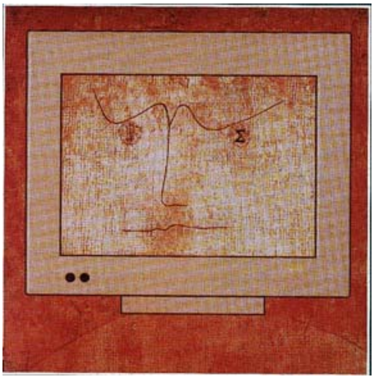
GWDG-Bericht Nr. 68

Kurt Kremer, Volker Macho (Hrsg.)

Forschung und wissenschaftliches Rechnen

Beiträge zum Heinz-Billing-Preis 2004



Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Forschung und
wissenschaftliches Rechnen

Titelbild:
Logo nach Paul Klee „Der Gelehrte“, Modifizierung durch I. Tarim,
Max-Planck-Institut für Psycholinguistik, Nijmegen.

GWDG-Bericht Nr. 68

Kurt Kremer, Volker Macho (Hrsg.)

Forschung und wissenschaftliches Rechnen

Beiträge zum Heinz-Billing-Preis 2004

Gesellschaft für wissenschaftliche Datenverarbeitung
Göttingen

2005

Die Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen ist eine gemeinsame Einrichtung der Max-Planck-Gesellschaft zur Förderung der Wissenschaften e. V. und des Landes Niedersachsen. Die Max-Planck-Gesellschaft hat diesen Bericht finanziell gefördert.

Redaktionelle Betreuung:
Volker Macho (Max-Planck-Institut fuer Polymerforschung, Mainz)

Satz:
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen,
Am Faßberg, D-37077 Göttingen

Druck: klartext GmbH, D-37073 Göttingen, www.kopie.de

ISSN: 0176-2516

Inhalt

Nominiert für den Heinz-Billing-Preis 2004

Weitere Beiträge für den Heinz-Billing-Preis 2004

Vorwort

Der vorliegende zwölfte Band der Reihe „Forschung und wissenschaftliches Rechnen" enthält alle Beiträge, welche für den Heinz-Billing-Preis des Jahres 2004 eingereicht wurden. Der Hauptpreis ging in diesem Jahr an Markus Rampp und Thomas Soddemann vom Rechenzentrum Garching der Max-Planck-Gesellschaft für ihre Arbeit „A Work Flow Engine for Microbial Genome Research". Es handelt sich bei hierbei um eine integrierte Softwareumgebung, welche Wissenschaftlern dabei helfen soll, verschiedenste Werkzeuge für die Analyse von Proteinsequenzen anzuwenden und miteinander zu verknüpfen.

In die Endrunde der Billing-Preis Verleihung kamen des weiteren Johannes Wicht und Julien Aubert vom Max-Planck-Institut für Sonnensystemforschung in Katlenburg-Lindau mit ihrer Arbeit „Dynamos in Action", bei der es um die Modellierung magnetischer Felder in Planetenkernen und um die spannende Frage nach der Ursache von Magnetfeldumkehrungen geht, sowie Niko Beerenwinkel vom Max-Planck-Institut für Informatik, Saarbrücken, mit dem Thema „Computational Tools for the Analysis and Simulation of HIV Drug Resistence". Auch dies ist eine Software-Umgebung, welche verschiedene Simulationswerkzeuge enthält und diese in geeigneter Weise miteinander verbindet.

Die übrigen Arbeiten kamen vom Max-Planck-Institut für Gravitationsphysik in Potsdam, vom Max-Planck-Institut für Dynamik komplexer technischer Systeme in Magdeburg, vom Max-Planck-Institut für molekulare Genetik in Berlin sowie aus dem Institut für Umwelttechnik der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshafen. Sie alle zeigen das hohe Niveau und das breite Spektrum wissenschaftlicher Datenverarbeitung.

Ab dem Jahr 2006 wird der Heinz-Billing-Preis nicht mehr durch die eigens hierfür gegründete Heinz-Billing-Vereinigung, sondern erstmals von der neu gegründeten Heinz-Billing-Stiftung verliehen. Dies ist Anlass für uns, eine kleine Rückblende zu halten. Wir haben daher den wissenschaftlichen Arbeiten die Zusammenfassung eines Vortrages von Dr. Theo Plesser, Vorstandsmitglied der Heinz-Billing-Vereinigung, vorangestellt, den er bei der Festveranstaltung zu Ehren von Prof. em. Heinz Billing anlässlich seines 90. Geburtstages gehalten hat.

Bedanken wollen wir uns ganz herzlich bei Günter Koch, GWDG, für die Umsetzung der eingesandten Manuskripte in eine kompatiblen Druckvorlage.

Die Vergabe des Preises wäre ohne Sponsoren nicht möglich. Wir danken der Firma IBM Deutschland, welche auch 2004 als Hauptsponsor aufgetreten ist, für ihre großzügige Unterstützung.

Die hier abgedruckten Arbeiten sind ebenfalls im Internet unter der Adresse

www. billingpreis.mpg.de

zu finden.

Kurt Kremer, Volker Macho

Anlässlich des 90. Geburtstags
von Prof. em. Heinz Billing

Der Heinz-Billing-Preis
Die Heinz-Billing-Vereinigung
Rückblick 1993 – 2005

Theo Plesser und Peter Wittenburg

1993 wurde der Heinz-Billing-Preis zum erstenmal anlässlich des 10ten DV-Treffens der Leiter der Rechenzentren und der EDV Verantwortlichen in den Instituten der Max-Planck-Gesellschaft vergeben. Im Juli 2005 löst sich die hinter dem Preis stehende Heinz-Billing-Vereinigung e.V. auf, da die Max-Planck-Gesellschaft die Trägerschaft des Preises übernimmt und ihn als „Heinz-Billing-Preis der Max-Planck-Gesellschaft zur Förderung des wissenschaftlichen Rechnens“ fortführt. Diese Zäsur gibt Anlass über die Entstehung des Preises, seine kurze Geschichte und die Preisträger zu berichten.

Das 10jährige Jubiläum der DV-Treffen der EDV- und Rechenzentrumsleiter der Institute der Max-Planck-Gesellschaft im Jahre 1993 wurde mit einem großen Programm gefeiert, groß in dem Sinne, dass aus aller Welt namhafte Wissenschaftler eingeladen wurden, um das Thema „Wissenschaftliches Rechnen in der Forschung“ unter allen Aspekten in den verschiedensten Anwendungsgebieten zu beleuchten. Um der Veranstaltung einen repräsentativen Rahmen zu geben wurde der Veranstaltungsort gewechselt. Vom familiären Seminarraum im Frankfurter Max-Planck-Institut für Hirnforschung ging es in den großen Hörsaal im Max-Planck-Institut für biophysikalische Chemie in Göttingen. Viele „Frankfurter“ hatten in den üppigen Sesseln des Hörsaals das Gefühl, als sei alles etwas zu groß geraten – aber die rasante Ausbreitung der EDV in den Instituten brachte Ambiente und Funktionalität bald ins Gleichgewicht – bis heute finden die DV-Treffen alljährlich im November in Göttingen statt.

Die Organisatoren des 10jährigen DV-Treffens, die Autoren dieses Artikels, die auch das erste „Frankfurter" Treffen 1983 organisiert hatten, wollten nach zehn Jahren nicht nur renommierte Redner aus der Informatik einladen und damit ein kurzes Glanzlicht setzen. Langfristig sollte durch einen jährlich auszulobenden Preis ein Zeichen gesetzt werden. Das Konzept der Auslobung war bald gefunden, es hat sich bis heute nicht geändert:

> „Es sollen die Leistungen derjenigen anerkannt werden, die in zeitintensiver und kreativer Arbeit die notwendige Hard- und Software entwickeln, die für neue Vorstöße in der Wissenschaft unverzichtbar sind."

Das Logo, eine von I. Tarim vom Max-Planck-Institut für Psycholinguistik gestaltete Verfremdung des Bildes „Der Gelehrte" von Paul Klee, war bald entwickelt. Es fehlte aber eine Kennung, ein Markenzeichen, um den Preis zugkräftig in die Öffentlichkeit zu bringen. In einer Diskussion über ein EDV Problem ergab sich plötzlich die Wortsequenz: *Rechner – Programme – BAR – Billing*, die Idee war da – Heinz-Billing-Preis.

Professor Heinz Billing ist emeritiertes wissenschaftliches Mitglied des Max-Planck-Institutes für Astrophysik und er war der erste Vorsitzende des Beratenden Ausschusses für Rechenanlagen (BAR) in der Max-Planck-Gesellschaft und leitete dieses Gremium über zwanzig Jahre. Mit der Erfindung des Trommelspeichers und dem Bau der Göttinger Rechenmaschinen G1, G2, G3 ist er einer der Pioniere der elektronischen Datenverarbeitung. Diese Rechner wurden in den Abteilungen von Werner Heisenberg und Ludwig Biermann am Max-Planck-Institut für Physik zur Lösung numerischer Probleme der theoretischen Physik eingesetzt, sie stehen somit am Beginn des wissenschaftlichen Rechnens.

Mit einem recht ungewissen Gefühl trugen wir unsere Überlegungen Herrn Professor Billing vor und unterbreiteten ihm den Textentwurf für die Auslobung des Preises. Die Zielrichtung des Preises und unseren Ausschreibungstext hat Herr Professor Billing schnell akzeptiert, allerdings bestand er auf der folgenden wesentlichen Ergänzung: „Es können Arbeiten eingereicht werden, die an einem oder in enger Kooperation mit einem Max-Planck-Institut durchgeführt wurden". Diese Eingrenzung sollte sicherstellen, dass der Preis mit seinem Namen nicht mit dem Preis bzw. der Medaille der Gesellschaft für Informatik, die nach seinem Freund Konrad Zuse benannt sind, in Konkurrenz tritt. Ab 2002 war Professor Billing bereit den Vorbehalt fallen zu lassen, da sich inzwischen gezeigt hatte, dass der Heinz-Billing-Preis und der Konrad Zuse Preis ganz verschiedene Zielgruppen anspricht.

Bei der Festlegung des Preisgeldes war zwischen Wunsch und Möglichkeiten abzuwägen, 5000 DM sollten es schon sein, aber auch ein solch relativ kleines Preisgeld will eingeworben sein. Dies erschien uns bei der damals prosperierenden Datenverarbeitungsindustrie kein Problem zu sein. Es war zunächst auch kein Problem, bis die Firmen Spendenbescheinigungen wünschten. Die Max-Planck-Gesellschaft winkte ab, da die Vergabe von Preisen laut Satzung nicht Zweck der Max-Planck-Gesellschaft ist. Es wurde überlegt, eine Stiftung zu gründen, um damit das Preisgeld auf Dauer zu sichern. Das wiederum scheiterte am Stiftungskapital. Das Stiftungskapital muss wenigstens so hoch sein, dass der Stiftungszweck ohne Kapitalverzehr erreicht werden kann. Damit musste der letzte Ausweg beschritten werden – die Gründung eines gemeinnützigen Vereins, der wissenschaftlichen Zwecken dient: Er vergibt den Heinz-Billing-Preis, publiziert die eingereichten Arbeiten und akkumuliert Geldvermögen, um den Verein in eine Stiftung umzuwandeln.

Unter der Bezeichnung „Heinz-Billing-Vereinigung zur Förderung des wissenschaftlichen Rechnens in der Forschung e.V.“ wurde der Verein am 21.11.1996 in Göttingen errichtet und am 24. Februar 1997 in das Vereinsregister beim Amtsgericht Dortmund unter dem Zeichen VR 4860 eingetragen. Gründungsmitglieder waren Prof. Dr. Heinz Billing (emeritiertes wissenschaftliches Mitglied des MPI für Astrophysik), Dipl.-Inf. Stefan Heinzel (Leiter des Rechenzentrum Garching der MPG), Dr. Jürgen Hess (MPI für Bildungsforschung, Berlin), Dr. Volker Macho (MPI für Polymerforschung, Mainz), Prof. Dr. Leo C.M. De Maeyer (emeritiertes Mitglied des MPI für biophysikalische Chemie, Göttingen), Dr. Theo Plesser (MPI für molekulare Physiologie, Dortmund), Prof. Dr. Dieter Wall (wissenschaftlicher Geschäftsführer der Gesellschaft für wissenschaftliche Datenverarbeitung (GWDG) Göttingen) und Dipl.-Ing. Peter Wittenburg (MPI für Psycholinguistik, Nijmwegen).

Als Kuratoren konnten die Herren Prof. H.W. Spiess, Mainz, Prof. Stefan Müller, Leipzig, Prof. F. Hoßfeld, Jülich, Prof. J. Renn, Berlin und Prof. K. Hahlbrock, Köln gewonnen werden.

Den ersten Vorstand bildeten die Herren Plesser (Vorsitz), Heinzel (Stellvertreter) und Wittenburg (Schatzmeister). Das Finanzamt erkannte den Verein als gemeinnützig an und erteilte die Befugnis zur Ausstellung von Spendenbescheinigungen.

Mit den Jahren erhöhte sich die Zahl der Mitglieder von acht auf vierzehn. Eine Ausdehnung des Vereins in Richtung auf eine deutliche Erhöhung der Mitgliederzahl war von den Mitgliedern nicht intendiert. Im Jahre 2002 wurde satzungsgemäß ein neuer Vorstand gewählt: Kremer (Vorsitz), Heinzel (Stellvertreter), Plesser (Schatzmeister). Dieser Vorstand sicherte mit Erfolg den Fortbestand und die Aufwertung des Heinz-Billing-Preises. In Würdigung der bisherigen Preisträger kam die Max-Planck-Gesellschaft zu dem Ergebnis, dass dieser Preis gut zur Gesellschaft passt. In Gesprächen des Vorstands mit der MPG wurde das Statut des „Heinz-Billing-Stiftung“ verhandelt und auf der außerordentlichen Mitgliederversammlung der Heinz-Billing-Vereinigung am 13. Juli 2005 einstimmig angenommen. In § 2 „Zweck der Stiftung“ des Statuts heißt es:

> „Der Zweck der Stiftung wird insbesondere verwirklicht durch die jährliche Vergabe des Heinz-Billing-Preises der Max-Planck-Gesellschaft zur Förderung des wissenschaftlichen Rechnens.“

Die Preisvergabe hat der in §9 des Statuts eingerichtete Stiftungsrat zu regeln.

Gleichzeitig mit der Zustimmung zur Stiftung hat sich die Vereinigung satzungsgemäß selbst aufgelöst und nach § 11 der Satzung das Vermögen[1] der Vereinigung auf die Max-Planck-Gesellschaft übertragen. Die Mittel werden in einer so genannten unselbständigen Stiftung von der Max-Planck-Gesellschaft verwaltet.

Nun zurück zum Bericht über den Heinz-Billing-Preis.

Die erste Preisverleihung fand im November 1993 vor dem gemeinsamen Abendessen des ersten Tages der Veranstaltung in der Kantine des Max-Planck-Instituts für biophysikalische Chemie statt. In seiner Rede zur Verleihung des ersten nach seinem Namen benannten Preises machte Professor Billing die folgende Aussage: „Ohne Anerkennung wird man missmutig und wem trübt das nicht die Freude an der Arbeit. Ich weiß nicht, ob die Herren Wittenburg und Plesser ähnlich gedacht haben, als Sie mir die Idee dieses Preises darlegten. Viel Geld stand nicht zur Verfügung, aber zur Anerkennung können wir beitragen. Ich habe begeistert zugestimmt und gerne meinen Namen zur Benennung des Preises zur Verfügung gestellt“.

In der Hektik haben wir völlig übersehen, dass Professor Billing seine Rede und die Laudatio für die Preisträger vor einem Getränkeautomaten hielt und vor eben diesen auch das erste Foto mit den Preisträgern geschossen wurde. Der Missgriff fiel erst auf, als wir die Bilder in der Hand hielten.

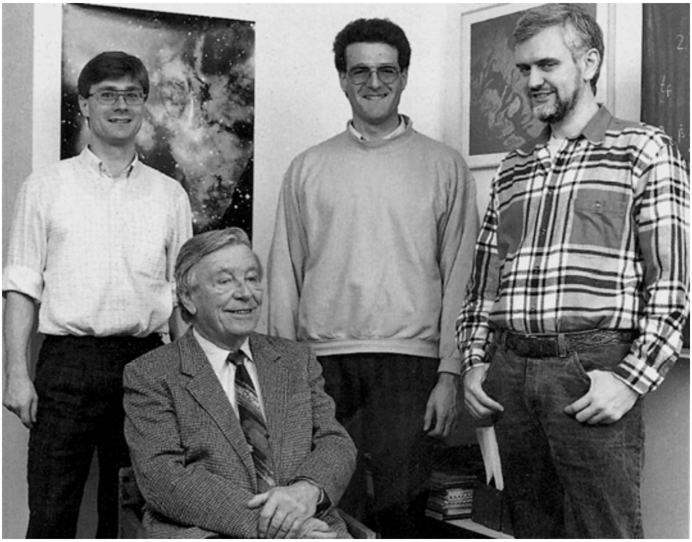
[1] An dieser Stelle sei den Firmen Digital Equipment GmbH, IBM Deutschland GmbH, NEC Europa Supercomputer Systems und Sun Microsystems sehr herzlich für die großzügige Unterstützung der Idee des Heinz-Billing-Preises gedankt.

Da einer der Preisträger aus familiären Gründen nicht an der Preisverleihung teilnehmen konnte, gab es einen guten Grund das offizielle Verleihungsfoto später im Max-Planck-Institut für Astrophysik „nachzustellen".

Bis heute ist der Heinz-Billing-Preis 12 mal vergeben worden an insgesamt 17 Preisträger. Obwohl auch immer mal wieder exzellente Werkzeuge prämiert wurden, so lässt sich die Mehrzahl der preisgekrönten Arbeiten eher unter der Rubrik „neuartige herausragende wissenschaftliche Algorithmen" einordnen. 26 Juroren haben an den Entscheidungen mitgewirkt, wobei zumeist eine erstaunlich hohe Übereinstimmung in den Voten für den Preisträger erzielt wurde. Von 1993 bis 2000 war Professor Billing Mitglied der Jury, und er hat bei den acht Entscheidungen siebenmal ins Schwarze getroffen, d.h. der spätere Preisträger stand auch auf seiner Liste an erster Stelle – er hat ein sicheres Gespür für gute Arbeit, selbst wenn die Thematik weit von seinem wissenschaftlichen Forschungsgebiet entfernt ist !

Die Preisträger 1993 bis 2003.

1993: Thomas Janka, Ewald Müller und Maximilian Ruffert, Max-Planck-Institut für Astrophysik, Garching.
Simulation turbulenter Konvektion in Supernova-Explosionen in massereichen Sternen



Prof. Billing mit Janka, Ruffert und Müller (v.l.)

1994: Rainer Göbel, Max-Planck-Institut für Hirnforschung, Frankfurt
Neurolator – Ein Programm zur Simulation neuronaler Netzwerke

Prof. Billing mit dem Preisträger

1995: Ralf Giering, Max-Planck-Institut für Meteorologie, Hamburg
AMC: Ein Programm zum automatischen Differenzieren von Fortran Programmen

Der Preisträger mit Herrn Prof. Billing

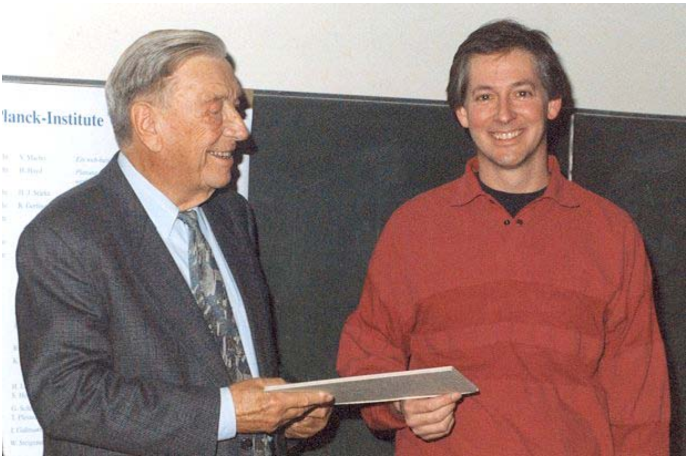
1996: Klaus Heumman, Max-Planck-Institut für Biochemie, Martinsried
Systematische Analyse und Visualisierung kompletter Genome am Beispiel von s. cervisiae.

Der Preisträger mit Herrn Prof. Billing

1997: Florian Müller , Max-Planck-Institut für molekulare Genetik, Berlin: *ERNA-3D (Editor für RNA – Dreidimensional)*

Der Preisträger mit Herrn Prof. Billing

1998: Edward Seidel, Max-Planck-Institut für Gravitationsphysik, Potsdam: *Technologies for Collaborative , Large Scale Simulation in Astrophysics and a General Toolkit for Solving PDEs in Science and Engineering*

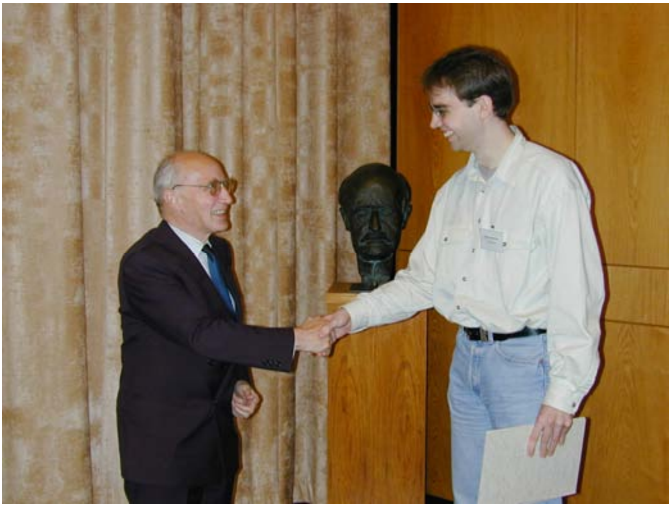


Herr Prof. Billing mit dem Preisträger

1999: Alexander Pukhov, Max-Planck-Institut für Quantenoptik, Garching: *Three-dimensional relativistic electromagnetic Particle-in-Cell code VLPL – Virtual Laser Plasma Laboratory*

Der Preisträger mit Herrn Prof. Billing

2000: Oliver Kohlbacher, Max-Planck-Institut für Informatik, Saarbrücken: *BALL- A Framework for Rapid Application Development in Molecular Modeling*



Prof. De Maeyer mit dem Preisträger

2001: Jörg Haber, Max-Planck-Institut für Informatik, Saarbrücken *MEDUSA -Ein Software-System zur Modellierung und Animation von Gesichtern*

Der Preisträger mit Prof. Kremer

2002: Daan Broeder, Hennie Brugmann und Reiner Dirksmeyer, Max-Planck-Institut für Psycholinguistik, Nijmegen
NILE – Nijmegen Language Resource Environment

Prof. Kremer mit Daan Broeder

2003: Roland Chrobok, Sigdur Hafstein und Andreas Pottmeier, Institut für Theoretische Physik, Universität Duisburg-Essen
OLSIM – A New Generation of Traffic Information Systems

Prof. Kremer mit Roland Chrobok (li.) und Sigdur Hafstein

Der Heinz-Billing-Preis 2004

Ausschreibung des Heinz-Billing-Preises 2004 zur Förderung des wissenschaftlichen Rechnens

Im Jahre 1993 wurde zum ersten Mal der Heinz-Billing-Preis zur Förderung des wissenschaftlichen Rechnens vergeben. Mit dem Preis sollen die Leistungen derjenigen anerkannt werden, die in zeitintensiver und kreativer Arbeit die notwendige Hard- und Software entwickeln, die heute für neue Vorstöße in der Wissenschaft unverzichtbar sind.

Der Preis ist benannt nach Professor Heinz Billing, emeritiertes wissenschaftliches Mitglied des Max-Planck-Institutes für Astrophysik und langjähriger Vorsitzender des Beratenden Ausschusses für Rechenanlagen in der Max-Planck-Gesellschaft. Professor Billing stand mit der Erfindung des Trommelspeichers und dem Bau der Rechner G1, G2, G3 als Pionier der elektronischen Datenverarbeitung am Beginn des wissenschaftlichen Rechnens.

Der Heinz-Billing-Preis zur Förderung des wissenschaftlichen Rechnens steht unter dem Leitmotiv

„EDV als Werkzeug der Wissenschaft“.

Es können Arbeiten eingereicht werden, die beispielhaft dafür sind, wie die EDV als methodisches Werkzeug Forschungsgebiete unterstützt oder einen neuen Forschungsansatz ermöglicht hat.

Der folgende Stichwortkatalog mag den möglichen Themenbereich beispielhaft erläutern:

- Implementation von Algorithmen und Softwarebibliotheken
- Modellbildung und Computersimulation
- Gestaltung des Benutzerinterfaces
- EDV gestützte Meßverfahren
- Datenanalyse und Auswertungsverfahren
- Visualisierung von Daten und Prozessen

Die eingereichten Arbeiten werden referiert und in der Buchreihe „Forschung und wissenschaftliches Rechnen“ veröffentlicht.

Die Jury wählt einen Beitrag für den mit € 3000,- dotierten Heinz-Billing-Preis 2004 zur Förderung des wissenschaftlichen Rechnens aus. Die Beiträge zum Heinz-Billing-Preis, in deutscher oder englischer Sprache abgefasst, müssen keine Originalarbeiten sein und sollten möglichst nicht mehr als fünfzehn Seiten umfassen.

Da zur Bewertung eines Beitrages im Sinne des Heinz-Billing-Preises neben der technischen EDV-Lösung insbesondere der Nutzen für das jeweilige Forschungsgebiet herangezogen wird, sollte einer bereits publizierten Arbeit eine kurze Ausführung zu diesem Aspekt beigefügt werden.

Der Heinz-Billing-Preis wird jährlich vergeben. Die Preisverleihung findet anlässlich des 21. EDV-Benutzertreffens der Max-Planck-Institute am 18. November 2004 in Göttingen statt.

Beiträge für den Heinz-Billing-Preis 2004 sind bis zum 1. Juni 2004 einzureichen.

Heinz-Billing-Preisträger

1993: Dr. Hans Thomas Janka, Dr. Ewald Müller, Dr. Maximilian Ruffert
Max-Planck-Institut für Astrophysik, Garching

Simulation turbulenter Konvektion in Supernova-Explosionen in massereichen Sternen

1994: Dr. Rainer Goebel
Max-Planck-Institut für Hirnforschung, Frankfurt

- Neurolator - Ein Programm zur Simulation neuronaler Netzwerke

1995: Dr. Ralf Giering
Max-Planck-Institut für Meteorologie, Hamburg

AMC: Ein Werkzeug zum automatischen Differenzieren von Fortran Programmen

1996: Dr. Klaus Heumann
Max-Planck-Institut für Biochemie, Martinsried

Systematische Analyse und Visualisierung kompletter Genome am Beispiel von S. cerevisiae

1997: Dr. Florian Mueller
Max-Planck-Institut für molekulare Genetik, Berlin

ERNA-3D (Editor für RNA-Dreidimensional)

1998: Prof. Dr. Edward Seidel
Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Potsdam

Technologies for Collaborative, Large Scale Simulation in Astrophysics and a General Toolkit for solving PDEs in Science and Engineering

1999: Alexander Pukhov
Max-Planck-Institut für Quantenoptik, Garching

High Performance 3D PIC Code VLPL:
Virtual Laser Plasma Lab

2000: Dr. Oliver Kohlbacher
Max-Planck-Institut für Informatik, Saarbrücken
BALL – A Framework for Rapid Application Development in Molecular Modeling

2001: Dr. Jörg Haber
Max-Planck-Institut für Informatik, Saarbrücken
MEDUSA, ein Software-System zur Modellierung und Animation von Gesichtern

2002: Daan Broeder, Hennie Brugman und Reiner Dirksmeyer
Max-Planck-Institut für Psycholinguistik, Nijmegen
NILE: Nijmegen Language Resource Environment

2003: Roland Chrobok, Sigurður F. Hafstein und Andreas Pottmeier
Universität Duisburg-Essen
OLSIM: A New Generation of Traffic Information Systems

2004: Markus Rampp, Thomas Soddemann
Rechenzentrum Garching der Max-Planck-Gesellschaft, Garching
A Work Flow Engine for Microbial Genome Research

Das Kuratorium des Heinz-Billing-Preises

Prof. Dr. Heinz Billing
Emeritiertes Wissenschaftliches Mitglied des Max-Planck-Institut für Astrophysik, Garching

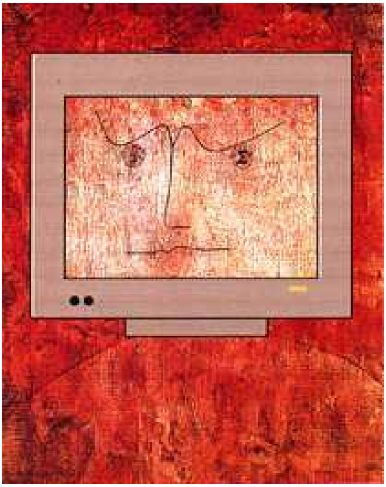
Prof. Dr. Friedel Hossfeld
Forschungszentrum Jülich GmbH, Jülich

Prof. Dr. K. Ulrich Mayer
Max-Planck-Institut für Bildungsforschung, Berlin

Prof. Dr. Stefan Müller
Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig

Prof. Dr. Jürgen Renn
Max-Planck-Institut für Wissenschaftsgeschichte, Berlin

Prof. Dr. H. Wolfgang Spiess
Max-Planck-Institut für Polymerforschung, Mainz



Markus Rampp und Thomas Soddemann,
Rechenzentrum Garching der Max-Planck-Gesellschaft,

erhalten den

Heinz-Billing-Preis 2004
zur Förderung
des wissenschaftlichen Rechnens

als Anerkennung für ihre Arbeit

A Work Flow Engine for Microbial Genome Research

Laudatio

Der Heinz Billing-Preis des Jahres 2004 wird für das Programmpaket *A Work Flow Engine for Microbial Genome Research* verliehen. Für den Anwender ist es oft problematisch, die Vielzahl vorhandener Bioinformatik Tools systematisch und effizient zu nutzen. Mit dem ausgezeichneten Programmpaket, das in enger Zusammenarbeit mit Anwendern aus unterschiedlichen Max-Planck-Instituten entstanden ist, wurde eine Umgebung geschaffen, die es den Wissenschaftlern ermöglicht, eigene komplexe Fragestellungen zu entwickeln und effizient mit sehr verschiedenen Software Tools, die von Homologie-Analysen mittels Datenbanken bis zur Simulation der Tertiärstruktur von Proteinen reichen, zu bearbeiten. Dabei wurde sowohl auf die einfache Handhabbarkeit durch die Benutzer als auch auf die Fähigkeit, sehr große Datenmengen effizient verarbeiten zu können, besonderer Wert gelegt.

Durch ein Design, das hohe Wartungsfreundlichkeit mit klaren Erweiterungskonzepten verbindet, wurde so ein sehr schlagkräftiges und zukunftsfähiges Werkzeug entwickelt.

Herr Prof. Kurt Kremer überreicht die Urkunde zum Heinz-Billing-Preis 2004 an Markus Rampp und Thomas Soddemann

A Work Flow Engine for Microbial Genome Research

Markus Rampp and Thomas Soddemann

Rechenzentrum Garching der Max-Planck-Gesellschaft

Abstract

We present architecture, design and application of a highly modular engine for processing complex workflows of computational tasks. The specific implementation of this engine focuses on the integration of a wide range of bioinformatics software packages into a single system tailored to microbial genome research. The corresponding web application provides a user-friendly interface to all integrated tools and offers comprehensive functionality required for the analysis of microbial genomes. Individual tools can be chained in pipelines allowing users to efficiently handle complicated workflows involving large datasets.

Acknowledgement

This work has been supported by the members of the MiGenAS[1] project:

[1] Research groups from different Max-Planck-Institutes working on microbial genome analysis have formed the so-called MiGenAS (Microbial Genome Analysis System) consortium in order to promote this area of research in Germany, together with the Center for Biotechnology (CeBiTec) at University of Bielefeld as a partner.

D. Oesterhelt, F. Pfeiffer, M. Falb, C. Garcia-Rizo, J. Wolfertz, MPI of Biochemistry
A. Lupas, S. Schuster, G. Raddatz, T. Frickey, J. Soeding, MPI for Developmental Biology
D. Amann, F.O. Gloeckner, M. Bauer, MPI for Marine Microbiology
T. Lengauer, I. Sommer, MPI for Computer Science
H. Lederer, R. Tisma, Garching Computing Center of the MPG
We also thank the MPG for providing a dedicated project position for one of us (M. R.).

1 Introduction

Although number and sophistication of software tools for bioinformatics applications is rapidly increasing, scientists who would like to efficiently utilize these tools for their work still suffer from a number of general problems: Whereas being useful from an algorithmic point of view many tools are difficult to handle, and that does not only hold for the non-expert. Most often, this is caused by non-standard user interfaces and the existence of many proprietary input/output formats. In those cases additional technical knowledge and tools for data conversion are required. Working in such a heterogeneous environment can become enormously tedious and is also inherently error-prone.

Meanwhile, for many of the popular bioinformatics tools publicly accessible servers[2] are offered over the internet. They provide web applications relieving the end-user of the burden of locally installing and maintaining software and databases and to learn about subtleties of their handling. The user, however, has in general no influence on the form of the interfaces (which could well be tailored to a specific class of applications) nor is the availability and stability of the provided services and datasets guaranteed. Lacking adequate computing resources, the more popular servers are in fact often dramatically overloaded rendering them useless for the solution of large-scale problems as well as for the many small tasks which require instant response. Moreover, in the vast majority of cases, sites focus on merely maximizing the number and functional broadness of available tools without promoting interoperability of the individual software packages. But in fact, rather than employing a single monolithic tool many problems in bioinformatics require the setup of a whole processing pipeline by chaining a number of different tools and defining appropriate filters. The presence of a variety of mostly non-standard data formats and modes of delivery for the individual results (e.g., it is a common practice to receive results as plain ASCII text via e-mail) again excludes the majority of public services found in the internet as the method of choice for such tasks.

[2] Among the most prominent are the portals of the National Center for Biotechnology Information, NCBI (http://www.ncbi.nlm.nih.gov) and the European Bioinformatics Institute, EBI (http://www.ebi.ac.uk)

Driven by the specific demands of the scientists of the MiGenAS consortium and motivated by the lack of existing solutions we have developed a framework for a workflow engine which integrates a growing selection of state-of-the-art bioinformatics software tools and databases into a single system. Based on this framework we provide a powerful web application with a coherent and easy-to-use interface to a variety of tools and databases, tailored to the analysis of microbial genomes. The central aim is to give researchers the possibility to efficiently investigate new and more complex scientific questions with a minimum of technical effort and to obtain structured answers within a reasonable amount of time.

The MiGenAS workflow engine offers the classic bioinformatics tasks such as homology searches in databases of nucleic-acid or amino-acid sequences, the computation and validation of multiple sequence alignments, phylogenetic analysis, as well as modeling the secondary and tertiary structure and biochemical function of proteins or parts of them. In particular, the user can seamlessly chain individual tools without the necessity to take care of any format conversions or tedious collection and interpretation ("parsing") of intermediate results. An outstanding feature is the ability to conveniently process large datasets (up to the scale of a complete microbial genome) with a minimum of user interaction. Importantly, sessions with the web application can be made persistent allowing the scientist to easily and reliably reproduce, reexamine and reprocess obtained results.

Web Services complement the picture and allow a complete access to the functionality of the workflow engine. By utilizing web services, the user is able to automate even the most complex workflows.

The demand for adequate software solutions for the highlighted problems has certainly been recognized in the bioinformatics community and beyond. Attempts to provide web-based, integrated toolkits for various kinds of applications are meanwhile receiving considerable attention by developers and also funding agencies, not at least due to their expected high scientific impact. Among the most prominent achievements in this direction is the web portal of the "Helmholtz Network of Bioinformatics" (HNB, [23]), which offers a large variety of services, however, with a different focus than our approach. Interesting new developments towards supporting bioinformatics workflows are found within the "Taverna Project" [20].

The MiGenAS project was launched in summer 2002, development of this framework and application started from scratch in early 2003. The web application for microbial genome analysis was released in early 2004 and is freely accessible for Max-Planck researchers via the MiGenAS web portal [1]. As of the beginning of 2005 public access to the MiGenAS environment is provided to the international scientific community via the web portal http://www.migenas.org/.

Architecture, design and implementation of the workflow engine is portrayed in the following section. In the subsequent section the web application for microbial genome analysis is described together with an actual scientific application. The last section contains the discussion and gives an outlook on future extensions.

2 Workflow Engine Design

A workflow is a well defined progression of individual tasks in a common context which is given by the underlying problem. In bioinformatics, for example, this could be the successive application of different software tools for finding a particular set of genomic sequences in databases, followed by the construction of a multiple sequence alignment, which in turn is eventually translated into an evolutionary tree (see below).

A workflow engine, on the other hand, is a piece of software that provides the implementation of a workflow (the workflow system), monitors its state by the help of agents, and manages the scheduling of tasks. The workflow engine may support configuration or set up of a workflow and its tasks at runtime, as it does in the present case. The MiGenAS workflow engine employs a workflow engine pattern as sketched in Figure 1, and, in addition, provides the user with two means of interacting with the engine and defining workflows – web application and web services.

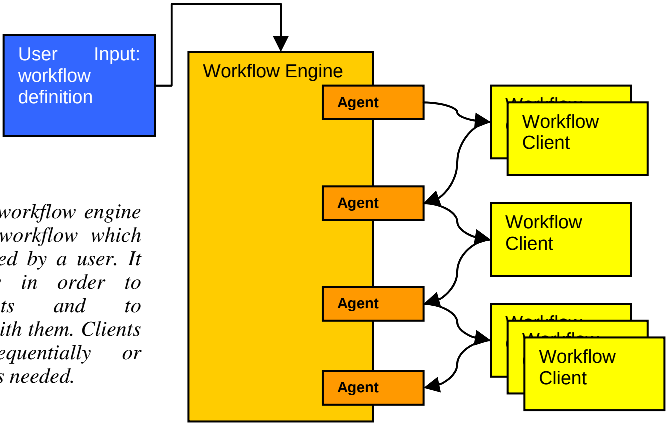


Figure 1: The workflow engine implements a workflow which has been defined by a user. It utilizes agents in order to spawn clients and to communicate with them. Clients can run sequentially or concurrently, as needed.

2.1 Specification and design goals

Making it maintainable and extensible: Many of today's applications in natural sciences lack maintainability. They were originally designed to suit and solve a single, specific task and have sometimes evolved over a long period of time. Extension and maintenance was hardly a matter at design time. However, very often such applications are indeed extended and hence are in danger to become a patchwork of different not well fitted parts with ill defined interfaces. Those applications are neither maintainable nor extensible with a reasonable amount of work. The aim of the MiGenAS workflow engine project is indeed different. Maintainability is an issue from first design steps and extensibility is a primary goal. The system is required to provide the flexibility to add new functionalities at any point of time in the development process and even at the time of deployment.

Applying useful design pattern: In natural sciences, the awareness of Multi-Tier applications is usually weak. For example, many of the currently deployed web applications rely on so called Common Gateway Interface (CGI) scripts, which act isolated from any session aspect or security context and combine several logically distinct components, such as Controller, View and Model parts (see below) into a single execution unit. On the other hand there are various projects in computer and economic sciences which indeed use modern and widely accepted[3] architectural patterns such as the Model-View-Controller pattern. Using such patterns simplifies the design and helps to avoid mistakes. In the design stage of the project all interfaces between the tiers will be defined and tested for plausibility. Then the project can be split up in its natural parts and hence allows a parallel development and independent testing of the different parts, especially the tiers.

Providing a functional, intuitive, user oriented Web Interface: The look-and-feel of a web application, the ease of navigation, and, of course, the given functionality are the points on which a web application will be judged by a user. The aim of this project is to provide the user with an attractive und useful view, with intuitive and coherent navigation, and comprehensive functionality.

Service Oriented Architecture and extensibility towards a future Grid Service Architecture standard: Web services become more and more important. They will form the major interaction point in computing grids [2]. At present specifications for grid architectures, which are based on the use of pure web services are being written. Leading companies like Oracle, IBM, Microsoft, and HP are deeply involved in that process. Hence it would be disadvantageous concerning extensibility to neglect the potential of web services and web service related architectures.This project aims at providing

[3] "Modern and widely accepted" in this context means that the design is well tested in the commercial environment.

a web service oriented access to the workflow engine based on the current developments pushed by the Global Grid Forum [2]. Since there is hardly any specification and reference implementation other than the pure web service layer itself, the project is currently offering proprietary web service access which will change as suitable specifications and reference implementations become available (likely candidates are [3], [4]).

2.2 Building Blocks

The MiGenAS workflow engine consists of several, discrete building blocks. This has the advantage that interfaces between these blocks are clearly defined, and the blocks themselves can easily be developed in parallel. The building blocks are arranged in three layers (Figure 2). A bottom layer, the core layer, incorporates all given software components such as databases, batch system, application containers and the primary bioinformatics tools. Components of this layer have been supplied by third party vendors/developers, and required only minimal configuration in order to work with the workflow engine application. The middle layer, the abstraction layer, consists of the core library providing the encapsulation of the bioinformatics tools and their infrastructure and the enterprise application. The latter one makes use of the core library by utilizing objects which, e.g., are exchanged with the other layers by interface methods. The top layer implements the user interfaces, i.e., the web application and the

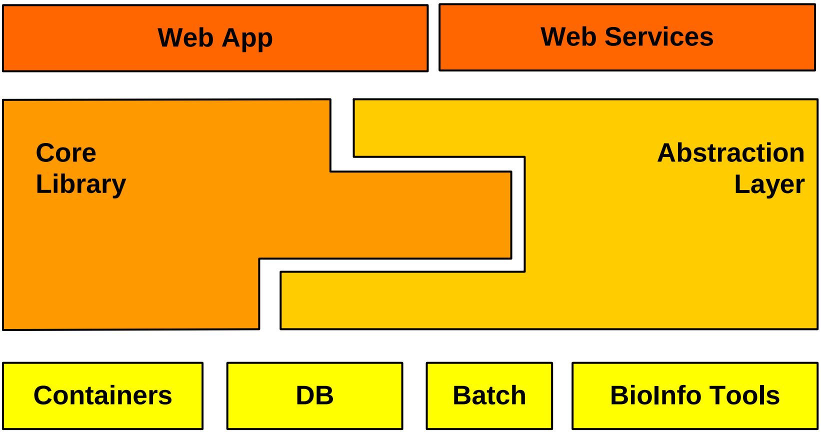


Figure 2: The building blocks of the MiGenAS workflow engine.

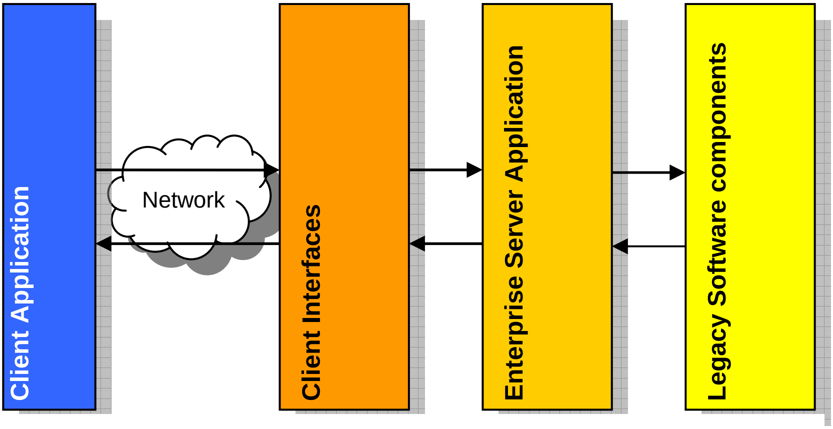


Figure 3: In the Four-Tier Model, the client interfaces, the enterprise application, and the legacy software components are separated and communicate via a few, well-defined interface objects.

web services. Both make use of the core library's objects in order to communicate with the enterprise application and, in the case of web services, with the client via SOAP (the simple object access protocol).

The MiGenAS workflow engine further employs a four-tier model, consisting of its natural parts as sketched in Figure 3. Only the client interface tier (web application and web services) is directly exposed to the client application (i.e., a web browser or a custom, web service oriented client application) where all requests from and responses to the user are handled. The client interfaces (web application and web services) are also exposed to the next tier, the enterprise application, which implements the so-called business model of the application. This model contains all functionality which is not directly related to user interaction.

The enterprise application provides methods to initiate data processing tasks like storing session data, or for executing a bioinformatics tool. The fourth tier consists of the various legacy applications which form the backend of the infrastructure. This comprises the database application, the batch system, and, of course, the various supported bioinformatics tools. Usually, the enterprise application acts only on this fourth layer. In this case, some of the legacy software tools are equipped with a programmatic layer which is aware of the enterprise application and can communicate with it. Examples are the Java representations of all integrated bioinformatics tools which are able to contact the enterprise application in order to store data or update their execution status.

2.2.1 The Web Application

The web application is the primary user interface accessible via a web browser over the internet. It is the easiest way to access the workflow engine. At login a session is set up for a new user, a returning user may choose to continue an old one. During a session the user creates workflows by successively selecting individual tools and using the output of some tool as input for another tool. The user is able to alter workflows to his likings and restart processing with results from any stage. Data are made persistent using the persistence layers/mechanisms of the enterprise application (abstraction layer). Details about layout and functionality of the user interface are given in the next section.

The design of the web application applies the so-called "Model-View-Controller (MVC)" pattern (e.g. [22]). This standard design pattern describes an application where client requests are routed to a controller component, which in turn assembles the response by utilizing the underlying (business-) model. A view component renders processing results and returns a web page to the client. Figure 4 shows a sketch of the MVC pattern in the present context. The model is implemented by the enterprise application hosted by the EJB container. The web application's controller and helper components are implemented by Java servlets, the views are rendered by the help of the JavaServer Pages (JSP) technology ([15], [21]). Java Beans compliant classes encapsulate the input provided by the user as well as the output of individual tools to be rendered by the view components. Modularity and readability of the individual JavaServer pages is established by taking advantage of standard tag libraries ([15], [21]). For special purposes not covered by available libraries, a small custom tag library has been developed to support the rendering of the view and to simplify the work of the web designer.

2.2.2 The Web Services

Web services are the port to the future. As pointed out in the previous section web services are going to dominate the communication between programs on computing and data grids. Making use of this technology is the foundation for a future implementation of the currently emerging Grid standard. Workflows will be realized by combining different web services with matching in- and output definitions. The user can utilize the web services by client applications, which are either self-made, based on this project's Java libraries, or by using supplied (example) stub-code. This project offers web services which can be combined to assemble complex workflows. The communication protocol is designed to be easily adjustable to upcoming Grid service architecture standards.

The architecture of the MiGenAS workflow engine framework is open in this respect and even allows the integration of remote tools via web services. To our knowledge this is a unique feature, which could become a standard procedure as application architects become more and more aware of the benefits of web services.

2.2.3 The Core Library

The Core Library encapsulates bioinformatics structures and provides an object oriented view of bioinformatics tools and especially their data. It contains groups of interfaces and classes suiting that purpose. A judicious use of the Java interface concept[4] leaves great flexibility for the actual implementation of the classes, and hence allows, e.g., re-factoring by changing only a minimum of lines of code. The core library is used by the enterprise application, but it does not depend on the latter. It is an independent middle layer building block of its own.

2.2.4 The Enterprise Application

The enterprise application is the heart of the MiGenAS workflow engine[5]. It takes care of project/session management, scheduling of (compute-) jobs, and bridges the gap to the batch system. The enterprise application is implemented in form of a J2EE [9] application which represents the engine's (business-) model and consists of various Enterprise Java Beans (EJBs). These partly make use of the core library, add functionality to the web application, and manage the persistency of data. Access to the abstraction layer is granted either synchronously via Session EJBs or asynchronously utilizing Message Driven EJBs in combination with the Java Messaging Service and the CORBA[6] Event mechanism [10]. Remote persistent data access[7] is handled by so-called Session Facades and "Data Transfer" or "Value" Objects.

[4] Interfaces are actually present in some OO languages other than Java, and can be used even by languages like C++, although C++ does not explicitly provide the interface language construct.

[5] One could even claim that the enterprise application is the workflow engine and the rest "just" commodities.

[6] CORBA (Common Object Resource Broker Architecture, [10]) is a widely accepted standard for remote object oriented access, independent of the implementation language. Here, C++ on the batch system side and Java on the server side are employed.

[7] Access from a "client" like the web application to data stored in a database and managed by the enterprise application.

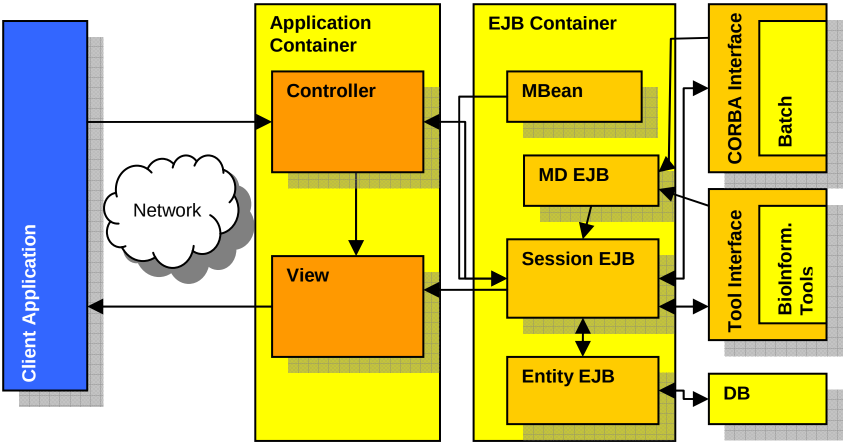


Figure 4: Overview of the fundamental architecture and design of the MiGenAS workflow engine. Interaction of the different components is indicated by arrows.

Project/Session Management beans provide the means to store session relevant data and make them available again, e.g. upon the user's next login to the web application[8].

Existing bioinformatics ("legacy"-) applications are wrapped by the "Bio-Tool" interface and the implementing Java classes which constitutes the basic interface to the enterprise application. In a broader context this interface can be seen as the combination of EJBs and the programmatic layer around those tools which enables bilateral communication between EJBs and tools. To be more precise, the enterprise application serves as the task repository as well as the input- and output-data cache. A bioinformatics tool retrieves its tasks and data or meta-data (whatever applicable in the particular case) from the enterprise application. In return, the bioinformatics tools store their execution state (asynchronously) and any resulting data or meta-data (synchronously and transactional) using the enterprise application. The latter takes care of associating the tool output with the according project. As a (desired) side-effect, this approach allows high throughput computing, since tool tasks are deployed using a batch system in the present case. Deployment to other Grid sites by the use of web service is likely to be included in the future.

[8] The term "project management" is preferred to "session management", since the user, who created such a project, is able to grant session-access also to a defined set of other users ("project members").

The connection to the batch system is established via a CORBA [10] interface in combination with the Java Connector Architecture (JCA). This interface hides the actual batch system from the enterprise application. Such architecture allows the actual batch system to be exchanged without the need of modifying a single line of code in the enterprise application.

Figure 4 sketches the communication of a client with the different components of the MiGenAS workflow engine. The web application receives HTTP requests from the client and invokes appropriate methods on session EJBs of the enterprise application. It then forwards a modified request to the view component, which eventually renders the response and returns it in form of a HTTP response to the client[9]. The session beans, contacted by the web application's controller, (so-called session facades) use other session beans in order to start jobs, collect results, query the batch system and handle the persistent storage, represented by the corresponding entity beans. The message driven beans asynchronously work on messages sent by the external components, i.e., the wrapping layers around third party applications. Management beans (MBeans) monitor the deployed beans as well as the application container and provide service tasks, such as timer services for recurrent queries.

2.2.5 Third Party Software

The Application Containers host web applications and enterprise applications. Since this project follows the J2EE specifications, the web application and the enterprise application are not dependent on a specific application container implementation. For the MiGenAS workflow engine the following setup is currently being used:

- *JBoss EJB Container [11]:* The JBoss EJB container is, at present, certainly the best open source J2EE compliant container available [28]. It is easy to manage, performs reasonably well and offers features like clustering, which are expensive extensions to most of the commercial application containers.
- *Apache Jakarta Tomcat Web Application Container [13]:* The Tomcat Web Application Container has the same reputation in the world of Java web-application hosting as JBoss does have as an EJB Container. In fact, recent versions of JBoss are being shipped bundled with Tomcat.

[9] Web services behave very similar to the web application, although any one of them is not as complex as the entire web application. For the ease of explanation, web services are not referred to, here.

– *Apache Axis Web Services Framework [16]:* Axis deploys as a web application within the Tomcat Web Application Container. It is used to publish all web services provided. A special web application is built on Axis which interprets web service communication, initiates appropriate actions in the enterprise application and returns results.
– *Apache HTTP Server [14]:* Apache's famous web server is used to route all http requests to the application container (Tomcat), which hosts the MiGenAS workflow engine's web application and the web services.

The Batch System takes care of all computationally intensive tasks (jobs) of the MiGenAS workflow engine. It schedules these jobs and dispatches them to the available computing facilities. Usually, batch systems are a legacy part of software and provide only a native interface. This project utilizes the native API of such a batch system and allows (via an abstraction layer) a standardized object oriented remote access to it. Hence, we do not depend on specific batch system software, provided it is possible to wrap a batch system's native API with the abstraction layer.

– *Sun Grid Engine (SGE) and the MiGenAS CORBA interface:* The Sun Grid Engine (SGE, versions 5.3 and 6.0b) is used at the Rechenzentrum Garching (RZG) as the batch system for the Linux Clusters. The MiGenAS workflow engine project provides a CORBA interface to SGE's native "C"-API. Hence, jobs can be scheduled by directly contacting the qmaster daemon, rather than using command line tools, job script files, and UNIX root privileges in order to swap user identities. The CORBA interface can have implementations for almost any batch system[10] [11] presently available.

The Database stores all data which have to be persistent.

– *MySQL database system:* The MySQL database has been chosen for reasons of simplicity, previous good experiences, suitable performance, and because it is Open source software.

Note, that the MiGenAS workflow engine is not dependent on MySQL databases. In case the performance does not scale with the number of users anymore, database systems like IBM's DB/2 or Oracles 10g can be easily used as well. In fact, any database system can be used provided it supports access from the EJB Container (e.g. via JDBC). A future version might make use of the Java Data Object (JDO) Specification rather than Container

[10] The same CORBA interface with an implementation for IBM's Load-Leveler batch system for the Regatta System at RZG has already been developed, but is not used at present (see, http://www.rzg.mpg.de/computing/IBM_P/hardware.html).

[11] Version 6.0 of the SGE provides a Java API.

Managed Persistent Entity Beans. But again, even a change in the persistence model will be limited to a few lines of code, due the nature of the chosen architecture and design.

The Bioinformatics tools are the work horses of the MiGenAS work flow engine. Most of these tools have been designed by their authors as stand-alone tools or packages. They usually have to be invoked from the command line, take files as input and write their output to the console or into other files. Generally speaking, this software has originally been designed with some other architecture in mind than J2EE, CORBA or .NET. Hence, these tools had to be equipped with a programmatic layer or interface in order to be able to communicate and work within the MiGenAS workflow engine.

3 Layout and Functionality of the Web Application

Based on the design outlined in the previous section we have implemented a specific workflow engine which integrates a wide range of bioinformatics software tools required for microbial genome research. This section deals with the corresponding web application which can be accessed with a web browser over the internet [1]. Taking the perspective of a user of this system we first give a brief description of the basic concepts the web application employs, followed by a portrait of the most important features of the graphical user interface. Subsequently, a fictitious user who is addressing an actual scientific problem with the help of this web application is followed on a tour through the system, thereby highlighting the benefits of this system for the researcher.

3.1 Basic concepts

"Tools": The user inserts the input parameters into HTML forms which are offered for each of the bioinformatics tools supported by the web application. Logical consistency and plausibility of the input is checked as far as possible on the client-side (using JavaScript) at submission time. Further checks are made on the server side where the input is parsed. In case of inconsistencies appropriate error messages or warnings are displayed allowing the user to revise the input. Note, that all selections can be reviewed at any time. Although the benefit of this feature is obvious (and, at least with the design chosen for the web application, easy to accomplish) it is not commonly supported by popular bioinformatics web servers. From a formal perspective this functionality might be regarded as a

fundamental, first level of persistency for the user's work with two further persistency levels to be described below.

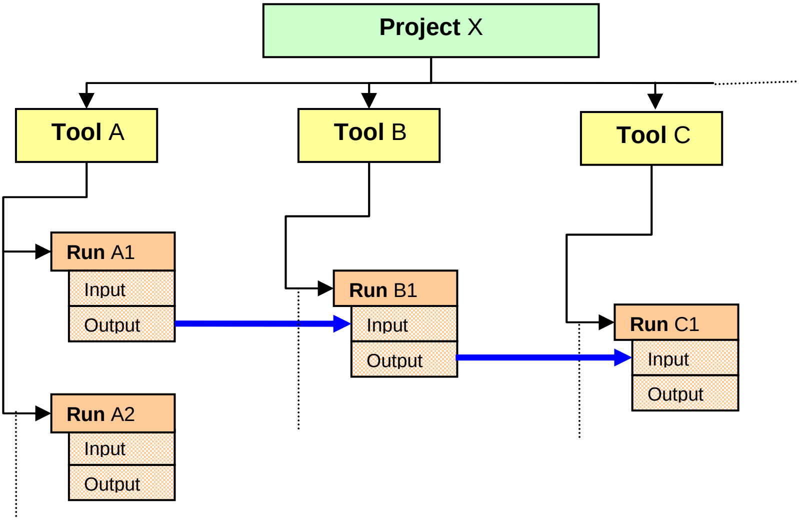


Figure 5: Overview of basic concepts employed for the web application. Three persistency levels (see the text for details) are indicated by different colors. Within a given "project" the user creates "runs" for selected "tools" (basic workflow is indicated by black arrows). Tool pipelines are set up by relating input and output of runs performed with different tools (blue arrows).

"Runs": Instead of iterating on the same input form by varying input parameters and rerunning the tool (and consequently losing the history of trials) the parameter space of interest can systematically and reproducibly be explored by creating a new "run" for each relevant combination of parameters. This means that the user is supplied with a new, empty input form which allows to work independently of the state of other runs.

The notion of a "run" with a tool is also the central concept underlying the pipelining capabilities of this application (cf. Figure 5): If output data of some tool A can (in principle) be used as input data for another tool B the names of all runs the user has already performed with tool A are offered for selection in the input form of tool B. For example, the target sequences found in a run with some search tool like NcbiBlast[12] are offered as input

[12] The most popular implementations of the **B**asic **L**ocal **A**lignment and **S**earch **T**ool [29] are "NcbiBlast", distributed by the National Center for Biotechnology Information (NCBI), Bethesda, MD, and "WU-Blast", provided by the University of Washington, Saint Louis, MO.

for a multiple sequence alignment tool like ClustalW[13]. Importantly, it is also possible to select more than one run for input (see Figure 7). In this case a new run of tool B is created for each of the selected inputs of tool A with only a single step of user interaction. The new runs get appropriately labeled and are executed (in parallel). Finished runs can be inspected and, if necessary, may also be rerun choosing different input parameters for each run individually.

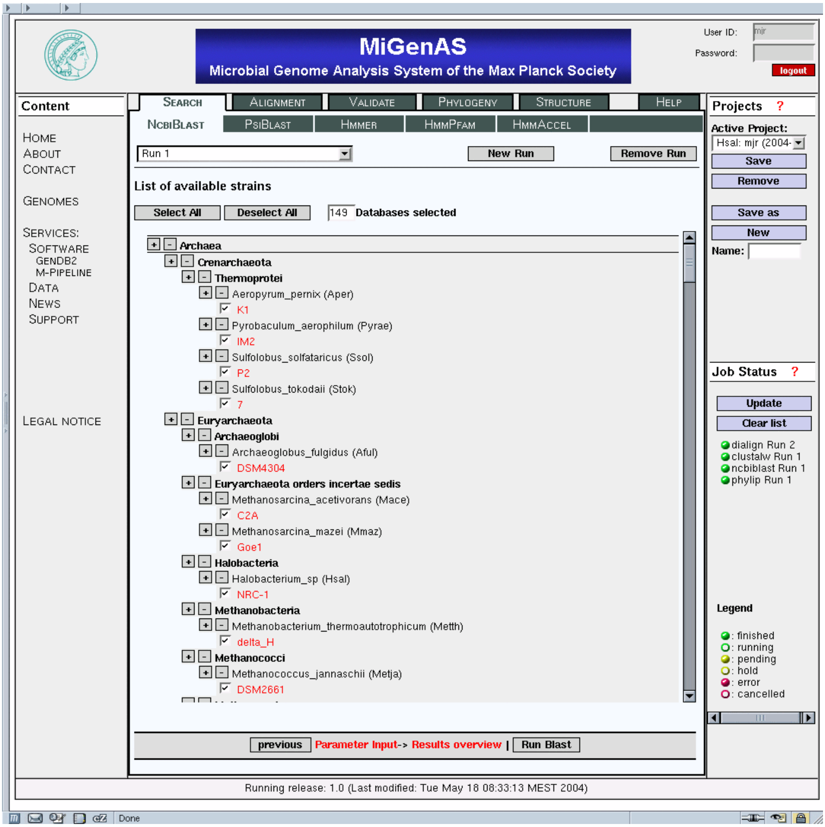
"Projects": Whereas the previously mentioned "run" may be envisaged as a level of persistency within an individual session a "project" can be used to make data permanent beyond that particular session with the web application (cf. Figure 5). The state of a project is saved by an explicit request of the user or automatically upon switching to a different project or logout from the system, thus allowing the user to interrupt a session at any time. Work on a project may then be resumed at a later time and also from a different client. Of course, the user needs to be warned if concurrent working on the same project is (accidentally) attempted. At login the user is by default equipped with an empty and transient "scratch" project in order to facilitate a quick startup and to ease casual or short-time usage. At any time this scratch project with all data that have accumulated so far can be turned into an actual, persistent project.

Defining projects also facilitates managing a wealth of individual runs and logically grouping the corresponding workflows and data which may accumulate in course of extensive use of the web application. While working on a specific project all information associated with other projects is hidden from the user's view.

3.2 Graphical User Interface

The graphical display is horizontally subdivided into three major parts (see, e.g., Figure 6). The vertical navigation bar on the left hand side of the window offers links to different areas of the MiGenAS website. Another vertical bar on the right hand side of the window provides facilities for managing the user's projects (upper part) and shows information about the status of individual computing tasks (lower part) of the active project. The controls for managing projects closely resemble the basic file/buffer-handling functionalities of common software tools like, e.g., text editors. A selection menu shows the name of the project which is currently in use and allows to switch to a different project. By clicking the corresponding buttons the user can save the current status of the project (optionally

[13] ClustalW [30] is a widespread tool for automatically aligning a set of nucleic-acid or amino-acid sequences.



User ID: mjr

Run 1

List of available strains

149 Databases selected

- ☑ Archaea
  - ☑ Crenarchaeota
    - ☑ Thermoprotei
      - ☑ Aeropyrum_pernix (Aper)
        - ☑ K1
      - ☑ Pyrobaculum_aerophilum (Pyrae)
        - ☑ IM2
      - ☑ Sulfolobus_solfataricus (Ssol)
        - ☑ P2
      - ☑ Sulfolobus_tokodaii (Stok)
        - ☑ 7
  - ☑ Euryarchaeota
    - ☑ Archaeoglobi
      - ☑ Archaeoglobus_fulgidus (Aful)
        - ☑ DSM4304
    - ☑ Euryarchaeota orders incertae sedis
      - ☑ Methanosarcina_acetivorans (Mace)
        - ☑ C2A
      - ☑ Methanosarcina_mazei (Mmaz)
        - ☑ Goe1
    - ☑ Halobacteria
      - ☑ Halobacterium_sp (Hsal)
        - ☑ NRC-1
    - ☑ Methanobacteria
      - ☑ Methanobacterium_thermoautotrophicum (Metth)
        - ☑ delta_H
    - ☑ Methanococci
      - ☑ Methanococcus_jannaschii (Metja)
        - ☑ DSM2661

Active Project: Hsal: mjr (2004-

● dialign Run 2
● clustalw Run 1
● ncbiblast Run 1
● phylip Run 1

Running release: 1.0 (Last modified: Tue May 18 08:33:13 MEST 2004)

Figure 6: Selection of microbial genome databases to be searched for homologous sequences using the NcbiBlast tool. Notice the option to conveniently select hierarchical groups of databases according to the taxonomical classification of the corresponding species (in analogy to the functionality offered by NCBI [6]).

providing a new project name), create a new, empty project or remove the currently active project.

The most important part of the user interaction occurs in the largest, central part of the window which displays the input and output of individual tools. The window also offers controls for navigating between input form and output display of an individual run, as well as for switching between runs or moving focus to a different tool (see below).

Applying the intuitive navigation concept of so-called "tabs" tools are grouped by the basic functionality they provide. Specifically, our system currently offers a selection of tools (cf. Figure 6) for sequence-similarity search ("Search"), the alignment of (multiple) sequences ("Alignment") and the assessment of the quality of an alignment (“Validate”), as well as phylogenetic tools dealing with evolutionary relationship (“Phylogeny”) and tools for predicting structural and functional properties of proteins (“Structure”). The tool, which is currently active (i.e. expecting input from the user or showing tool results), is presented to the user by its tab and the corresponding functionality group being highlighted. Alternative tools providing the same type of functionality visually stay in the background, as do the other functionality groups (e.g., Figure 7). The active tool can be selected by simply clicking the corresponding tabs. The run which is currently active is shown at the top of the input form (Figure 6) along with controls for switching to a different run (left selection menu), removing the run or manually creating a new one (buttons to the right of the selection menu). A navigation bar at the bottom contains the button for executing the tool and allows navigating between the different stages (input form, results display, etc.) of the corresponding run (see Figure 6).

Within the large central part of the form selection facilities for related input parameters are displayed in separate groups. Suitable default values are offered for all options (cf. Figure 7). Less frequently used groups of parameters are by default hidden. With a single click the user can open the corresponding fields in order to inspect and manipulate the “expert options”. In many cases there are logical dependencies between individual parameters. Given the wealth of different options and parameters we consider it to be important to make such dependencies as transparent as possible for the user in order to avoid inconsistent selections, unnecessary computations and possibly misleading results from the beginning. Having chosen a particular variant of the NcbiBlast tool that is suitable only for searching amino-acid databases, for example, only the matching subset of all databases is presented to the user for selection.

3.3 Application Example

The following scientific application[14] serves to highlight the capabilities of our system, with particular focus on the integrated toolkit approach and the advantages of the pipelining concept. As explaining the scientific objectives of the example is beyond the scope of this article we only give a brief outline of the problem, noting the basic workflows and the corresponding tools involved. After describing the steps necessary to accomplish the tasks with our web application this approach is confronted with addressing the problem by traditional techniques.

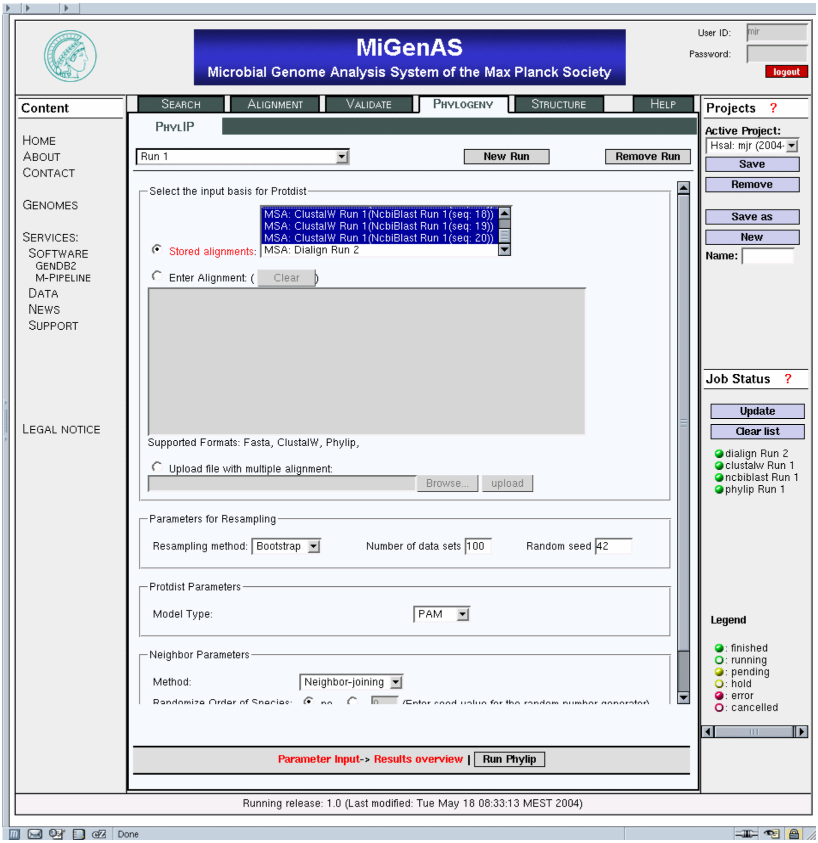
Outline: The application is one out of many embedded in an ongoing research project at MPI for Biochemistry (Dept. Oesterhelt) which is

[14] It is a pleasure to thank M. Falb and F. Pfeiffer (MPI for Biochemistry) for numerous hints and explanations that greatly helped to improve the functionality of the system.

dedicated to analyzing the genome and proteome of the halophilic microorganism Halobacterium salinarum (Hsal, [5]). Here, we are interested in the evolutionary relationship of a specific class of proteins identified in the genome of Hsal with similar, “homologous” sequences found in the genomes of other microbes.

Such relationships are usually represented by a so-called phylogenetic tree, which visualizes the “distance” (quantified by some mathematical measure that is specific to the chosen method) between sequences by the length of a corresponding branch of the tree. Branching points of the (binary) tree identify common ancestors. Before measuring the distance, however, the sequences of interest must first be identified and extracted from the genome databases of the corresponding organisms and then be properly aligned with each other. Automatically computing a biologically meaningful multiple sequence alignment in turn is a complex task in itself. It is central to many problems in bioinformatics and has attained considerable scientific interest resulting in many different algorithms and implementations (see, e.g., [31] for an accessible introduction). Our web application currently offers five of the most prominent multiple sequence alignment programs (cf. [33]).

Example session: As the first step in our example a homology search is performed for each of the 20 relevant amino-acid sequences of Hsal in the protein databases of all completely sequenced and publicly available microbial genomes (currently 149, plus several eucaryotic genomes to permit a more complete phylogenetic analysis). This task is accomplished by uploading an ASCII file with the 20 query sequences to the input form of the “NcbiBlast” tool, choosing appropriate search parameters, selecting the databases to be searched (see Figure 6) and submitting a run with the NcbiBlast tool. Depending on the search parameters and chosen cut-off values for dropping insignificant matches one gets on the order of 100 relevant hits for each query sequence. From the hits of the search the user then selects (manually, or automatically according to a criterion which is based on the statistical significance of the match) those sequences for which a multiple sequence alignment is to be constructed. For each of the 20 multiple alignments (one for each query sequence) computed by ClustalW [30] a phylogenetic tree needs to be constructed which translates the alignments into a (theoretical) evolutionary relationship between the organisms (or more precisely: between the amino-acid sequences involved in the alignment) and visualizes this relationship in form of a binary tree. Switching to the input form of the “Phylip” tool [34] the 20 alignments computed previously are conveniently selected for processing with a single step of user-interaction together with requesting a 100-fold randomly perturbed replication (“bootstrapping”) of each of the alignments for later statistical analysis (see Figure 7).



User ID: mjr
Password:

Active Project: Hsal: mjr (2004-
Name:

Run 1

Select the input basis for Protdist

● Stored alignments:
MSA: ClustalW Run 1(NcbiBlast Run 1(seq: 18))
MSA: ClustalW Run 1(NcbiBlast Run 1(seq: 19))
MSA: ClustalW Run 1(NcbiBlast Run 1(seq: 20))
MSA: Dialign Run 2

○ Enter Alignment: (Clear)

Supported Formats: Fasta, ClustalW, Phylip,

○ Upload file with multiple alignment:

Parameters for Resampling

Resampling method: Bootstrap   Number of data sets 100   Random seed 42

Protdist Parameters

Model Type: PAM

Neighbor Parameters

Method: Neighbor-joining

Randomize Order of Species: ● no ○ 0 (Enter seed value for the random number generator)

● dialign Run 2
● clustalw Run 1
● ncbiblast Run 1
● phylip Run 1

● : finished
○ : running
● : pending
○ : hold
● : error
○ : cancelled

Running release: 1.0 (Last modified: Tue May 18 08:33:13 MEST 2004)

Figure 7: Selection of input parameters for the Phylip tool. In this example 20 individual multiple sequence alignments ("MSA", computed by "ClustalW") are selected (marked by blue color in the list shown in the upper part of the form) for translation into phylogenetic trees.

The 20 phylogenetic trees constructed by Phylip can then be examined in detail (see Figure 8) and interpreted in the biological context. Depending on the final or intermediate results, the scientist likely decides to repeat one or more of the previously mentioned steps, using different input parameters or even an alternative tool. This reexamination or reprocessing of the data might well be done at a later time or at a different place after the session has been saved as a "project"[15]. Using our web application the example described above was finished in one session, which lasted for a few hours.

[15] A close inspection of Figure 7 reveals that the active "Run 1" is already finished and hence, in Figure 7, the fictitious user is actually revising the input parameters for this run.

Note, however, that only during a tiny fraction of this time the scientist directly needs to interact with the system. While most of the time is spent "in the background" for the actual execution of the underlying tools (and hence the quoted number is largely dependent on the available computing resources) the scientist is free to work on other tasks or projects within the same web application.

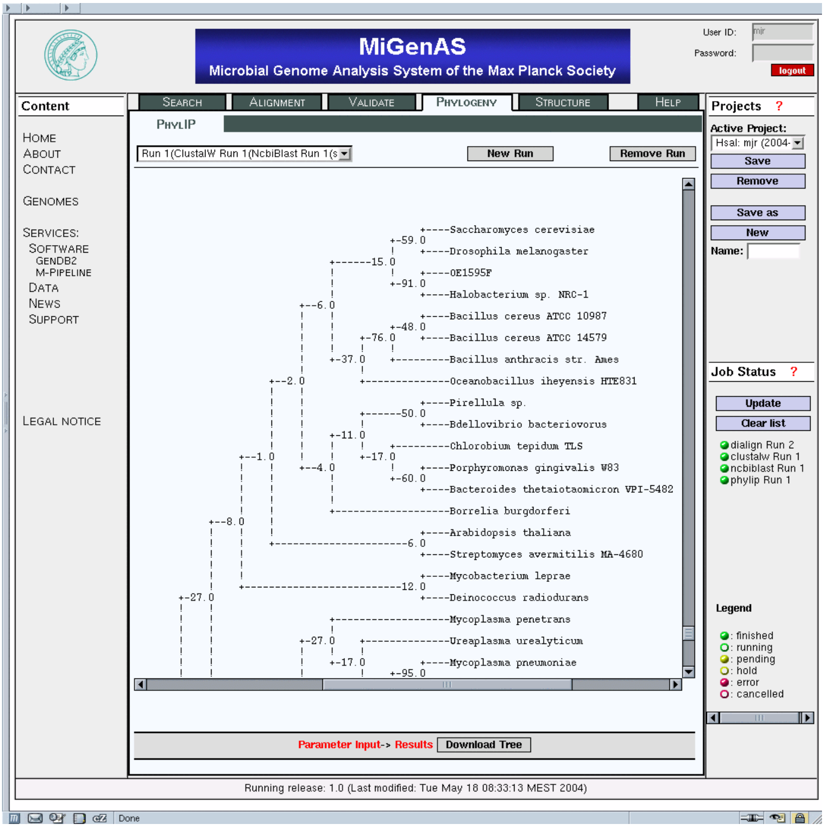
Discussion: The advantages of our system may be illustrated by confronting the described example session with a straightforward approach where the problem is tackled "by hand". The latter means that the executable programs of the corresponding software packages are called directly using their command-line interfaces; possibly aided by employing small scripts for managing the calls, for preparing input and for processing output. This still appears to be a common practice in the field.

For computing the multiple sequence alignments the output of the homology search must first be interpreted ("parsed") and filtered for the relevant information, which, depending on the employed tool, can be a cumbersome task prone to various kinds of errors. The individual target sequences then have to be extracted from the databases and assembled in a format appropriate for the particular multiple sequence alignment software. The user might also have to convert the computed alignment into a different data format which is accepted by the phylogenetic analysis tool. For finally constructing the phylogenetic trees input has to be prepared for executing four individual programs distributed with the "Phylip"-bundle: In a first step the input is "bootstrapped" by duplicating the input alignments (see above). For each of the duplicates distances and phylogenetic trees are computed in two subsequent steps and finally a single, so-called "consensus" tree is calculated. Note that in our example this procedure has to be applied to each of the 20 sequences of Hsal.

It is obvious that the presence of technical hurdles of this kind can severely hamper the productivity of the scientist by drawing the focus from his or her original field of expertise to programming tasks which he or she might not be adequately trained for. We speculate that in the worst case decisive questions might not even be asked due to seemingly insurmountable technical prerequisites even in cases of comparably handy complexity, like the one described.

In other words, a flexible, powerful and easy-to-use software solution might motivate the scientist to tackle new and more complex problems, help in interpreting and further processing the results and hence is expected to substantially facilitate scientific progress.

Finally, it should be mentioned that sophisticated special-purpose software packages exist which, for the particular problem described above may offer some, all, or more functionalities than our web application currently supports (e.g. ARB, [32]). The same probably holds true also for a number of other tasks. Nevertheless, we reemphasize the strengths of the



```
                                        +----Saccharomyces cerevisiae
                                   +-59.0
                                   !    +----Drosophila melanogaster
                         +------15.0
                         !         !    +----OE1595F
                         !         +-91.0
                         !              +----Halobacterium sp. NRC-1
                    +--6.0
                    !    !              +----Bacillus cereus ATCC 10987
                    !    !         +-48.0
                    !    !    +-76.0    +----Bacillus cereus ATCC 14579
                    !    !    !    !
                    !    +-37.0    +---------Bacillus anthracis str. Ames
                    !         !
               +--2.0         +--------------Oceanobacillus iheyensis HTE831
               !    !
               !    !                   +----Pirellula sp.
               !    !         +------50.0
               !    !         !         +----Bdellovibrio bacteriovorus
               !    !    +-11.0
               !    !    !    !    +---------Chlorobium tepidum TLS
          +--1.0    !    !    +-17.0
          !    !    +--4.0         !    +----Porphyromonas gingivalis W83
          !    !         !         +-60.0
          !    !         !              +----Bacteroides thetaiotaomicron VPI-5482
          !    !         !
          !    !         +-------------------Borrelia burgdorferi
     +--8.0    !
     !    !    !                        +----Arabidopsis thaliana
     !    !    +----------------------6.0
     !    !                             +----Streptomyces avermitilis MA-4680
     !    !
     !    !                             +----Mycobacterium leprae
     !    +--------------------------12.0
+-27.0                                  +----Deinococcus radiodurans
!    !
!    !                   +-------------------Mycoplasma penetrans
!    !                   !
!    !              +-27.0    +--------------Ureaplasma urealyticum
!    !              !    !    !
!    !              !    +-17.0         +----Mycoplasma pneumoniae
!    !              !         !    +-95.0
```

Running release: 1.0 (Last modified: Tue May 18 08:33:13 MEST 2004)

Figure 8: The results window of the Phylip tool shows the phylogenetic tree computed for one of the 20 multiple sequence alignments supplied as input.

integrated toolkit approach we have taken for our system. In the described application, for example, the following scenario is conceivable: During analysis of a particular phylogenetic tree the scientist decides to evaluate the biological relevance of the underlying multiple sequence alignment by recomputing it with an alternative tool, by invoking special validation programs or even by taking into account the predicted structure and biological function of the involved proteins. In particular the latter two use-cases cannot be covered in depth and with the necessary flexibility by special-purpose software designed for phylogenetic analysis. The user would thus be forced to leave such a system at a certain point and move to a different software package which entails many of the disadvantages pointed out in the course of this article. Moreover, if a particular special-purpose software package is considered to be useful by the involved scientists it may well be coupled with our system in an appropriate way.

4 Conclusion and Outlook

We have created and implemented a framework for a workflow engine that allows the integration of various kinds of bioinformatics software tools into a single system. Currently, a web application tailored to the analysis of microbial genomes is provided giving the user a convenient access to tools and databases over the internet[16]. The application offers a number of unique functionalities for microbial genome research. Most notably, the user can process complex tool-workflows with large samples of input data (up to the scale of a microbial genome) with a minimum of technical effort and short response cycles. We have demonstrated the benefits for the researcher and pointed out the relevance for a large class of bioinformatics applications by confronting an actual scientific application of our system with a traditional approach to solve the same problem.

The architecture of this framework employs a highly modular, object-oriented design. This enhances the maintainability of such a system and gives the necessary flexibility for continually adapting and extending the functionality of the framework in response to growing and changing demands of the scientists working with it. The chosen J2EE-based implementation, in addition, guarantees seamless portability across different platforms[17].

Compliance with well-established software standards and taking into account major world-wide technological developments allow us to immediately embark on new promising trends like, e.g., the upcoming computing grids. By this means we are able to substantially extend scope and functionality of the provided services in the most flexible and efficient way. For the user this will open completely new views onto the basic functionalities and services offered by our framework so far. Our current efforts in this direction aim at a so-called Service Oriented Architecture (SOA). The user will be able to access the workflow engine by a standardized set of interfaces and protocols which are discoverable over the internet by directory services such as UDDI. Even in the current proprietary form, web services can be used by clients to tap the already existing individual workflow sections (“exits”) at arbitrary points and plug in independent developments. This will allow users to efficiently handle a

[16] http://www.migenas.mpg.de/ –> Services –> M.-Pipeline.

[17] In fact, the system has already been moved from a SOLARIS platform to a LINUX cluster without any difficulties.

class of even more complex problems, which require non-interactive, automatic processing and filtering of large datasets. For performing such kinds of tasks web services are generally considered to be an ideal technology to replace a plethora of highly non-standard, home-grown scripts individual scientists have written for such purposes.

Web-services technology could also be a starting point for integration of earlier stages of genomic analysis like automatic annotation (as, e.g., featured by the powerful Gendb software package [35] developed at the University of Bielefeld). Moreover, the general framework architecture is not limited to the class of applications discussed so far. The system can as well serve as the basis for a more general workflow engine, which, according to the establishing grid-computing community, is a central challenge of "the Grid".

References

[1] The MiGenAS Project Portal: http://www.migenas.mpg.de/, http://www.migenas.org/.

[2] The Global Grid Forum: http://www.ggf.org/, https://forge.gridforum.org/.

[3] Business Process Expression Language: http://ifr.sap.com/bpel4ws/.

[4] Web Services Resource Framework: http://www.ibm.com/developerworks/library/ws-resource/.

[5] HaloLex: The Information System for Halobacterium Salinarum, http://www.halolex.mpg.de/.

[6] NCBI's "Genomic Blast", http://www.ncbi.nlm.nih.gov/sutils/genom_table.cgi.

[7] Sun Java Runtime Engine, version 1.4.2.04, http://java.sun.com/.

[8] IBM Java Runtime Engine, version 1.4.1, http://java.ibm.com/.

[9] Sun J2EE, http://java.sun.com/j2ee.

[10] CORBA: http://www.omg.org/corba.

[11] JBoss EJB Container, versions 3.2.3, 4.0.DR3, http://www.jboss.org/.

[12] JacORB, versions, http://jacorb.fu-berlin.de.

[13] Jakarta Tomcat, versions 4.1.28, 5.0.0, http://jakarta.apache.org/tomcat/.

[14] Apache http server, http://httpd.apache.org/.

[15] JavaServer Pages Technology, http://java.sun.com/products/jsp/.

[16] Apache Axis, http://ws.apache.org/axis/.

[17] OmniORB, version 4.0.3, http://look.it.up/omniorb.

[18] Eclipse, http://www.eclipse.org/.

[19] Rational Rose, http://www.rational.com/.

[20] Oinn, T. et al., "Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows", to appear in Bioinformatics Journal (2004); http://taverna.sourceforge.net/.

[21] Fields, D.K. and Kolb, M.A. "Web Development with Java Server Pages", Manning (2000).

[22] Gamma E., et al., "Design Patterns: Elements of Reusable Object-Oriented Software", Addison Wesley (1995).

[23] Crass, T., et al., “The Helmholtz Network of Bioinformatics: an integrative web portal for bioinformatics resources”, Bioinformatics 20(2):268-270 (2004); http://www.hnbioinfo.de/.

[24] Matena, V., et al.; “Applying Enterprise JavaBeans, 2nd Edition”, Addison Wesley (2003).

[25] Marinescu, F.; “EJB Design Patterns”, Wiley (2002).

[26] Booch, G., “Object Solutions”, Addison Wesley (1996).

[27] Roman, E., “Mastering Enterprise Java Beans”, Addison Wesley (2002).

[28] “Third Annual Java Use and Awareness Study”, BZ Research (2003).

[29] Altschul, A. et al.: "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs”, Nucleic Acids Res. 25:3389-3402 (1997).

[30] Thompson, J.D., et al., “CLUSTALW: improving the sensitivity of progressive multiple sequence alignments through sequence weighting, position-specific gap penalties and weight matrix choice”, Nucleic Acids Res. 22:4673-4680 (1994).

[31] Quelette, B.F.F.: “Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins”, Wiley (2001).

[32] Ludwig, W. et al., “ARB: a software environment for sequence data”, Nucleic Acids Research 32(4):1363-1371 (2004).

[33] Lassmann, T., Sonnhammer, E.L.L., “Quality assessment of multiple alignment programs”, FEBS Lett., 529(1):126-130 (2002).

[34] Felsenstein, J., “PHYLIP - Phylogeny Inference Package (Version 3.2)”, Cladistics 5: 164-166 (1989). Note: we currently use Version 3.5, distributed by the author (2003).

[35] Meyer, F. et al., “GenDB—an open source genome annotation system for prokaryote genomes”, Nucleic Acids Res. 31:2187-2195 (2003)

Nominiert für den Heinz-Billing-Preis 2004

Dynamos in Action

Johannes Wicht
Julien Aubert
Max-Planck-Institut für Sonnensystemforschung

Abstract

The magnetic fields of planets and suns are produced by induction in their electrically conducting interiors. Recent computer simulations have proven that this so called dynamo process can explain many properties of the observed magnetic fields. But as the numerical models become increasingly complex the tools to analyze their solutions must be refined accordingly. We have developed a set of new tools that allows us to calculate and visualize the key properties of the dynamo process. The simulations are performed with a modified MAGIC dynamo code. Visualizations employ Matlab and an IDL software package that features a graphical user interface. These instruments have enabled us to understand and present dynamo mechanism at a new level of comprehension and clarity. We demonstrate their power by illustrating en detail the $\alpha^2\omega$-mechanism that drives the simple benchmark dynamo. We also explain how these tools help to unravel the dynamics of a magnetic field reversal.

1 Introduction

The modelling of magnetic field generation in planetary cores and in the interior of stars has greatly benefited from the advances in computer technology. The increase in shear numerical power allows to conduct simulations at parameter ranges that could not be covered before. In addition, more and more features are incorporated into the models to make them more realistic. Recent simulations of the geodynamo, for example, not only show Earth-like large scale field structures, many smaller spatial details are also comparable to geomagnetic features. Moreover, the temporal behavior on different time scales – secular variation (some ten to thousands of years), paleo-secular variation (thousands to ten thousands of years), and polarity sequences (several thousands to millons of years) – are remarkably similar to geomagnetic behavior for some of the models [2, 7, 10, 15, 11, 16].

However, as the complexity of the models advances our understanding of the interior dynamics can not keep up. Dynamo modellers mostly analyze their simulations in terms of axisymmetric fields and global measures like the total magnetic energy. In addition, surface fields are compared with what we know, for example, about the geomagnetic field. But a detailed analyses of the internal dynamics is hardly possible. The particular convective flow features, their role in magnetic field production, the back reaction of the magnetic field on the flow via Lorentz-forces, all these important and interesting questions are treated in a global averaged sense, if at all. We are able to run advanced dynamo models on more and more powerful computers, but do not really understand their dynamics and have no idea why some dynamos deliver more realistic solutions than others.

The reasons are simple: First, dynamos are complex. Flow field and magnetic field vary in space and time and their interaction is complicated. And second, visualizing time dependent 3D vector fields in such a way that the important features are highlighted and can be conceived is no simple matter.

We have set out to tackle these problems and describe our advances here. In terms of model complexity, it seemed wise to take a step back. Our models are not todays most advanced, their parameters are much less realistic than in other recent simulations [3]. This has the advantage that the interior fields are not too small scale and that the dynamics of the dynamo process can still be untangle. However, the resulting magnetic fields presented here are nevertheless very much Earth-like.

An important part was the development of new visualization tools and respective modifications in our numerical model MAGIC [15]. The visualization tools are based on Matlab and IDL and allow to plot snap shots and to produce animations of 3D isosurfaces and 2D cuts. We not only analyze the magnetic field and the fluid flow but also compute and display magnetic field production, advection, diffusion, and the magnetic fieldlines. This allows us to identify and separate the key processes which leads to a much clearer understanding of the dynamo mechanism.

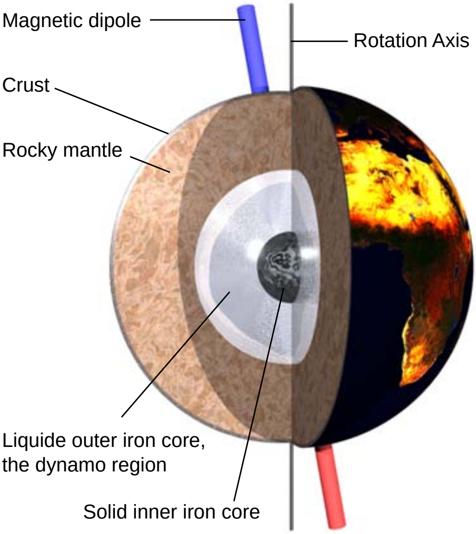


Fig. 1: Cut through the Earth showing the liquide iron outer core where the magnetic field is produced in the dynamo process.

2 The Dynamo Mechanism

We model convection and magnetic field production in a rotating spherical shell. The shell may represent the liquid metallic cores of Earth-like planets as well as the conduction atmospheres of gas planets. For simplicity we mostly refer to Earth's dynamo region shown in figure 1. Convection is driven by buoyancy differences of compositional and/or thermal origin. A Navier-Stokes equation describes the changes in fluid flow. A diffusion equation that also contains buoyancy sources and sinks formulates changes in the buoyancy field. These equations and more details on the employed numerical code MAGIC can be found in [15].

Here, we concentrate on magnetic field dynamics that is described by the dynamo equation which can be derived from the Maxwell equations in their non-relativistic form:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \lambda \nabla^2 \mathbf{B} \tag{1}$$

The first term on the right hand side models magnetic induction, the change of mangetic field $\mathbf{B}$ due to motion $\mathbf{u}$ of the conducting fluid. It is therefore called the induction term, field production term, or simply the dynamo term.

Note that $\mathbf{B}$ should more appropriately be called magnetic induction, but the geomagnetic community simply droppes the word induction for simplicity. The second term on the right hand side is a diffusion term, where $\lambda = 1/(\sigma\mu)$ is the magnetic diffusivity with σ being the electrical conductivity and μ the magnetic permeability. Since the liquid iron in the Earth's outer has a high conductivity (about 5×10^5 S/m) the magnetic diffusivity is quite low. In order to make our dynamos work we have to decrease it further by several orders of magnitude. i.e. we increase the electrical conductivity of the working fluid beyond appropriate values. Our models are not realistic in this and, for that matter, in several other respects but nevertheless produce very convincing magnetic fields. See [3] for a discussion on this topic.

The dynamo term can be decomposed into a field production term and a field advection term:

$$\nabla \times (\mathbf{u} \times \mathbf{B}) = (\mathbf{B} \cdot \nabla)\,\mathbf{u} \;-\; (\mathbf{u} \cdot \nabla)\,\mathbf{B} \tag{2}$$

Here, we have used the fact that magnetic field and flow field are divergence free, i.e. $\nabla \cdot \mathbf{B} = \nabla \cdot \mathbf{u} = 0$ since we assume an incompressible fluid.

Magnetic fieldlines are a good way to visualize magnetic field direction and strength. They also help to picture the action of flow on the magnetic field. The production term in equation (2) describes the stretching of magnetic fieldlines due to gradients in the fluid flow. This is the key mechanism that converts kinetic energy into magnetic energy. The advection term describes how magnetic fieldlines are pushed around by the fluid flow. This term has no net effect on the global magnetic energy. Magnetic field is moved away from where it is produced and is distributed over the fluid shell. It also reacts back on the fluid flow via the Lorentz force in the Navier-Stokes equation which is not discussed here. This back reaction provides an important balance and limits magnetic field growth.

Since $\mathbf{B}$ is divergence free, it can be decomposed uniquely in a toroidal and a poloidal contribution:

$$\mathbf{B} = \nabla \times \hat{r}t \;+\; \nabla \times \nabla \times \hat{r}p \tag{3}$$

where $\hat{r}$ is the unit vector in radial direction. The toroidal field $\nabla \times \hat{r}t$ has no radial component and is therefore confined to the dynamo region. The poloidal field $\nabla \times \nabla \times \hat{r}p$, on the other hand, leaves the conducting region of the fluid shell, and this is the field component we can observe on Earth's surface. Interaction of these two field contributions and the fluid flow plays a decisive role in the dynamo process. The dynamo can only work if fluid motion stretches and twists each of the two field contribution so that the respective other contribution is generated. If, for example, poloidal field is produced by flow acting on toroidal field the production could only work permanently if the toroidal field does not decay due to ohmic losses. Thus toroidal field also has to be produced by the dynamo process.

The ansatz (3) has the advantage that the three unknown components of the magnetic vector field are reduced to the two unknown scalar potentials p and t. In our model, the dynamo equation is decomposed into poloidal and toroidal evolution equations, and both are coupled by the dynamo term. We also calculate and visualize production, advection, and diffusion for both field contributions separately to understand their interplay.

The way magnetic field generation is described goes back to early mean field dynamo theory. (See for example [8] for an overview and further references.) Neglecting the non-axisymmetric small scale field seems tempting at first, since the magnetic field of many planets in dominated by the large scale axisymmetric dipole field. However, Cowling's [5] has shown that purely axisymmetric fields can not be maintained by a dynamo mechanism. Mean field dynamo theory is mainly concerned with the production of axisymmetric field and the necessary contributions due to non axisymmetric flows, is incorporated as a mean effect into the theory. The azimuthal mean is only one of several possible ways to define a mean in the scope of mean field dynamos. Important is the separation of large scale mean components and the smaller scale contributions. The expressions large scale and axisymmetric as well as small scale and non-axisymmetric are used synonymous here.

Mean field theory defines two main mechanisms for producing axisymmetric fields: The ω-effect produces azimuthal axisymmetric toroidal field from axisymmetric poloidal field. North-south oriented poloidal fieldlines are stretched by shear in axisymmetric zonal flows. Strong zonal flows are likely to be present in the fluid core or atmosphere of any rapidly rotating planet. They are driven by thermal gradients (thermal winds) and advection of momentum (Reynolds stress). The α-effect produces poloidal field from toroidal field. Toroidal fieldlines are twisted by flow vorticity pointing along the rotation axis. This action produces radial poloidal field of both signs. In addition, differential advection along the vorticity axis is needed to separate both radial field directions. The result is a net large scale radial poloidal field of opposite signs in the northern and southern hemisphere, the well known dipole configuration. The combination of vorticity and transport along the vorticity direction is called helicity:

$$\mathbf{H} = \mathbf{U} \cdot (\nabla \times \mathbf{U}) \tag{4}$$

Vorticity, and more to the point helicity, are essential for dynamo action. All the mechanisms will be illustrated below for one of our dynamo models.

Dynamos based on these two basic mechanisms are called $\alpha\omega$-dynamos. Large scale toroidal magnetic field can also be created by the action of small scale flow on small scale poloidal field. Such a mechanism has commonly been used for mean field dynamos. [14] suggested that it also works in self-consistent convection driven models. Consequently, there are also $\alpha^2\omega$ dynamos or simply α^2 dynamos when the ω-effect plays no important role. The mechanisms will become clearer when illustrated below for one of our dynamo models.

3 Fieldlines

As mentioned above, fieldlines are a useful way for visualizing magnetic fields. They provide information about the field direction. In addition, their density is a measure for field strength, and their curvature tells something about field gradients. Moreover, twists in fieldlines have to be maintained by field production or diffuse away otherwise. They therefore indicate where certain field components are created.

Since a plot of several fieldlines can quickly look like a bowl of spaghetti, a few key lines have to be selected. To not lose the information on the field strength we weight their thickness with the magnetic pressure $\vec{B}^2$. This representation has several advantages. First, it allows to judge which lines are energetically important. Second, it gives an idea of the Lorentz-forces acting perpendicular to the line and thus gives information about the back-reaction on the flow field.

The Lorentz-force can be interpreted as the sum of a magnetic tension and a magnetic pressure gradient:

$$\left(\nabla \times \vec{B}\right) \times \vec{B} = \left[\left(\vec{B} \cdot \nabla\right) \vec{B} - \nabla \left(\frac{\vec{B}^2}{2}\right)\right]. \tag{5}$$

We switch to a local curvi-linear coordinate system that is spanned by the tangential unit vector $\vec{e_s}$ along the fieldline, the normal unit vector $\vec{e_n}$ in the local fieldline plane, and the binormal vector $\vec{e_b} = \vec{e_s} \times \vec{e_n}$. Magnetic tension is then given by:

$$\left(\vec{B} \cdot \nabla\right) \vec{B} = \frac{\vec{B}^2}{R_c}\vec{e_n} + \frac{\partial}{\partial s}\left(\frac{\vec{B}^2}{2}\right)\vec{e_s}. \tag{6}$$

Here, R_c is the local curvature radius and s is the coordinate along $\vec{e_s}$. Plugging this into the Lorentz-force expression (5) results in:

$$\left(\nabla \times \vec{B}\right) \times \vec{B} = \left[\frac{\vec{B}^2}{R_c}\vec{e_n} - \nabla_H \left(\frac{\vec{B}^2}{2}\right)\right]. \tag{7}$$

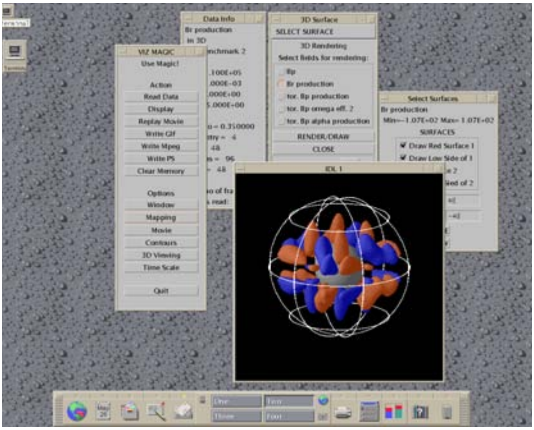


Fig. 2: Screen shot of the IDL based visualization tool.

with $\nabla_H = \nabla - \vec{e_s}\partial/\partial s$. The Lorentz-force is thus strong where magnetic pressure and fieldline curvature are large. Moreover, magnetic pressure variations between adjacent lines carry information on the Lorentz-force. These features have been used extensively by [1].

4 Visualization Tool

We employ Matlab and IDL to visualize the various fields calculated with MAGIC. Matlab's advantage is its relative simplicity and a very effective rendering that is essential for 3D viewing. IDL, the Interactive Data Language developed by RSI of Boulder, Colorado, is a quite extensive software package, that allows to program very professional applications. However, the learning curve, while shallow at first, steepens quickly. A large advantage of IDL is the effective data management. Large amounts of data can be handled and modified in reasonable times which is important when producing animations of 3D views.

We have developed a toolbox with a graphical user interface (GUI) based on IDL's widget system. The GUI allows to control several display options, to select output options, and to determine the viewing surface. Figure (2) shows a screen shot of the GUI in action. Output can be stored as GIF, JPEG, postscript, or animated GIF.

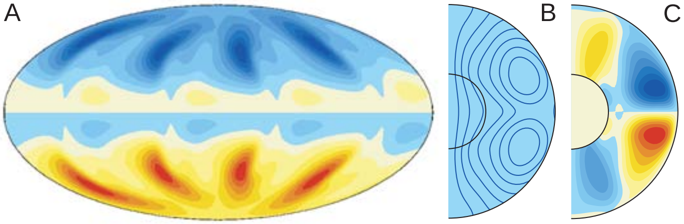


Fig. 3: Radial magnetic field at the outer boundary (A), axisymmetric fieldlines (B) and axisymmetric azimuthal field (C)

5 A Simple Dynamo

To demonstrate the power of the new visualizations tools, we have selected a rather simple dynamo, that has served as a benchmark for numerical dynamo simulations (See benchmark II in [4]). The parameters are chosen so that convection and dynamo action are close to onset: viscosity and electrical conductivity are large and the system is heated only mildly. The solution is therefore quasi-stable, i.e. convection and magnetic field pattern are drifting in prograde direction but are stationary otherwise. Flow and magnetic field are of large scale and have a four-fold symmetry in longitude.

Figure (3) shows the radial magnetic field at the outer surface (A), the axisymmetric fieldlines (B) and the axisymmetric toroidal field (C). The radial field at the outer boundary is similar to the geomagnetic field at the core-mantle boundary in that it is dipole dominated. However, the four-fold symmetry is not dominant in the geomagnetic field.

Figure (4) shows selected magnetic fieldlines and isosurfaces of the negative z-vorticity in blue. The fieldlines pass though the strong field patches at the boundaries and give a good representation of the internal structure. The blue anti-cyclones can be identified with the typical columnar convection roles in rapidly rotating spherical shells. In the pure convective case the cyclones would be equally important. They are, however, suppressed by Lorentz-forces and have therefore not been include in figure (4).

The reason for the selection of anti-cyclones has been revealed by [1] with the help of the visualization tools described here. The strong poloidal fieldlines sitting in the center of the anti-cyclones are aligned with the cyclonic axis. Responsible for the alignment is a secondary flow directed away from the equator toward the northern and southern end of the vorticity columns. This flow plays an important role in the poloidal field production [14]. Ax-

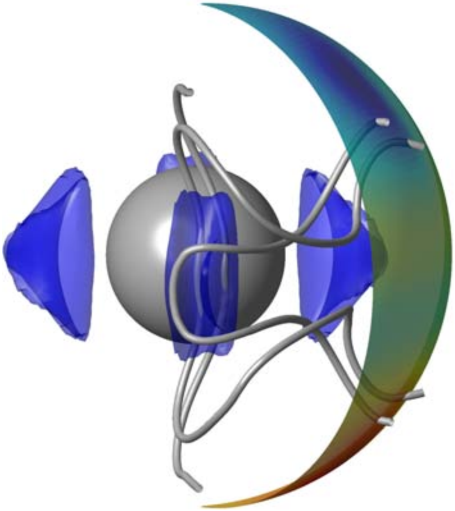


Fig. 4: Anti-cyclones, shown as blue isosurfaces, dominate the convective motion in the dynamo model. They can be identified with convection roles. The three magnetic fieldlines have been depicted to best illustrate the main features of the dynamo proce ss. These lines pass through the strong field patches at the outer surface indicated at right corresponding to Earth's core-mantle boundary. A model for Earth's inner core is shown in the center.

ially aligned field minimizes the flow disruption due to Lorentz-forces. The picture is different for the cyclones, where the internal secondary flow is directed toward the equator. It collects magnetic field near the outer boundary, thus producing the high amplitude patches shown in figure (3) and stretches the fieldlines down the columnar axis. In addition, the field is also expelled from the cyclone resulting in strong fieldlines that cross the vorticity structure. The associated Lorentz-force brakes the vortex motion of cyclones and is responsible for the dominance of anti-cyclones in the system.

We will now illustrate the seven specific mechanisms that shape the three model fieldlines depicted in figure (4). Since these fieldlines are chosen to represent the main field components, this ensemble of mechanisms actually constitutes the dynamo process. The order 1) to 7) follows the distortion of the fieldlines from the line at the right front to the line in the left background. This view of the dynamo process is based on analyzing production and advection of the radial poloidal field and the azimuthal toroidal field. We show isosurfaces of these quantities along with the fieldlines in several illustrations below. Note that these are no artist views but the true solutions of the numerical simulation.

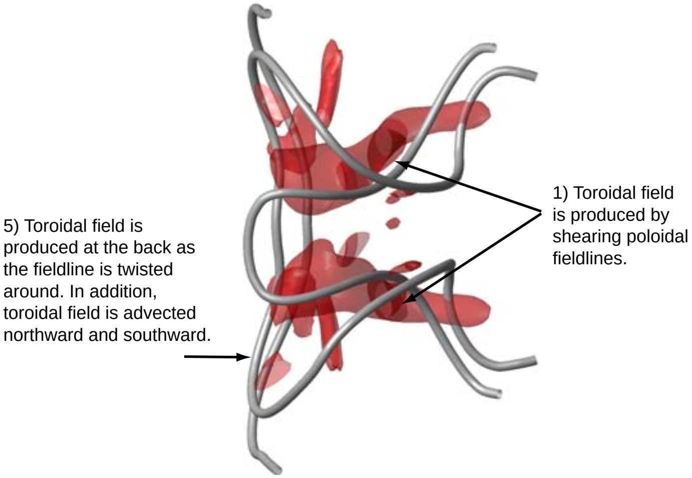

5) Toroidal field is produced at the back as the fieldline is twisted around. In addition, toroidal field is advected northward and southward.
1) Toroidal field is produced by shearing poloidal fieldlines.

Fig. 5: Production of azimuthal toroidal field shown as isosurfaces in additi on to the model fieldlines.

1)The process starts with a poloidal fieldline roughly aligned with the rotation axis. The field is pointing in north-south direction. This line is grabbed by the anti-cyclone and stretched in azimuthal direction, thus producing retrograde toroidal field. Figure (5) shows the production of azimuthal toroidal field along with the fieldlines.

2)Azimuthal toroidal field is then advected toward the equatorial plane by flow in the vicinity of the cyclone that lyes to the right of the anti-cyclone. Figure (6) shows an isosurface of azimuthal toroidal field advection demonstrating this point.

3)The now mostly azimuthal toroidal fieldline is twisted around the anticyclon producing radial field of opposite sign on either side. Figure (7) shows the production of radial poloidal field, and the described action is represented by the isosurface padding the anti-cyclone.

4)In addition, poloidal field is produced by stretching along the axis of the anti-cyclone due to the poleward motions and the associated horizontal gradients. See figure (7).

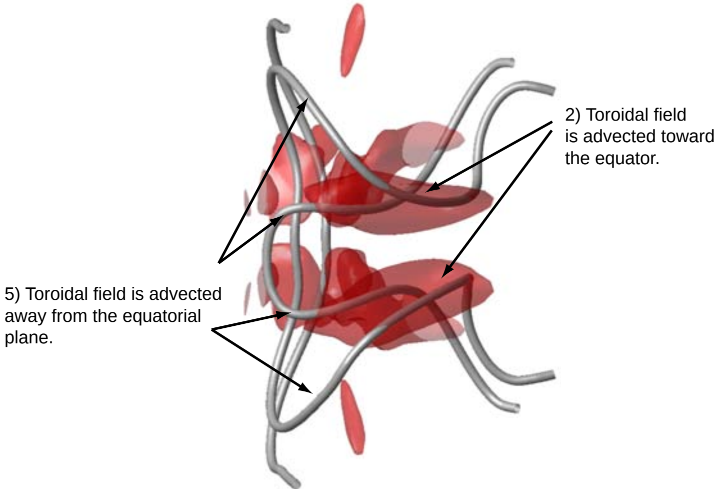


Fig. 6: Advection of azimuthal toroidal field as isosurfaces along with the mo del fieldlines.

5)The poleward motion in the vicinity of the anti-cyclones also advects toroidal field away from the equatorial plane. This process can be identified at the respective location in figure (6).

6)As the fieldline is twisted further around the anti-cyclone toroidal field is also produced at the back of this structure, see figure (5). Toroidal field production at this location could also have another source, the stretching of poloidal fieldlines lying at the back of the anti-cyclone in an mechanism analogous to point 1).

7)A third effect of the poleward motion is the advection of radial poloidal field along the anti-cyclone axis, shown in figure (8). Advection is present in nearly all the places where poloidal field is produced. An example is the radially outward motion between the anti-cyclon and the cyclone to its right.

The mechanism ends like it started, with a poloidal fieldline aligned with the rotation axis that is ready to be grabbed by the next anti-cyclone. A closed cycle is an important requirement for dynamos, showing that poloidal and toroidal field production and advection work hand in hand. Figure (9) gives an overview of the seven mechanisms.

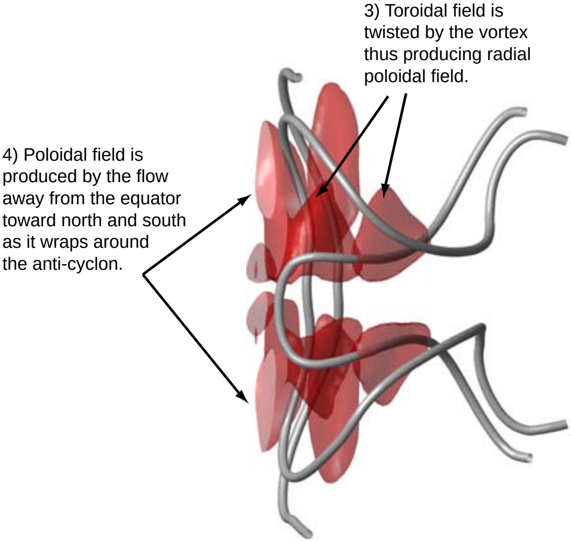

3) Toroidal field is twisted by the vortex thus producing radial poloidal field.
4) Poloidal field is produced by the flow away from the equator toward north and south as it wraps around the anti-cyclon.

Fig. 7: Production of radial poloidal field (isosurface) shaping three model fieldlines.

Such a process has been envisioned by [14]. The analyses presented here finally confirms their interpretation. Our points 1), 2), 5), and 6) form the toroidal field dynamo process that correspond to the mechanism shown in their figure 5a. Points 3), 4), and 7) constitute the poloidal counterpart, that has been illustrated by figure 5b in [14].

The mechanism explained above would classify as an α mechanism simply because the smaller scale flow of the individual cyclones and anti-cyclones is the active medium, not a large scale global (axisymmetric) flow. However, we have already mentioned that anti-cyclones are significantly larger than cyclones in our model. This gives rise to a mean flow, retrograde at the outer side of the anti-cyclones and prograde at their inner side. This flow could potentially produce toroidal field via an ω-effect. To explore the contribution of mean zonal flows to the dynamo process we have computed toroidal field production due to this specific flow contribution. Subtracting this part from the total toroidal field production leaves what would be called an α-effect.

Figure (10) shows the contribution to axisymmetric toroidal field changes due to the different mechanisms. Panel A shows the axisymmetric azimuthal toroidal field and panel B shows the respective dynamo effect, i.e. the sum of production and advection. Panel C depicts the field production due to the

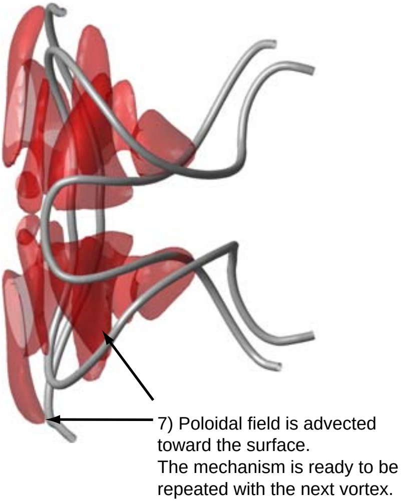

7) Poloidal field is advected
toward the surface.
The mechanism is ready to be
repeated with the next vortex.

Fig. 8: Advection of radial poloidal field shown as an isosurface along with the model fieldlines.

ω-effect and panel D the production due to the α-effect. Both are of comparable magnitude and cancel to a good degree. Finally, panel E shows the sum of the α-effect and advection of axisymmetric azimuthal toroidal field. This may be called the α-dynamo effect, note that there is no advection of axisymmetric field by axisymmetric azimuthal flow. The advection significantly enhances the α-production term, and the sum clearly dominates the total dynamo effect. This can have tow reasons. First, axisymmetric field produced by the ω-effect may be advected in a way that enhances the α-mechanism. And second, the alternating small scale flow of the cyclones and anti-cyclones separates azimuthal field of opposite sign, thereby leading to an increased azimuthal mean. By closely analyzing the advection we could show that this latter mechanism is clearly dominating. In particular the flow up and down the cyclonic plays an important role.

The toroidal field dynamo mechanism therefore is clearly dominated by the α-effect. But since the ω-effect is also relatively strong, this dynamo should probably be classified as an $\alpha^2\omega$ dynamo. However, these classifications are rather crude, go back to simplistic mean field dynamos, and lose their usefulness when the actual process has been analyzed en detail. It may be time for a new dynamo terminology.

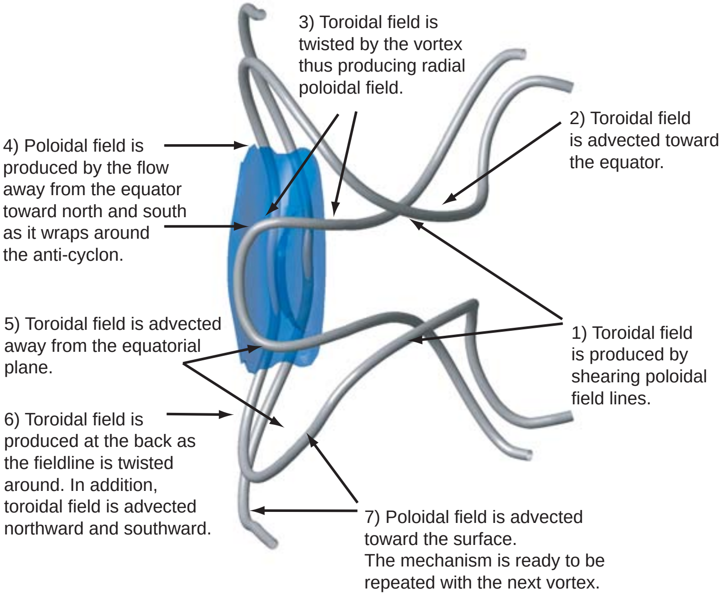

3) Toroidal field is twisted by the vortex thus producing radial poloidal field.
2) Toroidal field is advected toward the equator.
4) Poloidal field is produced by the flow away from the equator toward north and south as it wraps around the anti-cyclon.
1) Toroidal field is produced by shearing poloidal field lines.
5) Toroidal field is advected away from the equatorial plane.
6) Toroidal field is produced at the back as the fieldline is twisted around. In addition, toroidal field is advected northward and southward.
7) Poloidal field is advected toward the surface.
The mechanism is ready to be repeated with the next vortex.

Fig. 9: Overview of the seven important steps in the dynamo process. They are demonstrated with the help of the three fieldlines that stretch around an anti-cyclone. The cycle starts at lower right and ends with process 7) at the bottom.

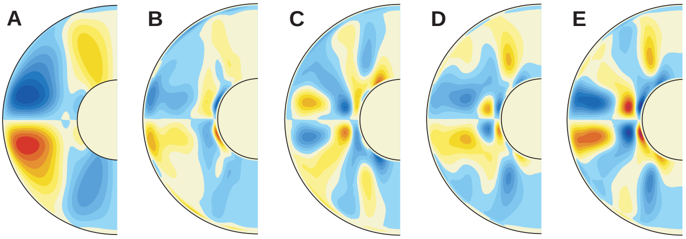


Fig. 10: Axisymmetric azimuthal toroidal field (A), production plus advection (dynamo or induction term) of azimuthal toroidal field (B), azimuthal toroidal field produced by the ω-effect (C), by the α-effect (D), and the sum of α-effect and advection of azimuthal toroidal field (E).

6 A Reversing Dynamo

Field reversals are the most striking features in the geomagnetic history. Several hundred reversals have been documented by paleomagnetism reaching back more than 250 Myr into Earth's history [12, 6]. Besides faithfully replicating many other features of the Earth's magnetic field several dynamo simulations also undergo reversals [7, 11, 15, 9]. However, our understanding of the reversal dynamics is still patchy. Because paleomagnetic data give only little information on the interior fields dynamo simulations are the only way to learn more about these interesting phenomenons. But the interpretation of the interior reversal dynamics is complicated by the very reasons outlined above: The dynamics is complex, flow and magnetic fields are 3D, and they vary in space and time. Visualization of the evolved processes under such circumstances is a hard task, let alone their understanding.

[16] have used the visualization tools described here to analyze a rather simple nearly periodically reversing dynamo. They succeeded to untangle the different steps in the reversal process, thus delivering the first concise and complete description of a magnetic polarity change in a selfconsistent dynamo simulation. Several animations of the different quantities involved show the interior dynamics. They can be viewed and downloaded at [16] or at http://www.linmpi.mpg.de/∼wicht/Reversals/.

The main steps in the reversal process are outlined in figure (11), showing the changes in the poloidal field polarity during one reversal. Despite the fact that the model in rather simple, many of its features may nevertheless apply to geomagnetic reversals. Important findings are:

1)The helicity associated with plume like upwellings inside and at the tangent cylinder are essential for creating inverse field. Such features are thought to also exist inside the tangent cylinders of the Earth, which suggest that reversals could start in these regions.

2)Upwelling in the plumes transport the inverse field to the core mantle boundary (CMB). When the inverse patches reach the core surface the reversal process becomes 'visible' for an outside observer. This observer would see only a smaller section of the total process.

3)The inverse field is advected along the core-mantle boundary until the reversal is completed, i.e. normal polarity field has been replaced. This suggest that the fluid velocities close to the core-mantle boundary determine the apparent duration of the reversal. Using typical values for flows along the CMB (westward drift) arrives at an estimated duration of a few thousand years for the Earth. This agrees well with paleomagnetic results.

4)The dynamo is close to a kinematic solution. This suggests that the velocity field is not responsible for triggering the reversal. Oscillatory fields are a primary solution of the dynamo problem. However, since the velocity field

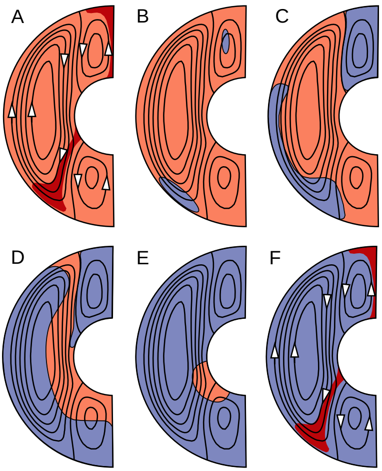


Fig. 11: Dynamics of a magnetic field reversal. Red and blue stand for normal and reversed polarity of the poloidal magnetic field. Contour lines are streamlines of the axisymmetric meridional circulation, which distributes the inverse field along the core-mantle boundary and throughout the core. The inverse field is produced in rising plumes marked with darker colors in the first and last panel.

is not stationary in the simulation, the interpretation is complicated.

Figure (12) shows the fieldlines at the start of a reversal. We have used the magnetic pressure to scale the fieldline thickness. Also shown is the axisymmetric azimuthal flow. The strong flow gradients near the equator are responsible for an intense ω-effect in this region. The fieldlines also demonstrate the strong magnetic field production by the rising plumes. One of these plumes is pictured in the southern part at lower left, a second rises near the inner-core north pole. Inverse field is mostly created by the twisting action of the plumes. Since meridional motion along the outer and inner boundary is also important this dynamo belongs to the class of $\alpha\omega$m-dynamos, m stands for meridional [13].

Calculating and displaying magnetic field production and advection was crucial for understanding the above described processes. In the case of the simple stationary dynamo presented in section (5) the field production and the field itself are in good correspondence, i.e. have the same sign. This, however, can not be true for a reversing dynamo. Here, magnetic field opposing the already existing field direction is produced. And just to complicate things, this field is not produced where the first inverse field patches appear at the CMB. As mentioned above, advection plays an important role for the reversal process.

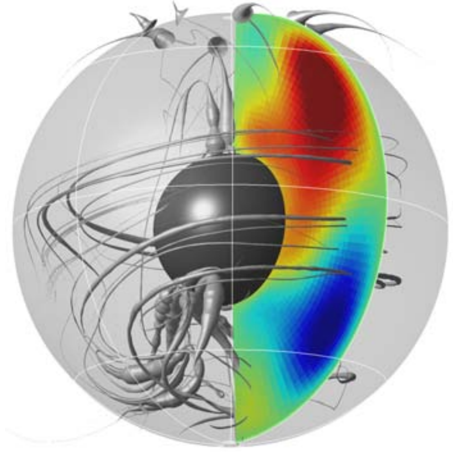


Fig. 12: Magnetic fieldlines and axisymmetric azimuthal flow for a reversing dynamo. The fieldlines have been scaled with the magnetic pressure. The coloured disk visualizes the gradients in the zonal flow that are responsible for the strong ω-effect in this dynamo. The flow directions are colour coded, red (blue) indicating prograde (retrograde).

7 Conclusion

We have demonstrated that visualization and careful analyses of dynamo action can greatly advance our understanding of the involved mechanisms. We present the first concise and clear illustration of an $\alpha^2\omega$-dynamo process based on the actual solution data from a selfconsistent dynamo simulation. And for that matter, this is the first illustration of this quality of any dynamo type. In addition, the newly developed tools enabled us to understand the dynamics of a magnetic field reversal.

The dynamo models are relatively simple but not overly so. Selecting viewing surfaces, isosurfaces, viewing angles, lighting, colors, shading, and fieldlines that best showed the most important mechanisms was neverthelesss difficult. It remains to be proven, whether the visualization tools are equally useful, when analyzing more complex dynamos at more Earth-like parameters.

But since even some of the basic dynamo features have not been understood thoroughly, much can be learned from analyzing even simpler models. We also hope that our illustrations can help in teaching dynamos theory to students and other scientists.

Acknowledgements

This work has been supported by the DFG via the SPP Geomagnetische Variationen. We grateful thank the GWDG for supplying very dependable and reliable computer resources.

References

[1] J. Aubert and J. Wicht. Axial versus equatorial dynamo models with implications for planetary ma gnetic fields. *Earth and Planet. Sci. Lett.*, 221:409–419, 2004.

[2] U. Christensen, P. Olson, and G.A. Glatzmaier. Numerical modeling of the geodynamo: A systematic parameter study. *Geophys. J. Int.*, 138:393–409, 1999.

[3] U. Christensen and A. Tilgner. Power requirement of the geodynamo from ohmic losses in numerical and laboratory dynamos. *Nature*, 429:169–171, 2004.

[4] U. R. Christensen, J. Aubert, F. H. Busse, E. Cardin, P. a nd Dormy, S. Gibbons, G. A. Glatzmaier, Y. Honkura, C. A. Jones, M. Kono, M. Matsushi ma, A. Sakuraba, F. Takahashi, A. Tilgner, J. Wicht, and K. Zhang. A numerical dynamo benchmark. *Phys. Earth Planet. Inter.*, 128:25–34, 2001.

[5] T.G. Cowling. The magnetic field of sunspots. *Month. Not. R. Astr. Soc.*, 94:39–48, 1934.

[6] E. Dormy, J.-P. Valet, and V. Courtillot. Numerical models of the geodynamo and observational constraint. *Geochem. Geophys. Geosys.*, 1:Paper No 2000GC0000062, 2000.

[7] G.A. Glatzmaier, R.S. Coe, L. Hongre, and P.H. Roberts. The role of the Earth's mantle in controlling the frequency of geomagnetic reversals. *Nature*, 401:885–890, 1999.

[8] R. Hollerbach. On the theory of the geodynamo. *Phys. Earth Planet. Inter.*, 98(3-4):163–185, 1996.

[9] S. Kida, K. Araki, and H. Kitauchi. Periodic reversals of magnetic field generated by thermal convection in a rotating spherical shell. *J. Phys. Soc. Jap.*, 66:2194–2201, 1997.

[10] C. Kutzner and U.R. Christensen. From stable dipolar to reversing numerical dynamos. *Phys. Earth Planet. Inter.*, 131:29–45, 2002.

[11] C. Kutzner and U.R. Christensen. Simulated geomagnetic reversals and preferred vgp paths. *Geophys. J. Int.*, 157:169–171, 2004.

[12] R.T. Merrill and P.L. McFadden. Geomagnetic polarity transitions. *Rev. Geophys.*, 37:201–226, 1999.

[13] H.K. Moffat. *Magnetic field generation in electrically conducting fluids*. Cambridge University Press, Cambridge, 1978.

[14] P. Olson, U. Christensen, and G.A. Glatzmaier. Numerical modeling of the geodynamo: Mechanism of field generation and equilibration. *J. Geophys. Res.*, 104:10,383–10,404, 1999.

[15] J. Wicht. Inner-core conductivity in numerical dynamo simulations. *Phys. Earth Planet. Inter.*, 132:281–302, 2002.

[16] J. Wicht and P. Olson. A detailed study of the polarity reversal mechanism in a numerical dynamo model. *Geochem., Geophys., Geosyst.*, 5(3):doi:10.1029/2003GC000602, 2003.

Computational Tools for the Analysis and Simulation of HIV Drug Resistance

Niko Beerenwinkel
Max-Planck-Institut für Informatik, Saarbrücken

Abstract

Despite the approval of almost 20 antiretroviral drugs and the use of combination therapy, successful treatment of HIV-infections is hampered by the emergence of drug-resistant genetic variants in response to therapy. Finding a new potent drug combination after treatment failure is considered challenging, because most accumulated mutations confer resistance to multiple drugs. We present three computational tools for the analysis and simulation of viral genomic sequences, phenotypic drug resistance, and clinical outcomes. `Mtreemix` is a software package for estimating and using mixture models of trees that describe probabilistically the evolution of drug resistance. `Geno2pheno` is a web-based system for the prediction of phenotypic resistance from viral genotypes. It also implements normalization methods that make these predictions comparable between different drugs. Finally, `theo` predicts virological response within a patient from the infecting viral strain and the selected drug combination. Together these models and programs provide a quantitative picture of the evolution of drug resistance and support the design of individualized antiviral therapies.

1 Introduction

Genomic technologies have an increasingly strong impact on molecular biology, pharmacology, and medicine. High-throughput genotyping and molecular profiling techniques can speed up the drug discovery process and have the potential to improve health care. In order to make use of the large amounts of data produced by these technologies, efficient computational and statistical tools are necessary. Bioinformatics methods have traditionally been developed to support pharmaceutical research from the genome to new drugs (Lengauer 2002). However, it has been argued that patients may benefit from these computational advances much earlier than it actually takes until the first "genomic" drugs come to market (Sander 2000).

Personalized medicine

Improved diagnostics of particular disease subtypes and of genetic predispositions provides a basis for personalized medicine. Computational prognostics will play an important role in surmounting the *one-drug-for-all* paradigm. The key to individualized therapies is

1. to integrate genotypic, molecular profiling, and clinical data,
2. to develop predictive methods for the interpretation of genotypes and molecular profiles with respect to clinical outcomes, and
3. to provide these tools in the form of decision support systems to health care professionals.

The present work is concerned with the development and implementation of computational tools to support the design of individualized therapies against infections with human immunodeficiency virus (HIV). Originally, pharmacogenomics studies the effect of genetic variations between patients on susceptibility to disease and response to drugs (Altman & Klein 2002). Here we consider genetic variation in the viral pathogen, rather than in the host.

Our focus in this paper will be on aspects (2) and (3), method development and implementation. We have also tackled the data integration challenge (topic (1)), which led to the development of the `Arevir database` (Beerenwinkel 2004). This relational database provides a secure electronic platform for collaborative research between clinicians, virologists and bioinformaticians and is the logical and technical basis for data analysis.

HIV drug resistance

HIV infection is linked to the acquired immunodeficiency syndrome (AIDS), which represents a major world-wide health hazard. In the developed countries, 18 antiretroviral agents, that interfere with different steps in the viral replication cycle, are at the physician's disposal. The two most important drug targets are the viral enzymes reverse transcriptase (RT) and protease (PR). Currently 10 RT inhibitors and 7 PR inhibitors are available. The first entry inhibitor has only recently been approved. Despite this comfortable number of available drugs and the use of combination therapies, eradication of the virus is not possible. Even prolonged suppression of virus replication below detectable limits is achieved only in few patients.

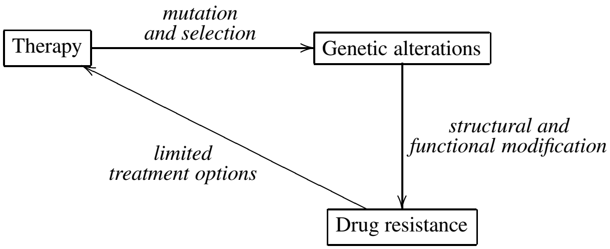
A major reason for limited therapeutic success is the evolution of drug-resistant genetic variants in response to therapy (Perrin & Telenti 1998, Vandamme et al. 1999). The intra-patient viral population is a highly dynamic system, characterized by short replication cycles and high turnover rates. Since viral replication is highly error-prone, these dynamics can quickly generate resistant variants, that have a selective advantage under drug pressure. These escape mutants cause viral rebound and lead to treatment failure.

Identifying a new potent drug combination after therapy failure is challenging, because many accumulated mutations confer resistance not only to the used drugs, but also to structurally similar compounds from the same drug class. The high levels of cross-resistance considerably restrict remaining treatment options after failure of a regimen.

In order to understand the development of drug resistance and to translate this knowledge into improved therapeutic strategies we need to answer the following questions.

1. How does the selective pressure of drug therapies shape the viral population?
2. What is the effect of these genetic alterations on phenotypic drug resistance?
3. Given these phenotypic changes for an individual patient and the limited treatment options they imply, what is the best choice for a new drug combination?

These key aspects are summarized in the following diagram:


Therapy
mutation and selection
Genetic alterations
structural and functional modification
limited treatment options
Drug resistance

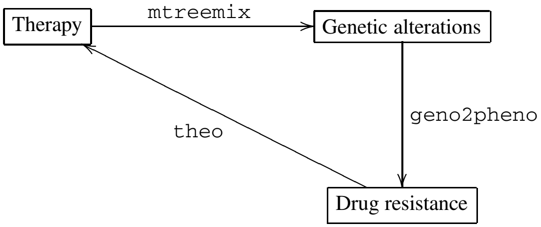
Resistance testing

Resistance testing has become an important diagnostic tool in the management of HIV infections. Resistance assays are either based on analyzing the viral genome to identify resistance-associated mutations (genotypic resistance testing) or on direct *in vitro* measures of drug susceptibility of the virus (phenotypic resistance testing) (Walter et al. 1999, Alcorn & Faruki 2000, Oette et al. 2004). Since, at least in principle, the viral genotype contains the resistance information, optimal combination therapies could be designed on the basis of a genotypic resistance test. However, retrieving the relevant information from the viral DNA sequence is challenging because of the high genetic variability of HIV strains.

Here we present computational tools for the interpretation of HIV genotypes to predict phenotypic drug resistance and to support the design of potent combination therapies. In general, the analysis of genotype-phenotype relations is of central interest in molecular biology, and gives rise to several mathematical and computational challenges (Karp 2002).

Outline

After summarizing related work in the following Section 2 we describe three computer programs that provide quantitative answers to questions 1–3 above:



In Section 3 we introduce `mtreemix`, a software package for estimating mutagenetic trees mixture models from observed viral sequences. These probabilistic models have been designed to describe the accumulation of resistance-associated mutations along multiple evolutionary pathways in the viral genome.

Section 4 presents `geno2pheno`, a web-based system for the prediction of phenotypic drug resistance from genotypes. We briefly describe the underlying statistical models and discuss the clinical significance of the tool.

Section 5 deals with `theo`, a tool to support the design of optimal drug combinations. We present a scoring function for combination therapies based on the phenotype predictions derived in Section 4. We show that our model is predictive of virological response in a real-world clinical setting.

Finally, in Section 7 we discuss conclusions and future work.

2 Related Work

For each of the three tools, we briefly discuss alternative approaches and related work.

Mtreemix

Related to our tree based approach to modeling mutational patterns are Bayesian tree models for approximating multivariate discrete probability distributions (Chow & Liu 1968). These models can be learned in a maximum likelihood (ML) fashion. Our mutagenetic trees mixture models are similar in spirit to the work of Meilă & Jordan (2000), who generalize these models to mixtures of trees. However, these unrestricted models do not capture the direction of dependencies.

Tree models and more general graph models related to mutagenetic trees have been developed for oncogenesis, where chromosomal losses and gains are considered as events (Radmacher et al. 2001). Desper et al. (2000) make use of distance-based phylogeny methods for tree reconstruction, which also miss directed dependencies, while von Heydebreck et al. (2004) have developed a ML approach at the cost of introducing hidden variables.

Foulkes & DeGruttola (2004) characterize the progression of viral mutations over time by a Markov model whose states are a few sequence clusters rather than the set of all mutational patterns.

Geno2pheno

Several clinical and virologic research groups have set up scoring systems for relating sequence variations to drug resistance or likelihood of therapy failure (see Ravela et al. (2003) and Stürmer et al. (2003) for an overview and comparison). Predictions are based on expert rules derived from public literature and are publically available (http://hivdb.stanford.edu/). These systems predict resistance to a drug as one of 2 to 5 categories ranging from “susceptible” to “resistant”.

The VirtualPhenotype$^{\text{TM}}$ (Virco, Mechelen, Belgium) is a quantitative phenotype prediction based on a pattern search in a database of genotype-phenotype pairs (Larder et al. 2000). The regression problem has also been approached with artificial neural networks (Drăghici & Potter 2003). Wang & Larder (2003) use a 1-hidden-layer neural network to predict resistance to lopinavir from PR sequences.

Sevin et al. (2000) have analyzed genotype-phenotype data by applying two classification techniques, namely recursive partitioning and linear discriminant analysis, and k-means clustering as an unsupervised learning method on the set of genotypes.

Theo

One way to estimate the activity of a therapeutic regimen against a viral strain is to learn this effect from observational clinical databases. However, this approach is limited by the huge amount of data necessary to derive useful models (DiRienzo & DeGruttola 2002).

Alternatively, drug combinations are scored on the basis of single drug effects. There are two rules-based systems following this principle. The CTSHIV (Customized Treatment Strategy for HIV) system identifies optimal resistance-avoiding combination therapies by means of a branch-and-bound algorithm operating on a set of rules (Lathrop & Pazzani 1999). Another approach applies fuzzy logic methods to a set of expert rules (De Luca et al. 2002). Rule weights are learned from known clinical outcomes.

Since rules-based approaches depend on the quality of established knowledge, they usually improve with the number of published studies. However, they tend to show inferior performance for newer drugs or infrequent drug combinations.

3 `Mtreemix`: Learning mutational pathways

The `mtreemix` package provides software to describe the evolution of a viral strain under the selective pressure of drug therapy.

Evolution of drug resistance

The evolution of drug resistance is characterized by the accumulation of resistance-associated mutations in the viral genes coding for the drug targets. There is evidence that these mutations do not occur independently. Some mutations are known to cluster together (Beerenwinkel et al. 2001b, Gonzales et al. 2003, Wu et al. 2003), but the order of accumulation is usually unknown. Only a few experimental studies based on longitudinal (time series) data have revealed directed dependencies between mutations (Boucher et al. 1992, Molla et al. 1996). However, this type of analysis is not practical for many different drugs or even drug combinations, because large longitudinal samples from patients under the same therapeutic regimen are difficult to obtain. As an alternative, `mtreemix` implements a method for estimating mutational pathways from cross-sectional data (i.e. data from different patients at different time points), which are much more abundant.

Mutagenetic trees

We describe the evolution of drug resistance as the accumulation of permanent genetic changes (Beerenwinkel, Rahnenführer, Däumer, Hoffmann, Kaiser, Selbig & Lengauer 2005). An observed mutational pattern is regarded as the result of a subset of mutation events. The basic building block of our model is a mutagenetic tree (Figure 1). Vertices of these trees represent binary random variables, each indicating the occurrence of a mutation. Edge weights represent conditional probabilities between events with the constraint that an event can not occur whenever its parent event has not occurred.

These restricted Bayesian tree models can be efficiently reconstructed from cross-sectional data by solving an instance of the maximum weight branching problem with a combinatorial algorithm (Karp 1971, Desper et al. 1999). The weight functional involves only the pairwise probabilities between events, which can be estimated reliably from only moderately large data sets.

The likelihood of a mutational pattern in a mutagenetic tree is given as the product of edge weights over all edges necessary to generate the pattern times the probability of not acquiring any further mutation along the tree. This quantity can be computed efficiently by traversing the tree in a breadth first search.

Mixture models

The tree models provide a detailed and interpretable description of the process of accumulating permanent genetic changes. They represent a considerable improvement over independence or linear path models. Nevertheless, most mutagenetic trees can generate only a few mutational patterns as compared to the set of all possible patterns. Thus, the tree structure fits only certain subgroups of the data. We interpret this shortcoming as indicating that the data has been generated by more than one (tree-like) process. Therefore, we consider the broader class of mixture models of mutagenetic trees (Figure 1). The idea is to identify multiple evolutionary processes acting on the same gene (or genome), each process in one specialized component of the mix-

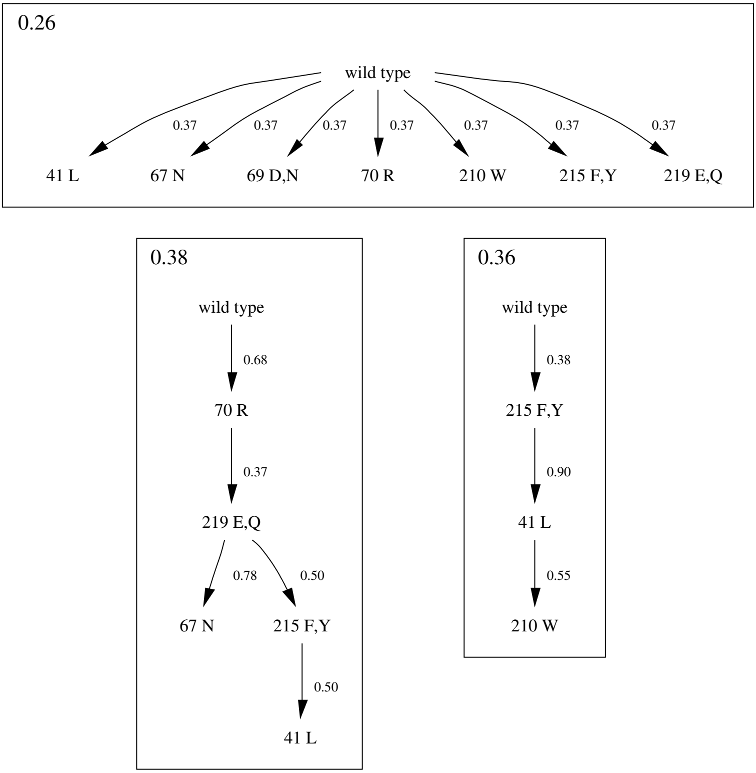


Fig. 1: Mixture model of 3 mutagenetic trees for the development of zidovudine resistance. Vertices are labeled with amino acid replacements in the HIV RT. Each tree is displayed in a box with its weight in the upper left corner. The upmost tree is a star and represents the noise component.

ture model. In particular, we introduce a "noise component", in which mutations are assumed to occur independently. Thus, the noise component has a star topology and all samples have positive likelihood in it. Figure 1 shows a mutagenetic trees mixture model for the development of resistance to zidovudine, the first anti-HIV drug approved.

A mixture model of K trees can be learned from data by iteratively computing the responsibilities of the model components for the observations and estimating, for each model component, its structure and parameters. This EM-like algorithm aims at identifying the mixture model that maximizes the likelihood of the data. Model selection (choosing the optimal number K of trees) is also performed in a maximum likelihood fashion and based on cross-validation estimates of the likelihood function. In general, the mixture models provide both better density fits of observed mutational patterns and biologically more plausible models than the single tree models (Beerenwinkel, Rahnenführer, Däumer, Hoffmann, Kaiser, Selbig & Lengauer 2005).

Program	Function
`Bootstrap`	Analyze tree stability by counting the appearances of each edge in multiple bootstrap replicates of the data. The responsibilities for the optimal model fit are fixed and used for resampling.
`Compare`	Compare density fits to empirical distribution between independence model, single tree model and mixture model. Four different distance measures between probability distributions are implemented: cosine distance, L_1 distance, L_2 distance, and Kullback-Leibler distance.
`Fit`	Estimate K-mutagenetic trees mixture model from data by running the EM-like algorithm. Input is a matrix of mutational patterns, output is the fitted mixture model.
`LLRtest`	Perform a likelihood ratio test of the fitted model against the independence model. The distribution of the test statistic is obtained from permutations of the data.
`Loglike`	Compute the log-likelihood of mutational patterns in a given mixture model.
`Print`	Print mixture model to DOT file, which allows for easy visualization with the Graphviz package (Gansner & North 1999).
`Select`	Model selection: estimate the out-of-sample likelihood for $1, \ldots, K_{\max}$ model components by cross-validation.
`Sim`	Simulate data, i.e. draw samples from a mixture model.
`Time`	Estimate waiting times for mutational patterns.
`Transprob`	Compute transition probabilities between mutational patterns in a mixture model.
`Wait`	Simulate mutational patterns and their waiting times.

Tab. 1: Programs of the `mtreemix` *package.*

Implementation

The `mtreemix` software package is a collection of C programs for statistical inference with mutagenetic trees mixture models (Beerenwinkel, Rahnenführer, Kaiser, Hoffmann, Selbig & Lengauer 2005). Table 1 summarizes the functionality of the programs. `Mtreemix` is available at http://mtreemix.mpi-sb.mpg.de. We provide source code as well as precompiled Solaris and Linux binaries.

4 Geno2pheno: Estimating phenotypic drug resistance from genotypes

Geno2pheno is a web-based system for the prediction of phenotypic drug resistance from genotypes (Beerenwinkel, Däumer, Oette, Korn, Hoffmann, Kaiser, Lengauer, Selbig & Walter 2003). It is based on a set of matched genotype-phenotype pairs obtained from genotypic and phenotypic resistance testing of the same clinical samples derived from patients failing antiretroviral combination therapy. In genotypic testing, direct DNA sequencing produces genomic data of about 1200 base pairs of the HIV *pol* gene, which codes for PR and RT, while phenotypic test results are usually reported as resistance factors (RFs), defined as the fold-change in susceptibility to the drug relative to a susceptible reference virus. It has been shown that patients can benefit from both genotypic and phenotypic testing (DeGruttola et al. 2000), but genotyping is faster and cheaper, whereas phenotypic results are easier to interpret.

Support Vector Machines

We apply machine learning techniques to the genotype-phenotype data in order to identify the genetic determinants of phenotypic resistance and to derive predictive models (Beerenwinkel et al. 2002). In particular, Support Vector Machines (SVMs) are used to predict resistance factors from PR and RT sequences (Beerenwinkel et al. 2001a). For each drug, a linear SVM regression model is generated from matched genotype-RF pairs. We use an ϵ-insensitive loss function with ϵ fixed at 0.1 such that prediction errors of less than 0.1 $\log_{10}$-resistance factors are not penalized in the training phase. The ability of these models to generalize from the training data is assessed by 10-fold cross-validation and is reported as the mean squared error and as the squared correlation coefficient between predicted and observed $\log_{10}$ resistance factors (Table 2). Since the range of observed resistance factors differs substantially among drugs, only the latter measure of performance allows for comparisons between drugs. Estimated squared correlation coefficients vary between 0.3 and 0.79 with an average of 0.6 (± 0.14) indicating that the models account for 30 to 79% of phenotypic variance.

The dynamic range of resistance factors varies by as much as two orders of magnitude between different drugs. Thus, resistance factors and their predictions are not comparable between different antiviral agents. To overcome this limitation geno2pheno provides two normalized resistance scores derived from different sets of predicted phenotypes (Beerenwinkel, Däumer, Oette, Korn, Hoffmann, Kaiser, Lengauer, Selbig & Walter 2003).

Comparison to therapy-naïve patients

In order to quantify natural variation of predicted resistance factors among patients that have not received any antiretroviral medication before, we predict phenotypes from a set of genotypes derived from untreated patients. For all drugs, these predicted resistance factors follow a normal distribution in agreement with results for experimentally determined phenotypes in drug-naïve patients (Harrigan et al. 2001). We observe considerable differences in both the mean and the standard deviation between drugs. In Figure 2 four representative examples are displayed. Having estimated these distributions, we can report for each predicted phenotype how many standard deviations it is away from the mean among drug-naïve patients. This *z-score* provides a standardized and comparable measure of deviation from the expected value for the untreated subpopulation.

The two-state model

We can gain more information on the meaning of a predicted resistance factor by studying the distribution of predictions over all genotypes. For this purpose, we consider a set of 2000 sequences randomly drawn from a clinical database in order to estimate the unconditional probability den-

Drug	N	SV	MSE (SE)	r^2
Zidovudine	649	387	0.554 (0.040)	0.62
Didanosine	649	437	0.101 (0.009)	0.42
Zalcitabine	534	325	0.122 (0.013)	0.30
Stavudine	649	401	0.145 (0.015)	0.33
Lamivudine	648	408	0.332 (0.019)	0.72
Abacavir	637	405	0.075 (0.011)	0.60
Tenofovir	321	206	0.091 (0.005)	0.50
Nevirapine	649	418	0.638 (0.056)	0.55
Delavirdine	648	403	0.476 (0.033)	0.55
Efavirenz	634	437	0.354 (0.026)	0.60
Saquinavir	652	394	0.204 (0.022)	0.71
Indinavir	652	387	0.197 (0.017)	0.73
Ritonavir	652	383	0.176 (0.017)	0.79
Nelfinavir	651	391	0.207 (0.011)	0.71
Amprenavir	464	303	0.173 (0.013)	0.65
Lopinavir	307	210	0.169 (0.016)	0.73
Atazanavir	305	187	0.262 (0.034)	0.61

Tab. 2: SVM regression analysis. Predictive performance is estimated from N samples by 10-fold cross-validation, and is reported as the mean squared error (MSE), its standard error (SE) and the squared correlation coefficient (r^2) between predicted and observed $\log_{10}$-resistance factors. SV denotes the number of support vectors, i.e. the number of non-zero coefficients in the linear SVM model.

sity of predicted phenotypes found in clinical isolates. Analysis of phenotype predictions shows large differences in range, location and deviation of modes, but also reveals the bimodal nature of the distributions common to all drugs (Figure 3). Thus, we model the probability density of predicted phenotypes by a two-component Gaussian mixture model. The parameters of this model are estimated by applying the EM algorithm (Dempster et al. 1977). Figure 3 displays the fitted density curves for the four drugs from Figure 2. The observed bimodality of the distributions supports the assumption of a "two-state model" (susceptible versus resistant) for the viral population. Given the generative two-component mixture model we can compute the probability of a sequence to originate from the resistant subpopulation. We call this quantity the *probability score*. In Figure 3 the estimated cumulative density of membership in the resistant subpopulation is plotted as a function of the predicted phenotype. The probability score provides a normalized and comparable measure of resistance for all antiretroviral drugs. Unlike z-scores with respect to drug-naïve patients, probability scores exploit information on location and variance of both the susceptible and the resistant subpopulation.

Web server

`Geno2pheno` (http://www.genafor.org) implements these approaches to interpret a genotypic resistance test result. A number of Perl scripts tie together software for computing a sequence alignment and making a SVM prediction, compute the resistance scores and present the results in HTML pages. On submitting a DNA sequence coding for HIV PR and RT to the server one obtains

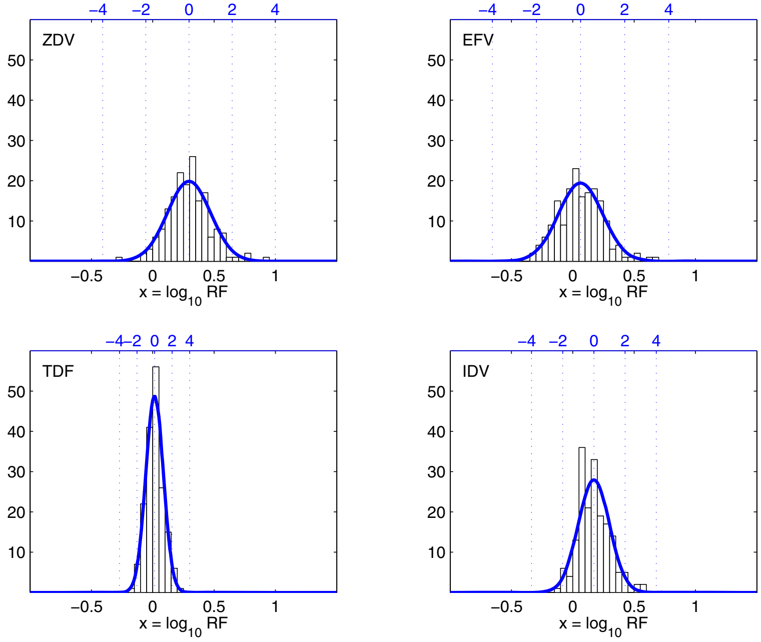


Fig. 2: Z-scores for zidovudine (ZDV), efavirenz (EFV), tenofovir (TDF), and indinavir (IDV). Histogram data and fitted normal densities are displayed for predicted resistance factors from viral genotypes derived from 124 treatment-naïve patients. The bottom x-axes refer to $\log_{10}$ *resistance factors (RF), whereas the top x-axes denote z-scores (numbers of standard deviations from the mean).*

- an alignment to the reference strain HXB2, translations to the protein sequences, and alignment summary statistics
- a prediction of the HIV subtype
- predictions of phenotypic drug resistance to all approved antiretroviral agents, including the predicted resistance factor, the predicted z-score, and the predicted probability-score.

These predictions provide a rational basis for selecting drugs in an antiretroviral regimen based on viral sequence information (cf. Section 6).

5 `Theo`: Selecting optimal drug combinations

In this section, we present `theo`, a tool for therapy optimization. The aim is to further exploit the information obtained from a genotypic resistance test in order to support the selection of optimal drug combinations (Beerenwinkel, Lengauer, Däumer, Kaiser, Walter, Korn, Hoffmann & Selbig 2003).

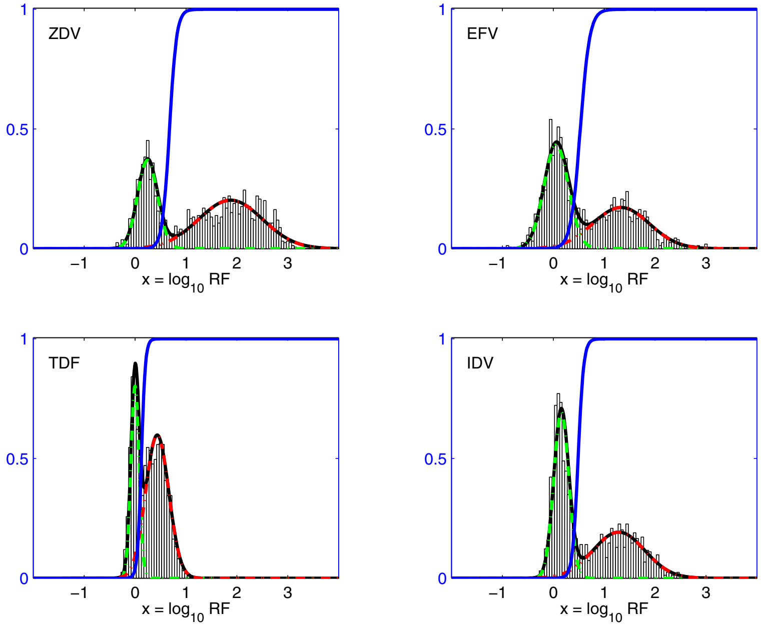


Fig. 3: Probability scores for zidovudine (ZDV), efavirenz (EFV), tenofovir (TDF), and indinavir (IDV). Histogram data and Gaussian mixture model fits are displayed for predicted resistance factors for 2000 samples drawn randomly from the population. Displayed are the bimodal mixture density (black line) and the densities for the susceptible (left bump, dashed green line) and resistant (right bump, dashed red line) subpopulations. The conditional class probability of belonging to the resistant subpopulation given the predicted phenotype is represented by the blue sigmoidal curve.

Scoring scheme

The first step is to construct a scoring function that estimates the activity of a drug combination against a given viral strain. This activity score is derived from the normalized phenotype predictions of the drugs making up the therapy.

The activity of a single drug against a given virus is defined as the probability of the virus belonging to the susceptible subpopulation. Thus, the activity of the drug is the complementary probability of the probability-score. We define the activity of a set of drugs as the sum over all drug classes of the maximum activity of the considered drugs within each class. Here, a drug class is a group of structurally similar compounds with the same mode of antiviral action. This scoring scheme amounts to a weighted counting of the number of active drug classes. It has been shown that the activity score is predictive of virological response in patients in a clinical setting (Beerenwinkel 2004).

Sequence space search

Long-term success of an antiretroviral therapy will not only depend on the current resistance phenotype of the virus, but also on its ability to escape from the selective pressure exerted by the

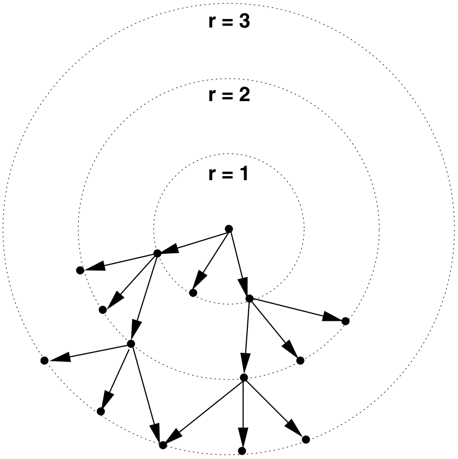


Fig. 4: Schematic diagram of beam search in sequence space in breadth $b = 2$ up two depth $d = 3$. The root of the directed graph represents the query sequence. Each edge symbolizes the introduction of a point mutation.

drug combination. As an estimate of how easily the virus can evade drug pressure, we predict activity against a worst case mutant in different mutational neighborhoods of the given sequence.

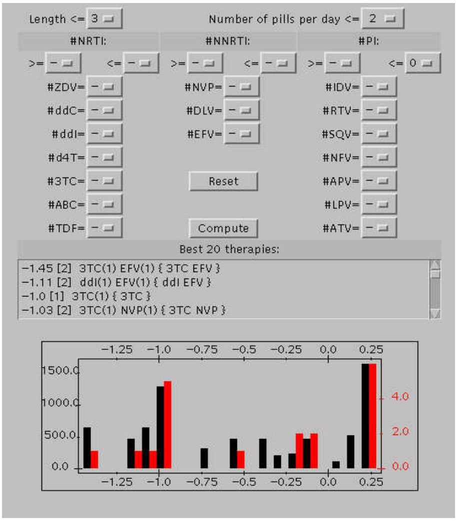
Since sequence space is so enormous, exhaustive searches are practical only for very restricted neighborhoods. Instead, we use a heuristic search strategy called beam search (Figure 4). From a selected set of sequences, we generate all one-point mutants and score them with the predicted activities. We maintain only a small number of mutants with least activity. Proceeding in this greedy fashion we follow only a fixed number of mutants, say b, at each search level d.

Searching is performed for each drug target separately. This is possible, because the activity scoring function is additive between drug classes and thus between drug targets. In order to come up with a score for a therapy comprising drugs with different targets, we follow a similar strategy as above to merge the search results for different target molecules. For each point mutation to be introduced we decide between a substitution in PR or RT by comparing activity scores. In a greedy fashion, we follow only the mutant against which the drug combination retains the least activity.

In each search level, we report the estimated worst case activity. This score vector of length $d+1$ is used to predict clinical response to therapy. Specifically, we have trained a linear classifier that discriminates successful therapies (sustained reduction of circulating virus) from failures as a function of the search depth d. For a clinical data set of 96 observed genotype-therapy pairs, the optimal search depth was found to be 3 point mutations. In particular, the estimated error rate improved considerably at search depth 3 over depth zero stressing the utility of searching the mutational neighborhood (Beerenwinkel, Lengauer, Däumer, Kaiser, Walter, Korn, Hoffmann & Selbig 2003).

Implementation

Scoring drug combinations and exploring the mutational neighborhood is implemented in the C program `theo`. The implementation separates the scoring function from the search process. Thus, any other function than the SVM based model can also be used. The beam search strategy



Length <= 3

Number of pills per day <= 2

#NRTI: >= - <= -

#NNRTI: >= - <= -

#PI: >= - <= 0

#ZDV= -
#ddC= -
#ddI= -
#d4T= -
#3TC= -
#ABC= -
#TDF= -

#NVP= -
#DLV= -
#EFV= -

#IDV= -
#RTV= -
#SQV= -
#NFV= -
#APV= -
#LPV= -
#ATV= -

Best 20 therapies:

```
-1.45 [2]  3TC(1) EFV(1) { 3TC EFV }
-1.11 [2]  ddI(1) EFV(1) { ddI EFV }
-1.0 [1]  3TC(1) { 3TC }
-1.03 [2]  3TC(1) NVP(1) { 3TC NVP }
```

Fig. 5: `Theo`*'s predicted responses to therapies. A subset of drug combinations can be specified by various criteria. Displayed are response histograms over all drug combinations (black bars) and the selected combinations (red bars). The top 20 selected therapies are given as a list.*

is realized by means of priority queues. Searching the neighborhood of a sequence of length s in breadth b ($s \gg b$) up to depth d is of time complexity $\mathcal{O}(d\,b\,s\;\log s)$.

`Theo` can be used to predict the expected virological response within a patient from the sequence of the virus that the patient carries and the selected drug combination. Thus, the program provides a means of ranking different combination therapies for an individual patient. Currently, around 7000 drug combinations are considered to be reasonable therapies, although only some hundreds of them are actually in clinical use. For a fixed viral sequence, we compute the expected response to all 7000 drug combinations. The resulting predictions are presented in a Java applet (Figure 5). The full set of therapies can be constrained by different criteria, including

- the total number of drugs,
- the minimal and maximal number of drugs from a drug class,
- inclusion or exclusion of a drug, and
- the maximum number of pills per day.

This selection tool allows for browsing through the list of therapies and focusing on drug combinations that are preferable for other reasons than resistance, such as drug-drug interactions, toxic side effects, or pill burden. The applet displays a histogram of predicted responses for all selected combinations and a list of the top 20 therapies. `Theo` is accessible at http://www.geno2pheno.org.

6 Applications

The goal of modeling HIV drug resistance is to understand the evolutionary processes that lead to reduced treatment response and to support the design of optimal drug combinations.

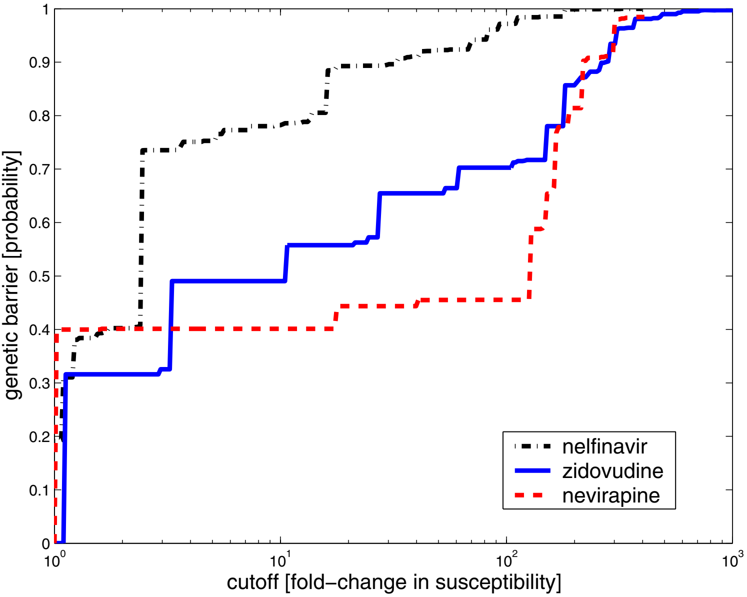


Fig. 6: Genetic barrier: Risk of reaching a genotype constellation that confers resistance of level c as a function of the cutoff value c. Over the wide range of resistance factor cutoffs between 2 and 100 we find nelfinavir > zidovudine > nevirapine, in agreement with clinical observations.

Clinical significance

We mention a clinical study from De Luca, Cozzi-Lepri, Perno, Balotta, Di Giambenedetto, Orani, Mussini, Toti & d'Arminio Monforte (2003) to demonstrate the benefit of predicting phenotypic resistance for the selection of new antiretroviral regimens. These researchers analyze therapy changes accompanied by a genotypic resistance tests in 332 previously untreated patients. Phenotypes are predicted with `geno2pheno` for the components of the combination therapies, and each drug is scored as *active* if the virus is predicted susceptible to it. Using a Cox proportional hazards model, they show that patients with a combination therapy consisting of $\leq$ 2 active drugs have a significantly higher risk of virological failure (sustained virus load increase) than patients receiving $\geq$ 3 active drugs ($p < 0.004$). The authors also compare the performance of 11 rules-based interpretation systems and our data-driven approaches. Genotypic scoring based on the SVM predictions as implemented in `geno2pheno` turns out to be the only interpretation system that provides significant predictions of virological failure after 24 weeks of treatment (De Luca, Cingolani, Di Giambenedetto, Trotta, Baldini, Rizzo, Bertoli, Liuzzi, Narciso, Murri, Ammassari, Perno & Antinori 2003).

Computing the genetic barrier

The genetic barrier of an antiretroviral drug relative to a viral strain is loosely defined as the difficulty for the virus to escape from the selective pressure of the drug by developing escape mutations. Despite its vague definition, the genetic barrier is widely believed to be a strong predictive factor of duration of treatment response. We propose a formal and rigorous definition of the genetic barrier and show how to compute this quantity.

A virus is considered to have escaped from drug pressure if the fold-change in susceptibility exceeds a certain cutoff c. We define the genetic barrier B_c of a drug relative to a given strain x as the probability of not reaching any escape state Y from x:

$$B_c(x) = \Pr(\mathrm{RF}(Y) < c | X = x) = \sum_{\mathrm{RF}(y) < c} \Pr(y | X = x).$$

The resistance factor $\mathrm{RF}(y)$ of a mutational pattern y can be predicted with `geno2pheno`, while the transition probability $\Pr(y|X = x)$ from pattern x to pattern y can be estimated with the `mtreemix` tools.

In Figure 6 we have computed the genetic barrier for three different drugs relative to the HXB2 wild type sequence as a function of the cutoff value that defines viral escape. We obtain for each specific fold-change in susceptibility the probability of not reaching any configuration of mutations conferring at least this level of resistance. For a wide range of cutoffs, we find the ordering nelfinavir > zidovudine > nevirapine, which is in concordance with clinical observations. This quantitative concept of the genetic barrier allows for systematic evaluations as a predictive factor in clinical trials (Beerenwinkel, Däumer, Sing, Rahnenführer, Lengauer, Selbig, Hoffmann & Kaiser 2005).

7 Conclusions

We have described three computational tools to model the evolution of HIV sequences under the selective pressure of combination therapies. `Mtreemix` is designed to predict mutational patterns from drug treatment, `geno2pheno` estimates phenotypic drug resistance from these patterns, and `theo` predicts virological response of a drug combination against a viral strain.

The computational models allow for investigating many virological and clinical questions in a quantitative manner. For example, we have argued that the genetic barrier can be defined in mathematical terms and computed with these tools. A clinically important question is to identify optimal treatment strategies over longer time periods. If therapy changes are unavoidable, optimal drug sequencing strategies are sought. To predict the fate of a viral population under changing drug pressures simulation models are indispensible. The *in silico* design of treatment strategies is an important complement to empirical studies, since only a tiny fraction of protocols can be tested in clinical trials.

The presented software tools are of direct benefit for clinical diagnostics. The `geno2pheno` web server supports the interpretation of genotypic drug resistance tests. As of 2004, the web server experiences on average 160 hits per day. `Theo` has been designed to support the task of selecting a new potent antiretroviral regimen on the basis of viral sequence information. The clinical decision process is complicated by complex mutational patterns and a huge number of possible drug combinations. `Theo` provides optimized therapies for the individual patient and hence constitutes a step towards personalized treatment.

Acknowledgements

The development of `mtreemix`, `geno2pheno` and `theo` has been supervised by Thomas Lengauer, Joachim Selbig, Daniel Hoffmann and Rolf Kaiser. I am grateful to Martin Däumer, Jörg Rahnenführer, Tobias Sing and Igor Savenkov for their valuable contributions. This work has been supported by Deutsche Forschungsgemeinschaft under grant No. HO 1582/1-3.

References

Alcorn, T. & Faruki, H. (2000). HIV resistance testing: methods, utility, and limitations, *Mol. Diag.* **5**(3): 159–168.

Altman, R. & Klein, T. (2002). Challenges for biomedical informatics and pharmacogenomics, *Annu. Rev. Pharmacol. Toxicol.* **42**: 113–133.

Beerenwinkel, N. (2004). *Computational Analysis of HIV Drug Resistance Data*, Shaker, Aachen, Germany. PhD Thesis, Naturwissenschaftlich-Technische Fakultät I der Universität des Saarlandes.

Beerenwinkel, N., Däumer, M., Oette, M., Korn, K., Hoffmann, D., Kaiser, R., Lengauer, T., Selbig, J. & Walter, H. (2003). Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes, *Nucl. Acids Res.* **31**(13): 3850–3855.

Beerenwinkel, N., Däumer, M., Sing, T., Rahnenführer, J., Lengauer, T., Selbig, J., Hoffmann, D. & Kaiser, R. (2005). Estimating HIV evolutionary pathways and the genetic barrier to drug resistance, *J. Infect. Dis.* p. *in press*.

Beerenwinkel, N., Lengauer, T., Däumer, M., Kaiser, R., Walter, H., Korn, K., Hoffmann, D. & Selbig, J. (2003). Methods for optimizing antiviral combination therapies, *Proc. 11th Int. Conf. on Intellig. Syst. for Mol. Biol. (ISMB 2003), June 29–July 3, 2003, Brisbane, Australia*, Vol. 19 of *Bioinformatics*, pp. i16–i25.

Beerenwinkel, N., Rahnenführer, J., Däumer, M., Hoffmann, D., Kaiser, R., Selbig, J. & Lengauer, T. (2005). Learning multiple evolutionary pathways from cross-sectional data, *Proc. 8th Ann. Int. Conf. on Res. in Comput. Biol. (RECOMB 2004), 27–31 March 2004, San Diego, CA*, pp. 36–44, to appear in *J. Comp. Biol.*

Beerenwinkel, N., Rahnenführer, J., Kaiser, R., Hoffmann, D., Selbig, J. & Lengauer, T. (2005). Mtreemix: a software package for learning and using mixture models of mutagenetic trees, *Bioinformatics* p. *in press*.
URL: *http://www.bioinformatics.oupjournals.org/cgi/content/abstract/bti274v1*

Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K. & Selbig, J. (2001a). Geno2pheno: Interpreting genotypic HIV drug resistance tests, *IEEE Intellig. Syst.* **16**: 35–41.

Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K. & Selbig, J. (2001b). Identifying drug resistance-associated patterns in HIV genotypes, *Proc. German Conf. on Bioinformatics, 7–10 October 2001, Braunschweig*, pp. 126–130.

Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K. & Selbig, J. (2002). Diversity and complexity of HIV-1 drug resistance: A bioinformatics approach to predicting phenotype from genotype, *Proc. Natl. Acad. Sci. U. S. A.* **99**(12): 8271–8276.

Boucher, C., O'Sullivan, E., Mulder, J., Ramautarsing, C., Kellam, P., Darby, G., Lange, J., Goudsmit, J. & Larder, B. (1992). Ordered appearance of zidovudine resistance mutations during treatment of 18 human immunodeficiency virus-positive subjects, *J. Infect. Dis.* **165**: 105–110.

Chow, C. & Liu, C. (1968). Approximating discrete probability distributions with dependence trees, *IEEE Trans. Inform. Theory* **14**(3): 462–467.

De Luca, A., Cingolani, A., Di Giambenedetto, S., Trotta, M., Baldini, F., Rizzo, M., Bertoli, A., Liuzzi, G., Narciso, P., Murri, R., Ammassari, A., Perno, C. & Antinori, A. (2003). Variable prediction of antiretroviral treatment outcome by different systems for interpreting genotypic human immunodeficiency virus type 1 drug resistance, *J. Infect. Dis.* **187**: 1934–1943.

De Luca, A., Cozzi-Lepri, A., Perno, C., Balotta, C., Di Giambenedetto, S., Orani, A., Mussini, C., Toti, M. & d'Arminio Monforte, A. (2003). The prognostic value to predict virological outcomes of 14 distinct systems used to interpret the results of genotypic HIV-1 drug resistance testing in untreated patients starting their first HAART, *Proc. 1st European HIV Drug Resistance Workshop: From Basic Science to Clinical Implications, 6–8 March 2003, Luxembourg*, Vol. 4 of *HIV Medicine*, p. 20.

De Luca, A., Vendittoli, M., Baldini, F., Giambenedetto, S. D., Rizzo, M., Trotta, M., Cingolani, A., Forbici, F., Perno, C., Antinori, A. & Ulivi, G. (2002). Construction, training and clinical validation of an inferential interpretation system for genotypic HIV-1 drug resistance based on fuzzy rules learning from virological outcomes, *Antivir. Ther.* **7**(Suppl. 1): 71.

DeGruttola, V., Dix, L., D'Aquila, R., Holder, D., Phillips, A., Ait-Khaled, M., Baxter, J., Clevenbergh, P., Hammer, S., Harrigan, R., Katzenstein, D., Lanier, R., Miller, M., Para, M., Yerly, S., Zolopa, A., Murray, J., Patick, A., Miller, V., Castillo, S., Pedneault, L. & Mellors, J. (2000). The relation between baseline HIV drug resistance and response to antiretroviral therapy: re-analysis of retrospective and prospective studies using a standardized data analysis plan, *Antivir. Ther.* **5**: 41–48.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussions), *J. R. Statist. Soc. B.* **39**: 1–38.

Desper, R., Jiang, F., Kallioniemi, O.-P., Moch, H., Papadimitriou, C. & Schäffer, A. (1999). Inferring tree models for oncogenesis from comparative genome hybridization data, *J. Comp. Biol.* **6**(1): 37–51.

Desper, R., Jiang, F., Kallioniemi, O.-P., Moch, H., Papadimitriou, C. & Schäffer, A. (2000). Distance-based reconstruction of tree models for oncogenesis, *J. Comp. Biol.* **7**(6): 789–803.

DiRienzo, G. & DeGruttola, V. (2002). Collaborative HIV resistance-response database initiatives: sample size for detection of relationships between HIV-1 genotype and HIV-1 RNA response using a non-parametric approach, *Proc. XI Int. HIV Drug Resistance Workshop: Basic Principles and Clinical Implications, 2–5 July 2002, Seville, Spain*, Vol. 7 of *Antivir. Ther.*, p. S71.

Drăghici, S. & Potter, R. (2003). Predicting HIV drug resistance with neural networks, *Bioinformatics* **19**(1): 98–107.

Foulkes, A. & DeGruttola, V. (2004). Characterizing the progression of viral mutations over time, *J. Am. Stat. Assoc.* **98**(464): 859–867.

Gansner, E. & North, S. (1999). An open graph visualization system and its applications to software engineering, *Softw. Pract. Exper.* **30**(11): 1203–1233.

Gonzales, M., Wu, T., Taylor, J., Belitskaya, I., Kantor, R., Israelski, D., Chou, S., Zolopa, A., Fessel, W. & Shafer, R. (2003). Extended spectrum of HIV-1 reverse transcriptase mutations in patients receiving multiple nucleoside analog inhibitors, *AIDS* **17**: 791–799.

Harrigan, P., Montaner, J., Wegner, S., Verbiest, W., Miller, V., Wood, R. & Larder, B. (2001). World-wide variation in HIV-1 phenotypic susceptibility in untreated individuals: biologically relevant values for resistance testing, *AIDS* **15**: 1671–1677.

Karp, R. (1971). A simple derivation of Edmonds' algorithm for optimum branching, *Networks* **1**: 265–272.

Karp, R. (2002). Mathematical challenges from genomics and molecular biology, *Notices AMS* **49**(5): 544–553.

Larder, B., Kemp, S. & Hertogs, K. (2000). Quantitative prediction of HIV-1 phenotypic drug resistance from genotypes: the virtual phenotype, *Antivir. Ther.* **5**(Suppl. 3): 49–50.

Lathrop, R. & Pazzani, M. (1999). Combinatorial optimization in rapidly mutating drug-resistant viruses, *J. Comb. Opt.* **3**: 301–320.

Lengauer, T. (ed.) (2002). *Bioinformatics: From Genomes to Drugs*, Vol. 14 of *Methods and Principles in Medicinal Chemistry*, Wiley-VCH, Weinheim, Germany.

Meilă, M. & Jordan, M. (2000). Learning with mixtures of trees, *J. Machine Learning Res.* **1**: 1–48.

Molla, A., Korneyeva, M., Gao, Q., Vasavanonda, S., Schipper, P., Mo, H.-M., Markowitz, M., Chernyavskiy, T., Niu, P., Lyons, N., Hsu, A., Granneman, G., Ho, D., Boucher, C., Leonard, J., Norbeck, D. & Kempf, D. (1996). Ordered accumulation of mutations in HIV protease confers resistance to ritonavir, *Nat. Med.* **2**(7): 760–766.

Oette, M., Kaiser, R. & Häussinger, D. (eds) (2004). *Resistenz in der HIV-Therapie: Diagnostik und klinisches Management*, UNI-MED, Bremen, Germany.

Perrin, L. & Telenti, A. (1998). HIV treatment failure: testing for HIV resistance in clinical practice, *Science* **280**: 1871–1873.

Radmacher, M., Simon, R., Desper, R., Taetle, R., Schäffer, A. & Nelson, M. (2001). Graph models of oncogenesis with an application to melanoma, *J. Theor. Biol.* **212**: 535–548.

Ravela, J., Betts, B., Brun-Vézinet, F., Vandamme, A.-M., Descamps, D., Van Laethem, K., Smith, K., Schapiro, J., Winslow, D., Reid, C. & Shafer, R. (2003). HIV-1 protease and reverse transcriptase mutation patterns responsible for discordances between genotypic drug resistance interpretation algorithms, *J. Acquir. Immune. Defic. Syndr.* **33**(1): 8–14.

Sander, C. (2000). Genomic medicine and the future of health care, *Science* **287**: 1977–1978.

Sevin, A., DeGruttola, V., Nijhuis, M., Schapiro, J., Foulkes, A., Para, M. & Boucher, C. (2000). Methods for investigation of the relationship between drug-susceptibility phenotype and human immunodeficiency virus type 1 genotype with applications to AIDS clinical trials group 333, *J. Infect. Dis.* **182**: 59–67.

Stürmer, M., Doerr, H., Staszewski, S. & Preiser, W. (2003). Comparison of nine resistance interpretation systems for HIV-1 genotyping, *Antivir. Ther.* **8**: 55–60.

Vandamme, A., Van Laethem, K. & De Clerq, E. (1999). Managing resistance to anti HIV drugs: an important consideration for effective disease management, *Drugs* **57**: 337–361.

von Heydebreck, A., Gunawan, B. & Füzesi, L. (2004). Maximum likelihood estimation of oncogenetic tree models, *Biostatistics* **5**(4): 545–556.
URL: *http://biostatistics.oupjournals.org/cgi/content/abstract/5/4/545?etoc*

Walter, H., Schmidt, B., Korn, K., Vandamme, A. M., Harrer, T. & Überla, K. (1999). Rapid, phenotypic HIV-1 drug sensitivity assay for protease and reverse transcriptase inhibitors, *J. Clin. Virol.* **13**: 71–80.

Wang, D. & Larder, B. (2003). Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks, *J. Infect. Dis.* **188**: 653–660.

Wu, T., Schiffer, C., Gonzales, M., Taylor, J., Kantor, R., Chou, S., Israelski, D., Zolopa, A., Fessel, W. & Shafer, R. (2003). Mutation patterns and structural correlates in human immunodeficiency virus type 1 protease following different protease inhibitor treatments, *J. Virol.* **77**(8): 4836–4847.

Weitere Beiträge für den Heinz-Billing-Preis 2004

A Novel Approach to Symbolic Algebra

Thomas Fischbacher
Max-Planck-Institut für Gravitationsphysik,
Albert-Einstein-Institut Potsdam

Abstract

A prototype for an extensible interactive graphical term manipulation system is presented that combines pattern matching and nondeterministic evaluation to provide a convenient framework for doing tedious algebraic manipulations that so far had to be done manually in a semi-automatic fashion.

1 Introduction

Despite the availability of generic symbolic term manipulation packages – computer algebra systems like Mathematica, Maple, MuPAD, to name just a few well-known ones – and despite their wide application in calculation-intensive fields of study, such as (especially) theoretical physics, the necessity to do lengthy pen and paper calculations that take days or even weeks still persists – especially in string theory and related fields. The primary underlying reason for this seemingly paradoxical situation seems to be that the way how calculations are communicated to these automated systems does not mirror closely enough the way how one usually thinks about doing a calculation by hand. In particular, while these systems have their own ideas about implicit canonical reductions that should be applied to newly generated terms automatically, communicating a step to them as simple as 'do quadratic completion on the second factor of the third summand' that appears perfectly natural when mathematicians talk at the blackboard already requires comparatively sophisticated programming. While the presently available systems excel at doing lengthy symbol crunching along the lines of clearly stated procedures, it is especially their inherent weakness in doing user interaction that makes these systems virtually useless for the much more explorative calculation style one has to use frequently e.g. when trying to prove some algebraic property of a lengthy expression.

In this work, the `term-abacus` prototype is presented which targets specifically this problem. While this prototype is still in a too early stage to do any productive work with it, it both demonstrates the viability of the underlying approach and has reached a stage in its development where so many important design decisions have to be made that further scientific discussion is required first.

2 The Problem (and its solution)

Generally, a 'calculation' is the application of a sequence of transformation rules to a term. For operational purposes, one can regard these as substitutions having two parts: a *pattern* matching part of the structure of a term and a *template* that is parametrized by pieces of the pattern and dictates how the substitute has to be constructed. These rules can be quite complicated in some instances, and hence, due to human imperfection, lengthy pen and paper calculations are inherently prone to sloppiness errors. These are certainly not the only mistakes one can make, but the ones that can be avoided most easily by letting a machine take care of proper application of substitution rules. Furthermore, the less of one's time – and more importantly – mental energy one has to use for purely mechanical tasks, the more one can concentrate on the important aspects, and hence the deeper one can go.

Using pattern matching to express calculation rules is not a new idea – it plays a key role in many of the well known symbolic algebra systems, such as FORM or Mathematica, where it has proven its value. While many calculation rules can be formulated conveniently as pattern matching and substitution rules suitable for a computer (once an agreement on the underlying term representation has been made), one fundamental problem is that, quite in general, there often are different ways how to apply such a rule to a given term in a typical calculation. Whenever there is a purely mechanical way to decide which route to follow, the corresponding step in the calculation is not interesting, as it does not require human intelligence. But evidently, as every interesting calculation does require human intelligence for its solution, one is almost bound to encounter the difficulty to find a way to convey enough information to the machine to completely specify the particular transformation one has in mind. As this is a key issue in this work, we want to introduce special terminology here and from now on call this procedure *term clamping*.

It is essential that term clamping has to work in a most unobtrusive way, requiring as little thought from the end user as possible, or such a system would be perceived as clumsy and unusable when it comes to real-world applications.

First and foremost, this means that the amount of extra information that has to flow has to be minimized – typing a full command, or determining values for additional parameters, is already too much of a hassle. In addition, any information flowing back from the machine to the user during that process must be presented in such a way that it does not require any additional interpretation.

On the other side of the coin, it is strongly desirable to find a clean and concise way to express the logic behind term clamping in the program code of the implementation of a term manipulation system. The essence behind term clamping is basically a nondeterministic choice between different possible futures, introduced into the system by an intelligent (and hence unpredictable) human user. As humans usually do calculations on terms which they can easily capture visually as a whole, the truly minimal amount of information flows between human and machine if the machine offers a simple visual choice between all possibilities how to apply a given calculation rule to a term. For the program, however, this means that we have to produce multiple solutions to the problem of recognizing a given pattern inside some larger structure. This is most easily effected by making use of *ambiguous evaluation* [11, 1, 3].

Basically, this means that the system which we want to use to implement such a term manipulation system should support a notion of making choices and forking a calculation into many different branches, each of which may produce its own result, or may turn out as futile and fail to return anything if the corresponding choice is incompatible with extra requirements we want to impose.

To give a specific example, suppose we wanted to extract all non-overlapping (unordered) pairs of triples of consecutive five-letter-words from a sentence such as:

```
The swift small brown horse might
never ever allow being shoed
```

One would basically want to express an idea like that – which admittedly sounds a bit synthetic, but actually has a lot in common in structure with the calculation rule patterns we are interested in – in a Scheme [10][1] program of

[1] This work is based on the free MzScheme implementation, which belongs to the PLT Scheme family [13]

roughly the following form[2]:

```
(define sentence
  '("The" "swift" "small" "brown" "horse"
    "might" "never" "ever" "allow" "being" "shoed"))

(define empty? null?)

(define (n-th n list)
  (if (empty? list)
      (fail)
      (if (= n 1)
          (first list)
          (n-th (- n 1) (rest list)))))

(define (n-th-rest n list)
  (if (= n 0)
      list
      (if (empty? list)
          (fail)
          (n-th-rest (- n 1) (rest list)))))

(define (find-three-consecutive-5-letter-words
     list-words)
  (let ((word1 (n-th 1 list-words))
        (word2 (n-th 2 list-words))
        (word3 (n-th 3 list-words)))
    (either
     (if (and (= (string-length word1) 5)
              (= (string-length word2) 5)
              (= (string-length word3) 5))
         list-words
         (fail))
     (find-three-consecutive-5-letter-words
      (rest list-words)))))

(define (find-pairs-3x5 list-words)
  (let* ((first-occurrence
          (find-three-consecutive-5-letter-words
       list-words))
         (second-occurrence
          (find-three-consecutive-5-letter-words
          (n-th-rest 3 first-occurrence))))
```

[2] Clearly, this has been written with readability for a broad public in mind; a seasoned programmer would e.g. most probably not let index counting start at 1. Hint to non-Scheme programmers: Scheme code is read mostly by indentation, ignoring most of the parentheses.

```
    (list (list (n-th 1 first-occurrence)
                (n-th 2 first-occurrence)
                (n-th 3 first-occurrence))
          (list (n-th 1 second-occurrence)
                (n-th 2 second-occurrence)
                (n-th 3 second-occurrence)))))

(all-values (find-pairs-3x5 sentence))



#| Result:

((("swift" "small" "brown") ("horse" "might" "never"))
 (("swift" "small" "brown") ("allow" "being" "shoed"))
 (("small" "brown" "horse") ("allow" "being" "shoed"))
 (("brown" "horse" "might") ("allow" "being" "shoed"))
 (("horse" "might" "never") ("allow" "being" "shoed")))

|#
```

Even if not the details, at least the general structure of this program should be understandable even for non-Scheme-programmers. What is especially interesting here is that failure of a calculation branch can happen at very different places, at different nesting levels in the calculation, that is, the calculation is highly non-uniform between different branches.

The Scheme programming language does not provide such highly unusual constructs like `either` or `fail` or `all-values`. But remarkably enough, it does provide an universal tool that allows one to catch the future of any given computation to store it away, or even call it multiple times (jettisoning the future of the current actual calculation itself), called `call-with-current-continuation`. With this, it is possible to seamlessly extend the language by virtually construct that involves highly nontrivial changes in execution flow – such as in particular nondeterministic features of exactly the form presented above – with little effort. Indeed, it can be done in less than fifty lines of extra code; this is explained in the appendix.

The example presented here indeed can be regarded as a specific instance of a matching problem of just the type that covers a large set of calculation rules. Under the premise of doing algebraic calculations, our terms generally will be sums of individual summands that consist of a coefficient plus a series of further factors that should be treated as noncommuting by default, as we want to be able to convey extra information in the order of factors. A typical 'local' calculation transformation will then have a form like

$$a_\mu a^\dagger_\nu \to a^\dagger_\nu a_\mu + \eta_{\mu\nu} \qquad (2.1)$$

where it is understood that $a_\mu a_\nu^\dagger$ is a pattern that matches against a subsequence of two operators of types $aa^\dagger$ that carry small Greek indices that have to be substituted into the expression on the right hand side wherever the corresponding actual indices mentioned in the rule appear.

Besides this, we also want to be able to apply rules where the location of individual pieces in the sequence of factors does not matter, and hence matching should succeed regardless of their position, and especially without first having to move pieces around. One simple example of such a rule would be:

$$\begin{array}{rr} \ldots\epsilon_{ijk}\ldots\epsilon_{imn}\ldots \rightarrow & \ldots\delta_{jm}\ldots\delta_{kn}\ldots \\ & -\ldots\delta_{jn}\ldots\delta_{km}\ldots \end{array} \tag{2.2}$$

One will typically prefer to place the deltas in other places than in this specific example, but this is not essential here. What is important is this particular form of a pattern.

There furthermore are rules where one that have both properties at the same time, that is: one wants to match a number of specific fixed-length sequences of factors which may appear at various positions in a term. An example of a rule that is usually expressed in such a way is the Fierz-Pauli identity, which rather should be regarded as a collection of various calculation rules that allow to re-arrange certain expressions that carry four fermions (denoted by small Greek letters), such as[3]

$$\begin{array}{rrl} \ldots\psi\Gamma^\alpha\phi\ldots\lambda\Gamma_\alpha\eta\ldots \rightarrow & & \ldots\psi\eta\ldots\lambda\phi\ldots \\ & -\frac{1}{2} & \ldots\psi\Gamma^\alpha\eta\ldots\lambda\Gamma_\alpha\phi\ldots \\ & -\frac{1}{2} & \ldots\psi\Gamma^{\alpha\beta\gamma}\eta\ldots\lambda\Gamma_{\alpha\beta\gamma}\phi\ldots \\ & - & \ldots\psi\Gamma^5\eta\ldots\lambda\Gamma^5\phi\ldots \end{array} \tag{2.3}$$

This is – up to the issue of ordering pieces – precisely the structure of our text matching example: we want to be able to match a collection of non-overlapping fixed-length subexpressions with additional constraints in a sequence in all possible ways and let the user choose. This type of pattern is also so common that we should coin a special term for it – let us call this a sequences-of-factors-pattern, in short *sofpa*. At the core of the `term-abacus` prototype lies a nondeterministic sofpa-matching engine.

Internally, this matching engine produces a list of sub-sequences which carry annotations which part of the pattern they matched, or if they lie between patterns, plus information about identifications of jokers within these patterns.

While this general structure covers many calculation rules, there are as well examples of term transformations that cannot be expressed in such a way. Among those, however, one also finds many re-occurring structures.

[3] See any textbook on quantum field theory like [7] or [9]

One major central concept which re-occurs in many guises but can not be captured in a sofpa rule is forming a variation of an entire term (not only a specific summand) along the lines of the Leibniz rule:

$$\delta(ab) = a(\delta b) + (\delta a)b \tag{2.4}$$

Within the present framework, the approach taken is to first implement them in a more ad-hoc way, and look for re-occurring structure that should be abstracted out while the prototype evolves.

In the present form of the prototype, a sofpa rule is represented internally as an associative list containing a pattern (which is a list of chains of factors), a substitution part, and additional information about highlighting telling which matched parts of a pattern to display in a visually distinct way. In particular, a rule like quantum mechanical normal ordering

$$aa^{+} \to a^{+}a + 1$$

is denoted internally as follows:

```
(define *rule-normal-ordering-a*
  `((pattern . ((,(as-pattern '?a '((a)))
                 ,(as-pattern '?a+ '((a "+"))))))
    (subs .
          #(;; summands
            (1 .
               #(;; one subs for every
         ;; factor-block pattern
                 (((a "+")) ((a)))))
            (1 .
               #(()
                 ))))
    (highlighting .
      ((?a . green)
       (?a+ . green)))))
```

To the system, every term is a vector of pairs of a coefficient and a list of factors, and every factor within a summand is a pair of a stem and either an exponent or a list of tensor (upper and lower, which denote contravariant and covariant) indices. The stem itself consists of a symbol and additional ornaments which are symbol specific, that is, the system contains a set of hooks where one can provide arbitrary code that defines the meaning and visual representation of that particular symbol. Via those means, it is e.g. possible to extend the system by a definition of a `del` symbol which carries as ornament some other field plus an index and renders e.g.

((del ((F) (down . mu) (down . nu))) (down . rho)) visually as $\partial_\rho F_{\mu\nu}$. While one would wish to retain the highest possible flexibility for extending the system with new interpretations for term ornaments, there are a few slightly subtle issues including (but not limited to) behaviour under renaming of silent indices one has to be aware of. (Upon a closer look, category theory appears to be a language especially well suited to talk about these subtleties.)

At a more elementary level, the matcher tries to perform a one-to-one recursive structural match between pattern and value similar to the equal? Scheme function, but with the additional features that a special 'joker symbol' in the pattern (the default convention – which can be changed – is that all symbols whose name starts with a '?' are joker symbols) matches either an arbitrary value, or if it occurs more than one time in the pattern, a set of equal? values. Furthermore, there are other special classes of jokers that e.g. match a sublist of arbitrary length of a given list (an inherently nondeterministic specification). One may also place jokers into the right-hand side substitution template, where they are either instantiated to the corresponding pieces matched by the template, or to letters from various alphabets with the additional guarantee that no such joker is instantiated to a letter that occurs somewhere else in the summand (this is to conveniently implement the generation of silent indices). If the pattern contains a function, this function is called with the corresponding part of the value and information about previous matches as arguments and may nondeterministically fail, or provide multiple choices of further successful match information. One observes that this scheme is flexible enough to easily transfer the spirit of any complex pattern matching notion to this system, such as those of guards or as-patterns in Haskell, by using pattern-matching-function-generating-functions, like as-pattern, which maps a joker name and a sub-pattern to a function matching against the sub-pattern and, if successful, binding the matched value against the provided joker name.

The design decisions about the internal structure of terms are in part motivated by the goal to use it for tensor algebra as required for quantum field theory. While it may seem strange at first to provide direct support e.g. for such a special detail as tensor indices, which one might rather like to think of an issue to be resolved at the level of specifying factor ornaments, this actually turns out to be necessary. One may catch a glimpse of the underlying issues by observing that it certainly makes sense to allow factors to carry powers and provide direct support for this, while powers and indices are a non-orthogonal concept in the sense that one cannot make sense of an expression like $(C_{abc})^3$.

Clearly, the language in which calculation rules are expressed is still way too low-level to be of use to end users of the system, but as implementing application-specific languages (in the broadest sense) is what systems such as Scheme from the LISP family truly excel at [1], this is merely a question of experimenting with different notations until one is discovered that turns out to be simple, powerful, and well-suited for use by non-schemers.

A further comment has to be made concerning the possibility to use the flexibility of the system to transfer 'sloppy' calculations one-to-one to the machine – in the sense that one may choose representations of factors that are ignorant of certain aspects that are conceptually vital from the mathematical point of view (such as the dependency of fields on the particular point in spacetime) but do not matter for some particular calculation one wants to do. While it is nice to specify all the mathematical structure in full detail, as this helps to come to a deep understanding of the subject and discover many interesting conceptual subtleties[4], it is perhaps nevertheless a good idea not to impose too great restrictions on the level of rigor to the user, as the ability to leave even conceptually important details that turn out not to have any influence on the calculation out of the description does have its advantages.

To summarize this section, the problem of allowing the user to communicate choices about where to apply a given calculation rule if there are multiple possibilities is most directly expressed in terms of nondeterministic evaluation and continuations. This is a suitable language to formulate rule patterns in a concise fashion, but there are many details one has to be aware of that require additional discussion.

3 Anatomy of the `term-abacus` prototype

As mentioned previously, the `term-abacus` prototype is implemented in MzScheme, as this provides a lot of highly useful infrastructure such as lisp-style `defmacro` macros, a lexer and parser as well as (most important!) continuations and even a framework to implement continuation-based web services following the ideas presented in [4].

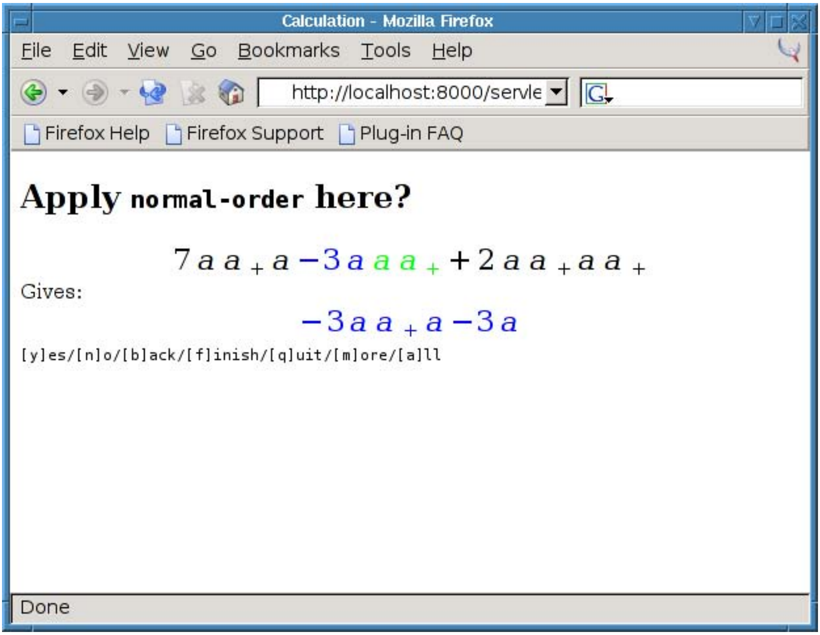
The first and foremost problem that has to be overcome is to find means to visually display terms in a convenient way and also allow user input. The requirement to support user input as well as the generic problem that solutions built by coercing various independent components to cooperate which were never intended to do so by making massive use of interprocess communication typically leads to brittle systems that may react very badly to version updates of individual components basically excludes any solution based on employing TEX to do the rendering. While the idea may be tempting to try to implement this system as a special Emacs mode, as Emacs at least in spirit

[4] One may consider especially the treatise on classical mechanics [15] as a prophetical exemplification of the power of this philosophy

intends to be a substrate for such kinds of application, this does not work as the text rendering features of Emacs are not sophisticated enough to do term typesetting at the level required for this application with it. Furthermore, Emacs Lisp does not support continuations, not even closures. At first sight, TEXmacs [16] appears as a much more attractive alternative, being an Emacs-inspired WYSIWYG-style text editor with advanced TEX rendering capabilities and a proper scheme (FSF's guile [5]) as scripting language. Unfortunately, TEXmacs is still quite power-hungry, and the amount of rendering functionality exported to Scheme was too small to build such a system on top of it for a long time (this may have changed by now).

An earlier LISP-based version of the `term-abacus` prototype made use of Screamer [14] to implement nondeterminism (which turned out too weak as it could not handle nondeterministic anonymous functions well and led to overly clumsy code) and Zebu [17] as parser generator, and employed an own simple renderer that was very loosely inspired by the way how monadic I/O works in Haskell to implement abstract rendering functionality which was then used by specialized renderers to generate TEX, ASCII, as well as graphical output. TEX output capability is evidently important to be able to directly use results obtained in `term-abacus`. ASCII output is important as we obviously need functionality to use a simple syntax for ASCII term input, and even with the most powerful system, one might want to do ad-hoc modifications not covered by any known calculation rule on terms that are best done by editing an ASCII representation. Graphical output was implemented by building a tree of typographic glyphs with additional relative positioning information which were then drawn by help of the clg [6] LISP-GTK interface, extended with some own functions. Eventually – mostly due to difficulties based on missing continuation support in LISP – MzScheme turned out to be a system much better suited to build this prototype on.

The switch to MzScheme made a very different and quite exciting approach to the rendering problem possible: as this Scheme implementation contains a full-fledged webserver and special infrastructure to build highly interactive continuation-based web services, the possibility to abuse a web browser with MathML support as a graphical front-end becomes feasible, cf. figure 1. (Incidentally, the idea behind employing continuations for web services is to use them to implement specialized control flow structures to hide all the underlying control transfer complexity – which comes from the web request-response model in this case. This is quite similar in spirit to the central idea behind `term-abacus`.) This is attractive for two reasons: first, MathML is gradually emerging as a standard for typesetting formulae that can be used with a variety of different applications, second, this immediately allows one to provide all the functionality of this system as a web service. The drawback of such an approach is that it brings along certain restrictions


Calculation - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8000/servle
Firefox Help
Firefox Support
Plug-in FAQ
Done

Apply normal-order here?

$$7 a a_+ a - 3 a a a_+ + 2 a a_+ a a_+$$

Gives:

$$-3 a a_+ a - 3 a$$

```
[y]es/[n]o/[b]ack/[f]inish/[q]uit/[m]ore/[a]ll
```

Fig. 1: An example for interactive non-deterministic choice in`term-abacus`

concerning the user interface. Basically, experience tells us that in order to use such a system in a fast and efficient way, one wants to be able to use it via keystroke commands, which means that JavaScript has to be used to a certain extent to implement the user side of the system, and furthermore, there are limitations on the keystroke commands one may use as some interfere badly with web browser internal keystroke commands. Another issue is that MathML support is still poor with many graphical web browsers. At present, the prototype is intended to be a system which specifically abuses the mozilla firefox browser[8] as graphical interface which accidentally also can be used over the internet and not yet a generic browser independent web application. Firefox users which have appropriate fonts for MathML installed[5] can have a peek at an early stage of the system, which is under active development, at `http://term-abacus.aei.mpg.de:8000` (the source is also available there).

The current MathML renderer is a modified variant of the TeX renderer that was transliterated directly from the LISP predecessor and still uses string blocks and templates internally. This is bound to change, as XML (of which

[5] Debian GNU/Linux users should install the `latex-xft-fonts` package

MathML is an application) can be embedded directly into Scheme S-expressions, which is considered a much cleaner and more powerful approach.

Besides the matching engine, the renderer and the web interface code, another important component of the system is the term input parser. At present, the intention behind this parser is to provide the most basic means to input terms as strings like

```
-7/2 e**4 X_a_b Q^a^b_alpha + 5 Z_alpha
```

only in order to keep things simple, but also be extensible by allowing users to register (almost) arbitrary extra parsers for special symbols that process symbol ornaments. The only restriction on such special user-definable parsers is that symbol ornaments will be delimited by matching pairs of brackets []; hence, it is easily possible to introduce e.g. a user-defined parser for things like lepton spinor factors using a syntax like `u[bar,mu;p_1]` or `v[tau;p_2]` if the application wants this, but it is not possible to introduce parsers for ornaments with non-well-formed bracket structure. At the implementation level, a two-stage LALR(1) parser is constructed employing MzScheme's parser generator functionality as it is not possible to let matching brackets delimit tokens by employing a regular lexer only.

4 Conclusions and Outlook

While the idea to build a term manipulation system that is suited for a much more interactive style of working than all other existing symbolic algebra packages by using nondeterministic language to concisely model user interaction nondeterminism at the level of the implementation is very attractive, and has been shown to be feasible with very moderate programming effort, this approach still has to prove its value, as the prototype implemented here is still a bare-bones system that provides all the abstract functionality to implement specific term manipulation systems on top of it, but no such system that uses `term-abacus` has been constructed yet. At least, preliminary experiments with an implementation of a thermodynamics-oriented term algebra of partial derivatives on top of the LISP-based predecessor of that system seemed quite promising. One interesting smoke test that should be within reach with justifiable effort would be to implement a set of transformation rules which allow one to do calculations such as the derivation of the Lagrangian of eleven-dimensional supergravity [2] as easily as possible. This should also show where the `term-abacus` system still requires to be refined and extended.

Acknowledgements

I want to express my gratitude to Klaus Aehlig for many invaluable discussions, especially on parser issues.

Appendix: On nondeterministic evaluation and Scheme

While continuation-based techniques are well established in the functional programming community, they gained surprisingly little attention (especially when considering their power) in the mainstream so far. Considering especially their perceived usefulness for building highly interactive web services – an approach that was popularized especially by Paul Graham [4] – they may well be on the verge of becoming the next hot issue that makes its way into the mainstream via convoluted paths which has been known to lisp hackers for decades – just as it was the case with standardized support for hash tables, print-read-consistency for textual representations of recursively structured data [12], proper garbage collection, and many others.

The basic idea behind continuations is that there is a symmetry between calling a function and returning a value from a function, the latter one being just a 'call to the function representing the entire future of the present calculation'. If we forget about technicalities such as when to free which type of memory object, then on the conceptual level, even the return address from a C function on the stack may just as well be regarded as an extra function pointer parameter denoting 'the entire rest of the program as a callable function to which we pass on our return value'. With this philosophy in mind, it is possible to mechanically transform every program to so-called 'continuation passing style' (CPS) where all functions take as an extra parameter. For a very simple function like a naive implementation of the factorial, this would look as follows:

```
(define (factorial n)
  (if (= n 0) 1 (* n (factorial (- n 1)))))

;; === the same after a full CPS transform ===

(define (return x)
  (lambda (c) (c x)))

(define (cps= cont-a cont-b cont)
  (cont-a
   (lambda (a)
     (cont-b
      (lambda (b)
        (cont (= a b)))))))

(define (cps* cont-a cont-b cont)
  (cont-a
   (lambda (a)
     (cont-b
      (lambda (b)
        (cont (* a b)))))))

(define (cps- cont-a cont-b cont)
  (cont-a
   (lambda (a)
     (cont-b
      (lambda (b)
        (cont (- a b)))))))

(define (cps-if cont-test cont-then cont-else cont)
  (cont-test
   (lambda (bool)
     (if bool
         (cont-then cont)
         (cont-else cont)))))

(define (cps-factorial cont-n cont)
  (cps-if
   (lambda (c) (cps= cont-n (return 0) c))
   (return 1)
   (lambda (c)
     (cps* cont-n
           (lambda (c)
             (cps-factorial
              (lambda (c) (cps- cont-n (return 1) c)) c))
           c))
   cont))

;; (cps-factorial (return 5) display) ==> 120
```

Note that in the definition of `cps-factorial`, there is not a single place left where a value is returned; furthermore, execution order is totally specified now. While transformation to CPS plays an important role 'under the hood' of a scheme system, complex code written in full CPS style is evidently almost unreadable to human beings. It needn't be, however, as all continuation-related issues are hidden from the user, the only exception being just the `call-with-current-continuation` function (and the values it generates), which allows the user to get a handle at the future of the entire program at an arbitrary point to store it away and *jump* back to this place in the program at any point in time, even multiple times, with all the surrounding context properly set up.

This construct gives us a bewildering flexibility to extend the language with new control flow constructs. For example, nondeterministic evaluation as used in the `term-abacus` prototype may be implemented along the following lines (the idea being to let `all-values` jump down deeply into a calculation which contains many choice points over and over again until all choices have been seen and the list of all values is passed on to its own continuation):

```
(require (lib "defmacro.ss"))

(define call/cc call-with-current-continuation)
(define __cont-other '())

(define (__all-values lambda-expr)
  (let ((results '()))
    (fluid-let ((__cont-other '()))
      (call/cc
       (lambda (ret)
       ; catch the continuation of all-values
         (set! __cont-other
           `(,(lambda () (ret (reverse results)))))
         (set! results (cons (lambda-expr) results))
         ((car __cont-other)))))))

(define-macro (all-values . body)
  `(__all-values (lambda () . ,body)))

(define (choose choices)
  (let ((rest-choices choices))
    (call/cc (lambda (c)
           (set! __cont-other (cons c __cont-other))))
    (if (null? rest-choices)
        (begin
          (set! __cont-other (cdr __cont-other))
          ((car __cont-other)))
        (let ((next (car rest-choices)))
          (set! rest-choices (cdr rest-choices))
          next))))

;; This must be a macro, since we do not want
;; to eval this and that!
;; primitive method, just along the lines
;; of CHOOSE.

(define-macro (either this that)
  (let ((sym-c (gensym "c-"))
        (sym-todo (gensym "todo-"))
        (sym-next (gensym "next-")))
    `(let ((,sym-todo
        (list (lambda () ,this) (lambda () ,that))))
       (call/cc
        (lambda (,sym-c)
      (set! __cont-other (cons ,sym-c __cont-other))))
       (if (null? ,sym-todo)
           (begin
             (set! __cont-other (cdr __cont-other))
             ((car __cont-other)))
           (let ((,sym-next (car ,sym-todo)))
             (set! ,sym-todo (cdr ,sym-todo))
             (,sym-next))))))

(define (fail) (choose '()))

;; (all-values
;;    (cons (choose (list 1 (choose (list 2 3))))
;;          (+ 100 (choose (list 10 20 30)))))
;; => ((1 . 110) (1 . 120) (1 . 130)
;;     (2 . 110) (2 . 120) (2 . 130)
;;     (1 . 110) (1 . 120) (1 . 130)
;;     (3 . 110) (3 . 120) (3 . 130))
```

References

[1] Hal Abelson, Gerald Jay Sussman and Julie Sussman, “Structure and Interpretation of Computer Programs,” MIT Press and McGraw-Hill, 1985, second edition 1996. (`http://mitpress.mit.edu/sicp/full-text/book/book.html`).

[2] E. Cremmer, B. Julia and J. Scherk, “Supergravity Theory In 11 Dimensions,” Phys. Lett. B **76** (1978) 409.

[3] Paul Graham, “On Lisp,” Prentice Hall, 1993. (`http://www.paulgraham.com/onlisptext.html`).

[4] Paul Graham, “Beating the Averages,” `http://www.paulgraham.com/lwba.html`

[5] The guile scheme homepage is `http://www.gnu.org/software/guile/guile.html`

[6] Espen S. Johnsen, Common Lisp bindings to GTK+ `http://sourceforge.net/projects/clg`

[7] M. Kaku, “Quantum Field Theory: A Modern Introduction,” Oxford University Press, 1993.

[8] The firefox browser is available from `http://www.mozilla.org`.

[9] M. E. Peskin and D. V. Schroeder, “An Introduction To Quantum Field Theory,” Addison-Wesley, 1995.

[10] Guy Lewis Steele Jr., “RABBIT: A Compiler for SCHEME,” Masters Thesis. MIT AI Lab. AI Lab Technical Report AITR-474. May 1978. (`ftp://publications.ai.mit.edu/ai-publications/pdf/AITR-474.pdf`)

[11] John McCarthy, “A Basis for a Mathematical Theory of Computation,” In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*. North-Holland, 1967 (http://www-formal.stanford.edu/jmc/basis1/basis1.html).

[12] John McCarthy, “Common Business Communication Language,” In Albert Endres and Jürgen Reetz, editors, Textverarbeitung und Brosysteme. R. Oldenbourg Verlag, Munich and Vienna, 1982.
(http://www-formal.stanford.edu/jmc/cbcl2/cbcl2.html)

[13] The PLT Scheme website is http://www.plt-scheme.org.

[14] J. Siskind, “Screaming Yellow Zonkers,” Draft Technical Report of 29th September 1991, supplied with the SCREAMER code distribution 3.20

[15] Gerald Jay Sussman and Jack Wisdom with Meinhard E. Mayer, “Structure and Interpretation of Classical Mechanics,” MIT Press, 2001.
(http://mitpress.mit.edu/SICM/)

[16] Joris van der Hoeven, http://www.texmacs.org

[17] William M. Wells III 1989, Zebu, LALR(1) parser system,
(http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/scheme/code/parsing/zebu/0.html)

Computer Aided Modeling of Chemical and Biological Systems – Methods, Tools, and Applications

Michael Mangold, Odon Angeles-Palacios, Martin Ginkel,
Andreas Kremling, Roland Waschler, Achim Kienle,
Ernst Dieter Gilles
Max-Planck-Institut für Dynamik komplexer technischer
Systeme, Sandtorstraße 1, 39106 Magdeburg

1 Introduction

In chemical engineering, dynamic process models based on conservation laws have become an indispensable tool for the development of new processes and the improvement of existing ones. In biology, the rapidly increasing knowledge of cellular processes guides the way for a quantitative description of biological systems. However, the development of realistic and predictive models is a challenging and time consuming task in both sciences. This has several reasons: To a large extent, modeling consists of choosing, validating, and revising physical model assumptions. It is an iterative process. Virtually every model of a complex process is inadequate at the beginning and requires a lot of refinements before it delivers satisfactory results. Often, it is necessary that experts from different fields share their knowledge during the model development process. In such a case, engineers used to differential equations on the one hand, and chemists and biologists thinking in qualitative models on the other hand must find a common language to exchange their ideas. The resulting detailed process models typically contain a large amount of information. Usually, they are implemented in a monolithic way without much internal structuring. This makes the understanding and debugging of the model difficult. Existing models are not very transparent and hardly reusable for another modeler. Furthermore, the implementation of complicated differential equations in a flow-sheet simulator is tedious and error prone. Finally, in most simulation tools it is in the responsibility of the modeler to formulate his models in a manner suitable for numerical treatment, e.g. to avoid a high differential index of a differential algebraic system.

In the last years, efforts have been made to support the model development process by computer tools. The main objectives of a modeling tool are

- To let a user concentrate on the physical modeling task and to relieve him from mechanical coding work.
- To increase the re-usability and transparency of existing models.
- To simplify the debugging process during model development.
- To provide libraries of pre-defined building blocks for standard modeling tasks like reaction kinetics, physical properties, or transport phenomena.

In the field of chemical engineering, general structuring methodologies have been proposed by several authors. Based on these theoretical concepts, modeling languages as well as modeling tools have been developed. In the field of mathematical modeling of biological systems, and especially of cellular systems, computer tools as well as language standards for model formulation have been published. For more information, the reader is referred to (Mangold, Angeles-Palacios, Ginkel, Kremling, Waschler, Kienle & Gilles 2005) and references therein.

The purpose of the present contribution is to give a review on recent results in the field of computer aided modeling that have been obtained at the Max Planck Institute in Magdeburg. These results are based on fundamental research done within the joint research project SFB 412 at the University of Stuttgart. In the next section, a general model structuring methodology will be presented that is applicable to biological as well as to chemical engineering processes. This method provides the theoretical foundation for the process modeling tool PROMOT, whose key features will be discussed in the following section. A number of different applications have been implemented in PROMOT, so far. In the area of chemical engineering, this includes reactive distillation processes (Tränkle, Kienle, Mohl, Zeitz & Gilles 1999), integrated chemical production plants (Waschler, Angeles-Palacios, Ginkel & Kienle 2003), and model libraries for membrane reactors (Mangold, Ginkel & Gilles 2004) and fuel cell systems (Hanke, Mangold & Sundmacher 2005). In the field of biological cellular systems, a very comprehensive model for the growth of the small bacterium *Escherichia coli* on carbohydrates has been developed (Kremling & Gilles 2001, Kremling, Bettenbrock, Laube, Jahreis, Lengeler & Gilles 2001).

Two examples selected from the various applications are presented in the last part of this contribution to illustrate the concepts.

2 Model Structuring Concept

The Network Theory of Chemical and Biological Processes (Gilles 1997) proposes a way to decompose various processes into hierarchical units in a systematic manner. A model of a chemical plant e.g. can be decomposed into models of process units like reactors, storage tanks, and separation units, the elementary modeling entities on the level of process units. This is the level of modularization, standard flowsheet simulators are based on. Further, each process unit model consists of models of thermodynamic phases and therefore can be decomposed into phase models. Models of thermodynamic phases are elementary modeling entities on the level of phases. Finally, a thermodynamic phase consists of storages for mass, energy, and momentum, and therefore can be decomposed further on the level of storages.

A completely analogous hierarchical decomposition can be made for cellular systems. However, in biology the focus is on the storage level. Since the biological phase consists of hundreds of components which interact in a biochemical reaction network, a structuring on the storage level aims at grouping together elements with common physiological tasks.

The idea of the network theory is to describe each hierarchical level by two basic types of elements, *components* and *coupling elements*. Components possess a hold-up for physical quantities like energy, mass, and momentum. Coupling elements describe the interactions and fluxes between components. Examples for components are reactors on the level of process units, thermodynamic phases on the level of phases, and mass storages on the level of storages. Examples for coupling elements are valves on the level of process units, phase boundaries or membranes on the level of phases, and reactive sinks and sources on the level of storages. Components are described by a thermodynamic state or state vector $\boldsymbol{X}$. The state of a component may be changed by fluxes $\boldsymbol{J}$, e.g. fluxes of mass or energy. The general differential equation of a component therefore reads:

$$\frac{\partial \boldsymbol{X}}{\partial t} = \boldsymbol{J}. \qquad (1)$$

The task of the coupling elements is to determine the flux vectors $\boldsymbol{J}$. In accordance with the principles of irreversible thermodynamics, it is assumed that the flux vector is an algebraic function of potential differences or potential gradients. A simple example of a coupling element is the heat flux between two phases which is driven by the temperature difference between the phases. The exchange between components can be visualized by a diagram as shown in Figure 1. The components C_k and C_l pass information on their states to

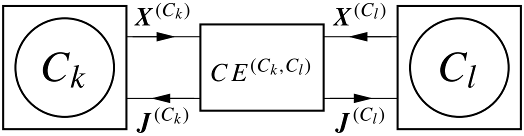


Fig. 1: Connection of two components by a coupling element.

the coupling element $CE^{(C_k,C_l)}$. Depending on those states, the coupling element computes the flux vector and returns the result to the two components. This establishes a bidirectional signal transfer between components and coupling elements.

3 The modeling tool PROMOT

Models of chemical and biological systems that are structured and constructed according to the aforementioned modeling concept, can be implemented in the Process Modeling Tool (Tränkle, Zeitz, Ginkel & Gilles 2000). This tool allows the construction of structured models via a graphical user interface and with a modeling language. The final models are transformed into a differential-algebraic equation-set, that can be analyzed in the simulation environment Diva (Mohl, Spieker, Köhler, Gilles & Zeitz 1997) or in Matlab. In the following section the construction of models in this tool and some aspects of the model processing will be elaborated.

3.1 Modeling elements

In the previous sections, a systematic approach for structuring models of chemical and biological systems was summarized. The components and coupling elements of this approach are implemented as *modules* in PROMOT. As an example we use a model of a regulated metabolic pathway (see Fig. 2) which is modeled on the level of storages. The whole pathway is represented by a module in PROMOT, an encapsulated entity containing a mathematical description of its behavior. This module is composed of submodules that represent storages (ellipses), reactions (squares with arrows) and signal-transformers (rectangles). This module is part of a larger model, describing the regulated carbohydrate uptake of *E.coli* (see 4.1).

The submodules in the figure are instances of other *module-classes*, e.g. the elements `lac` und `allo` are instances of the module-class `storage-intra-x`. The interface of the module is represented by *terminals* (smaller squares on the outer edges of the module). Terminals are connection-points that contain a group of named variables and can be linked with other compatible terminals on the next higher level of the aggregation hierarchy. A terminal can represent flows of mass, energy, momentum or signals. In biological systems, terminals represent mass flows (e.g. the terminal of `lac`) or concentrations of substances acting as signals. Terminals of submodules can be propagated as terminals of the containing module, e.g. the left terminal `a` of `r_lac` is propagated as `t_lace` for the whole lactose module.

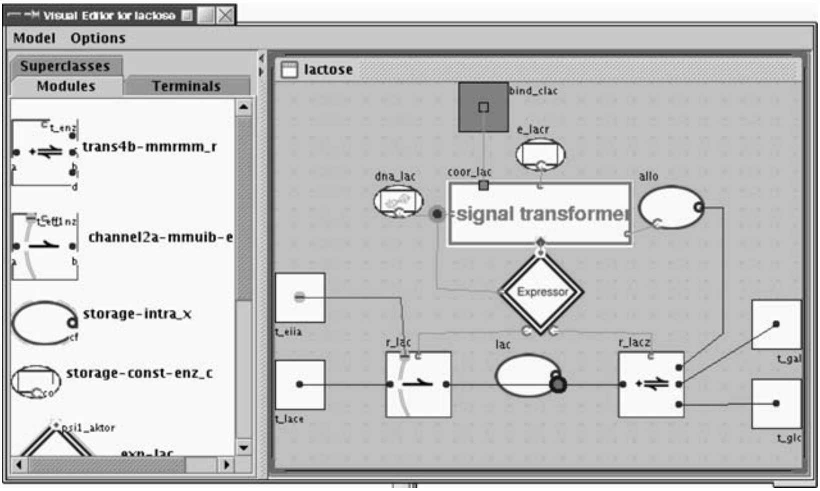
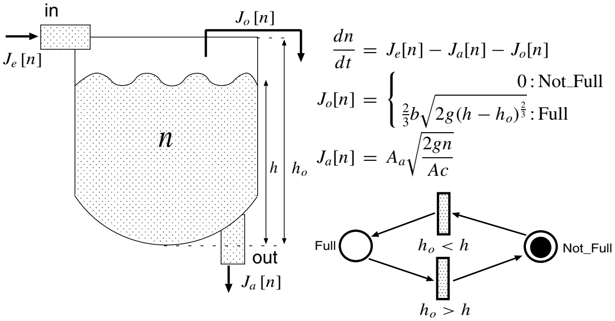


Fig. 2: Module for lactose transport in E.coli.

Users of PROMOT can build *composed modules* like `lactose` by selecting module-classes on the left of the window shown in Fig. 2 and placing them as submodules into the working-area on the right. Submodules can be parameterized with appropriate initial values and parameter settings, and can be connected using their terminals. Two or more compatible terminals can be connected with *links*. Terminals are considered compatible, if each of them contains a set of variables with the same names.

Although PROMOT allows for the construction of models out of encapsulated modules, the modeling scheme is equation-based and the equations are fully transparent. Users can add their own modules with special equations, or extend models from a library with own equations and libraries. In general, modules can contain a linear-implicit differential-algebraic equation set. Important for equations in the field of chemical engineering is the possibility to use arrays of variables, equations and also modules for the efficient modeling of repeated elements in plant models.

When chemical engineering systems are modeled, purely continuous models are often not sufficient, since discontinuities in the model equations as well as discrete controllers have to be described. Therefore PROMOT employs a concept of hybrid modeling which uses an additional petri net to describe the discrete part of the model. The places of the petri net represent discrete model states and the transitions describe changes of the discrete state that can be triggered by changes in the continuous variables. On the other hand, the current state of the petri net (i.e. the marked places) change the equation system locally through conditional equations or by changing characteristic parameter values. As a very simple example, the modeling of a tank with an overflow is shown in Fig. 3. The petri net switches between `Full` and `Not_Full` depending on the current height h of the liquid in the tank. The weir equation for the overflow $J_o\,[n]$ is a conditional equation, since it is only active as long as `Full` is marked and the liquid reaches the overflow.



$$\frac{dn}{dt} = J_e[n] - J_a[n] - J_o[n]$$

$$J_o[n] = \begin{cases} 0 : \text{Not\_Full} \\ \frac{2}{3}b\sqrt{2g(h-h_o)^{\frac{2}{3}}} : \text{Full} \end{cases}$$

$$J_a[n] = A_a\sqrt{\frac{2gn}{Ac}}$$

Fig. 3: Model of a tank with overflow, using a petri net for switching the weir equation.

3.2 Model processing

All modeling entities in PROMOT are organized in a specialization hierarchy with multiple inheritance. The modeler can therefore use object-oriented implementation techniques like abstraction and polymorphism to describe his modules. This becomes especially helpful when implementing general modeling libraries. In this case, general model elements like balance equations and physical property correlations can be implemented in reusable superclasses. For a specific application, general modules can be extended and specialized by deriving subclasses that inherit all general parts and add application specific equations and parameters. Examples for the use of object-oriented libraries are given in the next sections.

The structured and object-oriented view of a model is particularly effective for model formulation and manipulation. On the other hand, the numerical solution can be carried out more efficiently with the plain equations. Therefore all further operations are carried out using a global DAE that is obtained from all local equations from the different modules and the coupling equations generated by links. A further issue is related to the numerical solution: Due to the structured way of modeling that generates coupling relations, and since the elements in model-libraries tend to be implemented in a general way, often using some extra equations, PROMOT models typically contain a large number of simple algebraic equations. Before generating simulation code, PROMOT therefore analyzes and optimizes the system of equations. For this purpose the incidence matrix of the equations is computed and an algorithm proposed by Tarjan (Tarjan 1972) is used to transform this matrix to block-lower-triangular form. Explicit algebraic equations can be identified in this matrix and are symbolically transformed into a sequence of explicit assignments to intermediate variables. These assignments can be calculated in the simulation code without involving the numerical equation solver, which improves simulation performance. In this process of optimization, also repeated calculations of constant expressions and unused variables are removed from the model. The preprocessing step is not only useful for optimizing the efficiency of the numerical solution, but it also unveils structural inconsistencies (e.g. singularities) of the equation system. If such conditions occur, the user is provided with debugging information for detecting the error in the model structure quickly. Finally, the equation-system is transformed into simulation code using the Code-Generator (Köhler, Räumschüssel & Zeitz 1997) for Diva.

4 Applications

In this section, application examples for the structuring methodology and the modeling tool PROMOT are presented. Guided by the hierarchical modeling concept, examples for the formulation on the storage level, and on the level of phases will be discussed.

The example on the level of storages is chosen from biology. The carbohydrate uptake of bacteria is considered. The challenge here is to structure a huge reaction network into smaller functional units in order to understand the various interactions between the reaction steps. In this sense, the biological example has some similarity to chemical engineering systems with complex reaction kinetics like e.g. combustion processes.

The example on the level of phases is an arrangement of thermally coupled fixed bed reactors. Spatially distributed models of thermodynamic phases have to be coupled in this case. The objective of the structuring on the level of phases is in this case to support the design and analysis of a novel integrated process.

4.1 Application Example on the Level of Storages: Nutrient Uptake of a Bacterium

An interesting example for model set-up of a biological system is concerned with the question of nutrient uptake. The control of the carbohydrate uptake in bacteria has been investigated for a long time. Starting with the pioneering work of Monod, a number of components were detected that are responsible for the coordination of sugar uptake. It is widely accepted that the phosphotransferase system (PTS) is one of the important modules in the signal transduction machinery of bacteria. The PTS represents a transport system (in microbiology, a protein that is membranstanding and transports components from the extra cellular environment into the cell is named a transport system), and at the same time it is part of a signal transduction system responsible for carbon catabolite repression (Postma, Lengeler & Jacobson 1993). Catabolite repression means the dominance of one carbohydrate uptake system over another one. If two sugars, e.g. glucose and lactose, are present in the medium, glucose is taken up first while lactose is taken up only after the depletion of glucose. The connections of all sub-module are shown in Figure 4. The lactose pathway is shown in detail in Figure 2. As can be seen, protein EIIA and its phosphorylated form P$\sim$EIIA are the main output signals of the PTS. The output signal ψ from the Crp sub-module describes the transcription efficiency of the genes and operons under control of Crp. The glucose and the lactose pathway are connected to the liquid phase, which is represented by Glc_{ex} and Lac_{ex}. Both pathways feed into the central catabolic pathways. The entire model comprises 30 state variables (Kremling et al. 2001, Kremling & Gilles 2001).

The PTS controls via output EIIA the lactose pathway. EIIA is an inhibitor of the lactose transport protein. Providing both sugars at the very beginning of a batch experiment, glucose is taken up immediately while lactose is taken up after glucose has run out (Figure 5). During the second growth phase, galactose is excreted in large amounts in the medium. The enzyme for splitting intracellular lactose, LacZ, is synthesized first in the second growth phase. Protein EIIA, the output of the PTS, shows an interesting dynamical behavior. After the run out of glucose, EIIA switches very fast to the phosphorylated form and returns slowly back. This is based on the fact, that the PTS is active during glucose uptake in first growth phase and is active also during the sec-

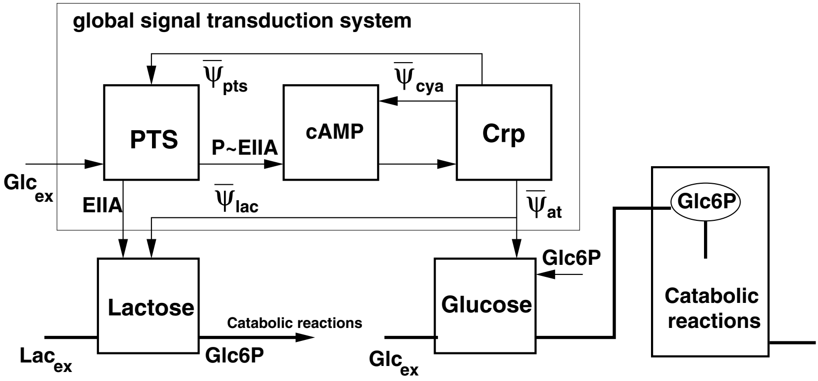


Fig. 4: Survey of all sub-modules of a biological model. The global signal transduction unit comprises the PTS module, the synthesis of the second messenger cAMP, and the binding of the regulator protein Crp to the respective binding sites.

ond growth phase due to the splitting of intracellular lactose into intracellular galactose and glucose. The intracellular glucose is also phosphorylated by the PTS.

The model is available in the PROMOT/Diva environment (Ginkel, Kremling, Nutsch, Rehner & Gilles 2003). Parameters are estimated using a number of experiments with different mixtures of carbon sources and mutant strains. The mutant strains differ only in one single gene. This strategy allows to analyze the influence of different proteins in an isolated way and to estimate parameters from the time courses.

4.2 Application Example on the Level of Phases: Autothermal Fixed Bed Reactor

An example from chemical engineering may serve to illustrate the structuring concept on the level of phases. An autothermal reactor concept is considered. The purpose of an autothermal reactor is to carry out weakly exothermic reactions without providing external heating energy. This can be achieved by integrating heat exchange and chemical reaction in one apparatus. Dynamically operated autothermal reactors make use of the fact that a creeping reaction front in a catalytic fixed bed causes an over-adiabatic temperature rise (Wicke & Vortmeyer 1959). In the well-known reverse-flow reactor (Matros 1989), the creeping reaction front is generated by a periodic flow reversal, i.e. in a forced periodic operation mode. Alternatively, it is also possible to design a reactor in such a way that the creeping reaction fronts and the resulting over-adiabatic temperature rise are created by autonomous periodic oscillations

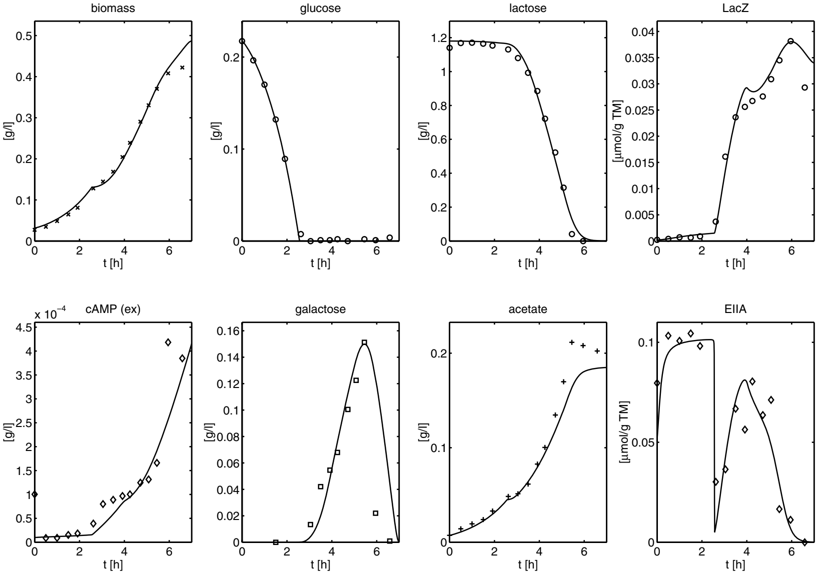


Fig. 5: Time course of simulation results (solid lines) and experimental data (symbols) for a selected experiment with the wild-type strain LJ110. Glucose is taken up immediately while the uptake of lactose is repressed. This is referred as diauxic growth.

without any external forcing (Lauschke & Gilles 1994). A reactor concept of this kind will be modeled in the following.

The reactor is divided into two reactor lines with separate inlets and outlets (see. Figure6). Each reactor line consists of two catalytic fixed beds in series. The first bed is jacketed by a gas channel co-current heat exchanger. The second bed is insulated towards the environment. The reactants enter the heat exchanger section, flow through the insulated section and leave the reactor via the gas channel of the other reactor line. In this way, a thermal feed back is established between the two reactor lines. This feed back can be used to generate circulating reaction fronts in the arrangement: A hot spot caused by a creeping reaction front in one of the reactor lines triggers a new reaction front in the other line when reaching the gas channel. For a simple oxidation reaction of first order, it can be shown that two types of autonomous periodic solutions co-exist under certain operation conditions: a symmetric solution with a creeping reaction front in each of the reactor lines, and an asymmetric solution, where alternately one reactor line contains a reaction front and the other is in an extinguished state (Mangold, Klose & Gilles 2000).

A spatially distributed one-dimensional dynamic model for this process has been implemented in PROMOT. Figure 7 shows the top level structur-

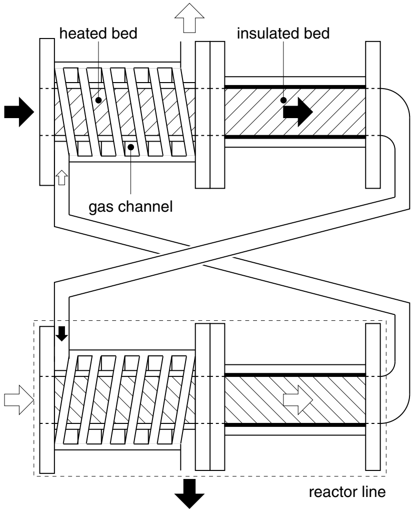


Fig. 6: Autothermal reactor concept: Two catalytic fixed bed reactors coupled thermally by co-current heat exchangers. White and black arrows indicate different mass fluxes

ing of the model. On this level, the two reactor lines are components in the nomenclature of the network theory. They are connected to reservoirs representing the inlet and outlet tanks of the system. The coupling between the two reactor lines is done by coupling elements describing the following internal boundary conditions between the two distributed systems: Continuity is assumed for the compositions and the temperatures on both sides of the boundary. Mass and energy conservation give further conditions for the fluxes across the internal boundary. Further coupling elements are needed in order to define heat fluxes to the reactor walls that enter the boundary conditions for the energy balance of the walls. As can be seen in Figure 7, the connection between two modules is always bidirectional with one signal line passing the information on the state vector and one signal line passing the information on the fluxes.

The models of the two reactor lines can be decomposed into models of interacting thermodynamic phases, as is shown in Figure 8. The spatially distributed phases of the fixed beds, the gas channels and the reactor walls are the components on this level. The coupling elements in Figure 8 define the internal boundary conditions between axially coupled phases on the one hand and the radial heat exchange between fixed bed, gas channel and reactor walls on the other hand. The structuring makes changes of the model very easy. For example, one might want to add mass exchange to the heat ex-

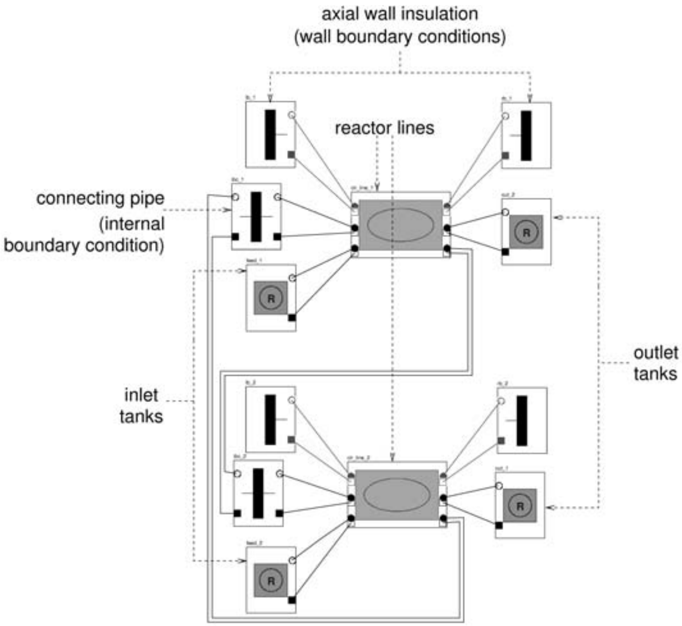


Fig. 7: Structure of the fixed bed reactor model on the top level (from a screenshot of the ProMoT GUI)

change between fixed bed and gas channel. This can be done by replacing the heat exchange coupling elements by membrane modules, as described in a previous publication. The models of the distributed phases consist internally of components and coupling elements on the level of storages, similar to the biological example in the previous section.

A simulation result obtained by the described model is shown in Figure 9. The total oxidation of ethane is considered. Under suitable inlet temperatures, inlet compositions, and flow rates, an autonomous periodic oscillation with creeping reaction fronts develops. At time t_1, the front stretches from the heated bed to the inlet of the insulated bed. At later times $t_2, \ldots, t_4$ the front moves into the insulated bed. At time t_5 the hot spot reaches the gas channel and ignites a new reaction front in the heated bed of the other reactor line. This marks the beginning of a new periodic cycle.

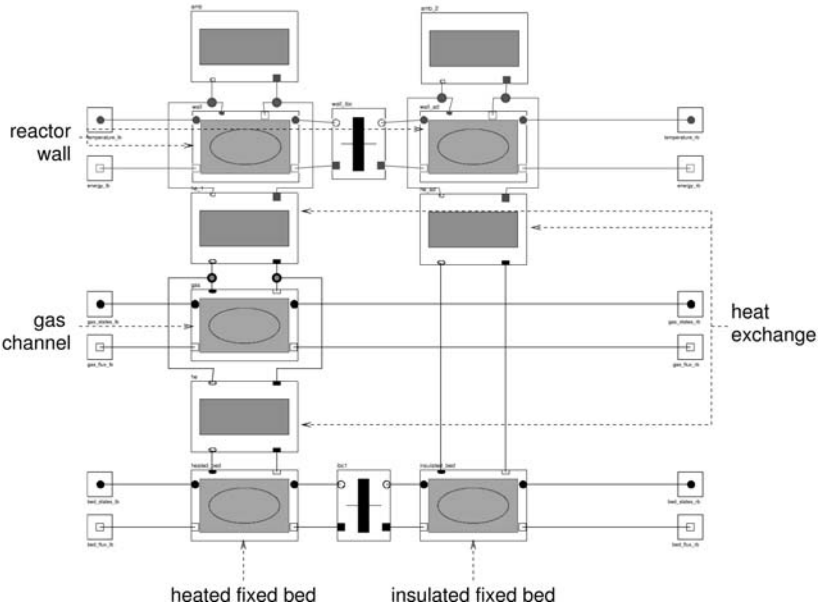


Fig. 8: Structure of the fixed bed reactor model on the level of phases (from a screenshot of the PROMOT *GUI)*

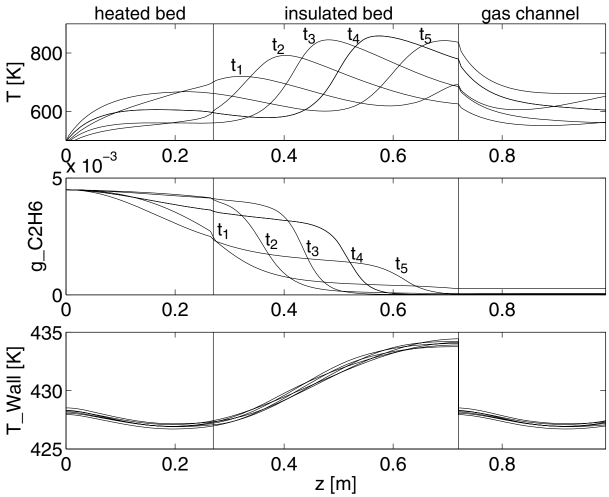


Fig. 9: Spatial profiles of the gas temperature T, the ethane mass fraction $g_{\mathrm{C_2H_6}}$, and the wall temperature T_{Wall} in autonomous periodic operation at time points $t_i, i = 1, \ldots, 5$.

5 Conclusions

Computer assistance during the model development process can accelerate the modeling process and improve the quality of the resulting models. Three key components have been identified for the successful application of computer aided modeling: (i) A structuring methodology is needed that permits a uniform, consistent, and systematic way to formulate different kinds of process models. (ii) A software tool must be available that is able to convert the structured model information into running program code suitable for numerical analysis in some flow-sheet simulator. (iii) A model library must exist that is based on the theoretical concepts of model structuring and that is implemented in the modeling tool. The model library must be comprehensive enough to enable a user to create his own model from pre-defined building blocks.

The work reported in this contribution addresses all of the three fields. The network theory provides a well-developed structuring methodology that is applicable to very different types of process models, as the examples from biology and chemical engineering show. The process modeling tool PROMOT permits a direct realization of the theoretical structuring concept. It is able to handle models of high order and complexity. Using its object oriented modeling language, modeling experts can implement new models very efficiently. Due to its graphical user interface, PROMOT is also a tool that can be used conveniently by non-specialists in the area of process modeling. The feasibility of the structuring approach and of the modeling concept could be demonstrated for quite different applications in the area of systems biology and chemical engineering. Currently, the database of models implemented in PROMOT is increasing, and the software is used more and more in the framework of ongoing research projects.

Future challenges will be extensions of the concepts and the tools to more complicated models, especially distributed systems with multiple dimensions. An example are coupled systems involving fluid dynamics and property coordinates, as they become more and more important for biological and chemical engineering applications.

Acknowledgements

The financial support of Deutsche Forschungsgemeinschaft in the framework of joint research projects FOR 447 and SFB 412 is gratefully acknowledged.

References

Gilles, E. (1997). Netzwerktheorie verfahrenstechnischer Prozesse, *Chemie-Ingenieur-Technik* **69**: 1053–1065.

Ginkel, M., Kremling, A., Nutsch, T., Rehner, R. & Gilles, E. (2003). Modular modeling of cellular systems with promot/diva, *Bioinformatics* **19**: 1169–1176.

Hanke, R., Mangold, M. & Sundmacher, K. (2005). Application of hierarchical process modelling strategies to fuel cell systems – towards a virtual fuel cell laboratory, *Fuel Cells* **5**(1): 133–147.

Köhler, R., Räumschüssel, S. & Zeitz, M. (1997). Code Generator for Implementing Differential Algebraic Models Used in the Process Simulation Tool DIVA, *Proc. 15th IMACS World Congr.*, Berlin, pp. 621–626.

Kremling, A., Bettenbrock, K., Laube, B., Jahreis, K., Lengeler, J. & Gilles, E. (2001). The organization of metabolic reaction networks: III. Application for diauxic growth on glucose and lactose, *Metab. Eng.* **3**(4): 362–379.

Kremling, A. & Gilles, E. (2001). The organization of metabolic reaction networks: II. Signal processing in hierarchical structured functional units, *Metab. Eng.* **3**(2): 138–150.

Lauschke, G. & Gilles, E. (1994). Circulating reaction zones in a packed-bed loop reactor, *Chemical Engineering Science* **49**: 5359–5375.

Mangold, M., Angeles-Palacios, O., Ginkel, M., Kremling, A., Waschler, R., Kienle, A. & Gilles, E. (2005). Computer aided modeling of chemical and biological systems – methods, tools, and applications, *Industrial & Engineering Chemistry Research* **44**: 2579–2591.

Mangold, M., Ginkel, M. & Gilles, E. (2004). A model library for membrane reactors implemented in the process modelling tool ProMoT, *Computers & Chemical Engineering* **28**: 319–332.

Mangold, M., Klose, F. & Gilles, E. (2000). Dynamic behavior of a counter-current fixed-bed reactor with sustained oscillations, *in* S. Pierucci (ed.), *European Symposium on Computer Aided Process Engineering - ESCAPE-10*, Elsevier, pp. 205–210.

Matros, Y. (1989). *Catalytic Processes under Unsteady-State Conditions*, Elsevier, Amsterdam.

Mohl, K. D., Spieker, A., Köhler, R., Gilles, E. D. & Zeitz, M. (1997). DIVA - A simulation environment for chemical engineering applications, *ICCS Collect. Vol. Sci. Pap.*, Donetsk State Techn. University, Ukraine, pp. 8–15.

Postma, P. W., Lengeler, J. W. & Jacobson, G. R. (1993). Phosphoenolpyruvate: Carbohydrate phosphotransferase systems of bacteria, *Microbiological Reviews* **57**(3): 543–594.

Tarjan, R. (1972). Depth first search and linear graph algorithms, *SIAM J. Comptg.* **1**: 146–160.

Tränkle, F., Kienle, A., Mohl, K., Zeitz, M. & Gilles, E. (1999). Object-oriented modeling of distillation processes, *Comp. Chem. Engng. (Suppl.)* **23**: S743–S746.

Tränkle, F., Zeitz, M., Ginkel, M. & Gilles, E. (2000). ProMot: a modeling tool for chemical processes, *Mathematical and Computer Modelling of Dynamical Systems* **6**: 283–307.

Waschler, R., Angeles-Palacios, O., Ginkel, M. & Kienle, A. (2003). Application of the process modeling tool promot to large scale chemical engineering processes, *Proceedings of the 4th MATHMOD-4th IMACS Symposium on Mathematical Modelling*, IMACS, ARGESIM, Vienna, pp. 1113–1121.

Wicke, E. & Vortmeyer, D. (1959). Zündzonen heterogener Reaktionen in gasdurchströmten Körnerschichten, *Zeitschrift für Elektrochemie* **63**: 145–152.

The General Hidden Markov Model Library: Analyzing Systems with Unobservable States

Alexander Schliep[1]
Benjamin Georgi[1]
Wasinee Rungsarityotin[1]
Ivan G. Costa[1]
Alexander Schönhuth[2]
[1] Max Planck Institute for Molecular Genetics, Berlin
[2] ZAIK, University of Cologne

Abstract

Hidden Markov Models (HMM) are a class of statistical models which are used in a broad variety of disciplines for problems as diverse as understanding speech to finding genes which are implicated in causing cancer. Adaption for different problems is done by designing the models and, if necessary, extending the formalism. The General Hidden Markov Model (GHMM) C-library provides production-quality implementations of basic and advanced aspects of HMMs. The architecture is build around a software library, adding wrappers for using the library interactively from the languages Python and R and applications with graphical user interfaces for specific analysis and modeling tasks. We have found, that the GHMM can drastically reduce the effort for tackling novel research questions. We focus on the Graphical Query Language (GQL) application for analyzing experiments which measure the expression (or mRNA) levels of many genes simultaneously over time. Our approach, combining HMMs in a statistical mixture model, using partially supervised learning as the paradigm for training results in a highly effective, robust analysis tool for finding groups of genes sharing the same pattern of expression over time, even in the presence of high levels of noise. Software available from http://ghmm.org

1 Introduction

When we set out to analyze experimental mass data we have to ask ourselves what is the nature of the underlying system generating the data — what is the physical, chemical or biological process we are investigating and what is the, usually, limited and imperfect view provided by our experimental instruments. Often the processes will be stochastic; moreover, more often than not the experimental procedure will introduce another source of stochasticity.

Provided that they are sufficiently complex, the systems we are investigating will share one commonality: the observations will be influenced by the state the system is in. Here state very broadly encompasses the values of all variables which describe he system. However, the state itself will be unobservable. This structure can for example be found in understanding speech, where words correspond to states and the waveforms of the spoken word are observed, or in finding genes in a DNA sequence, where the various structural elements of a gene (e.g., exons, introns, splice sites, start and stop codons) are the states and we observe the sequence of nucleotides.

For a multitude of analysis tasks — finding the most likely state given observations, classifying observations into different groups — a stochastic model offering effective computation of such processes with hidden states would be beneficial. If we make the assumptions, that

- the observations only depend on the state the system is in at the time of the observation and thus are independent on prior observations as well as prior states, and
- the probability of a change from a state at time t to another state at time $t+1$ only depends on the state at time t

then the so-called Hidden Markov Models (HMM) provide an effective model class. In essence they combine two stochastic processes. Consider a stochastic process on the discrete and finite set of states S (extensions to continuous state spaces are routine), i.e. a sequence of random variables

$$\{X_t\}_{t\geq 1},\ \ X_t \in S$$

such that the so called Markov property holds:

$$P[X_{t+1} = s_{t+1} | X_t = s_t, \ldots, X_1 = s_1] = P[X_{t+1} = s_{t+1} | X_t = s_t].$$

If we have a stochastic function $f : S \mapsto R(\Sigma)$, where $R(\Sigma)$ denotes the set of random variables with outcomes Σ, such that for all $i \in \{1, \ldots, N\}$, $f(i)$ is a random variable taking on values from Σ then

$$\{Y_t\}_{t\geq 1},\ \ Y_t := f(X_t)$$

is a HMM.

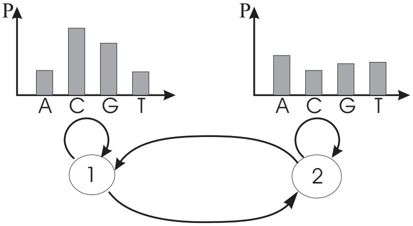


Fig. 1: A discrete HMM. The underlying Markov process is depicted as a directed, weighted graph, the states correspond to vertices and transitions to edges (edge weights correspond to transition probabilities, not shown) . The emissions in each state are displayed as discrete distributions over the alphabet of the four nucleotides. This model can be used to analyze segmentation of DNA based on nucleotide usage differences between the segments.

Hidden Markov Models in fact allow a wide range of variations with respect to emissions, they can be discrete, continuous, or vector-valued, the densities controlling the random variables f — for continuous emissions mixtures of Gaussian are routinely used — and the details of the Markov process controlling the sequence of states.

The first publications on HMMs stem from the 1940s, but they have not found wide-spread use until the 1970s when their effectiveness in modeling speech became obvious in the push to implement speaker-independent speech recognition at AT&T. Today HMMs form the basis for a wide range of solutions for data analysis and statistical modeling, from areas such as guiding missiles (Schrodt 1998), predicting crises in the Middle East (Nilubol, Pham, Mersereau & Smith 1998) or finding genes in human DNA sequence (Kulp, Haussler, Reese & Eeckman 1996, Yada & Hirosawa 1996). While many applications can be addressed with standard HMMs, often extensions to the basic method are required.

Our main contribution is two-fold. On one hand we have implemented our software in the highly reusable, general library GHMM— licensed under the Library GNU Public License (LGPL). We implemented the standard algorithms for computing with HMMs and a large number of extensions, both to the model class and algorithms. So-called wrappers allow the use of GHMM from interactive languages such as Python and R and a graphical application is provided for editing HMMs. The GHMM thus creates a comprehensive, flexible framework which substantially reduces the effort for implementing novel data analysis and modeling solutions. The architecture allows interactive use, incorporation into other software package, and — due to the licensing chosen — extension of the core functionality. Altogether, these aspects lead to a considerable speed-up of research efforts using HMMs.

On the other hand we employ the HMM framework and the GHMM in particular to design a novel mixture of HMMs which performs very well for identifying groups of genes in gene expression time-courses, due to the ability to use more prior knowledge than competing approaches and a large degree of flexibility in modeling the qualitative behavior of time courses.

In the following we will describe our work on analysis of gene expression time-courses in detail, list further application of the GHMM, and expand about the architecture of the GHMM.

2 Analysis of gene expression time course data using mixtures of Hidden Markov Models

Microarray experiments have become a staple in the experimental repertoire of molecular genetics. They can be used to detect or even quantify the presence of specific pieces of RNA in a sample. The experimental procedure is based on hybridization of these RNA-sequences to either oligonucleotide or cDNA probes which are affixed to the array. By using a probe for each gene in a cell, microarray experiments can measure the expression levels of up to thousands of genes simultaneously. The resulting so-called expression profiles allow for example investigation of differences in distinct tissue types or between healthy or diseased tissues. When microarray experiments are performed consecutively in time we call this experimental setting a time course of gene expression profiles. The questions this experimental setting tries to address are the detection of the cellular processes underlying the regulatory effects observed, inference of regulatory networks and, in the end, assigning function to the genes analyzed in the time courses.

Because of the large number of genes and their complex relationships in microarray measurements, it has been a standard procedure to identify groups of genes with similar temporal regulatory patterns or *time-courses*. When analyzing such gene expression time-courses a number of problems should be addressed.

- Noise is omnipresent and of manifold nature. We need a good statistical model to deal with it.
- Sometimes prior knowledge in form of high quality annotations regarding regulation or function of the inspected genes is available. The method should thus allow for the integration of readily labeled data.
- The computer aided analysis of gene expression experiments is an experimental help for the biologist. This means that the tool should allow for a high degree of interactivity and visualization.
- Very often only a small number of genes is decisively involved in the processes of interest. The procedure should thus be able to output only few

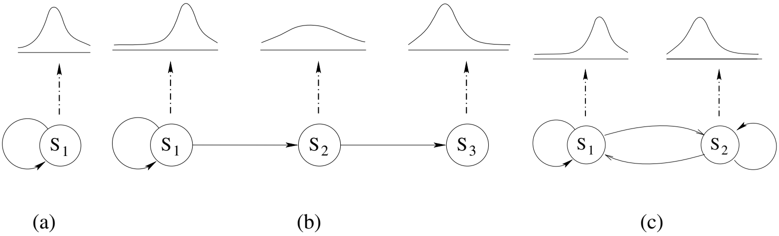


Fig. 2: A number of prototype HMMs encoding distinct qualitative time-course behavior: constant (a), up-down-up (b), cyclic up-down (c).

genes having highly significant relationships if required.

- Genes can trigger the expression of other genes. So gene expression profiles which are to be grouped together may exhibit similar expression patterns showing up at different times.
- Sometimes experiments are (partially) corrupted leading to missing data.
- Genes may interact with other genes in more than only one context. This precludes partitioning the data.
- Gene expression profiles are the result of a time course experiment. Methods which take care of these so called *horizontal dependencies* should outperform those which do not.

Prior approaches can be divided into two classes, depending on whether they are based on statistical models or not. Methods in the second class require the definition of a distance measure describing the degree of similarity between two gene expression profiles. Note that these methods mostly do not explicitly account for the high levels of noise in the data. Moreover they do not allow for the inherent nature of the data, namely being time courses. Examples are hierarchical (Eisen, Spellman, Brown & Botstein 1998, Gasch, Spellman, Kao, Carmel-Harel, Eisen, Storz, Botstein & Brown 2000) and k-means clustering (Tavazoie, Hughes, Campbell, Cho & Church 1999) or singular value decomposition (Rifkin & Kim 2002). Some of them provide graphical user interfaces, some do not. None of these methods allows for integration of prior knowledge.

Methods of the first class use statistical models to represent clusters. Cluster membership is decided based on maximizing the likelihood of data points given the cluster models and the assignment of data points to clusters. Model based procedures account for the horizontal dependencies in the data. Moreover one can expect a larger robustness with respect to noise as it mostly is explicitly modeled in these approaches. Examples of model-based clustering used for analysis of expression time courses are based on cubic splines (Bar-

Joseph, Gerber, Gifford & Jaakkola 2002) and autoregressive curves (Ramoni, Sebastiani & Kohane 2002, Ramoni, Sebastiani & Cohen 2002).

2.1 Mixtures of Hidden Markov Models

To cope with the issues given, we model a set of gene expression time-courses as a mixture model. The basic assumption of a mixture model is that the data has been generated by a weighted superposition of model components coming from the same model class but differing in their parameters and their weights. Besides from providing a "soft" assignment of time courses to clusters mixture models also have proved to be more robust with respect to noise when learned from data. The individual components we use are HMMs, mainly due to their flexibility in encoding 'grammatical' constraints of time-courses. Their graphical structure benefits the analysis process, as it affords a high degree of interactivity and accessibility.

Simple model for time-courses

We use HMMs (see (Rabiner 1989) for an excellent introduction) with continuous emissions governed by a normal distribution in each state. The HMM topology — the number of states, the set of possible transitions — is essentially a linear chain (following (Schliep, Schönhuth & Steinhoff 2003), see Fig. 2), neglecting a possible transition from the last to the first state to accommodate cyclic behavior. The states reflect regions of a time-course with *similar* levels of expression. There are usually fewer states than time-points, as several similar successive measurements will be accounted for by the same state by making use of its self-transition. It is important to point out that our approach is not limited to such models but rather accommodates arbitrary HMM topologies.

We deal with missing values in the following way. Each state of an HMM can either emit a real-valued variate according to its Gaussian state emission probability density function (pdf) or, with a low probability equal to the proportion of missing values in all the time-courses, a special missing symbol.

Learning Mixtures

We combine K of such HMMs $\lambda_1, \ldots, \lambda_K$ to a pdf for a gene expression time-course by use of a convex combination of the K component probability density functions induced by the HMMs, denoted $p_j(\cdot, \lambda_j)$. The mixture pdf is parameterized by

$$\Theta = (\lambda_1, \ldots, \lambda_K, (\alpha_1, \ldots, \alpha_K))$$

and defined as

$$p(\cdot|\Theta) := \sum_{j=1}^{K} \alpha_j p_j(\cdot, \lambda_j).$$

As the former is just a usual mixture (McLachlan & Basford 1988, McLachlan & Peel 2000), the well-known theory applies. The resulting likelihood function can be optimized with the EM-algorithm (Dempster, Laird & Rubin 1977, Wu 1983, Boyles 1983, Bilmes 1998).

We additionally propose to use *labeled* data by extending the EM algorithm to gain from prior knowledge. We show that there is a large improvement in convergence to *good* local optima on typical data, even if only small amounts of labeled data are supplied.

To apply the EM-algorithm one assumes the existence of unobservable (or hidden) data $Y = \{y_i\}$, which indicates which component has produced each O^i in the set of time-courses $\mathcal{O}$. Thus, we can formulate a complete-data log-likelihood function $\log L(\Theta|\mathcal{O}, Y)$.

If we are given additional labeled time-courses, we do not have to guess the corresponding y_i. We denote the set of labeled time-courses with $\mathcal{O}_L$ and the set of unlabeled ones with $\mathcal{O}_U$. For a time-course O^i from $\mathcal{O}_L$ we set the value of y_i to its component label l_i and maintain this assignment throughout the running time by setting $\mathbb{P}[\lambda_j|O^i] = 1$ for $j = l_i$ and zero else. The Θ^t are the estimates for the maximum likelihood in the t-th iteration), which splits into two sums,

$$Q(\Theta, \Theta^t) := \sum_{O^i \in \mathcal{O}_L} \log\left(\alpha_{l_i} p_{l_i}(O^i|\lambda_{l_i})\right) + \\ \textstyle\sum_{O^i \in \mathcal{O}_U} \sum_{j=1}^{K} \log\left(\alpha_j p_j(O^i|\lambda_j)\right) \mathbb{P}[j|\Theta^t, O^i],$$

and for which the usual local convergence result holds.

Inferring Groups

The simplest way of inferring groups in the data, is to interpret the mixture components as clusters and assign each time-course to the cluster which maximizes the probability of the cluster given the time-course O, $\mathbb{P}[\lambda_j|O]$. However, a mixture encodes much more information. Inspection of the posterior distribution $d(O) := \{\mathbb{P}[\lambda_i|O]\}_{1 \leq i \leq K}$ reveals the level of ambiguity in making the assignment, which can be quantified easily and sensibly by computing the entropy $\mathbb{H}(d(O))$. Choosing a threshold on the entropy yields a grouping of the data into $K + 1$ groups, one group containing all profiles showing no significant membership to one of the components.

Groups will typically contain time-courses having the same qualitative behavior. The time at which, for example, an up-regulation occurs will often

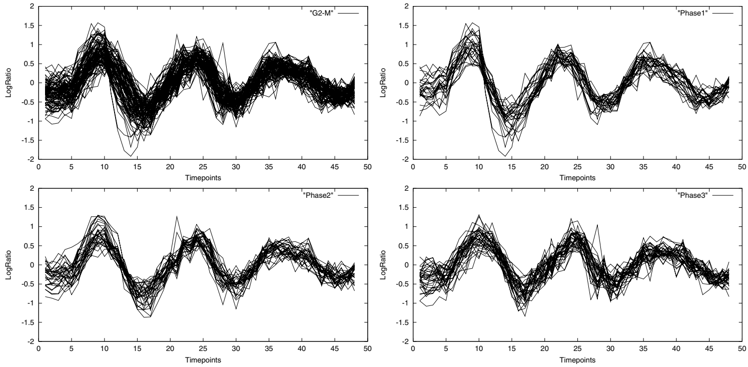


Fig. 3: A group obtained by computing a mixture model using nine labeled and 2263 unlabeled time-courses from the Whitfield data set (top left). It contains five of the labeled time courses. The group was decomposed, using the Viterbi decomposition, into three subgroups, corresponding to synchronous genes, resulting in a first subgroup containing mainly G2 genes (bottom left, phase 1), the second having G2 as well as G2/M genes (top right, phase 2) and the third having mostly G2/M genes (bottom right, phase 3).

Tab. 1: Results on the simulated data set `SIM` *for k-means clustering, CAGED, Splines , and HMM Mixtures with no, 0.9% (five per class) and 1.7% (ten per class) labeled time-courses per class. By comparing the known classes in* `SIM` *with the computed clustering for all pairs of time-courses we computed true and false positives as well as true and false negatives, abbreviated TP, FP, TN and FN. True positive is defined as a pair of time-courses with equal class which are assigned to the same cluster. To quantify the performance we computed the standard sensitivity, $\frac{\#TP}{\#TP+\#FN}$, and specificity, $\frac{\#TP}{\#TP+\#FP}$.*

plus 2pt minus 1pt

Method	**Description**	**Specificity**	**Sensitivity**
M1	k-means, Euclidean distance	85.55%	71.87%
M2	CAGED	41.00%	99.70%
M3	Splines	47.29%	39.38%
M4	HMM Mixtures	93.00%	79.14%
M5	HMM Mixtures, 0.9% labeled time-courses	96.40%	96.90%
M6	HMM Mixtures, 1.7% labeled time-courses	96.60%	96.99%

vary. Synchronous subgroups of such clusters are found with the Viterbi-decomposition introduced in (Schliep et al. 2003, Schliep, Costa, Steinhoff & Schönhuth 2005).

2.2 Results

We used published data from a time-course experiment (Whitfield, Sherlock, Saldanha, Murray, Ball, Alexander, Matese, Perou, Hurt, Brown &

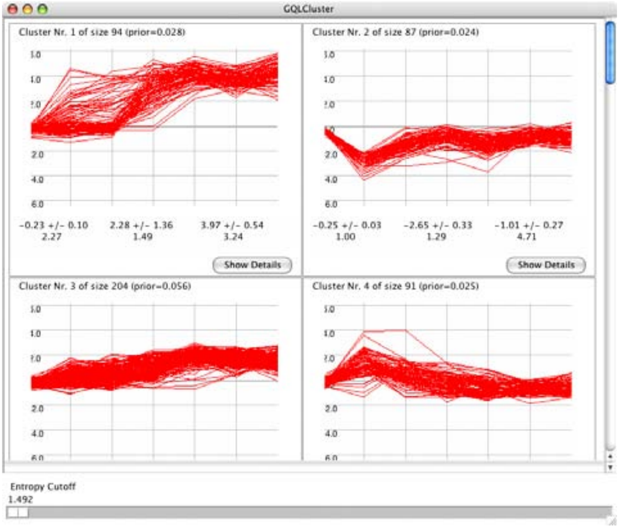
Botstein 2002), in which the authors measured genome wide gene expression of synchronized HeLa (cervical cancer cells) cells. Goal of the experiment was the detection of genes regulating cell cycle. One cycle can be divided into five phases each of which representing a section of life of a eukaryotic cell. Genes, which are involved in the regulation of the cell cycle, are further classified according to their regulation levels in different phases. The data was pre-processed by extracting all those genes with an absolute fold change of at least two in at least one time point. This resulted in a data set containing 2272 expression time courses.

The method was run using a collection of 35 random linear 24-state models. We used five G2/M phase genes described above as a seed for one cluster and four genes of the G1/S phase for a second one. We inferred two groups containing the labeled time-courses of size 91 and 14 respectively, see Fig. 3. We computed a Viterbi-decomposition of the larger group thus finding three subgroups, one containing only G2/M, the second containing G2/M and G2 genes and the third containing only G2 genes. The second cluster, not shown, contained twelve G1/S and two S-phase genes. All time-courses that are assigned to the different phases of our G2, G2/M phase cluster are known to be cell cycle regulated in their respective phase (Whitfield et al. 2002). The same holds for the G1/S, S phase cluster. Thus, the modest amount of prior information used resulted in highly specific (sub-)groups of synchronously expressed genes.

Simulated Data

To facilitate benchmarking and evaluation we designed a method for creating simulated data sets, which makes very mild assumptions about the nature of the data but reflects the realities of microarray experiments. Our proposed approach is *independent* from the underlying assumptions and peculiarities of the statistical model in our method, as it is independent from the assumptions in other methods.

As shown in Table. 1, two of the more involved methods, Caged (Ramoni, Sebastiani & Kohane 2002) and the Spline based clustering by (Bar-Joseph et al. 2002) only reach a specificity of less than 50% (see (Schliep, Steinhoff & Schönhuth 2004) for details). The main error made by Caged in deciding on too few clusters (this cannot be controlled by the user) which leads to merging of several classes (C1 and C2 respectively C3-C6) into one cluster. The HMM mixture perform quite well, achieving a high degree of over 90% specificity and over 75% sensitivity. The tests also show very clearly the impressive effect of partially supervised learning. It suffices to have labels for thirty, or less than one percent of all time-courses (cf. M5 in Table. 1), to obtain a specificity and sensitivity exceeding 95%. More labels do not yield further significant improvements.



Entropy Cutoff
1.492

Fig. 4: The method is implemented in a GUI-application written in Python using the highly portable Tk widget set. The mixture estimation is also written in Python, as the function calling overhead is negligible and all computationally intense work is handled by the GHMM respectively by the Numeric package for Python.

3 Further applications

The GHMM is in use in a wide range of research, thesis and industrial projects. The fields include computational finance (liquidity analysis), physiology (analysis of EEG data), computational linguistics and astronomy (classifying stars). Projects in our group mostly address problems from molecular biology, for example finding genes, assigning function to proteins, and discovering hierarchical groups in protein space.

In the following we will briefly introduce a project currently under research, which is typical in the sense that they would not have been started without the library supplying most of the necessary functionality.

3.1 Gene Expression and Chromosomal Proximity

Finding genetic causes for such serious and prevalent diseases as cancer is a very important and very difficult task. Usually, there will not be a single locus variation causing the disease but rather a combination of factors. One of the factors are chromosomal aberrations which do change the levels of expression of genes in close positional proximity on the chromosome. The problem is to identify groups of genes from contiguous regions which exhibit the same, possibly weak difference in expression when one compares cells from healthy and diseased samples. Using the positional information in addition to the differential expression should yield superior results.

An HMM with states for same, higher and lower levels of expression when comparing healthy and in diseased tissues is an effective model for the sequence (with respect to position) of observations. The positional effect can be explicitly modeled by using a *non-homogeneous* Markov chain. The probability of seeing a higher level of expression in gene $i + 1$, given that gene i shows a higher level should decrease with the distance between genes i and $i + 1$ on the chromosome. This extension to standard HMMs is already part of the GHMM and this application is the focus of a research project at our institute.

4 Software

At the core of the GHMM, see Fig. 6 is the GHMM C-library which provides efficient implementations of many HMM variants and their relevant algorithms:

- *Observations:* Discrete, continuous, vector valued, ”silent” emissions, observations conditioned on previous observations (higher-order states)
- *Observation densities:* Discrete, uni-variate Normal, truncated uni-variate Normals, mixture of (truncated) uni-variates Normals, multi-variate Normals
- *Markov chain:* Discrete state space, time-homogeneous, time-inhomogeneous (discrete classes)
- *Training:* Expectation-Maximization, Gradient Descent, discriminative learning
- *Probabilities:* Likelihoods, many marginals
- *Decoding:* Viterbi, 1-best, posterior

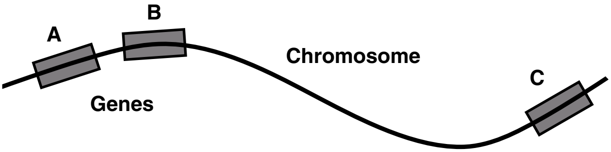


Fig. 5: We compare the levels of gene expression for complete chromosomes in healthy and in diseased tissues. The position of the genes on the chromosome are known. If chromosomal aberrations are causing disease, then one would expect a higher chance of observing differences in expressions if the genes are close. For example, genes A and B should exhibit a positional effect, whereas B and C can be treated as independent.

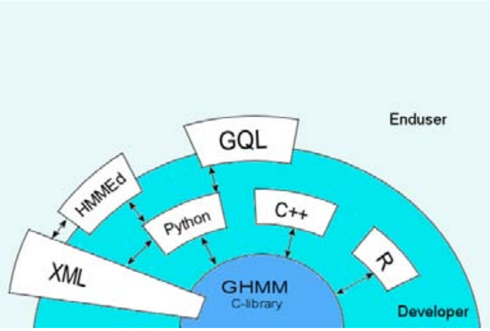


Fig. 6: The GHMM core library is written in C. We provide wrappers to use it from C++, Python and R. Additional packages use and extend GHMM through provision of graphical user interfaces and additional computational capabilities.

Wrappers — pieces of code which allow easy, native access to C-libraries from Python, C++ and R — provide one possible interface. This allows to develop applications using the GHMM, such as the GQL (Costa, Schonhuth & Schliep 2005), in either language. Moreover, the Python and R wrappers allow interactive use of the GHMM from the command line. The HMMEd editor adds a powerful graphical user interface for the design and modification of HMMs. The different layers are linked by our HMMXML, implementing XML input and output as an ubiquitous format for both models and sequences.

Our design choice assumes a novel type of computational scientist as the user of our software, the "scripter". In statistics and numerical analysis software packages such as S, R, and Matlab respectively have become widely popular for ease of use and the little loss of computational efficiency when one compares a script (a short program of high-level commands) with an all-out Fortran or C implementation for the same computational problem. The GHMM follows this models and thus combines full flexibility (everything is programmable) with efficiency (everything computationally expensive is coded in C) without locking out potential users, as it is often the case with problem-specific GUI applications [1]. Some of the training algorithms are implemented using threads to use micro-parallelism on multi-processor shared memory computers. In applications such as GQL, a Python interface to the standard MPI (Message Passing Interface) library allows the use of distributed computing resources from user code.

[1] Point in case: the GHMM was originally started because HMM software used for speech recognition implemented all the necessary algorithms, but not in an accessible form; licensing was another issue.

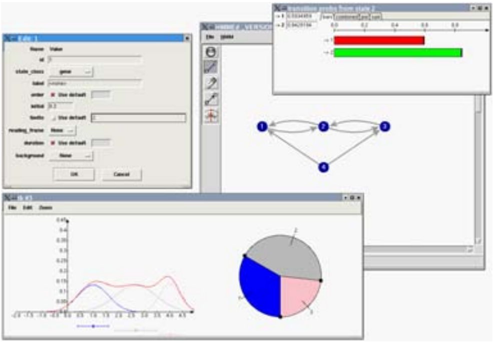


Fig. 7: The HMM editor HMMEd allows graphical design and editing.

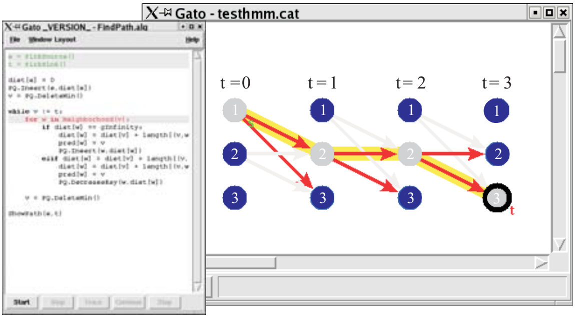
4.1 Supporting Teaching

There are two target audiences of learners which we try to address with the GHMM used as a teaching tool. First, students (and scientists) from the application side of things who are interested in building custom applications for their particular statistical modeling or data analysis problem. They are able to use the GHMM from an interactive, high-level language aided by the graphical user interface for editing models. The following real example reads an HMM build for searching for a particular transcription factor binding site, the nucleotide sequence of Human chromosome 16 and through computation of the Viterbi path finds putative binding sites.

```
>>> m = HMMOpen("trans-fac-13-hmm.xml")
>>> s = FastaOpen("human-chr16.fa")
>>> v = m.viterbi(s)
>>> print "There are %d putative binding sites" % hits(v)
```

It has been our experience that post-Bachelor students were able to implement a simplified variant of our method for analyzing gene expression time-courses, cf. Sec. 2, within one day of a week-long full-day course.

The second group we target are tool developers: students who need to acquire an in-depth understanding of the underlying mathematics to implement variants or extensions of the core algorithms. On one hand they benefit from the interactive access outlined above during learning, development and testing. On the other hand, we supply semi-automatically generated animations of algorithms using the software GATO (Schliep & Hochstättler 2002) which provide visual feedback, see Fig. 8.



```
s = PickSource()
t = PickSink()

dist[s] = 0
PQ.Insert(s,dist[s])
v = PQ.DeleteMin()

while v != t:
    for w in Neighborhood(v):
        if dist[w] == gInfinity:
            dist[w] = dist[v] + length[(v,w
            pred[w] = v
            PQ.Insert(w,dist[w])
        elif dist[w] > dist[v] + length[(v,
            dist[w] = dist[v] + length[(v,w
            pred[w] = v
            PQ.DecreaseKey(w,dist[w])

    v = PQ.DeleteMin()

ShowPath(s,t)
```

Fig. 8: Visualization of the Viterbi algorithm for the best alignment of the observation sequence to the HMM. Based on our graph algorithm animation framework Gato , we animate the Viterbi algorithm as the shortest path problem in a weighted graph without cycle and non-negative edge weights.

5 Summary

The GHMM library provides an essential contribution to scientific computing for a widely applicable class of statistical models, namely Hidden Markov Models. Our design and license choices allow effective use from many languages and for different roles of users: the data-analyst, the scripter and the application developer. Driven by the novel approach of modeling biological time-course data with a mixture of HMMs we leveraged our implementation effort into a much more usable end result, with a wider range of applications and a larger, more diverse user base.

Acknowledgements

The GHMM library has been originally conceived at the Zentrum für angewandte Informatik (ZAIK) at the University of Cologne. Thanks to Bernhard Knab, Bernd Wichern, Barthel Steckemetz for the original implementation (not publicly available), Achim Gädke, Peter Pipenbacher and Disa Thorarinsdottir for a lot of help with the creation of the first public release. Thanks to students from FU Berlin and University of Cologne who used the GHMM in their practical software courses ("Softwarepraktika") in other classes and for their thesis research and all our users for encouragement, bug reports, and code contributions. This work was partially funded by grants from the DAAD/CNPq (Brazil) and the BMBF through the Cologne University Bioinformatics Center (CUBIC).

References

Bar-Joseph, Z., Gerber, G., Gifford, D. K. & Jaakkola, T. S. (2002). A new approach to analyzing gene expression time series data., *6th Annual Int. Conf. on Research in Comp. Molecular Biology* .

Bilmes, J. A. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models, *Technical Report TR-97-021*, International Computer Science Institute, Berkeley, CA.

Boyles, R. (1983). On the convergence of the EM algorithm., *JRSS B* pp. 47–50.

Costa, I. G., Schonhuth, A. & Schliep, A. (2005). The Graphical Query Language: a tool for analysis of gene expression time-courses, *Bioinformatics* **21**(10): 2544–2545.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm., *JRSSB* **39**: 1–38.

Eisen, M., Spellman, P., Brown, P. & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns., *Proc Natl Acad Sci U S A.* **95**: 14863–8.

Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D. & Brown, P. (2000). Genomic expression programs in the response of yeast cells to environmental changes., *Mol Biol Cell.* **11**: 4241–57.

Kulp, D., Haussler, D., Reese, M. G. & Eeckman, F. H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA, *in* D. J. States, P. Agarwal, T. Gaasterland, L. Hunter & R. Smith (eds), *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, pp. 134–142.

McLachlan, G. & Basford, K. (1988). *Mixture Models: Inference and Applications to Clustering*, Marcel Dekker, Inc., New York, Basel.

McLachlan, G. & Peel, D. (2000). *Finite Mixture Models.*, Wiley Series in Probability and Statistics., Wiley, New York.

Nilubol, C., Pham, Q. H., Mersereau, R. M. & Smith, M. J. T. (1998). Translational and rotational invariant hidden markov models for automatic target recognition, *Proc. Of the SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition VI.*

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE* **77**(2): 257–285.

Ramoni, M. F., Sebastiani, P. & Cohen, P. R. (2002). Bayesian clustering by dynamics., *Mach. Learn.* **47**(1): 91–121.

Ramoni, M. F., Sebastiani, P. & Kohane, I. S. (2002). Cluster analysis of gene expression dynamics., *Proc Natl Acad Sci U S A* **99**(14): 9121–9126.

Rifkin, S. A. & Kim, J. (2002). Geometry of gene expression dynamics., *Bioinformatics* **18**(9): 1176–83.

Schliep, A., Costa, I. G., Steinhoff, C. & Schönhuth, A. (2005). Analyzing gene expression time course data., *IEEE\ACM Transactions on Computational Biology and Bioinformatics* . in print.

Schliep, A. & Hochstättler, W. (2002). Developing Gato and CATBox with Python: Teaching graph algorithms through visualization and experimentation, *Multimedia Tools for Communicating Mathematics*, Springer-Verlag, Berlin, Heidelberg, pp. 291–310.

Schliep, A., Schönhuth, A. & Steinhoff, C. (2003). Using Hidden Markov Models to analyze gene expression time course data., *Bioinformatics* **19 Suppl 1**: I255–I263.

Schliep, A., Steinhoff, C. & Schönhuth, A. (2004). Robust inference of groups in gene expression time-courses using mixtures of HMMs, *Bioinformatics* **20 Suppl 1**: I283–289.

Schrodt, P. A. (1998). Pattern Recognition of International Crises using Hidden Markov Models, *in* D. Richards (ed.), *Non-linear Models and Methods in Political Science*, University of Michigan Press, Ann Arbor, MI.

Tavazoie, S., Hughes, J., Campbell, M., Cho, R. & Church, G. (1999). Systematic determination of genetic network architecture., *Nat Genet.* **22**: 281–5.

Whitfield, M. L., Sherlock, G., Saldanha, A. J., Murray, J. I., Ball, C. A., Alexander, K. E., Matese, J. C., Perou, C. M., Hurt, M. M., Brown, P. O. & Botstein, D. (2002). Identification of genes periodically expressed in the human cell cycle and their expression in tumors., *Mol Biol Cell* **13**(6): 1977–2000.

Wu, C. (1983). On the convergence of the EM algorithm., *Ann. Stat.* pp. 95–103.

Yada, T. & Hirosawa, M. (1996). Gene recognition in cyanobacterium genomic sequence data using the hidden Markov model, *in* D. J. States, P. Agarwal, T. Gaasterland, L. Hunter & R. Smith (eds), *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, pp. 252–260.

Neuer Modellansatz (FUKA) zur Beschreibung der biologischen Abbauvorgänge in Kläranlagen

Frank Uhlenhut, Institut für Umwelttechnik – EUTEC – der FH Oldenburg/Ostfriesland/Wilhelmshaven

Zusammenfassung

Biologische Abbauvorgänge in Kläranlagen werden in wachsendem Maße mit einem Modellansatz auf der Grundlage der Monod-Kinetik beschrieben. Der neue Modellansatz FUKA (= Fundamentaler kinetischer Ansatz) beschreibt die biologischen Abbauvorgänge mit einem Satz fundamentaler, kinetischer Gleichungen und weist dadurch gegenüber der Monod-Kinetik einige bedeutende Vorteile auf.

Zunächst wurde ein neuer kinetischer Modellansatz für die Nitrifikationsvorgänge formuliert, in ein Simulationsprogramm umgesetzt und für die in Batch-Versuchen erhaltenen experimentellen Ergebnisse evaluiert. Es zeigte sich, dass die Konzentrationsverläufe der beteiligten Stickstoffkomponenten mit diesem Ansatz mit guter Übereinstimmung nachgebildet werden können.

Darauf basierend wurde ein neuer kinetischer Modellansatz für die in der biologischen Reinigungsstufe einer nach dem Belebungsverfahren arbeitenden Kläranlage ablaufenden Prozesse der C- und N-Elimination formuliert und als benutzerdefiniertes Modell in die Simulationssoftware SIMBA® implementiert. Die experimentellen Daten kommunaler Kläranlagen konnten erfolgreich mit dem neuen Modellansatz in der Simulation nachgebildet werden.

1 Einleitung

Seit einigen Jahren findet die dynamische Simulation im Bereich der Auslegung und Bemessung von Kläranlagen immer größeren Einsatz. Mit Hilfe geeigneter Modellierungen lässt sich das Verhalten einer Belebungsanlage sowohl hinsichtlich des Kohlenstoffabbaus als auch des Stickstoffabbaus beschreiben.

Auch bei bestehenden Kläranlagen wachsen die Anforderungen an die Reduzierung der Stickstoff- und Phosphorfrachten. Daraus ergibt sich vielfach die Notwendigkeit der Erweiterung bzw. der Verfahrensumstellung dieser Anlagen.

Unregelmäßige Schmutzfrachten im Zulauf (Stoßbelastung durch industrielle Einleiter, Regenereignisse, etc.) verursachen oftmals Probleme im Hinblick auf den optimalen Betrieb von Kläranlagen. Die Reaktionen des Betriebspersonals auf Grund von Erfahrung führen in vielen Fällen nur bedingt zum gewünschten Erfolg.

Dynamische Simulationsmodelle wurden entwickelt, um die Wirksamkeit biologischer Kläranlagen unter realitätsnahen Bedingungen schon in der Planungsphase testen zu können. Mit solchen Modellen lassen sich Optimierungsmaßnahmen zu einem Zeitpunkt durchführen, zu dem noch ausreichende Korrekturmöglichkeiten bestehen. Dynamische Modelle sind somit ein geeignetes Hilfsmittel für die Optimierung von Kläranlagen. Die Wirksamkeit der Kläranlage kann geprüft und verbessert werden, geplante Änderungen der Betriebsweise bzw. beabsichtigte Umbaumaßnahmen können im Vorfeld am Modell hinsichtlich der zu erwartenden Auswirkungen auf den Betrieb der Kläranlage und insbesondere auf die sich daraus ergebenden Ablaufwerte untersucht werden. Die Modelle dienen somit der Kostenminimierung und der Erhöhung der Zuverlässigkeit klärtechnischer Maßnahmen. Zukünftig wird die Simulation durch eine Kopplung an das Prozessleitsystem (Datentransfer) auch in den Bereich der aktiven Prozessüberwachung und -steuerung Einzug halten.

2 Theoretische Grundlagen

2.1 Funktionsweise einer nach dem Belebungsverfahren arbeitenden Kläranlage

Das wohl am häufigsten verwendete Verfahren zur Aufbereitung kommunaler Abwässer ist das Belebungsverfahren. Um das Prinzip zu veranschaulichen, sind in Abb. 1 die einzelnen Stufen einer nach dem Belebungsverfahren arbeitenden kommunalen Kläranlage dargestellt.

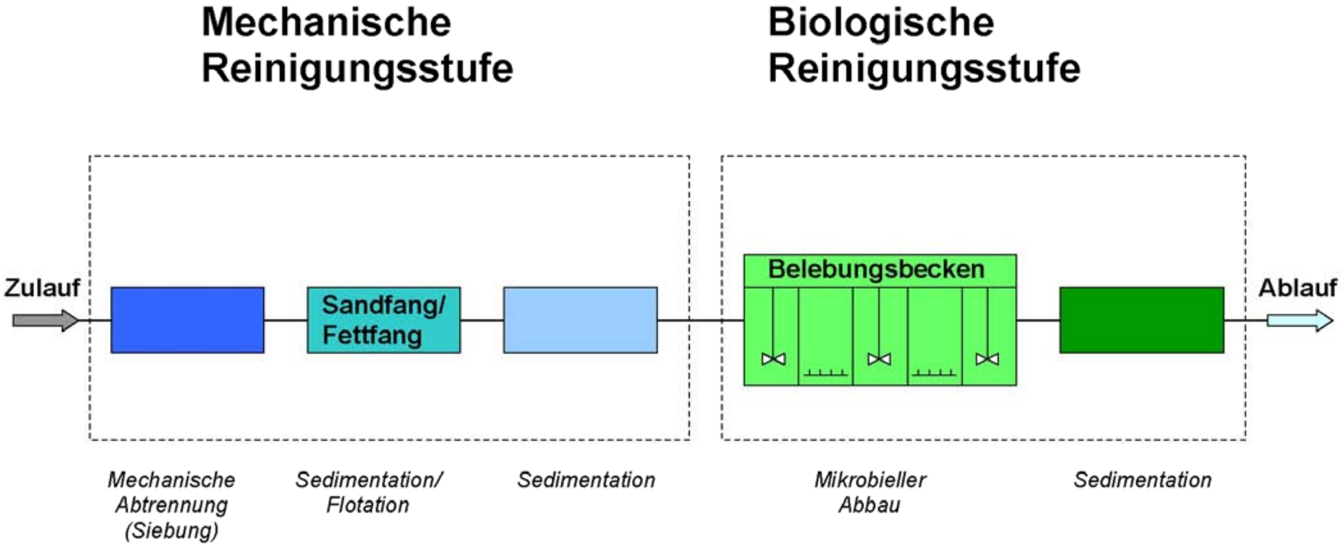


Abb. 1:Prinzipskizze einer nach dem Belebungsverfahren arbeitenden Kläranlage.

Wenn das Wasser aus der Kanalisation in das Pumpwerk der Kläranlage gelangt, wird es zunächst von groben Verschmutzungen befreit (Rechen). Im nachfolgenden Schritt wird das Wasser von leicht sedimentierenden festen Partikeln (z. B. Sand) und auftreibenden Verunreinigungen (z. B. Fette) getrennt (Sandfang/Fettfang). Danach gelangt das Abwasser in die Vorklärung, in der die Sedimentation feiner Partikel und evtl. eine chemische Fällung von Phosphaten stattfindet. Nach der mechanischen Reinigung erfolgt im Belebungsbecken der Abbau (und z. T. Adsorption) organischer Verbindungen durch die im Belebtschlamm enthaltenen Mikroorganismen (mikrobieller Abbau).

Das aus dem Belebungsbecken fließende Gemisch aus gereinigtem Abwasser und Belebtschlamm gelangt anschließend in das Nachklärbecken, in dem der Schlamm durch Sedimentation vom Abwasser getrennt wird. Das gereinigte Abwasser wird dann in einen Vorfluter (z. B. Fluß) eingeleitet. Es gibt mehrere Varianten des Belebungsverfahrens, wobei sich jeweils nur die Konstruktion des Belebungsbeckens (biologische Reinigung) unterscheidet (u. a. Belebungsverfahren mit vor- bzw. nachgeschalteter Denitrifikation, intermittierend betriebene Belebungsbecken, Kaskadensysteme (Step Feed-Kaskade, Plug Flow-Kaskade)).

2.2 *Modelle für die dynamische Simulation von Kläranlagen*

Ziel des Modells ist es, in Abhängigkeit von variablen Zuflussbedingungen (Tagesgang, Regenereignisse usw.) Vorhersagen zu treffen, wie sich bei gegebener Verfahrensvariante der Zustand des Systems und des gereinigten Abwassers ändert (insbesondere der Abbau organischer Stoffe, die Oxidation von NH_4^+ zu NO_3^- (Nitrifikation) und die Reduktion von NO_3^- zu N_2 (Denitrifikation)).

2.2.1 Modellansatz ASM 1

Für die Beschreibung der biologischen Abbauvorgänge in Kläranlagen wird in den aktuell auf dem Markt verfügbaren Simulationsprogrammen bislang überwiegend der Modellansatz ASM 1 (Activated Sludge Model No.1, IAWQ–Modell No.1) der IWA verwendet [1, 2]. Das Modell beschreibt die Vorgänge der CSB-Elimination sowie der Stickstoffelimination (Nitrifikation und Denitrifikation) in Belebtschlammsystemen (vgl. Abb. 2a, 2b) und beruht auf dem von J. Monod 1949 veröffentlichten Ansatz zur Charakterisierung des Wachstums von Mikroorganismen [3]. Der Ansatz von Monod ist in Analogie zu der enzymkinetischen Reaktionskinetik von Michaelis–Menten [4] aufgestellt worden. Die hierin getroffenen Annahmen (z. B.

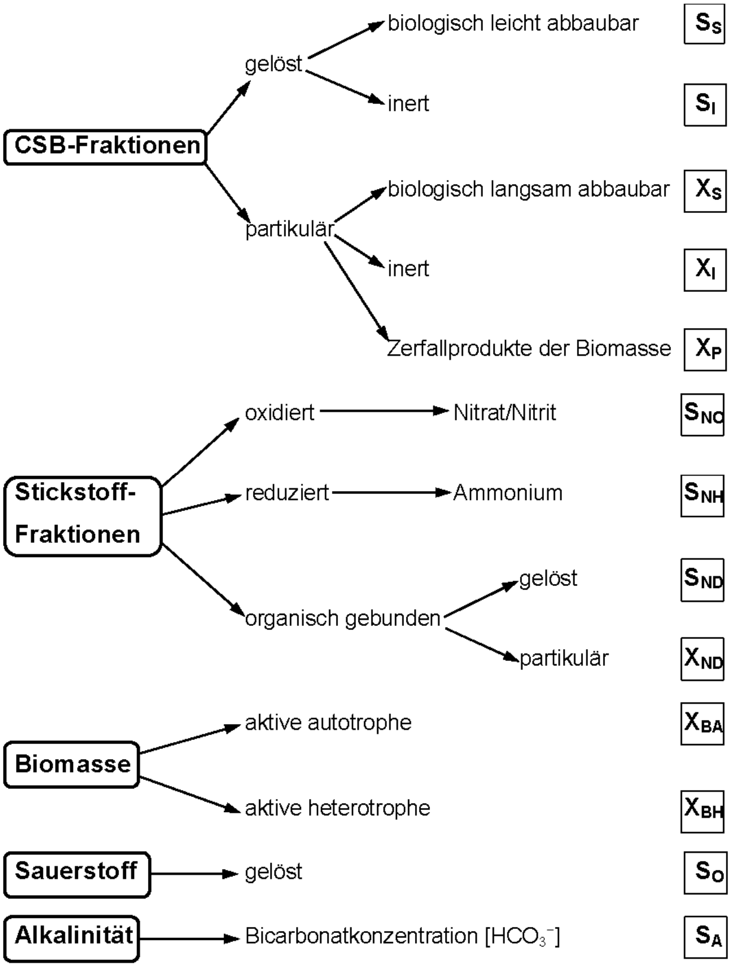


Abb. 2a: Übersichtsschema der 13 im Modellansatz ASM 1 definierten Stoffgruppen.

Quasistationarität der Zwischenprodukte) führen zu Einschränkungen der auf diesen Ansätzen basierenden Modelle:

Die Nitrifikation wird im ASM 1 als ein Gesamtvorgang aufgefasst, d. h. die beiden Reaktionsschritte Nitritation und Nitratation werden nicht differenziert betrachtet. Die im Verlaufe der Nitrifikation gebildeten Stickstoffverbindungen Nitrit und Nitrat werden als gemeinsame Stoffgruppe (S_{NO}) definiert.

Für eine detailliertere Betrachtung der Nitrifikation sowie die Aufnahme von Nitrit als eigenständige Stoffgruppe in den Modellansatz spricht jedoch die Tatsache, dass die Denitrifikation auch bereits ausgehend vom Nitrit erfolgen kann. Durch technisch anwendbare Varianten des Belebungsverfahrens, in denen das Ammonium nur bis zum Nitrit oxidiert wird und die

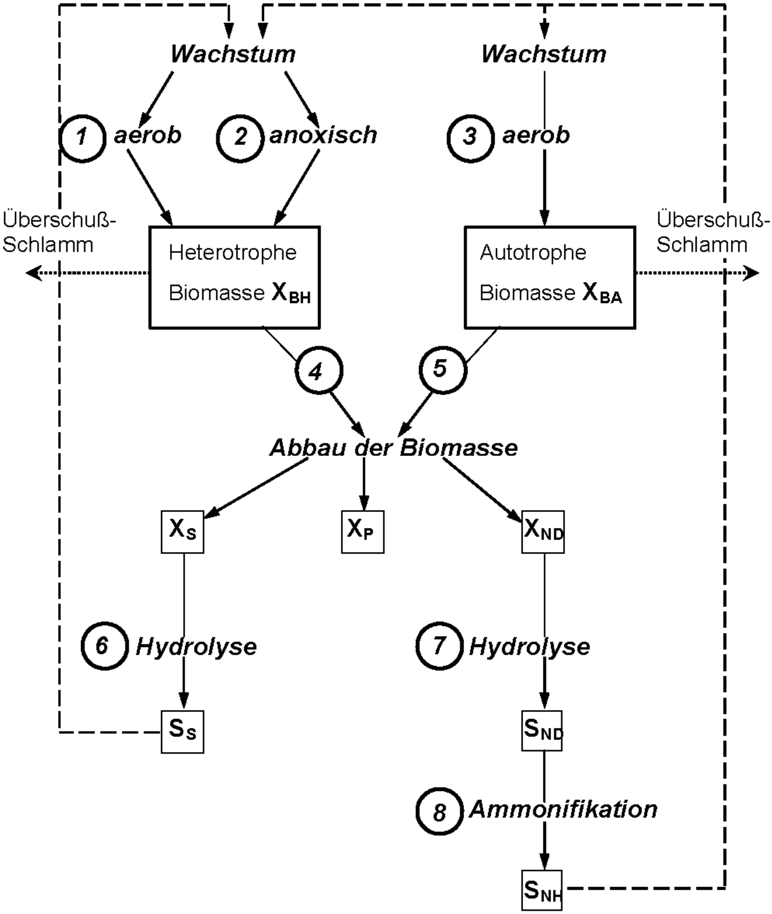


Abb. 2b: Vereinfachtes Schema der Zusammenhänge zwischen den 8 biologischen Prozessen des ASM 1 und den beteiligten Stoffgruppen.

nachfolgende Denitrifikation dementsprechend vom Nitrit aus stattfindet, könnten erhebliche Einsparungen hinsichtlich der Betriebskosten erzielt werden (Verringerung der erforderlichen Belüftungsleistung) [5, 6].

2.2.2 Neuer kinetischer Modellansatz FUKA (= Fundamentaler kinetischer Ansatz)

2.2.2.1 Modellgrundlagen

Es wurde ein neuer kinetischer Modellansatz zur Beschreibung biologischer Abbauvorgänge entwickelt. Für dessen Formulierung wird der komplette reaktionskinetische Ansatz ohne Näherungen zur Lösung gebracht. Folgender Reaktionsmechanismus wird für den Abbau eines Substrates S durch die Biomasse X zugrunde gelegt:

$$X + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} XS \xrightarrow{k_2} m * P + n * X \tag{1}$$

Das in der flüssigen Phase gelöste Substrat S gelangt zunächst durch einen Transportprozess (z. B. Diffusion) an die Zellmembran des Mikroorganismus X und passiert diese durch einen Transportvorgang (aktiver Transport oder Diffusion). Anschließend findet der biochemische Abbau des Substrates statt. Zur Vereinfachung des Modellansatzes wird nur ein „geschwindigkeitsbestimmender Schritt" (k_1) als Gesamtreaktion betrachtet, obwohl daran in der Regel verschiedene Enzyme beteiligt sind. Es wird zunächst die Bildung eines Enzym−Substrat−Komplexes XS postuliert, der dann einerseits zum Substrat und freien Enzym zurückreagieren (k_{-1}) und andererseits unter Bildung des Produktes P sowie neuer Biomasse X weiterreagieren (k_2) kann. Das in der Zelle gebildete Produkt P gelangt ebenfalls durch Diffusion oder Transport durch die Zellmembran und verteilt sich anschließend innerhalb der flüssigen Phase. Zur Erhaltung eines möglichst einfachen und übersichtlichen Modells werden die Diffusions- und Transportvorgänge nicht explizit betrachtet, sondern in die Reaktionsgeschwindigkeitskonstanten der betreffenden Schritte einbezogen.

Die Gleichungen für die Reaktionsgeschwindigkeiten der im Modellansatz betrachteten Komponenten werden nach dem in der chemischen Reaktionskinetik üblichen Verfahren formuliert:

$$\frac{d[S]}{dt} = -k_1 * [S] * [X] + k_{-1} * [XS] \tag{2}$$

$$\frac{d[X]}{dt} = -k_1 * [S] * [X] + k_{-1} * [XS] + n * k_2 * [XS] \tag{3}$$

$$\frac{d[XS]}{dt} = k_1 * [S] * [X] - k_{-1} * [XS] - k_2 * [XS] \qquad (4)$$

$$\frac{d[P]}{dt} = m * k_2 * [XS] \qquad (5)$$

2.2.2.2 Anwendung auf die biologischen Abbauvorgänge in Kläranlagen

Der neue kinetische Modellansatz wurde zunächst mit der Monod-Kinetik verglichen [7]. Dabei zeigte sich, dass dieser im Gegensatz zur Monod-Kinetik auch für die Beschreibung von Reaktionen mit Zwischenproduktbildung geeignet ist. Anschließend erfolgte die Anwendung des neuen kinetischen Ansatzes auf die Vorgänge der Nitrifikation sowie der Vergleich der simulierten Werte mit den experimentellen Ergebnissen aus Batch-Versuchen. Dabei konnten die experimentell erhaltenen Konzentrationsverläufe für die Größen Ammonium, Nitrit und Nitrat mit dem neuen kinetischen Modellansatz für die Nitrifikationsvorgänge mit guter Übereinstimmung nachgebildet werden [8]. Auf der Grundlage dieser Ergebnisse wurde ein neuer kinetischer Modellansatz für die in der biologischen Reinigungsstufe einer nach dem Belebungsverfahren arbeitenden Kläranlage ablaufenden Prozesse der C- und N-Elimination entwickelt.

Folgende Reaktionen müssen bei der Formulierung eines Modellansatzes für die Abbauprozesse der Kohlenstoff- und Stickstoffverbindungen in der biologischen Reinigungsstufe einer nach dem Belebungsverfahren arbeitenden Kläranlage berücksichtigt werden. Die in geschweifte Klammern gesetzten Reaktionspartner (Alkalinität, H_2O) wurden für die Formulierung des kinetischen Ansatzes vorerst nicht berücksichtigt und sind hier nur der Vollständigkeit halber aufgeführt worden.

(1) Nitrifikation (autotrophe Biomasse X_A)

Nitritation: $NH_4^+ + 1\frac{1}{2}\,O_2 \rightarrow NO_2^- \;\{+ H_2O + 2\,H^+\}$

Nitratation: $NO_2^- + \frac{1}{2}\,O_2 \rightarrow NO_3^-$

(2) Denitrifikation unter Einbeziehung des C–Abbaus (Substrat S), Nitrat-Atmung (heterotrophe Biomasse X_H):

$S + 4\,NO_3^- + \{4\,H^+\} \rightarrow 5\,CO_2 + 2\,N_2 + \{2\,H_2O\}$

(3) C–Abbau (heterotrophe Biomasse X_H):

$(CH_2O)_n + n\,O_2 \rightarrow n\,CO_2 + \{n\,H_2O\}$

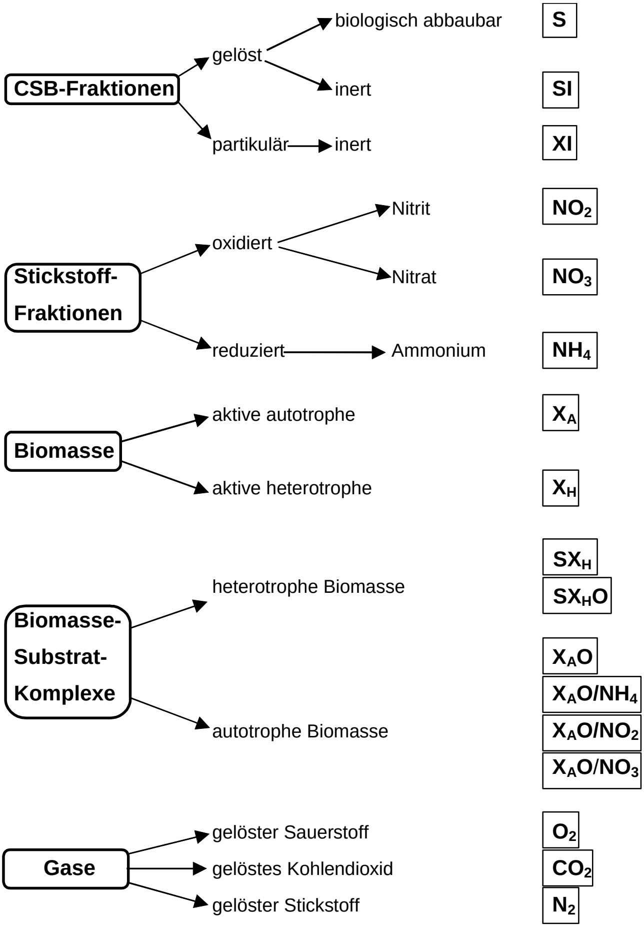


Abb. 3a: Übersichtsschema der 17 im Modellansatz FUKA definierten Stoffgruppen.

Bei der Formulierung des kinetischen Ansatzes wurden die unterschiedlichen Arten von Biomasse (X_A, X_H) berücksichtigt. Genaue stöchiometrische Verhältnisse bzw. Faktoren lassen sich in diesem Fall nicht angeben (wie z. B. bei einer chemischen Reaktion), da einer der Reaktionspartner (der Belebtschlamm, bzw. die Biomassen X_A und X_H) nur als ganze Einheit betrachtet werden kann.

Die 17 Stoffgruppen des neuen Modellansatzes FUKA sind in *Abb. 3a* zusammengestellt. *Abb. 3b* zeigt die 6 biologischen Prozesse und veranschaulicht deren Wirkungen auf die einzelnen Stoffgruppen.

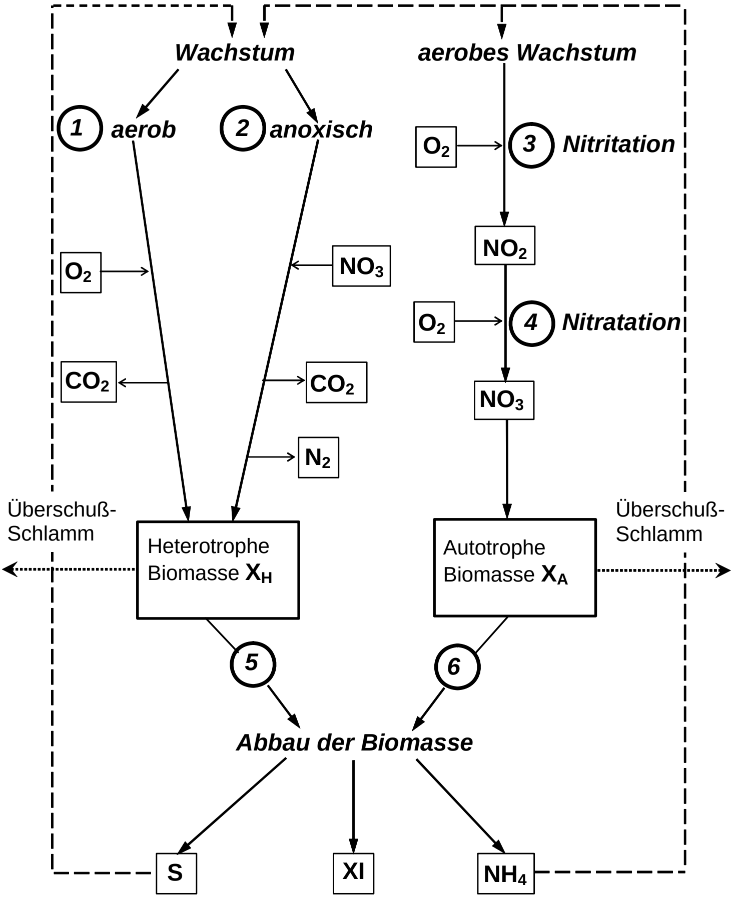


Abb. 3b: Vereinfachte Darstellung der Zusammenhänge zwischen den Stoffgruppen und Prozessen des Modellansatzes FUKA (sämtliche Biomasse-Substrat-Komplexe wurden nicht berücksichtigt).

Für die an den betrachteten Umsetzungen beteiligten Komponenten Kohlenstoff, Stickstoff und Sauerstoff wurde eine Bilanzierung hinsichtlich der Molzahlen vorgenommen (Erhaltungssätze).

Dabei wurde berücksichtigt, dass für die Neubildung der Biomasse (Wachstum der autotrophen und heterotrophen Mikroorganismen) auch gewisse Anteile Stickstoff und Sauerstoff verwendet werden. Die beim Abbau der autotrophen und heterotrophen Biomasse entstehenden Produkte werden auf die Fraktionen S, NH_4, O_2 (C-, N- und O-Anteile, die erneut als Substrat dienen können) und XI (inerter Anteil) verteilt.

2.3 Simulationsprogramm SIMBA®

Das verwendete Simulationsprogramm SIMBA 4.2® [9] wurde vom Institut für Automation und Kommunikation e. V. (ifak; Magdeburg/Barleben) als Unterprogramm (Toolbox) der Simulationssoftware MATLAB®/ SIMULINK® (Fa. Mathworks Inc.) entwickelt und ermöglicht die gleichzeitige Nutzung der vielfältigen Möglichkeiten dieses Simulationswerkzeugs (z. B. grafische Darstellung, mathematische Berechnungen etc.). Die Nachbildung der betrachteten Kläranlage erfolgt vollgrafisch. Alle Module zur Nachbildung des abwasserseitigen Teils einer Kläranlage sind vorhanden. Zudem können auch die im Unterprogramm SIMULINK® vorhandenen Regelungsbausteine und Funktionsblöcke mit einbezogen werden. Für die Simulationsrechnungen sind die Modellansätze der IWA (ASM 1 ASM 2d und ASM 3) enthalten.

Das Programm SIMBA 4.2® wurde als „offenes System“ konzipiert und ermöglicht dem Benutzer die Einbindung eigener Modellansätze (auch ohne Monod-Kinetik) unter Verwendung der grafischen Oberfläche und des gesamten Programmgerüstes. Dieser Punkt war für die vorliegende Arbeit von entscheidender Bedeutung, da die Entwicklung und Einbindung eines eigenen Modellansatzes das Hauptziel des Vorhabens war.

Seit der Markteinführung im Jahre 1995 hat sich SIMBA 4.2® in Deutschland zum verbreitetsten Programm für die dynamische Simulation von Kläranlagen entwickelt und wird teilweise auch bereits von Seiten der staatlichen Aufsichtsbehörden (in Nordrhein-Westfalen) als Prüfinstrument eingesetzt.

3 Beschreibung der biologischen Abbauvorgänge im Modell

Der in Abschnitt 2.2.2.2 postulierte Reaktionsmechanismus für den biologischen Abbau der Kohlenstoff- und Stickstoffverbindungen führt zu einem System von Differentialgleichungen zur Beschreibung der zeitlichen Konzentrationsänderungen der beteiligten Stoffe. Dieses wurde zunächst in die Matrix-Notation der IWA [1, 10] übertragen. Die stöchiometrischen Faktoren ergeben sich aus den genannten Erhaltungssätzen. Anschließend wurde der Modellansatz unter Verwendung der formalen Matrixbeschreibung (FOX-Notation) [9, 10] als benutzerdefiniertes Modell in die offene Modellbibliothek der Simulationssoftware SIMBA® implementiert. Um für den neuen kinetischen Modellansatz eine Sensitivitätsanalyse sowie eine Anpassung der Modellparameter durchführen zu können, wurde die Kläranlage Emden im Simulationsprogramm SIMBA® unter Verwendung der entsprechenden Blöcke der offenen Modellbibliothek abgebildet. Für die im Modellansatz definierten Parameter mussten zunächst möglichst plausible Anfangswerte vorgegeben werden. Einige dieser Größen wurden aus Korrelationen zwischen den Parametern des Monod-Ansatzes und des kinetischen Ansatzes erhalten. Zudem wurde angenommen, dass die Geschwindigkeitskonstante der Rückreaktion wesentlich kleiner als die Geschwindigkeitskonstante der jeweiligen Hinreaktion ist.

Unter Verwendung dieser Beziehungen ließen sich ausgehend von Defaultwerten des ASM 1 [1, 11] bzw. SIMBA® [9] Ausgangswerte für einige Parameter des kinetischen Modellansatzes berechnen. Für die Anfangskonzentrationen der im Modell definierten Stoffgruppen konnte bei einigen der Stoffgruppen auf die Werte der experimentell ermittelten Tagesganglinien zurückgegriffen werden. Ziel dieser Untersuchung war es, zum einen die besonders sensiblen Parameter zu erkennen, und zum anderen die Abhängigkeiten zwischen den Parametern und den im Modellansatz enthaltenen Komponenten aufzuzeigen. Außerdem konnte darauf basierend eine Anpassung ausgewählter Parameter an die experimentellen Daten vorgenommen werden.

4 Praktische Anwendungen des neuen Modellansatzes FUKA

4.1 Nachbildung der Kläranlage Emden

Die untersuchte Kläranlage Emden ist für 90.000 EW+EWG ausgelegt. Der Abwasservolumenstrom beträgt in etwa 10.000 m^3/d (Trockenwetterabfluss). Das Belebungsbecken ist in fünf Zonen aufgeteilt, die alternierend anoxisch und oxisch betrieben werden (Kaskadensystem; *Abb. 4* zeigt das Verfahrensschema in SIMBA 4.2®).

Die Nachbildung der Kläranlage Emden mit dem Modell ASM 1 ergab bei Verwendung der Parameterwerte des Standardparametersatzes [12] für die meisten Stoffgruppen eine befriedigende Übereinstimmung zwischen den mit der Simulation berechneten Ablaufwerten und den experimentell ermittelten Ablaufwerten. Die simulierten Werte für die Stoffgruppe Nitrat/Nitrit lagen jedoch um bis zu einem Faktor 6 über den realen Nitratgehalten im Ablauf der Kläranlage. Auch durch eine gezielte Variation einzelner Modellparameter im Rahmen einer eindimensionalen Sensitivitätsanalyse [13] ließ sich diese Diskrepanz nicht beseitigen.

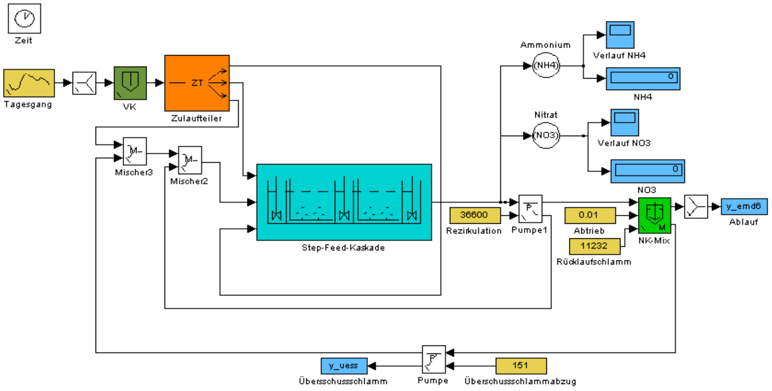


Abb. 4: Verfahrensschema der Kläranlage Emden in SIMBA 4.2®.

Dagegen konnten mit dem neuen kinetischen Modellansatz FUKA unter Verwendung der angepassten Parameterwerte die Konzentrationsverläufe der beteiligten Stickstoffkomponenten mit großer Übereinstimmung nachgebildet werden.

4.2 Nachbildung der Kläranlage Remels

Die Kläranlage Remels wurde für eine Ausbaugröße von 20.000 EW ausgelegt (zum Zeitpunkt der Untersuchungen aber nur mit einer Anschlussgröße von 7000 EW betrieben). In der Kläranlage wird ausschließlich kommunales Abwasser aus einer Trennkanalisation nach dem Belebtschlamm-verfahren behandelt. Das Belebungsbecken ist als Plug Flow-Kaskade nach MEG·A·TEC konstruiert Unter Verwendung der experimentell ermittelten Eingangsdaten (Wochenganglinien der relevanten Zulaufparameter CSB, Gesamt-Stickstoff, Ammonium, Nitrit, Nitrat / CSB-Fraktionierung im Zulauf, etc.) wurde die Kläranlage Remels mit den Modellansätzen FUKA und ASM 1 dynamisch simuliert (*Abb. 5* zeigt das Verfahrensschema in SIMBA 4.2®). Dazu wurde im Vorfeld unter Verwendung der Ergebnisse der für die Kläranlage Emden/Larrelt erstellten Sensitivitätsanalysen [13, 14] eine Modellkalibrierung für beide Modellansätze durchgeführt. Dabei zeigte sich, dass die Modellanpassung für FUKA einfacher möglich ist als für das ASM 1. Dies ist begründet in der Tatsache, dass sich im Modell FUKA eindeutigere Zusammenhänge zwischen den Modellparametern und den einzelnen Stoffgruppen herstellen lassen.

Aus den Ergebnissen der Simulationsrechnungen werden erhebliche Unterschiede zwischen den Modellansätzen FUKA und ASM 1 erkennbar. Die Nachbildung der biologischen Prozesse in der Kläranlage Remels mit FUKA ergab im Vergleich zum ASM 1 hinsichtlich der betrachteten Stickstoffverbindungen eine bessere Übereinstimmung zwischen realen und experimentell ermittelten Ablaufwerten.

Durch die Anpassung der Kläranlage Remels im Modell an die besonderen Gegebenheiten der Plug Flow-Kaskade konnte mit dem ASM 1 zwar eine bessere Übereinstimmung zwischen Simulation und Realität erreicht werden (Verringerung der simulierten Nitrat- und Ammoniumkonzentrationen im Ablauf), dennoch lagen die simulierten Ablaufkonzentrationen über den experimentell ermittelten Konzentrationen. Bei dieser Anpassung wurden die in der Plug Flow-Kaskade in den aeroben Zonen verstärkt ablaufenden Speicherprozesse organischer Substrate berücksichtigt. Da in den Modellansätzen FUKA und ASM 1 keine C-Speicherprozesse enthalten sind, wird dieser Effekt in der Modellanlage vereinfacht als C-Speicher dargestellt, wobei ein Teil des Biomasse-CSB (heterotrophe Biomasse X_H) abgezogen und zur gelösten, biologisch leicht abbaubaren CSB-Fraktion S_S bzw. S addiert wird (Realisierung in der Simulation siehe *Abb. 6*). Diese Speicher können in den unbelüfteten Denitrifikationszonen als zusätzliches Substrat (C-Quelle) genutzt werden.

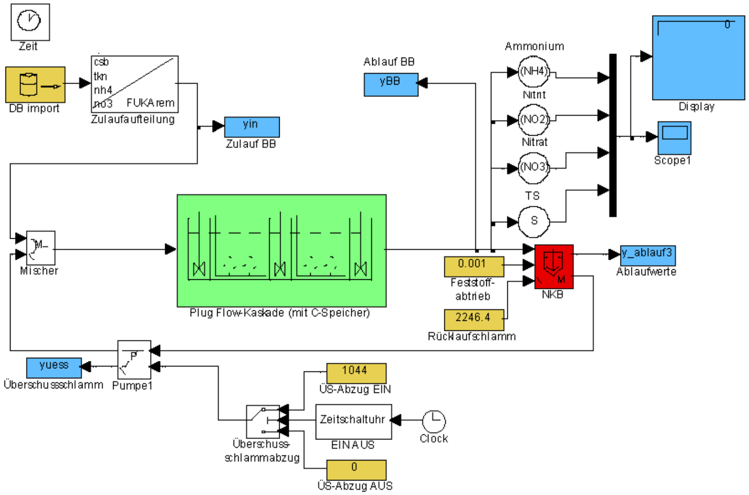


Abb. 5: Verfahrensschema der Kläranlage Remels in SIMBA 4.2®.

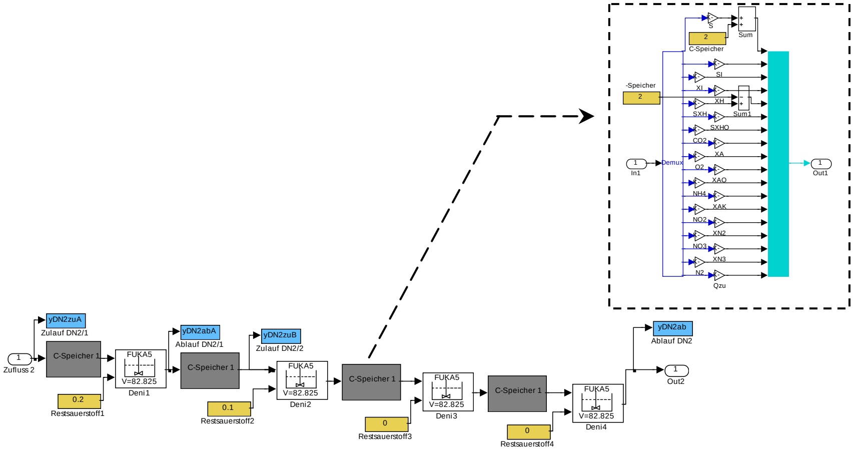


Abb. 6: Abbildung der Speicherprozesse in der Simulation.

5 Schlussbemerkungen

Es wurde ein neuer kinetischer Modellansatz zur Beschreibung des biologischen Abbaus der Kohlenstoff- und Stickstoffverbindungen in Kläranlagen formuliert und erfolgreich für die Simulation von zwei kommunalen Kläranlagen eingesetzt.

Das größte Problem hinsichtlich der praktischen Anwendbarkeit des neuen kinetischen Modellansatzes FUKA liegt in der Festlegung bzw. Bestimmung der Modellparameter (reaktionskinetische Parameter und der messtechnisch bisher nicht erfassbaren Stoffgruppen). Hier konnte für einen Teilbereich (Nitrifikation) gezeigt werden, dass die Werte dieser Parameter durch geeignete Laborexperimente erhalten werden können [8].

Im Gegensatz zum ASM 1 wird bei dem neuen Modellansatz FUKA das gesamte Differentialgleichungssystem ohne die Vereinfachungen nach Monod bzw. Michaelis-Menten (quasistationäre Näherung) gelöst und zudem das Nitrit als eigenständige Komponente berücksichtigt. Damit lässt sich im Vergleich zum ASM 1 eine genauere Abbildung der Nitrifikationsvorgänge erreichen. Das neue Modell FUKA bietet eine interessante Alternative zum bisher verwendeten ASM 1.

Literatur

[1] *Henze, M., Grady, Jr., C. P. L., Gujer, W., Marais, G. v. R., Matsuo, T.:* Final Report - Activated sludge model N°. 1/IAWQ task group on mathematical modelling for design and operation of biological wastewater treatment. IAWQ Scientific and Technical Reports, London, 1987.

[2] *Gujer, W.:* Ein dynamisches Modell für die Simulation von komplexen Belebtschlammverfahren. Habilitation ETH Zürich, 1985.

[3] *Monod, J.:* The growth of bacterial cultures, Annual Review of Microbiology (1949), Vol. III.

[4] *Michaelis, L.; Menten, M. L.:* Die Kinetik der Invertinwirkung, Biochem. Z. (1913), 49, S. 333-369.

[5] *Hippen, A., Helmer, C., Scholten, E., Kunst, S., Diekmann, H., Rosenwinkel, K.-H., Seyfried, C. F.:* Neue Möglichkeiten der Stickstoffelimination bei Abwässern mit niedrigem C/N-Verhältnis: Aerobe Deammonifikation. Korrespondenz Abwasser (1998) Nr. 12, S. 2287-2293.

[6] *Sölter, T., Orth, H.:* Stickstoffentfernung über Nitrit aus Trübwässern, Korrespondenz Abwasser (1998) Nr. 6, S. 1122-1131.

[7] *Uhlenhut, F., Berendes, O., Frauendorfer, E., Siefert, E., Schlaak, M.:* Kinetische Beschreibung biologischer Abbauvorgänge , gwf-Wasser/Abwasser 140 (1999) Nr. 6, S. 424-430.

[8] *Uhlenhut, F., Schlaak, M., Siefert, E., Schuller, D.:* Kinetischer Modellansatz zur Beschreibung der Nitrifikation, gwf-Wasser/Abwasser 141 (2000) Nr. 2, S. 103-108.

[9] Institut für Automation und Kommunikation e. V.:
Benutzerhandbuch für das Programm SIMBA 4.0®, Barleben, Mai 2001.

[10] *Alex, J., Jumar, U.*: Ein Zugang zur formalisierten Beschreibung und Implementierung von Simulationsmodellen am Beispiel der biologischen Abwasserreinigung. In: Keller, H. B., Grützner, R., Hohmann, R. (Hrsg.): 6. Treffen des Arbeitskreises „Werkzeuge für Simulation und Modellbildung in Umweltanwendungen„. Wissenschaftliche Berichte Forschungszentrum Karlsruhe Technik und Umwelt, 1996, S. 23-34.

[11] *Otterpohl, R.:* Dynamische Simulation zur Unterstützung der Planung und des Betriebes kommunaler Kläranlagen, Dissertation TH Aachen, Gewässerschutz Wasser Abwasser, 1995, 151, S. 38.

[12] *Bornemann, C., Londong, J., Freund, M., Nowak, O., Otterpohl, R., Rolfs, T.:* Hinweise zur dynamischen Simulation von Belebungsanlagen mit dem Belebtschlammmodell Nr. 1 der IAWQ, Korrespondenz Abwasser (1998) Nr. 3, S. 455-462.

[13] *Uhlenhut, F., Siefert, E., Schlaak, M., Schuller, D.:* Sensitivitätsanalyse der Parameter des Simulationsprogramms SIMBA® am Beispiel der Kläranlage Emden/Larrelt, gwf-Wasser/Abwasser 140 (1999) Nr. 10, S. 704-711.

[14] *Uhlenhut, F.:* Modellierung biologischer Abbauvorgänge in Kläranlagen – ein neuer reaktionskinetischer Ansatz, Dissertation Universität Oldenburg, 1999.

Danksagung

Diese Arbeit wurde zu einem großen Teil im Rahmen einer Dissertation am Institut für Umwelttechnik – EUTEC der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven durchgeführt. Ich möchte mich bei Herrn Prof. Dr. Dieter Schuller und Herrn Prof. Dr. Michael Schlaak sowie bei den Herren Prof. Dr. Eike Siefert und Prof. Dr. Axel Borchert für ihre Unterstützung bedanken.

Für die stets gewährte Hilfeleistung bei den in der Kläranlage Emden durchzuführenden Arbeiten gilt der besondere Dank des Autors Herrn W. van Lessen (Betriebsleiter des Hauptklärwerkes der Stadt Emden) sowie dem Betriebspersonal der Kläranlagen Emden und Remels.

Das Projekt wurde finanziell durch das Land Niedersachsen (Mittel aus der Stiftung Volkswagenwerk) im Rahmen eines Forschungsschwerpunktes am Institut für Umwelttechnik sowie durch das Bundesministerium für Wirtschaft und Arbeit – BMWA (Mittel aus dem Förderprogramm PRO INNO) unterstützt.

Verwendete Formelzeichen:

S_S	Biologisch rasch abbaubare, gelöste organische Stoffe (Substrat)	[(g CSB)/m^3]
S_I	Biologisch inerte, gelöste organische Stoffe	[(g CSB)/m^3]
X_S	Biologisch langsam abbaubare organische Stoffe	[(g CSB)/m^3]
X_I	Biologisch inerte, partikuläre organische Stoffe	[(g CSB)/m^3]
X_P	Partikuläre Zerfallsprodukte der Biomasse	[g CSB/m^3]
S_{NO}	Summe aus Nitrat- und Nitrit- Stickstoff	[(g N)/m^3]
S_{NH}	Summe aus Ammoniak und Ammonium-Stickstoff	[(g N)/m^3]
S_{ND}	Biologisch abbaubarer, gelöster organisch gebundener Stickstoff	[(g N)/m^3]
X_{ND}	Biologisch abbaubarer, partikulärer organisch gebundener Stickstoff	[(g N)/m^3]
X_{BA}	Aktive autotrophe Biomasse	[(g CSB)/m^3]
X_{BH}	Aktive heterotrophe Biomasse	[(g CSB)/m^3]
S_O	Gelöster Sauerstoff	[-(g CSB)/m^3]
S_A	Alkalinität (Bikarbonat (HCO$_3^-$))	[mol/m^3]
S	Substratkonzentration	[g l^{-1}]
P	gebildetes Produkt	[g l^{-1}]
X	Biomasse	[g l^{-1}]
XS	Enzym–Substrat–Komplex (fiktiver Biomasse–Substrat–Komplex)	[g l^{-1}]
k_1	Reaktionsgeschwindigkeitskonstante	[l g^{-1} min^{-1}]
k_{-1}, k_2	Reaktionsgeschwindigkeitskonstanten	[min^{-1}]
m, n	stöchiometrische Koeffizienten	[–]

S	Biologisch rasch abbaubare, gelöste organische Stoffe (Substrat)	[g/l]
SI	Biologisch inerte, gelöste organische Stoffe	[g/l]
XI	Biologisch inerte, partikuläre organische Stoffe	[g/l]
NH_4	Ammonium	[g/l]
NO_2	Nitrit	[g/l]
NO_3	Nitrat	[g/l]
X_A	Aktive autotrophe Biomasse	[g/l]
X_H	Aktive heterotrophe Biomasse	[g/l]
SX_H	Fiktiver Biomasse−Substrat−Komplex	[g/l]
SX_HO	Fiktiver Biomasse−Substrat−Komplex (mit Sauerstoff gesättigte heterotrophe Biomasse X_H)	[g/l]
SX_H/NO_3	Fiktiver Biomasse−Substrat−Komplex (nach Aufnahme von Nitrat)	[g/l]
X_AO	Fiktiver Biomasse−Substrat−Komplex(mit Sauerstoff gesättigte autotrophe Biomasse X_A)	[g/l]
X_AO/NH_4	Fiktiver Biomasse−Substrat−Komplex (nach Aufnahme von Ammonium)	[g/l]
X_AO/NO_2	Fiktiver Biomasse−Substrat−Komplex (nach Aufnahme von Nitrit)	[g/l]
O_2	Gelöster Sauerstoff	[g/l]
CO_2	Gelöstes Kohlendioxid	[g/l]
N_2	Gelöster Stickstoff	[g/l]

Anschriften der Autoren

Odon Angeles-Palacios
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: angeles@mpi-magdeburg.mpg.de

Dr. Julien Aubert
Institut de Physique du Globe de Paris
4, place Jussieu, F 75252 Paris cedex 05
E-Mail: aubert@ipgp.jussieu.fr

Dr. Niko Beerenwinkel
University of California
Dept. of Methematics
898 Evans Hall
Berkeley, CA 94720-3840
E-Mail: Niko.Beerenwinkel@gmx.net

Ivan G. Costa
Max-Planck-Institut für Molekulare Genetik
Ihnestr. 73, 14195 Berlin
E-Mail: costa@molgen.mpg.de

Dr. Thomas Fischbacher
Max-Planck-Institut für Gravitationsphysik
Albert-Einstein-Institut
Am Mühlenweg 1, 14476 Golm
E-Mail: tf@aei.mpg.de

Benjamin Georgi
Max-Planck-Institut für Molekulare Genetik
Ihnestr. 73, 14195 Berlin
E-Mail: georgi@molgen.mpg.de

Prof. Dr. Ernst Dieter Gilles
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail:gilles@mpi-magdeburg.mpg.de

Dipl.-Inf. Martin Ginkel
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: mginkel@mpi-magdeburg.mpg.de

Prof. Dr. Achim Kienle
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: kienle@mpi-magdeburg.mpg.de

Andreas Kremling
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: kremling@mpi-magdeburg.mpg.de

Dr. Michael Mangold
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: mangold@mpi-magdeburg.mpg.de

Dr. Markus Rampp
Rechenzentrum der Max-Planck-Gesellschaft
Max-Planck-Institut für Plasmaphysik
Boltzmannstr. 2, 85748 Garching
E-Mail: mjr@rzg.mpg.de

Wasinee Rungsarityotin
Max-Planck-Institut für Molekulare Genetik
Ihnestr. 73, 14195 Berlin
E-Mail: rungsari@molgen.mpg.de

Dr. Alexander Schliep
Max-Planck-Institut für Molekulare Genetik
Ihnestr. 73, 14195 Berlin
E-Mail: schliep@molgen.mpg.de

Dipl.-Math. Alexander Schönhuth
ZAIK, Universitaet zu Koeln
Weyertal 80, 50931 Köln
E-Mail: schoenh@zpr.uni-koeln.de

Dr. Thomas Soddemann
Rechenzentrum der Max-Planck-Gesellschaft
Max-Planck-Institut für Plasmaphysik
Boltzmannstr. 2, 85748 Garching
E-Mail: tks@rzg.mpg.de

Dr. Frank Uhlenhut
FH Oldenburg/Ostfriesland/Wilhelmshafen
Institut für Umwelttechnik - EUTEC
Constantiaplatz 4, 26723 Emden
E-Mail: uhlenhut@fho-emden.de

Dipl.-Ing. Roland Waschler
Max-Planck-Institut für Dynamik komplexer technischer Systeme
Sandtorstr. 1, 39106 Magedburg
E-Mail: waschler@mpi-magdeburg.mpg.de

Dr. Johannes Wicht
Max-Planck-Institut für Sonnensystemforschung
Max-Planck-Strasse 2, 37191 Katlenburg Lindau
E-Mail: wicht@linmpi.mpg.de

Dipl.-Ing. Peter Wittenburg
Max-Planck-Institut für Psycholinguistik
Postbox 310, NL-6500 AH Nijmegen, Niederlande
E-Mail: peter.wittenburg@mpi.nl

In der Reihe GWDG-Berichte sind zuletzt erschienen:

Nr. 40 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1994
1995

Nr. 41 *Brinkmeier, Fritz* (Hrsg.):
Rechner, Netze, Spezialisten. Vom Maschinenzentrum zum Kompetenzzentrum – Vorträge des Kolloquiums zum 25jährigen Bestehen der GWDG
1996

Nr. 42 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1995
1996

Nr. 43 *Wall, Dieter* (Hrsg.):
Kostenrechnung im wissenschaftlichen Rechenzentrum – Das Göttinger Modell
1996

Nr. 44 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1996
1997

Nr. 45 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
13. DV-Treffen der Max-Planck-Institute – 21.-22. November 1996 in Göttingen
1997

Nr. 46 **Jahresberichte 1994 bis 1996**
1997

Nr. 47 *Heuer, Konrad, Eberhard Mönkeberg und Ulrich Schwardmann:*
Server-Betrieb mit Standard-PC-Hardware unter freien UNIX-Betriebssystemen
1998

Nr. 48 *Haan, Oswald* (Hrsg.):
Göttinger Informatik Kolloquium – Vorträge aus den Jahren 1996/97
1998

Nr. 49 *Koke, Hartmut und Engelbert Ziegler* (Hrsg.):
IT-Infrastruktur im wissenschaftlichen Umfeld – 14. DV-Treffen der Max-Planck-Institute, 20.-21. November 1997 in Göttingen
1998

Nr. 50 *Gerling, Rainer W.* (Hrsg.):
Datenschutz und neue Medien – Datenschutzschulung am 25./26. Mai 1998
1998

Nr. 51 *Plesser, Theo und Peter Wittenburg* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1997
1998

Nr. 52 *Heinzel, Stefan und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1998
1999

Nr. 53 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Internet- und Intranet-Technologien in der wissenschaftlichen Datenverarbeitung – 15. DV-Treffen der Max-Planck-Institute, 18. - 20. November 1998 in Göttingen
1999

Nr. 54 *Plesser, Theo und Helmut Hayd* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 1999
2000

Nr. 55 *Kaspar, Friedbert und Hans-Ulrich Zimmermann* (Hrsg.):
Neue Technologien zur Nutzung von Netzdiensten – 16. DV-Treffen der Max-Planck-Institute, 17. - 19. November 1999 in Göttingen
2000

Nr. 56 *Plesser, Theo und Helmut Hayd* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2000
2001

Nr. 57 *Hayd, Helmut und Rainer Kleinrensing (Hrsg.)*
17. und 18. DV-Treffen der Max-Planck-Institute,
22. - 24. November 2000, 21. – 23. November 2001 in Göttingen
2002

Nr. 58 Macho*, Volker und Theo Plesser* (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2001
2003

Nr. 59 *Suchodoletz, Dirk von*:
Effizienter Betrieb großer Rechnerpools – Implementierung am Beispiel des Studierendennetzes an der Universität Göttingen
2003

Nr. 60 *Haan, Oswald (Hrsg.)*:
Erfahrungen mit den IBM-Parallelrechnersystemen RS/6000 SP und pSeries690
2003

Nr. 61 *Rieger, Sebastian*:
Streaming-Media und Multicasting in drahtlosen Netzwerken – Untersuchung von Realisierungs- und Anwendungsmöglichkeiten
2003

Nr. 62 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2002
2003

Nr. 63 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2003
2004

Nr. 64 *Koke, Hartmut* (Hrsg.):
GÖ* - Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ*-Vorantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation"
2004

Nr. 65 *Koke, Hartmut* (Hrsg.):
GÖ* - Integriertes Informationsmanagement im heterogenen eScience-Umfeld: GÖ*-Hauptantrag für die DFG-Förderinitiative „Leistungszentren für Forschungsinformation"
2004

Nr. 66 *Bussmann, Dietmar und Andreas Oberreuter* (Hrsg.):
19. und 20. DV-Treffen der Max-Planck-Institute
20.-22. November 2002
19.-21. November 2003 in Göttingen
2004

Nr. 67 *Gartmann, Christoph und Jochen Jähnke* (Hrsg.):
21. DV-Treffen der Max-Planck-Institute
17.-19. November 2004 in Göttingen
2005

Nr. 68 *Kremer, Kurt und Volker* Macho (Hrsg.):
Forschung und wissenschaftliches Rechnen – Beiträge zum Heinz-Billing-Preis 2004
2004

Nähere Informationen finden Sie im Internet unter
`http://www.gwdg.de/forschung/publikationen/gwdg-berichte/index.html`