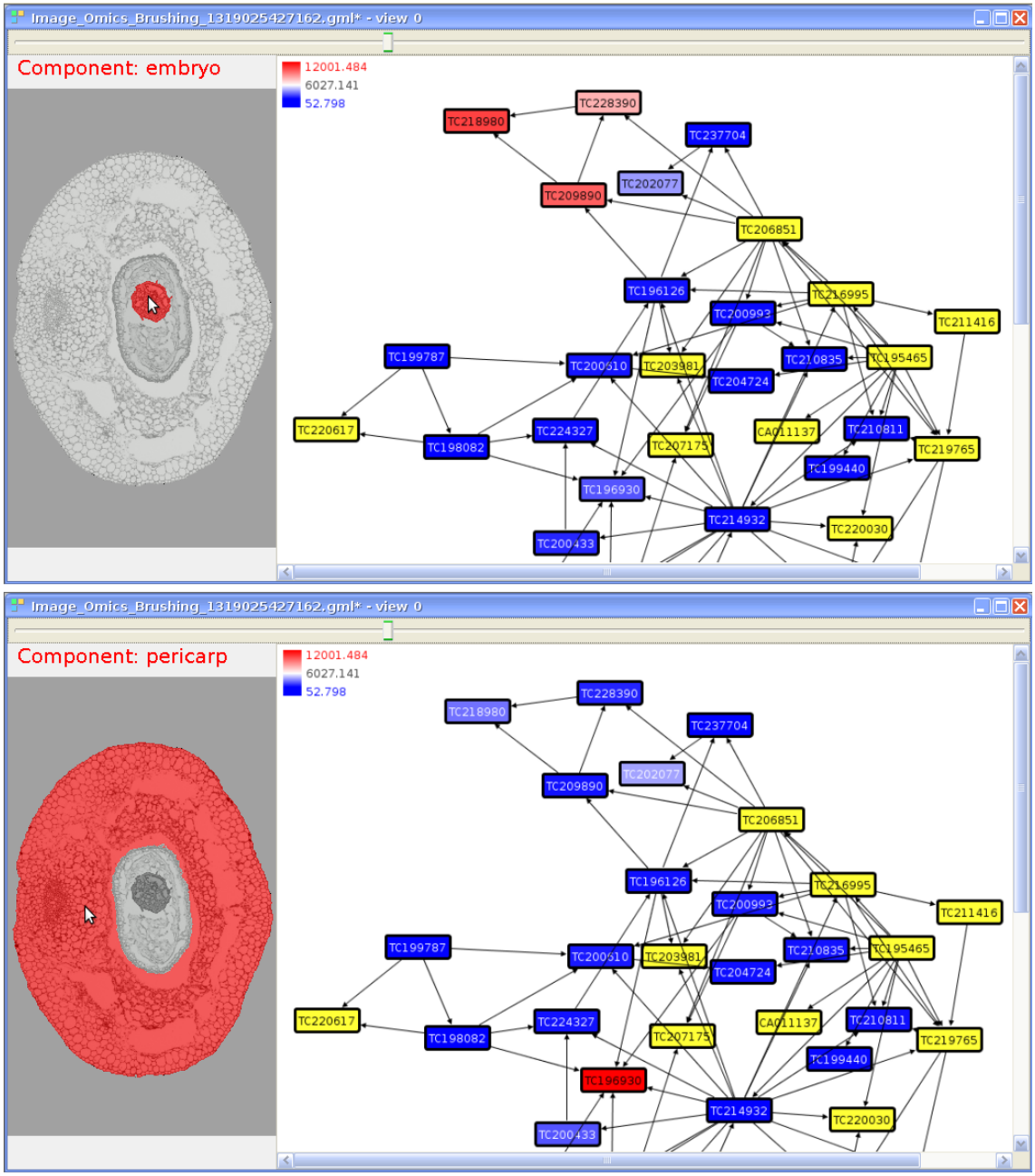


Abbildung 6.3: Exploration räumlich aufgelöster Genexpressionsdaten aus GENEVESTIGATOR [164] im Kontext eines Koexpressionsnetzwerkes von Mochida *et al.* [96] in der Integrationssicht **Image Omics Brushing**. Die Visualisierung reagiert auf die Auswahl des Gewebes im histologischen Schnittbild durch Färbung der Netzwerkknoten entsprechend der Genexpressionsdaten. Oben: Auswahl des Embryo-Gewebes. Unten: Auswahl des Perikarp-Gewebes. Erkennbar sind deutliche Unterschiede der Expressionsrate der Gene TC218980, TC228390, TC209890 und TC196930 (rote Knoten).

Navigation durch Gewebe-spezifische Netzwerke dienen. Auch Berechnungen, z. B. die Vorhersage räumlicher Daten mittels Modellierungsansätze auf Basis anderer räumlicher Daten ist denkbar. Beispielsweise könnte ein funktionelles Bild, welches die quantitative Sukrose-Verteilung darstellt, in ein Stärkebild umgerechnet werden, falls ein kinetisches Modell der Stärkesynthese veröffentlicht ist. Für alle diese Ideen sind aber entsprechende Daten nötig, die derzeit noch nicht im nötigen Umfang verfügbar sind.

6.4 Fazit

Dieses Kapitel beschreibt beispielhaft, wie multimodale Datensätze verschiedener Spezies mit der in dieser Arbeit vorgestellten Methodik integriert werden können, welche Möglichkeiten zur Kombination der Daten existieren und welche Visualisierungen sinnvoll sind, um verschiedene Sichten auf die biologischen Systeme zu generieren (vergleiche Tabelle 6.1). Das bekannte Modell der Blütenentwicklung von *Arabidopsis thaliana* konnte mit Genexpressionsdaten erfolgreich im Kontext eines genregulatorischen Netzwerkes validiert werden. Die intuitive Exploration großer Bildermengen mittels Netzwerken wurde anhand zweier *Drosophila melanogaster*-Bilddatenbanken beschrieben. Schließlich zeigte ein *Hordeum vulgare*-Datensatz, bestehend aus unterschiedlichsten Datentypen, die Möglichkeiten zur Erstellung einer großen Zahl verschiedener Integrationssichten auf.

Integrationssicht	Daten	Spezies	Abschnitt
Omics Distribution by Network	segmentiertes Bild, genregulatorisches Netzwerk, räumlich aufgelöste Genexpressionsdaten	*Arabidopsis thaliana*	6.1
Image Browsing by Network	funktionelle Bilder, PPI-Netzwerk, zeitlich aufgelöste Genexpressionsdaten	*Drosophila melanogaster*	6.2
Multimodal Alignment	Schnittbilder, Volumen	*Hordeum vulgare*	6.3
Gradient on Image	Sauerstoffgradient, Schnittbild	*Hordeum vulgare*	6.3
Omics Network Context	zeitlich aufgelöste Metabolitdaten, metabolisches Netzwerk	*Hordeum vulgare*	6.3
Substance Scatter Matrix	zeitlich aufgelöste Metabolitdaten	*Hordeum vulgare*	6.3
Network Stacking	zeitlich aufgelöste Metabolitdaten, metabolisches Netzwerk	*Hordeum vulgare*	6.3
Image Omics Brushing	räumlich aufgelöste Genexpressionsdaten, Koexpressionsnetzwerk, segmentiertes Schnittbild	*Hordeum vulgare*	6.3

Tabelle 6.1: Aufschlüsselung aller in diesem Kapitel beschriebenen Anwendungsfälle nach Integrationssicht, Daten, Spezies und Abschnitt.

7

Diskussion

7.1 Zusammenfassung

In dieser Arbeit wurde eine Methodik beschrieben, welche multimodale biologische Experimentdaten aus verschiedenen Quellen mit unterschiedlichen Auflösungsebenen integrieren, flexibel kombinieren und vielfältig visualisieren kann. Erstmals erlaubt es eine Anwendung, vier wichtige Typen von Experimentdaten (*numerische Werte*, *Bilder*, *Volumen* und *Netzwerke*) zusammen zu bearbeiten. Dadurch ist es möglich, sehr unterschiedliche Informationen eines biologischen Systems in einem Kontext betrachten zu können. Verschiedene Visualisierungs- und Interaktionstechniken ermöglichen es, neue Erkenntnisse über biologische Systeme zu gewinnen, beispielsweise durch Datennavigation und visuelle Analyse. Visualisierungen sind darüber hinaus für Publikationen (statische Bilder) oder den Begutachtungs-Prozess (interaktive Exploration) exportierbar. Somit konnten die Anforderungen an das System in Form der Anwendung Hive erfolgreich umgesetzt werden.

Dazu wurden Grundlagen der Biologie und Informatik beschrieben. Es folgte eine Analyse der Anforderungen an das System und eine Übersicht über vergleichbare Anwendungen und Ansätze. Basierend auf einem Datenmodell können multimodale Daten mittels einer Visualisierungspipeline in drei Schritten zu visualisierten Daten transformiert werden: *Datenintegration* erfolgt mittels Graphen, welche Datennavigation und Filteroperationen unterstützen. *Datenkombination* ermöglicht das flexible Kombinieren von Daten, um diese in wechselnde Kontexte bringen zu können. *Datenvisualisierung* schließlich erlaubt diese Datenkombinationen intuitiv darzustellen und visuell zu analysieren. Die Implementierung der Methodik in Form der Anwendung Hive wurde beschrieben und auf Designentscheidungen eingegangen. Abschließend folgte die Anwendung der Methodik auf drei verschiedene biologische Datensätze und Fragestellungen.

7.2 Diskussion der vorgestellten Methodik

7.2.1 Datenintegration

Basis der Datenintegration ist ein einfaches und leicht verständliches Datenmodell, welches nur wenige vom Anwender zu spezifizierende Metadaten benötigt. Komplexe experimentelle Abläufe und Prozeduren können allerdings nicht erfasst werden. Derzeit werden im Modell vier wichtige Datentypen beschrieben, die einen großen Teil der in der biologischen Forschung auftretenden Daten abdecken. Andere Daten wie molekulare Strukturen, Oberflächenmodelle und Sequenzinformationen werden derzeit nicht berücksichtigt. Die Hinzunahme zusätzlicher Datentypen könnte den Ansatz der Methodik weiter generalisieren und die Anwendbarkeit für verschiedene Fragestellungen erhöhen.

Die Datenintegration erfolgt durch einen flexiblen, Graph-basierten Ansatz, der die Zusammenhänge der Daten auf der Metadaten- und Datenwertebene grafisch aufbereiten kann und intuitive Interaktionen erlaubt. Ein vergleichbarer Ansatz wird auch in einigen anderen Anwendungen erfolgreich genutzt, insbesondere wenn es gilt, komplexe Interaktionen und Arbeitsabläufe intuitiv(er) abzubilden. Beispielsweise basiert die Datenmanipulation in den Visualisierungsanwendungen Amira [140] und SCIRun [106] auf einer ähnlichen Graphstruktur, in der die Knoten funktionelle Module und Kanten den Datenfluss repräsentieren. Übersichtliches und iteratives Kombinieren und Nachvollziehen des Arbeitsfortschrittes sind somit in allen Anwendungen möglich. Es wird jedoch eine signifikante Einarbeitungszeit benötigt, in der sich Anwender mit dem Vorgehen vertraut machen müssen, bevor ein produktives Arbeiten möglich wird.

Die Speicherung der Daten und des Integrationsgraphen erfolgt lokal im Dateisystem, indem der Integrationsgraph die Konsistenz der Daten wahrt. Diese lokale Datenhaltung ermöglicht eine hohe Datensicherheit und gute Performance, gewährleistet aber beispielsweise keine Transaktionssicherheit und keinen externen Zugriff.

7.2.2 Datenkombination und -visualisierung

Der Graph-basierte Ansatz der Datenintegration ermöglicht frei wählbare und flexible Datenkombinationen, welche vielfältig visualisiert werden können. Jede einzelne Kombination und deren Visualisierung erfordert einen hohen Implementierungsaufwand, weswegen aus Zeitgründen einige Datenkombinationen bisher nicht in einem solchen Umfang implementiert werden konnten, dass die Funktionalität mit der spezialisierter Programme mithalten kann. Hinzu kommt, dass, wie in Tabelle 4.2, Seite 63 zu sehen ist, bisher manche Kombinationsmöglichkeiten nicht abgedeckt sind, insbesondere die Kombination numerischer Werte oder Netzwerke mit volumetrischen Datensätzen. Weiterhin sind manche Interaktionen, wie zum Beispiel Selektion einzelner Volumenvoxel oder -segmente, in der Praxis vergleichsweise aufwändig zu implementieren. Im Gegensatz zu zweidimensionalen Visualisierungen sind oft spezielle Eingabegeräte und Bedienungshinweise notwendig, wodurch die Interaktion mit volumetrischen Daten in der Praxis oft kompliziert ist.

7.2.3 Datenverfügbarkeit

Die fehlende Verfügbarkeit passender Datensätzen ist derzeit ein grundlegendes Problem der in dieser Arbeit beschriebenen Methodik. Fehlen passende Daten, können theoretische Verknüpfungspunkte zwischen verschiedenen Datentypen nicht überzeugend in die Praxis umgesetzt werden. Die Gründe dafür sind in zwei Klassen kategorisierbar:

Ein Grund ist die fehlende Existenz räumlich aufgelöster, umfangreicher Datensätze, insbesondere volumetrischer Daten. Es gibt einerseits viele umfangreiche Datenbanken zur Speicherung genetischer und molekularbiologischer Daten (z. B. Genexpressionsdaten), weil eine Veröffentlichung aller Daten in diesen Teilgebieten Grundvoraussetzung für ein Publikationsvorhaben ist. Andererseits besteht für räumliche Daten kein vergleichbarer Zwang, wodurch nicht sehr viele umfangreiche Bild-Datenbanken vorhanden sind (vergleiche auch die Ausführungen in Abschnitt 3.2, Seite 33). Bilddaten sind außerdem von Natur aus deutlich diverser als andere Typen: Sequenzdaten beispielsweise sind innerhalb eines Organismus weitgehend identisch, Bilddaten hingegen können verschiedene Strukturen (z. B. Organe, Gewebe) des Organismus mit unterschiedlichen Verfahren (TEM, bildgebende NMR) darstellen. Hinzu kommen Zugriffsbeschränkungen (beispielsweise bei der ADNI-Datenbank [3]), da Experimente zur Erzeugung von Bild- und insbesondere Volumendaten sehr teuer und zeitaufwändig sein können.

Der zweite Grund für das Fehlen passender Datensätze ist, dass Daten oftmals unterschiedliche Auflösungsebenen aufweisen. Einzelne Arbeitsgruppen haben üblicherweise nicht das Ziel, verschiedenste Datentypen systematisch zu messen, sondern konzentrieren sich auf wenige bestimmte Typen und die Auflösungsebenen, die zur Beantwortung einer biologischen Fragestellung ausreichend sind. Beispielsweise können sehr günstig hochauflösende Fotografien von Gewebestrukturen aufgenommen werden, welche verschiedene Gewebe zeigen. Genexpressionsdaten sind im Gegensatz dazu zwar zunehmend, aber bisher nur selten Gewebe-spezifisch verfügbar, weil dies aufwändige bioanalytische Verfahren wie beispielsweise Mikrodissektion [100] erfordert (vergleiche auch die typischen Auflösungsebenen verschiedener Datentypen in Abbildung 2.1, Seite 4). Somit müssen zur Erzeugung umfangreicher Datensätze die Daten verschiedener Gruppen kombiniert werden, welche durch unterschiedliche Auflösungsebenen oft schwer oder gar nicht miteinander verknüpfbar sind.

Aufgrund der geringen Datenverfügbarkeit wurden bisher von potentiellen Anwendern kaum Anforderungen formuliert, komplexe und hochaufgelöste Datensätze miteinander zu verknüpfen. Der Fokus biologischer Experimente verschiebt sich aber im Zuge der Systembiologie zunehmend hin zur Messung solcher Daten. Eine Verknüpfung der Daten ist derzeit nur mit der in dieser Arbeit vorgestellten Methodik möglich. Das Potential dieses Ansatzes konnte in Kapitel 6 anhand der wenigen derzeit verfügbaren Datensätze beschrieben werden, wobei sich diese Anwendungsfälle auf die Validierung bekannter Modelle durch Visualisierung und Interaktion beschränken. Mit der zu erwartenden Verfügbarkeit komplexer Datensätze können dann auch neuartige biologische Erkenntnisse abgeleitet werden.

7.3 Ausblick

Ein Wandel des Publikationsverhaltens und die Kostenreduktion biologischer Experimente durch zunehmende Automatisierung (z. B. Hochdurchsatz-Techniken für Bilddaten) wird die Verfügbarkeit hochaufgelöster, komplexer und umfassender Datensätze verbessern. Insbesondere die in naher Zukunft verfügbaren räumlich und zeitlich aufgelösten Genexpressionsdaten wecken bereits heute das Interesse an Ansätzen wie der Integrationssicht **Omics Distribution by Network**. Die in dieser Arbeit beschriebene Methodik kann auf Basis dieser neuen Daten und Anforderungen validiert und weiter ausgebaut werden, z. B. durch Entwicklung neuer und vielversprechender Integrationssichten (einige Beispiele wurden in Abschnitt 6.3, Seite 85 beschrieben). Dazu ist jedoch viel Arbeitskraft und Wissen in verschiedensten Bereichen der (Bio-)Informatik nötig, wie beispielsweise Bildverarbeitung, Visualisierung, Interaktionstechniken, Statistikfunktionen, Modellierungsansätze und mehr.

Die Anwendung Hive sollte insbesondere hinsichtlich der Volumendaten sowohl bezüglich Speicherbedarf, als auch Rendergeschwindigkeit optimiert werden, um größere Volumendaten auf normalen Rechnern visualisieren zu können. Zusätzliche Interaktionstechniken sollten helfen, mit den Volumen besser interagieren zu können und somit diese in mehr Kombinationen zu berücksichtigen. Die Serialisierung numerischer Werte im Integrationsgraphen kann optimiert und die Unterstützung von Ontologien eingebaut werden. Letzteres ist über den Gene Ontology Lookup Service [33] möglich und vereinfacht die Eingabe von Metadaten und Kontrolle des Vokabulars erheblich. Des weiteren bietet sich die Einbindung verschiedener Anwendungen wie Fiji an, um komplexe Datenkombinationen ohne großen Aufwand realisieren zu können. Da Hive als Quellcode-freie Anwendung veröffentlicht wurde, können solche Erweiterungen auch über den Rahmen dieser Arbeit hinaus realisiert werden.

Literaturverzeichnis

[1] Abramoff, Michael D. ; Magelhaes, Paulo J. ; Ram, Sunanda J.: Image Processing with ImageJ. In: *Biophotonics International* 11 (2004), S. 36–42

[2] Alla, Hassane ; David, Rene: Continuous and hybrid Petri Nets. In: *Journal of Circuits, Systems, and Computers* 8 (1998), Nr. 1, S. 159–188

[3] Alzheimer's Disease Neuroimaging Initiative: *Data and Samples*. – `http://adni.loni.ucla.edu/`

[4] Ameur, Adam ; Yankovski, Vladimir ; Enroth, Stefan ; Spjuth, Ola ; Komorowski, Jan: The LCB Data Warehouse. In: *Bioinformatics* 22 (2006), Nr. 8, S. 1024–1026

[5] Augen, Jeffrey: Information technology to the rescue! In: *Nature Biotechnology* 19 (2001), S. BE39–BE40

[6] Baeza-Yates, Ricardo ; Ribeiro-Neto, Berthier: *Modern Information Retrieval*. 1. Edition. Addison Wesley, 1999

[7] Ball, Catherine A. ; Sherlock, Gavin ; Brazma, Alvis: Funding high-throughput data sharing. In: *Nature Biotechnology* 22 (2004), Nr. 9, S. 1179–1183

[8] Balzert, Helmut: *Lehrbuch der Software-Technik*. 2. Edition. Spektrum Akademischer Verlag, Heidelberg, 2000

[9] Baxevanis, Andreas D. ; Ouellette, Francis B. F.: *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. 2. Edition. John Wiley & Sons, 2001

[10] Bederson, Benjamin B. ; Hollan, James D. ; Perlin, Ken ; Meyer, Jonathan ; Bacon, David ; Furnas, George: Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics. In: *Journal of Visual Languages and Computing* 7 (1996), Nr. 1, S. 3–31

[11] Boline, Jyl ; MacKenzie-Graham, Allan ; Shattuck, David ; Yuan, H. ; Anderson, S. ; Sforza, D. M. ; Williams, Robert ; Wong, Willy W. ; Martone, Maryann E. ; Zaslavsky, Ilya ; Toga, Arthur: A digital atlas and neuroinformatics framework for query and display of disparate data. In: *Society for Neuroscience Conference*, 2006

[12] Bradford, Yvonne ; Conlin, Tom ; Dunn, Nathan ; Fashena, David ; Frazer, Ken ; Howe, Douglas G. ; Knight, Jonathan ; Mani, Prita ; Martin, Ryan ; Moxon, Sierra A T. ; Paddock, Holly ; Pich, Christian ; Ramachandran, Sridhar ; Ruef, Barbara J. ; Ruzicka, Leyla ; Schaper, Holle B. ; Schaper, Kevin ; Shao, Xiang ; Singer, Amy ; Sprague, Judy ; Sprunger, Brock ; Slyke, Ceri V. ; Westerfield, Monte: ZFIN: enhancements and updates to the Zebrafish Model Organism Database. In: *Nucleic Acids Research* 39 (2011), S. D822–D829

[13] Brandes, Ulrich ; Dwyer, Tim ; Schreiber, Falk: Visual Triangulation of Network-based Phylogenetic Trees. In: *Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization*, 2004, S. 75–84

[14] Brazma, Alvis ; Hingamp, Pascal ; Quackenbush, John ; Sherlock, Gavin ; Spellman, Paul ; Stoeckert, Chris ; Aach, John ; Ansorge, Wilhelm ; Ball, Catherine A. ; Causton, Helen C. ; Gaasterland0, Terry ; Glenisson, Patrick ; Holstege, Frank C. ; Kim, Irene F. ; Markowitz, Victor ; Matese, John C. ; Parkinson, Helen ; Robinson, Alan ; Sarkans, Ugis ; Schulze-Kremer, Steffen ; Stewart, Jason ; Taylor, Ronald ; Vilo, Jaak ; Vingron, Martin: Minimum information about a microarray experiment (MIAME) - toward standards for microarray data. In: *Nature Genetics* 29 (2001), Nr. 4, S. 365–371

[15] Brinkley, James F. ; Rosse, Cornelius: Imaging Informatics and the Human Brain Project: the Role of Structure. In: *Yearbook of Medical Informatics* 4 (2002), S. 111–128

[16] Brodlie, Ken ; Wood, Jason: Recent Advances in Volume Visualization. In: *Computer Graphics Forum* 20 (2001), S. 125–148

[17] Bry, François ; Kröger, Peer: A Computational Biology Database Digest: Data, Data Analysis, and Data Management. In: *Distributed and Parallel Databases* 13 (2003), Nr. 1, S. 7–42

[18] Carbon, Seth ; Ireland, Amelia ; Mungall, Christopher J. ; Shu, ShengQiang ; Marshall, Brad ; Lewis, Suzanna ; Hub, AmiG. O. ; Web Presence Working Group: AmiGO: online access to ontology and annotation data. In: *Bioinformatics* 25 (2009), Nr. 2, S. 288–289

[19] Card, Stuart K. ; Mackinlay, Jock D. ; Shneiderman, Ben: *Readings in information visualization: using vision to think*. 1. Edition. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999

[20] Cavalieri, Duccio ; Filippo, Carlotta D.: Bioinformatic methods for integrating whole-genome expression results into cellular networks. In: *Drug Discovery Today* 10 (2005), Nr. 10, S. 727–734

[21] Chaouiya, Claudine: Petri net modelling of biological networks. In: *Briefings in Bioinformatics* 8 (2007), S. 210–219

[22] Chen, Chaomei: *Mapping Scientific Frontiers: The Quest for Knowledge Visualization*. 1. Edition. Springer, Berlin, 2003

[23] Chen, Peter Pin-Shan: The Entity-Relationship Model: Toward a Unified View of Data. In: *ACM Transactions on Database Systems* 1 (1976), S. 9–36

[24] Cheng, Heng-Da ; Jiang, Xihua ; Sun, Ying ; Wang, Jingli: Color image segmentation: advances and prospects. In: *Pattern Recognition* 34 (2001), Nr. 12, S. 2259–2281

[25] Cheng, Ruida ; Bokinsky, Alexandra ; Hemler, Paul ; McCreedy, Evan ; McAuliffe, Matthew: Java based volume rendering frameworks. In: *Medical Imaging 2008: Visualization, Image-guided Procedures, and Modeling* 6918 (2008), Nr. 1, S. 691804.1–15

[26] Chernoff, Herman: The use of faces to represent points in k-dimensional space graphically. In: *Journal of the American Statistical Association* 68 (1973), Nr. 342, S. 757–765

[27] Churcher, Neville ; Irwin, Warwick: Informing the Design of Pipeline-Based Software Visualisations. In: *Proceedings of Asia-Pacific Symposium on Information Visualisation*, 2005, S. 59–68

[28] Churchill, Gary A.: Fundamentals of experimental design for cDNA microarrays. In: *Nature Genetics* 32 (2002), S. 490–495

[29] Codd, Edgar F.: Further Normalization of the Data Base Relational Model. In: *IBM Research Report, San Jose, California* RJ909 (1971)

[30] Conn, Michael P.: Imaging in Biological Research (Methods in Enzymology) Part A. In: *Imaging in Biological Research* Bd. 385. 1. Edition. Academic Press, 2004

[31] Conrad, Stefan: *Föderierte Datenbanksysteme*. 9. Edition. Springer-Verlag, Berlin/Heidelberg, 1997

[32] Cormen, Thomas H. ; Leiserson, Charles E. ; Rivest, Ronald L. ; Stein, Clifford: *Introduction to Algorithms*. 2. Edition. MIT Press, 2001

[33] Cote, Richard G. ; Jones, Philip ; Apweiler, Rolf ; Hermjakob, Henning: The Ontology Lookup Service, a lightweight cross-platform tool for controlled vocabulary queries. In: *BMC Bioinformatics* 7 (2006), Nr. 1, S. 97.1–7

[34] Crick, Francis H.: On protein synthesis. In: *Symposia of the Society for Experimental Biology* 12 (1958), S. 138–63

[35] De Bodt, Stefanie ; Carvajal, Diana ; Hollunder, Jens ; Cruyce, Joost Van d. ; Movahedi, Sara ; Inzé, Dirk: CORNET: A User-Friendly Tool for Data Mining and Integration. In: *Plant Physiology* 152 (2010), S. 1167–1179

[36] Dettmer, Katja ; Aronov, Pavel A. ; Hammock, Bruce D.: Mass spectrometry-based metabolomics. In: *Mass Spectrometry Reviews* 26 (2007), Nr. 1, S. 51–78

[37] Detwiler, Landon T. ; Suciu, Dan ; Franklin, Joshua D. ; Moore, Eider B. ; Poliakov, Andrew V. ; Lee, Eunjung S. ; Corina, David P. ; Ojemann, George A. ; Brinkley, James F.: Distributed XQuery-based integration and visualization of multimodality brain mapping data. In: *Frontiers in Neuroinformatics* 3 (2009), Nr. 2, S. 1–13

[38] Dinov, Ivo D. ; Valentino, Daniel ; Shin, Bae C. ; Konstantinidis, Fotios ; Hu, Guogang ; MacKenzie-Graham, Allan ; Lee, Erh-Fang ; Shattuck, David ; Ma, Jeff ; Schwartz, Craig ; Toga, Arthur W.: LONI Visualization Environment. In: *Journal of Digital Imaging* 19 (2006), Nr. 2, S. 148–158

[39] Elmasri, Ramez ; Navathe, Shamkant B.: *Fundamentals of Database Systems*. 2. Edition. Addison Wesley, 2006

[40] Feng, Guangjie ; Burton, Nick ; Hill, Bill ; Davidson, Duncan ; Kerwin, Janet ; Scott, Mark ; Lindsay, Susan ; Baldock, Richard: JAtlasView: a Java atlas-viewer for browsing biomedical 3D images and atlases. In: *BMC Bioinformatics* 6 (2005), Nr. 1, S. 47.1–7

[41] Flanagan, Catherine A. ; Ma, Hong: Spatially and temporally regulated expression of the MADS-box gene AGL2 in wild-type and mutant arabidopsis flowers. In: *Plant Molecular Biology* 26 (1994), Nr. 2, S. 581–595

[42] Fly-FISH: *A database of Drosophila embryo mRNA localization patterns.* – `http://fly-fish.ccbr.utoronto.ca/cgi-bin/eric_list.pl`

[43] Fodor, Imola K.: A survey of dimension reduction techniques / Center for Applied Scientific Computing, Lawrence Livermore National Laboratory. 2002. – Forschungsbericht

[44] FruitFly: *Patterns of gene expression in Drosophila embryogenesis.* – `http://www.fruitfly.org/cgi-bin/ex/bquery.pl?qpage=entry&qtype=summary`

[45] Gansner, Emden R. ; Koren, Yehuda: Improved circular layouts. In: *Lecture Notes in Computer Science* 4372 (2007), S. 386–398

[46] Gehlenborg, Nils ; O'Donoghue, Sein I. ; Baliga, Nitin S. ; Goesmann, Alexander ; Hibbs, Matthew A. ; Kitano, Hiroaki ; Kohlbacher, Oliver ; Neuweger, Heiko ; Schneider, Reinhard ; Tenenbaum, Dan ; Gavin, Anne-Claude: Visualization of omics data for systems biology. In: *Nature Methods* 7 (2010), S. S56–S68

[47] Gehringer, Doug: *VolRend.* – `http://www.java2s.com/Code/Java/3D/VolRend.htm`

[48] Glidewell, Sheila M.: NMR imaging of developing barley grains. In: *Journal of Cereal Science* 43 (2006), S. 70–78

[49] Grafahrend-Belau, Eva ; Klukas, Christian ; Junker, Björn H. ; Schreiber, Falk: FBASimViz: interactive visualization of constraint-based metabolic models. In: *Bioinformatics* 25 (2009), Nr. 20, S. 2755–2757

[50] Grafahrend-Belau, Eva ; Weise, Stefan ; Koschützki, Dirk ; Scholz, U. ; Junker, Björn H. ; Schreiber, F.: MetaCrop - A detailed database of crop plant metabolism. In: *Nucleic Acids Research* 36 (2008), S. D954–D958

[51] Hartmann, Anja: *Automatisierte Datenaufnahme und Bildverarbeitung zur Hochdurchsatz-Phänotypisierung von Gerste*, Martin-Luther-Universität Halle-Wittenberg, Naturwissenschaftliche Fakultät III, Diplomarbeit, 2009

[52] Hibbard, William ; Anderson, John ; Paul, Brian: A Java And World Wide Web Implementation Of VisAD. In: *Advances in Space Research* 22 (1998), Nr. 11, S. 1583–1589

[53] Hill, Thomas ; Lewicki, Pawel: *Statistics: Methods and Applications*. 1. Edition. Statsoft, 2005

[54] Hjornevik, Trine ; Leergaard, Trygve B. ; Darine, Dmitri ; Moldestad, Olve ; Dale, Anders M. ; Willoch, Frode ; Bjaalie, Jan G.: Three-Dimensional Atlas System for Mouse and Rat Brain Imaging Data. In: *Frontiers in Neuroinformatics* 1 (2007), S. 1–12

[55] Hofestädt, Ralf: A petri net application to model metabolic processes. In: *Systems Analysis Modelling Simulation* 16 (1994), Nr. 2, S. 113–122

[56] Holten, Danny ; Wijk, Jarke J. V.: Force-Directed Edge Bundling for Graph Visualization. In: *Computer Graphics Forum* 28 (2009), Nr. 3, S. 983–990

[57] Horn, John D. ; Grethe, Jeffrey S. ; Kostelec, Peter ; Woodward, Jeffrey B. ; Aslam, Javed A. ; Rus, Daniela ; Rockmore, Daniel ; Gazzaniga, Michael S.: The Functional Magnetic Resonance Imaging Data Center (fMRIDC): the challenges and rewards of large-scale databasing of neuroimaging studies. In: *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences* 356 (2001), Nr. 1412, S. 1323–1339

[58] Horn, John D. ; Ball, Catherine A.: Domain-specific data sharing in neuroscience: What do we have to learn from each other? In: *Neuroinformatics* 6 (2008), Nr. 2, S. 117–121

[59] Hu, Zhenjun ; Hung, Jui-Hung ; Wang, Yan ; Chang, Yi-Chien ; Huang, Chia-Ling ; Huyck, Matt ; DeLisi, Charles: VisANT 3.5: Multi-scale network visualization, analysis and inference based on the gene ontology. In: *Nucleic Acids Research* 37 (2009), Nr. Web Server issue, S. W115–W121

[60] Huang, Su ; Baimouratov, Rafail ; Xiao, Pengdong ; Ananthasubramaniam, Anand ; Nowinski, Wieslaw L.: A Medical Imaging and Visualization Toolkit in Java. In: *Journal of Digital Imaging* 19 (2006), Nr. 1, S. 17–29

[61] Hunter, Peter J. ; Crampin, Edmund J. ; Nielsen, Poul M. F.: Bioinformatics, multiscale modeling and the IUPS Physiome Project. In: *Briefings in Bioinformatics* 9 (2008), Nr. 4, S. 333–343

[62] Iersel, Martijn P. ; Kelder, Thomas ; Pico, Alexander R. ; Hanspers, Kristina ; Coort, Susan ; Conklin, Bruce R. ; Evelo, Chris: Presenting and exploring biological pathways with PathVisio. In: *BMC Bioinformatics* 9 (2008), S. 399.1–9

[63] Ihlow, Alexander: *Ein Hochdurchsatz-Screeningsystem zur Objekterkennung in Mikroskop-Farbbildern im Rahmen der Analyse pflanzlicher Pathogenresistenz*, Otto-von-Guericke-Universität Magdeburg , Fakultät für Elektrotechnik und Informationstechnik, Doktorarbeit, 2006

[64] Inselberg, Alfred ; Dimsdale, Bernard: Multidimensional Lines. In: *SIAM Journal on Applied Mathematics* 54 (2009), Nr. 2, S. 559–577

[65] Jahnke, Siegfried ; Menzel, Marion I. ; Dusschoten, Dagmar van ; Roeb, Gerhard. W. ; Bühler, Jonas ; Minwuyelet, Senay ; Blümler, Peter ; Temperton, Vicky M. ; Hombach, Thomas ; Streun, Matthias ; Beer, Simone ; Khodaverdi, Maryam ; Ziemons, Karl ; Coenen, Heinz H. ; Schurr, Ulrich: Combined MRI–PET dissects dynamic changes in plant structures and functions. In: *The Plant Journal* 59 (2009), S. 634–644

[66] Johnson, Chris: Top Scientific Visualization Research Problems. In: *IEEE Computer Graphics and Applications* 24 (2004), Nr. 4, S. 13–17

[67] Junker, Björn H. ; Klukas, Christian ; Schreiber, Falk: VANTED: A system for advanced data analysis and visualization in the context of biological networks. In: *BMC Bioinformatics* 7 (2006), S. 109.1–13

[68] Junker, Björn H. ; Schreiber, Falk: *Analysis of Biological Networks*. 1. Edition. John Wiley & Sons, 2008

[69] Jusufi, Illir ; Klukas, Christian ; Kerren, Andreas ; Schreiber, Falk: Guiding the Interactive Exploration of Metabolic Pathway Interconnections. In: *Information Visualization* (2011)

[70] Kanehisa, Minoru ; Goto, Susumu ; Furumichi, Miho ; Tanabe, Mao ; Hirakawa, Mika: KEGG for representation and analysis of molecular networks involving diseases and drugs. In: *Nucleic Acids Research* 38 (2010), Nr. Database issue, S. D355–D360

[71] Keim, Daniel A.: Information visualization and visual data mining. In: *IEEE Transactions on Visualization and Computer Graphics* 8 (2002), Nr. 1, S. 1–8

[72] Kindlmann, Gordon ; Durkin, James W.: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In: *Proceedings of the 1998 IEEE symposium on Volume visualization*, 1998, S. 79–86

[73] Kitano, Hiroaki: Systems Biology: A Brief Overview. In: *Science* 295 (2002), S. 1662–1664

[74] Klipp, Edda ; Herwig, Ralf ; Kowald, Axel ; Wierling, Christoph ; Lehrach, Hans: *Systems Biology in Practice: Concepts, Implementation and Application*. 1. Edition. Wiley-VCH Verlag, 2005

[75] Klukas, Christian: *Analyse und Visualisierung von Experimentdaten im Kontext biologischer Netzwerke*, Martin-Luther-Universität Halle-Wittenberg, Naturwissenschaftliche Fakultät III, Doktorarbeit, 2009

[76] Klukas, Christian ; Schreiber, Falk: Dynamic exploration and editing of KEGG pathway diagrams. In: *Bioinformatics* 23 (2007), Nr. 3, S. 344–350

[77] Köhler, Jacob ; Baumbach, Jan ; Taubert, Jan ; Specht, Michael ; Skusa, Andre ; Rüegg, Alexander ; Rawlings, Chris ; Verrier, Paul ; Philippi, Stephan: Graph-based analysis and visualization of experimental results with ONDEX. In: *Bioinformatics* 22 (2006), Nr. 11, S. 1383–1390

[78] Koschützki, D. ; Lehmann, Katharina A. ; Peeters, Leon ; Richter, Stefan ; Tenfelde-Podehl, Dagmar ; Zlotowski, Oliver: Centrality Indices. In: *Network Analysis: Methodological Foundations* Bd. 3418. Springer, 2005, S. 16–61

[79] Koslow, Stephen H.: Should the neuroscience community make a paradigm shift to sharing primary data? In: *Nature Neuroscience* 3 (2000), S. 863–865

[80] Koslow, Stephen H.: Sharing primary data: a threat or asset to discovery? In: *Nature Reviews Neuroscience* 3 (2002), S. 311–313

[81] Kutulakos, Kiriakos N. ; Seitz, Steven M.: A Theory of Shape by Space Carving. In: *International Journal of Computer Vision* 38 (2000), Nr. 3, S. 199–218

[82] Kyoto University Bioinformatics Center: *Growth of the Sequence and 3D Structure Databases*. – `http://www.genome.ad.jp/dbget/db_growth.html`

[83] Lamb, Rebecca S. ; Hill, Theresa A. ; Tan, Queenie K-G ; Irish, Vivian F.: Regulation of APETALA3 floral homeotic gene expression by meristem identity genes. In: *Development* 129 (2002), Nr. 9, S. 2079–2086

[84] Lange, Matthias: *Methoden zum homogenen Zugriff und zur Integration heterogener, biologischer Datenquellen mittels beschränkter Zugriffsmuster*, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, Doktorarbeit, 2006

[85] Law, Charles C. ; Henderson, Amy ; Ahrens, James: An Application Architecture for Large Data Visualization: A Case Study. In: *IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, 2001, S. 125–128

[86] Lehmann, Dirk J. ; Albuquerque, Georgia ; Eisemann, Martin ; Tatu, Andrada ; Keim, Daniel ; Schumann, Heidrun ; Magnor, Marcus ; Theisel, Holger: Visualisierung und Analyse multidimensionaler Datensätze. In: *Informatik-Spektrum* 33 (2010), Nr. 6, S. 589–600

[87] Lenzerini, Maurizio: Data integration: a theoretical perspective. In: *Proceedings of the Symposium on Principles of database systems*, 2002, S. 233–246

[88] Lorensen, William E. ; Cline, Harvey E.: Marching cubes: A high resolution 3D surface construction algorithm. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, S. 163–169

[89] Lécuyer, Eric ; Yoshida, Hideki ; Parthasarathy, Neela ; Alm, Christina ; Babak, Tomas ; Cerovina, Tanja ; Hughes, Timothy R. ; Tomancak, Pavel ; Krause, Henry M.: Global analysis of 3 localization reveals a prominent role in organizing cellular architecture and function. In: *Cell* 131 (2007), Nr. 1, S. 174–187

[90] Marinilli, Mauro: *Java Deployment with JNLP and Webstart*. 1. Edition. Sams Publishing, Indianapolis, IN, 2001

[91] Martin, Allen R. ; Ward, Matthew O.: High dimensional brushing for interactive exploration of multivariate data. In: *Proceedings on Visualization*, 1995, S. 271–278

[92] Martone, Maryann E. ; Tran, Joshua ; Wong, Willy W. ; Sargis, Joy ; Fong, Lisa ; Larson, Stephen ; Lamont, Stephan P. ; Gupta, Amarnath ; Ellisman, Mark H.: The Cell Centered Database project: An update on building community resources for managing and sharing 3D imaging data. In: *Journal of Structural Biology* 161 (2008), Nr. 3, S. 220–231

[93] Maura McGrail, Kristin Noack Saumya Pandey Yong Huang Xun G. Lindsey Batz B. Lindsey Batz ; Essner, Jeffrey J.: Expression of the Zebrafish CD133/prominin1 Genes in Cellular Proliferation Zones in the Embryonic Central Nervous System and Sensory Organs. In: *Developmental Dynamics* 239 (2010), S. 1849–1857

[94] McGonigle, John: *Java and 3D Interactive Image Display*, University of Aberdeen, Masterarbeit, 2006

[95] Mehlhorn, Hendrik ; Schreiber, Falk: DBE2 - Management of experimental data for the VANTED system. In: *Journal of Integrative Bioinformatics* 8 (2011), Nr. 2, S. 162.1–10

[96] Mochida, Keiichi ; Uehara-Yamaguchi, Yukiko ; Yoshida, Takuhiro ; Sakurai, Tetsuya ; Shinozaki, Kazuo: Global landscape of a co-expressed gene network in barley and its application to gene discovery in Triticeae crops. In: *Plant Cell Physiology* 52 (2011), Nr. 5, S. 785–803

[97] Moodley, Keagan ; Murrell, Hugh: A colour-map plugin for the open source, Java based, image processing package, ImageJ. In: *Computers & Geosciences* 30 (2004), Nr. 6, S. 609–618

[98] Moore, Eider B. ; Poliakov, Andrew V. ; Lincoln, Peter ; Brinkley, James F.: Mindseer: A Portable and Extensible Tool for Visualization of Structural and Functional Neuroimaging Data. In: *BMC Bioinformatics* 8 (2007), S. 389.1–12

[99] Mroz, Lukas ; Hauser, Helwig: RTVR – a Flexible Java Library for Interactive Volume Rendering. In: *IEEE Transactions on Visualization and Computer Graphics*, 2001, S. 279–286

[100] Nakazono, Mikio ; Qiu, Fang ; Borsuk, Lisa A. ; Schnable, Patrick S.: Laser-capture microdissection, a tool for the global analysis of gene expression in specific plant cell types: identification of genes expressed differentially in epidermal cells or vascular tissues of maize. In: *Plant Cell* 15 (2003), Nr. 3, S. 583–596

[101] Noor, Mohamed A. F. ; Zimmerman, Katherine J. ; Teeter, Katherine C.: Data Sharing: How Much Doesn't Get Submitted to GenBank? In: *PLoS Biology* 4 (2006), Nr. 7, S. 1113–1114

[102] Norm DIN 1319: *Grundlagen der Messtechnik - Grundbegriffe*. Januar 1995

[103] Oltvai, Zoltán N. ; Barabási, Albert-László: Life's complexity pyramid. In: *Science* 298 (2002), Nr. 5594, S. 763–764

[104] Özsu, Tamer M. ; Valduriez, Patrick: *Principles of Distributed Database Systems*. 2. Edition. Prentice Hall, 1999

[105] Pal, Nikhil R. ; Pal, Sankar K.: A Review on Image Segmentation Techniques. In: *Pattern Recognition* 26 (1993), Nr. 9, S. 1277–1294

[106] Parker, Steven G. ; Johnson, Christopher R.: SCIRun: a scientific programming environment for computational steering. In: *Proceedings of the 1995 ACM/IEEE conference on Supercomputing*, 1995, S. 2–19

[107] Pelaz, Soraya ; Ditta, Gary S. ; Baumann, Elvira ; Wisman, Ellen ; Yanofsky, Martin F.: B and C floral organ identity functions require SEPALLATA MADS-box genes. In: *Nature* 405 (2000), Nr. 6783, S. 200–203

[108] Pennisi, Elizabeth: How Will Big Pictures Emerge From a Sea of Biological Data? In: *Science* 309 (2005), Nr. 5731, S. 94

[109] Phillips, Carrie L. ; Arend, Lois J. ; Filson, Adele J. ; Kojetin, Doug J. ; Clendenon, Jeffrey L. ; Fang, Shiaofen ; Dunn, Kenneth W.: Three-Dimensional Imaging of Embryonic Mouse Kidney by Two-Photon Microscopy. In: *American Journal of Pathology* 158 (2001), S. 49–55

[110] Pielot, Rainer ; Seiffert, Udo ; Manz, Bertram ; Weier, Diana ; Volke, Frank ; Weschke, Winfriede: 4D Warping for Analysing Morphological Changes in Seed Development of Barley Grains. In: *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2008, S. 335–340

[111] Pieper, Steve ; Halle, Mike ; Kikinis, Ron: 3D SLICER. In: *Proceedings of the 1st IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2004, S. 632–635

[112] Pucknat, Kevin: *Implementierung eines interaktiven Anaglyph-Stereomodus für Java3D*, Hochschule Ostwestfalen-Lippe, Fachbereich Angewandte Informatik, Bachelorarbeit, 2009

[113] Purves, William K. ; Sadava, David ; Orians, Gordon H. ; Heller, H. C.: *Biologie*. 7. Edition. Spektrum Akademischer Verlag, 2006

[114] Rantzau, Dirk ; Lang, Ulrich: A Scalable Virtual Environment for Large Scale Scientific Data Analysis. In: *Proceedings of the Euro VR Mini Conference*, 1997, S. 215–222

[115] Reddy, Venkatramana N. ; Liebman, Michael N. ; Mavrovouniotis, Michael L.: Qualitative analysis of biochemical reaction systems. In: *Computers in Biology and Medicine* 26 (1996), S. 9–24

[116] Rhyne, Theresa-Marie ; Tory, Melanie ; Munzner, Tamara ; Ward, Matt ; Johnson, Chris ; Laidlaw, David H.: Information and Scientific Visualization: Separate but Equal or Happy Together at Last. In: *Proceedings of the IEEE Visualization Conference*, 2003, S. 619–621

[117] Richardson, Lorna ; Venkataraman, Shanmugasundaram ; Stevenson, Peter ; Yang, Yiya ; Burton, Nicholas ; Rao, Jianguo ; Fisher, Malcolm ; Baldock, Richard A. ; Davidson, Duncan R. ; Christiansen, Jeffrey H.: EMAGE mouse embryo spatial gene expression database: 2010 update. In: *Nucleic Acids Research* 38 (2010), Nr. 1, S. D703–D709

[118] Robenek, Horst: *Mikroskopie in Forschung und Praxis*. 1. Edition. Git Verlag, 1995

[119] Rohn, Hendrik ; Klukas, Christian ; Schreiber, Falk: Integration and Visualisation of Multimodal Biological Data. In: *Proceedings of the German Conference on Bioinformatics*, 2009, S. 105–115

[120] Rohn, Hendrik ; Klukas, Christian ; Schreiber, Falk: Creating Views on Integrated Multidomain Data. In: *Bioinformatics* 27 (2011), Nr. 13, S. 1839–1845

[121] Rohn, Hendrik ; Klukas, Christian ; Schreiber, Falk: Visual Analytics of Multimodal Biological Data. In: *Proceedings of the International Conference on Information Visualization Theory and Applications*, 2011, S. 256–261

[122] Rolletschek, Hardy ; Radchuk, Ruslana ; Klukas, Christian ; Schreiber, Falk ; Wobus, Ulrich ; Borisjuk, Ljudmilla: Evidence of a key role for photosynthetic oxygen release in oil storage in developing soybean seeds. In: *New Phytologist* 167 (2005), Nr. 3, S. 777–86

[123] Rolletschek, Hardy ; Weschke, Winfriede ; Weber, Hans ; Wobus, Ulrich ; Borisjuk, Ljudmilla: Energy state and its control on seed development: starch accumulation is associated with high ATP and steep oxygen gradients within barley grains. In: *Journal of Experimental Botany* 55 (2004), Nr. 401, S. 1351–1359

[124] Roos, David S.: Computational Biology: Bioinformatics – Trying to Swim in a Sea of Data. In: *Science* 291 (2001), S. 1260–1261

[125] Royce, Winston W.: Managing the development of large software systems. In: *Proceedings of the IEEE WESCON*, 1970, S. 1–9

[126] Royce, Winston W.: Managing the development of large software systems: concepts and techniques. In: *Proceedings of the 9th international conference on Software Engineering*, 1987, S. 328–338

[127] Rubtsov, Denis V. ; Jenkins, Helen ; Ludwig, Christian ; Easton, John ; Viant, Mark R. ; Günther, Ulrich ; Griffin, Julian L. ; Hardy, Nigel: Proposed reporting requirements for the description of NMR-based metabolomics experiments. In: *Metabolomics* 3 (2007), Nr. 3, S. 223–229

[128] Saladi, Sagar ; Pinnamaneni, Pujita ; Meyer, Joerg: Texture-based 3-D Brain Imaging. In: *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, 2001, S. 136–143

[129] Scharfe, Michael ; Pielot, Rainer ; Schreiber, Falk: Fast multi-core based multimodal registration of 2D cross-sections and 3D datasets. In: *BMC Bioinformatics* 11 (2010), Nr. 20, S. 20.1–11

[130] Schmid, Benjamin ; Schindelin, Johannes ; Cardona, Albert ; Longair, Mark ; Heisenberg, Martin: A high-level 3D visualization API for Java and ImageJ. In: *BMC Bioninformatics* 11 (2010), Nr. 1, S. 274.1–7

[131] Schoor, Wolfram ; Bollenbeck, Felix ; Seidl, Thomas ; Weier, Diana ; Weschke, Winfriede ; Preim, Bernhard ; Seiffert, Udo ; Mecke, Rüdiger: VR Based Visualization and Exploration of Plant Biological Data. In: *Journal of Virtual Reality and Broadcasting* 6 (2009), Nr. 8, S. 1860–2037

[132] Schreiber, Falk: *Visualisierung biochemischer Reaktionsnetze*, Universität Passau, Fakultät für Mathematik und Informatik, Doktorarbeit, 2001

[133] Schreiber, Falk: *Visual Analysis of Biological Networks*, Universität Passau, Fakultät für Mathematik und Informatik, Habilitation, 2006

[134] Schroeder, Will ; Martin, Ken ; Lorensen, Bill: *The Visualization Toolkit*. 3. Edition. Kitware Inc., 2004

[135] Schulze-Döbold, Jürgen P.: *Interactive Volume Rendering in Virtual Environments*, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Doktorarbeit, 2003

[136] Seo, Seongwon ; Lewin, Harris A.: Reconstruction of metabolic pathways for the cattle genome. In: *BMC Systems Biology* 3 (2009), Nr. 33, S. 33

[137] Shah, Sohrab ; Huang, Yong ; Xu, Tao ; Yuen, Macaire ; Ling, John ; Ouellette, Francis B F.: Atlas - a data warehouse for integrative bioinformatics. In: *BMC Bioinformatics* 6 (2005), S. 34.1–16

[138] Shannon, Paul ; Markiel, Andrew ; Ozier, Owen ; Baliga, Nitin S. ; Wang, Jonathan T. ; Ramage, Daniel ; Amin, Nada ; Schwikowski, Benno ; Ideker, Trey: Cytoscape: a software environment for integrated models of biomolecular interaction networks. In: *Genome Research* 13 (2003), Nr. 11, S. 2498–2504

[139] Sommer, Björn ; Künsemöller, Jörn ; Sand, Norbert ; Husemann, Arne ; Rumming, Madis ; Kormeier, Benjamin: CELLmicrocosmos 4.1: An interactive approach to Integrating Spatially Localized Metabolic Networks into a Virtual 3D Cell Environment. In: *Proceedings of BIOSTEC Bioinformatics*, 2010, S. 90–95

[140] Stalling, Detlev ; Westerhoff, Malte ; Hege, Hans-Christian: Amira: A highly interactive system for visual data analysis. In: *The Visualization Handbook*. Academic Press, Inc. Orlando, FL, USA, 2005, Kapitel 38, S. 749–767

[141] Stelling, Jorg ; Kremling, Andreas ; Ginkel, Martin ; Bettenbrock, Katja ; Gilles, Ernst D.: Towards a Virtual Biological Laboratory. In: *Proceedings of the 1st International Conference on Systems Biology*, 2000, S. 29–38

[142] Stevens, Robert ; Goble, Carole ; Baker, Patricia ; Brass, Andy: A classification of tasks in bioinformatics. In: *Bioinformatics* 17 (2001), Nr. 2, S. 180–188

[143] Stryer, Lubert ; Berg, Jeremy M. ; Tymoczko, John L.: *Biochemistry*. 5. Edition. W. H. Freeman, 2002

[144] Stuart, Joshua M. ; Segal, Eran ; Koller, Daphne ; Kim, Stuart K.: A gene-coexpression network for global discovery of conserved genetic modules. In: *Science* 302 (2003), Nr. 5643, S. 249–255

[145] Sujansky, Walter: Heterogeneous Database Integration in Biomedicine. In: *Journal of Biomedical Informatics* 34 (2001), Nr. 4, S. 285–298

[146] Swan, Edward ; Yagel, Roni: Slice-Based Volume Rendering / The Advanced Computing Center for the Arts and Design, The Ohio State University. 1993. – Forschungsbericht

[147] Swedlow, Jason R. ; Goldberg, Ilya G. ; Eliceiri, Kevin W. ; the OME Consortium: Bioimage Informatics for Experimental Biology. In: *Annual Review of Biophysics* 38 (2009), S. 327–346

[148] Taylor, Chris F. ; Paton, Norman W. ; Garwood, Kevin L. ; Kirby, Paul D. ; ; Stead, David A. ; Yin, Zhikang ; Deutsch, Eric W. ; Selway, Laura ; Walker, Janet ; Riba-Garcia, Isabel ; Mohammed, Shabaz ; Deery, Michael J. ; Howard, Julie A. ; Dunkley, Tom ; Aebersold, Ruedi ; Kell, Douglas B. ; Lilley, Kathryn S. ; Roepstorff, Peter ; Yates, John R. ; III ; Brass, Andy ; Brown, Alistair J. ; Cash, Phil ; Gaskell, Simon J. ; Hubbard, Simon J. ; Oliver, Stephen G.: A systematic approach to modeling, capturing, and disseminating proteomics experimental data. In: *Nature Biotechnology* 21 (2003), Nr. 3, S. 247–254

[149] Testi, Debora ; Clapworthy, Gordon ; Aylward, Stephen ; Frangi, Alejandro ; Christie, Richard: Interactive visualization of multiscale biomedical data: an integrated approach. In: *Proceedings of the 1st IEEE Symposium on Biological Data Visualization (BioVis)*, 2011, S. 3–4

[150] Thalmann, Nadia M. ; Thalmann, Daniel: Computer animation. In: *ACM Computing Surveys* 28 (1996), S. 161–163

[151] The Gene Ontology Consortium: The Gene Ontology project in 2008. In: *Nucleic Acids Research* 36 (2008), S. D440–D444

[152] Theissen, Günter ; Saedler, Heinz: Floral quartets. In: *Nature* 409 (2001), Nr. 6819, S. 469–471

[153] Thiele, Ines ; Jamshidi, Neema ; Fleming, Ronan M. T. ; Palsson, Bernhard O.: Genome-Scale Reconstruction of Escherichia coli's Transcriptional and Translational Machinery: A Knowledge Base, Its Mathematical Formulation, and Its Functional Characterization. In: *PLoS Computational Biology* 5 (2009), Nr. 3, S. e1000312

[154] Töpel, Thoralf ; Kormeier, Benjamin ; Klassen, Andreas ; Hofestädt, Ralf: BioDWH: A Data Warehouse Kit for Life Science Data Integration. In: *Journal of Integrative Bioinformatics* 5 (2008), Nr. 2, S. 93.1–9

[155] Tory, Melanie ; Kirkpatrick, Arthur E. ; Atkins, M. S. ; Müller, Torsten: Visualization Task Performance with 2D, 3D, and Combination Displays. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 1, S. 2–13

[156] Visualization and Imagery Solutions, Inc.: *OpenDX: The Open Source Software Project based on IBM's Visualization Data Explorer.* – `http://www.opendx.org/index2.php`

[157] Walter, Thomas ; Shattuck, David W. ; Baldock, Richard ; Bastin, Mark E. ; Carpenter, Anne E. ; Duce, Suzanne ; Ellenberg, Jan ; Fraser, Adam ; Hamilton, Nicholas ; Pieper, Steve ; Ragan, Mark A. ; Schneider, Jurgen E. ; Tomancak, Pavel ; Hériché, Jean-Karim K.: Visualization of image data from cells to organisms. In: *Nature Methods* 7 (2010), Nr. 3 Suppl., S. 526–541

[158] Weise, Stephan: *Integrierte Analyse pflanzenbiologischer Daten unter besonderer Berücksichtigung der Datenqualität*, Universität Bielefeld, Technische Fakultät, Doktorarbeit, 2009

[159] Wohlfahrter, Werner ; Encarnacao, Miguel L. ; Schmalstieg, Dieter: Interactive Volume Exploration on the StudyDesk. In: *Proceedings of the 4th Immersive Projection Technology Workshop*, 2000, S. 1–10

[160] Wollmann, Heike ; Mica, Erica ; Todesco, Marco ; Long, Jeff A. ; Weigel, Detlef: On reconciling the interactions between APETALA2, miR172 and AGAMOUS with the ABC model of flower development. In: *Development* 137 (2010), Nr. 21, S. 3633–3642

[161] Wolsiffer, Kerstin: *Entwurf und Realisierung eines Interaktiven VR-basierten Tools zur Segmentierung und Visualisierung medizinischer Volumendaten*, Deutsches Krebsforschungszentrum Heidelberg, Abteilung Medizinische und Biologische Informatik, Diplomarbeit, 1996

[162] Yi-Ren ; Shiffman, Smadar ; Brosnan, Thomas J. ; Links, Jonathan M. ; Beach, Leu S. ; Judge, Nicholas S. ; Xu, Yirong ; Kelkar, Uma V. ; Reiss, Allan L.: BrainImageJ - A Java-based Framework for Interoperability in Neuroscience, with Specific Application to Neuroimaging. In: *American Medical Informatics Association* 8 (2001), Nr. 5, S. 431–442

[163] Yilmaz, Alper ; Mejia-Guerra, Maria K. ; Kurz, Kyle ; Liang, Xiaoyu ; Welch, Lonnie ; Grotewold, Erich: AGRIS: the Arabidopsis Gene Regulatory Information Server, an update. In: *Nucleic Acids Research* 39 (2011), S. D1118–D1122

[164] Zimmermann, Philip ; Hirsch-Hoffmann, Matthias ; Hennig, Lars ; Gruissem, Wilhelm: GENEVESTIGATOR: Arabidopsis microarray database and analysis toolbox. In: *Plant Physiology* 136 (2004), Nr. 1, S. 2621–2632

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig und ohne fremde Hilfe verfasst habe. Ich habe keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt. Die den benutzten Werken anderer Autoren wörtlich oder inhaltlich entnommenen Stellen sind als solche kenntlich gemacht wurden. Ich habe mich um keinen weiteren Doktorgrad beworben und auch keine vergeblichen Promotionsversuche unternommen.

Hendrik Rohn

Lebenslauf

Hendrik Rohn

Adresse:	Wallstraße 31 06484 Quedlinburg
E-Mail:	hendrikrohn@gmail.com
Geboren am:	26.01.1983
Ort:	Pößneck
Familienstand:	verheiratet, zwei Kinder
Nationalität	Deutsch

Berufliche Tätigkeit

ab 12/2011	wissenschaftlicher Mitarbeiter, SunGene GmbH - A BASF Plant Science Company Gatersleben
05/2008–11/2011	wissenschaftlicher Mitarbeiter und Doktorand, Institut für Pflanzengenetik und Kulturpflanzenforschung Gatersleben

Ausbildung

ab 04/2011	Promotionsstudent, Martin-Luther-Universität Halle-Wittenberg
09/2006–03/2007	Teilnahme an Lehrveranstaltungen des Studiengangs Master of Bioinformatics, Dublin City University
10/2002–02/2008	Studium der Bioinformatik, Friedrich Schiller-Universität Jena, Abschluss mit Diplom, Gesamtprädikat *sehr gut*
07/2001–03/2002	Ableistung des Grundwehrdienstes, Prinz-Eugen-Kaserne Külsheim
bis 06/2001	Erlangung der Hochschulreife, Gymnasium Fridericianum Rudolstadt, Abschlussnote: *2,5*

Quedlinburg, den 05. Juni 2012