Vorlesungsskript

Softwarewerkzeuge der Bioinformatik

Dozent: Prof. Dr. Volkhard Helms

Übungen: PD Dr. Michael Hutter, Dr. Tihamér Geyer, Barbara Hutter

Zentrum für Bioinformatik, Universität des Saarlandes Saarbrücken

Sommersemester 2010

Inhaltsverzeichnis

1 Datenbanken

Die Vorlesung Softwarewerkzeuge der Bioinformatik hält sich an das Buch *„Understanding bioinformatics“* von Marketa Zvelebil und Jeremil O. Baum, 2008. (ISBN-13: 978-0-8153-4024-9). Die Bilder sind zum größten Teil aus diesem Buch entnommen. Die Quellen anderer Bilder, die nicht selbst erstellt sind, sind separat angegeben.

1.1 Was ist Bioinformatik?

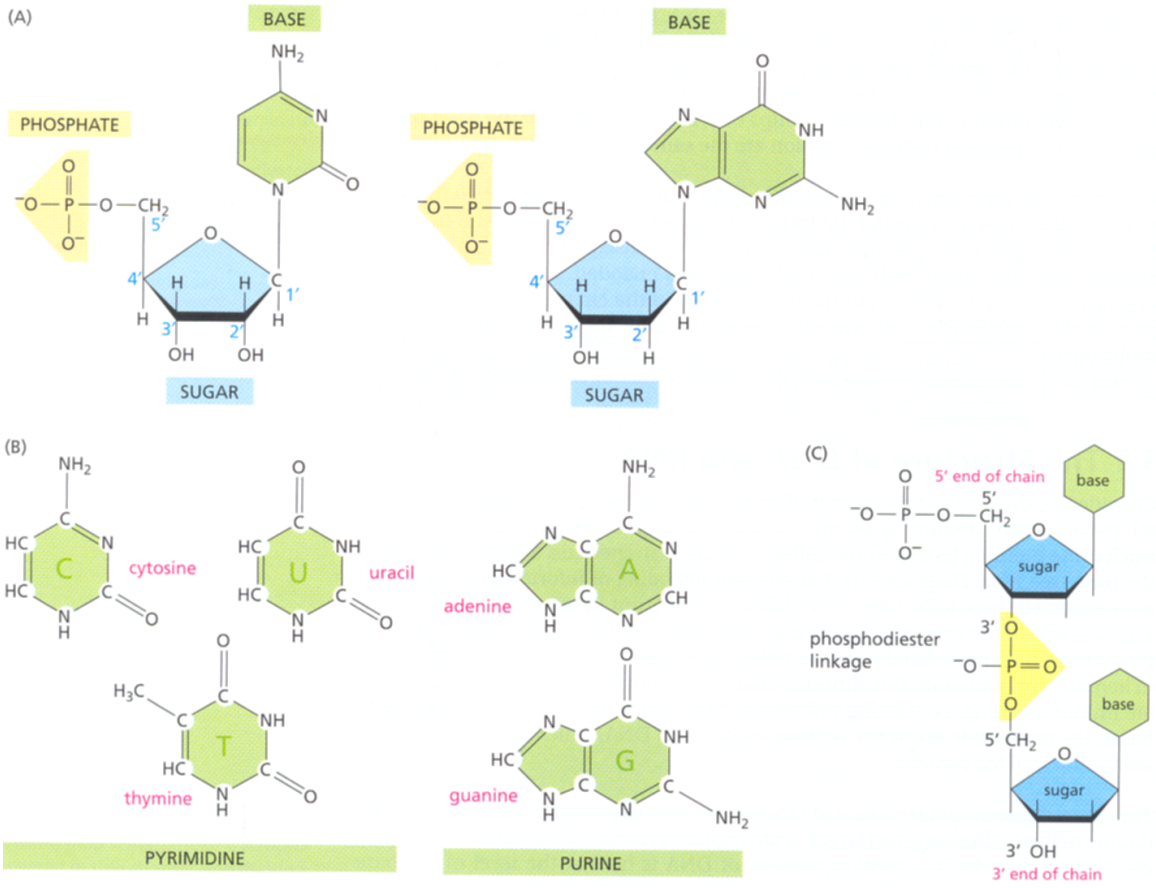
Die Disziplin Bioinformatik ist interdisziplinär ausgerichtet und wendet numerische, algorithmische, statistische und graphische Methoden aus der Mathematik und Informatik auf biologische und medizinische Datensätze an.
Die Bioinformatik unterteilt sich in mehrere Aufgabengebiete. Sie beschäftigt sich mit der Verwaltung und Integration biologischer Daten, der Sequenzanalyse, der Strukturbioinformatik und der Analyse von Daten aus Hochdurchsatzmethoden (*OMICS). Aufgrund der sehr großen Datenmengen, die in biologischen Experimenten generiert werden können, ist die Bioinformatik heute unentbehrlich. Die Entwicklung der Bioinformatik orientiert sich sehr stark an der Verfügbarkeit von neuen Datensätzen.
Die ersten geeigneten Daten kamen 1955 von F. Sanger durch die Sequenzanalyse von Rinderinsulin. Der italienische Arzt und Genetiker Luigi Luca Cavalli-Sforza benutzte 1963 erstmalig statistische Methoden, um phylogenetische Bäume zu rekonstruieren. Cavalli-Sforza sammelte auf mehreren Expeditionen Blutproben und genetisches Material und verband dies mit Studien, die auf linguistischen, kulturellen und archäologischen Daten aufbauten. Basierend auf den gesamten Daten erstellte er die evolutionären Stammbäume und genetische Landkarten, die die Verteilung der Gene auf die verschiedenen Kontinente zeigen.
Ein weiterer wichtiger Bereich ist seit etwa den 1980er Jahren die Analyse der Koordinaten in Proteinkristallstrukturen. Damals erkannte man, dass erst die Kenntnis der dreidimensionalen Struktur eines Proteins genauen Einblick in seine Funktion geben konnte.
Daher steht im ersten Teil der Vorlesung (Vorlesungen 1-4) die Sequenzanalyse im Vordergrund.

1.2 Sequenzanalyse

In der Bioinformatik stellt die Sequenzanalyse einen der klassischen Hauptanwendungsbereiche dar. Als Grundlage dienen entweder die DNA/RNA Nukleotidsequenzen, die aus den vier Nukleotiden Adenin, Cytosin, Guanin und Thymin in DNA und Uracil statt Thymin in RNA gebildet werden oder die Aminosäuresequenzen bestehend aus den zwanzig natürlich in Proteinen vorkommenden Aminosäuren (proteinogene Aminosäuren).
Gemeinsam bilden die Nukleotide durch Verknpfung ber Atombindungen die DNA bzw. RNA Stränge, wobei die Reihenfolge der Nukleobasen mit Hilfe des vier Buchstaben Codes (a, c, g, t/u) in einer Sequenz gespeichert wird.

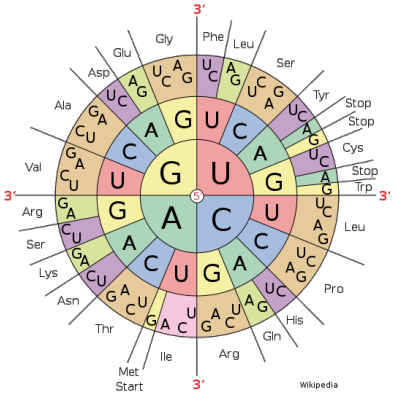
Bsp: 5’ ...atggccaggcaactttagtgctg... 3’

Verbunden sind die Nukleotide jeweils über eine Phosphatgruppe in Form eines Phosphodiesters.



In der Zelle wird mittels der bekannten Transkriptions- und Translationsmaschinerie diese Nukleotidsequenz in eine Aminosäuresequenz übersetzt. Jeweils drei aufeinanderfolgende Nukleotide, Nukleotid-Triplet oder auch Codon genannt, stehen für eine bestimmte Aminosäure. Um von einem Codon auf die jeweiligs kodierte Aminosäure schlieen zu knnen, benutzt man die sog. Codonsonne. Die Codonsonne ist von Innen nach Außen zu lesen und beschreibt so den gelesenen Code in 5' - 3' Richtung entlang der DNA.
Fast jedes Codon beschreibt eindeutig eine Aminosäure, Ausnahmen hierbei sind *tga, tag* und *taa*, die jeweils keine Aminosäure sondern ein Stopcodon kodieren. Umgekehrt ist es jedoch nicht möglich, einer Aminosäure ein bestimmtes Codon zuzuweisen. Eine Ausnahme bildet Methionin, das nur durch das Codon *atg* kodiert wird und das Startcodon eines Gens darstellt.
Die meisten Aminosäuren werden durch zwei bis vier verschiedene Codons kodiert, wobei die ersten beiden Nukleotide gleich bleiben. Die Ausnahme bilden hier, neben Methionin und den Stopcodons, die Aminosäuren mit jeweils sechs verschiedenen Codons:

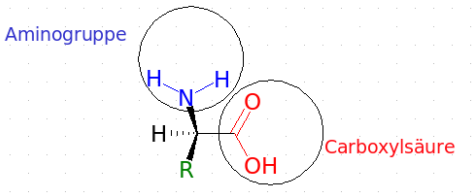
Codonsonne
- Serin (tc[a,g,c,t], agc, agt),
- Leucin (ct[a,g,c,t], tta, ttg) und
- Arginin (cg[a,g,c,t], aga, agg)



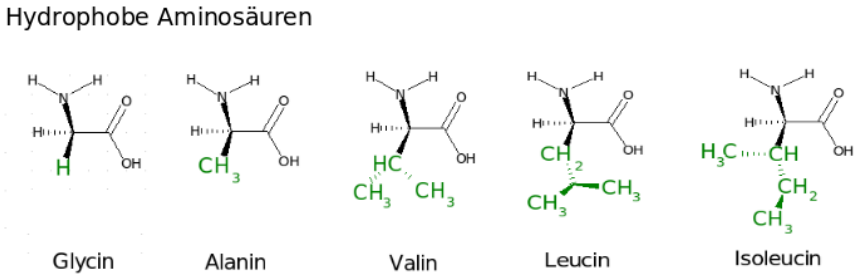
Die Aminosäuren bilden die Bausteine eines Proteins. Sie unterscheiden sich in Größe, elektrischer Ladung, Polarität, Form und Flexibilität. Aufgrund dieser physikochemischen Eigenschaften sind sie für die Faltung des Proteins und so auch für dessen Funktion verantwortlich.

1.3 Aminosäuren: Aufbau und Eigenschaften

Die Aminosäuren bestehen alle aus einem Grundgerüst: einer Aminogruppe und einer Carboxylgruppe, die über ein C-Atom miteinander verbunden sind und einer Seitenkette, die an diesem C-Atom hängt. Diese Seitenkette bestimmt die Spezifikation und die Eigenschaften der Aminosäure.

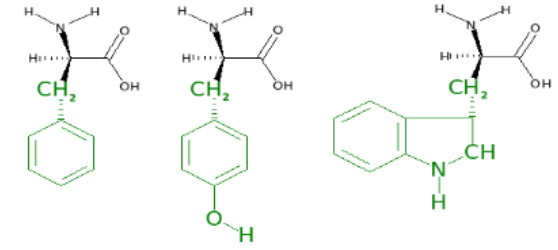


Eine Aminosäure besitzt nicht nur eine Eigenschaft, sondern mehrere. Im Folgenden teilen wir die Aminosäuren nach ihren Haupteigenschaften ein. Zum Beispiel gibt es kleine und große hydrophobe Aminosäuren.



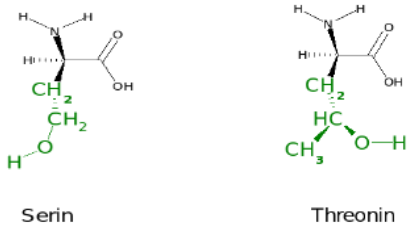
‚Hydrophob' bedeutet, dass diese Aminosäuren ungern in direktem Kontakt mit dem Lösungsmittel Wasser stehen. Daher kommen sie besonders im dicht gepackten Kern von gefalteten Proteinen vor.

Aromatische Aminosäuren

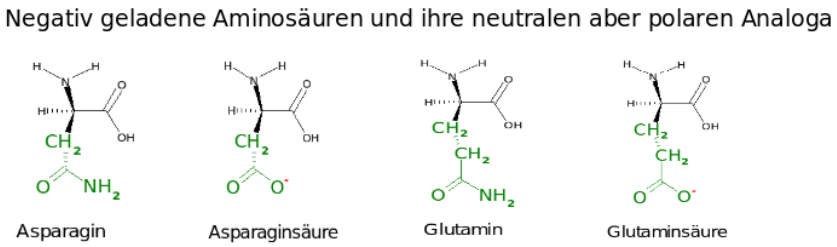


‚Aromatisch' bezieht sich auf das Vorkommen eines aromatischen Ringsystems in diesen Aminosäuren mit abwechselnden Einzel- und Doppelbindungen. Durch Überlappung der π-Orbitale aus den Doppelbindungen entsteht jeweils eine Elektronenwolke oberhalb und unterhalb des Ringsystems. Diese Ringsysteme sind wesentlich hydrophiler als gesättigte Ringe, in denen keine C=C Doppelbindungen vorkommen.
Die folgenden polaren und geladenen Aminosäuren sind ‚hydrophil', also wasserliebend. Diese Aminosäuren liegen vor allem auf der Oberfläche gefalteter Proteine in direktem Kontakt mit dem Lösungsmittel Wasser. Geladene Aminosäuren sind ebenfalls häufig in den aktiven Zentren von Enzymen positioniert. Zusammen mit den anderen polaren Aminosäuren sind sie dort an den katalytischen Reaktionen beteiligt. Außerdem binden geladene Aminosäuren entgegengesetzt geladene Ionen.

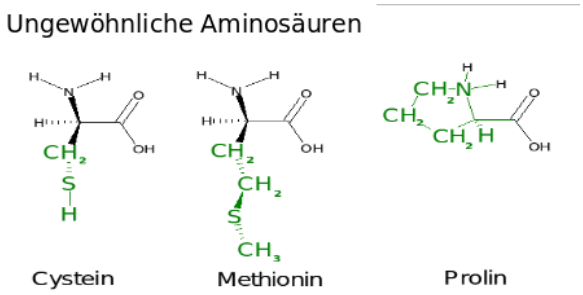
Aminosäuren mit polaren Hydroxylgruppen



Positiv geladene Aminosäuren



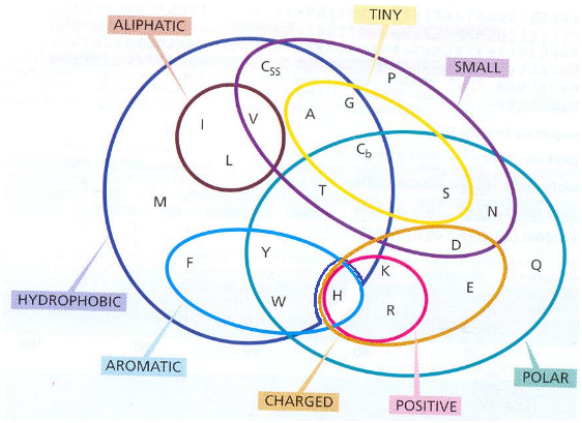
Zwei Aminosäuren enthalten Schwefel (Cystein und Methionin) und eine Aminosäure ist ungewöhnlich aufgebaut (Prolin). Cystein ist in der Lage Disulfidbrücken auszubilden, welche ein besonders stabiles Strukturelement gefalteter Proteine darstellt. Die ungewöhnliche Struktur bei Prolin ergibt sich durch die Verbindung der Seitenkette mit der Aminogruppe zu einem Ring.



Für die Aminosäure- Sequenzanalyse wird eine geeignete Notation benötigt. So gibt es zwei Abkürzungsformen für die Aminosäuren, einmal den Drei-Buchstaben Code und dann den Ein-Buchstaben Code.

Codierungstabelle Aminosäuren		
Name	3er-Code	1er-Code
Alanin	Ala	A
Arginin	Arg	R
Asparagin	Asn	N
Asparaginsäure	Asp	D
Cystein	Cys	C
Glutamin	Gln	Q
Glutaminsäure	Glu	E
Glycin	Gly	G
Histidin	His	H
Isoleucin	Ile	I
Leucin	Leu	L
Lysin	Lys	K
Methionin	Met	M
Phenylalanin	Phe	F
Prolin	Pro	P
Serin	Ser	S
Threonin	Thr	T
Tryptophan	Trp	W
Tyrosin	Tyr	Y
Valin	Val	V
Zusätzliche Codes:		
Asn/Asp		B
Gln/Glu		Z
Irgendeine Aminosäure		X

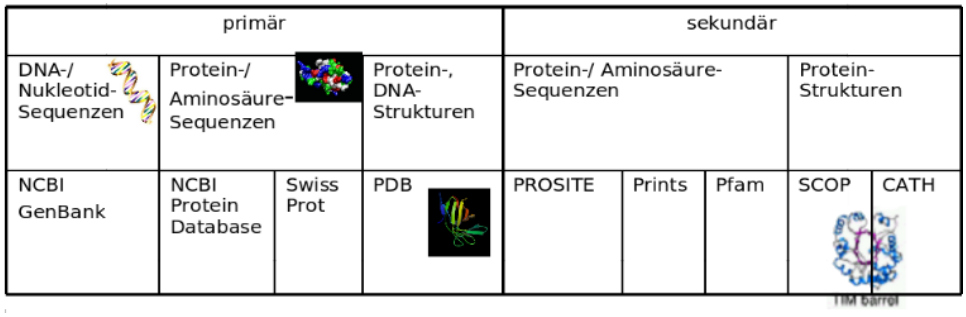
Für die Sequenzanalyse wird der Ein-Buchstaben Code verwendet. Für die erfolgreiche Teilnahme an dieser Vorlesung ist es unabdingbar, dass Sie sich diesen Code fest einprägen!
Das folgende Mengendiagramm zeigt die kompletten Eigenschaften für die einzelnen Aminosäuren. So ist zum Beispiel Histidin (H) den geladenen Aminosäuren zuzuordnen und zwar den positiv geladenen. Aufgrund des aromatischen Ringes in der Seitenkette wird Histidin aber auch den aromatischen Aminosäuren zugeordnet.



Woher erhält man nun die benötigten Daten, die Sequenzen, für die Analyse?

1.4 Sequenzdatenbanken

Der Begriff Datenbank stammt aus der Informatik und bezeichnet die strukturierte Speicherung von Datensätzen. Dadurch kann in rasantem Tempo gezielt auf einzelne Datensätze und somit Informationen zugegriffen werden. Sie ermöglichen so eine schnellere Suche, als wenn man zum Beispiel eine Textdatei immer von Anfang bis Ende durchsuchen muss.
Es gibt verschiedene öffentlich frei im Internet zugängliche Datenbanken, die weltweit für die Analyse biologischer Sequenzdaten verwendet werden. Dadurch wird ein weiterer Vorteilung von Datenbanken deutlich, und zwar die zentrale Speicherung und Bereitstellung der Datensätze, statt vieler kleiner, individueller Sammlungen.
‚Primäre Datenbanken' stellen die experimentell bestimmten Nukleotid- bzw Aminosäuresequenzen mit ihren Identifikationsnummern und Querverweisen zu anderen Datenbanken zur Verfügung.
‚Sekundäre Datenbanken' enthalten verarbeitete Daten, die durch Aufbereitung der Sequenzdaten gewonnen bzw. abgeleitet wurden.



1.4.1 NCBI GenBank

GenBank (*http://www.ncbi.nlm.nih.gov/Genbank*) ist eine öffentliche Nukleotid Sequenzdatenbank, die vom National Center for Biotechnology Information (NCBI) in den USA unterhalten wird.
Im Februar 2010 enthielt sie ca. 116 Mio. Sequenzeinträge von ca. 260.000 Organismen. Die Sequenzen werden von verschiedenen Laboratorien oder größeren Sequenzierungsprojekten eingereicht, wobei die eingereichten Sequenzen eine Mindestlänge von 50 Basenpaaren haben müssen. Wird die Sequenz in die Datenbank aufgenommen, erhält dieser neue Eintrag eine eindeutige Accession Number (Zugriffsnummer).
Eine tägliche automatische Daten-Synchronisation mit der „European Molecular Biology Laboratory Nucleotide Sequence Database"(EMBL, *http://www.ebi.ac.uk*) in Europa und der „DNA Data Bank of Japan" (DDBJ, *http://www.ddbj.nig.ac.jp*) sichert eine weltweite Abdeckung der Daten. Zu beachten ist, dass es oft mehrere Einträge für dieselbe Sequenz gibt, wenn diese mehrfach bestimmt und eingereicht wurde. Dies wird redundant genannt.

Hier eine Beispielausgabe von GenBank:

```
LOCUS       SCU49845     5028 bp    DNA             PLN       21-JUN-1999
DEFINITION  Saccharomyces cerevisiae TCP1-beta gene, partial cds, and Axl2p
            (AXL2) and Rev7p (REV7) genes, complete cds.
ACCESSION   U49845
VERSION     U49845.1  GI:1293613
KEYWORDS    .
SOURCE      Saccharomyces cerevisiae (baker's yeast)
  ORGANISM  Saccharomyces cerevisiae
            Eukaryota; Fungi; Ascomycota; Saccharomycotina; Saccharomycetes;
            Saccharomycetales; Saccharomycetaceae; Saccharomyces.
REFERENCE   1  (bases 1 to 5028)
  AUTHORS   Torpey,L.E., Gibbs,P.E., Nelson,J. and Lawrence,C.W.
  TITLE     Cloning and sequence of REV7, a gene whose function is required for
            DNA damage-induced mutagenesis in Saccharomyces cerevisiae
  JOURNAL   Yeast 10 (11), 1503-1509 (1994)
  PUBMED    7871890
REFERENCE   2  (bases 1 to 5028)
  AUTHORS   Roemer,T., Madden,K., Chang,J. and Snyder,M.
  TITLE     Selection of axial growth sites in yeast requires Axl2p, a novel
            plasma membrane glycoprotein
  JOURNAL   Genes Dev. 10 (7), 777-793 (1996)
  PUBMED    8846915
REFERENCE   3  (bases 1 to 5028)
  AUTHORS   Roemer,T.
  TITLE     Direct Submission
  JOURNAL   Submitted (22-FEB-1996) Terry Roemer, Biology, Yale University, New
            Haven, CT, USA
FEATURES             Location/Qualifiers
     source          1..5028
                     /organism="Saccharomyces cerevisiae"
                     /db_xref="taxon:4932"
                     /chromosome="IX"
                     /map="9"
     CDS             <1..206
                     /codon_start=3
                     /product="TCP1-beta"
                     /protein_id="AAA98665.1"
                     /db_xref="GI:1293614"
                     /translation="SSIYNGISTSGLDLNNGTIADMRQLGIVESYKLKRAVVSSASEA
                     AEVLLRVDNIIRARPRTANRQHM"
     gene            687..3158
                     /gene="AXL2"
     CDS             687..3158
                     /gene="AXL2"
                     /note="plasma membrane glycoprotein"
                     /codon_start=1
                     /function="required for axial budding pattern of S.
                     cerevisiae"
                     /product="Axl2p"
                     /protein_id="AAA98666.1"
                     /db_xref="GI:1293615"
                     /translation="MTQLQISLLLTATISLLHLVVATPYEAYPIGKQYPPVARVNESF
                     TFQISNDTYKSSVDKTAQITYNCFDLPSWLSFDSSSRTFSGEPSSDLLSDANTTLYFN
                     VILEGTDSADSTSLNNTYQFVVTNRPSISLSSDFNLLALLKNYGYTNGKNALKLDPNE
                     VDFSNKSNVNVGQVKDIHGRIPEML"
     gene            complement(3300..4037)
                     /gene="REV7"
     CDS             complement(3300..4037)
                     /gene="REV7"
                     /codon_start=1
                     /product="Rev7p"
                     /protein_id="AAA98667.1"
                     /db_xref="GI:1293616"
                     /translation="MNRWVEKWLRVYLKCYINLILFYRNVYPPQSFDYTTYQSFNLPQ
                     FVPINRHPALIDYIEELILDVLSKLTHVYRFSICIINKKNDLCIEKYVLDFSELQHVD
                     KDDQITTETEVFDEFRSSLNSLIMHLEKLPKVNDDTITFEAVINAILESLGHKLDRNR
                     RVDSLEEKAEIERDSNWVKCQEDENLPDNNGFQPPKIKLTSLVGSDVGPLIIHQFSEK
                     LISGDDKILNGVYSQYEEGESIFGSLF"
ORIGIN
        1 gatcctccat atacaacggt atctccacct caggtttaga tctcaacaac ggaaccattg
       61 ccgacatgag acagttaggt atcgtcgaga gttacaagct aaaacgagca gtagtcagct
      ...
     4861 ttctccactt cactgtcgag ttgctcgttt ttagcggaca aagatttaat ctcgttttct
     4921 ttttcagtgt tagattgctc taattctttg agctgttctc tcagctcctc atatttttct
     4981 tgccatgact cagattctaa ttttaagcta ttcaatttct ctttgatc
//
```

->LOCUS: enthält den locus Namen: hier SCU49845 zum Eingruppieren der Daten
Länge: 5028 bp
Typ: DNA,
GenBank division:
PLN -> Pflanzen, Pilze und Algen
Datum der letzten Änderung

->DEFINITION: Kurze Beschreibung der Sequenz, Organismus, Gen-/Protein Name, evtl. Funktion...

->ACCESSION: U49845

->SOURCE: Saccharomyces cerevisiae

->REFERENCE: Original-Veröffentlichung der in diesem Eintrag enthaltenen Sequenz

->FEATURES: Informationen über Gene und ihre Produkte, über biologisch relevante Regionen

Eine ausführliche Beschreibung der Ausgabe mit Tipps für Suchanfragen ist unter:

http://www. ncbi.nlm.nih.gov/Sitmap/samplerecord.html

zu finden.

-> komplette Nukleotid Sequenz

1.4.2 NCBI Protein Datenbank

Analog zu der NCBI GenBank ist die NCBI Protein Datenbank eine öffentliche, primäre Protein Sequenzdatenbank (*http://www.ncbi.nlm.nih.gov/protein*). Die Sequenzen der Datenbank sind aus den Proteindatenbanken:

- UniProt/KB,
- PIR (Protein Identification Resources),
- PDB (Protein Data Bank, Strukturen),
- Proteintranslationen aus der GenBank

und weiteren zusammengestellt.
Da in den einzelnen Datenbanken eine Proteinsequenz mehrfach vorkommen kann, ist die Protein Datenbank ebenso wie GenBank redundant. Der Vorteil dieser Proteindatenbank

ist, dass die Ergebnisse gleichzeitig die Links zu den originalen Datenbanken enthalten. Die Ausgabe ist gleich aufgebaut wie die Ausgabe von GenBank, (LOCUS, DEFINITION, ACCESSION, VERSION, DBSOURCE, ...), unter dem Punkt ORIGIN ist jedoch nicht die Nukleotidsequenz sondern die Aminosäuresequenz enthalten.

1.4.3 SwissProt - UniProt/KB

UniProtKB/Swiss-Prot (*http://www.uniprot.org*) ist eine gut geordnete Protein Sequenzdatenbank, die eine sehr gute Annotation, ein Minimum an redundanten Sequenzen und eine hohe Integration mit anderen Datenbanken anstrebt. Sie wird manuell erstellt, wurde 1986 etabliert und ist seit 2003 in dem UniProt Verbund. Dies ist eine Zusammenarbeit zwischen dem Swiss Institute of Bioinformatics (SIB), dem Department für Bioinformatik und Strukturelle Biologie der Universität Genf, dem Europäischen Bioinformatik Institut (EBI) und dem Protein Information Resource (PIR) des Medical Centers der Georgetown University.
UniProtKB/Swiss-Prot bildet mit seinem Computer generierten Pendant UniProtKB/-TrEMBL, die UniProt Knowledgebase (UniProtKB). UniProtKB/Swiss-Prot enthält aktuell 410.518 Einträge (Release 56.8 of 10-Feb-2009), UniProtKB/TrEMBL hingegen 7.157.600 (Release 39.8 of 10-Feb-2009).
Die Sequenzeinträge bestehen aus verschiedenen Notationen, jede mit ihrem eigenen Format. Um einen Standard der UniProt Knowledgebase zu erhalten, ist das Format so nah wie möglich an das Format der EMBL Nucleotide Sequence Database angelehnt.
Die UniProtKB/Swiss-Prot Datenbank unterscheidet sich von anderen Proteindatenbanken durch drei Kriterien:

1. Annotation:
   Die Daten werden manuell und kontinuierlich eingefügt. Sie beinhalten neben den Kerndaten (Sequenz, Referenz,..), soweit derzeit bekannt, die Beschreibungen von:

   - Funktion,
   - Posttranslationalen Modifikationen,
   - Domäne und aktive Zentren, Bsp: Calcium Bindestelle,
   - Sekundärstruktur,
   - Quartärstruktur,
   - Ähnlichkeit zu anderen Proteinen,
   - Krankheiten, die mit Mutationen in diesem Protein assoziiert werden und
   - Sequenz Konflikte, Varianten

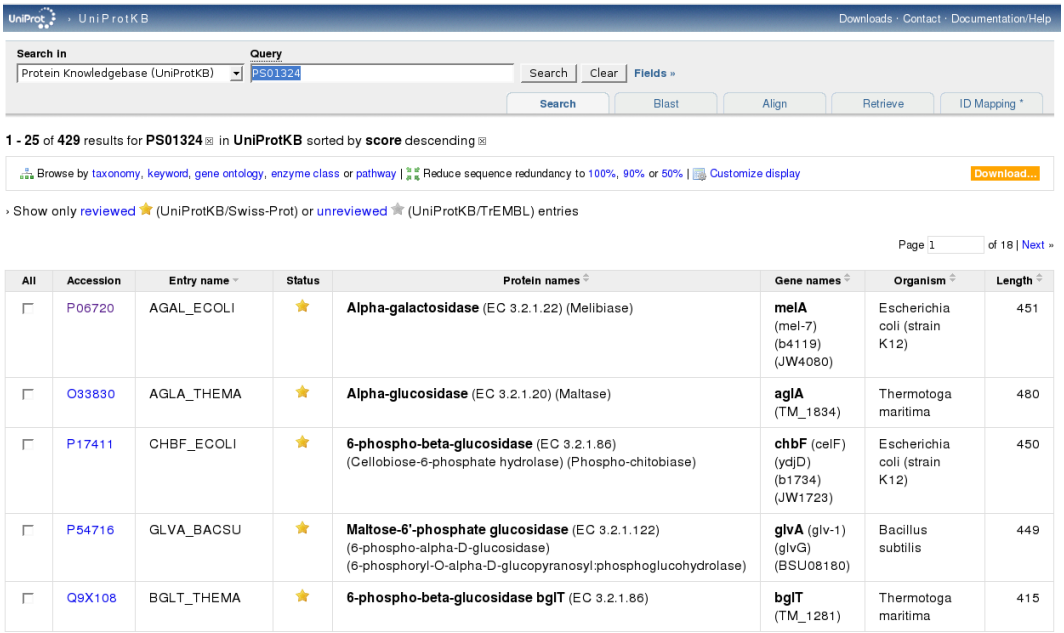
   Durch Abgleich mit Daten aus der Literatur können auch Eigenschaften annotiert werden, die nicht aus der Sequenz vorhergesagt werden können. Die Notation befindet sich in den Kommentaren (CC) der Eigenschaftstabelle (FT) und in den Keywörtern (KW).

2. Minimale Redundanz:
   Alle Proteinsequenzen, die durch dasselbe Gen kodiert werden, sind in einem Eintrag

zusammengefasst. Gefundene Differenzen aus verschiedenen Sequenzierungsberichten werden analysiert und in den Eigenschaftstabellen beschrieben.

3. Integration mit anderen Datenbanken:
   Die UniProtKB/Swiss-Prot Datenbank enthält zur Zeit Querverweise zu ca. 60 verschiedenen Datenbanken.

Hier eine Beispielsuche für PS01324 gegen die UniProt/KB:



All	Accession	Entry name	Status	Protein names	Gene names	Organism	Length
☐	P06720	AGAL_ECOLI	★	**Alpha-galactosidase** (EC 3.2.1.22) (Melibiase)	**melA** (mel-7) (b4119) (JW4080)	Escherichia coli (strain K12)	451
☐	O33830	AGLA_THEMA	★	**Alpha-glucosidase** (EC 3.2.1.20) (Maltase)	**aglA** (TM_1834)	Thermotoga maritima	480
☐	P17411	CHBF_ECOLI	★	**6-phospho-beta-glucosidase** (EC 3.2.1.86) (Cellobiose-6-phosphate hydrolase) (Phospho-chitobiase)	**chbF** (celF) (ydjD) (b1734) (JW1723)	Escherichia coli (strain K12)	450
☐	P54716	GLVA_BACSU	★	**Maltose-6'-phosphate glucosidase** (EC 3.2.1.122) (6-phospho-alpha-D-glucosidase) (6-phosphoryl-O-alpha-D-glucopyranosyl:phosphoglucohydrolase)	**glvA** (glv-1) (glvG) (BSU08180)	Bacillus subtilis	449
☐	Q9X108	BGLT_THEMA	★	**6-phospho-beta-glucosidase bglT** (EC 3.2.1.86)	**bglT** (TM_1281)	Thermotoga maritima	415

Wählt man nun den ersten Treffer mit der Identifikation P06720, erhält man Informationen zu Name und Ursprung, Protein Attribute, wie z.B. die Sequenzlänge, generelle Notation, Ontologien usw. Unter dem Punkt Cross-References sind weiterführende Links zu anderen Datenbanken, wie z.B. Prosite, PRINTS oder Pfam angegeben.

1.4.4 Prosite

PROSITE (*http://www.expasy.org/prosite*) ist eine sekundäre Protein Datenbank. Sie enthält Informationen über Proteinfamilien, Domänen und funktionelle Bereiche. PROSITE baut auf der Beobachtung auf, dass die meisten der in großer Anzahl vorkommenden, unterschiedlichen Proteine aufgrund ihrer Sequenzähnlichkeit in eine begrenzte Zahl von Proteinfamilien eingeteilt werden können.
Für mehr als 1000 Proteinfamilen oder Domänen befinden sich aktuell Pattern und Profile in PROSITE. Dies sind kurze konservierte Sequenzmotive (10-20 Aminosäuren), auch Signaturen genannt, werden durch ein multiples Sequenzalignment gewonnen (vgl. Vorlesung 3). Sie liefern in ihren Dokumentationen Hintergrundinformationen über Struktur und Funktion des Proteins. Dadurch hat jede Familie/ Domäne hat eine spezifische Signatur.

Eine Beispielausgabe für die Eingabe PDOC01027 (entspricht P06720 aus UniProt/KB) ergibt:

Home ScanProsite ProRule Documents Downloads Links Funding

PROSITE documentation PDOC01027

Glycosyl hydrolases family 4 signature

Description:

It has been shown [1,2,E1] that the following glycosyl hydrolases can be, on the basis of sequence similarities, classified into a single family:

- Escherichia coli and Bacillus subtilis α-galactosidase (EC 3.2.1.22) (melibiase) (gene melA).
- Thermotoga maritima α-glucosidase (EC 3.2.1.20) (gene aglA).
- Probable 6-phospho-β-glucosidases (EC 3.2.1.86) (gene celF) from Escherichia coli and Bacillus subtilis.
- Maltose-6'-phosphate glucosidases (EC 3.2.1.122) (6-phospho-α-D- glucosidase) from Bacillus subtilis (gene glvG) and Fusobacterium mortiferum (gene malH).
- Bacillus subtilis protein lplD.

These enzymes require NAD and a divalent ion for their activity. They are proteins of about 50 Kd. As a signature pattern we selected a conserved region located in the central section. This region does not contain residues directly shown to be important for the catalytic activity.

Expert(s) to contact by email:
Henrissat B.

Last update:
April 2006 / Pattern revised.

Technical section:

PROSITE method (with tools and information) covered by this documentation:

GLYCOSYL_HYDROL_F4, PS01324; **Glycosyl hydrolases family 4 signature** (PATTERN)

Consensus pattern:	**[PS] - x - [SAC] - x - [LIVMFY](2) - [QN] - x(2) - N - P - x(4) - [TA] - x(9,11) - [KRD] - x - [LIV] - [GN] - x - C**
Sequences known to belong to this class detected by the pattern:	ALL
Other sequence(s) detected in Swiss-Prot:	NONE.

- Retrieve an alignment of Swiss-Prot true positive hits:
  Clustal format, color, condensed view / Clustal format, color / Clustal format, plain text / Fasta format
- Retrieve the sequence logo from the alignment
- Taxonomic tree view of all Swiss-Prot/TrEMBL entries matching PS01324
- Retrieve a list of all Swiss-Prot/TrEMBL entries matching PS01324
- Scan Swiss-Prot/TrEMBL entries against PS01324
- view ligand binding statistics

Matching PDB structures: 1UP4 1UP6 1UP7 [ALL]

References:

1	*Authors*	Henrissat B.
	Title	*A classification of glycosyl hydrolases based on amino acid sequence similarities.*
	Source	Biochem. J. 280:309-316(1991).
	PubMed ID	1747104
2	*Authors*	Thompson J., Pikis A., Ruvinov S.B., Henrissat B., Yamamoto H., Sekiguchi J.
	Title	*The gene glvA of Bacillus subtilis 168 encodes a metal-requiring, NAD(H)-dependent 6-phospho-alpha-glucosidase. Assignment to family 4 of the glycosylhydrolase superfamily.*
	Source	J. Biol. Chem. 273:27347-27356(1998).
	PubMed ID	9765262
E1	*Source*	http://www.expasy.org/cgi-bin/lists?glycosid.txt

Copyright:

PROSITE is copyright. It is produced by the Swiss Institute of Bioinformatics (SIB). There are no restrictions on its use by non-profit institutions as long as its content is in no way modified. Usage by and for commercial entities requires a license agreement. For information about the licensing scheme send an email to license@isb-sib.ch or see: http://www.expasy.org/prosite/prosite_license.htm.

Miscellaneous:

View entry in original PROSITE document format
View entry in raw text format (no links)

Alle Glycosyl Hydrolasen können auf der Basis ihrer Sequenzähnlichkeit in eine gemeinsame Proteinfamilie klassifiziert werden:

- *Escherichia coli* und *Bacillus subtilis* -galactosidase (EC 3.2.1.22) (melibiase) (gene melA),

- *Thermotoga maritima* -glucosidase (EC 3.2.1.20) (gene aglA),
- Probable 6-phospho--glucosidases (EC 3.2.1.86) (gene celF) von *Escherichia coli* und *Bacillus subtilis*,
- Maltose-6'-phosphate glucosidases (EC 3.2.1.122) (6-phospho-D- glucosidase) von *Bacillus subtilis* (gene glvG) und *Fusobacterium mortiferum* (gene malH),
- *Bacillus subtilis* protein lplD.

Enzyme aus der Proteinfamilie Glycosyl Hydrolasen werden durch die Signatur (Consensus Pattern):

[PS] - x - [SAC] - x - [LIVMFY](2) - [QN] - x(2) - N - P - x(4) - [TA] - x(9,11) - [KRD] - x - [LIV] - [GN] - x - C

beschrieben.
Dies bedeutet, an Position 1 können nur Phenylalanin oder Serin auftreten, an Position 2 können alle Aminosäuren auftreten, an Position 3 können Serin, Alanin oder Cystein auftreten.

1.4.5 PRINTS

PRINTS (*http://bioinf.man.ac.uk/dbbrowser/PRINTS*) ist eine weitere sekundäre Protein Datenbank. PRINTS ist eine Übersicht von Protein-fingerprints. Ein Fingerprint ist eine Gruppe von konservierten Sequenzmotiven (Motiv = immer wieder auftretendes Sequenzstück), dessen Funktion und/ oder Struktur bekannt sind und die so eine Proteinfamilie charakterisieren. Die Datenbank enthält 1950 FINGERPRINTS, die 11.625 einzelne Motive kodieren (letzte Version 39.0, Februar, 2009).
Hier ein Teil der Ausgabe des Ergebnisses für P06720 aus UniProt/KB:

Sequence Titles

AGAL_BACSU ALPHA-GALACTOSIDASE (EC 3.2.1.22) (MELIBIASE) - BACILLUS SUBTILIS.
AGAL_ECOLI ALPHA-GALACTOSIDASE (EC 3.2.1.22) (MELIBIASE) - ESCHERICHIA COLI.
CELF_BACSU PROBABLE 6-PHOSPHO-BETA-GLUCOSIDASE (EC 3.2.1.86) - BACILLUS SUBTILIS.
CELF_ECOLI 6-PHOSPHO-BETA-GLUCOSIDASE (EC 3.2.1.86) - ESCHERICHIA COLI.
GLVG_BACSU MALTOSE-6'-PHOSPHATE GLUCOSIDASE (EC 3.2.1.122) (6-PHOSPHO-ALPHA-D- GLUCOSIDASE) - BACILLUS SUBTILIS.
GLVG_ECOLI PROBABLE 6-PHOSPHO-BETA-GLUCOSIDASE (EC 3.2.1.86) - ESCHERICHIA COLI.
LPLD_BACSU LPLD PROTEIN - BACILLUS SUBTILIS.
MALH_FUSMR MALTOSE-6'-PHOSPHATE GLUCOSIDASE (EC 3.2.1.122) (6-PHOSPHO-ALPHA-D- GLUCOSIDASE) - FUSOBACTERIUM MORTIFERUM.

Q88026 PUTATIVE GLUCOSIDASE - STREPTOMYCES COELICOLOR.

Scan History

OWL29_3 1 100 NSINGLE
SPTR37_9f 2 14 NSINGLE

Initial Motifs

Motif 1 width=16

Element	Seqn Id	St	Int	Rpt
SVVVAGGGSTFTPGIV	GLVG_ECOLI	5	5	-
SIVIAGGGSTFTPGIV	GLVG_BACSU	7	7	-
KVVTIGGGSSYTPELL	D90816S	6	6	-
KITFIGAGSTIFVKNI	AGAL_ECOLI	6	6	-
KVVTIGGGSSYTPELL	CELF_ECOLI	6	6	-
KIVTIGGGSSYTPELV	CELF_BACSU	6	6	-
KIAYIGGGSQGWARSL	LPLD_BACSU	11	11	-

Motif 2 width=17

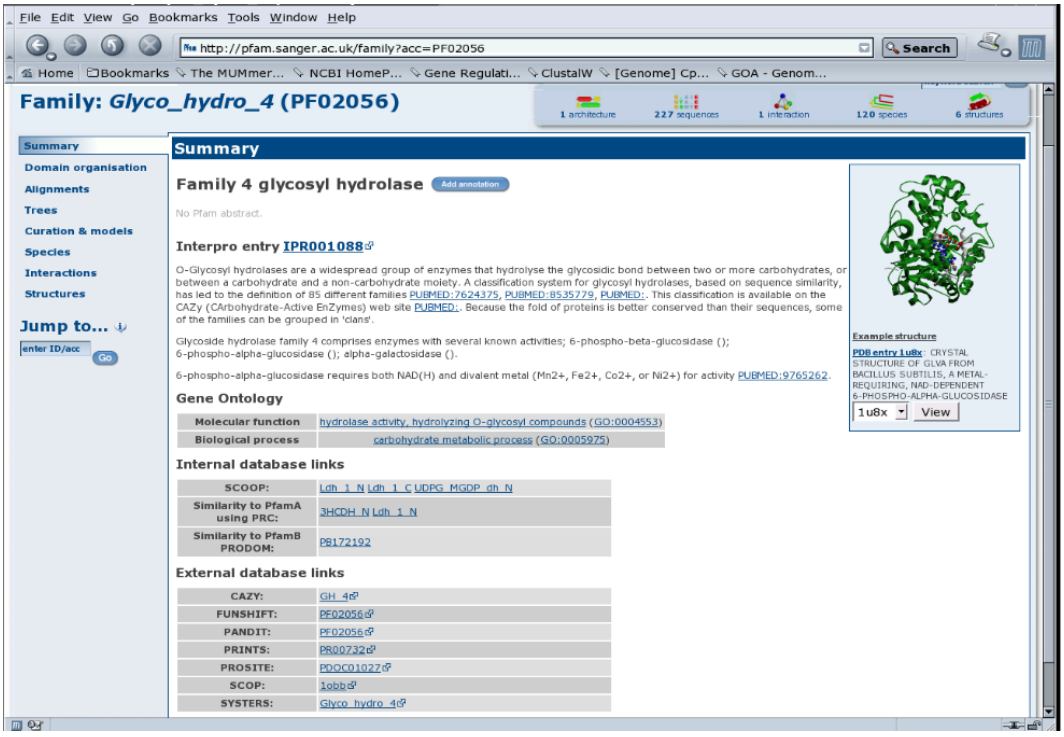
Element	Seqn Id	St	Int	Rpt
ALKDADFVTTQLRVGQL	D90816S	77	55	-
ALSAADIVIISILPGSL	LPLD_BACSU	76	49	-
ALEDADFVVVAFQIGGY	AGAL_ECOLI	75	53	-
ALKDADFVTTQLRVGQL	CELF_ECOLI	77	55	-
AFSDVDFVMAHIRVGKY	GLVG_ECOLI	74	53	-
AFTDVDFVMAHIRVGKY	GLVG_BACSU	76	53	-
ALKDADFVTTQFRVGLL	CELF_BACSU	77	55	-

Motif 3 width=14

Element	Seqn Id	St	Int	Rpt
LDERIPLSHGYLGQ	D90816S	98	4	-
LDEQIPLKYGVVGQ	GLVG_BACSU	97	4	-
LDEKIPLRHGVVGQ	GLVG_ECOLI	95	4	-
TDFEVCKRHGLEQT	AGAL_ECOLI	97	5	-
LDERIPLSHGYLGQ	CELF_ECOLI	98	4	-
KDERIPLKYGVIGQ	CELF_BACSU	98	4	-

1.4.6 Pfam

Pfam (*http://pfam.sanger.ac.uk*) ist eine weitere sekundäre Protein Datenbank. Sie basiert auf der UniProtKB Sequenzdatenbank, dem NCBI GenPept und auf Sequenzen aus ausgewählten Metagenomic Projekten. 74% aller Proteinsequenzen haben mindestens einen Pfam-Eintrag. Sie enthält 11.912 Proteinfamilien (Oktober 2009). Pfam ist also eine große Ansammlung von Proteinfamilien, die durch Multiple Sequenz Alignments (s. Vorlesung 3) und Hidden Markov Modelle (s. Vorlesung 4) repräsentiert werden. Es gibt zwei Komponenten von Pfam: Pfam-A und Pfam-B. Pfam-A wird manuell gewartet, wodurch der Qualittsstandard der Einträge hoch ist. Pfam-B wird automatisch generiert, was zu einer geringeren Qualität der Einträge führt. Trotzdem können Pfam-B Einträge funktionelle konservierte Regionen identifizieren, wenn keine Pfam-A Einträge vorliegen. Hier die Fortsetzung aus dem bereits bekannten Beispiel:

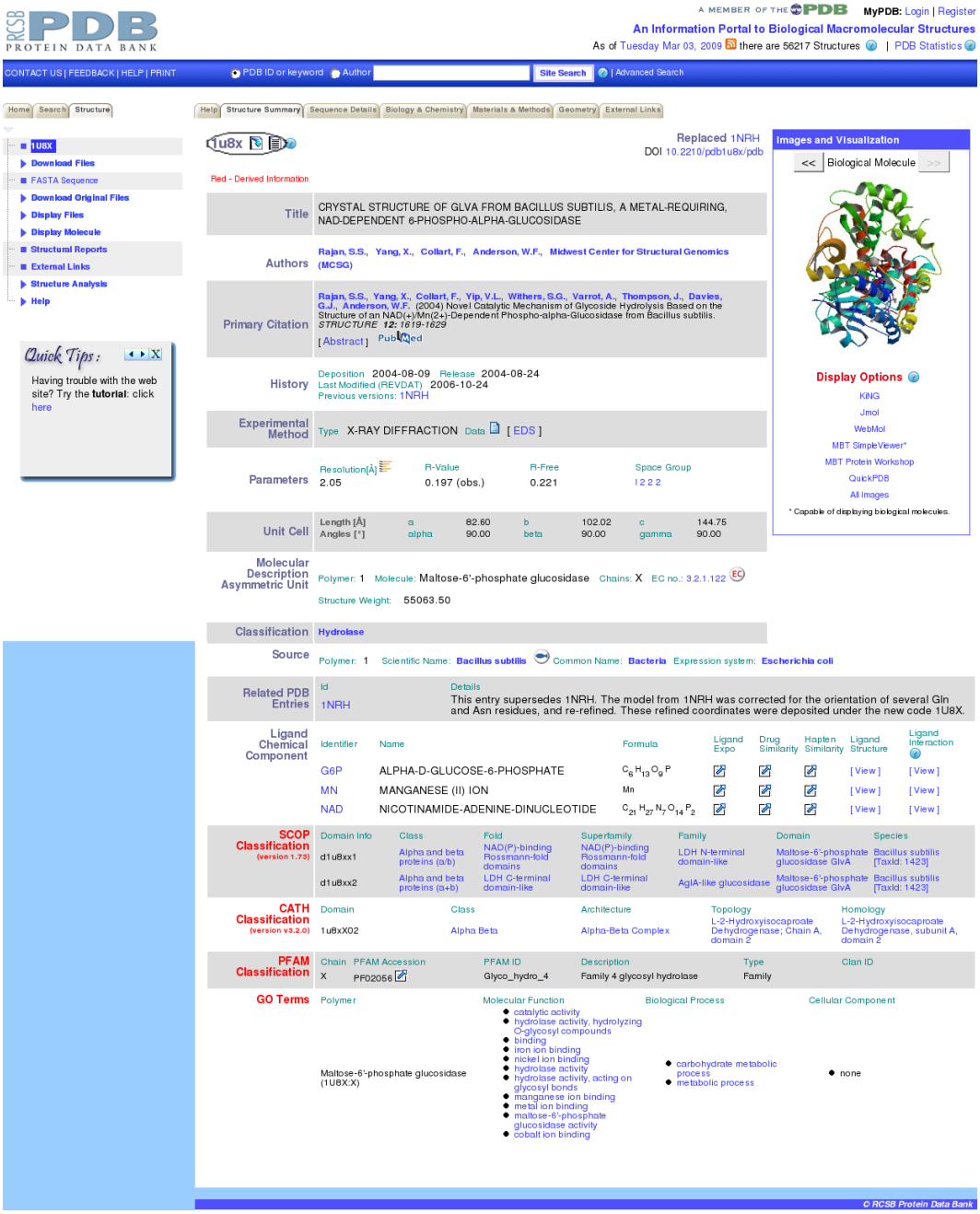


Man erhält als Ergebnis, dass der gesuchte Identifier zu der Familie der Glycosyl Hydrolase gehört. Als nächstes wird der Querverweis zu einer weiteren Datenbank, InterPro und der dazugehörige Identifier, genannt. Desweiteren ist eine Beschreibung der Glycosyl Hydrolase zu lesen. Das Bild in der rechten Ecke zeigt eine Bespielstruktur für eine Glycosyl Hydrolase, das *glvA* des *bacillus subtilis*, mit dem Identifier 1u8x.pdb für die PDB-Struktur.

1.4.7 RCSB PDB

Die primäre Protein Daten Bank (PDB) der Research Collaboratory for Structural Bioinformatics (RCSB, USA, *http://www.rcsb.org/pdb*) enthielt im April 2010 64623 3D

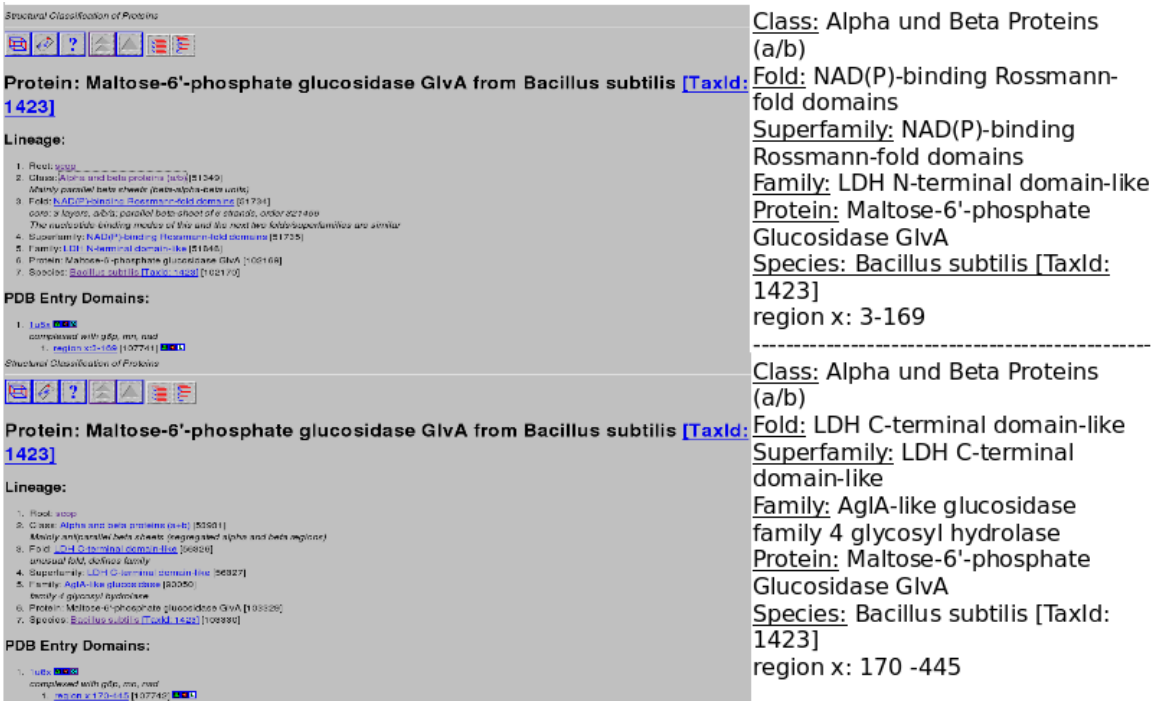
Strukturen von großen biologischen Molekülen (in der Mehrzahl Proteine). Die Datenbank enthält die Atomkoordinaten von experimentell bestimmten Proteinstrukturen, Nukleinsäuren und Komplexen. Damit lassen sich diese Makromoleküle detailiert visualisieren. Die RCSB PDB gehört zu der wwPDB Organisation (WorldWide PDB), einem Zusammenschluss (2003) aus der RCSB PDB, der Macromolecular Structure Database des Europäischen Bioinformatik Instituts von EMBL (PDBe) und der Protein Data Bank Japan (PDBj). Die Biological Magnetic Resonance Data Bank (BMRB, USA) kam 2006 noch hinzu. Nach der Suche nach dem PDB Identifier 1u8x in der Protein Data Bank (PDB) erhält man folgende Ausgabe:



Direkt neben dem Identifier oben links befinden sich zwei kleine Schaltflächen (Buttons). Hier kann man sich die 3D-Daten des Proteins anzeigen lassen bzw. herunterladen. Mit Hilfe eines graphischen Anzeigetools (Jmol, Rasmol, BallView, ...) kann man sich die 3D Struktur des Moleküls als ganzes oder gewünschte Teilregionen darstellen lassen. Ferner sind in der Ausgabe die Informationen über das Molekül enthalten. Die Einträge in Rot (SCOP, Cath, Pfam, GOTerms) sind Informationen aus anderen Datenbanken, die dieses Molekül klassifizieren.

1.4.8 SCOP

SCOP (Structural Classification Of Proteins) ist eine sekundäre Protein Struktur Datenbank. Mit SCOP können Proteine strukturell klassifiziert werden. SCOP beinhaltet 38221 PDB Einträge und 110800 Domänen. (1.75, Juni 2009)
(*http://scop.mrc-lmb.cam.ac.uk/scop/data/scop.b.html*)
Die Proteine sind nach Klasse, Faltung, Superfamilie und Familie klassifiziert.
Hier die SCOP-Ausgabe zu dem Beispiel 1u8x:



Die Ausgabe enthält die beiden weiterführenden Links zu den beiden unten gezeigten Seiten:
Protein: Maltose-6'-phosphate glucosidase GlvA from Bacillus subtilis [TaxId: 1423] [c.2.1.5]
Protein: Maltose-6'-phosphate glucosidase GlvA from Bacillus subtilis [TaxId: 1423] [d.162.1.2]

1.4.9 CATH

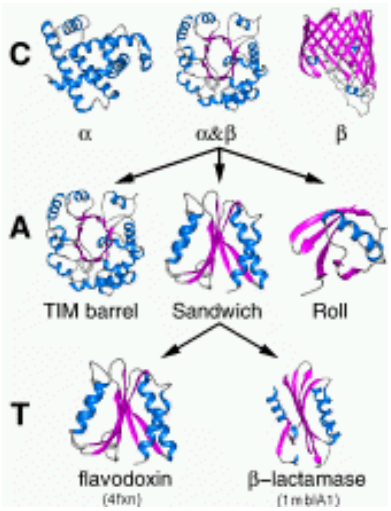
Die CATH Datenbank (*http* : *//www.cathdb.info*) stellt eine weitere hierarchische Domänen Klassifikation von Protein Strukturen aus der Protein Data Bank (PDB, Berman et al. 2003) zur Verfügung, wobei nur Kristall-Strukturen mit einer Auflösung besser als 4 Angström und NMR Strukturen werden betrachtet. Alle Nicht-Proteine, theoretischen Modelle und Strukturen mit mehr als 30% „C_α“ Atomen werden von CATH ausgeschlossen, wodurch gewhrleistet wird, dass nur hochaufgelöste Strukturen enthalten sind. Die Protein Strukturen werden anhand einer Kombination von automatischen und manuellen Prozessen klassifiziert. Es gibt vier Hauptklassen in dieser Hierarchie:

- Class:
  Wird gemäß der Sekundärstruktur und der Faltung bestimmt: 3 Hauptklassen: Mainly Alpha, mainly Beta und Alpha Beta

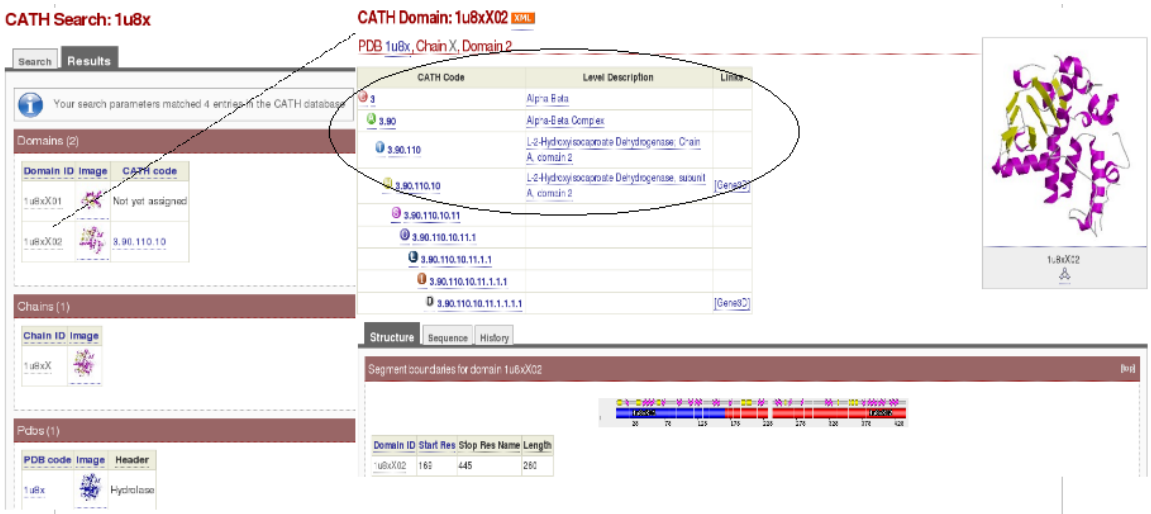
- Architecture:
  beschreibt die gesamte Form der Domänstruktur anhand der Orientierung der Sekundärstruktur Bsp: barrel

- Topology:
  Strukturen werden nach der gleichen Topologie oder Faltung im Kern der Domäne gruppiert

- Homologous superfamily:
  Protein Domänen mit demselben evolutionären Vorfahren werden zu homologen Einheiten gruppiert

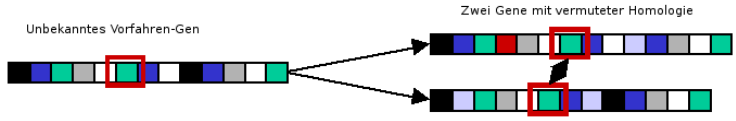


Cath Ausgabe für 1u8x:

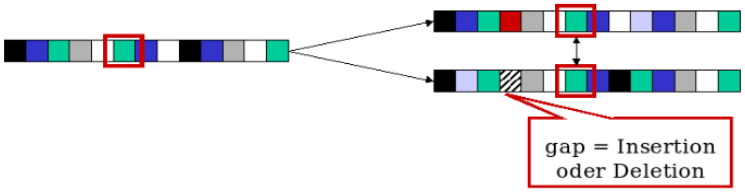


2 Sequenzanalyse

Verschiedene Organismen enthalten oft Gene aus ein und derselben Proteinfamilie, die von einem gemeinsamen Vorfahren abstammen. Diese Abstammung nennt man Homologie. Der Haupteinsatzbereich für Bioinformatikmethoden ist heutzutage der Vergleich zweier oder mehrerer Nukleotid- bzw. Aminosäuresequenzen. Dabei muss ständig der Grad an Ähnlichkeit zwischen den Einträgen der beiden Sequenzen in einer alignierten, d.h. sich entsprechenden Position beurteilt werden. Die Ähnlichkeit wird anhand eines Sequenzalignments quantitativ bestimmt. Je ähnlicher die beiden Sequenzen sind, desto näher sind im Allgemeinen die beiden Organismen verwandt. Die Sequenzen für die Sequenzanalyse werden aus neuen Experimenten gewonnen oder den bereits existierenden Datenbanken (Vorlesung 1) entnommen. Die ähnlichen Regionen können die ganze Sequenz, oder Teile von ihr umfassen. Man spricht in diesen Fällen von einem lokalen Alignment für ein Teilstück oder von einem globalen Alignment, in dem versucht wird die Sequenzen in ihrer gesamten Länge zu alignieren.



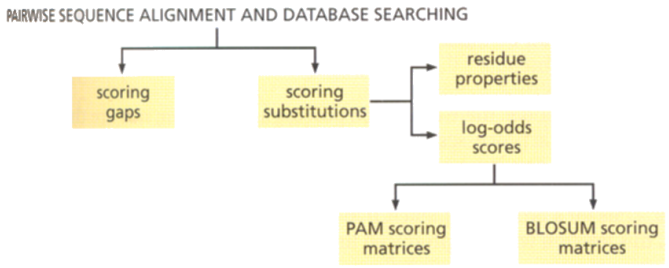
In diesem Beispiel unterscheiden sich die Sequenzen rechts nicht nur durch die unterschiedliche Länge sondern auch in der Zusammensetzung. Durch das Einfügen eines Gaps (Lücke) in die untere Sequenz kann der Längenunterschied ausgeglichen werden. Damit wird der Effekt einer vermuteten Insertion im oberen Gen oder einer Deletion im unteren Gen aufgehoben. Die Position des Gaps wird so gewählt, dass für dieses Sequenzpaar ein optimales Alignment, d.h. der beste Ähnlichkeitswert zwischen den beiden Sequenzen, entsteht. Dies geschieht, wenn ein möglichst hoher Anteil an Entsprechungen zwischen einzelnen Bausteinen beider Sequenzen existiert.



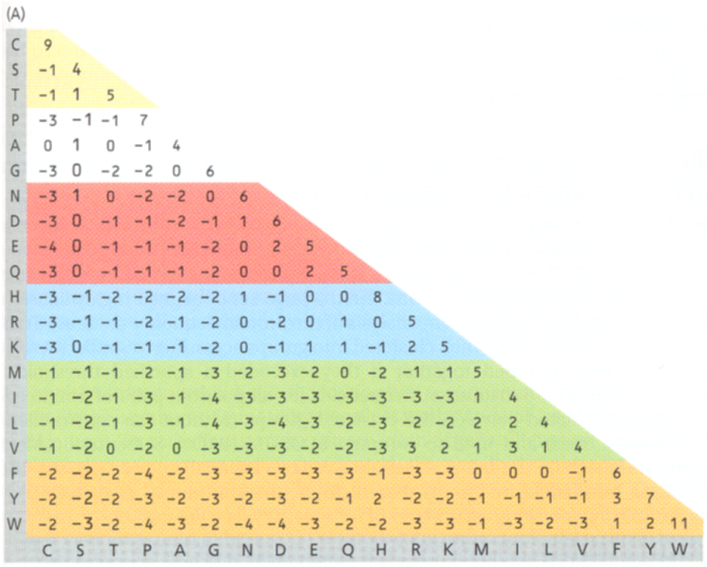
Den Ähnlichkeitswert zwischen den verschiedenen Sequenzen berechnet man anhand von Austauschmatrizen.

2.1 Aminosäure - Austauschmatrizen

Die beiden etablierten Kategorien für Austauschmatrizen sind die PAM Matrix und die BLOSUM Matrix. Sie unterscheiden sich in der Berechnungsweise ihrer Einträge. Innerhalb beider Kategorien gibt es unterschiedliche Matrizen, die jeweils für das Alignment von mehr oder weniger ähnlichen Sequenzen geeignet sind, z.B. PAM1, PAM120, PAM250, und BLOSUM45, BLOSUM62 und BLOSUM80.



Flow Diagram 5.1
The key concept introduced in this section is that if alignments of two sequences are assigned a quantitative score based on evolutionary principles then meaningful comparisons can be made. Several alternative approaches have been suggested, resulting in a number of different scoring schemes including those which account for insertions and deletions.



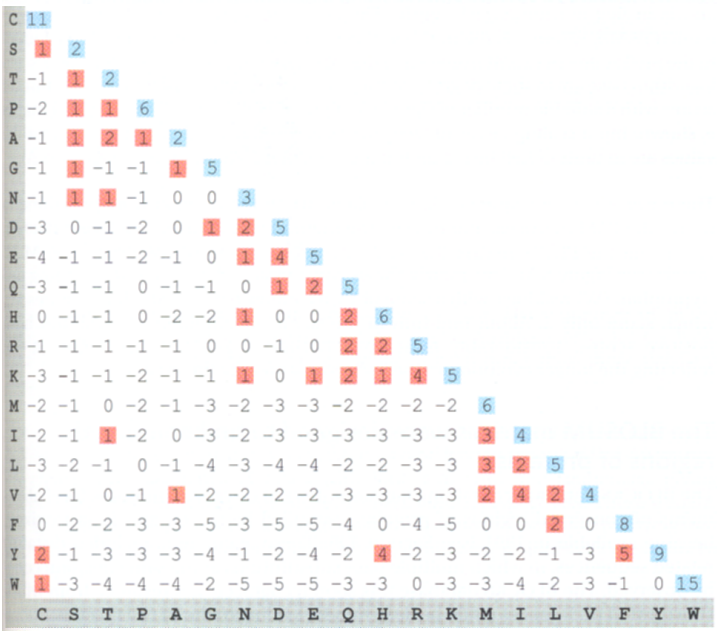


Figure 4.4
Amino acid substitution scoring matrices. (A) The BLOSUM-62 matrix and (B) the PAM120 substitution matrix. Each cell represents the score given to a residue paired with another residue (row × column). The values are given in half-bits, as discussed in Section 5.1. The colored shading indicates different physicochemical properties of the residues (see Figure 2.3): small and polar, yellow; small and nonpolar, white; polar or acidic, red; basic, blue; large and hydrophobic, green; aromatic, orange.

Figure 5.3
The PET91 version of the PAM250 substitution matrix. Scores that would be given to identical matched residues are in blue; positive scores for nonidentical matched residues are in red. The latter represent pairs of residues for which substitutions were observed relatively often in the aligned reference sequences.

Die beiden Matrizen enthalten positive und negative Werte. Je höher der Wert, desto häufiger wurde dieses Aminosäurepaar untereinander in einem Sequenzalignment gefun-

den,d.h. desto wahrscheinlicher ist es also, dass gerade diese beiden gegeneinander ausgetauscht wurden. Zum Beispiel ist der höchste Eintrag in der gezeigten PAM250 - Matrix der Wert 15 für den „Austausch“ von Trp gegen Trp, also die Konservierung von Trp. Daraus folgt, dass Tryptophan eine besonders gut konservierte Aminosäure ist.

2.1.1 Informationstheorie

Definition der Information:

$$H(p) = \log_2(\frac{1}{p}) = -\log_2(p) \tag{2.1.1}$$

p steht dabei für die Wahrscheinlichkeit einer Antwort.

Die DNA-Sequenzen enthalten die vier Buchstaben A, C, G und T. Wenn das Auftreten für jedes Nukleotid gleich wahrscheinlich ist, hat jedes Auftreten des Nukleotids die Wahrscheinlichkeit $p = \frac{1}{n}$ (mit n = 4), die Information jedes Ereignisses ist dann: $-\log_2(n)$. Der formale Name für die mittlere Information pro Ereignis ist die Entropie. Wenn die Ereignisse nicht gleich wahrscheinlich sind, muss man die Information jedes Ereignisses mit dessen Wahrscheinlichkeit gewichten. Damit ergibt sich die Definition der **Shannon Entropie**:

$$H(p) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2.1.2}$$

In einem zufälligen Stück DNA (z.B. GCAT) hat jedes Nukleotid bei Gleichverteilung die Wahrscheinlichkeit $p = \frac{1}{4}$.
Damit ergibt sich die Entropie als:

$-\sum \frac{1}{4}(-\log_2(\frac{1}{4}))$

$= -((0.25)(-2) + (0.25)(-2) + (0.25)(-2) + (0.25)(-2)) = 2$

Ein DNA Stück aus 90 % A oder T und 10 % C oder G hat jedoch eine kleinere Entropie: Für A und T ergibt sich eine Gesamtwahrscheinlichkeit von $p = \frac{9}{10}$, bei Gleichverteilung jeweils eine Wahrscheinlichkeit von $p = \frac{4.5}{10}$ für A oder T und für C oder G ergibt sich jeweils $p = \frac{0.5}{10}$.

$-(2 * \frac{4.5}{10} * (-\log_2(\frac{4.5}{10})) + 2 * \frac{0.5}{10} * (-\log_2(\frac{0.5}{10})))$

$= -(2(0.45)(-1.15) + 2(0.05)(-4.32)) = 1.47$

Der „am höchsten ungeordnete“ Zustand mit der größten Entropie liegt also bei einer Gleichverteilung vor. Wenn es 9 von 10 Positionen A oder T sind, erhält man durch die Bestimmung der Sequenz im Verhältnis dazu „weniger an Information“.

2.1.2 Dayhoff/PAM Matrix

Margaret O. Dayhoff stellte die beobachteten Austauschhäufigkeiten der Aminosäuren (Ähnlichkeit) zwischen verwandten Sequenzen als *log_2 odds* oder *lod score* dar. Der *lod score* einer Aminosäure wird berechnet als:
Logarithmus zur Basis 2 ($\log_2$) von dem Verhältnis der beobachteten Häufigkeit für ein Aminosäurepaar durch die zufällig für das Aminosäurepaar erwartete Häufigkeit. Die allgemeine Formel zur Berechnung des *lod scores* zweier Aminosäuren i und j lautet:

$$s_{ij} = \log_2(\frac{q_{ij}}{p_i p_j}) \tag{2.1.3}$$

Hierbei ist q_{ij} die beobachtete Häufigkeit (Paarungsfrequenz) des Aminosäurepaares aus den zwei Aminosäuren i und j. p_i und p_j sind die individuellen Häufigkeiten (Wahrscheinlichkeiten) für das Auftreten von i und j.

Daraus ergeben sich folgende drei Wertebereiche:

$$\text{Lod scores} = \begin{cases} < 0, \text{ unwahrscheinlicher Austausch,} \\ 0, \text{ beobachtete und erwartete Häufigkeiten sind gleich,} \\ > 0, \text{ ein Austauschpaar tritt häufiger auf als zufällig erwartet} \end{cases}$$

Beispiel:

1. Die relative Häufigkeiten von Methionin und Leucin in zwei Gesamtsequenzen eines Alignments oder in der Gesamtdatenbank seien $p_M = 0.01$ und $p_L = 0.1$. Daher ist die Wahrscheinlichkeit, an einer bestimmten Position eines Alignments ein Methionin zu finden $0.01 = 1\%$ bzw. $0.1 = 10\%$ für Leucin. Durch zufällige Paarung erwartet man $p_M * p_L = \frac{1}{100} * \frac{1}{10} = \frac{1}{1000}$ Austausche zwischen den Aminosäuren Methionin und Leucin. Die Wahrscheinlichkeit, dass ein Paar aus Methionin und Leucin in einem Sequenzalignment direkt übereinander steht ist damit also $0.001 = 0.1\%$.
   Beträgt die beobachtete Paarungshäufigkeit $q_{ML} = \frac{1}{500}$, ergibt sich für das Verhältnis der Häufigkeiten 2/1. Im Logarithmus zur Basis 2 ergibt sich ein *lod score* s_{ML} von $+1$.

2. Wenn die relative Häufigkeit von Arginin ebenfalls $p_R = 0.1$ ist und die Paarung mit Leu mit einer Häufigkeit q_{LR} von $\frac{1}{500}$ auftritt, dann ergibt sich für ein Arg-Leu Paar ein *lod score* s_{RL} von -2.322. Entsprechend der vorherigen Werte-Übersicht ist also der Austausch Met-Leu häufiger als zufällig erwartet, d.h. eher günstig und der Austausch Arg-Leu eher ungünstig. Dies paßt zu den physikochemischen Eigenschaften der drei Aminosäuren. Leucin und Methionin sind beide hydrophob, wogegen Arginin positiv geladen ist.

Gewöhnlich multipliziert man die Werte mit einem Skalierungsfaktor und rundet sie dann auf ganzzahlige Werte. Diese Werte sind dann in den Austauschmatrizen PAM und BLOSUM tabellarisiert.

Die so erstellten Austauschmatrizen dienen der Bewertung eines Alignments. Für den Vergleich von zwei Proteinsequenzen haben die Matrizen die Dimension 20 x 20, da es ja 20 verschiedene Aminosäuren gibt, die gegeneinander ausgetauscht werden können. Die Einträge geben die Wahrscheinlichkeit an, mit der eine bestimmte Aminosäure gegen eine andere Aminosäure ausgetauscht werden kann. Aufgrund der funktionellen und strukturellen Anforderungen der entsprechenden Proteinstruktur ist der Austausch von Aminosäuren mit ähnlichem Charakter (z.B. Leu und Ile) wahrscheinlicher (Austausch erhält einen höheren Wert) als der Austausch zweier Aminosäuren mit unterschiedlichem Charakter (z.B. Ile, Asp). Aus mechanistischen Gründen heraus werden zudem einige Aminosäuren leichter gegeneinander ausgetauscht als andere, besonders wenn sie ähnliche Codon Sequenzen besitzen, so dass eine kleine Mutation auf DNA Ebene den Austausch bewirkt. Da meist nicht eindeutig ist, welche Aminosäure die Sequenz des gemeinsamen Vorfahrens an dieser Position enthielt, wird nicht zwischen einem Austausch zwischen z.B. Ile-Asp und Asp-Ile unterschieden. Dadurch ist die Matrix symmetrisch und besitzt so die Form einer Dreiecksmatrix.

Margaret O. Dayhoff nutzte für ihre Berechnung einen Datensatz mit eng verwandten Proteinsequenzen (> 85% Idendität), da diese zweifelsfrei aligniert werden konnten. Mit den erhaltenen Austausch-Frequenzen erstellte sie die Dayhoff Matrix, die auch PAM 1 Matrix genannt wird (**PAM** = **P**oint **A**ccepted **M**utation) . Die 1 beschreibt hier den evolutionären Abstand der Sequenzen. PAM1 bedeutet, dass es genau 1 Punktmutation in 100 Residuen gibt oder anders, dass die Sequenzen zu 99% identisch sind.
Aus der PAM 1 Matrix kann man Matrizen für größere evolutionäre Entfernungen herstellen, indem man die Matrix einfach mehrfach mit sich selbst multipliziert. So steht PAM250 dafür, dass 2.5 Mutationen pro Residue vorkommen. Obwohl man dann eigentlich erwarten würde, dass alle Positionen in den beiden Sequenzen voneinander verschieden sind, sind immer noch etwa 20% der Positionen identisch. Dies liegt daran, dass manche Positionen einfach noch nicht ausgewählt wurden, bzw. in einer Position der Effekt der Mutation durch entsprechende Rückmutation aufgehoben wurde. Die PAM250 ist die Standard(Default)-Matrix in vielen Sequenzanalyse-Programmen.

Beispielsequenzen für eine Bewertung:
Sequenz 1: TCCPSIVARSN
Sequenz 2: SCCPSISARNT

Wenn sich zwei Sequenzen in zwei (oder mehreren) Positionen unterscheiden, möchte man die Wahrscheinlichkeit berechnen, dass Änderung A an Position 1 auftritt UND Änderung B an Position 2 (usw). Man braucht also $\log(A * B)$, wobei das Malzeichen für die UND-Verknüpfung steht.

Es gilt jedoch stets: $\log(AB) = \log(A) + \log(B)$

Da die PAM Matrix bereits die Log Werte der Austauschwahrscheinlichkeiten enthält, ist die Bewertung (Score) des Alignments daher einfach die Summe aller Bewertungen für

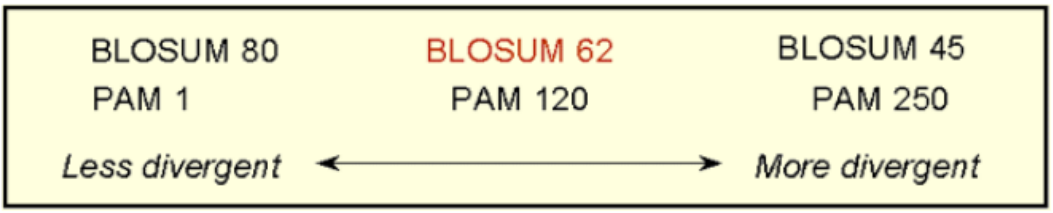
die Paare an Aminosäuren (Nukleinsäuren) des Alignments:

Sequenz 1:	T	C	C	P	S	I	V	A	R	S	N	
Sequenz 2:	S	C	C	P	S	I	S	A	R	N	T	
	1	11	11	6	2	4	-1	2	5	1	1	= 43 (PAM250)

Hier drückt der Score 43 die Wahrscheinlichkeit aus (Logarithmiert zur Basis 2), dass an Position 1 die Mutation T->S auftrat UND an Position 2 die Aminosäure C unverändert blieb UND ... UND an der letzten Position der Austausch N->T stattfand.

2.1.3 BLOSUM Matrix

Ein anderer Weg wurde von S. Henikoff und J.G. Henikoff eingeschlagen. Sie verwendeten lokale multiple Sequenzalignments von entfernter verwandten Sequenzen, also mit geringerer Ähnlichkeit. Dadurch haben sie gegenüber PAM den Vorteil, dass sie auf größere Datenmengen zurückgreifen können, da es mehr Sequenzen gibt, die entfernt miteinander verwandt sind, als nah verwandte Sequenzen. Ein weiterer Vorteil ist die Robustheit von Multiplen Sequenzalignments.
Die **BLOSUM** Matrizen (**BLO**cks **SU**bstitution **M**atrix) basieren auf der BLOCKS Datenbank. Die BLOCKS Datenbank verwendet das Konzept von Blöcken (lückenlose Aminosäure-Signaturen), die charakteristisch für eine Proteinfamilie sind. Aus den beobachteten Mutationen innerhalb dieser Blöcke wurden Austauschwahrscheinlichkeiten für alle Aminosäurepaare berechnet und für eine *log odds* BLOSUM Matrix benutzt. Man erhält unterschiedliche Matrizen indem man die untere Schranke des verlangten Grads an Identität variiert: z.B. wurde die BLOSUM80 Matrix aus Blöcken mit $> 80\%$ Identität abgeleitet.
Die Zahl hinter BLOSUM kennzeichnet somit, wie ähnlich die Sequenzen des Datensatzes waren, auf deren Grundlage die Matrizen berechnet wurden. So gilt für eine niedrige PAM und eine hohe BLOSUM, dass die Sequenzen eng miteinander verwandt waren. Umgekehrt gilt für eine hohe PAM und eine niedrige BLOSUM eine entfernte Verwandtschaft.



In den meisten Programmen sind die BLOSUM62 und PAM250 Matrizen als Standardeinstellung vorgeschlagen.

Sequenz 1:	T	C	C	P	S	I	V	A	R	S	N	
Sequenz 2:	S	C	C	P	S	I	S	A	R	N	T	
	1	11	11	6	2	4	-1	2	5	1	1	= 43 (PAM250)
	1	9	9	7	4	4	-2	4	5	1	0	= 42 (BLOSUM62)

2.2 Algorithmen zum Alignieren von Sequenzen

Neben den Substitutionsmatrizen ist für das Alignieren zweier Sequenzen auch die Gap Bewertung wichtig. Dadurch wird festgelegt, wann ein Gap eingefügt wird und wann nicht. Man unterscheidet dabei, ob eine Lücke neu aufgemacht werden muss oder ob die Lücke erweitert wird. Dies entspricht der biologischen Erfahrung. Insertionen/Deletionen liegen meist in Proteinloops (Schleifen) auf deren Oberfläche. Dabei ist die Länge der Insertion/Deletion nicht sehr entscheidend. Man bevorzugt Alignments mit wenigen, längeren Gaps.
Die gebräuchlichen Alignment-Programme haben Standardeinstellungen, die erstmals verwendet werden sollten, z.B. BLAST (Kosten um ein Gap zu öffnen = 5, um ein Gap zu erweitern = 2), ClustalW (Gap öffnen = 10, Gap erweitern = 0.05) und FASTA (Gap öffnen = -10 und Gap erweitern = -2).

Um nun zwei Sequenzen zu alignieren werden gute Algorithmen benötigt. Zwei häufig verwendete Algorithmen sind der Algorithmus von Needleman & Wunsch und der Algorithmus von Smith & Waterman. Diese Algorithmen basieren auf dynamischer Programmierung.

Als ***dynamische Programmierung*** bezeichnet man Algorithmen, die zur Lösung von Optimierungsproblemen verwendet werden. Dabei wird das Gesamtproblem in viele kleine Teilprobleme aufgeteilt, die zuerst einzeln bearbeitet werden. Mit den optimalen Lösungen dieser Teilprobleme wird eine optimale Lösung zu einem nächst größeren Teilproblem erstellt. Mit diesen nächst größeren optimalen Teillösungen verfährt man genauso, bis das Gesamtproblem optimal gelöst ist.

2.2.1 Needleman-Wunsch Algorithmus

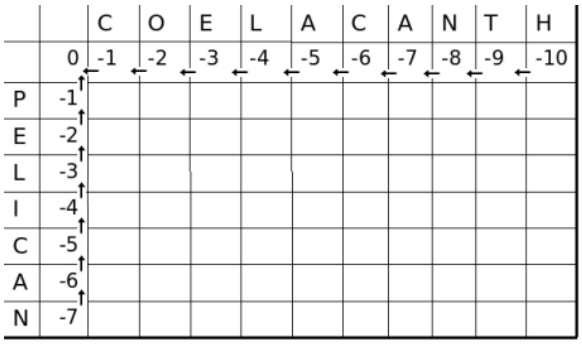
Der Needleman-Wunsch Algorithmus liefert mittels dynamischer Programmierung das bestmögliche globale (komplette Sequenz) Alignment zweier beliebiger Sequenzen. Der Algorithmus liefert die Lösung mit maximaler Bewertung der Ähnlichkeit, wobei Insertionen und Deletionen und somit das Einfügen von Gaps erlaubt sind.
Der Algorithmus basiert auf einer Matrizendarstellung, in der die beiden zu vergleichenden Sequenzen gegeneinander in eine Matrix aufgetragen werden, die für jedes mögliche Residuenpaar (Basen oder Aminosäuren) einen Wert enthält. Das optimale Alignment entspricht dem Pfad mit maximaler Bewertung durch die Matrix.
Der Algorithmus besteht aus drei Schritten:

1. Die Matrix wird initialisiert,
2. Die Matrix wird aufgefüllt,
3. Ein Traceback (Rückwärts)-Algorithmus bestimmt den besten Pfad

Als Beispiel sollen die beiden Wörter „COELACANTH“ der Länge $m = 10$ und „PELICAN“ der Länge $n = 7$ aligniert werden. In diesem Beispiel verwenden wir +1 für die

Bewertung eines Treffers (zwei gleiche Aminosäuren), -1 für verschiedene Aminosäuren und -1 für einen Gap.

1. Konstruiere eine $(m + 1)$ x $(n + 1)$ Matrix. Ordne den Elementen der ersten Zeile und Spalte die Werte -m * Gap und -n * Gap zu. Die Zeiger (*Pointer*) dieser Felder zeigen zurück zum Ursprung.



2. Fülle alle Matrizenfelder mit Werten und Zeigern auf. Benutze dazu simple Operationen, die die Werte der diagonalen, vertikalen, und horizontalen Nachbarzellen einschließen. Berechne dazu:

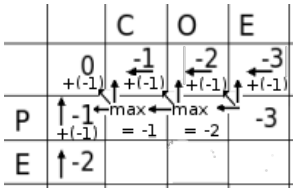
   1. ***match score***: Wert der Diagonalzelle links oben + Wert des Alignments in der aktuellen Zelle (+1 oder -1)
   2. ***horizontal gap score***: Wert der linken Zelle + gap score (-1)
   3. ***vertical gap score***: Wert der oberen Zelle + gap score (-1)

   Trage in die Zelle das Maximum dieser drei Werte ein. Der Zeiger dieser Zelle zeigt in Richtung des maximalen Werts, d.h. in die Zelle, deren Wert verwendet wurde.
   Beispiel:
   Der Eintrag für das Paar der beiden ersten Buchstaben (C,P) ist der maximale Wert aus:

   1. matchscore: Wert der Diagonale links oben = 0 + den Wert des Alignments = -1, da keine Übereinstimmung,
   2. horizontal score: Der Wert von links + (-1) = -2,
   3. vertical score: Der Wert von oben + (-1) = -2

   Die maximale Bewertung für die Zelle (C,P) ist daher: max (-1, -2, -2) = -1

   Analog werden die Einträge für die Paare (P,O), (E,E):
   (O,P): max(-1+(-1), -1 +(-1), -2+(-1)) = max (-2, -2, -3) = -2,
   (E,E): max(-2 + 1, -2 +(-1), -3+(-1)) = max (-1, -3, -4) = -1
   und alle anderen berechnet.



Da ein maximaler Wert aus mehreren Richtungen kommen kann, z.B. bei (O,P), muss eine Konvention festgelegt werden, die die Richtung, in die der Zeiger zeigen soll, bestimmt. Gewöhnlich ist das die Diagonale, da dies einem Alignment ohne Gap entspricht.

		C	O	E	L	A	C	A	N	T	H
	0	←-1	←-2	←-3	←-4	←-5	←-6	←-7	←-8	←-9	←-10
P	↑-1	↖-1	↖-2	↖-3	↖-4	↖-5	↖-6	↖-7	↖8	↖-9	↖10
E	↑-2	↖-2	↖-2	↖-1	←-2	←-3	←-4	←-5	←-6	←-7	←-8
L	↑-3	↖-3	↖-3	↑-2	↖0	←-1	←-2	←-3	←-4	←-5	←-6
I	↑-4	↖-4	↑-4	↑-3	↑-1	↖-1	↖-2	↖-3	↖4	↖-5	↖-6
C	↑-5	↖-3	←-4	↑-4	↑-2	↖-2	↖0	←-1	←-2	←-3	←-4
A	↑-6	↑-4	↖-4	↖-5	↑-3	↖-1	↑-1	↖1	←0	←-1	←-2
N	↑-7	↑-5	↖-5	↖-5	↑-4	↑-2	↖-2	←0	↖2	←-1	←-0

3. Traceback (Rückwärtssuche): Starte in der Ecke rechts unten. Folge den Pfeilen bis in die Ecke Links oben. Folge dabei jeweils den maximalen Werten. Diagonale Pfeile beschreiben dabei den Treffer, horizontale und vertikale Pfeile sind die Positionen, in denen ein Gap eingefügt wird.

		C	O	E	L	A	C	A	N	T	H
	0	←-1	←-2	←-3	←-4	←-5	←-6	←-7	←-8	←-9	←-10
P	↑-1	↖-1	↖-2	↖-3	↖-4	↖-5	↖-6	↖-7	↖8	↖-9	↖10
E	↑-2	↖-2	↖-2	↖-1	←-2	←-3	←-4	←-5	←-6	←-7	←-8
L	↑-3	↖-3	↖-3	↑-2	↖0	←-1	←-2	←-3	←-4	←-5	←-6
I	↑-4	↖-4	↑-4	↑-3	↑-1	↖-1	↖-2	↖-3	↖4	↖-5	↖-6
C	↑-5	↖-3	←-4	↑-4	↑-2	↖-2	↖0	←-1	←-2	←-3	←-4
A	↑-6	↑-4	↖-4	↖-5	↑-3	↖-1	↑-1	↖1	←0	←-1	←-2
N	↑-7	↑-5	↖-5	↖-5	↑-4	↑-2	↖-2	←0	↖2	←-1	←-0

Das Globale Alignment in diesem Beispiel ist:

S1:	C	O	E	L	A	C	A	N	T	H
S2:	–	P	E	L	I	C	A	N	–	–

2.2.2 Smith-Waterman Algorithmus

Der Smith-Waterman Algorithmus ist im Gegensatz zum Needleman-Wunsch Algorithmus ein lokaler Alignment-Algorithmus, d.h. er findet die besten lokalen Alignments. Dabei ist der Smith-Waterman Algorithmus lediglich eine Vereinfachung des Needleman-Wunsch Algorithmus mit folgenden drei Veränderungen:

1. Die Matrixränder werden anstelle von absteigenden Werten auf 0 gesetzt.

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0										
E	0										
L	0										
I	0										
C	0										
A	0										
N	0										

2. Der Maximalwert sinkt nie unter 0, Zeiger werden nur für Werte größer 0 eingezeichnet. Da die Anfangswerte mit 0 anstelle der negativen Werte für die Gap Penalties initialisiert werden, bleibt bei negativem Eintrag für einen möglichen Gap der Maximale Wert = 0.

3. Traceback beginnt bei dem größten Wert der Matrix und endet bei dem letzten Wert ungleich 0.

		C	O	E	L	A	C	A	N	T	H
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0	0	0
L	0	0	0	0	2	1	0	0	0	0	0
I	0	0	0	0	1	1	0	0	0	0	0
C	0	1	0	0	0	0	2	0	0	0	0
A	0	0	0	0	0	1	0	3	2	1	0
N	0	0	0	0	0	0	0	1	4	3	2

Das lokale Alignment für das Beispiel lautet:

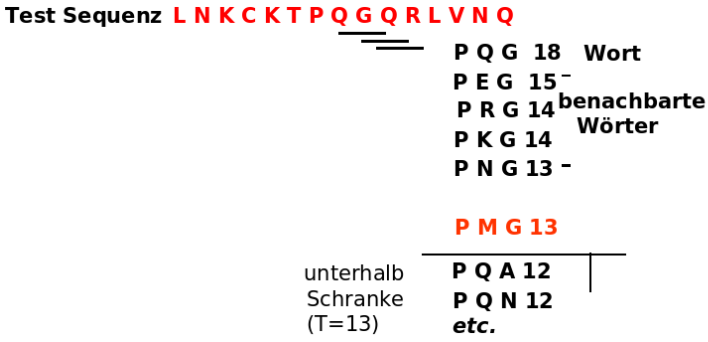
S1: E L A C A N
S2: E L I C A N

2.2.3 BLAST

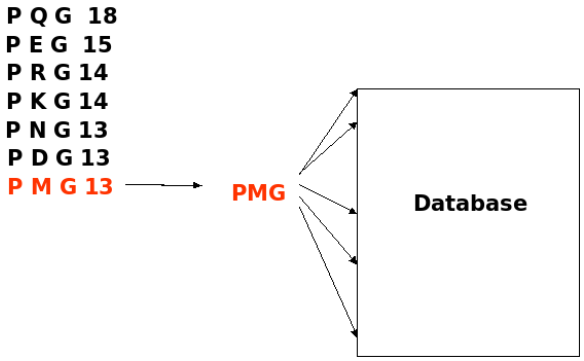
BLAST = **B**asic **L**ocal **A**lignment **S**earch **T**ool.
Das Programm BLAST ist eines der weltweit am häufigsten verwendeten Computerprogramme! BLAST findet das am besten bewertete lokale optimale Alignment einer Testsequenz mit allen Sequenzen einer Datenbank. Hinter dem Programm steckt ein sehr schneller Algorithmus, 50 mal schneller als dynamische Programmierung. Da BLAST eine vorindizierte Datenbank benutzt, kann es verwendet werden um sehr große Datenbanken zu durchsuchen. Für die meisten Zwecke ist es ausreichend sensititv und selektiv. BLAST ist robust und die Standardeinstellungen können üblicherweise verwendet werden.

Algorithmus:

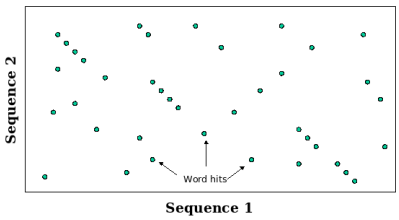
1. Für ein gegebenes Wort der Länge w (gewöhnlich 3 für Proteine) und eine gegebene Bewertungs-Matrix, erzeuge eine Liste aller Worte (w-mers), die eine Bewertung $> T$ erhalten, wenn man sie mit dem w-mer der Eingabe vergleicht.

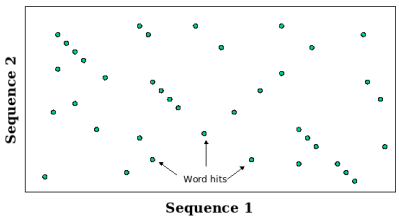


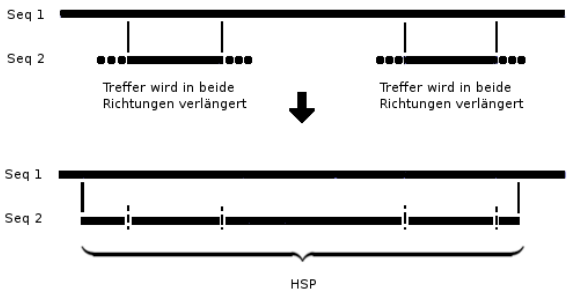
2. Zu jedem benachbarten Wort gibt es Zeiger auf alle Positionen in der Datenbank, in denen das Wort enthalten ist (*hit list*). Dies ist die Vorindizierung der Datenbank.



3. Im dritten Schritt (*Seeding*-Prozess) werden die Treffer (*Seed*) Schritt für Schritt verlängert, indem solange Residuenpaare hinzugefügt werden bis die zusätzliche Bewertung kleiner als ein Schrankenwert (*cut-off*) ist. Dabei beeinflusst die Wahl des Cut-offs den Seeding Prozess.







Nachdem die Ausdehnung beendet wurde, wird das Alignment so „zurückbeschnitten“, dass es die maximale Bewertung erhält. Das Alignmentstück mit dem besten Score wird als *High Scoring Segment Pair* (HSP) bezeichnet.

Die Anzahl an Alignments (E), die man während einer Suche in einer Sequenzdatenbank zufällig erhalten würde, ist proportional zur Größe des Suchraums ($m * n$), mit der Länge der Suchsequenz m und der Anzahl der Sequenzen n in der durchsuchten Sequenzdatenbank.

$$E \; \alpha \; m * n$$

Die gefundenen Sequenzen in BLAST werden zusammen mit einem Erwartungswert E angegeben. Dieser gibt an, wie signifikant der Treffer ist.
E-Wert: $E = P*$ Anzahl der Sequenzen in der Datenbank
Der P-Wert gibt die Wahrscheinlichkeit an, mit der die Bewertung eines Alignments zufällig zustande kommen kann. Je näher P bei Null liegt, desto größer ist die Sicherheit, dass ein gefundener Treffer ein richtiger Treffer (d.h. eine homologe Sequenz) ist.
E entspricht der Anzahl an Alignments einer bestimmten Bewertung, die man zufällig in einer Sequenz-Datenbank dieser Grösse erwartet (wird z.B. für ein Sequenzalignment $E = 10$ angegeben, erwartet man 10 zufällige Treffer mit der gleichen Bewertung). Dieses Alignment ist also nicht signifikant. Die Treffer werden von BLAST nur ausgegeben, wenn der E-Wert kleiner als eine vorgewählte Schranke ist.

Als Anhaltspunkt kann man folgende Einteilung verwenden:
$E \leq 0.0001$: genaue Übereinstimmung
$0.0001 < E \leq 0.02$: Sequenzen vermutlich homolog
$0.02 < E \leq 1$: Homologie ist nicht auszuschließen
$E > 1$: man muss damit rechnen, dass dieser „Treffer“ Zufall ist.

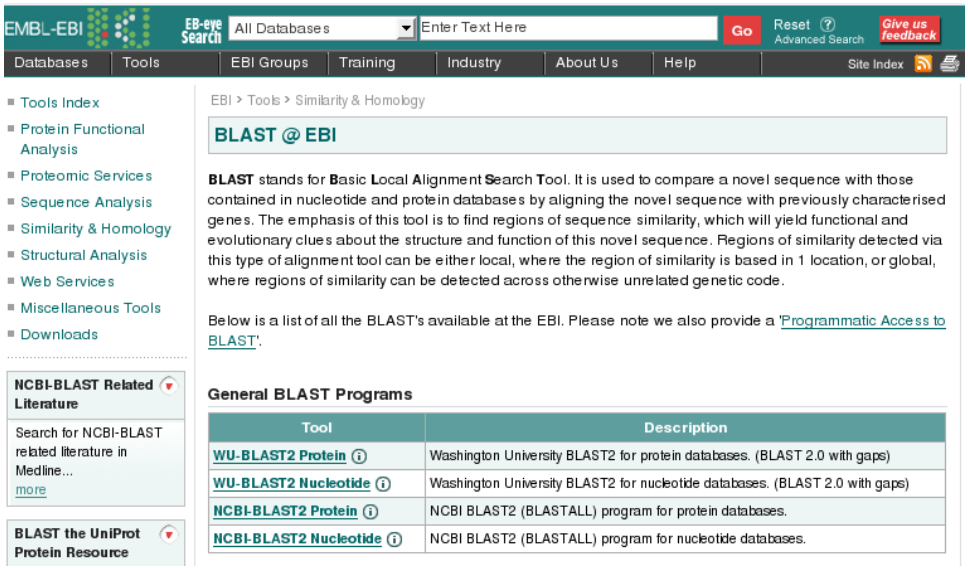
Durch das Wählen des Cut-offs bei den HSPs fallen der eine oder andere Treffer aus dem Ergebnis. Trotzdem liefert BLAST sehr gute Ergebnisse und ist somit eines der meistgenutzten Programme der Bioinformatik. Es gibt mehrere BLAST- Varianten, die man je nach Anfragestellung für die Suche auswählen kann.

Für die Standard-Proteinsuche wird neben BLASTP auch PSI-BLAST verwendet.
PSI-BLAST steht für Position-Specific Iterated BLAST.

PSI-BLAST basiert auf einer Motiv- bzw. Profilsuche, da so entfernte Verwandtschaften besser entdeckt werden als durch paarweise Vergleiche. PSI-BLAST führt zunächst eine BLAST-Suche mit Gaps durch.
Das PSI-BLAST Programm verwendet die Information jedes signifikanten Alignments um eine positionsspezifische Substitionsmatrix zu konstruieren, die an Stelle der Eingabesequenz in der nächsten Runde der Datenbank-Suche verwendet wird. PSI-BLAST kann solange iterativ verwendet werden bis keine neuen signifikanten Alignments mehr gefunden werden.

Basic BLAST	
Choose a BLAST program to run.	
nucleotid blast	Search a nucleotide database using a nucleotide query Algorithms: **blastn, megablast, discontiguous megablast**
protein blast	Search protein database using a protein query Algorithms: **blastp, psi-blast, phi-blast**
blastx	Search protein database using a translated nucleotide query
tblastn	Search translated nucleotide database using a protein query
tblastx	Search translated nucleotide database using a translated nucleotide query
Specialized BLAST	
Choose a type of specialized search (or database name in parentheses.)	
Make specific primers with **PRIMER-BLAST**	
Search **trace archives**	
Find conserved domains in your sequence (**cds**)	
Find sequences with similar conserved domain architecture (**cdart**)	
Search sequences that have gene expression profiles (**GEO**)	
Search immunoglobins (**IgBLAST**)	
Search for SNPs **(snp)**	
Screen sequence for vector contamination (**vecscreen**)	
Align two sequences using BLAST (**bl2seq**)	
Search protein or nucleotide targets in PubChem **BioAssay**	

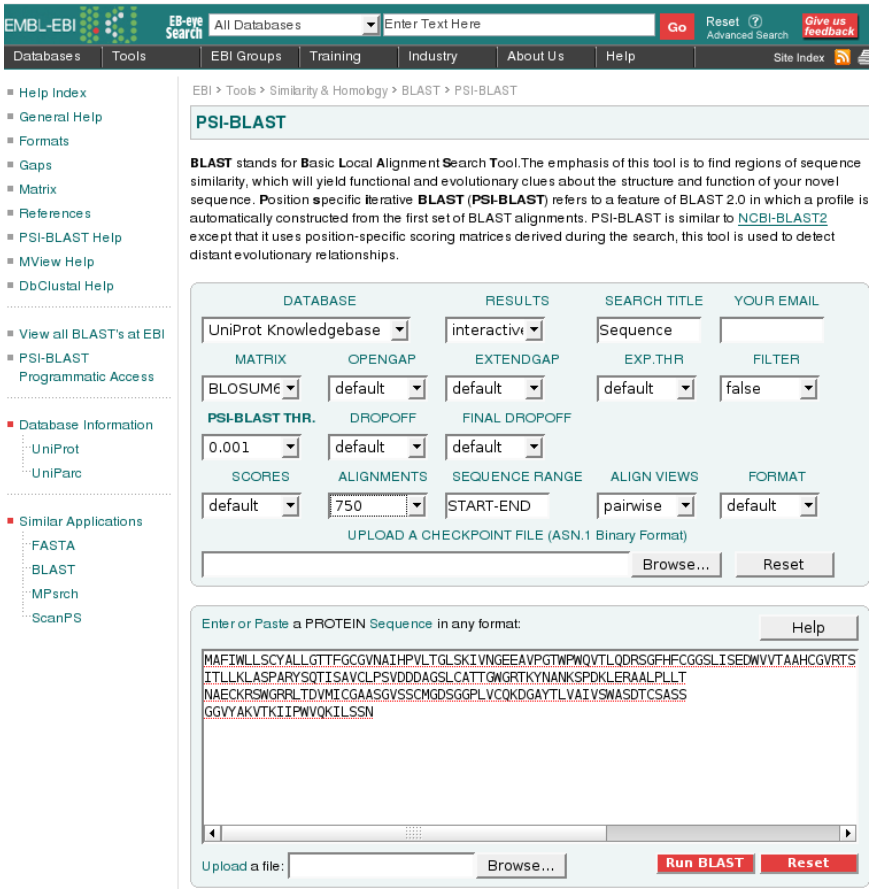
Um nun mit BLAST zu arbeiten, wählt man am einfachsten einen Webserver, der BLAST anbietet, z.B. über NCBI oder EBI.



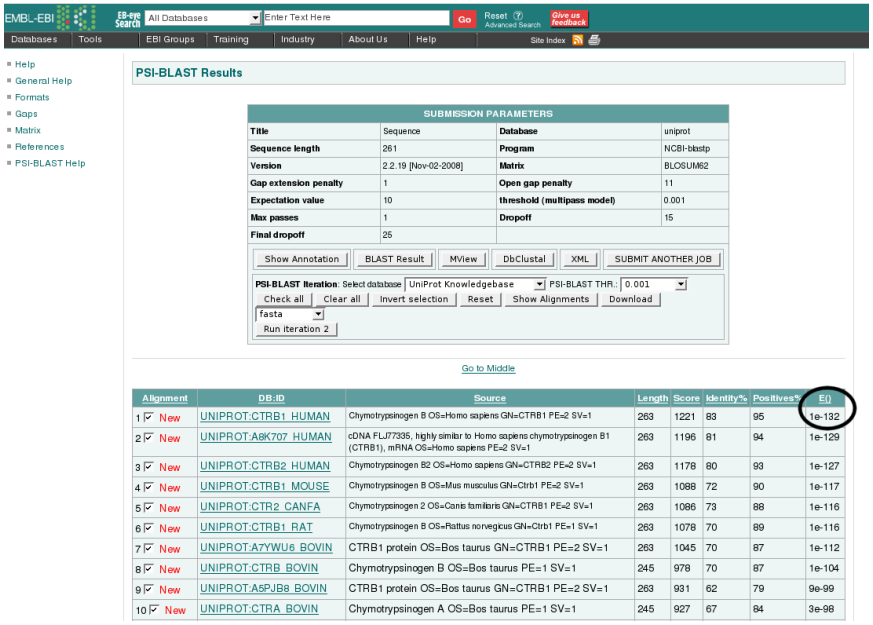
Als Eingabe wählt man die zu verwendende BLAST-Variante und die Sequenz, zu der ähnliche Sequenzen gefunden werden sollen. Diese kann entweder als Datei gespeichert sein und hochgeladen werden oder die Sequenz wird kopiert und in das dafür vorgesehene Feld eingefügt.

Testsequenz =
MAFIWLLSCYALLGTTFGCGVNAIHPVLTGLSKIVNGEEAVPGTWPWQVTL
QDRSGFHFCGGSLISEDWVVTAAHCGVRTSEILIAGEFDQGSDEDNIQVLRIA
KVFKQPKYSILTVNNDITLLKLASPARYSQTISAVCLPSVDDDAGSLCATTGW
GRTKYNANKSPDKLERAALPLLTNAECKRSWGRRLTDVMICGAASGVSSCM
GDSGGPLVCQKDGAYTLVAIVSWASDTCSASS GGVYAKVTKIIPWVQKILSSN

Dann wählt man die Parameter und die Datenbank, die durchsucht werden soll. Die Standardparameter sind im Bild zu erkennen:
Datenbank: UniProtKB
Matrix: BLOSUM62



> Starte BLAST



Die Ausgabe von PSI-BLAST liefert die Informationen über die Fundorte ähnlicher Sequenzen. Der sehr niedrige E-Wert e^{-132} für den ersten Treffer zeigt, dass der Wert nicht

zufällig gefunden wurde und ein sehr hoher Verwandschaftsgrad besteht.

Über die DB:ID Einträge gelangt man zu weiteren Informationen. (z.B. die Accessionnumber, unter der die Sequenz und weitere Informationen in UniProt gespeichert sind). M¨chte man das genaue Alignment ansehen, das durch BLAST generiert wurde, muss man im oberen Teil der Ergebnis-Seite den Button „Show Alignments“ drücken.

```
>UNIPROT:CTRB1_HUMAN P17538 Chymotrypsinogen B OS=Homo sapiens GN=CTRB1 PE=2 SV=1
          Length = 263

 Score =  474 bits (1221), Expect = e-132
 Identities = 220/263 (83%), Positives = 252/263 (95%), Gaps = 2/263 (0%)

Query: 1   MAFIWLLSCYALLGTTFGCGVNAIHPVLTGLSKIVNGEEAVPGTWPWQVTLQDRSGFHFC 60
           MAF+WLLSC+ALLGTTFGCGV AIHPVL+GLS+IVNGE+AVPG+WPWQV+LQD++GFHFC
Sbjct: 1   MAFLWLLSCWALLGTTFGCGVPAIHPVLSGLSRIVNGEDAVPGSWPWQVSLQDKTGFHFC 60

Query: 61  GGSLISEDWVVTAAHCGVRTSEILIAGEFDQGSDEDNIQVLRIAKVFKQPKYSILTVNND 120
           GGSLISEDWVVTAAHCGVRTS++++AGEFDQGSDE+NIQVL+IAKVFK  PK+SILTVNND
Sbjct: 61  GGSLISEDWVVTAAHCGVRTSDVVVAGEFDQGSDEENIQVLKIAKVFKNPKFSILTVNND 120

Query: 121 ITLLKLASPARYSQTISAVCLPSVDDD--AGSLCATTGWGRTKYNANKSPDKLERAALPL 178
           ITLLKLA+PAR+SQT+SAVCLPS DDD  AG+LCATTGWG+TKYNANK+PDKL++AALPL
Sbjct: 121 ITLLKLATPARFSQTVSAVCLPSADDDFPAGTLCATTGWGKTKYNANKTPDKLQQAALPL 180

Query: 179 LTNAECKRSWGRRLTDVMICGAASGVSSCMGDSGGPLVCQKDGAYTLVAIVSWASDTCSA 238
           L+NAECK+SWGRR+TDVMIC  ASGVSSCMGDSGGPLVCQKDGA+TLV IVSW SDTCS
Sbjct: 181 LSNAECKKSWGRRITDVMICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSDTCST 240

Query: 239 SSGGVYAKVTKIIPWVQKILSSN 261
           SS GVYA+VTK+IPWVQKIL++N
Sbjct: 241 SSPGVYARVTKLIPWVQKILAAN 263
```

Tips zur Verwendung von BLAST:

1. Verwende nicht stur die Standardparameter „You get what you look for “.
2. Führe Kontrollen durch: z.B. Schüttle die Sequenz durcheinander und wiederhole die Suche. Falls die variierte Sequenz ähnliche Ergebnisse liefert, beruht das Alignment auf einer systematischen Verfälschung, oder die Parameter sind nicht empfindlich genug gewählt.
3. Setze Komplexitätsfilter ein, wenn erforderlich.
4. Maskiere Repeats in genomischer DNA.
5. Teile große Genomsequenzen in Stücke auf um die Suche zu beschleunigen.
6. Sei skeptisch gegenüber hypothetischen Proteinen.
7. Erwarte Verunreinigungen in EST Datenbanken. In der Theorie sind ESTs Sequenzierungsreads von cDNA; cDNA wird von mRNA erhalten und die mRNAs stammen direkt von den Genen. Allerdings entsprechen ESTs oft keinen Genen, sondern gehören zu Exons bzw. UTRs, dem Überlappteil eines Repeats.

2.3 Zusammenfassung

- Paarweises Sequenzalignment ist heute Routine, aber nicht trivial.
- Mit dynamischer Programmierung (z.B. Smith-Waterman) findet man garantiert das Alignment mit optimaler Bewertung.
- Vorsicht: die Bewertungsfunktion ist nur ein Modell der biologischen Evolution.
- Die schnellste Alignmentmethode ist BLAST und seine Derivate. Es ergibt sehr robuste und brauchbare Ergebnisse für Proteinsequenzen.

3 Phylogenie

Diese Vorlesung hält sich an das Buch *„Inferring Phylogenies“* von Joseph Felsenstein, 2004. Die Abbildungen dieser Vorlesung wurden aus diesem Buch entnommen.

Zur Wiederholung sei noch einmal definiert:

> Homologie ist die Ähnlichkeit von Sequenzen, die durch Abstammung von einem gemeinsamen Ursprungsgen herrührt. Die Identifizierung und Analyse von Homologien ist eine zentrale Aufgabe der Phylogenie.
>
> Ein Alignment ist eine Hypothese für die positionelle Homologie zwischen Basenpaaren bzw. Aminosäuren.

Eine wichtige Frage bei der Erstellung von Alignments ist die Frage nach der Korrektheit: Ist es möglich ein korrektes Alignment zu erzeugen?
Grundsätzlich ist es recht schwierig, die Korrektheit eines Alignments strikt zu beweisen. Zunächst sollte man darauf achten, dass die Regionen, in denen die Gaps auftreten, zu den Loop-Regionen der Sekundärstruktur gehören. In den Sekundärstruktur bildenden Elementen wie α-Helix und β-Faltblatt sollten keine Gaps auftreten. Insbesondere bei Kenntnis der dreidimensionalen Strukturen für eine der zu alignierenden Proteinsequenzen ist dies ein gutes Kriterium. Alternativ kann man Ergebnisse aus Sekundärstrukturvorhersagen verwenden.

3.1 Multiples Sequenzalignment

Multiples Sequenzalignment:

```
G C G G C C C A   T C A G G T A C T T G   G T G G
G C G G C C C A   T C A G G T A G T T G   G T G G
G C G T T C C A   T C A G C T G G T T G   G T G G   Einfaches Alignment
G C G T C C C A   T C A G C T A G T T G   G T G G
G C G G C G C A   T T A G C T A G T T G   G T G A
* * * * * * * *   * * * * * * * * * * *   * * * *

T T G A C A T G   C C G G G G - - - A A   A C C G
T T G A C A T G   C C G G T G - - G T A   A G C C   Schwieriges
T T G A C A T G   - C T A G G - - - A A   C G C G   Alignment, aufgrund
T T G A C A T G   - C T A G G G A A C A   C G C G   von Insertionen und
T T G A C A T C   - C T C T G - - - A A   C G C G   Deletionen
* * * * * * * *   ? ? ? ? ? ? ? ? ? ? *   * * * *
```

3.1.1 Automatisch

Hier gibt es vor allem folgende zwei wichtigen Methoden:
Die dynamische Programmierung und das Progressive Alignment.

Die dynamische Programmierung (Vorlesung 2: Needleman-Wunsch und Smith-Waterman) liefert auch für multiple Sequenzalignments garantiert das optimale Alignment!

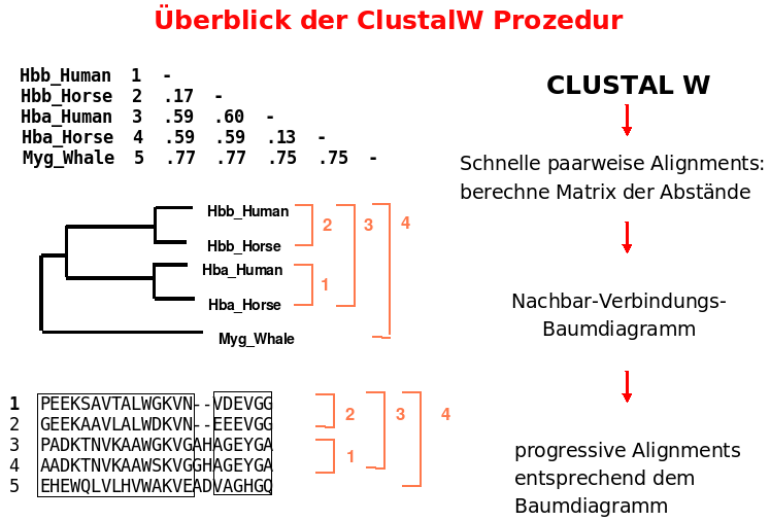
Betrachtet man jedoch zwei Proteinsequenzen von 100 Aminosäuren Länge, dauert es bereits 100^2 Sekunden, diese beiden Sequenzen erschöpfend zu alignieren. Bei drei Proteinsequenzen wird es 100^3 Sekunden dauern und 100^4 Sekunden für vier Sequenzen. Bei 20 Sequenzen ergibt das umgerechnet $1.90258 * 10^{34}$ Jahre.

Progressive Alignment:
Die Methode für ein Progressives Alignment wurde 1987 von Feng & Doolittle vorgestellt. Es handelt sich dabei um eine heuristische Methode, so dass nicht garantiert ist, das „optimale“ Alignment zu finden. Das Progressive Alignment benötigt (n-1) + (n-2) + (n-3) + ... + (n-n+1) paarweise Sequenzalignments als Ausgangspunkt. Progressive Alignments sind die am weitesten verbreitete Methode für multiple Sequenzalignments.
Eine sehr sensitive Methode sind ebenfalls Hidden Markov Modelle (HMMer) (s. Vorlesung 4.2.2)

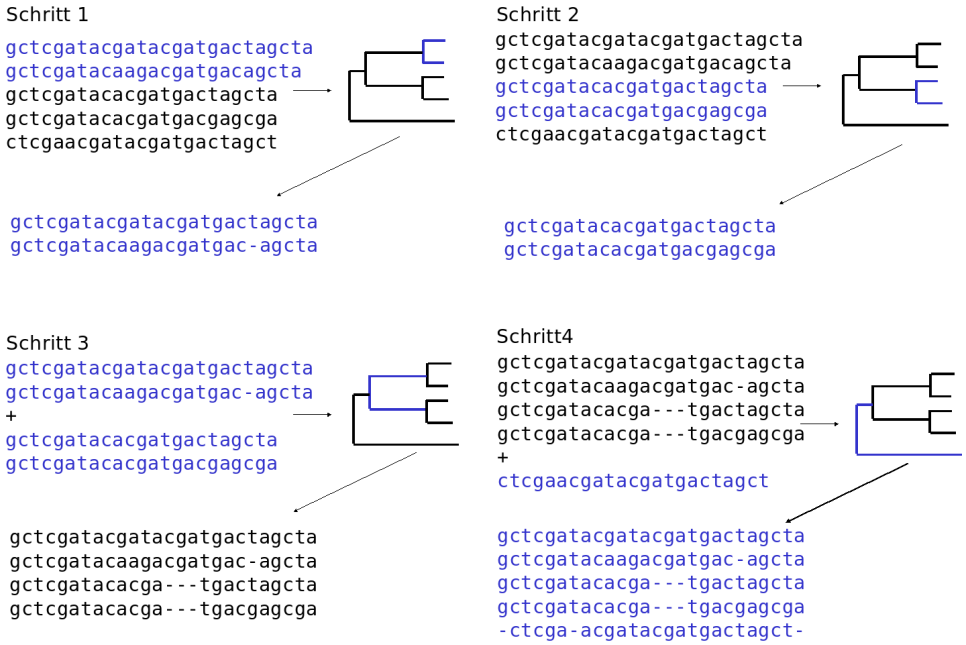
Ein Multiples Sequenzalignment ist nicht trivial. Manuelle Nacharbeit kann in Einzelfällen das Alignment verbessern. Ein Multiples Sequenzalignment erlaubt Denken in Proteinfamilien und -funktionen. Eine Implementierung der Progressive Alignment-Methode ist Clustal, welche ein weitverbreitetes Computerprogramm für die Berechnung von multiplen Alignments ist. ClustalW ist eine neuere Version, in der die Sequenzen gewichtet werden, um eine eventuelle, nicht gut balancierte Verteilung der evolutionären Distanzen im Datensatz auszugleichen. Dazu werden zum Einen Sequenzen heruntergewichtet, die sehr ähnlich zu anderen Sequenzen im Datensatz sind. Im Gegenzug werden Sequenzen, die sich von den Anderen am meisten unterscheiden, stärker gewichtet. Die dafür benutzten Gewichte werden direkt aus den Kantenlängen im ursprünglichen Suchbaum berechnet.

3.2 ClustalW

Zur Erstellung eines multiplen Alignments führt ClustalW drei Schritte aus.
Im ersten Schritt berechnet es alle möglichen paarweisen Sequenzalignments.
Im zweiten Schritt wird ein Phylogenetischer Baum erstellt, der die Verwandtschaftsbeziehungen zwischen den einzelnen Paaren veranschaulicht und in die richtige Reihenfolge bringt. Hierzu ist es notwendig, aus den paarweisen Sequenzalignments jeweils den Abstand zwischen den beiden Sequenzen zu berechnen. Diese Abstände werden dann in einer Distanzmatrix abgelegt. Mit Hilfe dieser Abstandsmatrix wird ein Nachbarschaftsbaum erstellt. (s. Neighbor-Joining Methode $->$ Phylogenie 3.3.2). Dieser Baum gibt die Reihenfolge an, in der das progressive Alignment ausgeführt werden wird.



Im Beispiel werden die Sequenzen der Hämoglobin Untereinheit Beta von Mensch und Pferd, sowie die Hämoglobin Untereinheit Alpha von Mensch und Pferd mit dem Myoglobin vom Wal verglichen.
Oben links ist die Distanzmatrix gezeigt, wobei gilt, dass de je kleiner die Distanz, desto näher verwandt sind die Sequenzen. Zu erkennen ist, dass die Hämoglobin Untereinheit Alpha des Menschen am ähnlichsten zu der Hämoglobin Alpha Untereinheit des Pferdes ist (d = 0.13). Als nächstes sind sich die Hämoglobin Beta Untereinheiten von Mensch und Pferd am ähnlichsten (d = 0.17).
Für die Erstellung des Baumes werden diese Paare zuerst unter sich verbunden und im nächsten Schritt miteinander. Das Myoglobin des Wales ist von allen anderen Sequenzen am weitesten entfernt und wird erst als letztes mit den beiden anderen Gruppen verbunden. Aufgrund dieses Baumes wird das progressive Alignment durchgeführt.
Beispiel für ein progressives Alignment anhand von 5 frei gewählten Nukleotidsequenzen:



1: Aligniere die beiden ähnlichsten Sequenzen zuerst. Dieses Alignment ist dann „fest“und wird nicht mehr angetastet. Falls später ein Gap eingeführt werden muss, wird er in beiden Sequenzen an der gleichen Stelle eingeführt. Deren relatives Alignment bleibt unverändert.
Hier wird ein Gap in der zweiten Sequenz an Position 6 von rechts eingefügt.

2: Ziehe den Baum heran um festzulegen, welches Alignment als nächstes durchgeführt werden soll. Es gibt die zwei Möglichkeiten eine dritte Sequenz zu den ersten beiden zualignieren oder ein zweites Sequenzpaar miteinander zu alignieren.
Hier werden Sequenz 3 und 4 aligniert, das Einfügen eines Gaps ist nicht erforderlich.

3: Die beiden Sequenzpaare werden miteinander aligniert. Die relativen Positionen der Paare bleiben dabei jeweils fest.
Hier werden im zweiten Sequenzpaar in beiden Sequenzen an gleicher Stelle drei Gaps hinzugefügt.

4: Im letzten Schritt wird die von den beiden Paaren am weitesten entfernte Sequenz zu dem 4er Alignment hinzugefügt. In diesem Fall bleiben die vier Sequenzen unverändert.

Hinweis: Es macht wenig Sinn, proteinkodierende DNA-Abschnitte auf DNA-Ebene zu alignieren!

```
ATGCTGTTAGGG          ATGCT-GTTAGGG
ATGCTCGTAGGG    ⟶     ATGCTCGT-AGGG
```

Da die Abfolge von Nukleotid-Triplets dabei unterbrochen werden kann, kann das Ergebnis sehr unplausibel sein und entspricht eventuell nicht dem biologischen Prozess.
Es ist viel sinnvoller, die Sequenzen in die entsprechenden Proteinsequenzen zu übersetzen, diese zu alignieren und dann in den DNS-Sequenzen an den Stellen Gaps einzufügen, an denen sie im Aminosäure-Alignment zu finden sind.

Der Hauptvorteil des Progressiven Alignments ist die Geschwindigkeit.
Nachteile sind, dass es keine objektive Funktion ist, d.h. man hat keine Möglichkeit zu quantifizieren ob das Alignment gut oder schlecht ist, wie zum Beispiel bei BLAST (E-Wert). Ebenso gibt es keine Möglichkeit zu überprüfen, ob das Alignment „korrekt“ ist.

Mögliche Probleme: Die Prozedur kann in ein lokales Minimum geraten, d.h. falls zu einem frühen Zeitpunkt ein Fehler im Alignment eingebaut wird, kann dieser später nicht mehr korrigiert werden, da die bereits alignierten Sequenzen fest bleiben. Ein zufälliges Alignment kann entstehen.

Der Frage ob alle Sequenzen gleich behandelt werden sollen, obwohl manche Sequenzen eng verwandt und andere entfernt verwandt sind und obwohl sie unterschiedliche Funktionen und Positionen in der dreidimensionalen Strukturen haben können, tritt ClustalW mit seinen Besonderheiten entgegen.

3.2.1 ClustalW Besonderheiten

- Sequenzgewichtung
- Variable Substitutionsmatrizen
- Residuen-spezifische Gap-Penalties und verringerte Penalties in hydrophilen Regionen (externe Regionen von Proteinsequenzen), bevorzugt Gaps in Loops anstatt im Proteinkern
- Positionen in frühen Alignments, an denen Gaps geöffnet wurden, erhalten lokal reduzierte Gap Penalties, um in späteren Alignments Gaps an den gleichen Stellen zu bevorzugen

Positionsspezifische Gap Penalties:
Zwei Parameter sind festzulegen (es gibt Default-Werte, aber man sollte sich bewusst sein, dass diese abgeändert werden können):

- Die GOP- Gap Opening Penalty ist aufzubringen, um eine Lücke in einem Alignment zu erzeugen.
- Die GEP- Gap Extension Penalty ist aufzubringen, um diese Lücke um eine Position zu verlängern.

Bevor irgendein Sequenzpaar aligniert wird, wird für jede Position der beiden Sequenzen eine Tabelle von GOPs erstellt. Die GOPs werden positions-spezifisch behandelt und können über die Sequenzlänge variieren. Falls ein Gap an einer Position existiert, werden die GOP- und GEP- Penalties herabgesetzt und alle anderen Regeln treffen nicht zu. Daher wird die Bildung von Gaps an Positionen wahrscheinlicher, an denen bereits Gaps existieren. Solange kein Gap offen ist, wird GOP hochgesetzt falls die Position innerhalb von 8 Residuen von einem bestehenden Gap liegt. Dadurch werden Gaps vermieden, die zu eng beieinander liegen. An jeder Position innerhalb einer Reihe von hydrophilen Residuen wird GOP herabgesetzt, da diese gewöhnlich in Loop-Regionen von Proteinstrukturen liegen. Eine Reihe von 5 hydrophilen Residuen aus den Aminosäuren Asp, Lys, Pro, Glu, Asn, Arg, Gly, Gln und Ser gilt als *hydrophiler stretch*. Dies kann durch den Benutzer geändert werden.

Tips:
Progressives Alignment ist ein mathematischer Vorgang, der völlig unabhängig von der biologischen Realität abläuft. Dadurch kann es sowohl eine sehr gute Abschätzung als auch eine unglaublich schlechte Abschätzung sein. Um dies zu erkennen, erfordert die Methode den Input und die Erfahrung des Benutzers. Die Methode sollte mit Vorsicht verwendet werden. Für gewöhnlich kann das Alignment manuell verbessert werden.
Es hilft oft, farbliche Darstellungen zu wählen. Je nach Einsatzgebiet sollte der Benutzer in der Lage sein, die zuverlässigen Regionen des Alignments zu beurteilen. Für phylogenetische Rekonstruktionen sollte man nur die Positionen verwenden, für die eine zweifelsfreie Hypothese über positionelle Homologie vorliegt.

3.3 Phylogenie

Eine phylogenetische Analyse einer Familie verwandter Nukleinsäure- oder Proteinsequenzen bestimmt, wie sich diese Familie durch Evolution entwickelt haben könnte.
Die evolutionären Beziehungen der Sequenzen können durch Darstellung als Blätter auf einem Baum veranschaulicht werden. Phylogenien, oder evolutionäre Bäume, sind die Grundlage, um Unterschiede zwischen Arten zu beschreiben und statistisch zu analysieren. Es gibt sie seit über 140 Jahren und seit etwa 40 Jahren mit Hilfe von statistischen, algorithmischen und numerischen Verfahren. Drei häufig verwendete Ansätze sind:

- Phylogenie über maximale Parsimonie (Sparsamkeit)
- Phylogenie-Berechnung mit Hilfe einer Distanzmatrix
- Phylogenie über maximum likelihood (wird hier nicht behandelt)

Zwei häufig verwendete Programme sind:
PHYLIP (phylogenetic inference package, von J. Felsenstein)
http://evolution.genetics.washington.edu/phylip.html
und PAUP (phylogenetic analysis using parsimony, von Sinauer Associates
http://paup.csit.fsu.edu/

3.3.1 Maximale Parsimonie

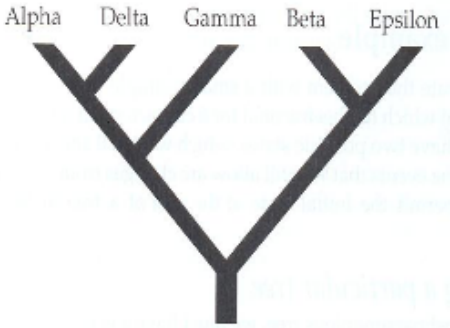
Nach Edwards & Cavalli-Sforza (1963) ist derjenige evolutionäre Baum zu bevorzugen, der „den minimalen Anteil an Evolution“ enthält.

1. Für jede vorgeschlagene Phylogenie müssen wir in der Lage sein, die Vorgänge zu rekonstruieren, die am wenigsten Zustandsänderungen benötigen.

2. Wir müssen unter allen möglichen Phylogenien nach denen suchen können, die eine minimale Anzahl an Zustandsänderungen beinhalten.

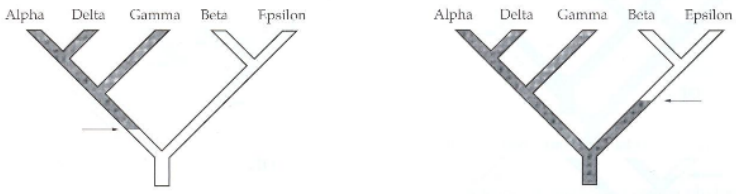
Gegeben seien sechs Buchstaben lange Sequenzen aus fünf Spezies Alpha bis Epsilon, die die Werte 0 oder 1 annehmen können. Dieser Baum stelle die Phylogenie des ersten Buchstabens dar.

	Characters					
Species	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0



Erlaubt seien Austausche $0 - 1$ und $1 - 0$. Der anfängliche Zustand an der Wurzel des Baums kann 0 oder 1 sein. Um den Baum höchster Parsimonie zu finden müssen wir berechnen können, wie viele Zustandsänderungen für einen gegebenen Baum nötig sind. Es gibt in diesem Fall zwei gleich gute Rekonstruktionen, die jede nur eine Buchstabenänderung benötigen. Sie nehmen unterschiedliche Zustände an der Wurzel des Baums an und unterschiedliche Positionen für die eine Änderung.

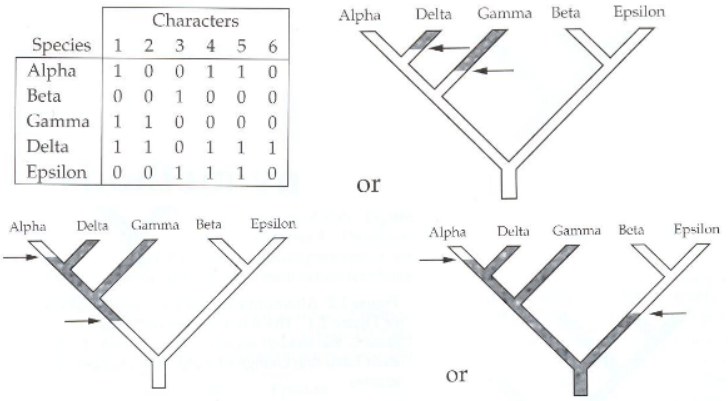
Mögliche Bäume für Position 1:
In beiden tritt jeweils eine Änderung auf. Links ist der Wurzelwert 0 (Zustand 0 = weiß) und es tritt eine Änderung zu 1 für die drei Spezies Alpha, Delta und Gamma auf (Zustand 1 = grau). Rechts analog.



or

Figure 1.2: Alternative reconstructions of character 1 on the phylogeny of Figure 1.1. The white region of the tree is reconstructed as having state 0, the shaded region as having state 1. The two reconstructions each have one change of state. The changes of state are indicated by arrows.

Für die zweite Position gibt es drei mögliche Bäume, die jeweils zwei Änderungen benötigen, z.B. ist die Wurzel = 0 ändert sich der Zustand bei Delta und Gamma entweder erst im direkten Zweig oder zuvor, dann muss der Zustand auf dem Weg zu Alpha wieder geändert werden.



Species	Characters 1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Figure 1.3: Reconstructions of character 2 on the phylogeny of Figure 1.1. The white regions have state 0, the shaded region state 1. The changes of state are indicated by arrows.

An Position drei gibt es wieder eine Zustandsänderung, an den Positionen vier und fünf jeweils zwei und an Position sechs wieder eine Zustandsänderung. Die gesamte Anzahl an Zustandsänderungen auf diesem Baum ist $1 + 2 + 1 + 2 + 2 + 1 = 9$

Rekonstruktion der Zustandsänderungen auf diesem Baum

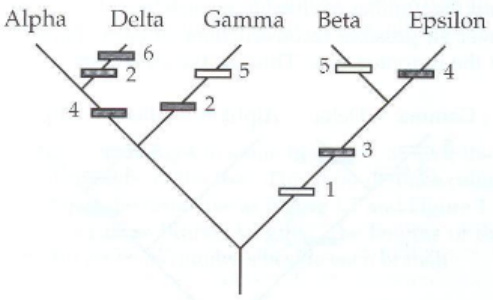


Figure 1.7: Reconstruction of all character changes on the phylogeny of Figure 1.1. The changes are shown as bars across the branches, with a number next to each indicating which character is changing. The shading of each box indicates which state is derived from that change.

Die naheliegende Methode, den Baum höchster Parsimonie zu finden ist, ALLE möglichen Bäume zu betrachten und einzeln zu bewerten. Leider ist die Anzahl an möglichen Bäumen üblicherweise zu groß. Dafür verwendet man heuristische Suchmethoden, die versuchen, die besten Bäume zu finden ohne alle möglichen Bäume zu betrachten:

1. Konstruiere eine erste Abschätzung des Baums und verfeinere diesen durch kleine Änderungen = finde „benachbarte “Bäume.

2. Wenn irgendwelche dieser Nachbarn besser sind, verwende diese und setze die Suche fort.

In dem Beispiel ist die geringste Anzahl an möglichen Zustandsänderungen 8. Aber auch hier gibt es mehrere Möglichkeiten den Baum darzustellen, wenn er bei einer Wurzel beginnt. Entfernt man die Wurzel, sehen die Bäume gleich aus.

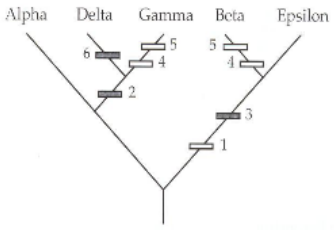
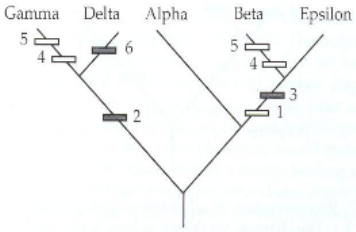


Figure 1.8: Reconstruction of all changes on the most parsimonious phylogeny for the data of Table 1.1. It requires only 8 changes of state. The changes are shown as bars across the branches, with a number next to each indicating which character is changing. The shading of each box indicates which state is derived from that change.



e 1.9: Another rooted tree with the same number of changes of state.

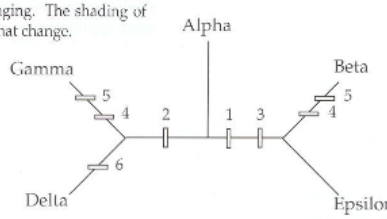


Figure 1.10: The unrooted tree corresponding to Figures 1.8 and 1.9.

Die Parsimonie hängt also nur von den Bäumen ohne Wurzel (*unrooted tree*) ab.

Biologen verwenden jedoch gerne Bäume mit Wurzeln (*rooted tree*).
Für das Zählen der evolutionären Zustandsänderungen existieren zwei verwandte Algorithmen von Fitch (1971) und Sankoff (1975), die beide dynamische Programmierung verwenden:

Beide Algorithmen:

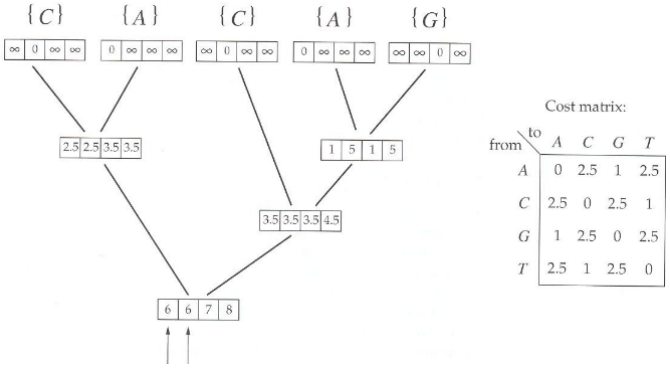
- bewerten eine Phylogenie Buchstabe für Buchstabe.
- betrachten jeden Buchstaben als Baum mit Wurzel an einem geeigneten Platz.
- propagieren eine Information nach unten durch den Baum;
  beim Erreichen der Blätter ist die Anzahl der Zustandsänderungen bekannt.

Dabei werden die Zustandsänderungen oder internen Zustände an den Knoten des Baums nicht konstruiert.

Hier wird nur den Sankoff Algorithmus erklärt. Dieser ist komplexer als Fitch, dafür aber verständlich in Hinsicht auf seine Funktionalität.

Sankoff

Gegeben ist eine Kostenmatrix c_{ij} für alle Zustandsänderungen zwischen den Zuständen i und j. Die kompletten Kosten der sparsamsten Zustandsveränderungen (maximale Parsimonie) werden durch die Konstruktion aller Zustände berechnet.
Man kann so die Werte für jeden Knoten berechnen, und so auch für den Wurzelknoten. Dazu benötigt man das Minimum aus den beiden Vorgängerknoten. Dieses Vorgehen beschreibt die minimalen evolutionären Zustandsänderungen für den jeweiligen Zustand.
In den Anfangsknoten (den Blättern) des Baums ist $S(i)$ einfach zu berechnen. Die Kosten sind 0 wenn der beobachtete Zustand der Zustand i ist, ansonsten sind die Kosten unendlich.
Nun ist ein Algorithmus nötig, der die minimalen Kosten $S(i)$ für die mittleren Knoten berechnet, die die „Vorfahren“ von jeweils zwei Vorgängerknoten bilden.



Definiere die beiden Vorgängerknoten als „l“ (linker Knoten) und „r“ (rechter Knoten). Berechne für den inneren (*ancestor*) Knoten $S_a(i)$:

$$S_a(i) = \min_j[c_{ij} + S_l(j)] + \min_k[c_{ik} + S_r(k)] \tag{3.3.1}$$

Im Beispiel ergibt das für den Vorfahr-Knoten für {C} und {A}:

1. $$\begin{aligned} S_{a1}(A) &= min[(AC) + S_l([A,C,G,T])] + min[(AA), +S_r([A,C,G,T])] \\ &= min[2.5 + (unend, 0, unend, unend)] + min[0 + [0, unend, unend, unend]] \\ &= 2.5 \end{aligned}$$

   Mit l: $(AC) = 2.5$, $S_l(A) = unend$, $S_l(C) = 0$,... und r: $(AA) = 0$, $S_r(A) = 0$, $S_r(C) = unend$,...

2. $S_{a1}(C) =$ ähnlich wie $S_{a1}(A)$, ersetze c_{ij} durch $(CC) = 0$ und c_{ik} durch $(CA) = 2.5$

3.
$$\begin{aligned} S_{a1}(G) &= min[(GC) + S_l([A,C,G,T])] + min[(GA), +S_r([A,C,G,T])] \\ &= min[2.5 + (unend, 0, unend, unend)] + min[1 + [0, unend, unend, unend]] \\ &= 3.5 \end{aligned}$$

Mit l: $(GC) = 2.5$, $S_l(A) = unend$, $S_l(C) = 0$,... und r:$(GA) = 1$, $S_r(A) = 0$, $S_r(C) = unend$, ...

4. $S_{a1}(T)$ = ähnlich zu $S_{a1}(G)$, ersetzte c_{ij} durch $(TC) = 1$ und c_{ik} durch $(TA) = 2.5$

Daraus ergibt sich für $S_{a1} = [2.5, 2.5, 3.5, 3.5]$;

Die übrigen ancestor Knoten werden nacheinander nach demselben Schema erstellt. Hier noch gezeigt für C − {AG}:

1.
$$\begin{aligned} S_{a3}(A) &= min[(AC) + S_l([A,C,G,T])] + min[(A[A,G]) + S_r([A,C,G,T])] \\ &= min[2.5 + (unend, 0, unend, unend)] + min[0 + [1,5,1,5]] \\ &= 3.5 \end{aligned}$$

!! Beachte, dass die Kosten für den rechten Ast entweder A−A = 0 oder A−G = 1 sein können. Auch hier nimmt man das Minimum !!

2.
$$\begin{aligned} S_{a3}(C) &= min[(CC) + S_l([A,C,G,T])] + min[(C[A,G] + S_r([A,C,G,T])] \\ &= min[0 + (unend, 0, unend, unend)] + min[2.5 + [1,5,1,5]] \\ &= 3.5 \end{aligned}$$

3.
$$\begin{aligned} S_{a3}(G) &= min[(GC) + S_l([A,C,G,T])] + min[(G[A,G]) + S_r([A,C,G,T])] \\ &= min[2.5 + [unend, 0, unend, unend]] + min[0 + [1,5,1,5]] \\ &= 3.5 \end{aligned}$$

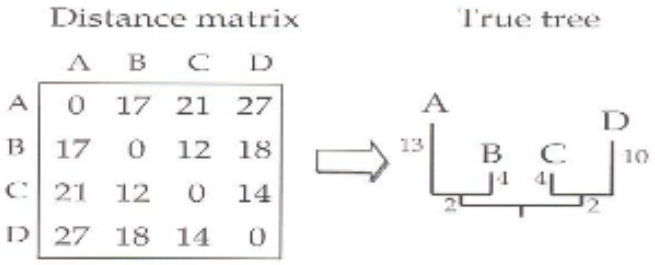
4.
$$\begin{aligned} S_{a3}(T) &= min[(TC) + S_l([A,C,G,T])] + min[(T[A,G]) + S_r([A,C,G,T])] \\ &= min[1 + (unend, 0, unend, unend)] + min[2.5 + [1,5,1,5]] \\ &= 4.5 \end{aligned}$$

3.3.2 Distanzmatrix

Eine weitere Methode, die Phylogenie zu berechnen, ist das Anwenden einer Distanzmatrix. Dabei werden die Distanzen zwischen allen Sequenzen untereinander berechnet und in einer Distanzmatrix notiert. Ein bekannter Algorithmus ist der Neighbor-Joining Algorithmus.

Neighbor-Joining Algorithmus

Der Neighbor-Joining Algorithmus von Saitou und Nei (1987) schätzt die Minimale Evolution ab. Dabei werden die Abstände zwischen allen Sequenzpaaren berechnet und in einer Distanzmatrix D_{ij} mit den Spezies i und j gespeichert (s. CLUSTALW).



Mit dieser Distanzmatrix wird eine neue Matrix M_{ij} berechnet, dafür werden die Abstände von einer Spezies zu allen anderen Spezies addiert:

1. $$r_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{(n-2)}$$

   Mit n = Anzahl der Spezies, $n-2$ ist hier die Anzahl der inneren Knoten.

2. $$M_{ij} = D_{ij} - r_i - r_j$$

   Wähle den kleinsten Eintrag aus M_{ij}. Die beiden Spezies i und j sind dann am nächsten zueinander. Berechne die Astlänge von i zu dem neuen Knoten v_i und von j zu dem neuen Knoten v_j.

3. $$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(r_i - r_j)$$
   $$v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(r_j - r_i)$$

   Als nächstes muss von dem neuen Knoten (ij), der jetzt wie eine neue Spezies behandelt wird, zu jedem anderen Knoten (k) die neue Distanz berechnet werden.

4. $$D_{(ij),k} = \frac{1}{2}(D_{ik} + D_{jk} - D_{ij})$$

5. Entferne die Einträge für die Spezies i und j aus der Matrix und füge die neuen Daten für (ij) $(=U)$ ein. Solange mehr als zwei Einträge in der Matrix übrig sind, wiederhole den Algorithmus ab Schritt 1. Sind nur noch zwei Knoten (Matrixeinträge) übrig (z.B. l und m), verbinde diese mit einem Ast der Länge D_{lm}.

Man erhält so Schritt für Schritt einen phylogenetischen Beziehungsbaum. Der Baum ist nun hierarchisch nach dem Grad der Verwandtschaft seiner Einträge aufgebaut. Wenn mehrere Einträge einen Unterast bilden, spricht man auch von einem Cluster. Die Daten werden also hierarchisch geclustert .

Ein Beispiel für den Neighbor-Joining Algorithmus:

I.
1) Gegeben ist eine Distanzmatrix D für 6 Spezies S1-S6:

S	S1	S2	S3	S4	S5	S6
S1	-	0.102	0.366	0.366	0.462	0.486
S2	0.102	-	0.384	0.366	0.462	0.492
S3	0.366	0.384	-	0.378	0.451	0.438
S4	0.366	0.366	0.378	-	0.451	0.446
S5	0.462	0.462	0.451	0.451	-	0.481
S6	0.486	0.492	0.438	0.446	0.481	-

Berechne anhand der Matrix die r_is:

$$r_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{(n-2)}$$

$r_1 = \frac{0.102+0.366+0.366+0.462+0.486}{6-2} = 0.4455$

$r_2 = 0.4515$
$r_3 = 0.42925$
$r_4 = 0.42675$
$r_5 = 0.57675$
$r_6 = 0.58575$

2) Im nächsten Schritt werden nun die neuen Matrixeinträge M_{ij} berechnet:
$M_{ij} = D_{ij} - r_i - r_j$
$M_{12} = 0.102 - 0.4455 - 0.4515 = -0.795$
usw...

S	S1	S2	S3	S4	S5	S6
S1	-					
S2	-0.795	-				
S3	-0.509	-0.497	-			
S4	-0.506	-0.512	-0.778	-		
S5	-0.560	-0,566	-0.555	-0.553	-	
S6	-0.548	-0.545	-0.577	-0.567	-0.682	-

3) Im nächsten Schritt wird das Paar der beiden Spezies, die sich am nächsten sind, ausgewählt. In diesem Beispiel sind dies S1 und S2 ($->$ K). Diese werden dann über einen Knoten K zusammengeführt. Die Abstände von S1 und S2 zu dem neuen Knoten K werden berechnet als:
$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(r_i - r_j)$

$v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(r_j - r_i)$

$v_1 = \frac{1}{2}0.102 + \frac{1}{2}(0.4455 - 0.4515) = 0.048$

$v_2 = \frac{1}{2}0.102 + \frac{1}{2}(0.4515 - 0.4455) = 0.054$

4) Lösche die Einträge für S1 und S2 und ersetze sie durch den neuen Knoten K. Berechne die Astlänge von dem neuen Knoten K zu allen anderen Knoten:
$D_{(ij),k} = \frac{1}{2}(D_{ik} + D_{jk} - D_{ij})$

$D_{(1,2),3} = \frac{1}{2}(0.366 + 0.384 - 0.102) = 0.324$
usw...

S	K	S3	S4	S5	S6
K	-				
S3	0.324	-			
S4	0.315	0.378	-		
S5	0.411	0.451	0.451	-	
S6	0.438	0.438	0.446	0.481	-

5) Wiederhole die Schritte 1) - 4) drei mal (da N-2 = 4 innere Knoten, K - K4) bis nur noch zwei Knoten übrig sind.

II. 1. Wiederholung:

1) Berechne anhand der neuen Matrix die r_is:

$$r_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{(n-2)}$$

$r_K = \frac{0.324+0.315+0.411+0.438}{5-2} = 0.496$ (neu: n=5)

$r_3 = 0.5303$
$r_4 = 0.53$
$r_5 = 0.598$
$r_6 = 0.601$

2) Im nächsten Schritt werden nun die neuen Matrixeinträge M_{ij} berechnet:
$M_{ij} = D_{ij} - r_i - r_j$
$M_{K,S3} = 0.324 - 0.496 - 0.5303 = -0.7023$
$M_{K,S4} = 0.315 - 0.496 - 0.530 = -0.711$
$M_{K,S5} = 0.411 - 0.496 - 0.598 = -0.683$
usw...

S	K	S3	S4	S5	S6
K	-	-0.7023	-0.711	-0.683	-0.659
S3	-0.7023	-	-0.6823	-0.6773	-0.6933
S4	-0.711	-0.6823	-	-0.677	-0.685
S5	-0.683	-0.6773	-0.677	-	-0.718
S6	-0.659	-0.6933	-0.685	-0.718	-

3) => Nachbarn S5 - S6
Diese werden dann über einen Knoten $K2$ zusammengeführt. Die Abstände von S5 und S6 zu dem neuen Knoten $K2$ werden berechnet als:
$v_5 = \frac{1}{2}0.481 + \frac{1}{2}(0.598 - 0.601) = 0.4795$

$v_6 = \frac{1}{2}0.481 + \frac{1}{2}(0.601 - 0.598) = 0.4825$

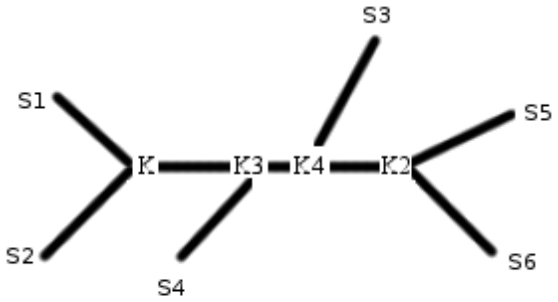
4) Lösche die Einträge für S5 und S6 und ersetze sie durch den neuen Knoten K2. Berechne die Astlänge von dem neuen Knoten K zu allen anderen Knoten:
$D_{(5,6),K} = \frac{1}{2}(0.451 + 0.438 - 0.481) = 0.204$

$D_{(5,6),3} = \frac{1}{2}(0.451 + 0.446 - 0.481) = 0.208$

$D_{(5,6),4} = \frac{1}{2}(0.411 + 0.438 - 0.481) = 0.184$

S	K	S3	S4	K2
K	-	0.324	0.315	0.184
S3	0.324	-	0.378	0.204
S4	0.315	0.378	-	0.208
K2	0.184	0.204	0.208	-

usw ...

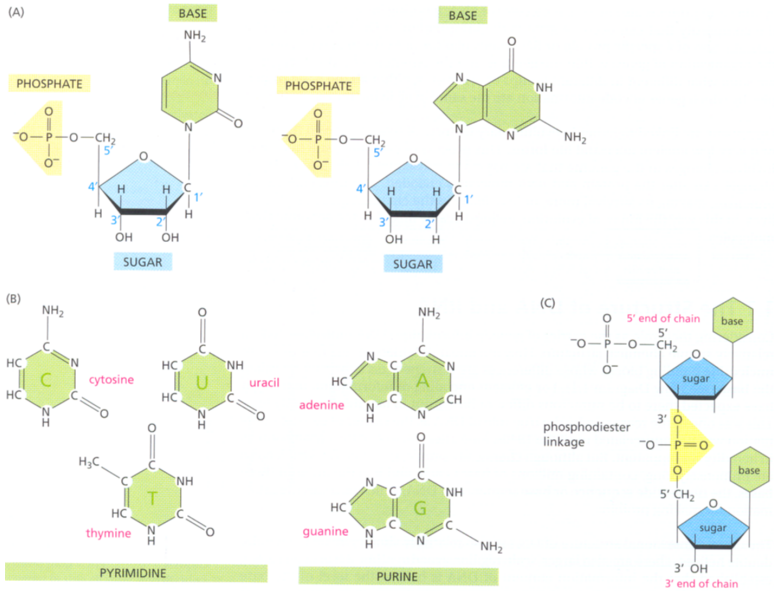


Auf diese Weise wird ein evolutionärer Baum konstruiert, dessen Äste präzise die Länge der verstrichenenen biologischen Zeit besitzen bzw. deren Länge dem Grad an beobachteter Sequenzvariationen entspricht. (In der Beispielzeichnung sind die Längen nicht berücksichtigt.)

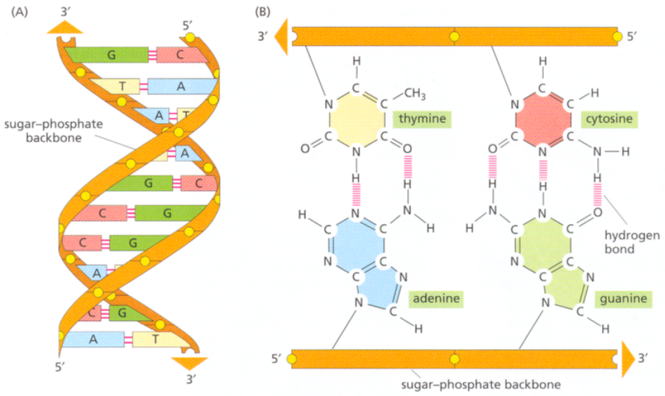
4 Analyse von genomischen Merkmalen

4.1 Genomaufbau

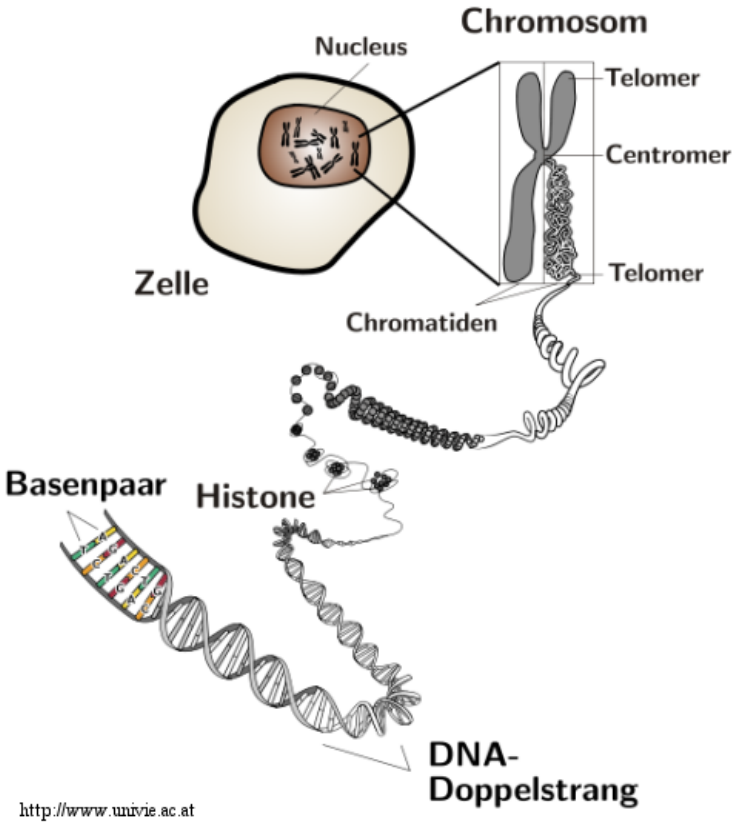
Wie in der ersten Vorlesung erwähnt, sind die Grundbausteine des Genoms die vier Desoxyribonukleotide, kurz Nukleotide. Jedes Nukleotid besteht aus einem Phosphatrest, einem Zucker (Desoxyribose) und aus einer der vier organischen Basen Adenin, Guanin, Cytosin oder Thymin.



Diese bilden ein langes Kettenmolekül (Polymer), die Nukleinsäure. Die das Rückgrat bildenden Desoxyribose- und Phosphat-Untereinheiten sind bei jedem Nukleotid gleich. Aufgrund der negativen Ladung der Phosphatreste sind diese hydrophil, sie geben der DNA insgesamt eine negative Ladung.
Lagern sich zwei der Nukleinsäure Polymere mit komplementärer Sequenz aneinander, entsteht die typische Form der DNA, die Doppelhelix. Dabei werden die beiden Stränge in entgegengesetzter Richtung aneinandergelegt, das 3'-Ende des einen Strangs liegt am selben Ende der DNA wie das 5'-Ende des anderen Strangs. Durch die Aneinanderlagerung werden in der Mitte der Doppelhelix immer ein Purin mit einem Pyrimidin kombiniert. Es paaren sich entweder Adenin mit Thymin, die dabei zwei Wasserstoffbrücken ausbilden, oder Cytosin mit Guanin, die über drei Wasserstoffbrücken miteinander verbunden sind. Da sich immer die gleichen Basen paaren, lässt sich aus der Sequenz der Basen in einem Strang die Sequenz des anderen Strangs ableiten, d.h. die Sequenzen sind komplementär.



In prokaryotischen Zellen liegt die doppelsträngige DNA als zirkuläres Molekül vor, wobei sich ein 3'-Ende mit seinem 5'-Ende zum Kreis schließt. Die DNA liegt dort statt im Zellkern in einem Plasmid vor.
In eukaryotischen Zellen ist die DNA in Chromosomen gepackt, die sich im Zellkern befinden. In der kleinsten kompakten Einheit ist die DNA in Nucleosomen aufgewickelt. Diese enthalten acht Histonproteine. Durch diverse Faltungen und Verknäuelungen bildet sich das Chromosom.



4.2 Identifikation von Genen

Die Gesamtlänge der menschlichen DNA umfasst etwa 3,2 Gbp (= Gigabasenpaare oder Milliarden Basenpaare) mit bisher 22300 gefundenen Genen. (*www.ensembl.org, Homo sapiens*. Datenbankstand von Februar 2009). Wir wenden uns nun der Frage zu, wie man mit Methoden der Bioinformatik Gene identifizieren kann. Die experimentelle Bestätigung folgt z.B. durch Analysen von EST- oder cDNA-Sequenzen.
Die einfachste Methode, DNA Sequenzen zu finden, die für Proteine kodieren, ist nach den offenen Leserahmen der Sequenz, den so genannten *open reading frames* (ORFs), zu suchen.

4.2.1 Offene Leserahmen (ORFs)

In jeder DNA-Sequenz gibt es sechs mögliche ORFs:
Drei ORFs starten an den Positionen 1, 2, und 3 und gehen in die 5' - 3' Richtung. Für Methionin (AUG) sind das: 1. A, 2. U. 3. G.
Drei ORFs starten an den Positionen 1, 2, und 3 und gehen in die 5' - 3' Richtung des komplementären Strangs.

In prokaryotischen Genomen werden Protein-kodierende DNA-Sequenzen gewöhnlich von der Polymerase in mRNA transkribiert und die mRNA wird ohne wesentliche Änderungen direkt in einen Aminosäurestrang übersetzt. Daher ist der längste ORF vom ersten verfügbaren Met-Codon (AUG) auf der mRNA bis zum nächsten Stopcodon im selben ORF gewöhnlich eine gute Vorhersage für die Protein-kodierende Region.

In eukaryotischen Genomen wird die mRNA vor der Translation erst noch modifiziert. Die Protein-kodierende DNA (Exons) ist nämlich durch nicht kodierende Sequenzstücke, die Introns, unterbrochen. Die mRNA wird in drei Schritten modifiziert. Zuerst erhält die RNA an ihr 5'-Ende eine „Kappe", ein 7'-Methylguanosin wird addiert, dann findet am 3'-Ende eine Polyadenylierung statt. Zuletzt müssen die Introns in einem Spleiß-Schritt (*splicing*) aus der kodierenden Sequenz herausgeschnitten und die Exons zusammengefügt werden. Dies geschieht in einem großen Proteinkomplex, dem Spliceosom.

Extrinsische/Intrinsische Methoden
Viele Verfahren kombinieren nun (a) Homologie-Methoden = „extrinsische Methoden", da Information „von außen" benutzt wird, mit (b) Genvorhersage-Methoden = „intrinsische Methoden", in denen nur die Information der einzelnen Sequenz verwendet wird.

Extrinsische Methoden identifizieren die Ähnlichkeiten mittels eines optimalen lokalen Alignments (− > Smith-Waterman) oder anhand von heuristischen Suchen in Protein/DNA Datenbanken: FASTA, BLAST,...
Eine Schwäche dabei ist die unvollständige Informationsdichte in den Datenbanken, da nichts gefunden werden kann, wenn die Datenbank keine ähnliche Sequenzen enthält. Im Prinzip könnte es ja durchaus Gene geben, die nur in einem einzelnen Organismus vor-

kommen und daher durch Homologiesuchen nicht gefunden werden können. Zum anderen ist die Zahl der bisher sequenzierten Genome verglichen mit der Anzahl der Organismen auf der Erde natürlich noch immer sehr gering.
Kleine Exons werden leicht übersehen. ESTs (expressed sequence tags) und cDNA (DNA-Kopie der mRNA) erlauben es Exons zu identifizieren. Der Vorteil der extrinsischen Methoden ist jedoch, dass ein einziger Treffer genügt um ein Gen aufzuspüren.

In **intrinsischen Methoden** muss erst festgelegt werden, woran man eine kodierende Region erkennen soll:

Table 4. Comparison of the GeneMarkS, Glimmer 2.02 and ORPHEUS gene prediction programs on the following test sets: the *B.subtilis* genome as annotated in GenBank (A); three sets of *B.subtilis* genes shorter than 300 nt with at least one (B), at least two (C) and at least 10 (D) significant homologies determined by BLAST analysis; and a set of 195 experimentally validated *E.coli* genes (E)

Program	Test set	Genes in test set	Genes precisely predicted[a]	Genes detected[b] (3′ end)
Glimmer	A	4099	2556 (62.4%)	**4023** (98.1%)
ORPHEUS	A		3028 (73.9%)	3484 (85.0%)
GeneMarkS	A		**3412** (83.2%)	3962 (96.7%)
Glimmer	B	123	70 (57.0%)	112 (91.1%)
GeneMarkS	B		**102** (82.9%)	**113** (91.9%)
Glimmer	C	72	41 (57.0%)	66 (91.7%)
GeneMarkS	C		**64** (88.9%)	**68** (94.4%)
Glimmer	D	51	26 (51.0%)	45 (88.2%)
GeneMarkS	D		**46** (90.2%)	**48** (94,1%)
Glimmer	E	195	139 (71.3%)	**195** (100%)
ORPHEUS	E		148 (75.9%)	181 (92.8%)
GeneMarkS	E		**184** (94.4%)	**195** (100%)

Numbers in bold indicate the highest number of genes detected or genes precisely predicted for each test set.
[a]Refers to the case where both the 5′ end and the 3′ end predictions match the annotation.
[b]Refers to the case where the 3′ end prediction (and not necessarily 5′ end prediction) matches the annotation.

Besemer et al. Nucl. Acids. Res. 29, 2607 (2003)

Häufig verwendete Merkmale sind zum Beispiel:

- Anordnung der Nukleotide (z.B. ATG - Methionin und TAG als Stop Codon),
- GC Inseln, $->$ in Introns sind A und T häufiger,
- Anordnung der Codons,
- Hexamer Frequenz,
- der Zyklus in dem die Base beobachtet wird

Es wurde herausgefunden, dass die Hexamer Frequenz, oder generell, die k-mer, den statistisch größten Unterschied zwischen Exons und Introns zeigt.
Etwa die Hälfte aller Gene kann durch Homologie zu anderen bekannten Genen oder Proteinen gefunden werden. Dieser Anteil wächst stetig, da die Anzahl an sequenzierten Genomen und bekannten cDNA/EST Sequenzen kontinuierlich wächst. Um die übrige Hälfte an Genen zu finden, muss man intrinsische Methoden zur Vorhersage einsetzen.

4.2.2 Hidden Markov Modell

Markov Modell:
Ein Markov Modell ist ein stochastisches Modell, das annimmt, dass die Wahrscheinlichkeit des Vorkommens eines Nukleotids (A, T, G, oder C) an einer bestimmten Position der Sequenz nur von den k vorhergehenden Nukleotiden abhängt. k wird als die Ordnung des Markovmodells bezeichnet.
Das Modell wird definiert über $P(X|k)$ = Wahrscheinlichkeit von X unter der Bedingung k.
X ist hier A, T, G oder C.
Ein solches Modell muss mit Hilfe von Testsequenzen trainiert werden, um die Wahrscheinlichkeiten abschätzen zu können. Die einfachsten Markov Modelle sind homogene Null oder Markov Modelle, die annehmen, dass ein Nukleotid unabhängig von der Frequenz auftritt. Meistens werden diese für die nicht kodierenden Regionen, also Introns, benutzt. Neuere Programme, wie z.B. GeneMark, benutzen auch Modelle höherer Ordnung, um Introns zu finden.
Um kodierende Regionen zu finden, werden naheliegenderweise 3-stufige Modelle verwendet, da bekannterweise je drei Nukleotide für eine Aminosäure kodieren. Die kodierenden Regionen werden durch drei Markov Modelle, eins für jede Position im Codon, definiert. Je höher die Ordnung des verwendeten Markov Modells ist, desto feiner kann die Abhängigkeit der voneinander abhängigen Nukleotide charakterisiert werden. Die meisten Programme wie GenMark oder GeneScan benutzten Markov Modelle der Ordnung 5 oder weniger um kodierende Regionen zu finden.

Ein **Hidden Markov Modell** besteht aus einer Markovkette, einer Reihe von Zuständen. Einige dieser Zustände können jedoch versteckt sein, d.h. man weiß nicht, in welchem Zustand sich das System befindet. In diesem Fall schließt man aus den beobachteten Effekten auf die Wahrscheinlichkeit jedes Zustands. Die Übergänge zwischen den Zuständen sind mit den Übergangswahrscheinlichkeiten P_{ij} gewichtet. Die Wahrscheinlichkeit der Kette setzt sich also aus den einzelnen Übergangswahrscheinlichkeiten zwischen den Zuständen zusammen. Die Werte werden dabei miteinander multipliziert.

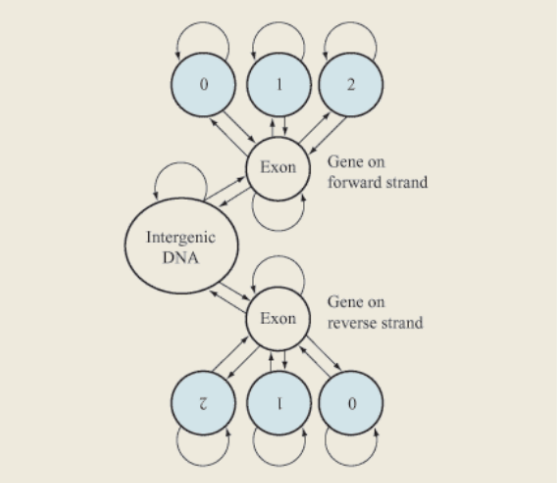


Figure 2

Abbreviated gene HMM model. The HMM is split into two symmetrical parts: genes on the forward or reverse strand of the DNA sequence (DNA sequence can be read in two directions). Each gene model contains a central exon state which has an emission of nucleotides tuned to recognize protein coding regions. Interrupting the exons are introns; three intron states are used, since there are three relative positions at which an intron can interrupt a coding triplet of DNA bases. These introns are distinguished by their "phase" — 0, 1, or 2.

Die einzelnen Methoden funktionieren nicht überall. Im Folgenden werden drei Beispiele gezeigt, in denen jeweils ein anderes Programm die besten Resultate zu einer gegebenen cDNA liefert:

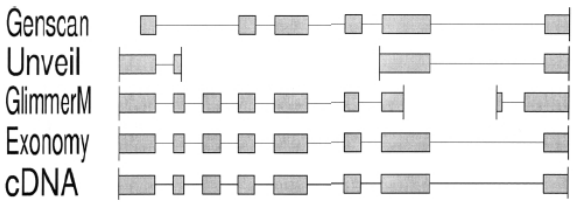


Figure 2. An example in which Exonomy produces the correct gene model.

Majoros et al. Nucl. Acids. Res. 31, 3601 (2003)

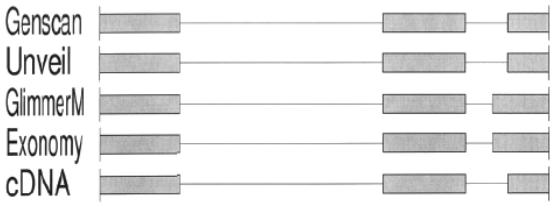


Figure 3. An example in which Unveil produces the correct gene model (as does Genscan).

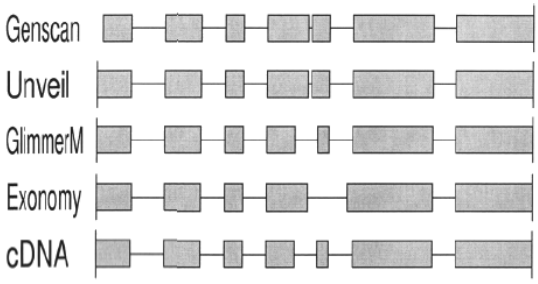


Figure 4. An example in which GlimmerM produces the correct gene model.

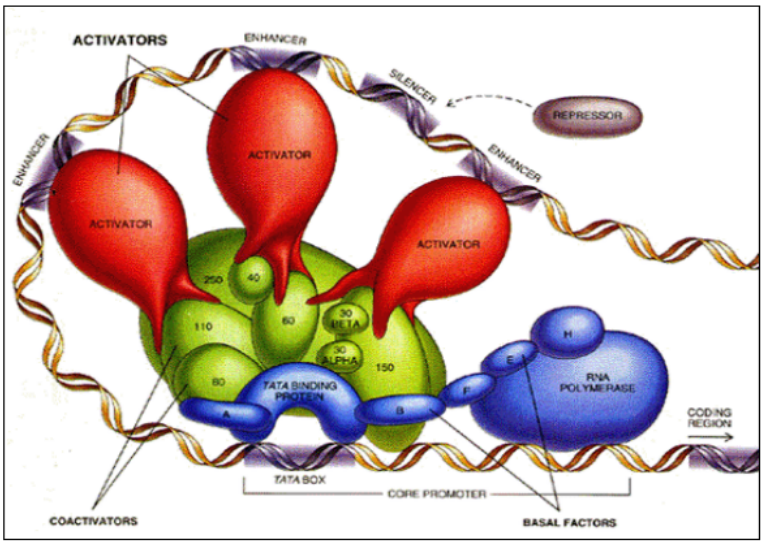
Majoros et al. Nucl. Acids. Res. 31, 3601 (2003)

Im ersten Bild erkennt das Programm Exonomy die Sequenzen am besten, im zweiten Beispiel liefen GenScan und Unveil ähnliche Ergebnisse und im letzten Beispiel erkennt GlimmerM die Sequenz am besten.
Die Resultate der Genvorhersage werden zuverlässiger, dennoch sollte man sie mit Vorsicht behandeln. Sie sind sehr nützlich um die Entdeckung von Genen zu beschleunigen. Dennoch sind biologische Techniken notwendig um die Existenz von virtuellen Proteinen zu bestätigen und um deren biologische Funktion zu finden bzw. zu beweisen.

Dadurch werden vergleichende Genom-Ansätze immer wichtiger, in denen Programme Genkandidaten auf Homologie mit exprimierten Sequenzen vergleichen (EST oder cDNA Sequenzdaten).

4.3 Transkription - Motivsuche

Die Transkription ist ein wesentlicher Teilprozess der Genexpression. Bei der Transkription wird ein Gen aus der DNA abgelesen und als RNA-Molekül vervielfältigt, das heißt ein spezifischer DNA-Abschnitt dient als Vorlage zur Synthese eines neuen RNA-Strangs. Hierbei werden die Nukleinbasen der DNA (A,T,G,C) durch die RNA-Polymerase in die Nukleinbasen der RNA (A,U,G,C) umgeschrieben, Thymin wird durch Uracil ersetzt.



http://www.berkeley.edu/news/features/1999/12/09_nogales.html

Die Maschinerie, die ein Gen transkribiert, besteht aus etwa 50 Proteinen. Das Enzym RNA-Polymerase setzt sich an eine Promotor-Sequenz und trennt dann die DNA-Doppelhelix durch Lösen der Wasserstoffbrücken auf. Am Antisense Strang der DNA lagern sich durch Basenpaarung komplementäre Ribonukleotide an. Die Ableserichtung der DNA auf dem Matrizenstrang verläuft vom 3'-Ende zum 5'-Ende, die Synthese der komplementären RNA dementsprechend von 5' nach 3'. Die Öffnung der DNA-Doppelhelix erfolgt nur in einem kurzen Bereich. Die RNA-Polymerase benötigt keinen Primer, am Terminator wird die Transkription beendet. Danach wird das mRNA-Transkript entlassen und die Polymerase löst sich von der DNA.

Der Vorgang der Transkription verläuft bei Eukaryoten und Prokaryoten grundsätzlich gleich. Bei Prokaryoten wird die Transkription über einen Operator gesteuert, bei den Eukaryoten werden ein Enhancer oder Silencer benötigt, eine Gruppe von Transkriptionsfaktoren bindet an die DNA gerade oberhalb der Stelle des Kern-Promoters, während assoziierte Aktivatoren an Enhancer-Regionen weiter oberhalb der Stelle binden.

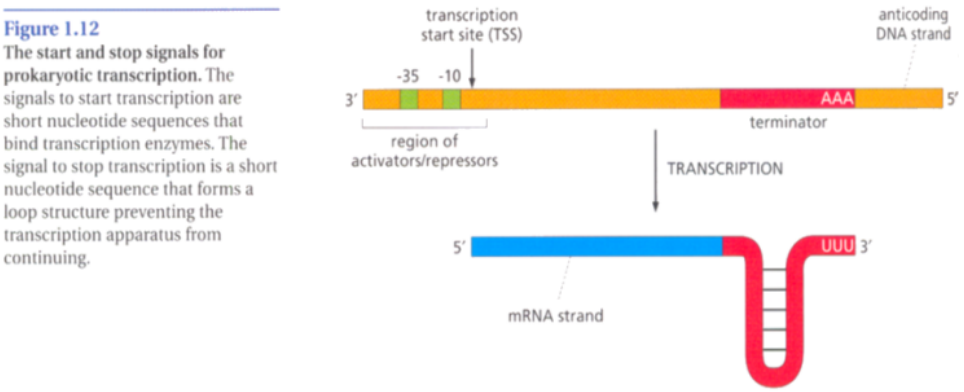


Figure 1.12
The start and stop signals for prokaryotic transcription. The signals to start transcription are short nucleotide sequences that bind transcription enzymes. The signal to stop transcription is a short nucleotide sequence that forms a loop structure preventing the transcription apparatus from continuing.

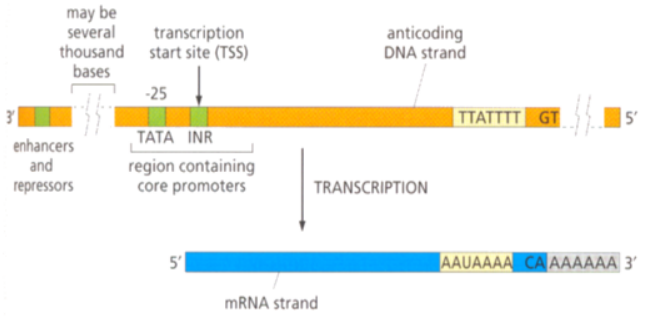


Figure 1.13
The start and stop signals for eukaryotic transcription. All the signals are short sequences that bind enzymes involved in this complex process.

Um prokaryotische Promoter zu analysieren kann man eine Menge von Promotersequenzen bzgl. der Position alignieren, die den bekannten Transkriptionsstart markieren und in den Sequenzen nach konservierten Regionen suchen. Beispielsweise enthalten *E.coli* Promotoren drei konservierte Sequenzmerkmale:

- eine etwa 6bp lange Region mit dem Konsensusmotif TATAAT bei Position -10,
- eine etwa 6bp lange Region mit dem Konsensusmotif TTGACA bei Position -35,
- die Distanz zwischen den beiden Regionen von etwa 17bp ist relativ konstant

Transkriptionsfaktorbindestellen mit einem Computerprogramm zu identifizieren ist schwierig, da diese aus kurzen, entarteten Sequenzen bestehen, die häufig ebenfalls durch Zufall auftreten.

Probleme treten auch auf, wenn man versucht die Region einzugrenzen. Zum einen ist die Länge des Motivs nicht bekannt. Außerdem müssen die Sequenzen, mit denen man nach dem Motiv sucht nicht notwendigerweise dem gesamten Promoter entsprechen. Die Promotoren verschiedener Gene, die man untersuchen will, können bereits durch einen Clusteralgorithmus, der gewissen Beschränkungen unterliegt, zu Gruppen zusammengefasst worden sein.

Strategie 1:
Diese wird seit der Verfügbarkeit von Microarray Gen-Expressionsdaten eingesetzt.

1. Stelle Gruppen von Genen mit ähnlichen Expressionsprofilen zusammen (z.B. solche, die zur selben Zeit im Zellzyklus aktiviert sind)
2. Als Annahme nutzt man, dass dieses Profil, zumindest teilweise, durch eine ähnliche Struktur der für die transkriptionelle Regulation verantwortlichen regulatorischen Regionen verursacht wird.
3. Suche nach gemeinsamen Motiven in < 1000 Basen upstream Region.

Bis heute wurde vor allem nach einzelnen Motiven gesucht (als TFBindestellen), die in den Promotoren von möglicherweise ko-regulierten Genen gemeinsamen auftreten.

Besser scheint die Suche nach dem gleichzeitigen Auftreten von zwei oder mehr Stellen in einem vorgegebenen Abstand zu sein, da dadurch die Suche empfindlicher wird!

Strategie 2: Erschöpfende Motivsuche in upstream-Regionen

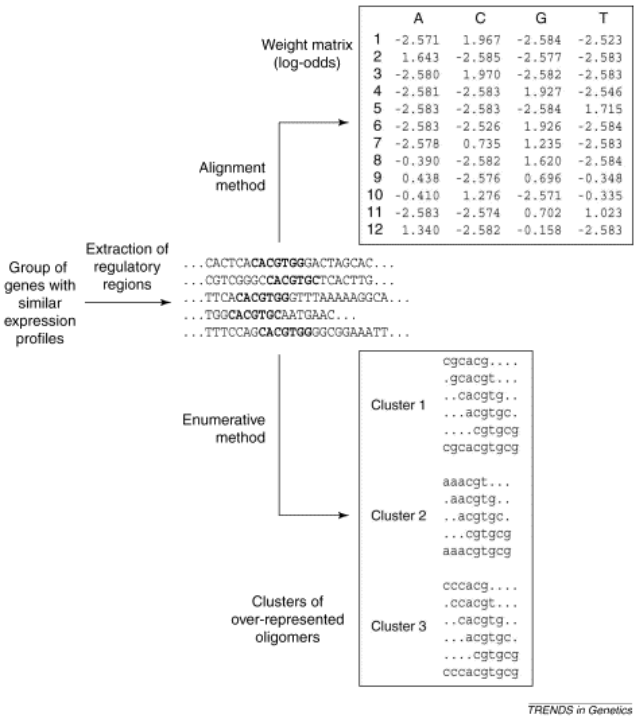
1. Benutze die Beobachtung, dass relevante Motive sich in der upstream-Region oft viele Male wiederholen, unter Umständen mit kleinen Variationen, damit die regulatorische Wirkung effektiv ist.

2. Suche in der upstream-Region nach überrepräsentierten Motiven

3. Ordne Gene nach den überrepräsentierten Motiven

4. Analysiere Gruppen von Genen, die Motive für Ko-Regulation in Microarray-Experimenten gemeinsam haben

5. Betrachte überrepräsentierte Motive, die Gruppen von ko-regulierten Genen als mögliche Bindungsstellen markieren.



	A	C	G	T
1	-2.571	1.967	-2.584	-2.523
2	1.643	-2.585	-2.577	-2.583
3	-2.580	1.970	-2.582	-2.583
4	-2.581	-2.583	1.927	-2.546
5	-2.583	-2.583	-2.584	1.715
6	-2.583	-2.526	1.926	-2.584
7	-2.578	0.735	1.235	-2.583
8	-0.390	-2.582	1.620	-2.584
9	0.438	-2.576	0.696	-0.348
10	-0.410	1.276	-2.571	-0.335
11	-2.583	-2.574	0.702	1.023
12	1.340	-2.582	-0.158	-2.583

Fig. 1. A flowchart to illustrate the two different approaches for motif identification. We analyzed 800 bp upstream from the translation start sites of the five genes from the yeast gene family PHO by the publicly available systems MEME (alignment) and RSA (exhaustive search, see Table 1). MEME was run on both strands, one occurrence per sequence mode, and found the known motif ranked as second best. RSA Tools was run with oligo size 6 and noncoding regions as background, as set by the demo mode of the system. The well-conserved heptamer of the motifs used by MEME to build the weight matrix is printed in bold.

Ohler, Niemann Trends Gen 17, 2 (2001)

Table 1: Significant motifs associated to nucleolar proteins implicated in ribosome biogenesis. The columns titled "C", "F" and "P" correspond to the three branches of the Gene Ontology: cellular component, molecular function and biological process respectively.

motif	C	F	P
GATGAGA	nucleolus	-	ribosome biogenesis
GATGAGAT	nucleolus	-	ribosome biogenesis
ATGAGAT	nucleolus	-	ribosome biogenesis
ATGAGATG	-	-	ribosome biogenesis
TGAGATG	-	-	ribosome biogenesis and assembly
TGAGATGA	-	-	ribosome biogenesis and assembly
GAGATG	-	-	ribosome biogenesis and assembly
GAGATGAG	nucleolus	-	ribosome biogenesis and assembly
GAGATGA	nucleolus	-	ribosome biogenesis and assembly
AGATGAG	nucleolus	-	ribosome biogenesis
GATGAG	nucleolus	-	ribosome biogenesis
GATGA	-	-	ribosome biogenesis
ATGAGCT	nucleolus	-	ribosome biogenesis
TGAGCT	nucleolus	-	rRNA processing
GATGAGATGAGCT			
AAAAATT	nucleolus	-	ribosome biogenesis
AAAAATTT	nucleolus complex	-	transcription from Pol I promoter
AAAATT	nucleolus	-	ribosome biogenesis
AAAATTT	nucleolus	-	ribosome biogenesis
AAAATTTT	nucleolus	-	ribosome biogenesis
AAATT	nucleolus	-	35S primary transcript processing
AAATTTTC	small nucleolar ribonucleoprotein complex	-	35S primary transcript processing
AAAAATTTTC			

Cora et al. BMC Bioinformatics 5, 57 (2004)

Aktuelle Verfahren um Promotoren zu finden:

Table 1. A selection of recently published promoter finding and analysis tools accessible on the World-Wide Web

Program	Description	URL
General promoter finding		
Promoter2.0	Search-by-signal, artificial neural network	www.cbs.dtu.dk/services/Promoter
NNPP	Search-by-signal, time delay neural network	www.fruitfly.org/seq_tools/promoter.html
PromoterInspector	Search-by-content, class-specific oligomers	www.gsf.de/biodv
McPromoter V3	Signal/content, stochastic segment model/neural network	www.mustererkennung.de/HTML/English/Research/Promoter
CorePromoter	Signal/content, discriminant analysis	argon.cshl.org
Promoter analysis tools		
RSA Tools	Yeast and microbial exhaustive search	www.ucmb.ulb.ac.be/bioinformatics/rsa-tools
Gibbs sampler	Alignment method	bayesweb.wadsworth.org/gibbs/gibbs.html
MEME	Alignment via Expectation Maximization	meme.sdsc.edu
BBA	Phylogenetic footprinting by Bayes alignment	bayesweb.wadsworth.org/cgi-bin/bayes_align12.pl
Pip Maker	Phylogenetic footprinting by identity plots	bio.cse.psu.edu

Ohler, Niemann Trends Gen 17, 2 (2001)

Positionsspezifische Gewichtsmatrix:
Das Verfahren der PSSM (für *position specific scoring matrix*) ist ein populäres verfahren, wenn es eine Liste von Genen gibt, die ein TF-Bindungsmotiv gemeinsam haben. Bedingung hierfr ist, dass gute Multiple Sequenz Alignments müssen vorhanden sein.

Die Alignment-Matrix gibt an, wie häufig die verschiedenen Buchstaben an jeder Position im Alignment auftreten. Daraus wird, ähnlich zu der Konstruktion der Austauschmatrizen für Sequenzalignments die PSSM-Matrix erzeugt.

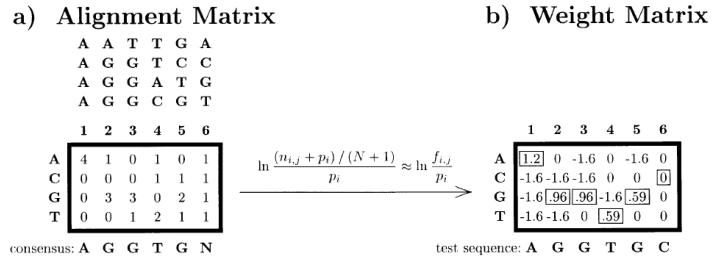


Fig. 1. Examples of the simple matrix model for summarizing a DNA alignment. (**a**) An alignment matrix describing the alignment of the four 6-mers on top. The matrix contains the number of times, $n_{i,j}$, that letter i is observed at position j of this alignment. Below the matrix is the consensus sequence corresponding to the alignment (N indicates that there is no nucleotide preference). (**b**) A weight matrix derived from the alignment in (a). The formula used for transforming the alignment matrix to a weight matrix is shown above the arrow. In this formula, N is the total number of sequences (four in this example), p_i is the *a priori* probability of letter i (0.25 for all the bases in this example) and $f_{i,j} = n_{i,j}/N$ is the frequency of letter i at position j. The numbers enclosed in blocks are summed to give the overall score of the test sequence. The overall score is 4.3, which is also the maximum possible score with this weight matrix.

Hertz, Stormo (1999) Bioinformatics 15, 563

4.4 TRANSFAC

TRANSFAC ist eine Datenbank für die Bindungsstellen von eukaryotischen Transkriptionsfaktoren. Entwickelt und vertrieben wird sie von der Firma BIOBase / der TU Braunschweig / und der Gesellschaft für Biotechnologische Forschung (GBF).
Die Datenbank ist online zugänglich und besteht aus einem öffentlichen und einem kommerziellen Teil.
Das linke Bild unten zeigt, wie viele Einträge in der öffentlichen sowie in der kommerziellen Datenbank enthalten sind. Die Benutzung der öffentlichen Datenbank ist nach einer Registrierung frei.
Die Datenbank ist in sechs Unterdatenbanken aufgeteilt:

FACTOR Wechselwirkung von TFs

SITE ihre DNA-Bindungsstelle

GENE durch welche sie diese Zielgene regulieren

CELL wo kommt Faktor in Zelle vor?

MATRIX TF Nukleotid-Gewichtungsmatrix

CLASS Klassifizierungsschema der TFs

Das rechte Bild zeigt, wie viele Einträge in den einzelnen Datenbanken enthalten sind und wie die Daten sich zusammensetzen.

	Public	Professional
Feature	**TRANSFAC 7.0** (2005)	**TRANSFAC 2008.3** (2008)
Factor	6,133	11,683
Site	7,915	30,227
ChIP-chip Fragment	N/A	141,595
Gene, total	2,397	33,159
Matrix	398	856
Reference	N/A	18,959
Plant Data		
Factor	2,248	3,229
Site, total	731	1,928
Site, genomic	345	719
Gene	346	1,140
Matrix	29	102

http://www.gene-regulation.com/faqs/TFsubscription.html

Table 1. Number of entries in the tables of TRANSFAC® 7.0 and TRANSCompel® 7.0

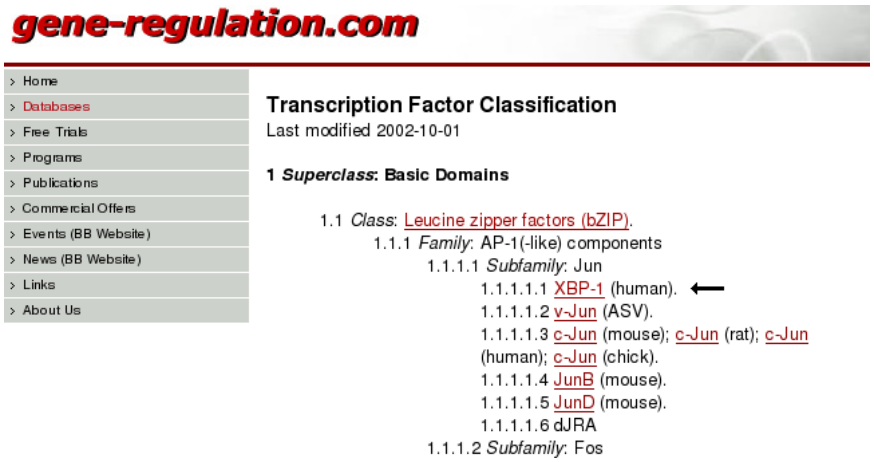
Table	TRANSFAC® Rel. 7.0
FACTOR	6133
Homo sapiens	1040
Mus musculus	765
D.melanogaster	233
A.thaliana	1751
S.cerevisiae	368
SITE	7915
MATRIX	398
GENE (all entries)	2397
H.sapiens	608
M.musculus	417
D.melanogaster	145
A.thaliana	115
S.cerevisiae	195
GENE (entries with SITE links)	1504
CLASS	50
CELL	1307
	TRANSCompel® Rel. 7.0
COMPEL (composite elements)	322

Matys et. al, Nucleic Acids Res. 2006 Jan 1;34; D108

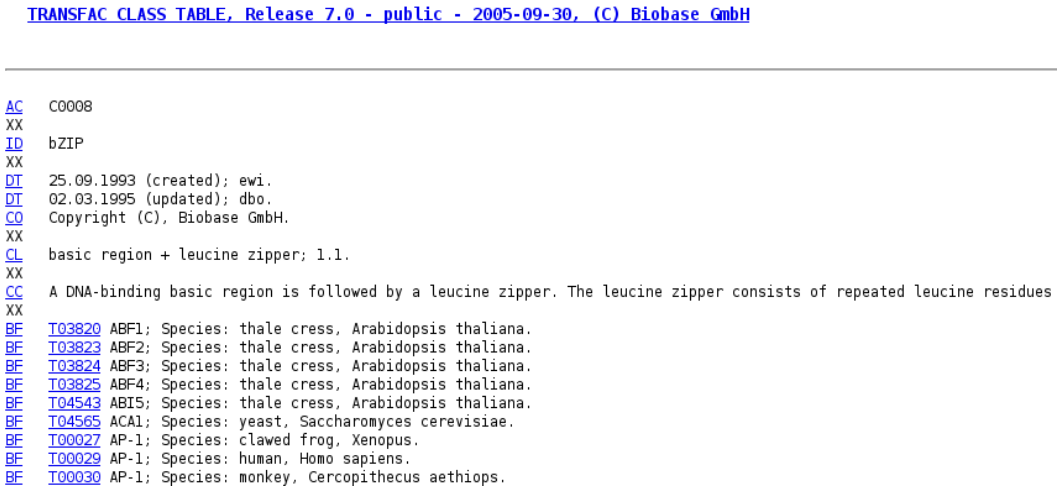
TRANSFAC arbeitet auch mit Klassifizierungen.

1. Superklasse basische Domänen
   - Leuzin-zipper Faktoren (bZIP)
   - Helix-Loop-Helix Faktoren (bHLH)
   - bHLH-bZIP Faktoren mit Kontakt in der
   - NF-1 Minor Groove
   - RF-X
   - bHSH
2. Superklasse: Zink-koordinierende DNA-bindende Domänen
   - Cys4 Zinkfinger vom Typ nuklearer Rezeptor
   - verschiedene Cys4 Zinkfinger
   - Cys2His2 Zinkfinger Domänen
   - Cys6 Cystein-Zink Cluster
   - Zinkfinger mit abwechselnder Zusammensetzung
3. Superklasse: Helix-turn-helix
4. Superklasse: beta-Scaffold
5. Superklasse: andere

Von der Startseite aus gelangt man über Databases/ Classification zu der Unterteilung der Klassen.

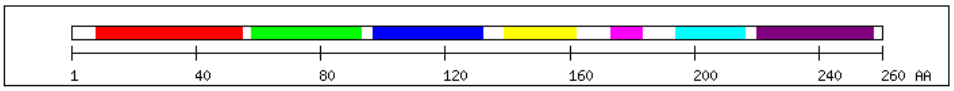


Wählt man dort den ersten Eintrag der ersten Superklasse für die Klasse: 1.1 Class: Leucine zipper factors (bZIP) erhält man die Liste aller Vorkommen des Leucin Zippers in den einzelnen Spezies.



Wählt man den Eintrag (Pfeil) für die Unterklasse der Leuzin Zipper, erhält man die detaillierten Ergebnisse. Die einzelnen Regionen, die in diesem Teilstück der Sequenz gefunden wurden, sind angegeben und farblich hervorgehoben.

```
AC   T00902
XX
ID   T00902
XX
DT   12.03.1993 (created); ewi.
DT   17.09.2002 (updated); mkl.
CO   Copyright (C), Biobase GmbH.
XX
FA   XBP-1
XX
SY   hXBP-1; TREB-5; TREB5; X box binding protein 1; X-box binding protein 1; XBP-1; XBP1; XBP2.
XX
OS   human, Homo sapiens
OC   eukaryota; animalia; metazoa; chordata; vertebrata; tetrapoda; mammalia; eutheria; primates
XX
GE   G006882 XBP1; HGNC: XBP1.
XX
CL   C0008 bZIP; 1.1.1.1.1..
XX
SZ   260 AA; 28.7 kDa (cDNA), 38 kDa (SDS) [4]
XX
SQ   MVVVAAAPNPADGTPKVLLLSGQPASAAGAPAARLPLMVPAQRGASPEAASGGLPQARKR
SQ   QRLTHLSPEEKALRRKLKNRVAAQTARDRKKARMSELEQQVVDLEEENQKLLLENQLLRE
SQ   KTHGLVVENQELRQRLGMDALVAEEEAEAKGNEVRPVAGSAESAALRLRAPLQQVQAQLS
SQ   PLQNISPWILAVLTLQIQSLISCWAFWTTWTQSCSSNALPQSLPAWRSSQRSTQKDPVPY
SQ   QPPFLCQWGRHQPSWKPLMN
XX
SC   SwissProt #P17861
XX
FT        8     55   p   proline-rich region (9/48).
FT       58     93   p   basic region (16/36).
FT       67    131   -   PF00170; bZIP.
FT       67    131   p   SM00338; BRLZ.
FT       97    132   p   leucine zipper (L6).
FT      139    162       acidic region (7/24).
FT      173    183   p   glutamine-rich region (5/11).
FT      194    216   -   serine-/threonine-rich region (9/23).
FT      220    257   -   glutamine-/proline-rich region (13/38).
```




```
XX
SF   conflicting data: A33,R34 is G33,Q34,A35 in [4];
XX
CP   ubiquitous [4].
XX
FF   binds to the X-box elements of class II MHC genes [3] [2] [1];
FF   different from RF-X;
XX
IN   T00122 c-Fos; mouse, Mus musculus.
IN   T00123 c-Fos; human, Homo sapiens.
```

5 Proteine

5.1 Proteinfunktion

Proteine übernehmen im Körper unterschiedliche Funktionen. Es gibt Strukturproteine wie z.B. die Hüllenproteine von Viren, die die Außenhülle des Kapsid bilden oder Proteine, die am Aufbau des Zytoskeletts beteiligt sind. Die Enzyme sind dafür verantwortlich, dass chemische Reaktionen im Körper katalysiert werden. Es gibt die Transport- und Speicherproteine, wie z.B. das Hämoglobin, das den lebensnotwendigen Sauerstoff aufnimmt und zu den vorgesehenen Orten transportiert. Hormone und Rezeptoren/Signalübertragungsproteine steuern gewisse Vorgänge im Körper. In der letzten Vorlesung wurden die Transkriptionsproteine, die für die Transkription von DNA zu RNA verantwortlich sind, behandelt. Ferner gibt es noch Proteine, die für Erkennungsvorgänge verantwortlich sind, die Zelladhäsionsproteine oder die Antikörper.
Die Funktion eines Proteins wird meist nur durch einen kleinen Teil des Gesamtproteins bestimmt. Dieser Teil wird das aktive Zentrum genannt. Der überwiegende Rest des Proteins scheint hingegen für die eigentliche Funktion zunächst unbedeutend. Offensichtlich garantiert dieser Rest die Stabilität des Proteins, besonders die korrekte Orientierung der Aminosäuren im aktiven Zentrum. Der Rest kann jedoch weiterhin durch die eigene konformationelle Dynamik auf Einflüsse von außen reagieren und sogar die Effizienz chemischer Umwandlungen im aktiven Zentrum beeinflussen. Außerdem bietet die restliche Oberfläche Bindungsstellen für Interaktionspartner an.

5.2 Proteinaufbau

Proteine sind große Moleküle im Vergleich zu Ligandenmolekülen wie z.B. ATP oder Wirkstoffe. Linderström-Lang und Schellmann (In The Enzymes, Hrsg. P.D. Boyer, Band 1, 2. Auflage, Seiten 443-510; Academic Press, New York, 1959) ordneten den Aufbau eines Proteins hierarchisch in:

- **Primärstruktur** - Aminosäuresequenz
- **Sekundärstruktur** - durch Wasserstoffbrücken stabilisierte Strukturen wie α-Helices und β-Faltblätter, sowie Verbindungsstücke zwischen diesen, z.B. turns.
- **Tertiärstruktur** - Die gesamte Faltung einer Kette, die sich aus der Packung der Sekundärstrukturelemente ergibt. Es gibt z.B. β-barrels, $\beta\alpha\beta$-Einheiten oder Greek-Keys usw.
- **Quartärnere Struktur** - Die Anordnung der Tertiärstrukturen mehrerer Ketten eines Proteins, das mehrere Untereinheiten besitzt.
- **Proteinkomplexe** - Komplexe aus mehreren Proteinen.

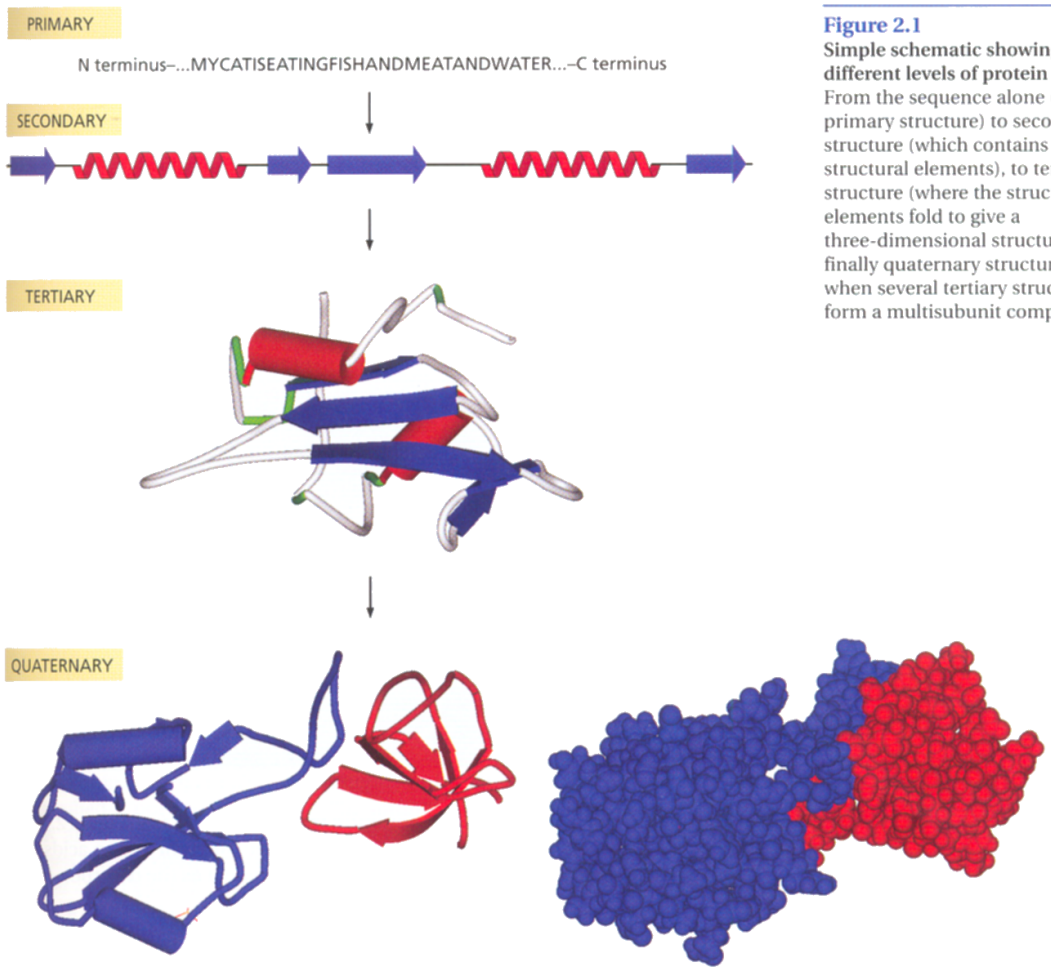


Figure 2.1
Simple schematic showing the different levels of protein structure. From the sequence alone (the primary structure) to secondary structure (which contains local structural elements), to tertiary structure (where the structural elements fold to give a three-dimensional structure), to finally quaternary structure found when several tertiary structures form a multisubunit complex.

Welche „Kräfte“ sind für die Ausbildung der verschiedenen „Strukturen“ wichtig?

5.2.1 Hydrophober Effekt

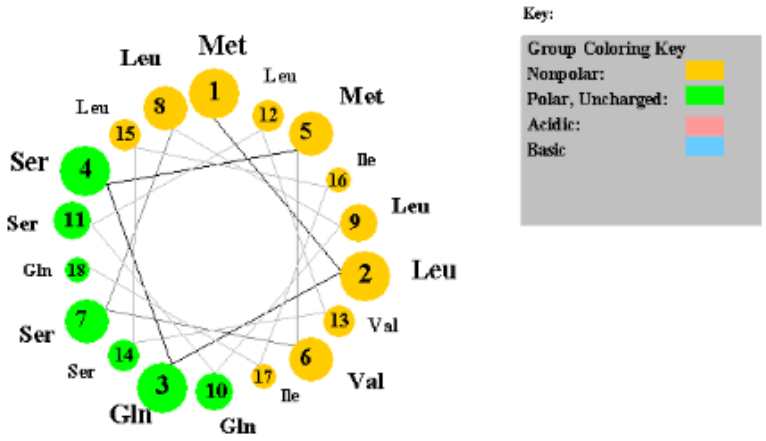
Für lösliche Proteine bestimmt im wesentlichen der hydrophobe Effekt die Faltung des Proteins. Für Membranproteine gilt das hauptsächlich für den äußeren Teil, sie sind im Transmembranbereich außen hydrophober als innen.

Der hydrophobe Effekt basiert auf polaren Wechselwirkungen. Wird eine unpolare Substanz aus einem organischen bzw. unpolaren Lösungsmittel in ein polares Lösungsmittel wie beispielsweise Wasser überführt, ist das energetisch sehr ungünstig, da zwischen den polaren Wassermolekülen und den unpolaren (=hydrophoben) Proteinteilen keine Wasserstoffbrückenbindungen gebildet werden können. Zwar sinkt die Entropie des Proteins durch dessen Faltung stark ab, aber der Gewinn an Entropie im umgebenden Wasser überwiegt. Das Protein faltet sich wie von selbst. Der hydrophobe Effekt bewirkt bei Raumtemperatur eine Abnahme der Entropie, sowie eine Zunahme der Wärmekapazität.

Der Beitrag hydrophober Wechselwirkungen zur freien Enthalpie bei der Proteinfaltung und der Protein-Liganden Wechselwirkung kann als proportional zur Größe der während dieser Prozesse vergrabenen hydrophoben Oberfläche angesehen werden. Die typischen Oberflächen, die gefunden werden, sind Methan CH_4 und Benzol CH_6. Zur Erinnerung: Hydrophobe Aminosäuren sind:

- aliphatisch: Ile, Val, Leu
- aromatisch: Phe, Tyr, Trp
- sonstige: Ala, Pro, Cys, Met

Die folgende Sequenz aus 18 Aminosäuren MLQSMVSLLQSLVSLIIQ wird als Helix dargestellt. Mittels eines Helikalen Rades kann man leicht erkennen, welche Aminosäuren dem Proteininneren zugewandt sind, bzw. welche Aminosäuren der Lösung zugewandt sind. Im folgenden Bild sind die hydrophoben Aminosäuren alle auf einer Seite (gelb), diese zeigen ins Proteininnere oder in eine Phospholipidmembran. Die restlichen polaren Aminosäuren (grün) zeigen in die wässrige Lösung. Dies ist ein perfektes Beispiel für eine amphipatische Helix. Die Beispielsequenz gehört zum Peptid Magainin.



http://cti.itc.virginia.edu/~cmg/Demo/wheel/wheelApp.html

5.2.2 Peptidbindungen

In Peptiden und Proteinen sind die Aminosäuren zu langen Ketten verknüpft, indem zwei aufeinanderfolgende Aminosäuren jeweils über eine „Peptidbindung“ verbunden sind.

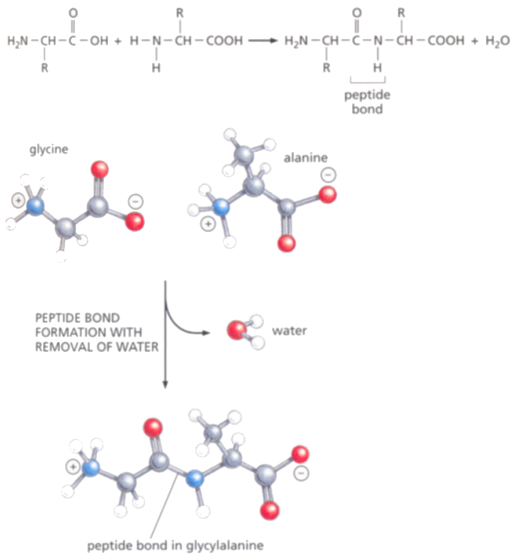
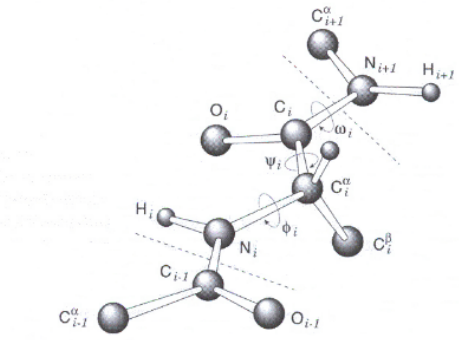


Figure 2.4
Peptide bonds. (A) gives the chemical formulae of the peptide bond that is formed between amino acids to make a polypeptide chain. (B) illustrates the above in a diagrammatic form. (B, from B. Alberts et al., Molecular Biology of the Cell, 4th ed. New York: Garland Science, 2002.)

Die Kenntnis der Aminosäuresequenz eines Proteins allein verrät allerdings noch nicht viel über seine Funktion. Entscheidend ist seine drei-dimensionale Struktur.

Die C-N Bindungslänge in der Peptidbindung beträgt 1.33 Å. Sie liegt damit zwischen 1.52 Å und 1.25 Å, den Bindungslängen für eine Einfach- bzw. Doppelbindung sind. Die benachbarte C=O Bindung hat eine Länge von 1.24 Å, was etwas länger als eine typische Carbonyl- C=O Doppelbindung ist (1.215 Å). Die Peptidbindung hat einen teilweise konjugierten Charakter und ist nicht frei drehbar. Somit bleiben pro Residue nur zwei frei drehbare Diederwinkel des Proteinrückgrats übrig.
Die dreidimensionale Faltung des Proteins wird vor allem durch die Diederwinkel des Proteinrückgrats bestimmt. Die beiden frei drehbaren Diederwinkel werden mit ϕ (phi) und ψ (psi) bezeichnet.



Definition der Konformationswinkel im Polypeptidrückgrat.

In den Sekundärstruktur-Konformationen α-Helix und β-Faltblatt bilden sich die Wasserstoffbrückenbindungen jeweils zwischen den C=O und N-H Atomen des Rückgrats. Pro Wasserstoffbrücke ergibt sich eine hydrophobe Stabilisierung von ca. 20kJ/mol. Zum Vergleich: Wechselwirkungen zwischen geladenen Aminosäureseitenketten und/ oder Ionen

(sog. Salzbrücken) betragen etwa das 10-fache, Wechselwirkungen zwischen ungeladenen bzw. unpolaren nicht-gebundenen Atomkontakte dagegen nur 2kJ/mol pro Paarung. Deshalb sind α-Helices und β-Faltblätter stabile strukturelle Einheiten.

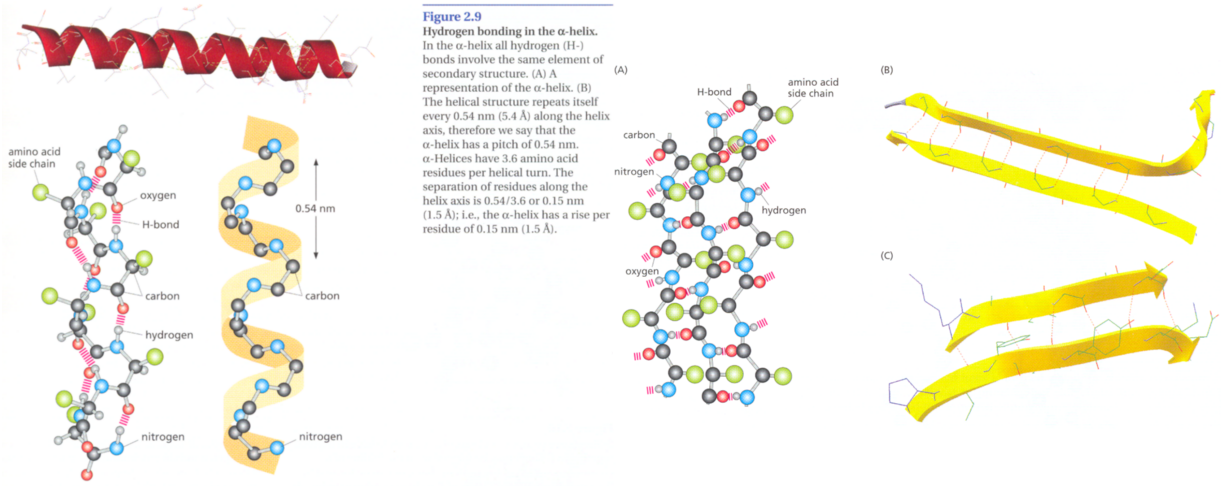
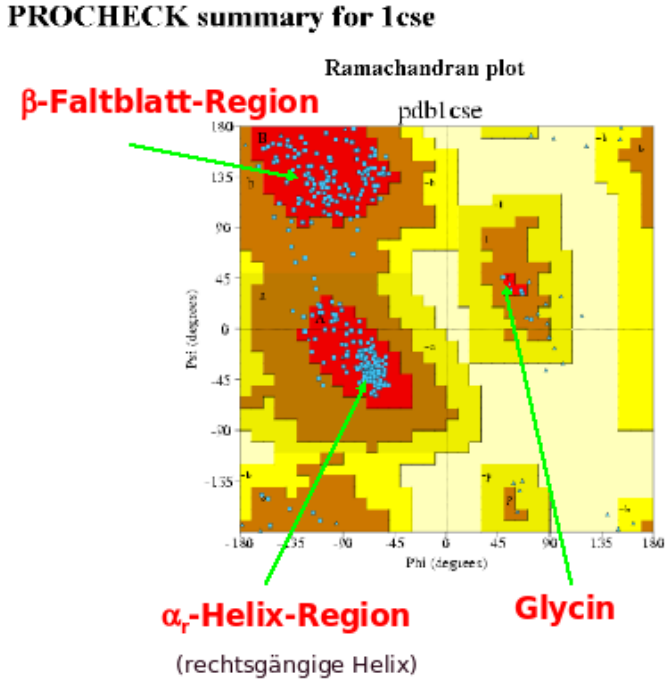


Figure 2.9
Hydrogen bonding in the α-helix. In the α-helix all hydrogen (H-) bonds involve the same element of secondary structure. (A) A representation of the α-helix. (B) The helical structure repeats itself every 0.54 nm (5.4 Å) along the helix axis, therefore we say that the α-helix has a pitch of 0.54 nm. α-Helices have 3.6 amino acid residues per helical turn. The separation of residues along the helix axis is 0.54/3.6 or 0.15 nm (1.5 Å); i.e., the α-helix has a rise per residue of 0.15 nm (1.5 Å).

Die gefaltete Struktur eines Proteins ist die Konformation, die die günstigste freie Enthalpie ΔG für diese Aminosäuresequenz besitzt.
Der **Ramachandran-Plot** charakterisiert die energetisch günstigen Bereiche des Aminosäurerückgrats. Die einzige Residue, die außerhalb der erlaubten Bereich liegen darf, also alle möglichen Torsionswinkel annehmen kann, ist Glycin. Der einfache Grund hierfür ist, dass Glycin keine Seitenkette hat und daher leichter verdrehbar ist.



Proteinstrukturen, die anhand von Homologiemodellen (s. Vorlesung 6) erstellt wurden,

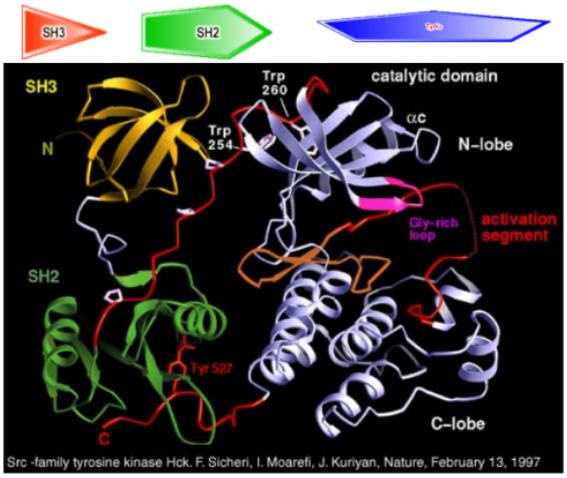
sollten möglichst wenig Aminosäuren außerhalb der erlaubten Bereiche aufweisen.
Im Faltungsmuster der Molekülketten gibt es kompakte Bereiche, die den Anschein haben auch unabhängig von den restlichen Molekülen stabil zu sein. Diese kompakten Bereiche werden als Domäne bezeichnet.

5.2.3 Modular aufgebaute Proteine

Modular aufgebaute Proteine bestehen aus mehreren Domänen. Die Anwendung von SMART (*www.smart.embl-heidelberg.de*) für die Src-Kinase HcK mit der Sequenz

```
MGGRSSCEDP GCPRDEERAP RMGCMKSKFL QVGGNTFSKT ETSASPHCPV
YVPDPTSTIK PGPNSHNSNT PGIREAGSED IIVVALYDYE AIHHEDLSFQ
KGDQMVVLEE SGEWWKARSL ATRKEGYIPS NYVARVDSLE TEEWFFKGIS
RKDAERQLLA PGNMLGSFMI RDSETTKGSY SLSVRDYDPR QGDTVKHYKI
RTLDNGGFYI SPRSTFSTLQ ELVDHYKKGN DGLCQKLSVP CMSSKPQKPW
EKDAWEIPRE SLKLEKKLGA GQFGEVWMAT YNKHTKVAVK TMKPGSMSVE
AFLAEANVMK TLQHDKLVKL HAVVTKEPIY IITEFMAKGS LLDFLKSDEG
SKQPLPKLID FSAQIAEGMA FIEQRNYIHR DLRAANILVS ASLVCKIADF
GLARVIEDNE YTAREGAKFP IKWTAPEAIN FGSFTIKSDV WSFGILLMEI
VTYGRIPYPG MSNPEVIRAL ERGYRMPRPE NCPEELYNIM MRCWKNRPEE
RPTFEYIQSV LDDFYTATES QYQQQP
```

ergibt:



Die Klassifikation von Proteinstrukturen nimmt in der Bioinformatik eine Schlüsselposition ein, weil sie das Bindeglied zwischen Sequenz und Funktion darstellt. (Klassifikation z.B. mit SCOP und CATH $->$ Vorlesung 1)

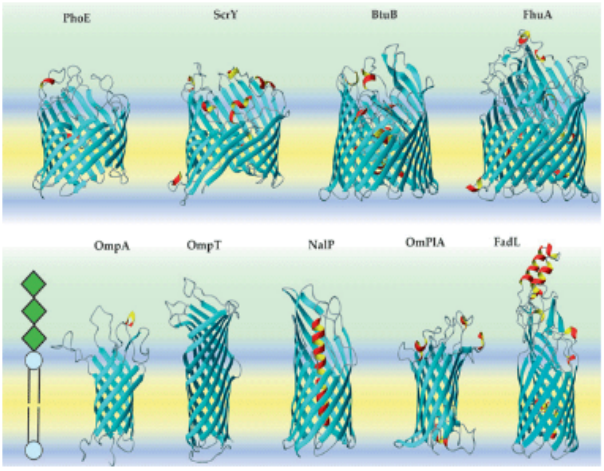
Die allgemeinste Einteilung in Familien von Proteinstrukturen stützt sich auf die Sekundär- und Tertiärstrukturen:

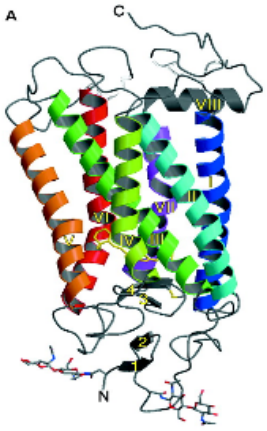
Klasse	Merkmale
α-helikal	Sekundärstruktur ausschließlich oder fast ausschließlich α-Helix
β-Faltblatt	Sekundärstruktur ausschließlich oder fast ausschließlich β-Faltblatt
$\alpha + \beta$	α-Helix und β-Faltblatt getrennt in verschiedenen Molekülteilen; kein β-α-β als Supersekundärstruktur
α/β	Helices und Faltblätter aus β-α-β-Einheiten zusammengesetzt
– α/β-linear	Mittellinie von Strängen der Faltblätter ungefähr linear
– α/β-Tonnen (*barrels*)	Mittellinie von Strängen der Faltblätter ungefähr kreisförmig
wenig oder gar keine Sekundärstruktur	

Lesk-Buch

5.2.4 Topologie von Membranproteinen

Im Inneren der Lipidschicht kann das Proteinrückgrat keine Wasserstoffbrückenbindungen mit den Lipiden ausbilden. Die Atome des Rückgrats müssen miteinander Wasserstoffbrückenbindungen ausbilden, wodurch sie entweder α-helikale oder β-Faltblattkonformation annehmen müssen. Die hydrophobe Umgebung erzwingt, dass (zumindest die bisher bekannten) Strukturen von Transmembranproteinen entweder reine β-Barrels (links) oder reine α-helikale Bündel (rechts) sind.

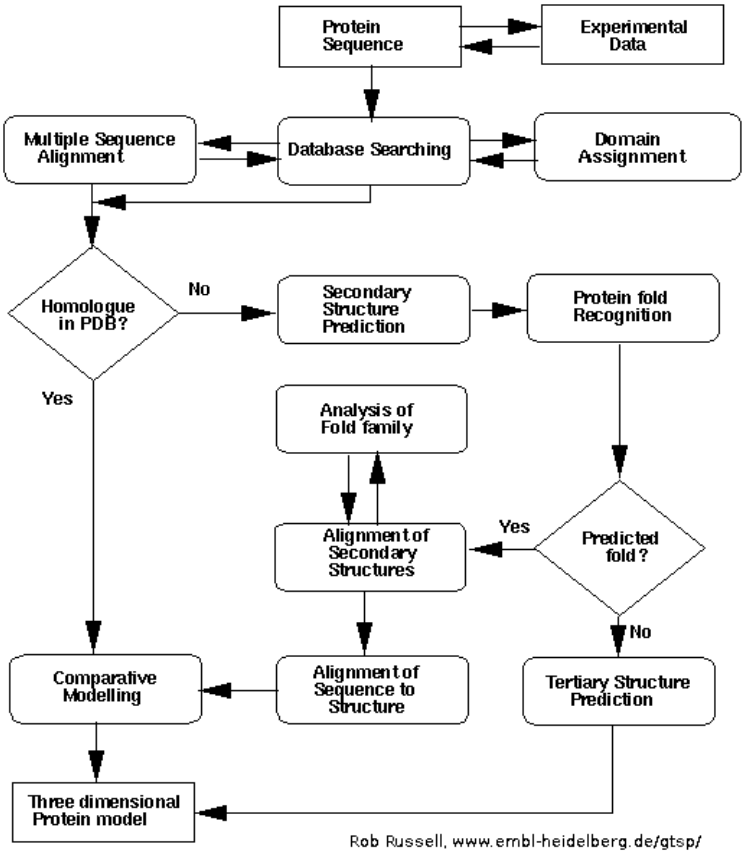




Zusammenfassung:

Die Sekundärstrukturelemente α-Helix und β-Faltblatt werden durch energetisch günstige Wasserstoffbrücken zwischen Atomen des Peptidrückgrats gebildet und sind sequenzunabhängig. Die Proteinfaltungen ergeben sich durch die Assemblierung der Sekundärstrukturelementen.
Der Ramachandran-Plot ist ein wichtiges Werkzeug um die Güte von Proteinstrukturen (bzw. -modellen) zu beurteilen.
Proteine sind oft modular aus mehreren Domänen aufgebaut.

5.3 Sekundärstrukturvorhersage



Diese Abbildung von Rob Russell gibt einen Überblick über die verschiedenen Verfahren zur Konstruktion von dreidimensionalen Strukturmodellen für Proteine.
Zunächst werden wir die Aufgabe betrachten, für eine gegebene Aminosäuresequenz die dazu gehörige Sekundärstruktur vorherzusagen. Die am häufigsten vorkommenden Strukturelemente sind die α-Helix, das β-Faltblatt und der Loop (Coil). Demzufolge wird sich die Sekundärstruktuvorhersage auf die Klassifizierung einer Inputsequenz in Regionen beziehen, die als Helikal (H) also Helix, Beta (E) Faltblatt oder Coil (C) vorhergesagt werden.

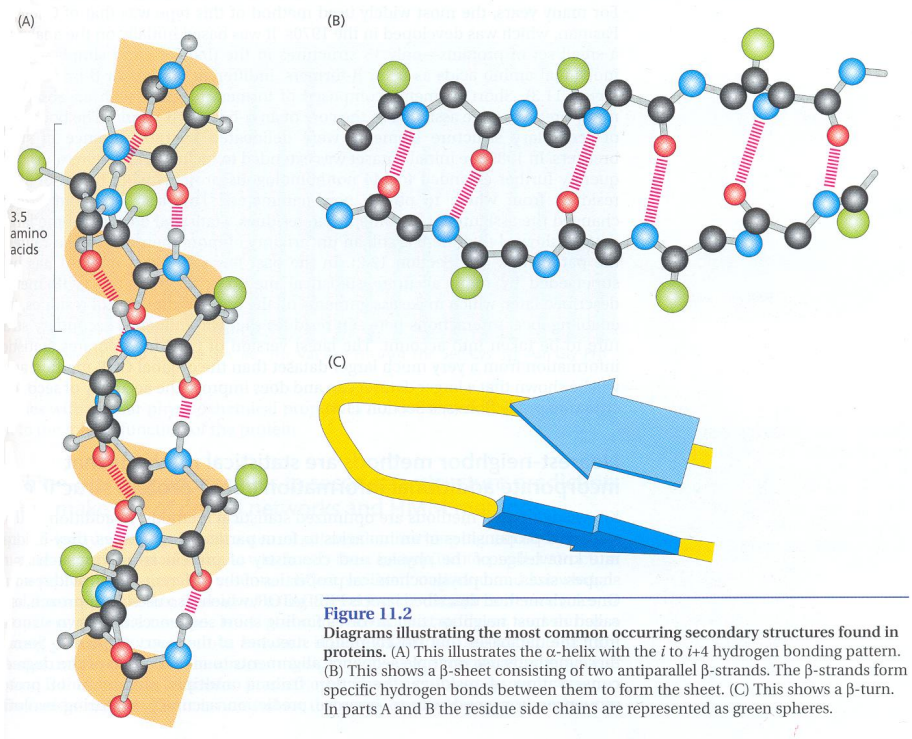
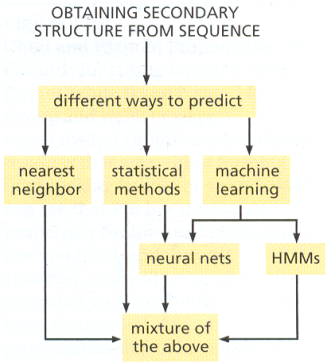


Figure 11.2
Diagrams illustrating the most common occurring secondary structures found in proteins. (A) This illustrates the α-helix with the *i* to *i*+4 hydrogen bonding pattern. (B) This shows a β-sheet consisting of two antiparallel β-strands. The β-strands form specific hydrogen bonds between them to form the sheet. (C) This shows a β-turn. In parts A and B the residue side chains are represented as green spheres.

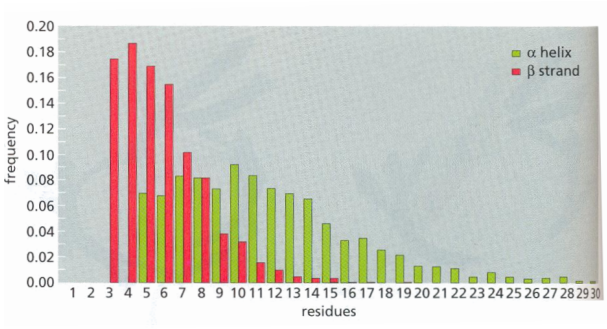
Zur Sekundärstrukturvorhersage wurden u.a. folgende Ansätze verwendet: Nearest-neighbor, statistische Methoden und Methoden aus dem Bereich des Machine Learning (Neuronale Netze, HMM). Bei der Anwendung dieser Methoden wird zwischen löslichen Proteinen und Transmembran-Proteinen unterschieden.

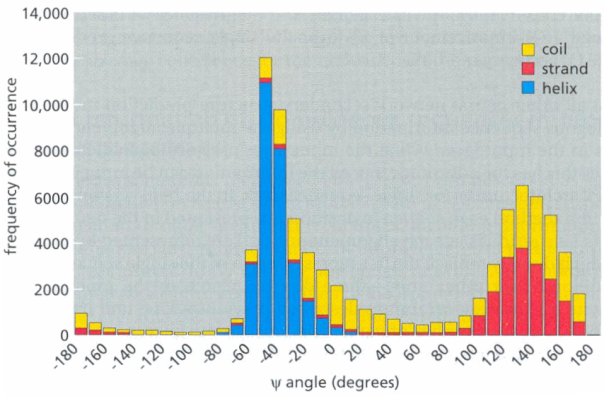


Flow Diagram 11.1
The key concept introduced in this section is that many different approaches have been taken in deriving methods for predicting protein secondary structure.

5.3.1 Lösliche Proteine

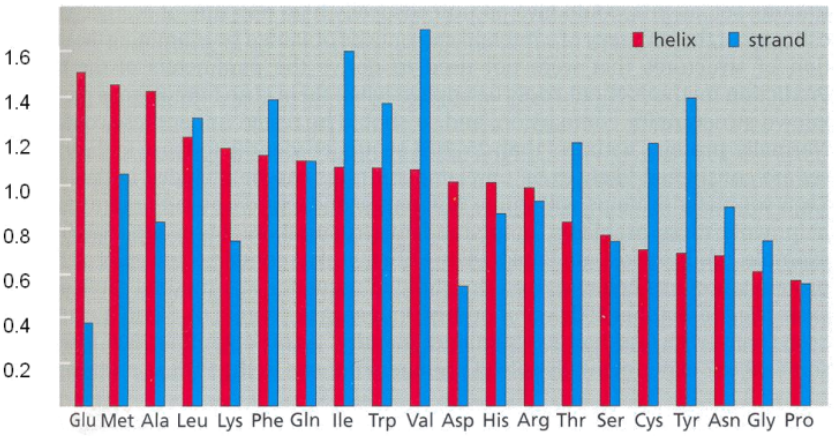
Die statistischen Daten stammten aus Strukturanalysen und Sequenzdaten einer großen Menge an Proteinen mit bekannter Struktur. Aus diesen Daten wurden statistische Regeln für die Sekundärstruktur abgeleitet.
Im linken Diagramm ist die Längenverteilung für α-Helices und β-Faltblätter in löslichen Proteinen gezeigt. Im rechten Diagramm ist die Verteilung des Φ-Winkels im Rückgrat des Aminosäurestrangs für die Klassen H, E und C gezeigt.





Chou & Fasman
Die 1970 von Chou und Fasman entwickelte statistische Methode wurde lange verwendet. Initialisiert wurde sie anhand der 3D Strukturen von lediglich 15 damals bekannten Proteinstrukturen. Die einzelnen Aminosäuren wurden in die Gruppen α-β-Former, α-β-Brecher und Neutrale aufgeteilt.

Amino Acid	helix			
	Designation	*P*	Designation	*P*
Ala	F	1.42	b	0.83
Cys	I	0.70	f	1.19
Asp	I	1.01	B	0.54
Glu	F	1.51	B	0.37
Phe	f	1.13	f	1.38
Gly	B	0.61	b	0.75
His	f	1.00	f	0.87
Ile	f	1.08	F	1.60
Lys	f	1.16	b	0.74
Leu	F	1.21	f	1.30
Met	F	1.45	f	1.05
Asn	b	0.67	b	0.89
Pro	**B**	**0.57**	**B**	**0.55**
Gln	f	*1.11*	*h*	*1.10*
Arg	I	0.98	I	0.93
Ser	I	0.77	b	0.75
Thr	I	0.83	f	1.19
Val	f	1.06	F	1.70
Trp	f	1.08	f	1.37
Tyr	b	0.69	F	1.4



Das rechte Diagramm zeigt die graphische Darstellung der Chou-Fasman Indices aus der Tabelle links. Dabei steht:

F : für starke Former

f : für schwache Former

B : starker (Unter-) Brecher

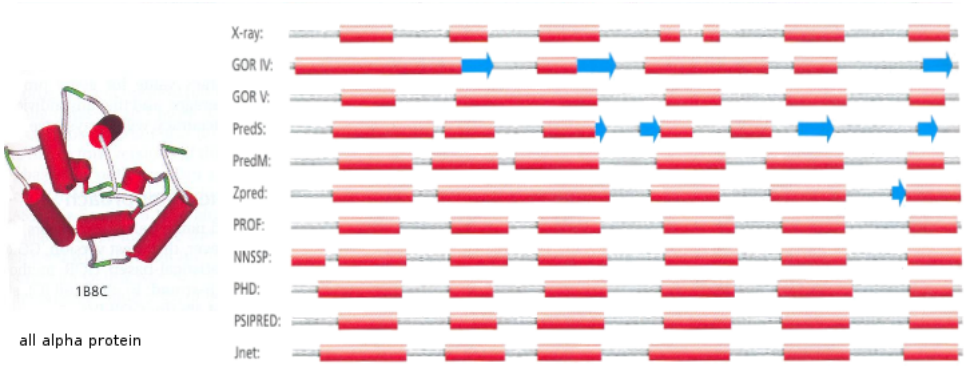
b : schwacher (Unter-) Brecher

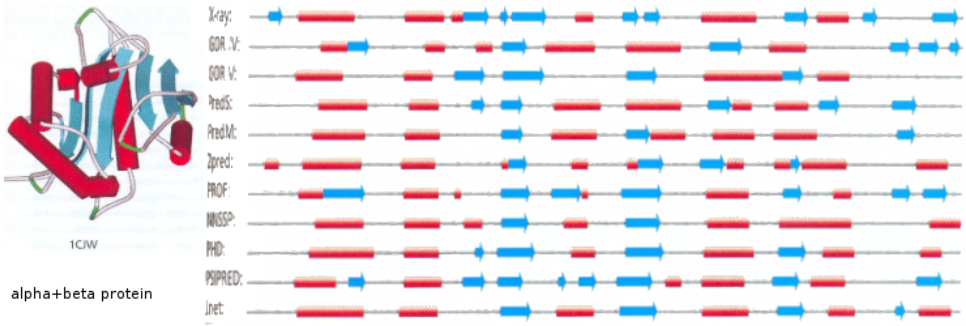
I : neutral (indifferent)

Anhand der niedrigen Werte für Prolin ist diese Aminosäure zugleich der stärkste Helixbrecher sowie der stärkste Brecher von Betasträngen. Dies liegt vor allem an der Rückfaltung seiner Seitenkette, die dadurch die möglichen Winkel ϕ und ψ stark einschränkt. Da es nach der Vergrößerung des Datensatzes für die statistischen Grundlagen zu Unstimmigkeiten von über 10% in den Parametern kam, werden nun genauere Methoden wie z.B. GOR angewandt.

GOR

Benannt ist die Methode nach ihren Autoren J. Garnier, J.-F. Gibrat, B. Robson (GOR Method for Predicting Protein Secondary Structure from Amino Acid Sequence, Methods in Enzymology 266, 540 (1996)).
Diese Methode unterteilt die Struktur in α-Helix (H), β-Faltblatt (E) und Coil (C). Sie arbeitet nicht nur mit statistischen Parametern, sondern betrachtet außerdem die Effekte der Umgebung. So betrachtet sie ein Fenster (bestimmter Bereich in der Sequenz) von 17 Aminosäuren, d.h. von der zu untersuchenden Aminosäure j jeweils 8 Aminosäuren in beide Richtungen (j-8 , j+8). Dabei werden drei Arten der Information benutzt. Die Selbst-Information, die Richtungs-Information und die Paar-Information. Die Selbst-Information ist die Information über die Konformation der Residue selbst, angelehnt an Chou und Fasman. Die Richtungs-Information betrachtet die Konformation der Umgebung i, mit $i \neq j$, ungeachtet der eigenen Konformation. Die Paar-Information zählt die Konformation, die an Stelle j vorliegt.
Die GOR-Methode wurde immer weiterentwickelt, die aktuelle Version GOR-V hat eine Vorhersagegenauigkeit von 73.5%.
Hier ein Vergleich für verschiedene Methoden:

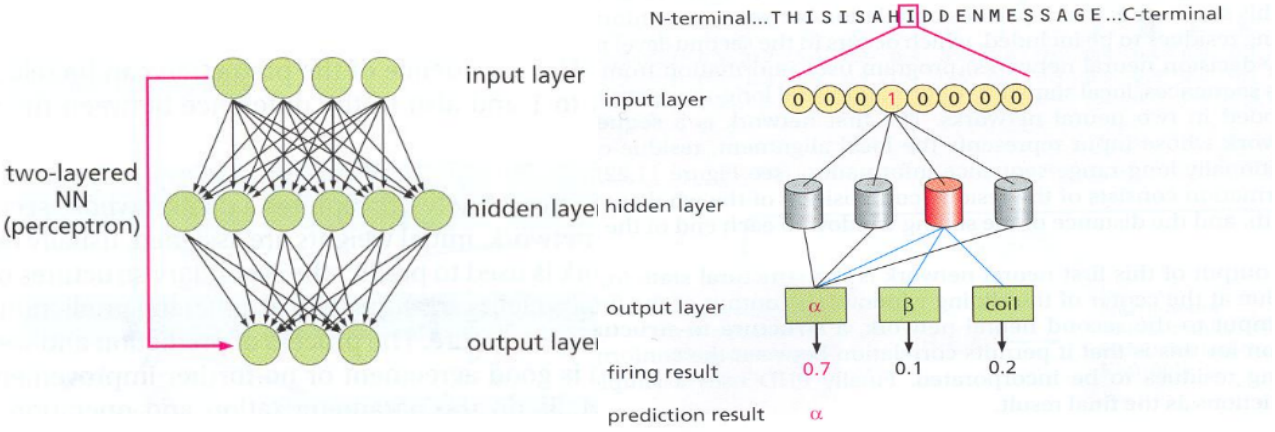




Neuronale Netze

Neuronale Netze sind ein Beispiel für maschinelle Lerntechniken, die gut dafür geeignet sind, redundante Informationen zu streichen. Neuronale Netze bearbeiten die Informationsabwicklung über sogenannte Layer. Das einfachste neuronale Netz ist ein Zwei-Layer Netzwerk (*perceptron*). Der erste Layer ist der *Input Layer*, der zweite ist der *Output Layer*. Die Layer können aus mehreren Knoten bestehen. Komplexere Neuronale Netze besitzen ein oder mehrere Zwischenlayer (*Hidden layer*).

Das Inputsignal für eine Aminosäure ist oft eine Gruppe aus 20 Einheiten. In den meisten Vorhersageprogrammen werden diese in ein gleitendes Fenster unterteilt. In der Mitte liegt die Aminosäure, deren Struktur vorhergesagt werden soll. So hat z.B. ein Netzwerk mit 13-Residuen-Fenstern einen Inputlayer von 260 (13*20) Knoten. Die Werte für die Inputknoten sind bei der repräsentierten Aminosäure 1, ansonsten 0. Im Output sind das die drei Zustände α-Helix (H), β-Faltblatt (E) oder Coil (C). So wäre ein Ergebnis (1,0,0) eine perfekte α-Helix Vorhersage. Sind die Werte anders gewichtet, wird die Struktur mit dem höchsten Wert angenommen.

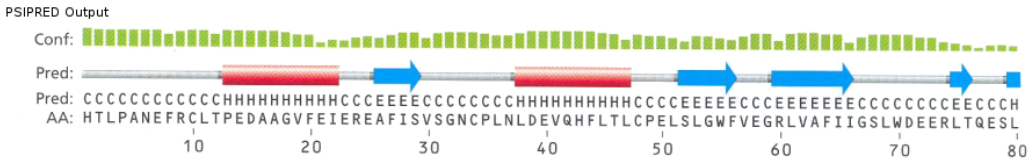


PSIPRED

PSIPRED ist eines der Programme, die Neuronale Netze verwenden. PSIPRED arbeitet in drei Stufen.

In Stufe eins generiert es ein multiples Alignment und ein PSI-BLAST (Vorlesung 2) Profil für die gegebene Sequenz. Im zweiten Schritt generiert es eine erste Sekundärstruktur, die im letzten Schritt gefiltert wird. Das Programm verwendet ein neuronales Netz aus zwei

Stufen. Das benutzte Fenster für den ersten Input sind 15 Residuen, der erste Output ist die Struktur für die zentrale Aminosäure des Fensters. Der Output des ersten neuronalen Netzes dient dann als Input für das zweite Netz.



5.3.2 Transmembran (TM-) Proteine

Hydrophobizitätsskalen liefern ein einfaches Kriterium um Transmembran Helices (TMH) vorherzusagen. TMHs können durch hydrophobe und polare Regionen in der Sequenz vorhergesagt werden (s. 5.2.1 helikales Rad).
Dabei werden immer wiederkehrende Motive beobachtet:

1. TMHs sind meistens apolar und 12-35 Residuen lang,
2. Globuläre (d.h. kompakte oder kugelförmige) Regionen zwischen den TMHs sind kürzer als 60 Residuen,
3. die meisten TMH Proteine haben eine spezifische Verteilung der positiv geladenen Aminosäuren Arginin und Lysin. Gemäß der empirisch gefundenen „positive-inside-rule“ (Gunnar von Heijne) haben die „loop“ Regionen innen mehr positiv geladene Aminosäuren als außen.
4. Lange globuläre Regionen (> 60 Residuen) unterscheiden sich in ihrer Anordnung von den Regionen, die der „ Innen-Außen-Regel“ unterliegen.

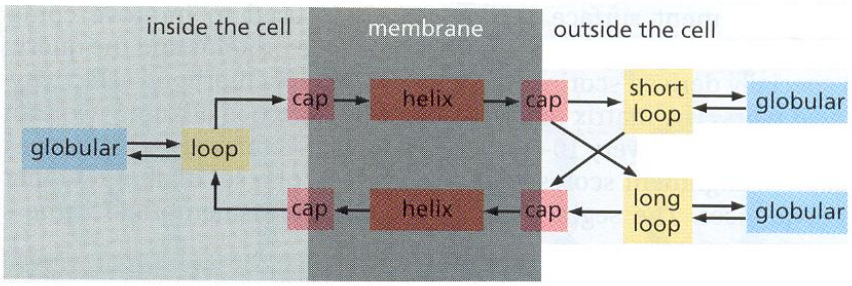
Kyte & Doolittle
Kyte und Doolittle stellten eine sehr verbreitete Hydrophobizitäts Skala auf:

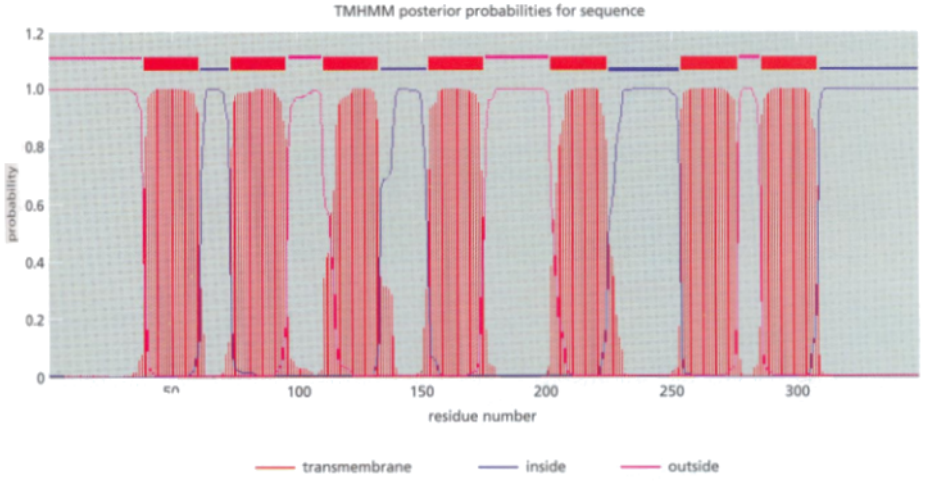
Alanin	1.8	Arginin	-4.5
Asparagin	-3.5	Asparaginsäure	-3.5
Cystein	2.5	Glutamin	-3.5
Glutaminsäure	-3.5	Glycin	-0.4
Histidin	-3.2	Isoleucin	4.5
Leucin	3.8	Lysin	-3.9
Methionin	1.9	Phenylalanin	2.8
Prolin	-1.6	Serin	-0.8
Threonin	-0.7	Tryptophan	-0.9
Tyrosin	-1.3	Valin	4.2

Nach dieser Skala ist Isoleucin die hydrophobste (4.5) und Arginin die am wenigsten hydrophobe Aminosäure. Diese Skala wird benutzt, um ein hydropathisches Profil der Sequenz zu erstellen. Hier wird wieder ein Fenster, diesmal der Länge 15 -30, benutzt. Kleinere Fenster benötigt man, um das Ende einer TMH zu suchen. Man addiert die Werte aller w-Residuen in dem Fenster der Länge w. Benutze eine Grenze T, wenn die Werte darüber liegen, wird eine Membranhelix vorhergesagt. Bei einem Wert von $w = 19$ erhält man gute Ergebnisse bei der Unterscheidung zwischen Membran und kugelförmigen Proteinen. Als Grenze wird dabei T > 1.6 vorgeschlagen.
Seit wenigen Jahren ist auch eine nennenswerte Anzahl von 3D-Strukturen für Transmembranproteine bekannt. Dies ermöglicht es, statistische Ansätze mit der Anzahl der beobachteten Aminosäuren in Membranproteinen im Vergleich zu deren Anzahl in Nicht-Membranproteinen, zu erstellen.
Programme hierfr sind TMpred und SPLIT. Letzteres ist in der Lage, kürzere, unstabile oder bewegliche Membran Helices identifizieren.

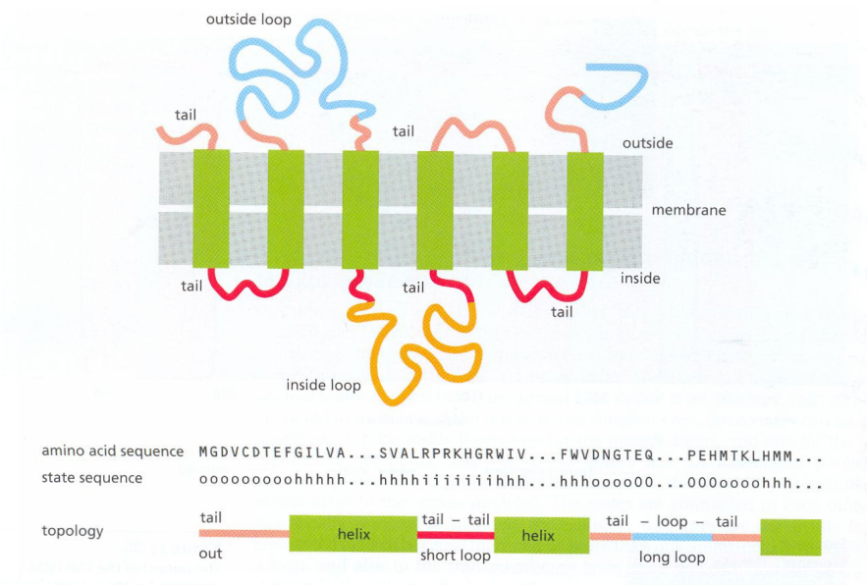
Die Struktur der Transmembranproteine wird durch den Lipid-Bilayer der Membran eingeschränkt. Zwei Programme (TMHMM und HMMTop) basierend auf HMMs versuchen dieses Problem zu lösen.

TMHMM
TMHMM (Sonnhammer et al. 1998, Krogh et al. 2001) verwendet ein zyklisches Modell des HMM mit 7 Zuständen:
den TMHkern, TMH N- und C-terminale Enden, Nicht-Membran Regionen auf der Cytoplasma Seite, 2 Nicht-Membran Regionen auf der nicht Cytoplasma Seite (kurze und lange Loops) und eine globuläre Domäne in der Mitte jeder Nicht-Membran Region.





HMMTOP
HMMTOP (Tusnady & Simon 1998, 2001) verwendet ein Hidden Markov Model mit 5 Zuständen:
Nicht-Membran Region innen, TMH-Ansatz innen, Membran Helix, TMH Ansatz außen und Nicht-Membran Region außen.

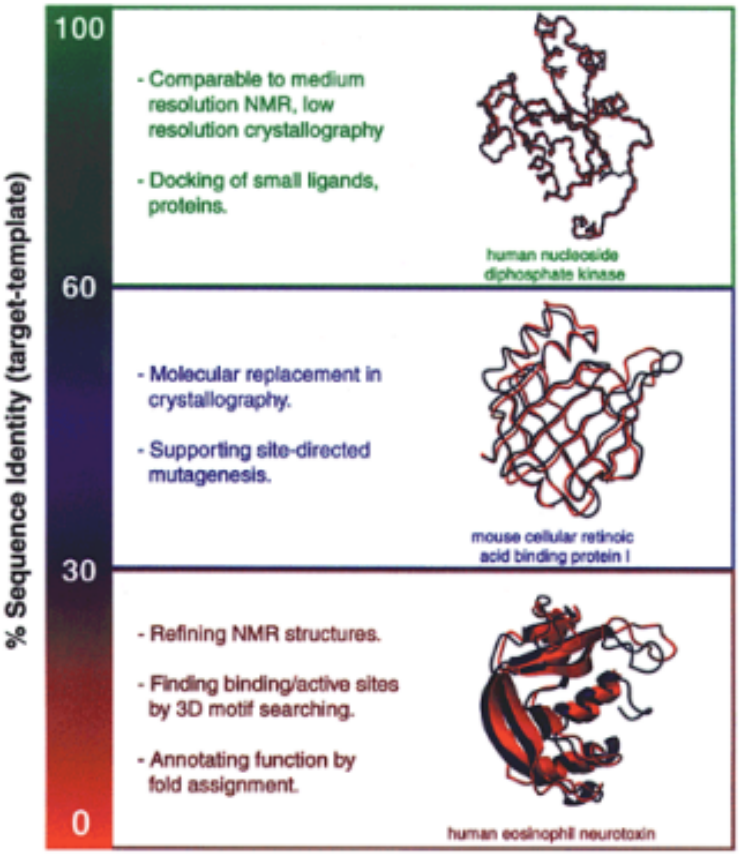


Zusammenfassung:

Die Strukturvorhersage von Sekundärstrukurelementen erlangt bei wasserlöslichen Proteinen eine Genauigkeit von ca. 70%.
TMHs sind Ketten mit meist hydrophoben Residuen.
Die besten Methoden basieren auf HMM und Neuronalen Netzen.
Es ist möglich, Signalpeptide und TM Helices zu unterscheiden.
Nur Split 4.0 kann kurze Nicht-Membran Helices aufspüren.
Die besten Vorhersagen erhält man durch Kombination der Ergebnisse von mehreren Programmen.

6 Homologie Modellierung

Homologie-basierte Proteinmodellierung ist ein wissensbasierter Ansatz. Für die Modellierung der dreidimensionalen, atomistischen Struktur eines Proteins muss daher wenigstens eine bekannte Struktur eines verwandten Proteins vorliegen.
Bei der Proteinmodellierung geht man folgendermaßen vor: Zuerst sucht man nach ähnlichen Sequenzen von Proteinen, deren Struktur bekannt ist. Dann wird ein Multiples Alignment mit der Zielsequenz und den gefundenen Sequenzen erstellt.
In mehreren Schritten wird nun die Struktur modelliert:
Zuerst wird ein Grundgerüst (*Framework*) für die neue Sequenz generiert. Danach müssen fehlende Loops hinzugefügt werden. Ist dies geschehen, kann das Proteinrückgrat vervollständigt und korrigiert werden. Anschließend werden die Aminosäureseitenketten korrigiert. Sind diese Schritte alle abgeschlossen, muss letztendlich die Qualität der Struktur überprüft werden. Durch Energieminimierung und Moleküldynamik kann die Struktur noch verbessert werden.

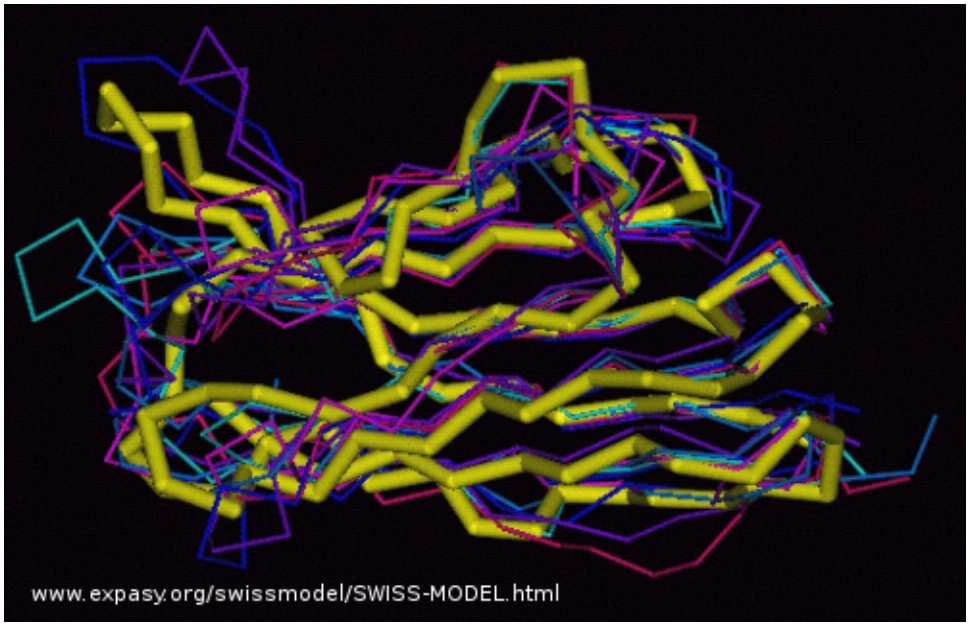


Protein structure modeling for structural genomics.
R. Sánchez *et al.* Nat. Struct. Biol. 7, 986 - 990 (2000)

Die Zuverlässigkeit eines Homologiemodells hängt in erster Linie davon ab, ob es eine vorhandene Kristallstruktur eines hinreichend ähnlichen Proteins gibt. Außerdem muß das Alignment der Aminosäuresequenzen korrekt sein. Falsche Alignments führen zwangsläufig zu falschen Strukturen.

6.1 Erstellung des Frameworks

Für alle Atome, die aufgrund des Sequenzalignments eine ähnliche Position besitzen und damit vermutlich ebenso eine strukturelle Entsprechung in der neuen Struktur besitzen, werden gemittelte Positionen als Framework-Koordinaten bestimmt. Diejenigen Seitenketten, die eine völlig inkorrekte Geometrie aufweisen, werden entfernt.
Das Bild zeigt, wie anhand mehrerer Strukturvorlagen das gelb und fett dargestellte Strukturmodell für die Zielsequenz konstruiert wurde.



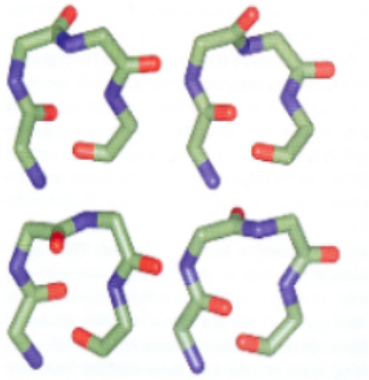
6.2 Konstruktion fehlender Loops

Bei der vergleichenden (komparativen) Modellierung ist die Konstruktion von, in der Struktur fehlenden, Loops ein großes Problem. Eine generell gültige, effiziente Lösung für dieses Problem ist noch nicht gefunden. Dies gilt sowohl für lange Loops, in denen zahlreiche Mutationen auftreten, als auch für kurze Loops, in denen Insertionen und Deletionen auftreten.
Zwei Ansätze sind zum einen die wissensbasierte Analogie zu bekannten Proteinstrukturen, die Loopfragmente sehr ähnlicher Sequenz enthalten und zum anderen die *de novo* Modellierung durch Konformationssuche aufgrund einer energetischen Bewertung. (s. unten) Sobald das Alignment aus der Zielsequenz und der Vorlagesequenz vorliegt, sollte man überprüfen, ob die eingefügten Gaps außerhalb von Sekundärstruktur-Elementen in der 3D-Struktur der Vorlage liegen.

Als Anhaltspunkte sollen folgende Regeln dienen:

- Bei sehr kurzen Loops können Daten über *β-turns* verwendet werden. Eine Aminosäurekette kann ihre Richtung nämlich dadurch umkehren, dass ein *reverse turn* durch Bildung einer H-Bindung zwischen C=O und H-N gebildet wird. Wenn dies zwischen zwei antiparallelen β-Strängen geschieht, nennt man den Turn eine β-Haarnadel (*hairpin*). Es ergeben sich folgende Diederwinkel:

TURN Typ	ϕ_1	ψ_1	ϕ_2	ψ_2
I	-60	-30	-90	0
I‘	60	30	90	0
II	-60	120	80	0
II‘	60	-120	-80	0



Buch, Anna Tramontano

Die Turns rechts entsprechen den in der Tabelle beschriebenen Turn-Typen.

- Falls mittellange Loops kompakte Substrukturen bilden, spielt die Ausbildung von Wasserstoffbrückenbindungen mit den Atomen des Rückgrats die wichtigste Rolle für die Konformation.

- Falls mittellange Loops ausgedehnte Konformationen haben, ist für ihre Stabilisierung meistens eine hydrophobe Seitenkette verantwortlich. Diese zeigt dann ins Proteininnere und ist zwischen den Sekundärstrukturelementen gepackt. Das Bild zeigt zwei Schleifen ähnlicher Konformation, die durch die ins Proteininnere weisende Faltung eines hydrophoben Phenylalaninrings bzw. Isoleucins stabilisiert sind.

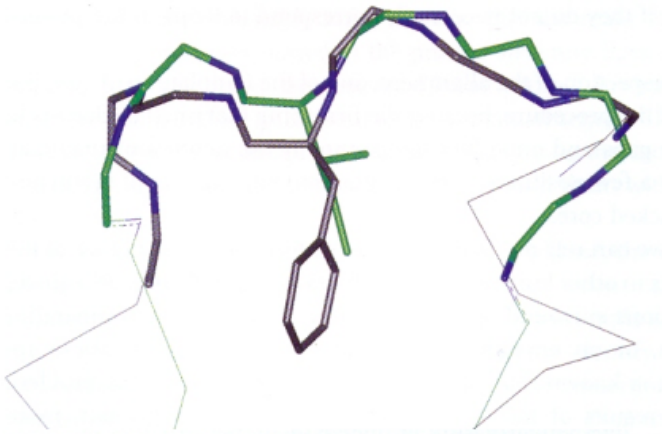
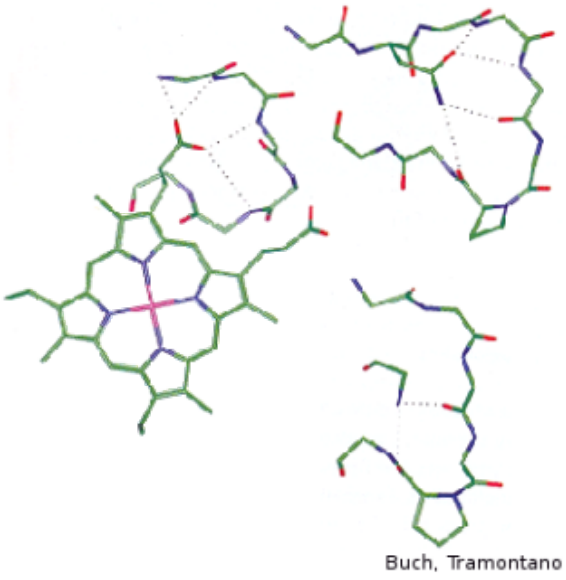


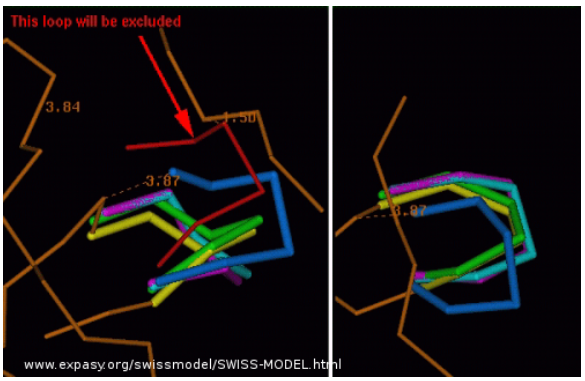
Figure 4.16 The figure shows two loops with similar conformations stabilized by the packing of a central hydrophobic amino acid. Note that one of the loops connects two alpha helices and the other two beta strands.

Buch, Tramontano

Das nächste Bild zeigt sehr ähnliche Konformationen dreier Loops mit unterschiedlicher Sequenz. Die beiden rechten Loops enthalten ein cis-Prolin. Die stabilisierenden H-Bindungen werden mit sehr unterschiedlichen Proteingruppen ausgebildet. Die Schleifen sind dennoch sehr ähnlich und sind über Wasserstoffbrückenbindungen miteinander verbunden, wobei die Interaktionspartner in drei verschiedenen Proteinen (Immunoglobulin, Virales Protein und ein Cytochrom) unterschiedlich sind.

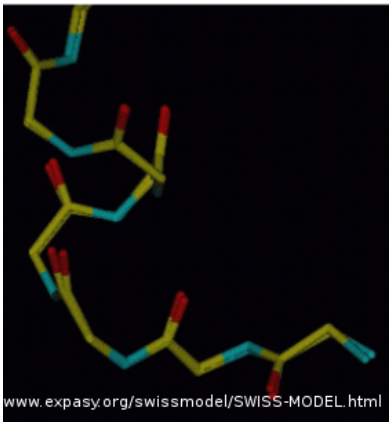


- Basierend auf den Verankerungen zu Beginn und am Ende wird entweder:
  a) eine Datenbank bekannter Loopfragmente aus der PDB durchsucht.
  Für den neuen Loop verwendet man dann entweder das am besten passende Fragment oder ein Framework aus den 5 besten Fragmenten.
  oder
  b) der Torsionsraum der Loopresiduen durchsucht und energetisch bewertet.
  Es gibt 7 erlaubte Kombinationen der ϕ-ψ Winkel. Man betrachtet den gesamten Raum, der für den gesamten Loop benötigt wird.



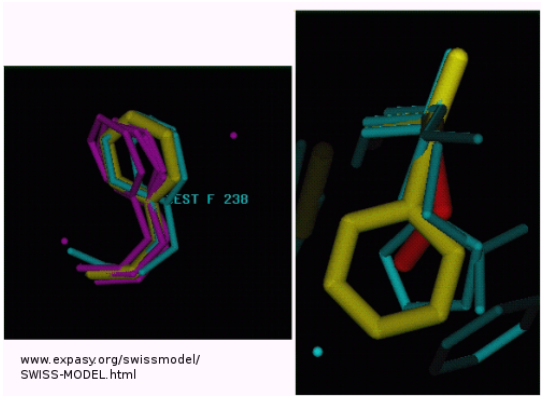
6.3 Rekonstruktion von fehlendem Proteinrückgrat

Das Rückgrat soll auf der Grundlage der C_α-Positionen ergänzt und rekonstruiert werden, wobei wieder 7 Kombinationen der ϕ-ψ Winkel erlaubt sind. Zunächst wird eine Datenbank für Backbone-Fragmente aus der PDB-Datenbank (Vorlesung 1) mit einem Fenster aus 5 Residuen durchsucht. Man verwendet nun die Koordinaten der 3 zentralen Residuen des am besten passenden Fragments für die Rekonstruktion des fehlenden Rückgrats.



6.4 Konstruktion unvollständiger oder fehlender Seitenketten

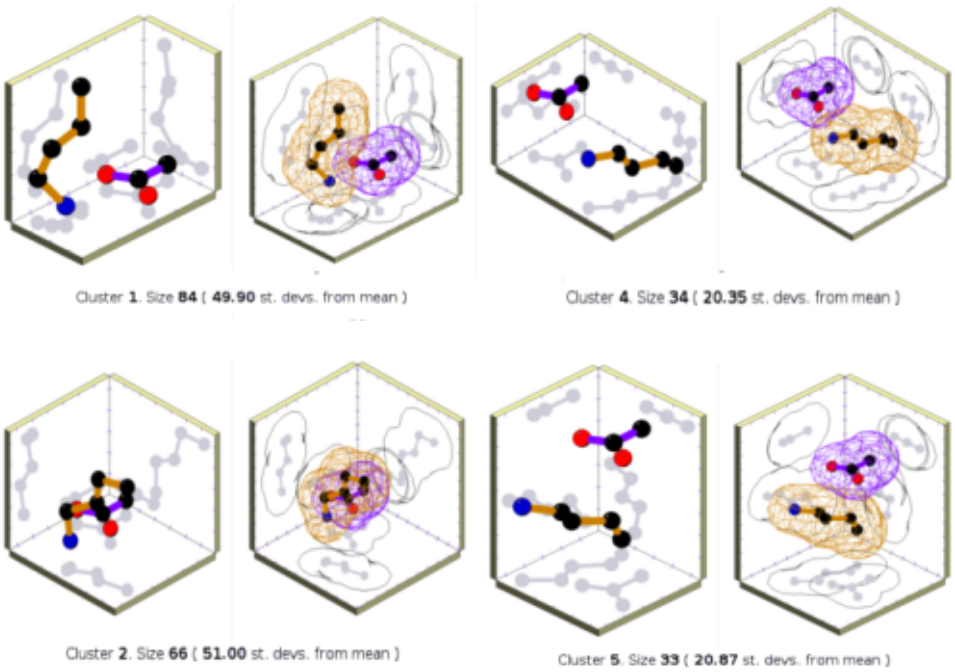
Ponder & Richards haben 1987 gezeigt, dass die meisten Aminosäuren bestimmte Winkelbereiche für ihre Seitenkettenwinkel bevorzugen. Aus dieser Erkenntnis sind **Rotamerbibliotheken** entstanden. Als Rotamer bezeichnet man ein häufig vorkommende Seitenkettenkonformation. Um nun die unvollständigen Seitenketten zu konstruieren, wird die Bibliothek erlaubter Seitenketten-Rotamere verwendet, die nach der Häufigkeit des Auftretens in der PDB-Datenbank geordnet ist.
Zuerst werden verdrehte (aber komplette) Seitenketten korrigiert. Die fehlenden Seitenketten werden mit Hilfe der Rotamer-Bibliothek ergänzt. Dabei muss getestet werden, ob Überlappungen der van-der-Waals Radien um die Atome mit Atomen der Umgebung auftreten und ob die Torsionswinkel in den erlaubten Bereichen liegen.

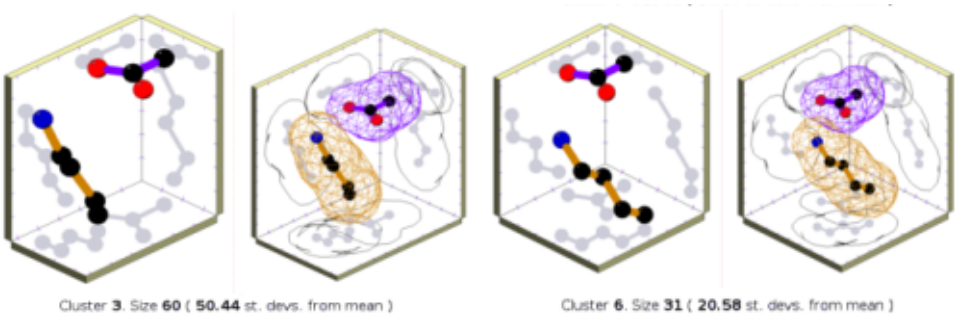


6.5 Paarungs-Präferenz von Aminosäuren

Bei der Ergänzung der Seitenkettenorientierungen wird jede Seitenkette üblicherweise für sich allein betrachtet. In der Rotamer-Bibliothek wird zwar die Konformation des kompletten Rückgrates berücksichtigt, nicht aber die Konformationen der Umgebung. Je nach Umgebung nehmen die Aminosäuren jedoch unterschiedliche Konformationen an. Diese Effekte werden auch „Packungseffekte“ genannt. In der Zukunft könnten diese genaueren, aber deutlich komplizierteren Statistiken für die komparative Modellierung berücksichtigt werden.
Für die Packungseffekte sind verschiedene Faktoren verantwortlich. Wir werden im Folgenden einige Ergebnisse aus der Datenbank über die statistische Präferenz für die Orientierung von Aminosäure-Seitenketten in der PDB-Datenbank:
http://www.biochem.ucl.ac.uk/bsm/sidechains/index.html diskutieren. Unter diesem Link findet man ebenfalls die Informationen über die Interaktionen der Seitenketten.

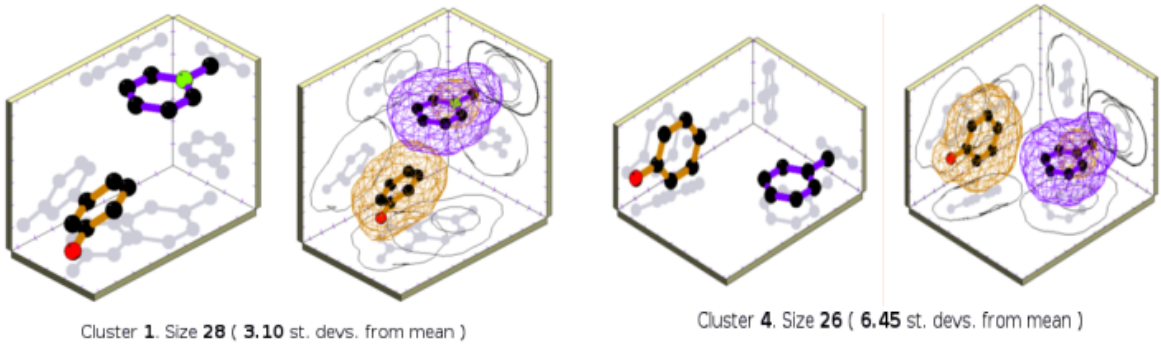
Salzbrücken
Zum Beispiel gibt es für die beiden Aminosäuren Asparaginsäure und Lysin insgesamt 1845 Kontakte in der Datenbank. Die Kontaktarten wurden in sechs signifikante Cluster aufgeteilt. Jedes Cluster ist im unteren Bild auf zwei Arten dargestellt. Links sind die Strukturen in *Ball and Stick* eingezeichnet und rechts sind die Strukturen mit der dazugehörigen van-der-Waals Umgebung gezeigt.





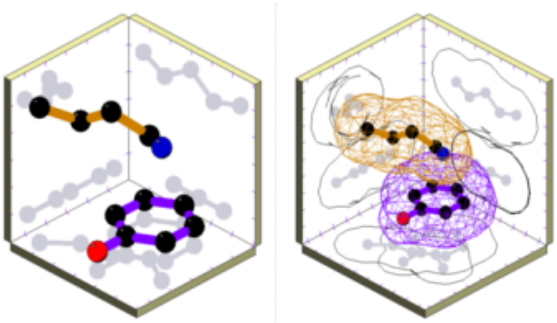
Π-stacking von Aromatischen Ringen

Die aromatischen Ringe in den Seitenketten von Tyrosin, Phenylalanin, Tryptophan und Histidin besitzen ein delokalisiertes Elektronensystem außerhalb der Ringebene. Mehrere dieser Ringe „packen“ bevorzugt aufeinander bzw. senkrecht zueinander. So gibt es z.B zwischen Phenylalanin und Tyrosin 1851 Kontakte in 6 signifikanten Clustern. Hier sind nur die Cluster 1 und 4 gezeigt.



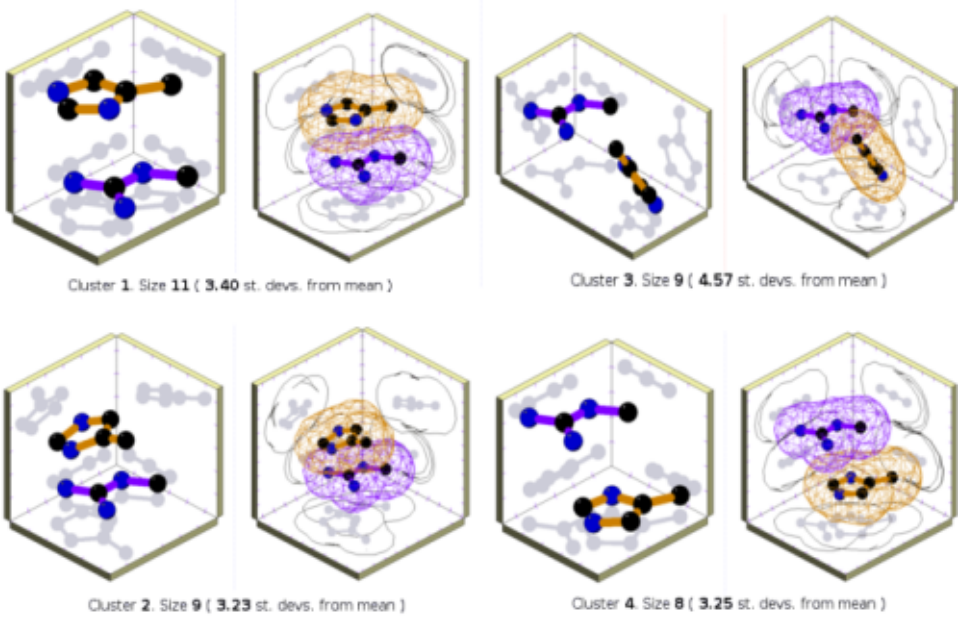
Kationen-Π-Wechselwirkung

Aromatischen Ringe können außerdem mittels der delokalisierten Elektronensysteme senkrecht zur Ringebene mit positiv geladenen Gruppen interagieren. Die Wechselwirkung ist jedoch schwächer als zwischen zwei geladenen Residuen. Ein Beispiel hierfür ist das aktive Zentrum des Enzyms Acetylcholinesterase, in dem der Ligand Acetylcholin bindet. In dieser Bindungstasche treffen Tyrosin und Lysin aufeinander. Zwischen Tyrosin und Lysin wurden insgesamt 1138 Kontakte analysiert, die wiederum in sechs signifikanten Clustern zusammengefasst werden können. Hier ist Cluster 6 gezeigt:



Cluster **6**. Size **18** (**22.00** st. devs. from mean)

Ein weiteres Beispiel ist die Wechselwirkung der positiv geladenen Guanidinium-Gruppe von Arginin mit dem Π-Elektronensystem von Histidin. Es ist fast immer eine planare Packung. Nur in Cluster 3 gibt es eine Ausbildung einer Wasserstoffbrücke zwischen N-H und N. Es liegen 547 Kontakte in 4 signifikanten Clustern vor.



Cluster **1**. Size **11** (**3.40** st. devs. from mean)

Cluster **3**. Size **9** (**4.57** st. devs. from mean)

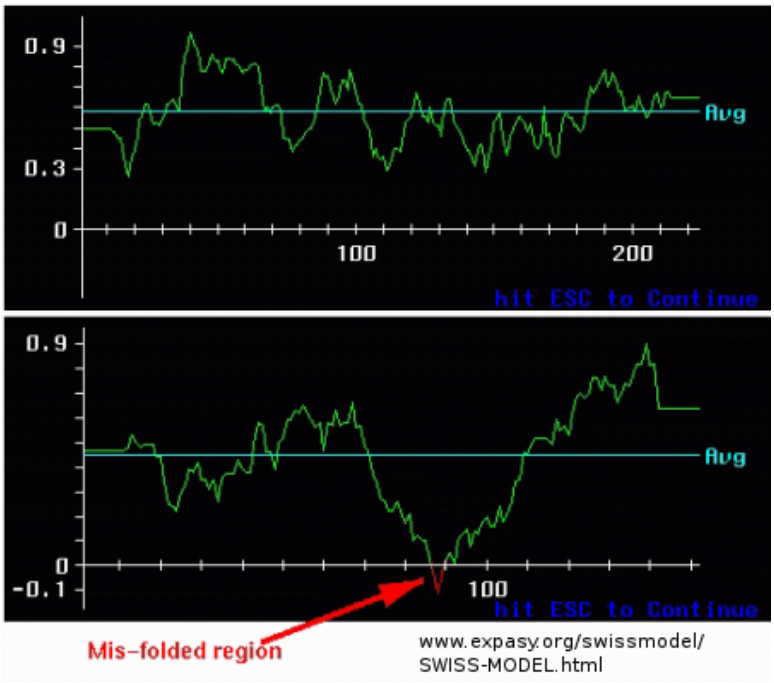
Cluster **2**. Size **9** (**3.23** st. devs. from mean)

Cluster **4**. Size **8** (**3.25** st. devs. from mean)

6.6 Qualität der Modellierung

Um die Qualität der 3D-Modellierung einer homologen Struktur zu bestimmen, wird die 3D-Umgebung jeder Seitenkette analysiert. Dies erlaubt im Allgemeinen die Identifizierung fehlgefalteter Regionen. Ein Programm, das diese Überprüfung vornimmt ist z.B. WHATCHECK, welches u.a. auch die Packungsdichte überprüft.
Dabei wird berechnet, welche Bereiche des Proteins für eine kleine Probe zugänglich sind (Connolly-Oberfläche bzw. Kubisches Gitter). Der Algorithmus entdeckt die Oberflächen

innerhalb und außerhalb des Proteins. Der Vergleich von Größe und Verteilung von internen Cavities zwischen Modell und Kristallstruktur-Vorlage erlaubt es, möglicherweise fehlerhafte Regionen im Modell aufzuspüren. In diesem Beispiel ist im oberen Bild eine Röntgenkristallstruktur gezeigt, in der alle Sequenzbereiche eine gleichmäßige Bewertung erhalten.



Das Strukturmodell im unteren Bild weist anscheinend bei Position 90 eine schlecht gefaltete Region auf, die durch den Vergleich mit der typischen Packung dieser Aminosäuren in richtigen Strukturen eine ungünstige Bewertung erhält.

6.6.1 Bewertung der Qualität eines Homologiemodells

Expasy (Expert Protein Analysis System) unterteilt die Bewertung eines Modells in 4 Stufen:
(*www.expasy.org/swissmodel/SWISS-MODEL.html*)

1. Allgemeine Gesichtspunkte:

- Ein Modell wird als falsch angesehen, wenn mindestens eines seiner strukturellen Elemente gegenüber dem Rest des Modells falsch angeordnet ist. Dies kann durch ein falsches Sequenzalignment entstehen. Das Modell kann dennoch eine korrekte Molekülgeometrie der einzelnen Aminosäuren besitzen.

- Man kann ein Modell als ungenau ansehen, wenn seine atomare Koordinaten mehr als 4.5 Å von einer experimentellen Kontrollstruktur abweichen.

- Ungenauigkeiten können auch in der Molekülgeometrie (Bindungslängen und winkel auftreten). Dies kann z.B. ebenfalls leicht mit den Programmen WhatCheck und ProCheck überprüft werden.

- Statistische Paarpotentiale für die Verteilung von Aminosäuren in bekannten Proteinen erlauben manchmal die Aufspürung von fehlerhaften Modellen.

2. Fehlerquellen:

Die Qualität eines Modells wird anhand von 2 Kriterien beurteilt:

- Seine Korrektheit hängt vor allem von der Qualität des Sequenzalignments ab.

- Seine Genauigkeit wird durch seine Abweichung von einer (zukünftig zu bestimmenden) experimentellen Struktur bestimmt. Strukturelle Abweichungen haben 2 Ursachen:

  1) der inherente Fehler der Modellierungsprozedur

  2) durch Umgebung und Methoden der Datenerfassung bewirkte Variationen der experimentellen Strukturen, die als Vorlage verwendet werden.

Generell kann ein durch komparative Methoden abgeleitetes Protein-Modell nicht genauer sein, als z.B. der Unterschied zwischen einer NMR-Struktur und einer Kristallstruktur desselben Proteins, der oft im Bereich von 1 Å liegt.

3. Proteinkern und Loops:

- Fast jedes Proteinmodell enthält nicht-konservierte Loops, die als die am wenigsten zuverlässigen Teile des Proteinmodells angesehen werden können. Andererseits sind diese Bereiche der Struktur ohnehin am flexibelsten. Dies zeigt sich an hohen Temperaturfaktoren in Kristallstrukturen oder hohen Unterschieden zwischen verschiedenen (gleichsam gültigen) NMR-Strukturen.

- Die Residuen im Proteinkern werden gewöhnlich fast in der identischen Orientierung wie in experimentellen Kontrollstrukturen modelliert.

- Residuen an der Proteinoberfläche zeigen dagegen größere Abweichungen.

4. Einordnung von Proteinmodellen in 3 Kategorien:

1 Modelle, die auf falschen Alignments zwischen Vorlage und Zielprotein basieren. Als Strategie zur Vermeidung dieses Problems konstruiert man am besten mehrere Modelle für unterschiedliche Alignments und wählt das am besten erscheinende Modell aus.

2 Modelle, die auf korrekten Alignments beruhen, können für zielgerichtete Mutagenese-Experimente hilfreich sein. Sie sind aber oft nicht zuverlässig genug für die detaillierte Untersuchung von Ligandenbindungen.

3 Modelle, die auf einer hohen Sequenzidentität ($\geq$ 70%) mit der Vorlage beruhen, können in Drug Design Projekten verwendet werden. Fehler sind jedoch immer, also auch bei sehr hoher Identität möglich.

6.7 Zusammenfassung

- Der gemeinsame Kern von Proteinen mit 50% Sequenzidentität besitzt ca. 1 Å RMSD

- Dies gilt sogar für absolute identische Sequenzen.

- Der zuverlässigste Teil eines Proteinmodells ist der Sequenzabschnitt, den es mit der Vorlage gemeinsam hat. Die größten Abweichungen sind in den konstruierten Loops zu vermuten.

- Die Wahl der Modellvorlage ist entscheidend!
  Die An- oder Abwesenheit von Kofaktoren, anderen Untereinheiten oder Substraten kann die Proteinkonformation sehr beeinflussen und somit alle Modelle, die von ihnen abgeleitet werden.

- Jeder Fehler im Alignment produziert falsche Modelle!
  Solche Alignment-Fehler treten vor allem bei Sequenzidentität unter 40% auf.

7 Genexpression - Analyse von Mikroarrays

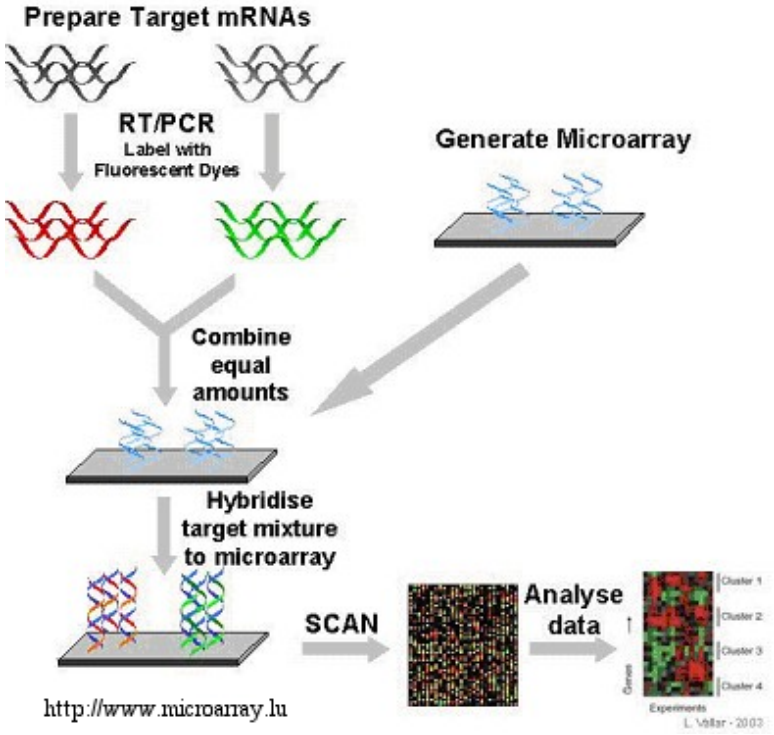
Die Genexpression beginnt zu dem Zeitpunkt in der Zelle, wenn die Genkodierende DNA in die messenger RNA (mRNA) transkribiert wird. Die gesamte Genexpression in kultivierten Zellen oder einem Stück Gewebe kann auf zwei Hauptarten erfasst werden. Zum Einen ist dies die Erfassung und Quantifikation der Menge an gesamter mRNA, dem Transkriptom, durch die DNA Mikroarray Technologie. Zum Anderen ist das die Erfassung aller Proteine, des Proteoms, durch das Auftrennen der Proteinprodukte durch 2D Gel-Elektrophorese oder Chromatographie. In beiden Fällen werden schon in einem Experiment eine große Menge Rohdaten bestimmt, welche analysiert werden müssen.
Das Transkriptom und das Proteom, aber nicht die DNA-Sequenz, ändern sich mit den unterschiedlichen Bedingungen, wie z.B. dem Stand der Entwicklung, der Umgebung oder dem Typ des Gewebes. Um z.B. ein möglichst genaues Bild einer Zelle zu erhalten (welches Gen wann aktiv ist), werden daher aus der Zellkultur Proben zu unterschiedlichen Zeitpunkten genommen.
Wir betrachten in dieser Vorlesung die Messung der Genexpression mittels Microarrays. Hier sind zwei typische Affymetrix GeneChip® Mikroarrays gezeigt.

In der Forschung haben sich DNA Mikroarrays für die Gewinnung von Hochdurchsatzdaten ganzer Genome etabliert, da dadurch die Expression einer großen Anzahl von verschiedenen Genen simultan gemessen werden kann.
Es gibt zwei Hauptansätze für den Einsatz von Microarrays, einmal die Untersuchung einer Probe für sich allein und zum anderen die Untersuchung zweier Proben miteinander. Z.B. könnten die Ähnlichkeiten oder die Unterschiede der Genexpression zweier Gewebetypen analysiert werden. Eine Anwendung findet dies z.B. in der Tumorforschung durch Vergleich von gesundem mit krankem Gewebe, um herauszufinden welche Gene im kranken Gewebe (stärker) exprimiert werden.
Ein Mikroarray Experiment kann man in drei Schritte unterteilen.

1. Zuerst muss das Mikroarray vorbereitet werden bzw. ein solches gekauft werden.

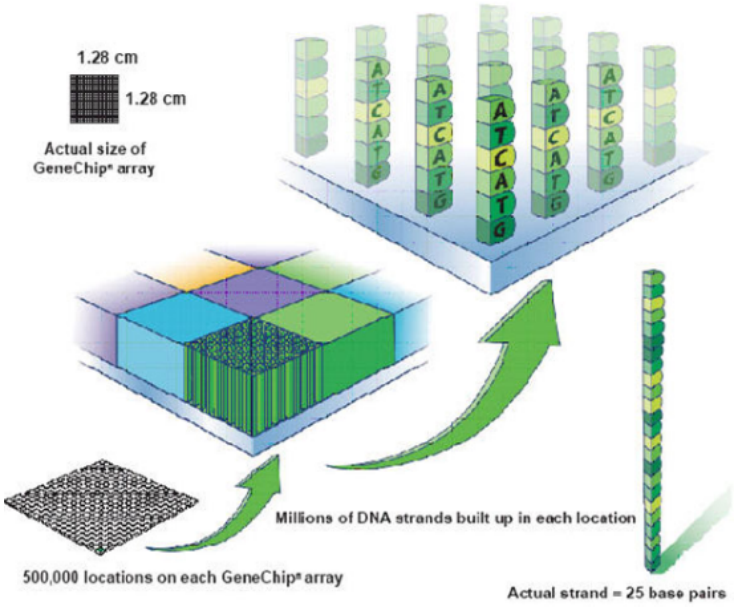
2. Dann erfolgt die Vorbereitung für den Scan. Dazu muss die mRNA der zu untersuchenden Zelle isoliert und in cDNA umgeschrieben werden, mittels fluoreszierender Marker markiert und auf das Mikroarray gegeben werden, wodurch es schließlich zur Hybridisierung der vorbereiteten cDNA mit dem Microarray kommt.

3. Letztendlich gilt es, das erhaltene Ergebnis, sprich das markierte Mikroarray zu analysieren. Das folgende Bild zeigt eine Übersicht über die 3 Arbeitsschritte:



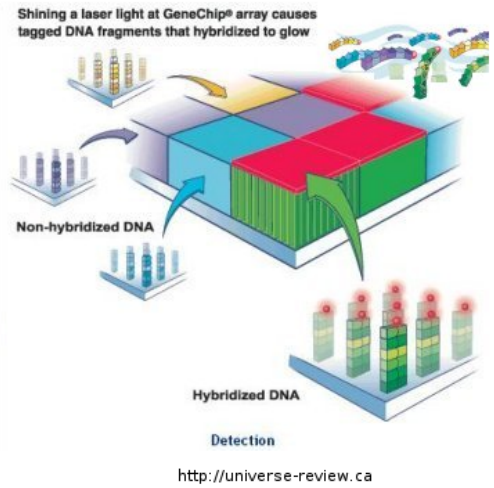
7.1 Vorbereiten des Mikroarray

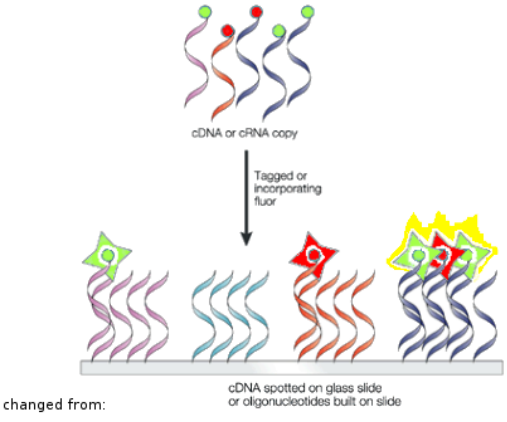
Ein Mikroarray besteht oft aus einem Glasplättchen oder einer anderen festen Unterlage. Diese wird in einem längeren Prozess mit einzelnen kurzen DNA Strängen (ca. 25 bp), entweder Base für Base oder direkt in einem Vorgang, beschichtet, d.h. die Nukleotidsequenzen werden auf der Unterlage immobilisiert. Dabei ist das Plättchen in 10 bis mehrere hunderttausend kleine Felder unterteilt, je nach Bedarf und Art der Anwendung. In den einzelnen Feldern sind allerdings immer viele Kopien derselben DNA-Sequenz aufgebracht, ein Feld entspricht also einem Gen bzw. einer DNA-Sequenz. Diese Sequenzstücke werden systematisch aufgebaut und beschreiben im Fall der Genexpression die zu der cDNA komplementäre DNA eines Gens.



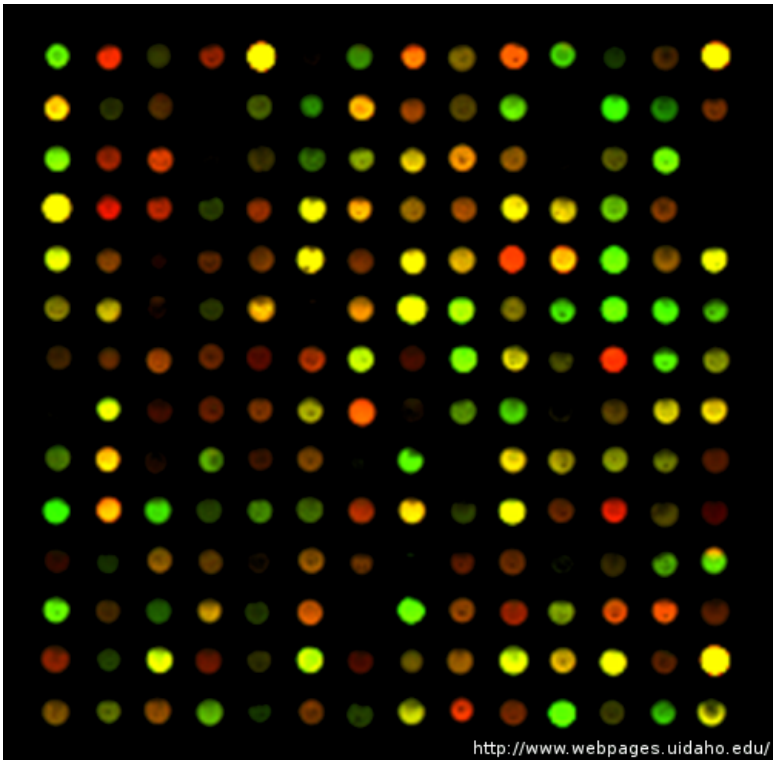
7.2 Versuch durchführen

Ist das Mikroarray vorbereitet, wird die zu analysierende RNA benötigt. Diese wird aus den Zellen gewonnen, eventuell aufgereinigt und dann mittels Prozessierung mit dem Enzym Reverse Transkriptase in cDNA (oder cRNA) umgewandelt. Für ein Experiment verwendet man meistens zwei Proben. Diese können entweder aus einer Zielsequenz und einer Testsequenz, zwei unterschiedlichen Proben, oder einer Probe zu unterschiedlichen Zeitpunkten bestehen. An die cDNA der beiden Proben werden zwei Fluoreszenzmarkermoleküle chemisch angehängt, die eine unterschiedliche Fluoreszenzwellenlänge besitzen. (Bsp. Cy3, grün und Cy5,rot). Anschließend wird das DNA Array mit der cDNA hybridisiert. Die in den Proben enthaltenen cDNA Stücke heften sich an ihre komplementären DNA Stücke auf dem Array.





D.h. die Felder mit den komplementären DNA Stücken binden die cDNA der aktuell exprimierten Gene. Da die Gene in den einzelnen Feldern bekannt sind, kann man durch die Markierung später feststellen, welche Gene in welcher Probe exprimiert sind. Nun müssen nur noch die übrigen nichtgebundenen cDNA Stücke von dem Chip gewaschen werden, bevor das Mikroarray für die Analyse bereit ist.
Der Scan wird mit Hilfe eines Lasersystems, das mehrere Wellenlängen anregen kann, ausgeführt. So wird das Mikroarray zweimal getrennt betrachtet. Einmal leuchten die mit Cy3 markierten Felder grün auf, beim anderen Mal die mit Cy5 markierten Felder rot. Mittels eines Computerprogramms werden die beiden so entstandenen Bilder übereinander gelegt. Ist zum Beispiel die Probe 1 rot markiert und die Probe 2 grün, kann man anhand des aufgenommenen Bildes erkennen, welche Gene in den beiden Proben exprimiert sind. Leuchtet ein Feld rot, ist dieses Gen hauptsächlich in Probe 1 exprimiert, leuchtet es grün in Probe 2. Leuchtet das Feld allerdings gelb, geschieht dies aufgrund der Überlagerung von rot und grün und bedeutet, dass dieses Gen in beiden Proben exprimiert ist. Die Felder, die nicht leuchten, sind auch nicht markiert und stehen für Gene, die in den Proben nicht oder nur sehr schwach vorhanden sind. Die unterschiedlichen Leuchtintensitäten der einzelnen Felder hängen mit der Menge der hybridisierten cDNA zusammen. Die folgende Abbildung zeigt ein typisches Bild eines gescannten Zweifarben-DNA-Mikroarrays.

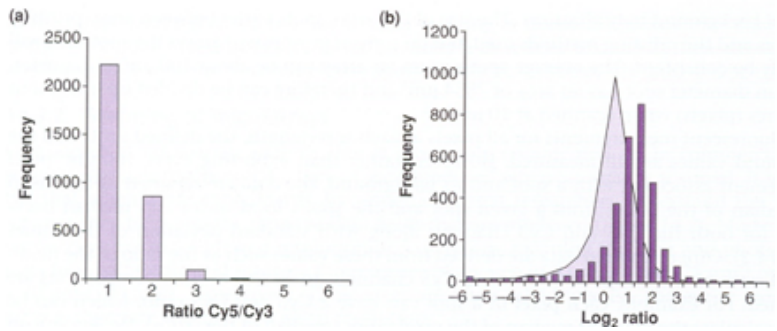


7.3 Analyse der Genexpression

Die einfachste Methode, die Microarray Daten der Genexpression zu analysieren, ist die Daten hierarchisch zu clustern. Das Hauptziel der Genexpressionsanalyse ist die Identifikation der Gemeinsamkeiten, z.B. welche Gene co-exprimiert sind und welche in einer Probe niedriger bzw. höher reguliert als in der anderen Probe sind. Das **hierarchische Clustern** ist die am weitesten verbreitete Methode, die Daten zu analysieren.

Um hierarchisches Clustern anwenden zu können, müssen die Daten zuerst prozessiert werden, was sich in folgende Schritte unterteilen lässt:

1. Das Mikroarray wird mittels eines Scanners eingelesen, um dann die Signale der kreisförmigen Zellen mit einem Programm zu identifizieren
2. Einlesen der Intensitäten jeder Zelle
3. Entfernen von Zellen mit einem schlechten Signal-zu-Rausch-Verhältnis, um Verfälschungen der Daten zu vermeiden
4. Logarithmieren der Daten, da häufig so eine Normalverteilung (Glockenkurve) der Expressionsraten erreicht wird. Ein Beispiel hierfür zeigen folgende Histogramme, in der links die Verteilung der Intensitäts-Verhältnisse von Cy5 zu Cy3 pro Zelle gezeigt ist. Rechts zu sehen ist im Hintergrund die gleiche Verteilung, jedoch logarithmiert und im Vergleich zu einem weiteren Datensatz (Balken).



(aus dem Buch „Bioinformatics“ von C.A. Orengo)

7.3.1 Bearbeitung der Expressionsdaten

In biologischen Experimenten können häufig Abweichungen auftreten, die sowohl zufällig als auch systematisch sein können. Zufällige Abweichungen entstehen z.B. in der absoluten Menge an mRNA, der Hybridisierungs-Technik als auch in den Waschschritten. Systematische Unterschiede treten z.B. bei den physikalischen Fluoreszenzeigenschaften der Farbstoffmoleküle auf. Um möglichst wenig systematische Schwankungen zu erhalten, werden **Normalisierungsmethoden** angewandt.

Globale Normalisierung

Der globalen Normalisierung liegen zwei Annahmen zugrunde. Zum Einen geht man davon aus, dass die meisten Gene nicht differentiell exprimiert sind, d.h. der Expressionslevel zwischen zwei mRNA-Populationen ändert sich nicht. Zum Anderen nimmt man an, dass die Intensitäten der zwei Farbstoffe auf einem Microarray über einen konstanten Faktor zusammenhängen:

$$Cy5 = k * Cy3$$

Logarithmiert ergibt sich daraus also:

$$\log_2\left(\frac{Cy5}{k * Cy3}\right) = \log_2\left(\frac{Cy5}{Cy3}\right) - \log_2(k)$$

Als konkretes Beispiel dienen uns im Folgenden 5 Proben, in denen die logarithmierten Cy5/Cy3-Verhältnisse für Gen A und Gen B betrachtet werden. Durch Berechnung des Median für alle Verhältnisse ergibt sich für Gen A ein Median-Wert von 2, für Gen B 5,32. Abschließend wird für beide Gene durch Subtraktion des Median von jedem log-Verhältnis erreicht, dass der Median für alle Werte gleich ist, in diesem Fall 0. Grafisch lässt sich dieses Normalisierungsergebnis wie folgt darstellen:

		Sample				
		1	2	3	4	5
Log_2 ratios	Gene A	0	1	2	3	4
	Gene B	3.32	4.32	5.32	6.32	7.32
Median value	Gene A			2		
	Gene B			5.32		
Median Centered	Gene A	−2	−1	0	1	2
	Gene B	−2	−1	0	1	2

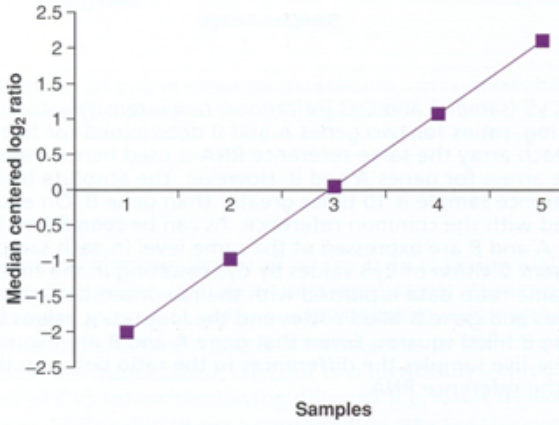


Figure 14.7

Data from Figure 14.7 median centered for both genes and plotted. The median gene expression value is now zero. Median centering removes the effect of the reference RNA having 10 times greater gene A expression compared to gene B. After median centering the $\log_2$ ratios for genes A and B are the same reflecting the observed fact that the raw Cy5 intensities both gene A and B are expressed at the same level in each sample.

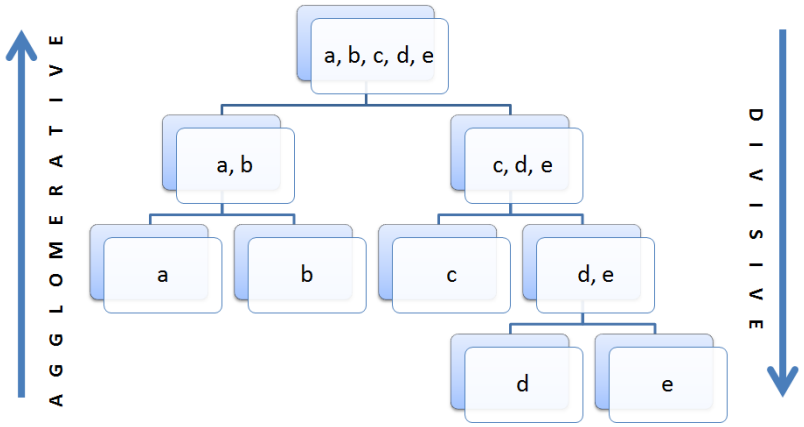
(aus dem Buch „Bioinformatics“ von C.A. Orengo)

Nachdem die Daten nun normalisiert worden sind, kann man mithilfe von hierarchischem Clustern die Expressionswerte analysieren.

7.3.2 Clustering Methoden

Clusterverfahren ermöglichen eine automatische Klassifizierung von Daten mit dem Ziel, diese Daten in sog. *Cluster* zu gruppieren bzw. segmentieren, sodass die Objekte eines Clusters eine höhere Ähnlichkeit zueinander aufweisen, als Objekte verschiedener Cluster. Clustermethoden lassen sich methodisch in zwei Bereiche aufteilen, zum Einen das *hierarchische Clustering* und zum Anderen das *partitionierende Clustering* (z.B. k-means), wobei hier nur auf hierarchische Methoden weiter eingegangen werden soll.

Hierarchisches Clustern
Mithilfe des hierarchischen Clustering werden die Daten nicht in einem einzigen Schritt einem bestimmtem Cluster zugewiesen, sondern durch eine Serie von Schritten zugeordnet, wodurch eine Hierarchie entsteht. Nachteil im Vergleich zum partitionierenden Clustering ist dadurch, dass einmal gebildete Cluster nicht wieder aufgelöst werden können, sondern die Hierarchie bestehen bleibt.
Innerhalb des hierarchischen Clustering gibt es zwei Ansatzmöglichkeiten, die als **bottom up** bzw. **agglomerativ** und **top down** bzw. **aufteilend (divisive)** bezeichnet werden. Agglomerative Verfahren fusionieren n einzelne Datenobjekte in Cluster, wohingegen spaltende Methoden ein Cluster aus allen n Datenobjekten in mehrere Cluster aufteilen. Folgende Grafik veranschaulicht in einem sog. *Dendrogramm* den hierarchischen Aufbau:

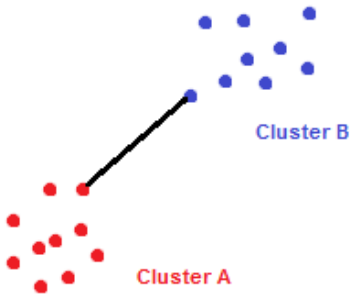


Eine hierarchische, **agglomerative clustering** Prozedur produziert aus n einzelnen Clustern, die jeweils nur ein einziges Datenobjekt enthalten, Cluster mit mehreren (oder vielen) Objekten, indem in jedem Prozedurschritt i diejenigen zwei Datenobjekte zu einem Cluster zusammengefügt werden, die am ähnlichsten zueinander sind. Der Unterschied in den einzelnen agglomerativen Prozeduren beruht auf der Definition der *Ähnlichkeit* und

wird in Form einer Distanz ausgedrückt, was im Folgenden anhand dreier Distanzmaße genauer beschrieben wird.

1) Das **single linkage clustering** benutzt als Distanz- und damit Ähnlichkeitsmaß die minimale Distanz zweier Elemente:

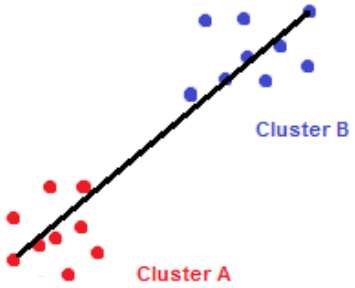
$$D(A, B) = min_{a \in A, b \in B}\ d(a, b)$$



Hierzu wird für jedes mögliche Objektpaar aus $a \in A$ und $b \in B$ die Distanz berechnet und das Minimum bestimmt, was somit die kürzeste Verbindung (linkage) zwischen den zwei Clustern A und B beschreibt.

2) Das **complete linkage clustering** benutzt im Gegensatz zum *single linkage clustering* als Distanzmaß die maximale Distanz zweier Elemente:

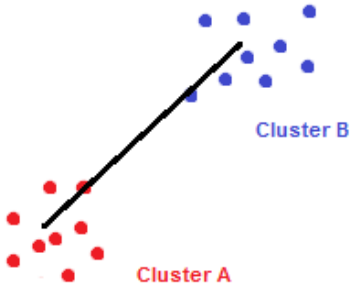
$$D(A, B) = max_{a \in A, b \in B}\ d(a, b)$$



Aus der Berechnung der Distanz für jedes mögliche Objektpaar aus $a \in A$ und $b \in B$ wird das Maximum ermittelt, welches dann die längste Verbindung (linkage) zwischen den zwei Clustern A und B beschreibt.

3) Das **average linkage clustering** benutzt als Distanzmaß die durchschnittliche Distanz aller Elemente zweier Cluster:

$$D(A, B) = \frac{1}{|A|\,|B|} \sum_{\substack{a \in A, \\ b \in B}} d(a, b)$$



Durch Addition der Distanz für jedes mögliche Objektpaar aus $a \in A$ und $b \in B$ und Division durch das Produkt der Cluster-Größen wird die durchschnittliche Distanz der zwei Cluster ermittelt.

Der zweite Ansatz ist das **divisive clustering**. Prinzip des *aufteilenden Clustering* ist, dass ein Cluster mit allen n Datenobjekten solange geteilt wird, bis jedes Datenobjekt ein einzelnes Cluster darstellt.

Top-down Ansätze wie das divisive clustering sind komplexer als agglomerative clustering Verfahren, können jedoch effizienter sein, wenn nicht die komplette Hierarchie bis zu n Einzeldatenobjekt-Clustern vollzogen wird.
Insgesamt sind hierarchische clustering Verfahren auf wenige tausend Datenobjekte limitiert aufgrund der hohen Anzahl an Schritten.

SOM - Kohonen Netzwerk
SOM (auch: benannt nach seinem Entwickler: *Kohonen Netzwerk*] sind eine Art neuronale Netze, die aus einer vorbestimmten Anzahl von Knoten bestehen, wobei jeder Knoten für einen Gencluster steht. SOM hat eine typische 2D Gitterstruktur, in der die Knoten miteinander verbunden sind. Während des Trainierens des Netzes werden die Daten nacheinander addiert und dazu benutzt, die Knoten auszurichten. Sind alle Expressionsdaten abgearbeitet, zeigt das Ergebnis, welche Gene mit welchem Knoten assoziiert sind. Diese Eigenschaften der Daten definieren dann die Gencluster. Die mit jedem Knoten assoziierten Daten können nun überprüft werden, um die Charakteristiken und die Variationen innerhalb des Knotens zu bestimmen.

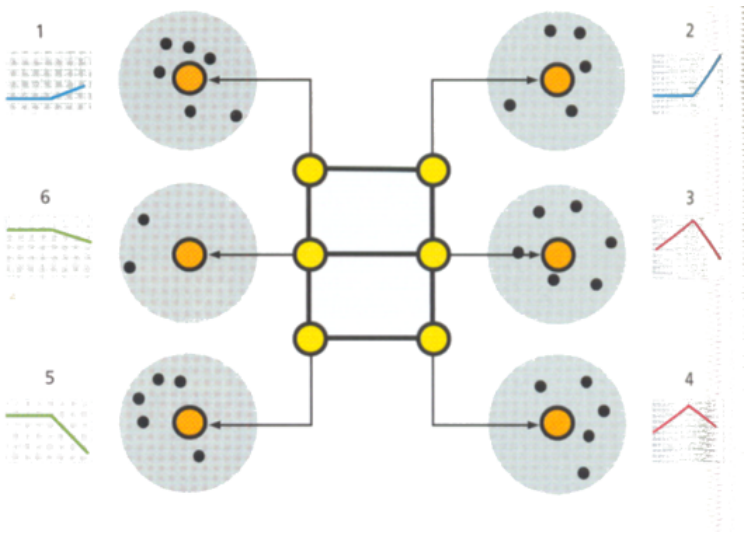


Figure 16.15
A diagrammatic representation of a self-organizing map (SOM). The initial geometry of the nodes in a 3 × 2 rectangular grid is given by solid lines connecting the nodes (yellow circles). The nodes migrate as they adapt to fit the data during successive iterations of the SOM training algorithm (arrows and orange circles). Individual data points are represented by black dots and the six clusters associated with the nodes in their final positions by large gray circles. Average patterns are shown for each of the nodes. Nodes such as 3 and 4, which are geometrically close, contain patterns that are similar. Adapted from Figure 1 of Tamayo et al., 1999.

Das Bild zeigt die initiale Geometrie der Knoten in einem 3 x 2 Gitter (gelbe Knoten). Während der Iterationen des Trainings-Algorithmus wandern die Knoten, um sich den Daten anzupassen (Pfeile + orange Knoten). Die individuellen Datenpunkte sind als schwarze Punkte gezeigt, die sich um einen Knoten herum lagern, wodurch sich 6 Cluster um die finale Position der Knoten bilden (grau hinterlegt). Neben jedem Cluster sind die gemittelten Muster gezeigt.

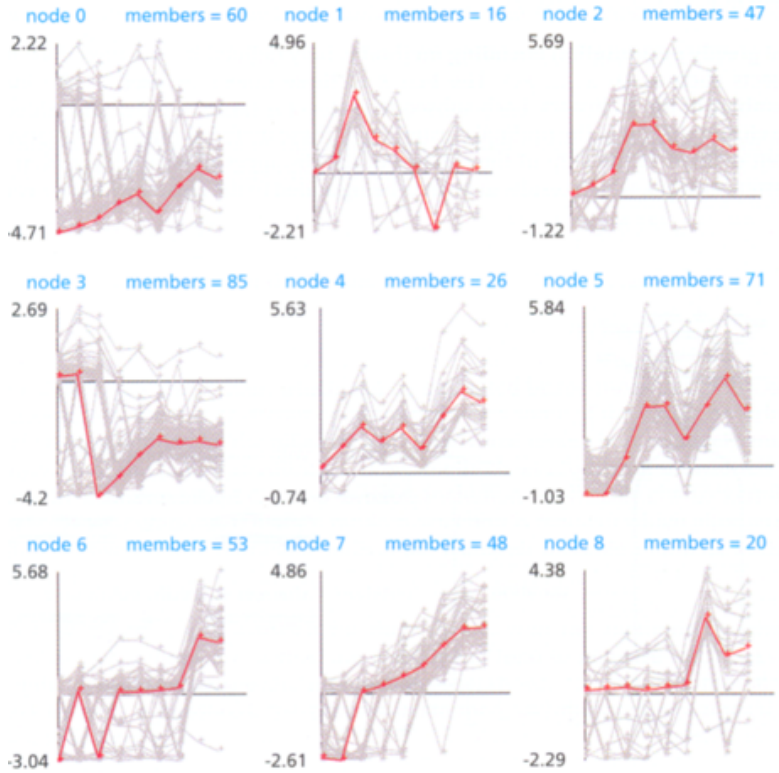
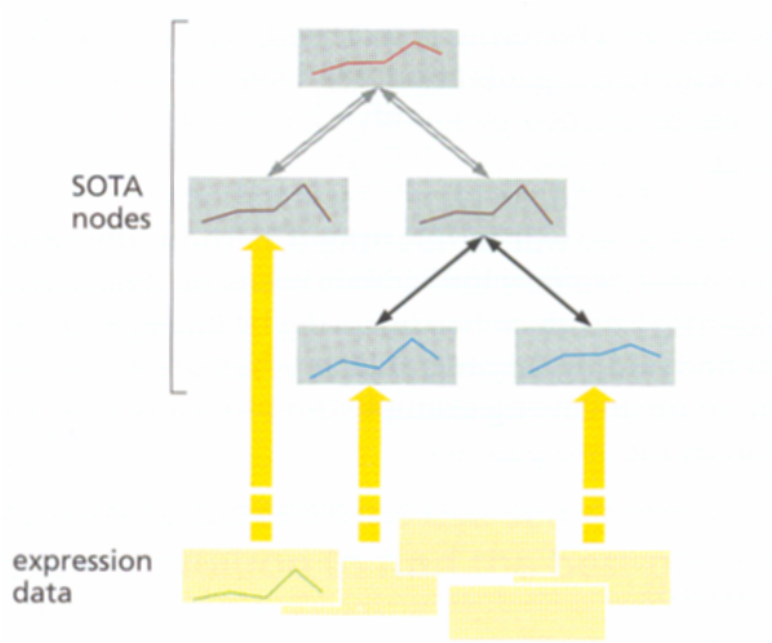


Figure 16.17
An example of the results from a SOM analysis. The SOM used is arranged as a 3 × 3 grid, labeled nodes 0 to 8. For each node the expression level patterns are shown in gray for all members of the node, with the average for the node shown in red. There is a considerable variation in the number of members of each node and the variability of the patterns in each node.

In diesem Bild ist das Ergebnis einer SOM Analyse für ein 3 x 3 Gitter gezeigt. Die Knoten sind mit 0 - 8 beschriftet und die Anzahl der Muster (patterns) in dem Cluster ist gegeben. Jedes Muster der Expression ist in grau dargestellt, das gemittelte Muster in rot. Betrachtet man die Knoten 0-3, führt das zu der Annahme, dass man in diesem Fall mehr Knoten benötigt. Knoten 5 hingegen zeigt ein sehr gutes Ergebnis, welches indiziert, dass die Klassifikation komplett ist.
Der Nachteil der SOM Methode ist, dass die Anzahl der Cluster schon im Voraus festgelegt werden muss. Der Vorteil gegenüber den hierarchischen Cluster-Methoden ist, dass SOM die Cluster klar definiert.
Ein frei verfügbares Programm, das SOM verwendet, ist z.B. die Gen Expressionsanalyse von GenePattern (*http://www.broad.mit.edu/cancer/software/genepattern*).

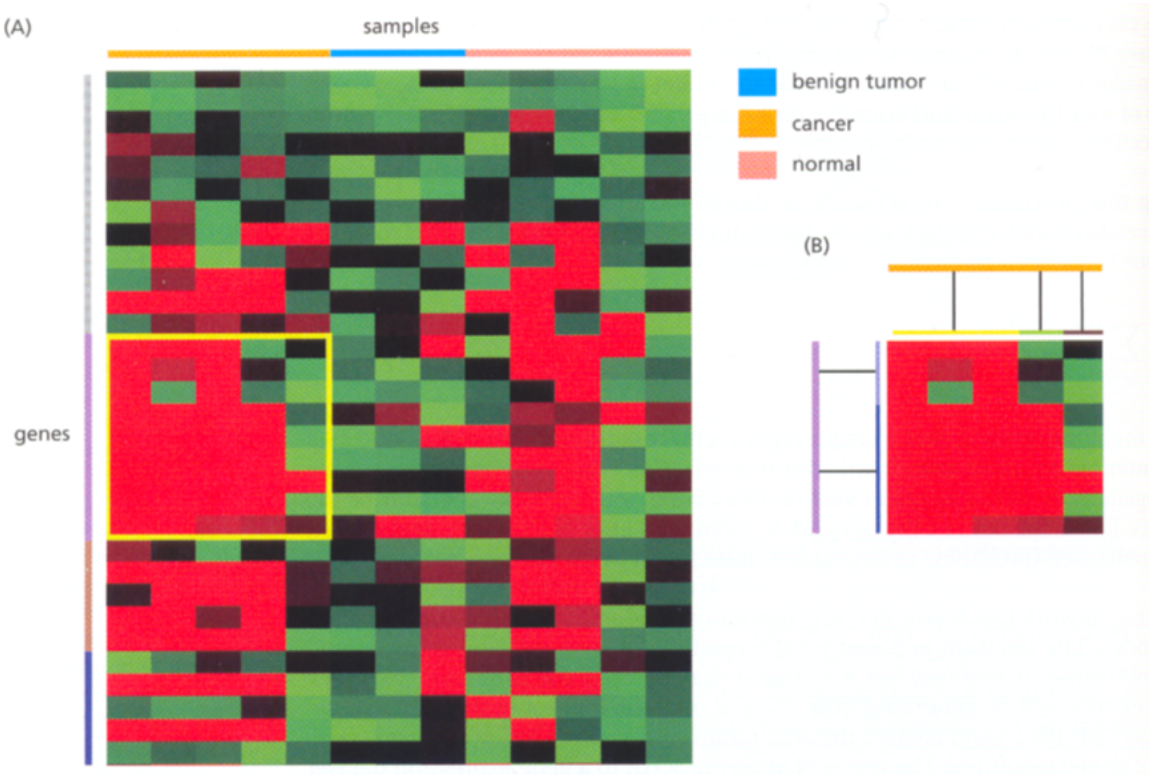
SOTA ist eine Kombination aus SOM und einer Technik, die die Anzahl der Knoten wahlweise vergrößert. Jedes Gen wird mit einem Vektor der Expressions-Messungen dargestellt und jeder Knoten ist über einen äquivalenten Vektor definiert. Der Aufbau des SOTA Netzwerkes unterscheidet sich zu SOM darin, dass es hierarchisch aufgebaut ist. Jeder innere Knoten ist ein Vorgänger-Knoten zu zwei anderen Knoten. Ein zweiter signifikanter Unterschied ist, dass nur eine begrenzte Zahl von Knoten in einem Trainings-Schritt eingepasst werden können. Die externen Knoten werden Zellen genannt und nur diese und ihr direkter Vorgänger können in weiteren Trainingsschritten verändert werden. Das initiierte SOTA Netzwerk besteht aus drei Knoten, deren Vektoren aus dem Durchschnitt aller Daten bestehen. Einer dieser Knoten ist der Vorgänger der anderen beiden. Danach wird das Netzwerk in sich wiederholenden Schritten aufgebaut. Das Ergebnis von SOTA

ist ein hierarchisches Cluster-Dendrogramm, in dem jeder Cluster eine definierte Grenze besitzt. Wird die Grenze z.B. auf 0 gesetzt, läuft der Algorithmus solange bis jeder Cluster genau einen Datenpunkt enthält.



Das Bild zeigt eine schematische Darstellung der SOTA-Methode. Die oberen drei Knoten sind die Initialknoten. Zu diesen werden die ähnlichsten Pattern zu Clustern hinzugefügt. Der rechte äußere Knoten enthält die größte Variation an Daten, daher werden dort die nächsten Knoten angehängt.

Eine weitere Möglichkeit Daten zu clustern, ist das zwei-stufige Clustern *biclustering*. Dabei werden mehr Informationen verwendet, z.B. werden mit *biclustering* die Gewebeproben von Patienten mit einem spezifischen Leiden nach der medizinischen Diagnose klassifiziert. Dabei werden in einer weiteren Unterteilung verschiedene Faktoren, wie z.B. das Alter, Lebensbedingungen, andere Krankheiten, Nicht-/Raucher usw., berücksichtigt. So werden die Patienten in spezielle Untergruppen geclustert. Das folgende Bild zeigt ein Beispiel für ein *biclustering* mit der Methode SOM. Die Proben stammen dabei aus drei verschiedenen Gewebetypen, einmal normales Gewebe, einmal Krebszellengewebe und einmal Gewebe von gutartigen Tumoren. Im ersten Cluster-Schritt (A) sind die Ergebnisse in drei Cluster gemäß der Gewebetypen und in vier Gen Cluster unterteilt (s. farbliche Balken oben und an der Seite). Die SOM Cluster werden durch die farblichen Felder entlang des Gitters gezeigt. Das gelb umrandete Feld hebt eine Kombination aus Gen und Proben Clustern hervor, die im nächsten Schritt weiter analysiert werden. Die weitere Analyse (B) zeigt drei Untercluster der Proben und zwei Gen Untercluster.
Diese Schritte können auch noch weitergeführt werden, wenn eine weitere Verfeinerung der Ergebnisse erwünscht ist.



Anhand der Mikroarrays ist es auch möglich zu erkennen, welche Gene gleichzeitig exprimiert (*co-expressed*) sind. Ebenso gilt das für die Analyse von Proteinen, (in der Vorlesung nicht behandelt). Aus diesen Erkenntnissen lassen sich Proteininteraktions-Netze erstellen, welche in Kapitel 8.6 genauer behandelt werden.

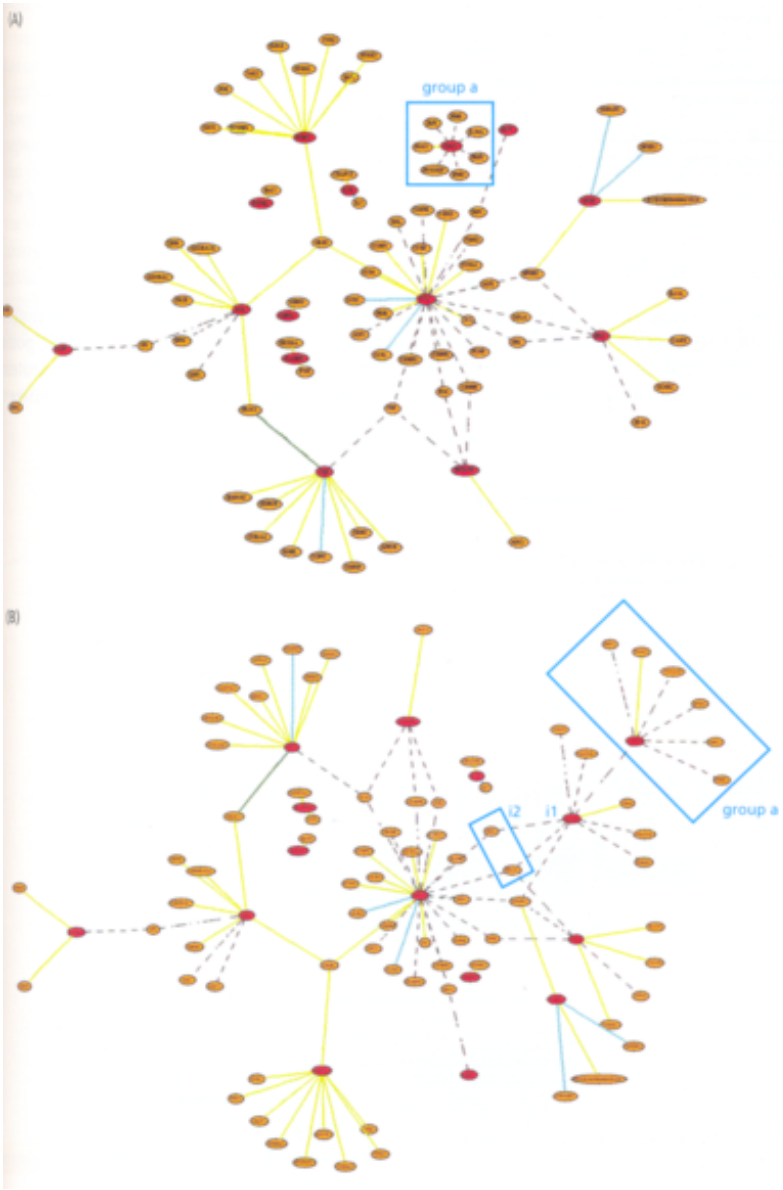


Figure 15.10
An example of protein networks from clustering. Once a cluster of interesting genes (or proteins) has been obtained the members of that cluster can be submitted to a protein–protein (gene–gene) interaction map finder, such as pSTIING. (A) This shows the genes submitted from the node 0 cluster from Figure 15.8B (red circles). Those genes that interact either directly (solid line) or by transcriptional activation (dotted line) are considered to be functionally associated and may be part of a specific pathway. Sometimes, the genes will interact but through further intermediary units, and the map has to be extended. (B) This illustrates how two unconnected groups can be connected by including further interactions. Group a now joins the rest of the network via an intermediary i1 which connects group a and the rest of the network through other proteins (i2).

8 Systembiologie

Obwohl die Anfänge der Systembiologie bis in die erste Hälfte des letzten Jahrhunderts zurückreichen, erlangte dieses Arbeitsfeld erst in den letzten Jahren durch die Verfügbarkeit großer Datensätze aus genomischen und proteomischen Hochdurchsatzverfahren immer stärkere Bedeutung. Um die Funktionalität von zellulären Prozessen, kompletten Zellen, Organen und sogar Organismen zu verstehen, müssen die integrierten Gene und Proteine mittels eines dynamischen Netzwerkes, das deren Interaktionen untereinander beschreibt, analysiert werden.
Leider tendieren biologische Systeme dazu, sehr komplex zu sein und es ist schwierig, alle relevanten Reaktionen für das dynamische Verhalten eines Systems zu erkennen. Man versucht daher, die experimentellen Daten in Modelle, die biologischen Netzwerke, zu übertragen. Tools zur Modellierung, Simulation und Visualisierung eines Systems sind z.B. **Copasi** (s. Kapitel 9.4) oder **ProMoT**.
Ein solches Netzwerk ist zum Beispiel eine Gruppe miteinander verknüpfter metabolischer Pfade.
Der Prozess, der während der zellulären Atmung abläuft, kann z.B. über ein Netzwerk beschrieben werden. Im Bild sind die beiden Kreisläufe der Glykolyse und des Zitronensäurezyklus gezeigt, welche über einen Zwischenschritt miteinander verbunden sind. In der Glykolyse wird Glucose in Pyruvat umgewandelt. Dieses dient als Input für den Zitronensäurezyklus, es kann jedoch auch in einem Zwischenschritt zu Lactat umgewandelt werden. Der gesamte Prozess dient der Gewinnung von Energie, also der Produktion von ATP.
Die Gesamtbilanz kann durch folgende Formel beschrieben werden:

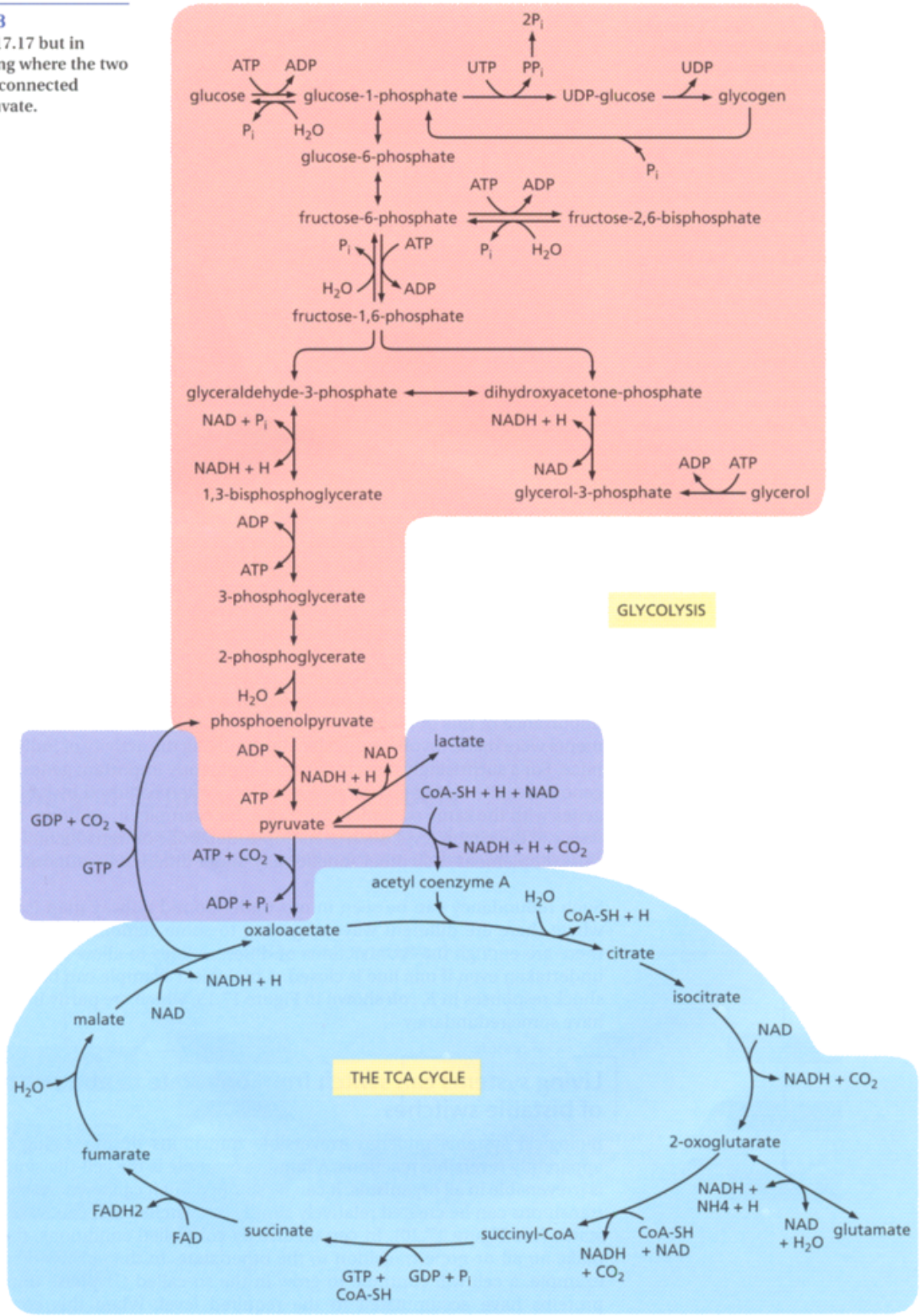
$$C_6H_{12}O_6 + 10NAD^+ + 4ADP + 4P_i + 2FAD + 2H_2O$$

$$\downarrow$$

$$6CO_2 + 10NADH + 10H^+ + 4ATP + 2FADH_2$$

Diese Summenformel ergibt sich durch Betrachtung der Glykolyse und Citratzyklus, verknüpft über Reaktionen, um das Pyruvat in für den Citratzyklus wichtige Moleküle umzuwandeln, wie folgendes Schema des Stoffwechsels verdeutlicht:

Figure 17.18
As in Figure 17.17 but in detail, showing where the two modules are connected through pyruvate.



8.1 Metabolische Pfade in der post-genomischen Ära

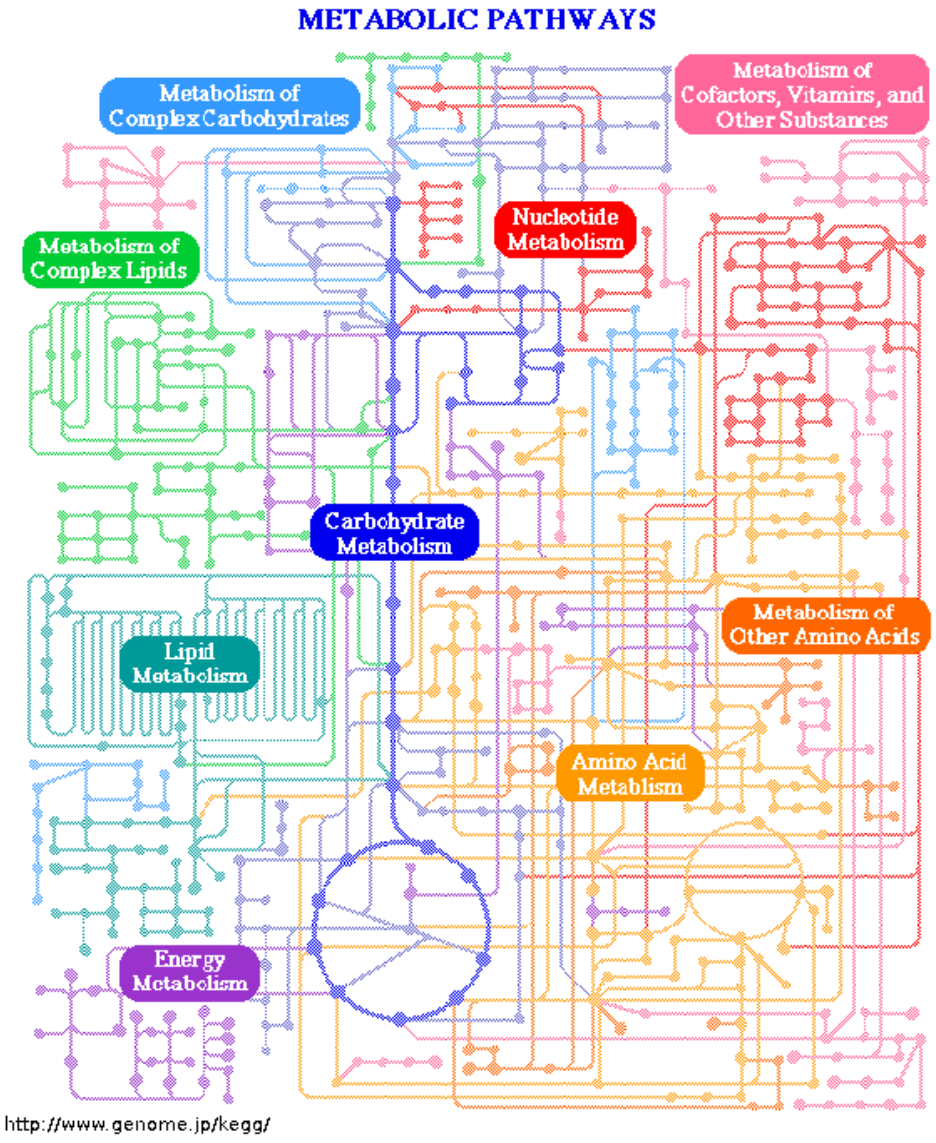
Die Hierarchie der Netzwerk-basierten Pfade lässt sich in drei Schritte gliedern, wobei sich die Beschreibung des zellulären Metabolismus historisch gesehen von a) nach c) entwickelte:

a) Die klassische Biochemie bestimmt die Stöchiometrien einzelner Reaktionen.

b) Viele Reaktionen werden gruppiert und nach gemeinsamen Metaboliten katalogisiert. Dies führt zu traditionellen Pfaden wie der bereits vorgestellten Glykolyse.

c) Durch die kompletten Informationen können nun die kompletten metabolischen Pfade zugeordnet werden.

Die Datenbank KEGG bietet ein interaktives Interface, das zeigt, wie die metabolischen Pfade zusammenhängen. In das Interface kann man „hineinzoomen“ und sich die einzelnen metabolischen Pfade anzeigen lassen.



Eine ähnlich gute Darstellung gibt es unter dem Link:
http://www.sigmaaldrich.com/img/assets/4202/Metabolicpathways_updated_4.19.05.pdf

Diese traditionellen metabolische Pfade dienen als konzeptioneller Rahmen für Forschung und Lehre. Sie erlauben es, Metabolismen verschiedener Organismen miteinander zu vergleichen. Sie sind jedoch nicht für eine quantitative, systemische Bewertung biologischer Reaktionsnetzwerke geeignet, da sie nur Teile der Netzwerke darstellen. Sie wurden oft in Zelltypen entdeckt, in denen sie wichtige metabolische Funktionen übernehmen (z.B. Glykolyse in Hefe). Man kann diese Pfade jedoch nicht einfach auf andere Zelltypen mit anderen Enzymleveln und metabolischen Profilen übertragen.

8.2 Beschreibung vernetzter metabolischer Pfade

Ein vernetzter metabolischer Pfad wird beschrieben, indem man aus genomischen, biochemischen und physiologischen Daten ein Reaktionsnetzwerk aufstellt. Elementare, biochemische Reaktionen sind z.B. folgende, wobei C den Primärmetabolit, P z.B. eine Phosphatgruppe und A einen Cofaktor (z.B. ATP) darstellt:

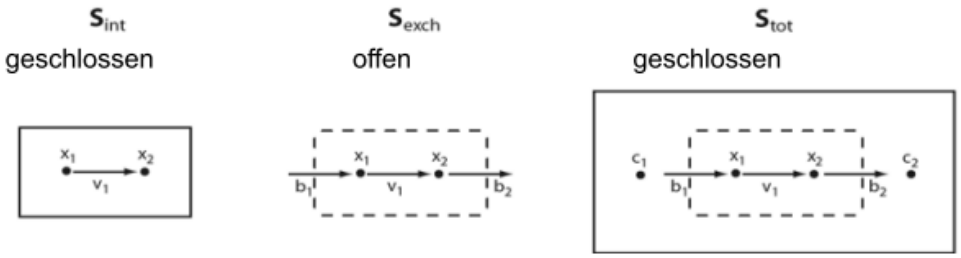
- reversible Umwandlungen:
$$CP \overset{v_1}{\longleftrightarrow} PC$$
- Bi-molekulare Vereinigung:
$$C + P \overset{v_2}{\longleftrightarrow} CP$$
- Cofaktor-gebundene Reaktionen:
$$C + AP \overset{v_3}{\longleftrightarrow} CP + A$$

Ein solcher Reaktionspfad kann sowohl **linear** als auch **nichtlinear** sein. Ein linearer Pfad für einen Metabolit C ist jener, bei dem C nur durch eine einzige Reaktion gebildet und nur durch eine Reaktion verbraucht wird. Jedoch sind solche Reaktionspfade eher selten in den Stoffwechselwegen und damit nicht lineare Pfade realistischer. Als ein nichtlinearer Pfad für einen gegebenen Metabolit C gilt z.B., wenn C entweder durch mehr als eine Reaktion gebildet und/oder durch mehr als eine Reaktion verbraucht wird. ATP ist z.B. in etwa 150 Reaktionen involviert.

Weiterhin ist es wichtig, die **Systemgrenzen** des metabolischen Netzwerks zu analysieren. So wird zwischen einem internen und damit geschlossenen Netzwerk, einem Austausch-System und damit offenem Netzwerk oder einem Gesamtnetzwerk unterschieden. Aufgrund dieser Unterscheidung muss zwischen folgenden Termen differenziert werden, die zur Beschreibung der Reaktionen mithilfe der daran beteiligten internen und externen Konzentrationen und Flüsse dienen:

- Interne Konzentrationen: x
- Externe Konzentrationen: c
- Interne Flüsse (Reaktionen): v
- Austausch-Flüsse: b

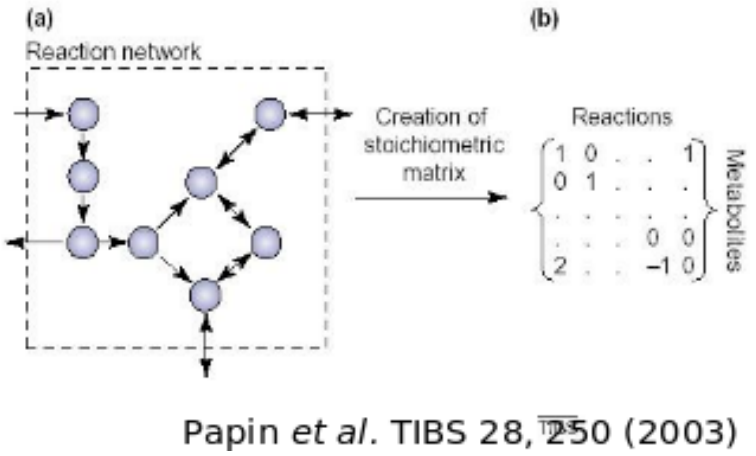
Aufgrund dieser Definitionen lassen sich die drei verschiedenen Systeme grafisch beschreiben, wobei S_{int} für ein internes System, S_{exch} für ein Austausch-System mit externen Flüssen und S_{tot} für ein Gesamtnetzwerk steht:



(aus dem Buch „Systems Biology“ von Bernhard O. Palsson)

Wie deutlich wird, sind für ein internes Netzwerk nur die internen Konzentrationen x_i und Flüsse v_i relevant, für ein Austausch-System zusätzlich noch die externen Flüsse b_i.

Dieses Netzwerk wird dann durch eine stöchiometrische Matrix dargestellt. Die Matrix charakterisiert das Netzwerk aller Stoffumwandlungen im betrachteten System. Mögliche Zustände der Zelle aufgrund dieser Matrix werden durch Techniken wie die *Flux Balance Analyse* (FBA) bestimmt.



8.3 Aufbau und Analyse der stöchiometrischen Matrix

Mathematisch beschreibt eine stöchiometrische Matrix S die lineare Transformation eines Flussvektors $v = (v_1, v_2, ..., v_n)$ in die zeitliche Veränderung des Konzentrationsvektors $x = (x_1, x_2, ..., x_m)$:

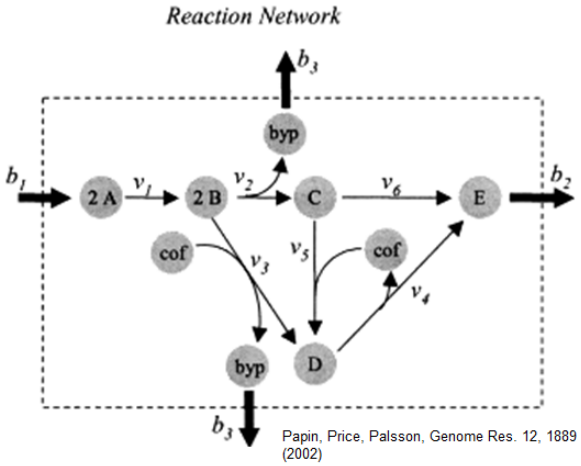
$$\frac{dx}{dt} = S * v$$

Dieses Gleichungssystem aus m Gleichungen für die zeitlichen Änderungen der Konzentrationen der m Metabolite bildet die *dynamischen Massenbalancen* ab, die alle funktionalen Zustände eines rekonstruierten chemischen Reaktionsnetzwerkes charakterisieren. Jede einzelne Gleichung steht für die Summation aller Flüsse v_k, die die Komponente x_i bilden oder abbauen:

$$\frac{dx_i}{dt} = \sum_k s_{ik} v_k$$

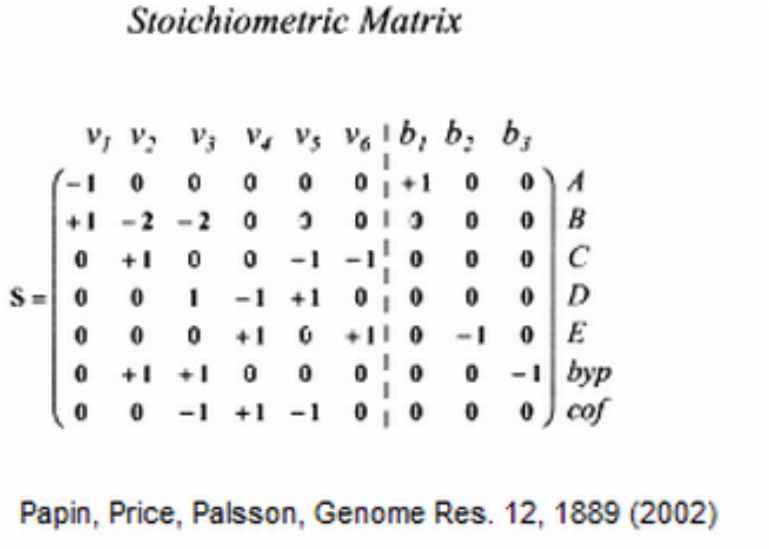
Die Dimension der Matrix S ist $m \times n$, wobei die m Zeilen für die Anzahl der Metabolite stehen und die n Spalten für die Anzahl an Reaktionen. Typischerweise gilt meist $n > m$, da es mehr Reaktionen als Metabolite gibt. Die Einträge innerhalb der Matrix werden **stöchiometrische Koeffizienten** genannt und beschreiben die Proportionen, mit denen die Edukte und Produkte in die biochemischen Reaktionen eingehen. Das Vorzeichen der Einträge richtet sich danach, in welche Richtung die Reaktion verläuft ($-$ für verbrauchend, $+$ für bildend).
Der Vektor v enthält alle Flüsse der Einzelreaktionen, die im metabolischen Netzwerk auftreten können, einschließlich der internen Flüsse, der Transportflüsse und der Flüsse für das Wachstum.

Um nun eine stöchiometrische Matrix anhand eines metabolischen Stoffwechselpfads aufstellen zu können, muss man zuerst einen Überblick über das zu betrachtende Reaktionsnetzwerk gewinnen, um alle Konzentrationen und Flüsse und damit die Dimension der Matrix zu bestimmen.



In dem oben gezeigten Reaktionsnetzwerk gibt es 6 interne $(v_1, ..., v_6)$ und 3 externe (b_1, b_2, b_3) Flüsse (also 9 insgesamt), sowie 7 Metabolite (A-E, byp, cof). Es ergibt sich

also eine 7 x 9 Matrix.



Gemäß des Reaktionsnetzwerkes ergeben sich also für Metabolit A (erste Zeile) folgende stöchiometrischen Koeffizienten:
Die Bildung von A erfolgt durch den externen Fluss b_1, indem dem System eine Einheit hinzugefügt wird, aus der 2 Einheiten gebildet werden, daher +1. Weiterhin wird der Metabolit A im internen Fluss v_1 abgebaut, wodurch 2B gebildet werden. Da der Faktor 2 erhalten bleibt und A verbraucht wird, ist der stöchiometrische Eintrag -1. Alle weiteren Einträge sind gleich Null, da Metabolit A an keiner weiteren Reaktion beteiligt ist. Anhand dieses Beispiels lassen sich alle weiteren Matrixeinträge bilden.

8.4 Analyse der Flussbalance

Chemische Reaktionen bedingen stets die Erhaltung der Masse. Durch diese Anforderung lassen sich alle möglichen Flußverteilungen in einer biologischen Zelle im Gleichgewicht berechnen. Ein Gleichgewicht ist dann erreicht, wenn die Konzentrationen der internen Metabolite konstant bleiben. Die einzige Erfordernis hierzu ist die Kenntnis über die Stöchiometrie der metabolischen Pfade und die metabolischen Anforderungen. Für jeden Metaboliten gilt:

$$v_i = \frac{dx_i}{dt} = V_{synthetisiert} - V_{abgebaut} - (V_{hinaustransportiert} - V_{hineintransportiert})$$

Im Gleichgewicht (*steady-state*), in dem die zeitlichen Änderungen der Konzentrationen aller m Metaboliten $\frac{dx_i}{dt} = 0, i = 1, ..., m$ sind, gilt mit derselben Matrixgleichung wie zuvor:

$$\mathbf{S} * v = 0$$

Da die Anzahl der Metabolite generell kleiner als die Anzahl der Reaktionen ist, ist die Matrixgleichung fast immer unterbestimmt. Dies bedeutet, dass das Gleichungssystem mehrere Lösungen besitzt, also mögliche Flussverteilungen in der Zelle. Um nun den tatsächlichen Fluss durch das System zu bestimmen (z.B. in der Arbeitsgruppe von Prof. Dr. Heinzle an der UdS), benötigt man entweder zusätzliche Informationen aus Experimenten, oder man legt Grenzen für die Flüsse fest.

$$\alpha_i \leq v_i \leq \beta_i$$

Die lineare Ungleichung wird benutzt, um die Reversibilität/ Irreversibilität metabolischer Reaktionen und die maximalen metabolische Flüsse der Transportreaktionen zu erzwingen.

Mit der Flux Balance Analysis (FBA) - Methode bestimmt man aus der großen Anzahl an möglichen Lösungen durch mathematische Optimierung eine bestimme Lösung, die eine Zielfunktion optimiert. Als Zielfunktion für bakterielle Systeme, die vermutlich relativ einfach aufgebaut sind, legt man meist maximales Wachstum fest, also etwa die maximale Menge an produziertem ATP oder dass die Summe aller metabolischen Flüsse im Netzwerk maximal sei.

8.4.1 *E.coli in silico*

Eines der am besten charakterisierten zellulären Systeme ist *Escherichia coli*. Edwards & Palsson konstruierten im Jahr 2000 eine *in silico* Abbildung des *E.coli*-Metabolismus. Die Daten stammten von dem komplett sequenzierten Genom von *E.coli* MG1655 und aus der biochemischen Originalliteratur, genomischen Informationen und den metabolischen Datenbanken EcoCyc und KEGG.
Da *E.coli* schon sehr lange studiert wird, gibt es zu allen metabolischen Reaktionen der *in silico* Abbildung entweder einen biochemischen oder einen genetischen Nachweis für die Funktion des Gens, in den meisten Fällen sogar beides.
Die folgende Abbildung führt alle Gene auf, die für die in silico-Modellierung von E.coli MG1655 verwendet wurden.

Table 1

The genes included in the *E. coli* metabolic genotype (21)

Central metabolism (EMP, PPP, TCA cycle, electron transport)	*aceA, aceB, aceE, aceF, ackA, acnA, acnB, acs, adhE, agp, appB, appC, atpA, atpB, atpC, atpD, atpE, atpF, atpG, atpH, atpI, cydA, cydB, cydC, cydD, cyoA, cyoB, cyoC, cyoD, dld, eno, fba, fbp, fdhF, fdnG, fdnH, fdnI, fdoG, fdoH, fdoI, frdA, frdB, frdC, frdD, fumA, fumB, fumC, galM, gapA, gapC_1, gapC_2, glcB, glgA, glgC, glgP, glk, glpA, glpB, glpC, glpD, gltA, gnd, gpmA, gpmB, hyaA, hyaB, hyaC, hybA, hybC, hycB, hycE, hycF, hycG, icdA, lctD, ldhA, lpdA, malP, mdh, ndh, nuoA, nuoB, nuoE, nuoF, nuoG, nuoH, nuoI, nuoJ, nuoK, nuoL, nuoM, nuoN, pckA, pfkA, pfkB, pflA, pflB, pflC, pflD, pgi, pgk, pntA, pntB, ppc, ppsA, pta, purT, pykA, pykF, rpe, rpiA, rpiB, sdhA, sdhB, sdhC, sdhD, sfcA, sucA, sucB, sucC, sucD, talB, tktA, tktB, tpiA, trxB, zwf,* ***pgl*** *(30),* ***maeB*** *(30)*
Alternative carbon source	*adhC, adhE, agaY, agaZ, aldA, aldB, aldH, araA, araB, araD, bglX, cpsG, deoB, fruK, fucA, fucI, fucK, fucO, galE, galK, galT, galU, gatD, gatY, glk, glpK, gntK, gntV, gpsA, lacZ, manA, melA, mtlD, nagA, nagB, nanA, pfkB, pgi, pgm, rbsK, rhaA, rhaB, rhaD, srlD, treC, xylA, xylB*
Amino acid metabolism	*adi, aldH, alr, ansA, ansB, argA, argB, argC, argD, argE, argF, argG, argH, argI, aroA, aroB, aroC, aroD, aroE, aroF, aroG, aroH, aroK, aroL, asd, asnA, asnB, aspA, aspC, avtA, cadA, carA, carB, cysC, cysD, cysE, cysH, cysI, cysJ, cysK, cysM, cysN, dadA, dadX, dapA, dapB, dapD, dapE, dapF, dsdA, gabD, gabT, gadA, gadB, gdhA, glk, glnA, gltB, gltD, glyA, goaG, hisA, hisB, hisC, hisD, hisF, hisG, hisH, hisI, ilvA, ilvB, ilvC, ilvD, ilvE, ilvG_1, ilvG_2, ilvH, ilvI, ilvM, ilvN, kbl, ldcC, leuA, leuB, leuC, leuD, lysA, lysC, metA, metB, metC, metE, metH, metK, metL, pheA, proA, proB, proC, prsA, putA, sdaA, sdaB, serA, serB, serC, speA, speB, speC, speD, speE, speF, tdcB, tdh, thrA, thrB, thrC, tnaA, trpA, trpB, trpC, trpD, trpE, tynA, tyrA, tyrB, ygjG, ygjH,* ***alaB*** *(42),* ***dapC*** *(43),* ***pat*** *(44),* ***prr*** *(44),* ***sad*** *(45),* ***methylthioadenosine nucleosidase*** *(46),* ***5-methylthioribose kinase*** *(46),* ***5-methylthioribose-l-phosphate isomerase*** *(46),* ***adenosyl homocysteinase*** *(47),* ***L-cysteine desulfhydrase*** *(44),* ***glutaminase A*** *(44),* ***glutaminase B*** *(44)*
Purine & pyrimidine metabolism	*add, adk, amn, apt, cdd, cmk, codA, dcd, deoA, deoD, dgt, dut, gmk, gpt, gsk, guaA, guaB, guaC, hpt, mutT, ndk, nrdA, nrdB, nrdD, nrdE, nrdF, purA, purB, purC, purD, purE, purF, purH, purK, purL, purM, purN, purT, pyrB, pyrC, pyrD, pyrE, pyrF, pyrG, pyrH, pyrI, tdk, thyA, tmk, udk, udp, upp, ushA, xapA, yicP,* ***CMP glycosylase*** *(48)*
Vitamin & cofactor metabolism	*acpS, bioA, bioB, bioD, bioF, coaA, cyoE, cysG, entA, entB, entC, entD, entE, entF, epd, folA, folC, folD, folE, folK, folP, gcvH, gcvP, gcvT, gltX, glyA, gor, gshA, gshB, hemA, hemB, hemC, hemD, hemE, hemF, hemH, hemK, hemL, hemM, hemX, hemY, ilvC, lig, lpdA, menA, menB, menC, menD, menE, menF, menG, metF, mutT, nadA, nadB, nadC, nadE, ntpA, pabA, pabB, pabC, panB, panC, panD, pdxA, pdxB, pdxH, pdxJ, pdxK, pncB, purU, ribA, ribB, ribD, ribE, ribH, serC, thiC, thiE, thiF, thiG, thiH, thrC, ubiA, ubiB, ubiC, ubiG, ubiH, ubiX, yaaC, ygiG,* ***nadD*** *(49),* ***nadF*** *(49),* ***nadG*** *(49),* ***panE*** *(50),* ***pncA*** *(49),* ***pncC*** *(49),* ***thiB*** *(51),* ***thiD*** *(51),* ***thiK*** *(51),* ***thiL*** *(51),* ***thiM*** *(51),* ***thiN*** *(51),* ***ubiE*** *(52),* ***ubiF*** *(52),* ***arabinose-5-phosphate isomerase*** *(22),* ***phosphopantothenate-cysteine ligase*** *(50),* ***phosphopantothenate-cysteine decarboxylase*** *(50),* ***phospho-pantetheine adenylyltransferase*** *(50),* ***dephosphoCoA kinase*** *(50),* ***NMN glycohydrolase*** *(49)*
Lipid metabolism	*accA, accB, accD, atoB, cdh, cdsA, cls, dgkA, fabD, fabH, fadB, gpsA, ispA, ispB, pgpB, pgsA, psd, pssA,* ***pgpA*** *(53)*
Cell wall metabolism	*ddlA, ddlB, galF, galU, glmS, glmU, htrB, kdsA, kdsB, kdtA, lpxA, lpxB, lpxC, lpxD, mraY, msbB, murA, murB, murC, murD, murE, murF, murG, murI, rfaC, rfaD, rfaF, rfaG, rfaI, rfaJ, rfaL, ushA,* ***glmM*** *(54),* ***lpcA*** *(55),* ***rfaE*** *(55),* ***tetraacyldisaccharide 4′ kinase*** *(55),* ***3-deoxy-D-manno-octulosonic-acid 8-phosphate phosphatase*** *(55)*
Transport processes	*araE, araF, araG, araH, argT, aroP, artI, artJ, artM, artP, artQ, brnQ, cadB, chaA, chaB, chaC, cmtA, cmtB, codB, crr, cycA, cysA, cysP, cysT, cysU, cysW, cysZ, dctA, dcuA, dcuB, dppA, dppB, dppC, dppD, dppF, fadL, focA, fruA, fruB, fucP, gabP, galP, gatA, gatB, gatC, glnH, glnP, glnQ, glpF, glpT, gltJ, gltK, gltL, gltP, gltS, gntT, gpt, hisJ, hisM, hisP, hisQ, hpt, kdpA, kdpB, kdpC, kgtP, lacY, lamB, livF, livG, livH, livJ, livK, livM, lldP, lysP, malE, malF, malG, malK, malX, manX, manY, manZ, melB, mglA, mglB, mglC, mtlA, mtr, nagE, nanT, nhaA, nhaB, nupC, nupG, oppA, oppB, oppC, oppD, oppF, panF, pheP, pitA, pitB, pnuC, potA, potB, potC, potD, potE, potF, potG, potH, potI, proP, proV, proW, proX, pstA, pstB, pstC, pstS, ptsA, ptsG, ptsI, ptsN, ptsP, purB, putP, rbsA, rbsB, rbsC, rbsD, rhaT, sapA, sapB, sapD, sbp, sdaC, srlA_1, srlA_2, srlB, tdcC, tnaB, treA, treB, trkA, trkG, trkH, tsx, tyrP, ugpA, ugpB, ugpC, ugpE, uraA, xapB, xylE, xylF, xylG, xylH,* ***fruF*** *(56),* ***gntS*** *(57),* ***metD*** *(43),* ***pnuE*** *(49),* ***scr*** *(56)*

The *in silico E. coli* MG1655 metabolic genotype used herein is available on the web: http://gcrg.ucsd.edu/downloads.html.

Proc Natl Acad Sci U S A. 2000 May 9; 97(10): 5528–5533.

Für die mathematische Flussanalyse wurde für die irreversiblen inneren Flüsse α_i auf 0 und für reversible innere Flüsse α_i auf $-\infty$ gesetzt. Die Reversibilität der metabolischen Reaktionen wurde anhand der Literatur bestimmt. Der Transportfluss anorganischer Phosphate, Ammoniak, Kohlendioxid, Sulfate, Kalium und Natrium war unbeschränkt $\alpha_i = -\infty$ und $\beta_i = \infty$. Der Transportfluss für die anderen Metabolite, sofern sie im Medium vorhanden waren, wurden zwischen 0 und das Maximum gesetzt, $(0 < v_i < v_i^{max})$. Wenn der Metabolit nicht vorhanden war, wurde er automatisch auf 0 gesetzt.
Durch FBA mit linearer Programmierung fand man die Lösung, die die Summe aller metabolischer Flüsse maximierte. Diese Lösung maximierte folgende Zielfunktion

$$Z = \sum c_i * v_i = \langle c * v \rangle$$

Die Flussbalance-Analyse kann ebenfalls dazu benutzt werden, um Veränderungen, die durch Entfernen eines Gens verursacht werden, auf die metabolischen Fähigkeit zu überprüfen. Um den Effekt einer Gen-Deletion virtuell zu simulieren, wird einfach der Fluss durch die dazugehörige enzymatische Reaktion auf 0 gesetzt. Nun vergleicht man den optimalen Wert der „Mutante“ (Z_{Mutant}) mit dem *wild-type* Z, um den Effekt der Deletion auf das metabolische Verhalten des Systems zu bestimmen.

$\frac{Z_{Mutant}}{Z}$

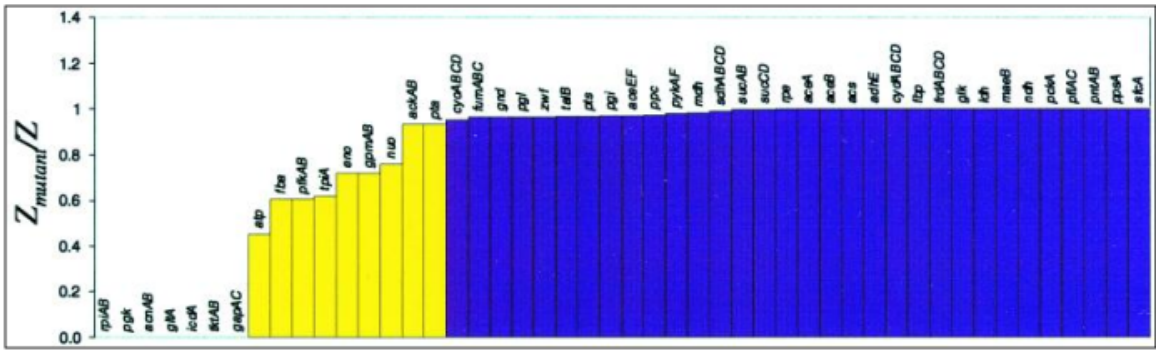


Figure 2

Gene deletions in *E. coli* MG1655 central intermediary metabolism; maximal biomass yields on glucose for all possible single gene deletions in the central metabolic pathways. The optimal value of the mutant objective function (Z_{mutant}) compared with the "wild-type" objective function (Z), where Z is defined in Eq. 3. The ratio of optimal growth yields (Z_{mutant}/Z). The results were generated in a simulated aerobic environment with glucose as the carbon source. The transport fluxes were constrained as follows: $\beta_{glucose}$ = 10 mmol/g-dry weight (DW) per h; β_{oxygen} = 15 mmol/g-DW per h. The maximal yields were calculated by using FBA with the objective of maximizing growth. The biomass yields are normalized with respect to the results for the full metabolic genotype. The yellow bars represent gene deletions that reduced the maximal biomass yield to less than 95% of the *in silico* wild type.

Proc Natl Acad Sci U S A. 2000 May 9; 97(10): 5528–5533.

Interpretation der Ergebnisse durch das Entfernen des Gens:
Die essentiellen Genprodukte, für die $\frac{Z_{Mutant}}{Z}$ auf Null zurückgeht (7 Gene) oder mehr als 5% reduziert ist (9 gelbgefärbte Balken), waren an der Glykolyse, an drei Reaktionen des Zitronesäurezyklus und an mehreren Punkten des Pentose-Phosphat Stoffwechsels beteiligt. Die anderen Gene des zentralen Metabolismus (blau gefärbt) können entfernt werden. Auch ohne sie ist *E.coli in silico* in der Lage sein zelluläres Wachstum zu unterstützen.
Das führt zu der Folgerung, dass anscheinend eine große Anzahl der zentralen metabolischen Gene entfernt werden kann, ohne dass dadurch die Fähigkeit des metabolischen Netzwerkes in Bezug auf das Wachstum unterbinden wird.
Die FBA konstruiert das optimale Netzwerk durch die einfache Verwendung der Stöchiometrie metabolischer Reaktionen. Im Fall von *E.coli* entsprachen die *in silico* Ergebnisse in der Mehrzahl (86 %) den experimentellen Daten.
Weiterhin konnte durch die FBA gezeigt werden, dass es im metabolischen Netzwerk von *E.coli* nur relativ wenige kritische Genprodukte im zentralen Metabolismus gibt. Die FBA identifiziert das Beste, was eine Zelle tun kann, d.h. kurzfristige Veränderungen nicht zu beachten, sondern sie versucht zu überleben. (Fähigkeit zum Wachstum)

Fazit:
Stoffwechselpfade sind nicht isoliert voneinander, sie besitzen in verschiedenen Organis-

men verschiedene Komponenten bzw. Bedeutung. Oft sind mehrere alternative Reaktionspfade möglich. Komplexe metabolische Netzwerke sind daher für uns schwer zu analysieren. Man benötigt intelligente und robuste Algorithmen, um daraus biologische Modelle abzuleiten. Diese biologischen Modelle sehen jedoch vermutlich nicht so einfach aus wie in heutigen Lehrbüchern.

8.5 Proteinkomplexe

Als Proteininteraktionen bezeichnet man sowohl die direkte Wechselwirkung von Proteinen, die permanente Komplexe („Protein Maschinen“) bilden als auch die von Proteinen, die nur vorübergehend (transient) direkt aneinander binden, wie z.B. Hormone mit Rezeptoren der Signalübermittlung. Des weiteren interagieren Proteine auf funktionelle Weise miteinander, wenn sie Co-reguliert exprimiert sind (Gen Co-Expression, Vorlesung 7), zu dem selben metabolischen Pfad gehören, sich ein Substrat teilen oder einfach nur colokalisiert sind.
Die Proteininteraktion ist entscheidend für viele zelluläre Prozesse, die Signalübermittlung, die Energieweitergabe oder die Immunreaktion. Dazu müssen die Proteine große permanente Komplexe, wie z.B. das Ribosom oder das Apoptosom, oder nur transient existierende Komplexe bilden. Treibende Kräfte für diese Komplexe sind für manche Paare die weitreichende elektrostatische Anziehung oder die dauerhafte Adhäsion durch den hydrophoben Effekt.
Die Proteininteraktion lässt sich so grob in die zwei Bereiche Komplexbildung und Interaktion über Abhängigkeiten aufteilen.

8.6 Proteininteraktionsnetzwerke

Zelluläre Prozesse beruhen selten auf der Aktion eines einzelnen Proteins, sondern eher auf der Interaktion von Proteinen mit anderen Molekülen, wobei einer Schätzung zufolge ein Protein mit ca. 3-8 weiteren Proteinen interagiert. Neben Interaktionspartnern wie kleinen Biomoleülen, DNA oder Lipiden entstehen viele Prozesse aufgrund der Bindungen mit anderen Proteinen, wodurch Homo-/ Heterodimere bzw. -Oligomere gebildet werden. Diese Protein-Protein Interaktionen sind hoch spezifisch aufgrund der strukturellen und physikochemischen Eigenschaften der Bindungspartner. Es ist von großem Interesse, solche Protein-Interaktionen zu analysieren und aufgrund der gewonnenen Kenntnisse Forschungsgebiete wie z.B. die Medikamentenforschung & -Entwicklung zu optimieren.

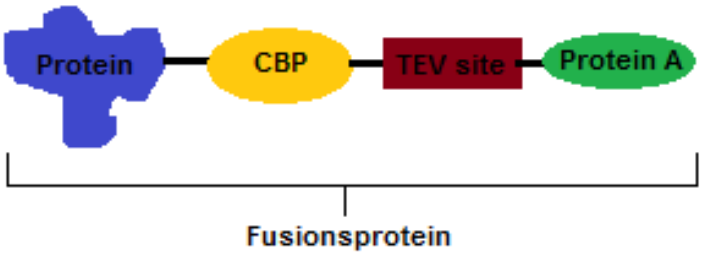
Der Begriff *Interactome* steht für die Gesamtmenge aller molekularen Interaktionen in Zellen, die Analyse dieser Interaktionen nennt man *Interactomics*. Je nachdem, welche Molekülfamilien untersucht werden, bezieht sich z.B. im Bereich der Proteomics das Interaktom auf Protein-Protein Interaktionsnetzwerke. Ein weiterer wichtiger Bereich ist der der Protein-DNA Interaktionen wie z.B. Interaktionen von Transkriptionsfaktoren mit DNA.

8.6.1 Generierung der Rohdaten

Die für die Interaktionsnetzwerke benötigten Rohdaten werden experimentell durch high throughput Methoden wie u.A. die *TAP (Tandem Affinity Purification)* gekoppelt an MS (Massenspektrometrie), *Yeast Two Hybrid System*, *SDS PAGE* oder *FRET (Fluoreszenz Resonanz Energie Transfer)* für Proteininteraktionen generiert. Dynamische Metabolit-Konzentrationen kann man mittels *Fluoreszenz* oder *stopped-flow* ermitteln, räumliche Strukturen durch verschiedene Mikroskopie- bzw. Beugungsverfahren. Weiterhin gibt es viele Datenbanken, auf die man zurückgreifen kann, wie z.B. die bereits bekannten *NCBI*, *KEGG* (metabolische Pfade), *SABIO-RK* (kinetische Daten) oder natürlich die *PDB* für Protein-Strukturen. Jedoch sollte beachtet werden, dass die Qualität der Daten aufgrund unzuverlässiger Experimente oder indirekter Zuordnungen variieren kann.

TAP - Tandem Affinity Purification
Die TAP - Methode liefert mithilfe der direkten Aufreinigung gute und genaue Ergebnisse. Hierzu wird ein als Köder fungierendes Protein, für welches interagierende Proteine gesucht werden, mithilfe eines *tag* markiert. Die Markierung erfolgt, indem durch homologe Rekombination das Köderprotein mit einem TAP-tag versehen wird, was zusammen *Fusionsprotein* genannt wird.
Der TAP-tag besteht im Detail z.B. aus folgenden Bestandteilen:

- CBP - Calmodulin Bindungspeptid
- TEV site - Tobacco etch Virus Protease Schnittstelle
- Protein A aus Staphylococcus Aureus



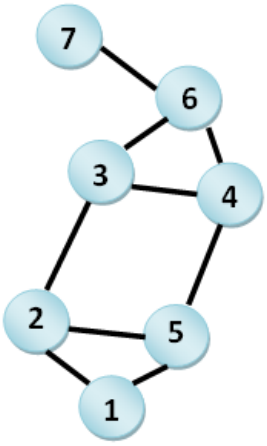
Schematische Darstellung des TAP-tags

Für den genauen Ablauf einer solchen Affinitätsaufreinigung sei an dieser Stelle auf andere Vorlesungen verwiesen (z.B. VL Bioanalytik). Prinzipiell jedoch gilt, dass aus einem Proteingemisch im besten Fall einige Proteine an das Köderprotein binden und anschlieend mithilfe des Fusionsprotein isoliert und mittels Massenspektrometrie (MS) identifiziert werden können.

8.6.2 Aufbau eines Interaktionsnetzwerkes

Interaktionsnetzwerke werden mithilfe mathematischer Grundlagen wie der Graphentheorie realisiert.
Ein Graph G besteht aus einer Menge {V, E}, wobei $V = \{v_1, v_2, ..v_n\}$ die Menge der Knoten (vertices) und $E = \{e_1, e_2, ..., e_n\}$ die Menge der Kanten (edges) darstellt. Im Falle eines Proteininteraktionsnetzwerks handelt es sich bei den Knoten um Proteine und bei den Kanten um Interaktionen zweier Moleküle. Verlaufen die Interaktionen in beide Richtungen, nennt man die Kanten *ungerichtet*. Ist ein Knoten v_j von seinem Vorläuferknoten v_i abhängig, so nennt man die betreffende Kante *gerichtet*, wobei diese dann vorzugsweise durch Pfeile dargestellt wird. Für Interaktionsnetzwerke werden üblicherweise ungerichtete Graphen verwendet, für regulatorische Netzwerke verwendet man gerichtete Graphen

Um aus den ermittelten Rohdaten einen Graph erstellen zu können, muss zuerst eine geeignete Beschreibung der Daten gewählt werden. So werden MS-Daten als Listen von Proteinen gespeichert, welche jedoch auch leicht in eine Matrixdarstellung der Interaktionen überführt werden können. Z.B. ist eine Darstellung mithilfe einer Adjazenzmatrix eine relativ leicht umzusetzende Möglichkeit.
Existiert ein ungerichteter Graph G mit n Knoten ohne Mehrfachkanten, so hat die daraus entstehende Adjazenzmatrix M die Dimension $n \times n$. Der Matrixeintrag $M_{i,j}$ steht für die Interaktion des Proteins i mit Protein j, wobei dieser Matrixeintrag entweder ein Bool'scher Ausdruck (0 = keine Interaktion, 1 = Interaktion) oder ein Integer-Wert sein kann, sofern Gewichte oder Kosten mit der Kante und damit Interaktion verknüpft sind. Wie anhand des folgenden Beispiels deutlich wird, sind Adjazenzmatrizen für ungerichtete Graphen ohne Mehrfachkanten symmetrisch.



$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Weiterhin muss ein geeignetes Gewichtungssystem aufgestellt werden, das jeder einzelnen Interaktion einen Wahrscheinlichkeitswert zuweist, ob diese Interaktion richtig oder falsch ist. Zur Berechnung dieser Interaktionswahrscheinlichkeiten gibt es mehrere Ansätze, z.B.:

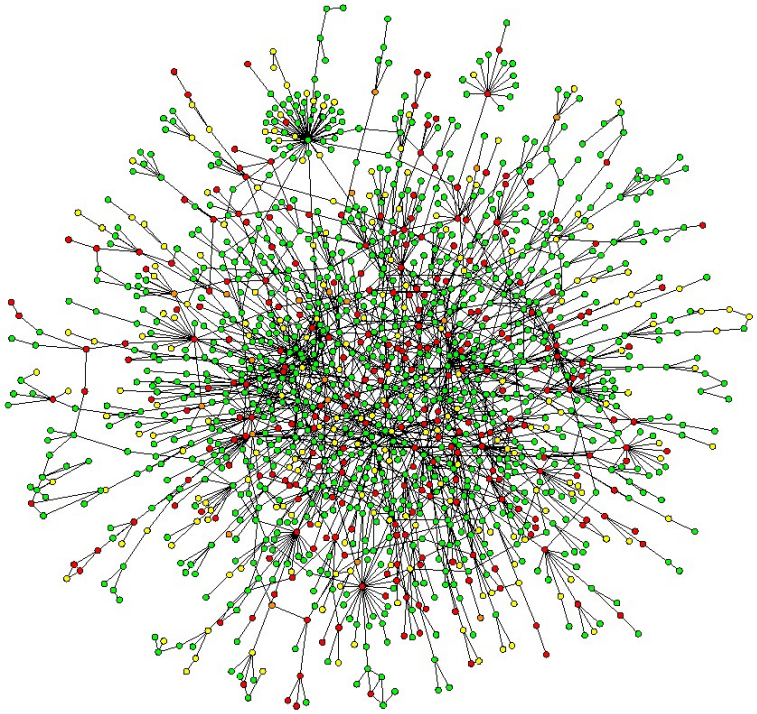
- Machine learning (ML) - hier werden Referenzdaten als Trainingsset herangezogen
- Purification Enrichment Score (PE) - dieser Score beruht auf einem statistischen Wert, ob die Häufigkeit des Auftretens dieser Interaktion innerhalb mehrerer Mes-

sungen statistisch signifikant ist

Diese Wahrscheinlichkeiten dienen im Graph dann als Kantengewichte zwischen den einzelnen Proteinen (Knoten). Aussagen über die Güte eines Graphen lassen sich durch Maße wie die **Genauigkeit** (accuracy) und die **Abdeckung** (coverage) treffen.

Um innerhalb eines Netzwerkes Proteinkomplexe zu identifizieren, gibt es verschiedene Möglichkeiten, eine davon ist die Anwendung von Clusteralgorithmen, wie z.B. Markov Clustering, hierarchisches Clustering oder Restricted Neighborhood Search Clustering. Nachteil beim Anwenden von gewöhnlichen Cluster-Verfahren ist jedoch zum einen, dass Proteine, die in der Zelle durchaus als Bestandteil von mehreren Komplexen auftreten können, nur einem Cluster zugeteilt werden und des Weiteren, dass Komplexisoformen nicht als solche erkannt werden und dadurch mehrfach auftreten. Jedoch kann man durch weitere Strukturierung der Komplexe oder zusätzliche Zuteilung von Proteinen, die mehrfach interagieren, diese Probleme lösen.

Ein Beispiel für ein relativ gut untersuchtes Protein-Interaktionsnetzwerk ist das des Mikroorganismus *S. cerevisiae*:



Man erkennt mehrere, blumenstraußartige Zentren, in denen ein zentrales Protein an einer Anzahl an Interaktionen beteiligt ist. Diese Proteine nennt man **Hubs**. Grün eingefärbte Proteine kennzeichnen hier solche, deren Gendeletion das Zellwachstum nicht beeinträchtigt. Rot gefärbte Proteine sind solche, deren Gendeletion für die Zelle tödlich

ist und gelb gefärbte Proteine stellen solche dar, deren Status nicht bekannt ist.
Eine statistische Auswertung ergab, dass die Deletion von stark interagierenden Proteinen (Hubs) deutlich öfter zum Zelltod führt als die Deletion von Proteinen mit wenigen Interaktionen.

8.6.3 Beispiele verschiedener Interaktionen

Wie bereits erwähnt, treten Interaktionen nicht nur zwischen Proteinen selbst auf, sondern auch zwischen Proteinen und DNA. Die Kontrolle der Genexpression in Prokaryoten ist hierfür ein gutes Beispiel.

Protein-DNA Interaktion bei Prokaryoten
Bakterielle Zellen stehen in direktem Kontakt mit der Umgebung, die sich sehr schnell verändern kann. Transferiert man eine bakterielle Zellkultur von Minimalmedium in ein Medium mit Laktose oder Tryptophan, führt dies zu einer rapiden Anpassung des Metabolismus der Bakterienkultur.
Laktose ist ein Disaccharid bestehend aus Glukose und Galaktose, welches über eine α-1,4-glykosidische Bindung verknüpft ist. Die Hydrolyse dieser Bindung mithilfe des Enzyms β-Galaktosidase versorgt die Bakterienzelle mit metabolischen Zwischeneprodukten und Energie. Jedoch wird von der Zelle nur dann Laktose als Kohlenstoffquelle genutzt, wenn keine effizientere Energiequelle wie Glukose zur Verfügung steht.
Durch Überführung der Bakterienkultur von Minimalmedium in Laktose-haltiges Medium befinden sich innerhalb von Minuten mehr als die tausendfache Anzahl an β-Galaktosidase Molekülen in der Zelle, das Vorhandensein von Laktose hat also die Synthese des Enzyms induziert.

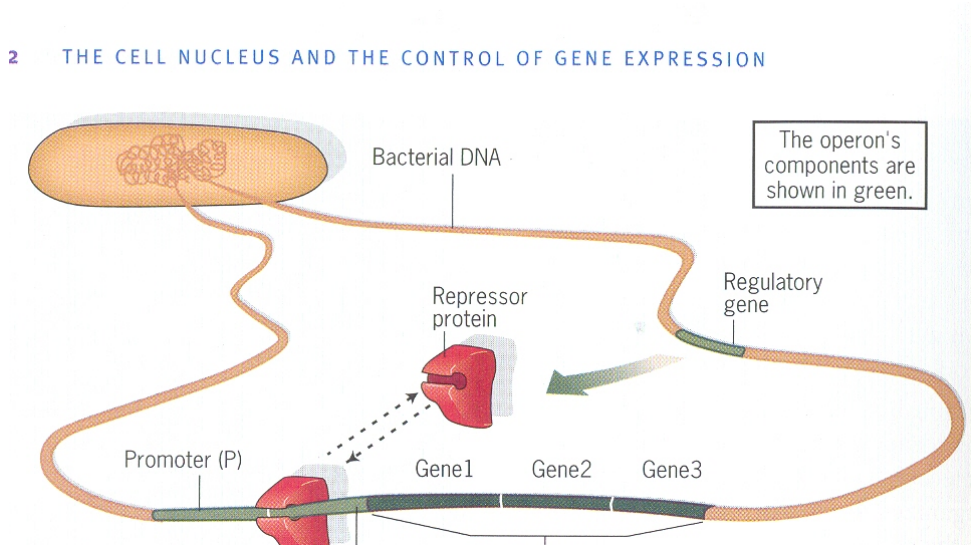
In Bakterien liegen Gene, die für Enzyme eines metabolischen Pfades codieren, üblicherweise räumlich benachbart in einem funktionellen Komplex, dem sog. **Operon**. Alle Gene eines Operon werden koordiniert reguliert nach einem Mechanismus, der zuerst von F. Jacob & J. Monod (1961) beschrieben wurde.

Der typische Aufbau eines Operon besteht aus:

- **Strukturelle Gene**
  Strukturelle Gene codieren für die Enzyme selbst. Diese Gene liegen meist benachbart zueinander, wodurch bei der Transkription mittels RNA Polymerase alle Gene in eine einzige mRNA transkribiert werden und folglich dann in verschiedene, individuelle Enzyme translatiert werden. Dies bedeutet, dass bei der Aktivierung eines Gens für ein Enzym alle Gene in diesem Abschnitt des Operons aktiviert werden.
- **Promotor Region**
  Die *Promotor-Region* dient als Bindungsstelle der RNA Polymerase
- **Operator**
  Der *Operator* beschreibt eine Region, die benachbart oder überlappend zur Promotor-Region liegt und als Bindungsstelle für ein Protein dient, dem sog. **Repressor**, der

eine genregulierende Funktion besitzt, indem er mit hoher Affinität an eine bestimmte Basensequenz innerhalb des Operators bindet.

- **regulatorische Genen**
  Die *regulatorischen Gene* codieren für das Repressorprotein.



(aus dem Buch „Cell and molecular biology“ von Gerald Karp)

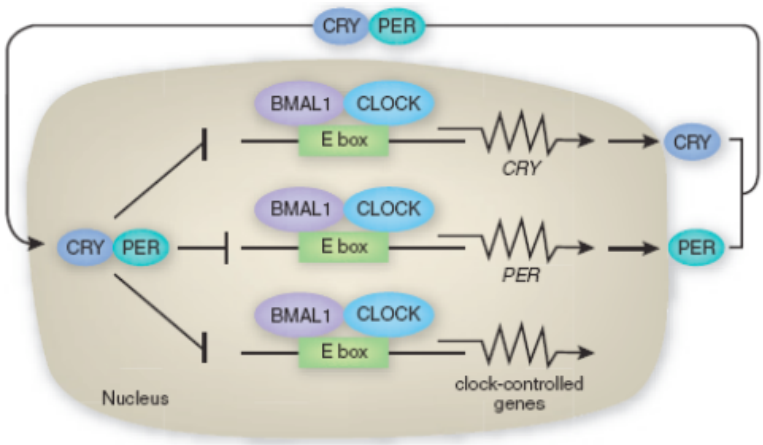
Wie anhand des Operators deutlich wird, gibt es Bereiche, die der Regulation eines Operon dienen, welche sowohl positiv (*induzierbar*) als auch negativ (*reprimierbar*) erfolgen kann.
Ein sehr bekanntes Beispiel für ein **induzierbares Operon** ist das *lac Operon*, bei dem wie bereits erläutert, Laktose selbst als sog **Inducer** fungiert, indem es an das Repressorprotein bindet und dadurch die Bindung des Repressors an die DNA verhindert wird. D.h. bei hoher Laktose-Konzentration wird das Operon induziert und β-Galaktosidase gebildet. Durch den metabolischen Abbau der Laktose sinkt deren Konzentration, wodurch die Inducer-Moleküle wieder vom Repressor dissoziieren und dieser wieder an die DNA binden und somit die Transkription inhibieren kann.

Ein **reprimierbares Operon** ist z.B. das *trp Operon* (Tryptophan-Operon), bei dem der Repressor allein nicht an den Operator binden kann und somit die Strukturgene aktiv transkribiert werden. Tryptophan agiert als sog. *Corepressor*, indem es an den inaktiven Repressor bindet und dessen Konformation so ändert, dass dieser an den Operator binden kann und die Tryptophan-Synthese inhibiert wird.
Ist die Tryptophan Konzentration also hoch (z.B. als Zusatz in einem Medium), wird das Operon reprimiert, um eine Tryptophanüberproduktion zu verhindern. Liegt jedoch nur eine geringe Konzentration vor, fehlt den meisten Repressormolekülen der Corepressor und Tryptophan wird gebildet.

Protein-DNA/Protein-Protein Interaktion bei Säugern
Es gibt viele Beispiele für DNA-Protein Interaktionen bei Eukaryoten, hier werden wir auf einen Spezialfall eingehen, den zirkadianen Rhythmus bei Säugern (einfach gesprochen die „innere Uhr“).

Organismen steuern ihre Entwicklung in optimaler Weise, indem diese an die täglichen Veränderungen der Umgebung mithilfe molekularer Zeitgeber, sog. **zirkadianer Uhren** angepasst wird.
So zeigen Säuger zirkadiane Rhythmen im Verhalten oder bei physiologischen Vorgängen wie z.B. dem Schlaf, Blutdruck oder Metabolismus. Diese werden mithilfe intrinsisch zentraler und peripherer molekularer Uhren durch externe Lichtsignale gesteuert. Zirkadiane Rhythmen zeichnen sich als Teilmenge biologischer Rhythmen dadurch aus, dass sie endogen erzeugt und selbsterhaltend sind und unter konstanten Umgebungsbedingungen (konstante Helligkeit oder Dunkelheit) sowie konstanter Temperatur bestehen bleiben. Unter solchen, kontrollierten Bedingungen ergibt sich eine Periode von 24 Stunden, wobei dieser Zeitraum innerhalb eines Bereichs von Umgebungstemperaturen konstant bleibt. Es wird vermutet, dass diese Fähigkeit dazu beiträgt, die innere Uhr gegen Veränderungen im zellulären Metabolismus zu schützen.

Biologische Uhren beinhalten 3 wichtige Elemente, erstens einen **zentralen Oszillator**, der den Takt angibt, zweitens die Fähigkeit, durch **Zurücksetzung der Uhr** auf zeitliche Reize der Umgebung zu reagieren und drittens beinhalten biologische Uhren eine **Reihe von Ausgaben**, die an bestimmte Phasen des Oszillators gebunden sind, um die Aktivität und Physiologie zu steuern.
Die zirkadiane Uhr bei Säugetieren befindet sich im *Nucleus suprachiasmaticus* (SCN), eine kleine Region des Gehirns, die rhythmische Ausgaben, bestehend aus einer Vielzahl von neuronalen und hormonellen Signalen, produziert. Die wichtigste Aufgabe dieser Signale ist es, die peripheren Uhren zu stellen. Um die Genexpression mit zirkadianer Periodizität auszuführen, benötigt die zirkadiane Uhr mehrere verbindende, transkriptionale, translationale und post-translationale Schleifen. Einen minimalen Überblick gibt die folgende Abbildung:



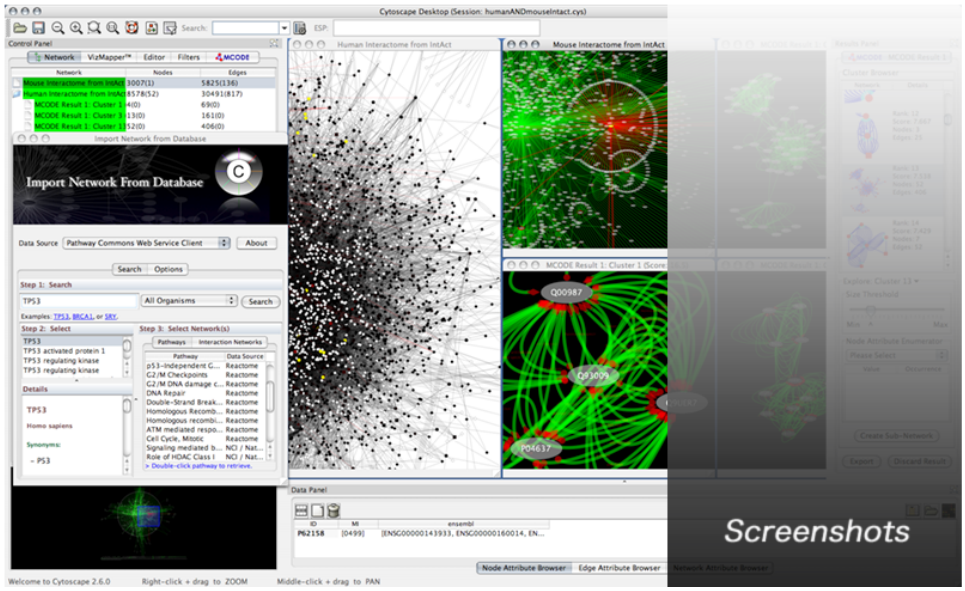
(„The intelligent clock and the Rube Goldberg clock“, Aziz Sancar, Nat. Struct. Mol. Biol. 15:23-24, 2008)

Die Transkriptionsfaktoren CLOCK und BMAL1 dimerisieren zu einem heterogenen Komplex. Anschließend bindet dieser BMAL1:CLOCK Komplex an die E-Boxen (Transkriptionsfaktorbindungsstellen) der Promotoren für die Gene PER, CRY und CDG, wodurch die Transkription dieser Gene beginnt. PER steht für *period*, CRY für *cryptochrome* und CDG für *clock-controlled genes*. Die PER und CRY Proteine dimerisieren, wandern wieder in den Nukleus und inhibieren die Transkription der drei zuvor genannten Genabschnitte.

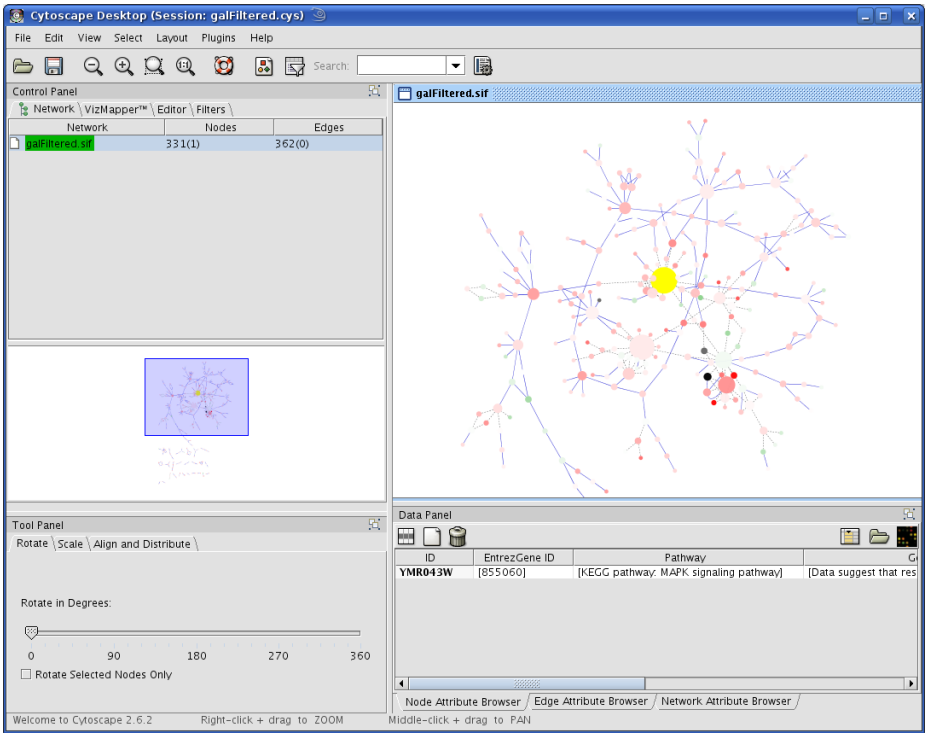
Der zirkadiane Ablauf der Genexpression ist bei weitem komplexer, als hier dargestellt, da wichtige Schritte wie Phosphorylierung als zeitliche Verzögerungen u.v.w. nicht erläutert werden. Als Beispiel für das komplexe Zusammenspiel von Proteinen und/oder DNA sei es an dieser Stelle ausreichend, für weitergehende Informationen siehe Vorlesung Biological Sequence Analysis.

8.6.4 Cytoscape - Visualisierung eines Interaktionsnetzwerkes

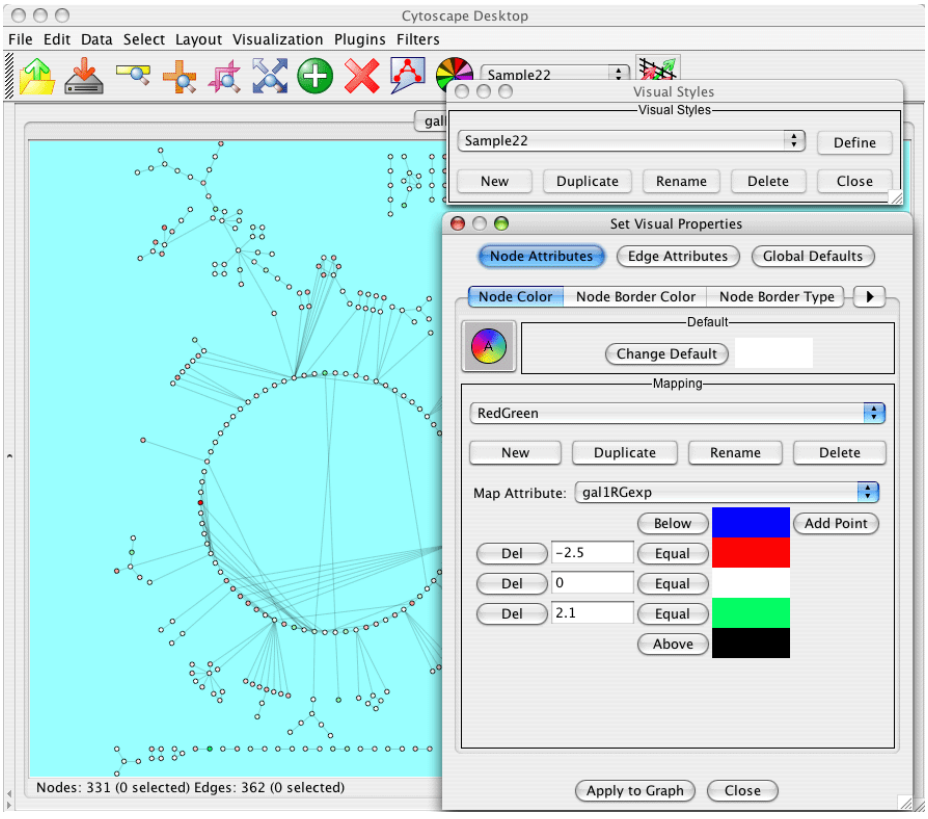
Ein bekanntes und weitverbreitetes Programm ist Cytoscape. (*http://www.cytoscape.org/*)



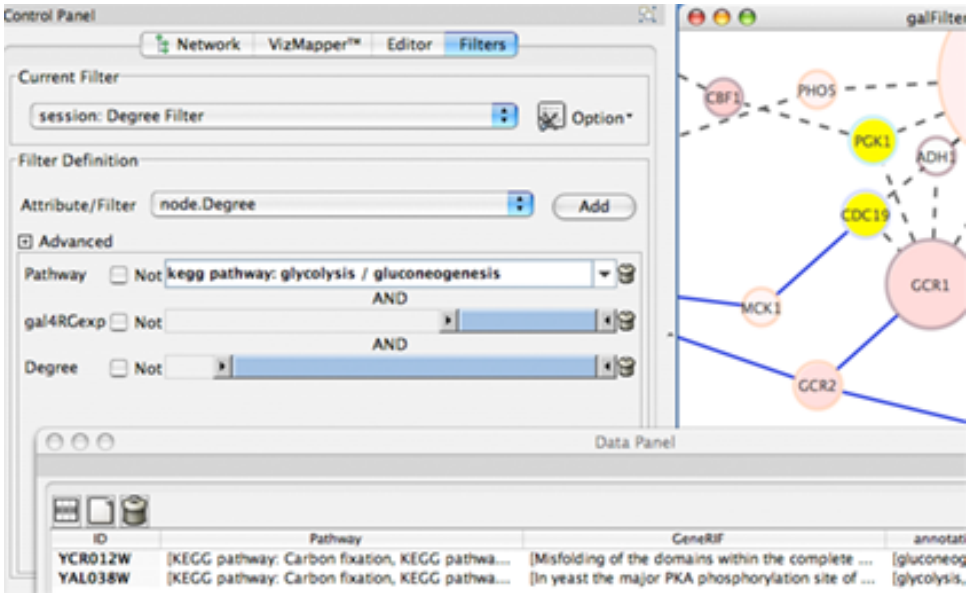
Cytoscape dient zur Visualisierung molekularer Interaktionsnetzwerke und biologischer Pfade. Die Netzwerke in Cytoscape können zudem Informationen über Genexpressionsprofile und andere Daten analysieren und visualisieren.
Im Bild ist ein Screenshot von Cytoscape 2.6.2 zu sehen. Aktuell angezeigt ist ein Netzwerk aus der mitgelieferten Beispieldatei galFiltered.sif. Es enthält ein Interaktionsnetzwerk und die Expressionsdaten aus der Veröffentlichung von T. Ideker et. al, Science, 292, 929-34 (2001). Es handelt sich dabei um den Prozess der Galaktose Aufbereitung (GAL) in der Hefe *Saccharomyces cerevisiae*. Das Netzwerk entält 331 Knoten (Gene), die über 362 Kanten miteinander verbunden sind.



Mit dem Visualisierungstool kann man sich das Netzwerk in verschiedenen Visualisierungsstilen anzeigen lassen, oder auch die Wahrscheinlichkeit, oder die Expressionsdaten, usw.

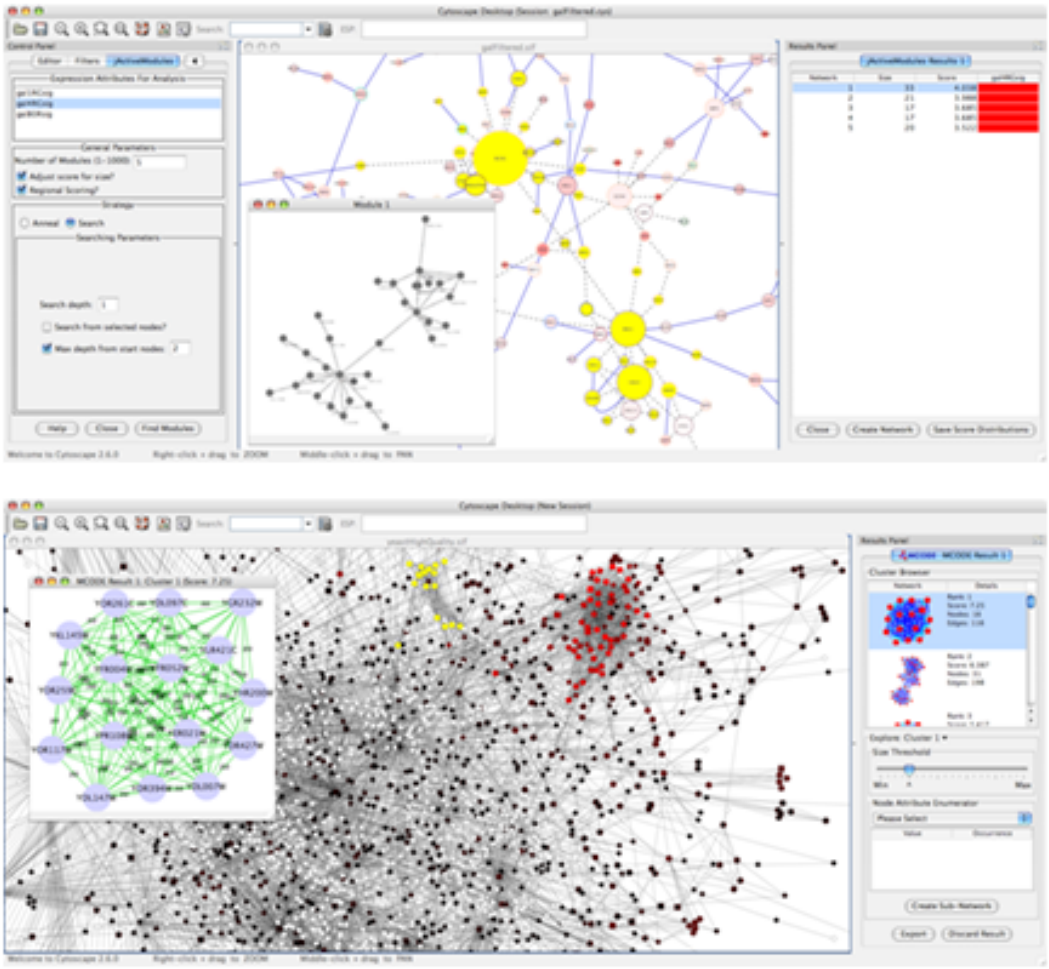


Mit Cytoscape kann das Netzwerk gefiltert werden, um sich nur einen Teil des Netzes anzuschauen, z.B Knoten mit einer bestimmten Wahrscheinlichkeit oder Knoten, die einem bestimmten Pfad angehören.



Zum Auffinden aktiver Teilnetze/Pfade wird das Netzwerk gegen Genexpressionsdaten gescreent, um miteinander verbundene Interaktionen zu identifizieren, z.B. Netze, deren Gene stark co-exprimiert sind. Die Interaktionen in den Teilnetzen lassen Rückschlüsse auf regulatorische und Signal Interaktionen durch beobachtete Veränderungen der Genexpressionen zu.

Abhängig von der Art des Netzwerkes können Cluster verschiedene Dinge beschreiben. Cluster in einem Protein-Protein Interaktions Netzwerk stehen für Proteinkomplexe und beschreiben einen Teil von Pfaden. Cluster in einem Protein-Ähnlichkeitsnetzwerkes stehen für Proteinfamilien.



Cytoscape unterstützt viele Standardnetzwerke und Dateiformate: SIF (Simple Interaction Format), GML, XGMML, BioPAX, PSI-MI, SBML, OBO, and Gene Association. Textdateien und MS Excel Tabellen werden ebenfalls unterstützt. Expressions Daten und GO Profile können ebenfalls importiert werden. Die Netzwerke können weiter bearbeitet werden, z.B. kann man die Knoten und Kanten selbst beschriften usw. Cytoscape arbeitet mit externen Datenbanken zusammen, so ist es möglich direkt Daten dieser Datenbanken in Cytoscape zu importieren. In Version 2.6.2 ist dies für die Datenbanken Pathway Commons, IntAct, BioMart, NCBI Entrez Gene und PICR möglich.

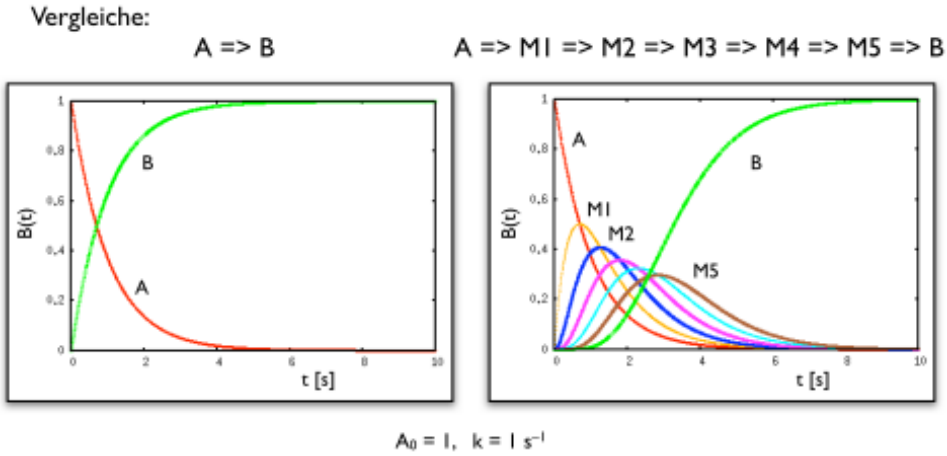
9 Differentialgleichungs-Modelle für die dynamische Simulation von biologischen Modellen

Wie im vorherigen Kapitel gezeigt wurde, geht es in der Systembiologie darum, ein möglichst umfassendes Verständnis biologischer Zellen und sogar ganzer Organismen zu entwickeln anhand einer sehr großen Menge von Daten, die auf unterschiedlichen Ebenen gewonnen wurden (Genom, Proteom, Metabolom etc.). Mithilfe dieser Information wird versucht, ein Modell zu erstellen, um letztendlich mithilfe der Simulationen Vorhersagen treffen zu können. Dies gestaltet sich jedoch oft schwierig, da biologische Systeme sehr komplex sind und so mathematische Modelle viele Variablen benötigen, um diesem detailierten Wissen gerecht zu werden.
Im Gegensatz zu dem eher qualitativen und beschreibenden Graphenmodellen können Reaktionsgleichungssysteme die zeitabhängigen Abläufe auch in komplizierten Netzwerken auf quantitative Weise beschreiben.
Die grundlegenden Schritte einer Systemanalyse bis hin zur Simulation sind folgendermaßen gegliedert:

1. Erstellung des biologischen Netzwerks
2. Identifikation der Metabolite inklusive ihrer Konzentrationen
3. Aufstellen der Einzelreaktionen
4. Übersetzung der Reaktionen in Differentialgleichungen oder Boolsche und Petri-Netze
5. Bestimmung der Anfangswerte und Simulation
6. Analyse der Ergebnisse

Qualitative chemische Modelle basieren auf der *Reaktionskinetik*, d.h. der Analyse des zeitlichen Ablaufs chemischer Reaktionen mit Differentialgleichungen. Logische Modelle hingegen werden mithilfe *Boolescher Netze* oder *Petri-Netzen* umgesetzt, die auf der Graphentheorie beruhen.
Weiterhin muss unterschieden werden, ob die dynamischen Simulationen ein **zeitabhängiges Verhalten** und damit Reaktionen des Systems auf Änderungen der äußeren Bedingungen betrachten oder **stationäre Zustände** beschreiben, in denen die Randbedingungen konstant sind.

Wie in Kapitel 8 gezeigt wurde, bestehen metabolische Pfade aus vielen Reaktionen. Teilweise handelt es sich um Einzelreaktionen, teilweise müssen Reaktionen jedoch auch in Elementarreaktionen gesplittet werden, um die Vorgänge genau zu verstehen und simulieren zu können. Ein Beispiel hierfür ist die bereits vorgestellte Gesamtreaktion der Umwandlung von Glukose in ATP.



Diese „Zwischenschritte“ haben große Auswirkung auf die Systemreaktionen, da sie verzögert werden und ein Weglassen solcher Zwischenschritte zu falschen Simulationsresultaten führen kann. Es ist daher nicht trivial, alle relevanten Reaktionen zu beachten und bedarf genauer Analyse mittels Experimenten.

Im Folgenden soll die Elementarreaktion, bei der die Edukte A und B zu einem Komplex AB_2 reagieren, als Beispiel für einige Definitionen und Erläuterungen dienen. Die Reaktion befindet sich im chemischen Gleichgewicht, was bedeutet, dass Hin- und Rückreaktion gleich schnell ablaufen.

$$A + 2B \Leftrightarrow AB_2$$

9.1 Erstellen der Differentialgleichung

Basis der Modellierung eines dynamischen, quantitativen Modells sind **Differentialgleichungen** (DGL), die eine mathematische Darstellung *lokaler Veränderungen* der Stoffkonzentrationen durch die angreifenden Reaktionen beschreiben. Als eine Differentialgleichung bezeichnet man in der Mathematik Gleichungen, die die (zeitliche oder räumliche) Ableitung(en) einer oder mehrerer Veränderlichen enthalten.

Die *Konzentration* eines Stoffes A entspricht der Teilchendichte, d.h. dem Verhältnis von Teilchenzahl pro Volumen:

$$Teilchendichte = \frac{Teilchenzahl\ N_A}{Volumen\ V}$$

Um z.B. die Veränderung der Konzentrationen der an der Reaktion beteiligten Edukte A und B und des Produkts AB_2 zu ermitteln, müssen wir zum Einen die **Assoziation der Edukte** und zum Anderen die **Dissoziation des Komplexes** betrachten.

Die Assoziation beschreibt die Wahrscheinlichkeit, dass A und B sich finden und reagieren. Da die Konzentration gleich der Teilchenzahl pro Volumen ist, steigt die Assoziationsrate proportional zu den Dichten der Reaktanden A **und** B. Die Dissoziation hingegen beschreibt die Wahrscheinlichkeit, dass der Komplex AB_2 aufbricht, d.h. die Dissoziationsrate ist proportional zur Dichte von AB_2.

Die **Dissoziation** des Komplexes in die Einzelstoffe ist im Hinblick auf die Konzentration von A (bzw. B) ein Gewinn (engl. *gain*) (da sich die Konzentration erhöht) und wird mit G_A bzw. G_B dargestellt:

$$G_A = k_d[AB_2] \qquad G_B = 2k_d[AB_2] \tag{9.1.1}$$

D.h. der Konzentrationsgewinn von A hängt vom Produkt des phänomenologischen Faktors k_d mit der Konzentration des Komplexes AB_2 ab. Für B gilt Ähnliches, wobei der stöchiometrische Faktor 2 miteinbezogen werden muss, da durch den Zerfall des Komplexes zwei B - Moleküle freigesetzt werden. Der phänomenologische Faktor k_d steht für die **D**issoziationskonstante und kann in der Literatur auch als k_r beschrieben werden (r = reverse), wobei diese Nomenklatur davon abhängig ist, wie die Richtung der Reaktion definiert ist.

Die **Assoziation** der Edukte A und B zum Komplex AB_2 kommt einem Verlust (engl. *loss*) der Konzentrationen der Edukte gleich und wird durch den Term L_A bzw. L_B beschrieben:

$$L_A = k_a[A][B][B] = k_a[A][B]^2 \qquad L_B = 2k_a[A][B]^2 \tag{9.1.2}$$

Der Konzentrationsverlust von A bzw. B durch Assoziation entspricht also dem Produkt des phänomenologischen Faktors k_a und den Konzentrationen der an der Assoziation beteiligten Edukte A und $2B$. k_a steht hier für die **A**ssoziationskonstante und kann in der Literatur auch als k_f erscheinen (f = forward).

Die Differentialgleichung für die zeitliche Änderung der Konzentration eines Stoffes X lässt sich mit den bereits eingeführten Begriffen nun wie folgt herleiten:

$$\frac{d}{dt}[X] = G_X - L_X \tag{9.1.3}$$

Für A ergibt sich daher:

$$\frac{d}{dt}[A] = G_A - L_A = k_d[AB_2] - k_a[A][B]^2 \tag{9.1.4}$$

Für B ergibt sich daher:

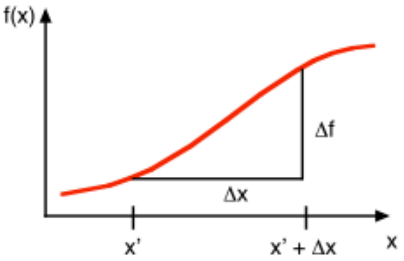
$$\frac{d}{dt}[B] = G_B - L_B = 2(k_d[AB_2] - k_a[A][B]^2) = 2\frac{d}{dt}[A] \tag{9.1.5}$$

Für den Komplex AB_2 ergibt sich eine Konzentrationsänderung, die der negativen Veränderung von $[A]$ entspricht, da der Gewinn G der Assoziation und Verlust V der Dissoziation entspricht:

$$\frac{d}{dt}[AB_2] = G_{AB_2} - L_{AB_2} = k_a[A][B]^2 - k_d[AB_2] = -\frac{d}{dt}[A] \tag{9.1.6}$$

Um eine vollständige Beschreibung einer zeitlichen Veränderung zu erhalten, müssen zusätzlich auch die Anfangskonzentrationen der Stoffe zum Zeitpunkt t_0 berücksichtigt werden.

Die zeitliche Veränderung der Konzentration ist grafisch in der Abbildung rechts zu sehen (rote Kurve). Die y-Achse $f(x)$ stellt die Konzentration in Abhängigkeit von der Zeit dar, die x-Achse steht für die Zeit. Die Definition des Differentialquotienten lautet:



$$\frac{df}{dx}(x') = \lim_{\Delta x \to 0} \frac{f(x' + \Delta x) - f(x')}{\Delta x} \tag{9.1.7}$$

mit der genäherten Lösung:

$$f(x' + \Delta x) = f(x') + \frac{df}{dx}(x')\Delta x \tag{9.1.8}$$

Angewandt auf die Konzentrationsveränderung innerhalb des Zeitintervalls Δt ergibt sich daraus für Stoff A:

$$[A](t + \Delta t) = (k_d[AB_2](t) - k_a[A](t)[B]^2(t))\Delta t \tag{9.1.9}$$

9.1.1 Bestimmung der Simulationsschrittweite

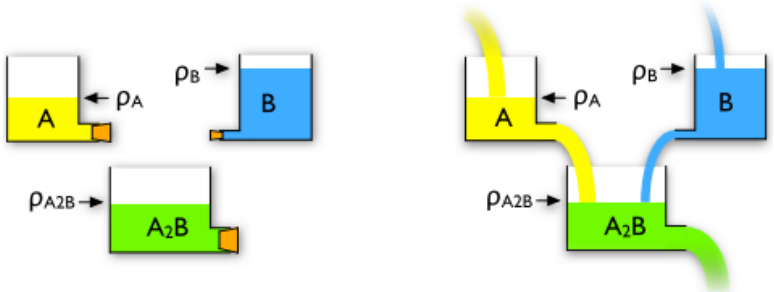
Nachdem nun die mathematischen Grundlagen für eine Simulation erläutert wurden, muss ein weiterer Aspekt dieser Differentialgleichungen beachtet werden. Da es hier um dynamische Simulation geht, ist die richtige **Auswahl der Schrittweite** von entscheidender Bedeutung für die korrekte Modellierung. Wählt man eine zu kleine Schrittweite, ergeben sich eine lange Laufzeit sowie eine Akkumulation von vielen kleinen Rundungsfehlern, bei zu großer Schrittweite hingegen falsche Ergebnisse, da viele zwischenzeitliche Ereignisse nicht beachtet werden.
Um eine Lösung für dies Problem zu finden, kann man mittels verschiedener Methoden der numerischen Integration einen näherungsweisen Wert für die Schrittweite ermitteln, wobei hier nur sog. *Einschrittverfahren* aufgelistet sind:

- **Midpoint Integration**
- **Runge-Kutta Verfahren**
- **Euler Verfahren**

Jede der Methoden hat ihre Vor- und Nachteile und neben den Laufzeitunterschieden ergeben sich unterschiedliche Fehlerraten.

9.1.2 steady-state Systeme

Bis jetzt wurden lediglich Simulationen für *zeitabhängige Vorgänge* behandelt. Wie bereits erwähnt, können dynamische Simulationen jedoch auch für Prozesse im **stationären Zustand** angewandt werden. Im **steady-state** sind die Konzentrationen konstant, d.h. $\frac{d\vec{X}t}{dt} = 0$. Dies bedeutet jedoch keinesfalls, dass im Gleichgewicht alle dynamischen Prozesse zum Stillstand kommen. Im Gleichgewicht konzentrieren sich dynamische Simulationen auf die *effektiven Volumina* und *absoluten Mengen* beziehen und nicht mehr auf zeitliche Konzentrationsänderungen. Unterschieden werden muss, ob es sich um ein statisches Gleichgewicht oder ein dynamisches Gleichgewicht handelt, wobei ein dynamisches Gleichgewicht noch weiter anhand der vorherrschenden Systemgrenzen klassifiziert werden kann. Im Gegensatz zu einem statischem Gleichgewicht findet bei einem dynamischen Gleichgewicht ein stetiger Stoffffluss statt, wobei per Definition der Zufluß ins System gleich dem Abfluß aus dem System ist.
Das Bild links unten zeigt ein statisches Gleichgewicht, das recht Bild hingegen ein dynamisches Gleichgewicht.

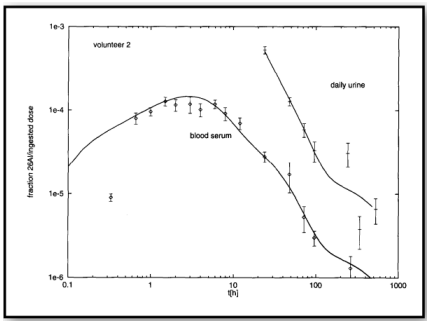


9.1.3 Simulation von Multi-Kompartment-Modellen

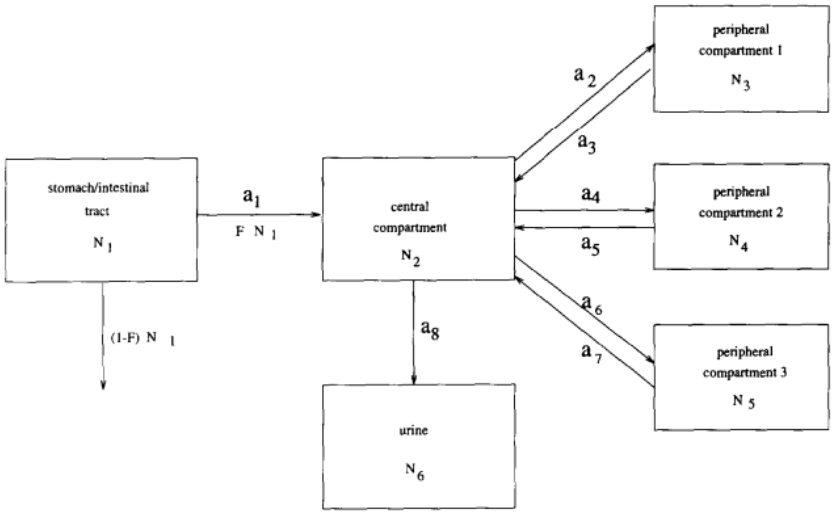
Der Bereich der Pharmakokinetik beschäftigt sich mit den Prozessen, die ein Arzneistoff innerhalb eines Organismus durchläuft, welche zusammengefasst **ADME** genannt werden und für die Vorgänge **A**dsorption, **D**istribution, **M**etabolismus und **E**xkretion stehen. Da

ein Arzneistoff meistens oral oder i.v. verabreicht wird, muss sich der Wirkstoff erst im Körper bis hin zum Zielorgan verteilen, d.h. er passiert mehrere Gewebe. Um solche Prozesse simulieren zu können, sind mehr als ein Kompartiment nötig, wodurch sich der Begriff **multi compartment model** etabliert hat. Als ein Beispiel hierfür betrachten wir den Aluminium-Metabolismus, der experimentell über 46 Tage an zwei Probanden über die Aufnahme und Ausscheidung des Stoffs untersucht wurde (Hohl et al., Nucl. Inst. Meth. B92 (1994) 478).

Hierzu wurden 100 ng des Isotops ^{26}Al an die Testpersonen oral verabreicht und in bestimmten Zeitabständen Proben in Form von Blut sowie Tagesurin genommen. Die ^{26}Al-Menge wurde dann mithilfe des AMS (Accelerator Mass Spectrometry) ermittelt, wie in der Abbildung zu sehen ist.



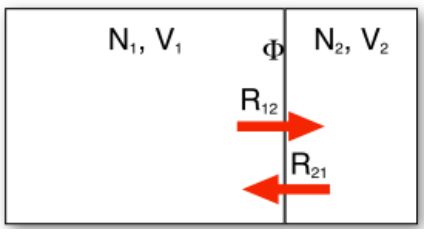
Zur Modellierung des Metabolismus muss zuerst die Verteilung im Körper nachvollzogen werden. Der Stoff wurde oral verabreicht und gelangt so durch den Magen und den Verdauungstrakt ins Blut. Von dort aus verteilt er sich in das umliegende Gewebe und die Organe, wobei sich ein dynamisches Gleichgewicht zwischen dem Blut und den periphären Gewebespeichern einstellt. Abschließend wird der Stoff über die Leber/Niere durch den Urin ausgeschieden.



Hohl, …, Nolte, Ittel, *Nucl. Inst. Meth.* B **92** (1994) 478

Die Übergangsraten(a_1, ..., a_7) zwischen den Kompartimenten wurden durch Parameterfit aus der Simulation ermittelt. Weiterhin zu beachten ist, dass die Kompartimente unterschiedlich groß sind und sich somit die Teilchenanzahl N_x unterscheidet.

Für den Teilchenaustausch zwischen zwei Kompartimenten mit den Variablen N_1, V_1 bzw. N_2, V_2 und den Übergangsraten R_{12} bzw. R_{21} durch eine Membran (Interface) mit der Fläche Φ hindurch ergibt sich folgende Berechnungsformel:



$$\frac{dN_{12}}{dt} = k_{12}\Phi\frac{N_1}{V_1} \qquad \frac{dN_{21}}{dt} = k_{21}\Phi\frac{N_2}{V_12}$$

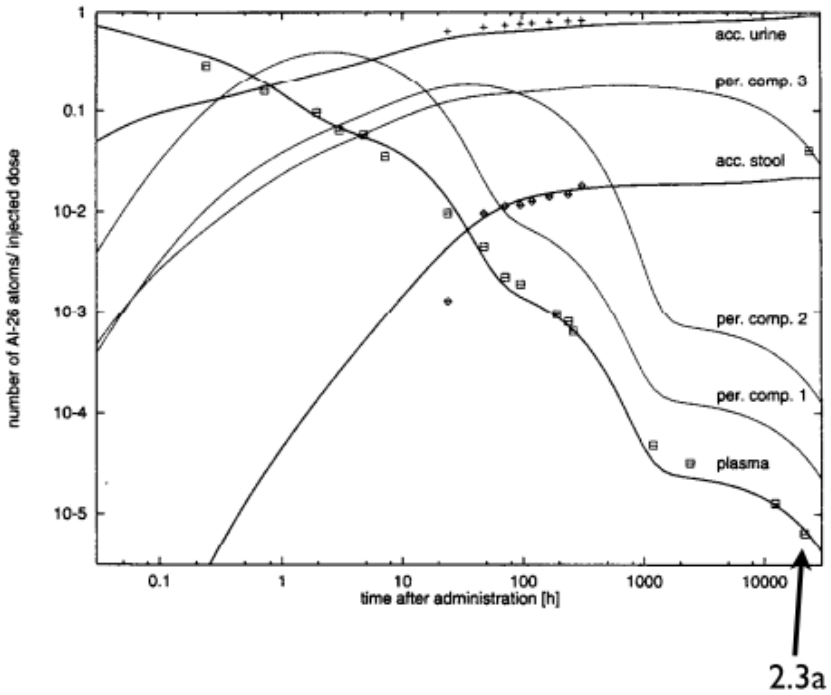
Die Änderungen der Teilchenanzahlen N_1, N_2 und damit der entsprechenden Dichten durch den Stoffaustausch ergeben sich daraus auf folgende Weise:

$$\frac{dN_1}{dt} = \frac{dN_{12}}{dt} + \frac{dN_{21}}{dt} \qquad \frac{dN_2}{dt} = -\frac{dN_1}{dt}$$

Die daraus resultierenden Dichten lassen sich berechnen durch:

$$\frac{d}{dt}\frac{N_1}{V_1} = \frac{1}{V_1}\frac{dN_1}{dt} = \frac{\tilde{k}_{21}}{V_1}\frac{N_2}{V_2} - \frac{\tilde{k}_{12}}{V_1}\frac{N_1}{V_1} \qquad \frac{d}{dt}\frac{N_2}{V_2} = \frac{V_1}{V_2}\frac{d}{dt}\frac{N_1}{V_1}$$

Die resultierenden Ergebnisse zeigten, dass drei Kompartimente für Gewebetypen mit unterschiedlichen Eigenschaften hinsichtlich des Volumens und der Austauschraten reichten, um ein zeitabhängiges Verhalten simulieren zu können, welches mit den ermittelten Messwerten übereinstimmte.



Hinsichtlich des Al-Metabolismus wurde durch freiwillige Proben gezeigt, dass Al noch nach mehr als 2 Jahren im Blut nachgewiesen werden konnte aufgrund der Speicherung in den Knochen.

Zusammenfassend halten wir fest, dass die Interpretation zeitabhängiger Daten für zelluläre Prozesse sehr komplex werden kann und somit ohne Simulationstools nicht mehr möglich ist. Eines dieser Programme, COPASI, wird in diesem Kapitel noch erläutert werden.

9.2 Enzymkinetik

Die meisten biochemischem Reaktionen in biologischen Zellen werden von Enzymen katalysiert, da ohne diese Enzyme die Reaktionen nur in nicht wahrnehmbarem Umfang ablaufen würden. Die Beteiligung der Enzyme beschleunigt die Reaktionen durch Erniedrigung der Gibbschen freien Aktivierungsenergie $\Delta G^{\ddagger}$, die als Barriere verstanden werden kann. Um diese Effekte innerhalb einer Simulation korrekt beschreiben zu können, muss ein Verständnis der Enzymkinetik vorliegen.
Ein einfaches und für viele Enzyme gültiges Modell zur Erklärung der kinetischen Eigenschaften eines Enzyms ist das sog. **Michaelis-Menten Modell** (MM Modell), welches 1913 basierend auf der Arbeit von Victor Henri bekannt wurde. Es ist ein einfaches Modell, bei dem die enzymatische Reaktion mit folgender Gleichung dargestellt wird:

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_3}{\rightarrow} E + P$$

Enzym E und Substrat S reagieren hier mit der Geschwindigkeitskonstante k_1 zum Enzymsubstratkomplex ES. Dieser Komplex kann entweder wieder in die Edukte E, S mit der Geschwindigkeitskonstanten k_2 zerfallen oder in das Enzym E und ein Produkt P mit der Geschwindigkeitskonstanten k_3 umgewandelt werden. Um eine Aussage über die Reaktionsgeschwindigkeit V treffen zu können, sind mehrere Überlegungen wichtig. Unter der Annahme, dass der Komplex nur sehr selten wieder in die Edukte E, S zerfällt, wird die Katalysegeschwindigkeit v durch die Umwandlungsreaktion in die Produkte E, P bestimmt:

$$v = k_3[ES]$$

Die Konzentration [ES] hängt, wie bereits erwähnt, von zwei Raten ab, zum einen von der Bildungsrate und zum andren von der Zerfallsrate:

$$\text{ES - Bildungsrate} = k_1[E][S] \qquad \text{ES - Zerfallsrate} = (k_2 + k_3)[ES]$$

Im *steady state* sind die Geschwindigkeiten der Bildung sowie des Zerfalls gleich, wodurch durch Gleichsetzung folgt:

$$[ES] = \frac{[E][S]}{K_M} \qquad \text{mit } K_M = \frac{k_2 + k_3}{k_1} \tag{9.2.1}$$

Die Konzentration des Enzym-Substrat-Komplexes $[ES]$ entspricht weiterhin der Gesamtkonzentration $[E_T]$ des Enzyms minus der Konzentration an freiem Enzym $[E]$:

$$[ES] = [E_T] - [E] = [E_T] - \frac{[ES] * K_M}{[S]}$$

Durch Umformen ergibt sich hieraus:

$$[E_T] = [E_S] + \frac{[ES] + K_M}{[S]} = [E_S] * \left(1 + \frac{K_M}{[S]}\right) = [ES] * \left(\frac{[S] + K_M}{[S]}\right)$$

Daraus ergibt sich abschließend:

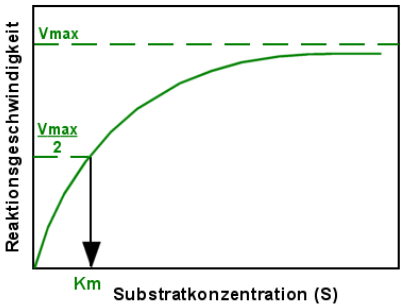
$$[ES] = [E_T] * \frac{[S]}{[S] + K_M}$$

Nachdem nun die Konzentration des Komplexes in Abhängigkeit von der Enzym- und Substratkonzentration definiert ist, kann dieser Term nun in der anfänglichen Gleichung für die Katalysegeschwindigkeit v benutzt werden:

$$v = k_3[E_T]\frac{[S]}{[S] + K_M}$$

Der Term $k_3[E_T]$ entspricht dann der maximalen Katalysegeschwindigkeit v_{max}, wenn alle Bindungsstellen des Enzyms mit Substrat gesättigt sind, was gleichbedeutend damit ist, dass die Substratkonzentration [S] sehr groß ist und der Bruch der vorherigen Gleichung sich damit 1 nähert.

Hierdurch ergibt sich endgültig die Michael-Menten Gleichung. Die Michaelis-Menten Konstante K_M entspricht der Substratkonzentration, bei der die Katalysegeschwindigkeit die Hälfte ihres Maximalwertes erreicht hat. Die grafische Veranschaulichung zeigt, dass die Auftragung der Reaktionsgeschwindigkeit v gegen die Substratkonzentration [S] eine hyperbolische Kurve beschreibt.



$$v = v_{max}\frac{[S]}{[S] + K_M} \tag{9.2.2}$$

Die Michaelis-Menten Kinetik ist jedoch, wie bereits erwähnt, nicht für alle Enzyme gültig. Sie besitzt Vorteile als analytische Formel für einfache Systeme und kann leicht anhand der Kennlinie interpretiert werden. Jedoch liefert die Michaelis-Menten Gleichung weniger kinetische Informationen als eine explizite Modellierung und liefert bei dynamischem Verhalten falsche Ergebnisse. Weiterhin ist die Michaelis-Menten Kinetik nicht für die allosterische Hemmung gültig, worauf nun eingegangen werden soll.

9.2.1 Inhibierung von Enzymen

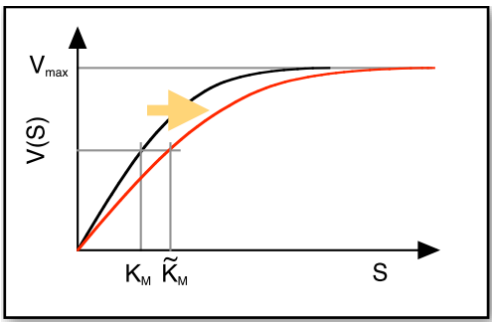
Die Inhibierung der enzymatischen Aktivität stellt eine wichtige Möglichkeit zur Kontrolle biologischer Systeme dar und wird nicht zuletzt in der Wirkstoffentwicklung als Angriffsziel verwendet. Die Enzym-Inhibition lässt sich grundlegend in **irreversible** und **reversible** Hemmung einteilen.
Bei der *irreversiblen Hemmung* bindet der Inhibitor kovalent oder nicht kovalent an das Enzym und verhindert so die enzymatische Aktivität.
Reversible Hemmungen lassen sich in mehrere Formen unterscheiden, grundlegend ist jedoch, dass die Dissoziation von Enzym und Inhibitor schnell erfolgt und die Inhibition somit zeitweise wieder aufgehoben ist.

Kompetitive Hemmung
Das Prinzip der kompetitiven Inhibition ist, dass das Enzym entweder mit dem Substrat oder dem Inhibitor bindet, jedoch nicht gleichzeitig. Der Inhibitor konkurriert mit dem Substrat um die Bindungsstelle im aktiven Zentrum und verdrängt das Substrat. Dadurch sinkt die Konzentration freier Enzyme für die Bindung mit dem Substrat, wodurch die maximale Katalysegeschwindigkeit v_{max} verringert wird. Durch Erhöhung der Substratkonzentration kann dieser Effekt jedoch aufgehoben werden und v_{max} bleibt unverändert.

Für die Michaelis-Menten Konstante und somit die gesamte Gleichung bedeutet dies:

$$\tilde{K}_M = K_M(1 + [I]/K_I)$$

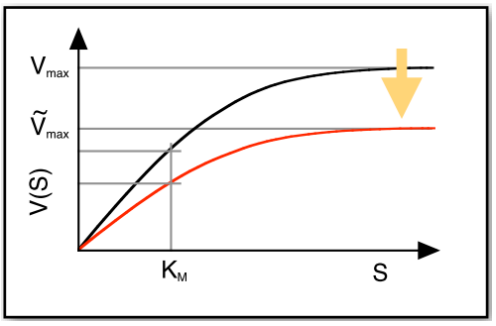
$$v = v_{max}\frac{[S]}{[S] + K_M(1 + [I]/K_I)}$$



Nichtkompetitive Hemmung
Bei der nichtkompetitiven Inhibition bindet der Inhibitor an eine Stelle, die nicht mit der Substratbindungsstelle identisch ist. Im Gegensatz zur kompetitiven Hemmung ist hier eine gleichzeitige Bindung von Substrat und Inhibitor möglich, allerdings führt die Bindung des Inhibitors dazu, dass das Enzym aufgrund einer Konformationsänderung inaktiviert wird. Die nichtkompetitive Hemmung wirkt also nicht durch Verringerung der freien Enzymkonzentration, sondern durch die Erniedrigung der *Wechselzahl* (turnover number). Unter der Wechselzahl versteht man die Anzahl von Substratmolekülen, die bei vollständiger Enzymsättigung pro Zeiteinheit in das Produkt umgewandelt werden und entspricht der kinetischen Konstante k_3 der MM-Gleichung.

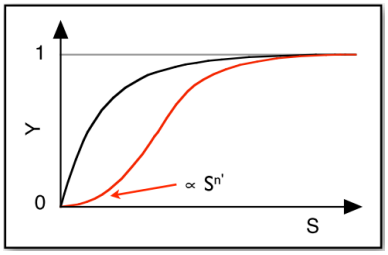
Für die Michaelis-Menten Konstante und somit die gesamte Gleichung bedeutet dies:

$$\tilde{v}_{max} = \frac{v_{max}}{1 + I/K_I}$$



Allosterische Hemmung / Kooperativität
Bei der nichtkompetitiven Inhibition bindet der Inhibitor an eine Stelle (allosterisches Zentrum), die nicht der Substratbindungsstelle entspricht. Jedoch ist hier im Gegensatz zur nichtkompetitiven Hemmung eine gleichzeitige Bindung von Substrat und Inhibitor möglich, da die Bindung des Inhibitors dazu führt, dass die Enzymkonformation so geändert wird, dass eine Substratbindung unmöglich ist.
Weiterhin unterliegen **allosterische Enzyme** nicht der Michaelis-Menten Kinetik, da ihre Auftragung der Reaktionsgeschwindigkeit v gegen die Substratkonzentration [S] einen sigmoidalen Verlauf hat und nicht einen Hyperbolischen. Dieser sigmoidale Kurvenverlauf ist ebenfalls typisch für Enzyme, deren Substrat **kooperativ** bindet (z.B. Hämoglobin), d.h. die Bindung eines Substrats führt zu einer Konformationsänderung, die die Bindung eines weiteren Substrats begünstigt. Dieser Umstand wird durch die **Hill Kinetik** beschrieben:

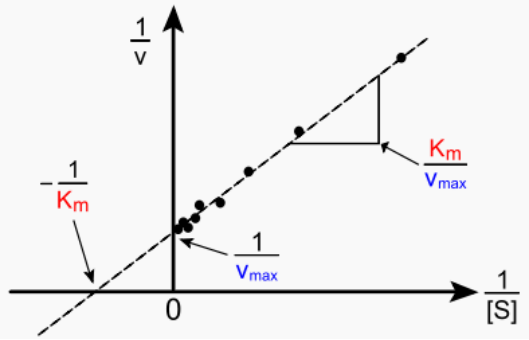
$$E + nS = ES_n$$
$$Y = \frac{S^{n'}}{S^{n'} + K^{n'}}$$



9.2.2 Bestimmung von v_{max} und K_M

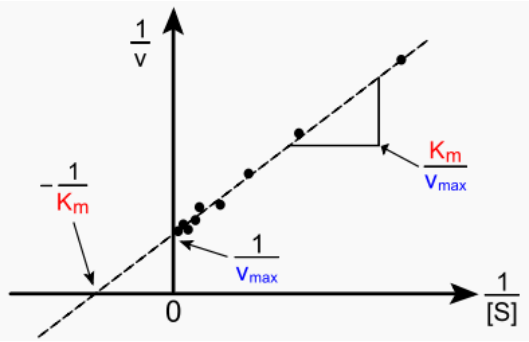
Die in diesem Kapitel bisher gezeigten grafischen Darstellungen der Michaelis-Menten Gleichung entsprechen alle einer direkt-linearen Auftragung der Katalysegeschwindigkeit v gegen die Substratkonzentration [S].

Es gibt jedoch auch Linearisierungsverfahren wie das *Lineweaver-Burk Diagramm*, welches eine doppelt-reziproke Darstellung beschreibt, bei der $1/v$ gegen $1/[S]$ aufgetragen werden. Die Gerade schneidet die x-Achse bei $-\frac{1}{K_M}$ und die y-Achse bei $\frac{1}{v_{max}}$, wodurch diese Werte direkt ersichtlich sind. Die Umformung der Michaelis-Menten Gleichung ergibt:



$$\frac{1}{v} = \frac{K_M}{v_{max}} \frac{1}{[S]} + \frac{1}{v_{max}}$$

Im sog. *Hanes-Woolf-Diagramm* wird $[S]/v$ gegen $[S]$ aufgetragen, was für die Umformung der Michaelis-Menten Gleichung folgenden Term ergibt:



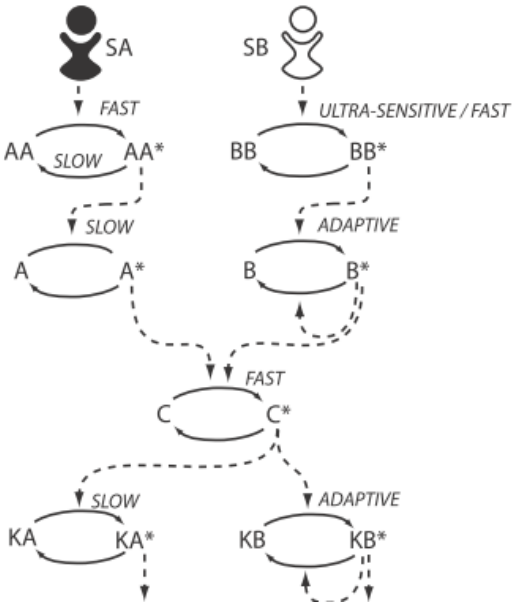
$$\frac{[S]}{v} = \frac{1}{v_{max}}[S] + \frac{K_M}{v_{max}}$$

Die grafische Darstellung von linearisierten, kinetischen Parametern ist zwar weit verbreitet und sehr deutlich, jedoch werden die Ergebnisse verfälscht, da kleine Abweichungen von v zu großen Unterschieden in der reziproken Darstellung $\frac{1}{v}$ führen. Deshalb ist davon abzuraten, diese Form der Parameteranalyse zu verwenden.

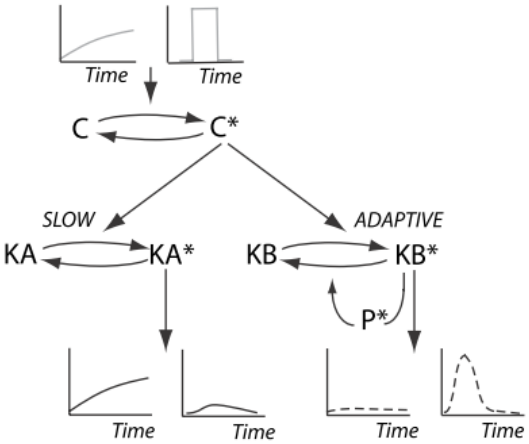
9.2.3 Fallbeispiel Michaelis-Menten Kinetik: Kinetische Isolierung von Pfaden

M. Behlar, H.G. Dohlman und T.C. Elston stellten 2007 unter dem Titel „Kinetic insulation as an effective mechanism for achieving pathway specificity in intracellular signaling networks“ ein Prinzip vor, um die spezifische Antwort intrazellulärer Pfade, die eine gemeinsame Enzymkomponente enthalten, zu erklären. In den meisten Fällen wurden die biochemischen Mechanismen, die für solche spezifische Reaktionen verantwortlich sind, durch die zusätzliche Aktivitäten weiterer Systemkomponenten zur „cross-inhibition“ oder „scaffold proteins“ erklärt. Im Gegensatz hierzu verwiesen die Arbeitsgruppe um Behlar auf ein möglicherweise unterschiedliches, zeitliches Verhalten dieser Signalsysteme, das sie **kinetic insulation** nannten. Zusammengefasst besagt dieses Prinzip, dass aufgrund des unterschiedlichen Signalverlaufs eine spezifische Antwort der gemeinsamen Komponente erfolgt und nur der zu diesem Signalverlauf gehörige Pfad reagiert.

Die Abbildung zeigt eine schematische Darstellung zweier Signalpfade A und B, die beide die gemeinsame Komponente C (hier das Enzym Kinase) enthalten. Beide Pfade bestehen aus einem Ligand SA oder SB, der an den jeweiligen Rezeptor RA oder RB bindet. Je nach Stimulus darf dann ausschließlich eine terminale Kinase (KA oder KB) aktiviert und somit nur ein Signalpfad fortgeführt werden. Im Falle von Pfad A wird durch Ligandenbindung ein Signal generiert, welches dann in ein langsames Signal umgewandelt wird und als solches an die Kinase C weitergegeben wird. Für Pfad B geschieht dies in der gleichen Weise, nur dass hier das eingehende Signal in ein schnelles, transientes umgewandelt wird um eine spezifische Antwort zu gewährleisten.



Diese Spezifität der Signalweitergabe kann erreicht werden durch unterschiedlich schnelle Aktivierungskinetiken der Kinasen KA bzw. KB laut der Autoren.
In diesem Beispiel besitzt KA eine langsame Aktivierungskinetik, d.h. die Kinase KA reagiert lediglich auf Signale, die langsam ansteigen. Um nun eine Pfadspezifität zu erreichen, darf KB nur dann aktiviert werden, wenn das eingehende Signal schnell und transient ist und muss inaktiv bleiben, wenn das Signal sich nur langsam über die Zeit hinweg verändert. Hierzu ist für KB ein adaptives System nötig, welches sich an an langsam steigende Signale kontinuierlich anpasst und sich auf das Anfangslevel zurücksetzt. Dieses Verhalten kann erreicht werden, indem das adaptive System durch eine negative feedback Hemmung reguliert wird. Hierbei phosphoryliert und aktiviert die Kinase KB die Phosphatase P, welche jedoch im Gegenzug dadurch die Kinase KB dephosphoryliert und deaktiviert. Diese feedback Schleife führt dazu, dass die Aktivierung der Kinase KB ebenfalls transient ist.



Die Kinetiken der Aktivierungs- bzw. Inaktivierungsreaktionen der einzelnen Pfadkomponenten wurde mithilfe vereinfachter Michaelis-Menten Kinetik modelliert (s. unten). Die Vereinfachung bezieht sich darauf, dass Proteinsynthese und Proteinabbau nicht beachtet wurden, weshalb die Konzentrationen normalisiert und die kinetischen Raten sowie Konstanten dementsprechend angepasst wurden.

$$\frac{d[\mathrm{AA}^*]}{dt} = \frac{k_{30} \cdot sa \cdot (1 - [\mathrm{AA}^*])}{k_{30m} + (1 - [\mathrm{AA}^*])} - \frac{V_{31}[\mathrm{AA}^*]}{k_{31m} + [\mathrm{AA}^*]} \quad [1]$$

$$\frac{d[\mathrm{A}^*]}{dt} = \frac{k_{32} \cdot [\mathrm{AA}^*] \cdot (1 - [\mathrm{A}^*])}{k_{32m} + (1 - [\mathrm{A}^*])} - \frac{V_{33}[\mathrm{A}^*]}{k_{33m} + [\mathrm{A}^*]} \quad [2]$$

$$\frac{d[\mathrm{BB}^*]}{dt} = \frac{k_{20} \cdot sb \cdot (1 - [\mathrm{BB}^*])}{k_{20m} + (1 - [\mathrm{BB}^*])} - \frac{V_{21}[\mathrm{BB}^*]}{k_{21m} + [\mathrm{BB}^*]} \quad [3]$$

$$\frac{d[\mathrm{B}^*]}{dt} = \frac{k_{22} \cdot [\mathrm{BB}^*] \cdot (1 - [\mathrm{B}^*])}{k_{22m} + (1 - [\mathrm{B}^*])} - \frac{V_{23}[\mathrm{B}^*]}{k_{23m} + [\mathrm{B}^*]} - \frac{k_{24}[\mathrm{M}^*] \cdot [\mathrm{B}^*]}{k_{24m} + [\mathrm{B}^*]} \quad [4]$$

$$\frac{d[\mathrm{M}^*]}{dt} = \frac{k_{25} \cdot [\mathrm{B}^*] \cdot (1 - [\mathrm{M}^*])}{k_{25m} + (1 - [\mathrm{M}^*])} - \frac{V_{26}[\mathrm{M}^*]}{k_{26m} + [\mathrm{M}^*]} \quad [5]$$

$$\frac{d[\mathrm{C}^*]}{dt} = \frac{k_3 \cdot [\mathrm{A}^*] \cdot (1 - [\mathrm{C}^*])}{k_{3m} + (1 - [\mathrm{C}^*])} + \frac{k_1 \cdot [\mathrm{B}^*] \cdot (1 - [\mathrm{C}^*])}{k_{1m} + (1 - [\mathrm{C}^*])} - \frac{V_2[\mathrm{C}^*]}{k_{2m} + [\mathrm{C}^*]} \quad [6]$$

$$\frac{d[\mathrm{KA}^*]}{dt} = \frac{k_9 \cdot [\mathrm{C}^*] \cdot (1 - [\mathrm{KA}^*])}{k_{9m} + (1 - [\mathrm{KA}^*])} - \frac{V_{10}[\mathrm{KA}^*]}{k_{10m} + [\mathrm{KA}^*]} \quad [7]$$

9.3 Datenbanken KEGG / SABIO-RK

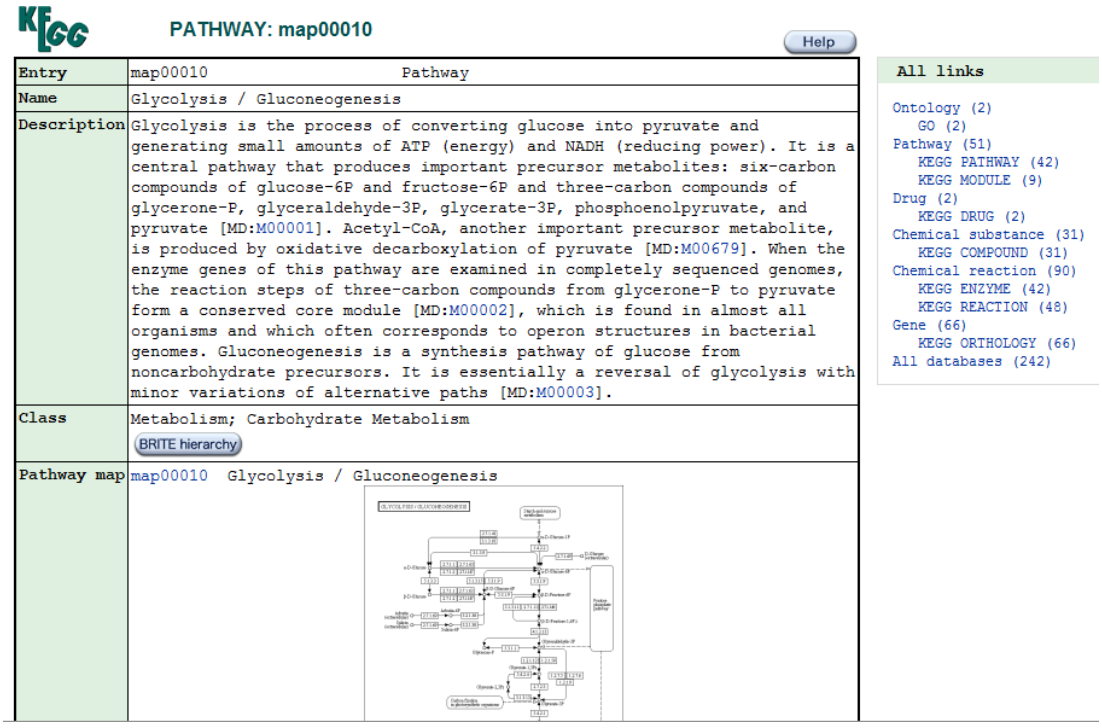
Das soeben vorgestellte Modell betrachtete nur einen ausgewählten Satz an Enzymen und deren spezielle Kinetik. Besteht jedoch der Bedarf, große metabolische Pfade in ihrer genauen Abfolge mit allen Komponenten zu analysieren, stehen verschiedene Datenbanken zur Verfügung.

Die **KEGG (Kyoto Encyclopedia of Genes and Genomes** (akt. version 54.1, Mai 2010, http://www.genome.jp/kegg/) ist eine integrierte Datenbank bestehend aus 16 Hauptdatenbanken, welche man zusammenfassend in die Bereiche

- Systeminformationen
- Genominformationen
- Chemische Informationen

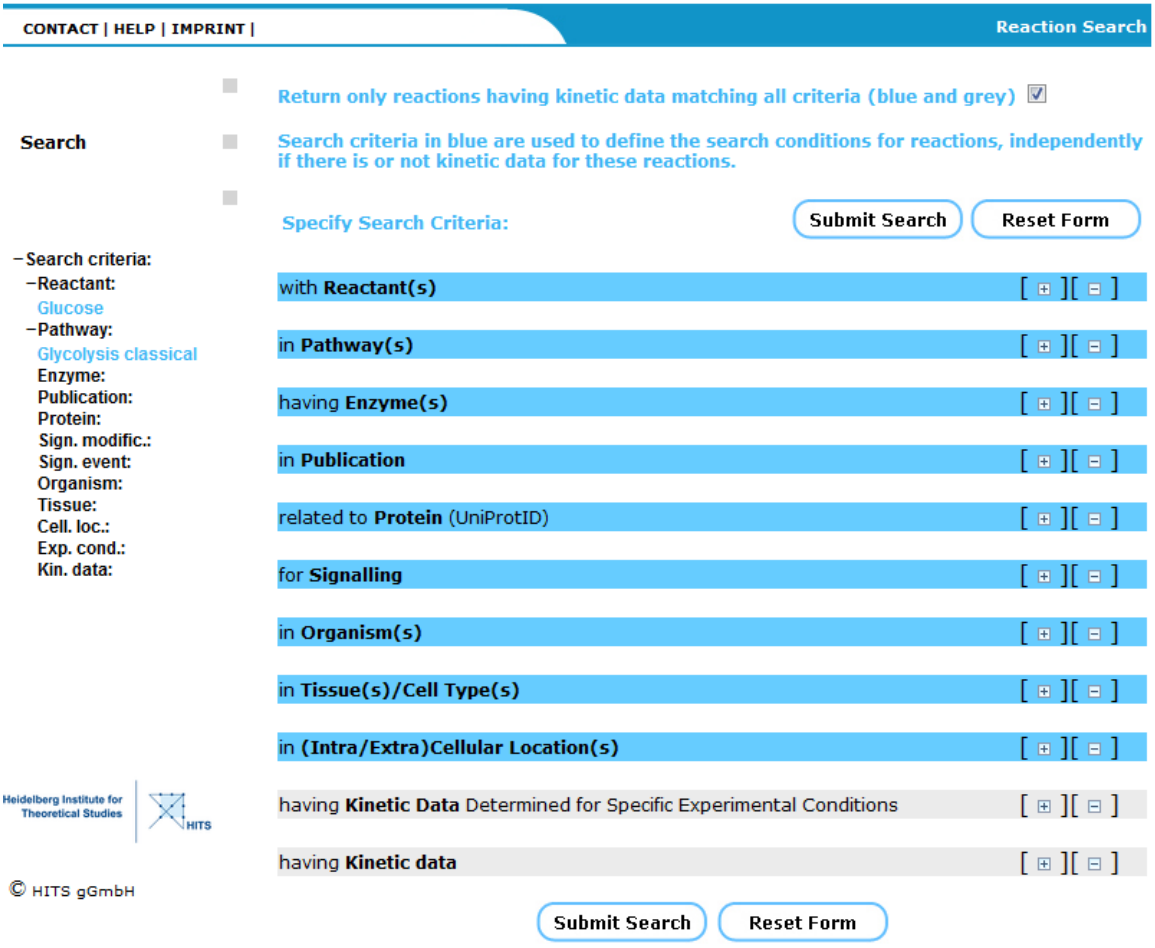
unterteilen kann. Für metabolische Pfadanalysen eignen sich besonders die Datenbanken **KEGG PATHWAY**, **KEGG MODULE** und **KEGG REACTION**. KEGG PATHWAY beinhaltet momentan 361 Pfade mit insgesamt 112000 Referenzen (Juni 2010).

Eine Suche in KEGG PATHWAY mit den Suchparametern „map“ und „Glycolysis“ ergibt ein Ergebnis von 50 gefundenen Pfadkarten, wovon der erste Treffer (map00010) eine Übersicht der gesamten Glykolyse mit allen Enzymen ist. Unter dem Menüpunkt „Pathway entry“ erhält man eine Zusammenfassung der Informationen zu dem jeweiligen Pfad, wie z.B. Name, zugehörige Klasse, Pfadkarte, Module, Referenzen und sogar einen Hinweis, an welchen Krankheiten dieser Pfad beteiligt ist.

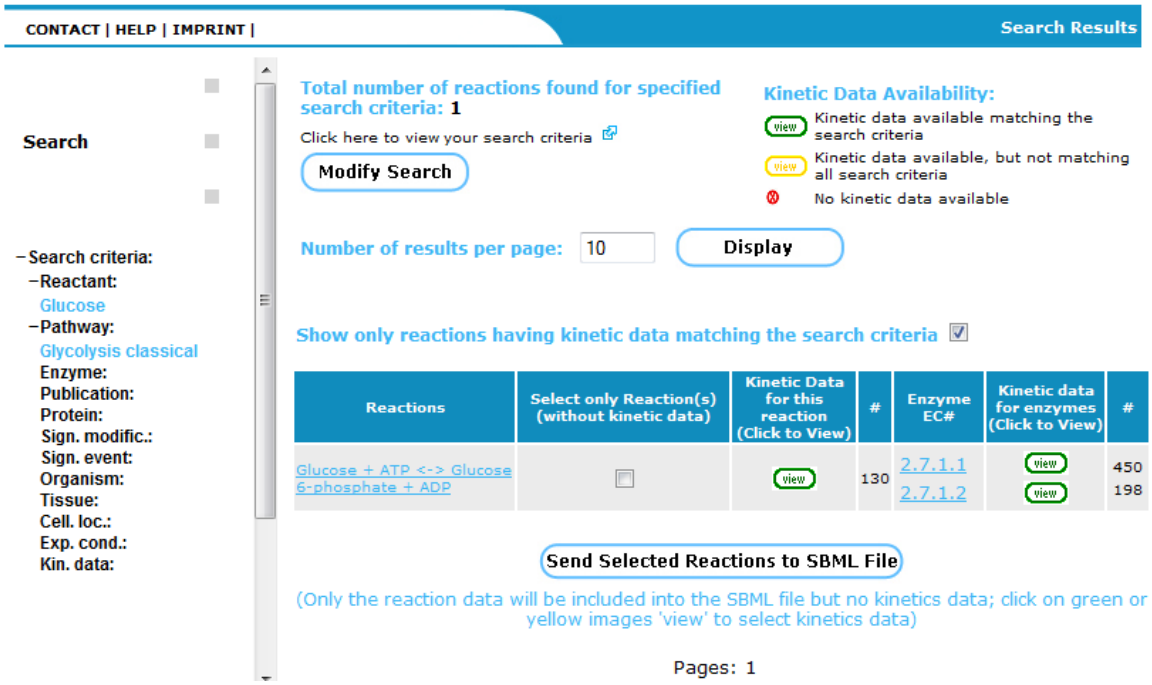


Eine weitere Datenbank, speziell für kinetische Daten, ist **SABIO-RK** (System for the Analysis of Biochemical pathways Reaction Kinetics Database) (akt. Version 1.0, Juni 2010, http://sabio.villa-bosch.de), deren Reaktionsdaten hauptsächlich aus der KEGG Datenbank sowie aus Originalarbeiten stammen.

Das Interface für die Suche in der SABIO-RK Datenbank gibt dem Benutzer die Möglichkeit, die Suche anhand verschiedener Optionen einzuschränken, indem z.B. Reaktanden, Enzyme oder der Gewebetyp spezifiziert werden können.



Eine Suche mit den Parametern „Glucose“ als Reactant-substrate und der Pathway-Spezifikation „Glycolysis“ ergibt wie erwartet einen Treffer. Neben der Reaktion und der EC Nummer der beteiligten Enzyme ist es im optimalen Fall möglich, sich die kinetischen Daten der Reaktion sowie der Enzyme anzeigen zu lassen.

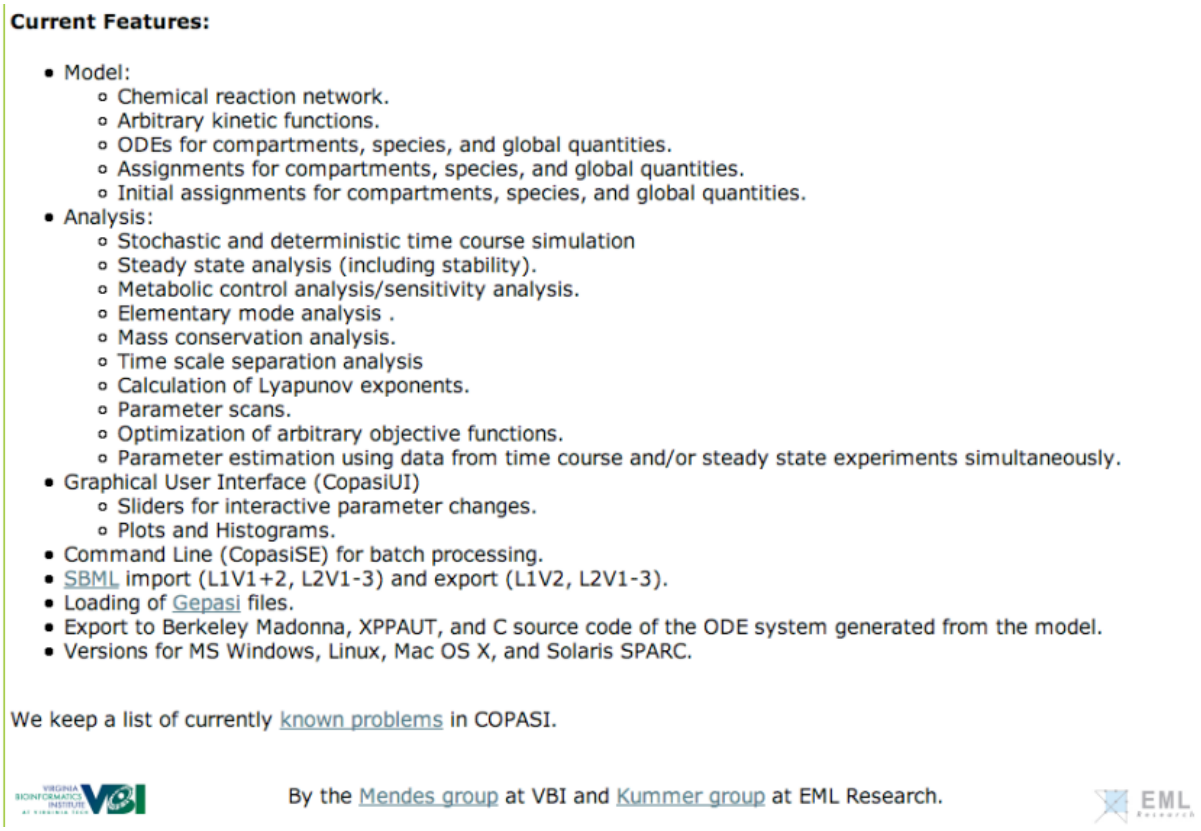


9.4 Simulationstool COPASI

Nachdem nun die theoretischen Grundlagen für eine Simulation erläutert wurden, soll an dieser Stelle nun ein Simulationstool für biochemische Netzwerke eingeführt werden.

COPASI (**Co**mplex **Pa**thway **Si**mulator) ist ein Programm für die Simulation und Analyse biochemischer Netzwerke und deren Dynamik, welches in den Gruppen von Pedro Mendes (Virginia Bioinf. Inst.) und Ursula Kummer (EML HD) entwickelt wurde (*http://www.copasi.org*). Das Programm verfügt über eine GUI und ist somit mit allen Betriebssystemen, die QT unterstützen, kompatibel (akt. Version 4.5 build 30).

Das Programm verfügt über viele Features um ein Modell zu generieren und es dann mithilfe verschiedener Analyse-Methoden auszuwerten, wie folgende Übersicht veranschaulicht.



Current Features:

- Model:
  - Chemical reaction network.
  - Arbitrary kinetic functions.
  - ODEs for compartments, species, and global quantities.
  - Assignments for compartments, species, and global quantities.
  - Initial assignments for compartments, species, and global quantities.
- Analysis:
  - Stochastic and deterministic time course simulation
  - Steady state analysis (including stability).
  - Metabolic control analysis/sensitivity analysis.
  - Elementary mode analysis .
  - Mass conservation analysis.
  - Time scale separation analysis
  - Calculation of Lyapunov exponents.
  - Parameter scans.
  - Optimization of arbitrary objective functions.
  - Parameter estimation using data from time course and/or steady state experiments simultaneously.
- Graphical User Interface (CopasiUI)
  - Sliders for interactive parameter changes.
  - Plots and Histograms.
- Command Line (CopasiSE) for batch processing.
- SBML import (L1V1+2, L2V1-3) and export (L1V2, L2V1-3).
- Loading of Gepasi files.
- Export to Berkeley Madonna, XPPAUT, and C source code of the ODE system generated from the model.
- Versions for MS Windows, Linux, Mac OS X, and Solaris SPARC.

We keep a list of currently known problems in COPASI.

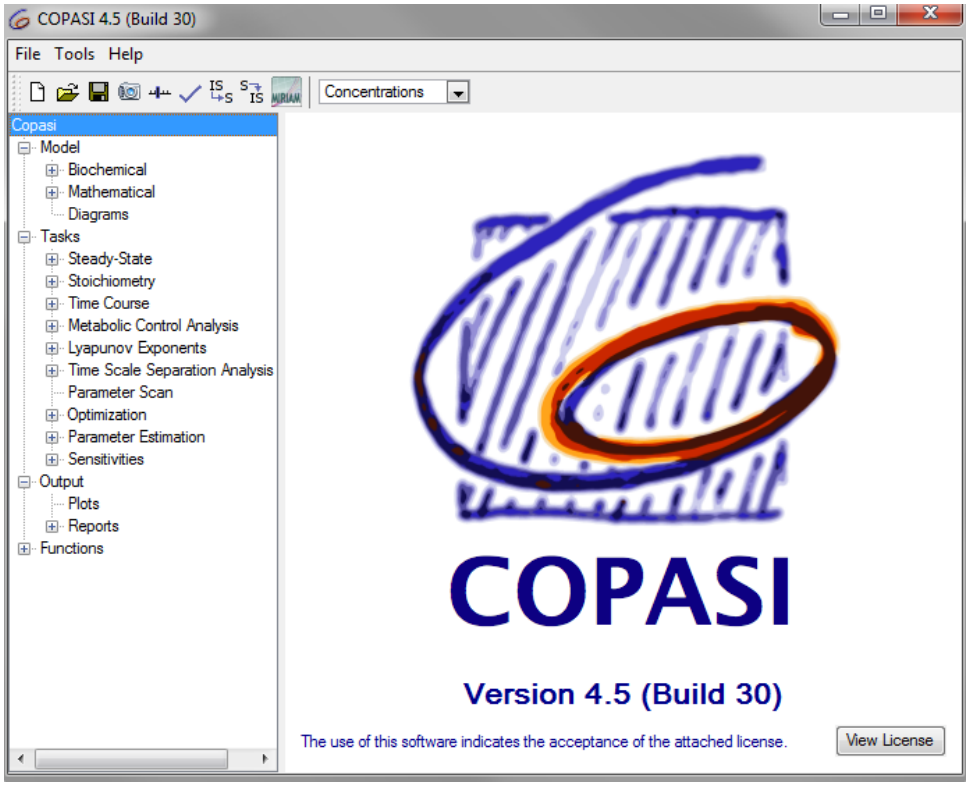
By the Mendes group at VBI and Kummer group at EML Research.

Möglichkeiten mit COPASI

Nachfolgend sollen grundlegend ein paar Möglichkeiten mit COPASI erläutert werden. Eine ausführliche Dokumentation findet man auf der Website des Programms unter *Documentation → User Manual*. Der Umgang mit dem Programm wird in der Übung zu Softwarewerkzeuge der Bioinformatik erklärt.

COPASI steht in zwei Versionen zur Verfügung, zum einen CopasiUI, welches eine grafische Oberfläche für die interaktive Arbeit bereithält und zum anderen CopasiSE für die Datenverarbeitung von Modellen.

Startet man CopasiUI ohne Kommandozeilenparameter, wird ein neues Modell generiert, woraufhin links im Menübaum vier Hauptelemente zur Verfügung stehen (**Model, Tasks, Output, Functions**).



Das erste Hauptelement **Model** beinhaltet alle Parameter, die das aktuelle Modell betreffen. Neben grundsätzlichen Angaben zum Modell können in den Untermenüs biochemische und mathematische Eigenschaften, sowie Diagrammeinstellungen vorgenommen werden. Unter **Tasks** können verschiedene Berechnungen von COPASI ausgeführt werden, wie z.B. die Berechnung des steady state, elementare Flussmoden oder auch zeitliche Verläufe. Die Berechnung eines zeitlichen Verlaufs ergibt die Trajektorie aller Spezies und damit Metabolite in einem Modell. COPASI stellt hierfür verschiedene Methoden zur Berechnung bereit:

- deterministic time course simulation mithilfe des LSODA Algorithmus
- stochastic calculation (next reaction method) für Modelle mit einer kleinen Partikelanzahl (Gibson & Bruck)
- hybrid method, welche die Vorteile des deterministischen und stochastischen Ansatzes kombiniert

Der Menüpunkt **OUTPUT** beinhaltet verschiedene Möglichkeiten, eine Ausgabe zu generieren. Für manche Analysen eines Modells wie z.B. ein zeitlicher Verlauf ist es von Vorteil, die berechneten Ergebnisse in einer Datei oder einem Plot abzuspeichern. Neben bereits vordefinierten Ausgabe-Formen besteht für den Benutzer ebenfalls die Möglichkeit, eigene Ausgaben allein oder mithilfe eines Assistenten zu generieren. Die vordefinierten Ausgabeberichte sind konsistent nach der ausgeführten Aufgabe benannt, d.h. für Gleichgewichtsberechnungen wird die Ausgabe *steady-state* genannt. Neben einer Beschreibung

der Parametereinstellungen sind Informationen über Zwischenergebnisse und abschließende Ergebnisse enthalten.
Besteht der Bedarf, eine grafische Ausgabe in Form eines Plots zu erstellen, bietet COPASI wie erwähnt hierzu die Möglichkeit, wobei in der aktuellen Version lediglich zweidimensionale Diagramme und Histogramme unterstützt werden.

Im letzten Menüpunkt **Functions** stehen dem Benutzer knapp 40 vordefinierte Funktionen zur Verfügung, um Werte aus dem Modell zu erhalten. Jede Funktion enthält eine detaillierte Liste aller Parameter sowie die Formel, die zur Berechnung nötig sind. Wird eine dieser Funktionen als kinetische Funktion einer Reaktion benutzt, prüft COPASI die Konsistenz der Parameter im Modell, was bedeutet, dass jeder Parameter eine von 6 verschiedenen Typen (Substrat, Produkt, Modifikator, Volumen, Zeit und Parameter) zugeordnet wird und dadurch gewisse Voraussetzungen erfüllen muss. So muss z.B. ein Substrat einer Konzentration zugeordnet sein, ein Volumen jedoch einem Kompartiment. Neben den vordefinierten Funktionen ist weiterhin die Option gegeben, eigene kinetische Funktionen zu erstellen. Hierzu muss neben einem eindeutigen Funktionsnamen die Formel zusammen mit allen Parametern definiert werden.

Zusammengefasst ist COPASI ein leistungsfähiges Softwaretool, dass dem Benutzer die Modellierung und Simulation biochemischer Reaktionsnetzwerke aufgrund zahlreicher Methoden erleichtert. Mit dem Programm ist es möglich, die biochemischen Prozesse nachzuvollziehen, die für ein System wichtigen Größen zu bestimmen und sogar zu optimieren, wie z.B. die Produktion eines Stoffes.

10 SBML/ VirtualCell

Aufgrund der Vielfalt an biochemischen Netzwerken und die durch unterschiedliche Bedürfnisse entstandenen Modellierungsansätze und Programme gestaltet sich die gemeinsame Arbeit an einem solchen Modell schwierig. Um eine gemeinsame Nutzung und damit den Austausch und nicht zuletzt die Archivierung solcher Modelle zu erleichtern, hat sich das leicht anwendbare **SBML** Format durchgesetzt, das für den Ausdruck **S**ystems **B**iology **M**arkup **L**anguage steht. SBML basiert auf der Auszeichnungssprache XML (Extensible Markup Language) und ist der Versuch, ein maschinell lesbares Format für die quantitative Beschreibung eines Modells für ein biologisches Netzwerk zu erstellen.

Durch SBML wird es möglich, ein Modell mithilfe verschiedener Softwaretools zu bearbeiten und zu transferieren. So kann z.B. das bereits vorgestellte Programm COPASI ebenfalls SBML- Dokumente importieren/ exportieren, wobei die angestrebte *Interoperabilität* noch nicht perfekt ausgereift ist und es innerhalb mancher Programme zu Fehlermeldungen kommen kann.

Eine Auflistung aller Softwaretools, die kompatibel sein sollen, ist auf folgender Website zu finden (http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix). Aktueller Stand ist SBML Level 3 Version 1 Core, Release 1 (releasedate 31.12.2009).

10.1 Aufbau eines SBML Dokuments

Mithilfe von SBML ist es möglich, biochemische Modelle beliebiger Komplexität zu erstellen. Eine ausführliche Erläuterung des kompletten SBML Syntax ist unter der Spezifikation
http://precedings.nature.com/documents/4123/version/1
zu finden.

Jedes wohldefinierte SBML Dokument muss mit einer XML Deklaration beginnen, die die *Version* sowie die *Kodierung* von XML definiert. Darauf folgend wird die SBML Klasse mit den drei Attributen *level*, *version* und der namespace *xmlns* definiert. Weiterhin muss jede *SBML Klasse* eine Instanz **model** der Modellklasse enthalten, die anhand der Hilfsklassen dann spezifiziert werden kann. Beispielhaft würde dies für die aktuelle Version lauten:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml level="3" version="1=
    xmlns="" http://www.sbml.org/sbml/level3/version1/core">
<model name="EnzymeReaction">\\
```

Grundsätzlich ist die Gliederung des Modells in verschiedene, spezifische Datenobjekttypen intuitiv und einfach zu erstellen. Die Modelldefinition ist in mehrere optionale Listen aufgeteilt, die die relevanten Daten der Modellkomponente beinhaltet (s. Modelldefinition). Die meisten dieser Objektlisten müssen bestimmte Attribute wie **id** (zwingend) oder

name (optional) erhalten, wobei die Definition dieser Attribute innerhalb der Attributmengen eines Modells eindeutig sein muss.

Beginn der Modelldefinition

Liste der Funktionsdefinitionen (optional)
Liste der Einheitendefinitionen (optional)
Liste der Kompartimente (optional)
Liste der Spezies (optional)
Liste der Parameter (optional)
Liste der anfänglichen Zuweisungen (optional)
Liste der Regeln (optional)
Liste der Bedingungen (optional)
Liste der Reaktionen (optional)
Liste der Events (optional)

Ende der Modelldefinition

Ein Beispielcode für das Modell *EnzymeReaction*, welches die Michaelis-Menten Gleichung beschreibt, lautet:

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml level="3" version="1"
    xmlns="http://www.sbml.org/sbml/level3/version1/core">
  <model name="EnzymeReaction">
    <listOfUnitDefinitions>
      <unitDefinition id="per_second">
        <listOfUnit>
          <unit kind="second" exponent="-1"/>
        </listOfUnit>
      </unitDefinition>
    </listOfUnitDefinitions>
    <listOfCompartments>
      <compartment id="cytosol" size="1e-14"/>
    </listOfCompartments>
    <listOfSpecies>
      <species compartment="cytosol" id="ES"  initialAmount="0" name="ES"/>
      <species compartment="cytosol" id="P"   initialAmount="0" name="P"/>
      <species compartment="cytosol" id="S"   initialAmount="1e-20"
          name="S"/>
      <species compartment="cytosol" id="E"   initialAmount="5e-20"
          name="E"/>
    </listOfSpecies>
    <listOfReactions>
    ...
    ...
    </listOfReactions>
  </model>
</sbml>
```

Dieser Code ist nicht vollständig, vermittelt jedoch einen Eindruck über den leicht verständlichen Aufbau eines SBML Dokuments. Schritt für Schritt werden die nötigen Parameter

gesetzt, wobei wie bereits zuvor beschrieben, nicht jede Klasse Angaben benötigt.
Die Definition einer Einheit erfolgt, indem ein oder mehrere Objekte der Klasse „Unit“ kombiniert werden. So wird z.B. die Einheit Liter pro Mol und Sekunde ($\frac{litre}{mol * s}$) erstellt, indem zuerst ein Objekt der Klasse *UnitDefinition* mit dem Attribut *id* definiert wird. Anschließend werden Instanzen der Klasse *Unit* für die einzelnen Einheiten Mol, Liter und Sekunde generiert.

```
<listOfUnitDefinitions>
  <unitDefinition id="per_second">
    <listOfUnits>
      <unit kind="second" exponent="-1"/>
    </listOfUnits>
  </unitDefinition>
  <unitDefinition id="litre_per_mole_per_second">
    <listOfUnits>
      <unit kind="mole"   exponent="-1"/>
      <unit kind="litre"  exponent="1"/>
      <unit kind="second" exponent="-1"/>
    </listOfUnits>
  </unitDefinition>
</listOfUnitDefinitions>
```

Wie in der Übersicht des Modells zu sehen ist, fehlen die Definitionen der Reaktionen und ggf. weitere Spezifikationen des Modells, dies soll an dieser Stelle jedoch nicht weiter von Bedeutung sein.

10.2 Umwandlung von SBML Dateien

Obwohl das SBML Format relativ klar strukturiert und lesbar ist, ist es für den Benutzer angenehmer, eine grafische Darstellung des Modells in Form von Tabellen und Auflistungen vorliegen zu haben. Hierzu gibt es u.a. das Tool **SBML2LATEX**, welches die Informationen einer SBML Datei in verschiedene Formate wie DVI, LATEX oder PDF u.v.m. konvertiert. SBML2LATEX gibt es als eigenständiges auf Java basierendes Programm als auch als Webservice für eine direkte, schnelle Konvertierung (http://www.ra.cs.uni-tuebingen.de/software/SBML2LaTeX/).
Der so erhaltene Bericht gliedert sich in mehrere Abschnitte, beginnend mit einer allgemeinen Übersicht des Modells, in der neben Angaben zu SBML selbst alle Komponenten des Modells aufgelistet sind. Anschließend werden alle Definitionen der Einheiten dargestellt, u.a. auch die 5 vordefinierten SBML Einheiten *substance, volume, area, length* und *time*. Da alle weiteren Parameter eines Modells optional sind, gibt der Bericht auch nur explizit definierte Parameter wieder. Jeder dieser Abschnitte enthält gegebenenfalls die Anzahl der enthaltenen Komponenten sowie alle Informationen zu dieser Komponente. Im Falle des Abschnitts „Reactions“ findet der Benutzer eine Tabelle mit allen Reaktionsgleichungen, welche dann detailliert einzeln beschrieben werden unter Nennung aller Reaktanden, Produkte und Modifikatoren, sowie der kinetischen Gleichung, der abgeleiteten Einheiten und lokalen Parameter.

Abbildung 1: Drager, A. et al. Bioinformatics 2009 25:1455-1456; doi:10.1093/bioinformatics/btp170

10.3 BioModels Database

Nachdem es also möglich ist, ein biochemisches Modell mithilfe verschiedenster Programme zu nutzen, ist es von Vorteil, diese Modelle öffentlich zugänglich zu speichern.

Die Datenbank **BioModels** (http://www.ebi.ac.uk/-biomodels-main/) stellt eine Sammlung quantitativer, biologischer Modelle aus begutachteten Veröffentlichungen dar, die seit April 2005 besteht und seitdem regelmäßig erneuert wird (release 17 im April 2010). Aktuell sind 473 Modelle aus ca. 37850 Spezies mit knapp 19000 Reaktionen in einer MySQL Datenbank gespeichert.

Die Modelle sind mit Kommentaren sowie den relevanten Datenquellen (Veröffentlichungen, Datenbanken der Komponenten und metabolischen Pfade, etc.) versehen. Die Datenbank gewährleistet durch verschiedene Überprüfungen einen korrekten Syntax (XML) als auch die Richtigkeit des Modellaufbaus, die Genauigkeit der biologischen Informationen sowie die Reproduzierbarkeit des Modells. In diesem Zusammenhang ist *MIRIAM* (Minimum Information Required in the Annotation of Models) zu nennen, ein Regelwerk für die Kodierung und Annotation eines biochemischen Modells, dessen Einhaltung Voraussetzung für die Aufnahme des Modells in die BioModels Datenkbank ist. Nichtsdestotrotz gibt es jedoch einen Teil in der BioModels DB, in der Modelle enthalten sind, die noch nicht offiziell in die Datenbank aufgenommen wurden. Dies kann darin begründet liegen, dass das betreffende Modell lediglich noch nicht auf seine Korrektheit überprüft worden ist oder der MIRIAM-Standard bzw. andere Voraussetzungen nicht gegeben sind.
Nachdem das Modell überprüft worden ist, erhält es einen entsprechenden Namen in der Form *AutorJahr_Thema_Methode* und einer ID der Form *BIOMD0000000001* für z.B. das erste Modell.

Nach einer Suche innerhalb der BioModels Datenbank besteht die Möglichkeit, die Modelle in verschiedenen Formaten (z.B. .gif für Grafiken oder BioPAX für Pfade) neben dem Standard SBML herunterzuladen.

10.4 Prozesse in einer Zelle: Beispiel Diffusion

Prozesse innerhalb einer Zelle sind mannigfaltig und neben dem Metabolismus spielt z.B. auch die Migration und im Speziellen die *Chemotaxis* aufgrund des sog. *spatial sensing* eine wichtige Rolle. Zellen sind in der Lage, zwischen Rezeptor-vermittelten Signalen innerhalb verschiedener zellulärer Bereiche zu unterscheiden und so z.B. in Richtung chemischer Lockstoffe zu migrieren. Der physikalische Prozess, der dem chemischen Gradienten zugrunde liegt, ist die **Diffusion**.
Diffusion ist ein passiver Vorgang, welcher zu einer gleichmäßigen Verteilung von Teilchen aufgrund ihrer Eigenbewegung führt. In Bereichen höherer Konzentration bzw. Teilchendichte kommt es häufiger zu Zusammenstößen dieser Teilchen als in Bereichen geringerer Konzentration, was letztendlich dazu führt, dass sich der Konzentrationsunterschied abbaut und sich ein Gleichgewicht einstellt.

Mathematisch entspricht die Teilchendichte bei räumlich konstanter Dichte dem Verhältnis der Teilchenanzahl N zum Volumen V . Die Diffusion beschreibt jedoch eine Dichteveränderung, die von Raum und Zeit abhängig ist:

$$\rho(\vec{r}, t) = \frac{\Delta N(\vec{r}, t)}{\Delta V}$$

wobei gilt: ρ = Dichte, N = Teilchenanzahl, V = Volumen und t = Zeit

10.4.1 Diffusion ohne Einfluss externer Kräfte

Um eine ortsabhängige Diffusionsgleichung zu entwickeln, sind zwei Beiträge wichtig, zum einen der *Diffusionsstrom* j und zum anderen die *Kontinuitätsgleichung*.
Wie bereits erwähnt, bewegen sich Teilchen fortlaufend. In diesem Zusammenhang beschreibt die sog. *Teilchenstromdichte* J oder *Diffusionsstrom* j quantitativ die gerichtete Bewegung dieser Teilchen und ist laut dem **1. Fick'schen Gesetzes** proportional zum Konzentrationsgradienten entgegen der Diffusionsrichtung:

$$\vec{j}(\vec{r}, t) = -D\ \nabla\rho(\vec{r}, t) = -D\ grad\ \rho(\vec{r}, t) \tag{10.4.1}$$

Hierbei beschreibt $-D$ den Diffusionskoeffizienten für einen Strom, der von der hohen Dichte zur niedrigeren fließt und ist daher negativ. $grad\ \rho(\vec{r}, t)$ steht für die Dichtefluktuationen, was nichts anderes bedeutet, als den Teilchen- und damit Konzentrationsgradienten.

Der zweite Beitrag zur Diffusionsgleichung ist die **Kontinuitätsgleichung**, die den Massenerhalt, d.h. die Änderung der Dichte ρ abhängig von der Zeit an einem festgehaltenen

Ort r als Divergenz des Diffusionsstroms beschreibt:

$$\frac{\partial \rho(\vec{r},t)}{\partial t} = -\nabla \vec{j}(\vec{r},t) = -div\ \vec{j}(\vec{r},t) \tag{10.4.2}$$

Anschaulich gesprochen bedeutet es festzustellen, was mit den Teilchen geschieht. Für ein offenes System wäre dies die Differenz zwischen eintretenden und austretenden Teilchen, z.B. $N_{in} = 5$ und $N_{out} = 7$ und damit $\Delta N = N_{in} - N_{out} = -2$.

Setzt man nun das 1. Fick'sche Gesetz in die Kontinuitätsgleichung ein, ergibt sich daraus die **Diffusionsgleichung**, welche das **2. Fick'sche Gesetz** darstellt:

$$\vec{j}(\vec{r},t) = -D\ \nabla \rho(\vec{r},t) \qquad in \qquad \frac{\partial \rho(\vec{r},t)}{\partial t} = -\nabla \vec{j}(\vec{r},t) \tag{10.4.3}$$

ergibt:

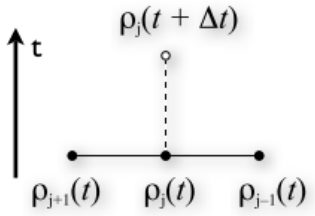
$$\frac{\partial \rho(\vec{r},t)}{\partial t} = -\nabla(-D\ \nabla \rho(\vec{r},t)) \tag{10.4.4}$$

Unter der Voraussetzung, dass der Diffusionskoeffizient D konstant ist, folgt:

$$\frac{\partial \rho(\vec{r},t)}{\partial t} = D\ \Delta \rho(\vec{r},t) \tag{10.4.5}$$

Die Diffusionsgleichung stellt somit eine vollständige Beschreibung der zeitabhängigen Dichteverteilung ohne die Auswirkung externer Kräfte dar. Mit konstantem Diffusionskoeffizient beschreibt die Gleichung also eine Zeitentwicklung, die durch globale Verteilungen der Teilchen bestimmt wird und statt der Ableitungen nach **Zeit** und **Ort** nur die Ableitung der Dichte nach der Zeit benötigt.

Mithilfe des **FTCS Integrators** (Forward in Time Centered in Space) wird die Ableitung der Zeit durch eine vorausgehende Differenz-Abschätzung und der Laplace-Operator durch eine gemittelte Differenzabschätzung ersetzt:



$$\underbrace{\frac{\rho_j(t+\Delta t) - \rho_j(t)}{\Delta t}}_{\text{Forward in Time}} = \underbrace{D\frac{\rho_{j+1}(t) - 2\rho_j(t) + \rho_{j-1}(t)}{\Delta x^2}}_{\text{Centered in Space}}$$

Die Dichte ρ_j für die Vorwärtsdifferenz der Zeit ist gegeben durch:

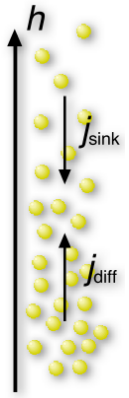
$$\rho_j(t+\Delta t) = \rho_j(t) + \Delta t D\ \frac{\rho_{j+1}(t) - 2\rho_j(t) + \rho_{j-1}(t)}{\Delta x^2}$$

10.4.2 Diffusion unter dem Einfluss externer Kräfte

Wirkt hingegen eine Kraft auf die Teilchen, wie zum Beispiel die Schwerkraft, kann für die Diffusionsgleichung eine stationäre Lösung gefunden werden, die unabhängig von dem Diffusionskoeffizient D ist. Bei einer Diffusion entgegengesetzt zur Schwerkraft ist für die Berechnung der Dichte ρ ein zweiter Teilchenstrom j_{sink} zu beachten, der gegen den Diffusionsstrom j_{diff} wirkt. Der Teilchenstrom aufgrund der Schwerkraft setzt sich zusammen aus der Gravitationskraft $F = mg$ geteilt durch die Gravitationskonstante γ (G):

$$j_{sink}(h) = \nu\rho(h) = -\frac{mg}{\gamma}\rho(h)$$

$$j_{diff}(h) = -D\frac{d\rho(h)}{dh}$$



Für den stationären Zustand gilt, dass die Summe der beiden Ströme gleich Null ist, d.h. die beiden Ströme heben sich auf:

$$j_{sink}(h) + j_{diff}(h) = 0$$

Um den stationären Zustand unabhängig von der Diffusionskonstante D zu beschreiben, wird diese durch den Term $\frac{k_B T}{\gamma}$ ersetzt, der die Boltzmannverteilung miteinbezieht mithilfe der Boltzmannkonstante k_B und der Temperatur T. Durch Umformen Gleichung des stationären Zustands ein Einsetzen folgt:

$$\begin{aligned} j_{sink}(h) &= -j_{diff}(h) \\ D\frac{d\rho(h)}{dh} &= -\frac{mg}{\gamma}\rho(h) \\ \frac{d\rho(h)}{dh} &= -\frac{mg\gamma}{\gamma k_B T}\rho(h) \\ \frac{d\rho(h)}{dh} &= -\frac{mg}{k_B T}\rho(h) \end{aligned}$$

Für die Lösung der Differentialgleichung und somit für die Dichte ergibt sich dann:

$$\rho(h) = \rho_0\ e\left[\frac{mgh}{k_B T}\right]$$

10.5 Zellmodelle - The Virtual Cell

Für die Simulation von diffusiven Prozessen in Zellen ist die Kenntnis einiger kinetischer und physikochemischer Parameter notwendig. Man kann bequem auf vorhandene Softwaretools zurückgreifen. **The Virtual Cell** ist ein Projekt der National Resource for Cell

Analysis and Modelling (NRCAM) und steht dem Benutzer als webbasierte Java Applikation zur Verfügung (http://www.nrcam.uchc.edu/index.html, aktuelle Version 4.7, released 28.04.2010). Mithilfe des Programms ist es möglich, die Topologie und Geometrie von Kompartimenten, die molekularen Charakteristiken sowie die relevanten Interaktionsparameter festzulegen, welche dann automatisch in ein mathematisches System bestehend aus gewöhnlichen bzw. partiellen Differentialgleichungen umgesetzt werden. Die grafische Oberfläche erleichtert die Erstellung eines Modells, ist jedoch nicht zwingend notwendig für fortgeschrittene Benutzer, da es ebenfalls möglich ist, direkt die gesamte mathematische Modellbeschreibung zu definieren.
Nach der Erstellung des Modells wird dieses durch die Programm-internen numerischen Lösungsmethoden in Softwarecode umgesetzt, um das Modell zu simulieren und analysieren. Die Ergebnisse können sowohl online analysiert als auch in verschiedenen Formaten heruntergeladen werden.

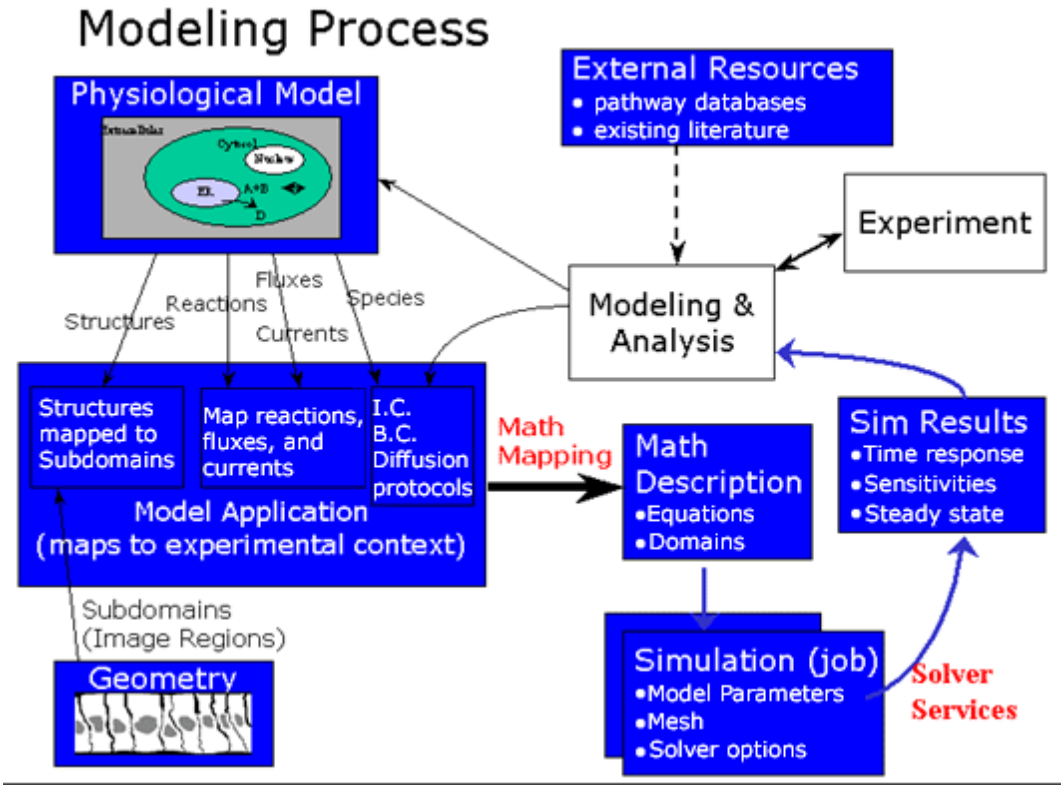
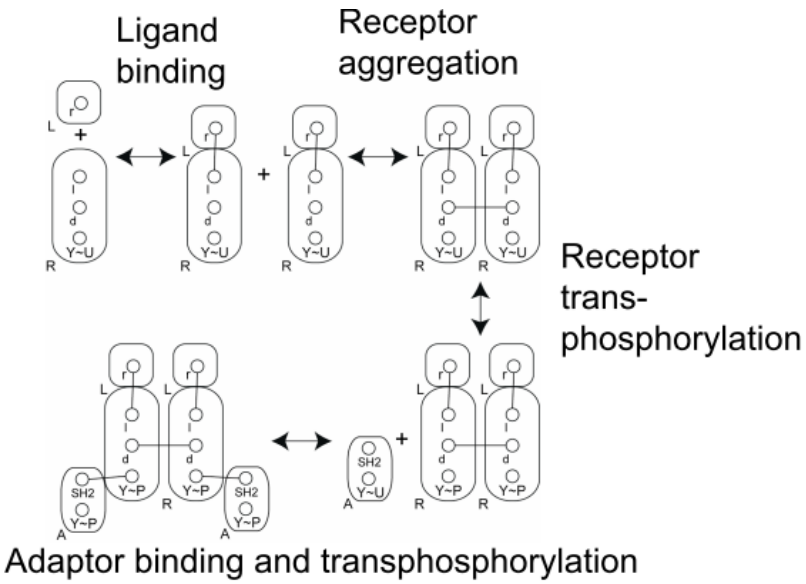


Abbildung 2: Modeling Process performed in Virtual Cell

Wie in der Abbildung gezeigt, ist The Virtual Cell in der Lage, externe Datenquellen miteinbeziehen, so z.B. Pfadbeschreibungen aus der KEGG Datenbank oder Informationen aus Literaturquellen. Zusammenfassend ist The Virtual Cell ein weiteres Tool, um Simulationen zu generieren und analysieren und ist auf die Simulation zellulärer Reaktionsnetzwerke spezialisiert, die diffusive Prozesse beinhalten.

10.5.1 BioNetGen

Wie bereits beschrieben, entwickeln biologische Modelle relativ schnell eine hohe Komplexität, wenn viele Stoffe an einer Reaktion beteiligt sind. Die Komplexität nimmt weiterhin sehr schnell zu, wenn verschiedene Zustände der Spezies miteinbezogen werden. Ein bekanntes Beispiel hierfür sind Signaltransduktionsnetzwerke wie z.B. die Signaltransduktion ausgelöst durch epidermale Wachstumsfaktoren, die an Rezeptoren binden. Nicht nur, dass sehr viele Proteine und andere Moleküle daran beteiligt sind, Prozesse wie die Dimerisierung der Rezeptoren oder die Phosphorylierung des Wachstumsfaktors führen sehr schnell zu einer kombinatorischen Explosion der zu simulierenden Spezies und Reaktionen. Viele Modelle basieren auf der Annahme, dass nur bestimmte Kombinationen relevant sind. Jedoch sind solche Annahmen nicht allgemeingültig und basieren nicht auf konkretem Wissen.
Eine Möglichkeit, einer solchen Komplexität gerecht zu werden, ist der regelbasierte Ansatz.
BioNetGen stellt eine Reihe von Software-Tools dar, mit deren Hilfe man Regeln für solch ein umfassendes Modell definieren kann. Diese Regeln beschreiben u.a. Aktivitäten, mögliche Modifikationen und Interaktionen der Reaktanden. Mithilfe des Algorithmus von BioNetGen werden dann alle möglichen Kombinationen anhand der definierten Regeln generiert.



Die Grafik stellt eine schematische Darstellung der Ligand-Rezeptor Interaktion mithilfe von BioNetGen dar. Deutlich wird die Darstellung der Spezies (Rezeptor R, Ligand L und Adapter A) und der verschiedenen Zustände (l, d, P) der funktionellen Komponenten eines solchen Moleküls. So bindet der Ligand L an den Rezeptor R (vertikale Linie) und es kommt zur Dimerisierung zweier solcher Ligand-Rezeptorkomplexe (angedeutet durch die horizontale Linie der zwei mit *d* markierten Zustandskreise). Anschließend wird eine bestimmte Aminosäure, in diesem Fall Tyrosin (Y), phosphoryliert (Y-P) und abschließend ein Adapter A gebunden und ebenfalls eine bestimmte AS phosphoryliert.

Weiterhin zu sehen ist die Regeldefinition von BioNetGen bezüglich der Spezies und wie mögliche Interaktionen bzw. Vorgänge gehandhabt werden, um der Komplexität eines solchen Systems gerecht zu werden.

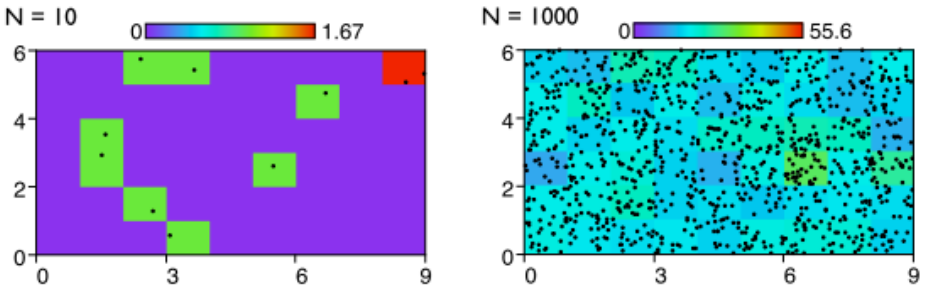
Die Software BioNetGen stellt eine gute und nicht zuletzt eine von derzeit wenigen Möglichkeiten dar, komplexere Systeme zu modellieren. Des Weiteren ist sie mittlerweile in das Programm The Virtual Cell integriert als stand-alone Anwendung namens **BioNetGen@VCell**. Zum einen kann direkt mithilfe der VCell Benutzeroberfläche BioNetGen aufgerufen werden, zum anderen ist jedoch auch die Möglichkeit gegeben, mithilfe von BioNetGen@VCell ein Biomodell aus einer SBML Datei zu erstellen.

11 Stochastische Effekte

In Kapitel 9 wurde bereits in die Simulation biochemischer Modelle eingeführt. Der dort beschriebene, *deterministische Ansatz* mithilfe gewöhnlicher Differentialgleichungen stößt jedoch an seine Grenzen, wenn z.B. die Anzahl der Partikel im Volumen gering ist oder eine detaillierte, räumliche Betrachtung des Systems notwendig ist statt der vereinfachten Aufteilung in homogene Kompartimente.
Eine Möglichkeit, um solche Feinheiten sowohl in räumlicher als auch quantitativer Dimension dennoch präzise simulieren zu können, sind **stochastische Ansätze**. Hierbei werden nicht wie bei deterministischen Ansätzen die Stoffkonzentrationen als Zustandsvariablen im zeitlichen Verlauf betrachtet, sondern die tatsächliche Anzahl an Teilchen jeder Spezies im zeitlichen Verlauf. Weiterhin werden Reaktionskonstanten durch eine *zeitlich variable Reaktionswahrscheinlichkeit* pro Zeitintervall beschrieben. Nachteil eines stochastischen Simulationsansatzes ist der erheblich größere Aufwand im Vergleich zu einem deterministischen Ansatz.

11.1 Grundlagen für stochastische Simulationen

Stochastische Effekte beschreiben zum Beispiel die zufällige Verteilung von sehr wenigen Teilchen in einem Raum, d.h. wenn eine geringe Dichte vorliegt. Wie bereits in Kapitel 10.4 beschrieben, versteht man unter der Dichte die Anzahl der unterscheidbaren Teilchen N pro Volumen V. Aufgrund der bei geringer Dichte seltenen Kollisionen mit anderen Teilchen kommt es zu einer ungleichmäßigen Verteilung, wie folgende Grafiken verdeutlichen. Bei einer Teilchenanzahl $N = 1000$ ist die Verteilung der Teilchen (farblich dargestellt) sehr viel homogener als bei $N = 10$.



Falls es sich bei den Teilchen nun um die Reaktanden einer chemischen Reaktion handelt, ist bei geringer Dichte die Wahrscheinlichkeit dafür, dass zwei Reaktanden kollidieren und eine Reaktion stattfinden kann, räumlich sehr verschieden. Im Folgenden werden verschiedene Ansätze vorgestellt, die die stochastische Komponente in verschiedener Weise behandeln.

11.2 Poisson-Verteilung

Die **Poisson-Verteilung** beschreibt eine diskrete Wahrscheinlichkeitsverteilung, mithilfe derer Voraussagen darüber getroffen werden, wie hoch die Wahrscheinlichkeit ist, dass in einem bestimmten Zeitintervall eine bestimmte Anzahl von zufälligen, voneinander unabhängigen Ereignissen geschieht. Hierzu muss die im Mittel zu erwartende Anzahl an Ereignissen innerhalb dieses Zeitintervalls aus der vorherigen Beobachtung bekannt sein. Hierbei werden drei Annahmen getroffen:

- **Seltenheit**:
  Die Wahrscheinlichkeit für ein Ereignis im Zeitraum $[w, w + \Delta w]$ ist $<< 1$
- **Proportionalität**:
  Die Wahrscheinlichkeit eines Ereignisses im Zeitintervall Δw ist proportional zur Länge des Intervalls Δw
- **Geschichtslosigkeit**:
  Das Ereignis im Zeitintervall Δw ist unabhängig von Ereignissen, die vorher eingetroffen sind (Geschichtslosigkeit)

Es sei λ die Anzahl der im Mittel eintretenden Ereignisse in einem Zeitintervall Δw in einem Kontinuum w. Die **Wahrscheinlichkeit**, dass $k = 0, 1, 2, ...$ Ereignisse auftreten ist dann:

$$p_k = \frac{\lambda^k}{k!} e^{-k} \qquad \text{mit } \textbf{Mittelwert } k = \sum k \; p_k = \lambda$$

Die **Varianz** σ^2 errechnet sich aus:

$$\sigma^2 = \sum p_k (k - < k >)^2 = \lambda$$

Um die **relative Streuung (Fehler)** zu berechnen, gilt:

$$\frac{\Delta k}{k} = \frac{\sigma}{< k >} = \frac{\sqrt{\lambda}}{\lambda} = \frac{1}{\sqrt{\lambda}}$$

Beispielhaft ergibt sich also für eine durchschnittliche Teilchenanzahl von $\lambda = 100$ ein Fehler von $\frac{1}{\sqrt{\lambda}} = \frac{1}{\sqrt{100}} = \frac{1}{10} = 10\%$.

Betrachtet man nun dies im Zusammenhang mit einer chemischen Reaktion $A + B \Rightarrow AB$, so ergibt sich für eine kontinuierliche Ratengleichung folgende Wahrscheinlichkeit für die Bildung des Produkts AB:

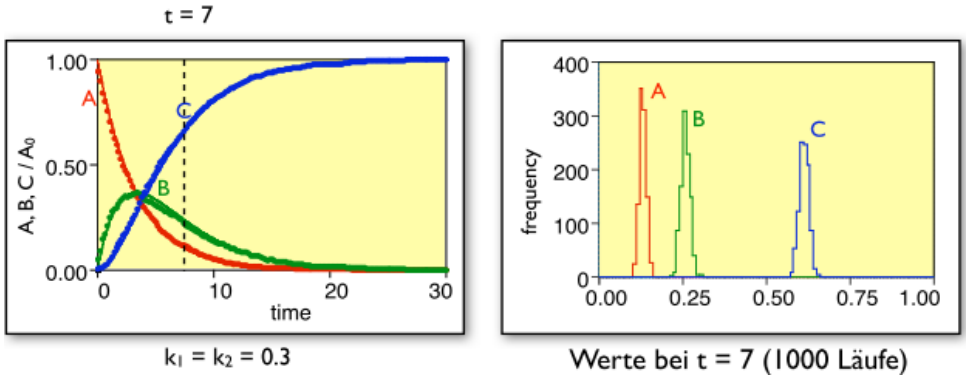
Kontinuierliche Ratengleichung: $\frac{d[AB]}{dt} = k[A][B]$

Für die Wahrscheinlichkeit des Ereignis k, in diesem Fall die Produktbildung AB im Volumen V im Zeitintervall Δt ergibt sich daher:

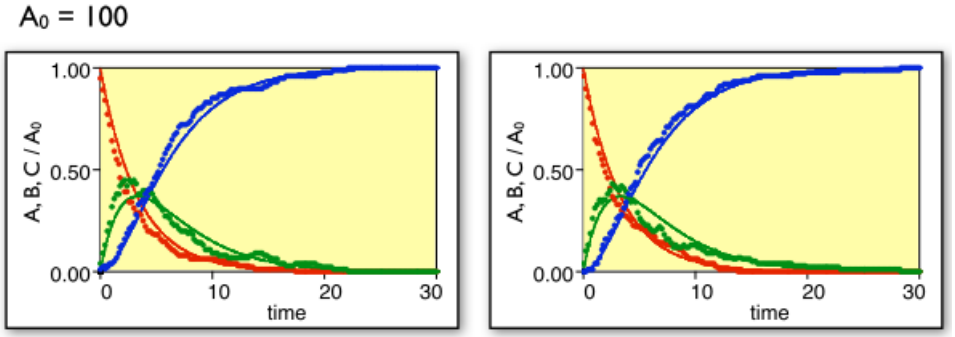
$$\begin{aligned}\Delta N_{AB} &= \frac{d[AB]}{dt} V\ \Delta t \\ &= k_{AB} \frac{N_A}{V} \frac{N_B}{V} V \Delta t \\ &= k_{AB} \frac{k_{AB} \Delta t}{V} N_A\ N_B \\ &= P_{AB} N_A\ N_B\end{aligned}$$

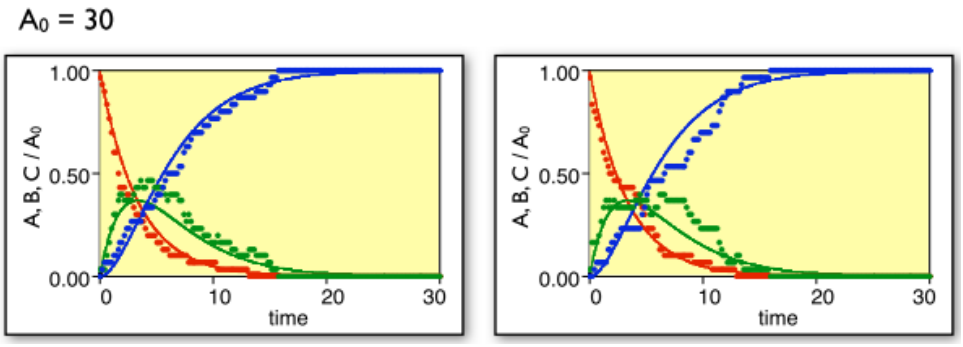
Aus dem letzten Umformungsschritt wird ersichtlich, dass die Reaktionsrate k_{AB} im Zeitintervall Δt im Volumen V der Reaktionswahrscheinlichkeit P_{AB} entspricht.

Um Fluktuationen der Teilchenanzahl besonders deutlich darzustellen, kann z.B. bei einer stochastischen Simulation ein bestimmter Zeitpunkt t_x ausgewählt werden, um dann mithilfe mehrerer Durchläufe festzustellen, ob die Teilchenanzahl der an der Reaktion beteiligten Stoffe variiert. In den folgenden Abbildungen wurde diese Vorgehensweise für die Reaktionskette $A \Rightarrow B \Rightarrow C$ angewandt, mit einer anfänglichen Ausgangsmenge $A_0 = 1000$ zum Zeitpunkt t_0.

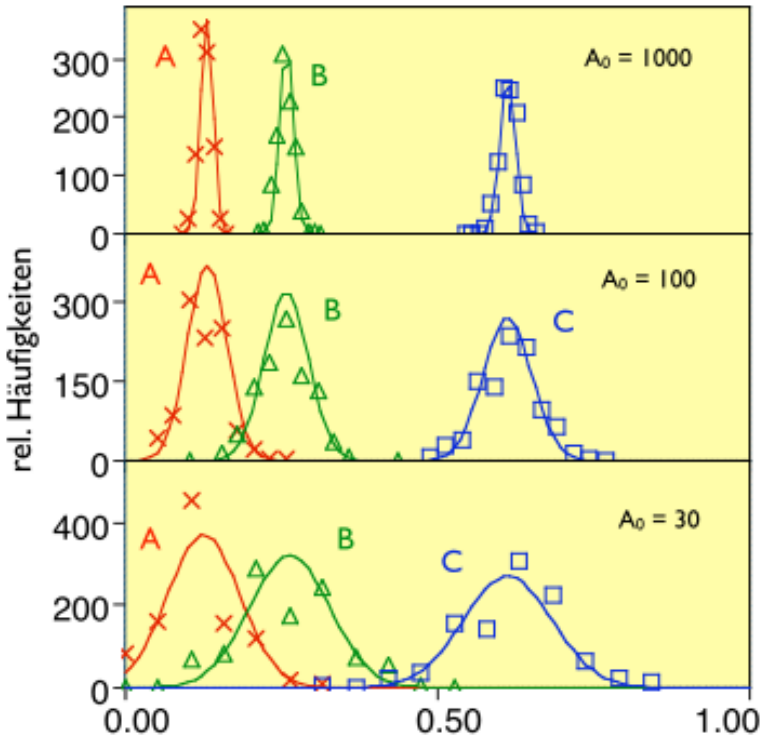


Wiederholt man diese Vorgehensweise für eine geringere Ausgangsmenge von Teilchen ($A_0 = 100$ *bzw.* $A_0 = 30$), wird deutlich, dass die ermittelten Anzahlergebnisse zum ausgewählten Zeitpunkt immer stärker abweichen.





Durch einen Fit der drei Verteilungen mithilfe einer Normalverteilung werden die Fluktuationen besonders deutlich, wie folgende Abbildung zeigt. Die Varianz der einzelnen Reaktanden-Teilchenanzahlen wird mit sinkender Ausgangsteilchenanzahl immer größer:



Zur vollständigen Beschreibung der Zustandswahrscheinlichkeitsdichten einer chemischen Kinetik wird die sog. *Mastergleichung* genutzt, die die zeitliche Evolution der Zustandswahrscheinlichkeitsdichtefunktion beschreibt (Für weitere Informationen hierzu s. VL Bioinformatik III). Jedoch ist nur in den einfachsten Fällen eine Lösung dieser Gleichung zu finden. Daniel T. Gillespie hat hierzu 1977 eine stochastischen Ansatz entwickelt, der im Folgenden beschrieben werden soll.

11.3 Gillespie Algorithmus

Mithilfe des Gillespie-Algorithmus ist es möglich, eine schnelle und genaue numerische Lösung für chemische Reaktionskinetiken zu finden, ohne die Mastergleichung direkt lösen zu müssen. Hierbei wird mithilfe einer Monte Carlo Prozedur die zeitliche Entwicklung eines System numerisch simuliert. Im Gegensatz zu naiven stochastischen Ansätzen wird nicht die Wahrscheinlichkeit für das Eintreten eines Ereignisses im Zeitraum $(t, t + \Delta t)$ berechnet, sondern **wann** das nächste Ereignis eintritt, d.h. eine **Wartezeit** s.
Dadurch wird das Problem umgangen, dass die Wahrscheinlichkeit eines Ereignisses in dem Zeitraum so gering ist, dass es bei einem naivem Algorithmus bei den allermeisten Iterationsschritten zu keiner Veränderung der Teilchenanzahl kommt.
Für eine einfache chemische Zerfallsreaktion $A \xrightarrow{k} \emptyset$ bedeutet dies folgenden Unterschied:

Naiver Algorithmus:

$A = A_0$
For every timestep:
    get random number $r \in [0, 1)$
    if $r \leq A\ k\ dt$:
        $A = A - 1$

Gillespie:

$A = A_0$
While (A > 0):
    get random number $r \in [0, 1)$
    $t = t + s(r)$
    $A = A - 1$

Wie der Pseudocode des Gillespie-Algorithmus aufzeigt, wird statt eines festen Zeitintervalls ein variables Intervall gewählt, nach welchem das Ereignis dann definitiv eintritt.
Zur Berechnung der Wahrscheinlichkeiten nach Gillespie sei $A(t)$ die zum Zeitpunkt t vorhandene Teilchenanzahl des Reaktanden A und s die Wartezeit bis zum nächsten Ereignis (in unserem Fall eine chemische Reaktion des Reaktanden A). Die exponentielle Verteilung der Wartezeiten zwischen diskreten Reaktionsereignissen ist dann:

$$g(A(t), s) = exp[-A(t)ks] = e^{-s/s_0}$$

Sei $r = exp[-A(t)ks]$ und die mittlere Wartezeit $s_0 = \frac{1}{k\ A(t)}$, so ergibt sich für die Wartezeit s aus der exponentiellen Verteilungsgleichung:

$$s = \frac{1}{k\ A(t)} ln\left[\frac{1}{r}\right] = \frac{1}{\alpha_0} ln\left[\frac{1}{r}\right]$$

Da biologische Prozesse fast nie nur aus einer einzigen Zerfallsreaktion bestehen, soll an dieser Stelle noch die Vorgehensweise für mehrere Reaktionen erwähnt werden.

1. Die Wahrscheinlichkeiten für jede einzelne Reaktion seien α_i mit $i = 1, ..., N$. Die Gesamtwahrscheinlichkeit ist daher $\alpha_0 = \sum \alpha_i$

2. Die Wartezeit bis zum nächsten Ereignis ist dann $s = \frac{1}{\alpha_0} ln\left[\frac{1}{r_1}\right]$

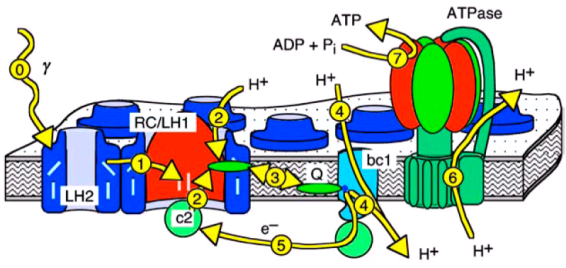
3. Die Auswahl einer Reaktion erfolgt dann durch:
   $\sum_{i=1}^{j-1} \alpha_i \leq \alpha_0 r_2 < \sum_{i=1}^{j} \alpha_i$

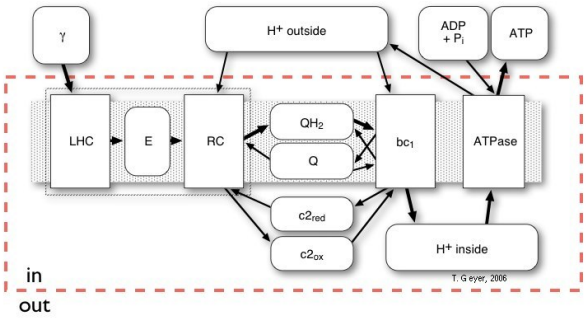
4. Aktualisierung der Teilchenanzahl

In diesem Fall werden also bei jedem Iterationsschritt zwei Zufallszahlen gezogen. Schritt 3 garantiert, dass jede Reaktion i gemäß ihrer individuellen Reaktionswahrscheinlichkeit α_i „an die Reihe kommt“ .

11.4 Pools-and-Proteins

Ein an der Universität des Saarlandes entwickeltes Tool zur stochastischen Simulation eines Chromatophorvesikeln des Bakteriums *Rhodobacter sphaeroide* ist **Vesimulus**. Vesimulus kann über das web-Frontend **Vesiweb** (http://service.bioinformatik.uni-saarland.de /vesiweb/) gesteuert werden und erlaubt dem Benutzer den Einblick in das molekulare Modell für ein solches Vesikel.
Die Simulation basiert auf der Methode des **Pools-and-proteins** Modells, welches die Proteine mit ihren internen Zuständen mit der Netzwerk-Ansicht der einheitlichen Konzentrationen für eine diskrete Anzahl unterscheidbarer Metabolite verbindet. Der Ablauf der Photosynthese mit der Bindung von Metaboliten an Proteine wird einfach und deutlich dargestellt, indem die Proteine verschiedene Input- und Output Konnektoren besitzen, wodurch Metabolite aus dem Pool aufgenommen bzw. auch wieder zurückgegeben werden können. D.h., für ein Modell eines biologischen Systems gibt es eine bestimmte Anzahl an Kopien für jeden Proteintyp und einen Pool pro Metabolittyp, charakterisiert durch Größe und Anzahl der Metabolitmoleküle. Für die Photosynthese innerhalb eines Chromatophorvesikels sieht die Modellierung mittels Vesimulus folgendermaßen aus, wobei die Vierecke die Proteine darstellen und die abgerundeten Vierecke die Pools:





Für das Photosynthese-Modell ergeben sich so 40 aktive Moleüle, die unabhängig voneinander sind und nur eine einzige stochastische Reaktion mit je einem Molekül eingehen. Weiterhin gibt es 19 Pools mit je einem Pool pro Metabolit, wobei die Verbindungen zwischen Proteinen und Pools das biologische System definieren.
Der Vorteil der Pools-and-Proteins Methode gegenüber dem Gillespie-Algorithmus ist, dass mehrere Reaktionen pro Zeitschritt möglich sind statt nur einer Reaktion, sowie die Belegung unabhängiger Bindungsstellen überprüft wird. Dadurch ist es möglich, das mikroskopische Verhalten jedes einzelnen Proteins zu simulieren unter Beibehaltung der Netzwerktopologie.

12 Petrinetze und Boolsche Netze

Wie bereits in Kapitel 9 erwähnt, gibt es bei der dynamischen Modellierung neben chemischen Modellen anhand der Reaktionskinetik auch die Möglichkeit, logische Modelle, wie z.B. Petri-Netze oder Boolsche Netze zu nutzen.
Hierbei werden die Reaktionsabläufe auf Graphnetze abgebildet, wobei gewisse Eigenschaften der Reaktionen beachtet werden müssen. So werden die beteiligten Metabolite durch Enzyme umgesetzt, welche voneinander unabhängig sind. Weiterhin müssen die Stöchiometrien der Substrate und Produkte bekannt sein.

12.1 Petri-Netze

Ein **Petrinetz** ist ein *bipartiter Graph*, d.h. ein Modell für die Darstellung von Beziehungen zwischen Elementen zweier Mengen, mit gerichteten und gewichteten Kanten. Zur Modellierung chemischer Reaktionen entsprechen die zwei Knotenmengen **Stellen** (places) und **Übergänge** (transitions) den Metaboliten bzw. Enzymen, welche diese umsetzen. Die gewichteten Kanten stellen die stöchiometrischen Faktoren dar und verbinden Stellen und Übergänge miteinander.
Jede Stelle besitzt eine definierte **Kapazität**, die angibt, wie viele Token (Zeichen) diese Stelle enthält, wobei die Standardkapazität unbegrenzt ist. Jede Kante hingegen besitzt ein Gewicht, welches die Kosten darstellt mit einem Standardkostenwert von 1, sofern kein anderer Wert zugeteilt wurde.
Übergänge können wie Schalter verstanden werden, die aktiviert sind und feuern, wenn die folgenden Bedingungen erfüllt sind:

1. Anzahl der Token auf den Eingangsstellen ist mind. gleich der Kosten des Übergangs
2. Anzahl der freien Plätze auf den Ausgangsstellen ist mindestens gleich groß wie die Kosten des Übergangs dorthin

Ein Beispiel für ein sehr einfaches Petri-Netz ist folgendes, wobei die Stellen kreisförmig und Übergänge als Rechtecke dargestellt werden:
Die obere linke Stelle besitzt 5 Token. Die Übergangskosten betragen 2 zum Übergang

hin und von dort weg zur Ausgangsstelle. Für die untere Eingangsstelle sind 4 Token vorhanden, der Übergang kostet 3 zum Übergang hin und 2 von dort weg. Feuert der obere Übergang, werden 2 Token entnommen, beim Unteren 3 Token und aufgrund der Übergangskosten zur Ausgangsstelle 2 Token dort hinzugefügt.

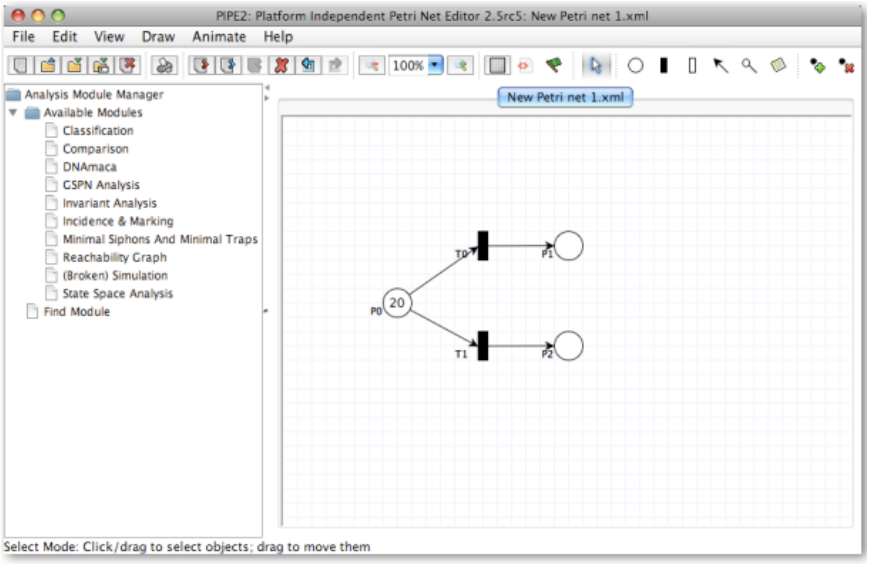
Sind mehrere Übergänge möglich, so wird bei Gleichheit zufällig ausgewählt oder Prioritäten gesetzt, indem der Übergang mit höchster Priorität aktiviert wird.

Bisher wurde angenommen, dass ein Übergang direkt feuert, wenn die Bedingungen erfüllt sind, was mit dem Ausdruck *instantan* beschrieben wird. Es gibt jedoch auch andere Varianten, wie z.B.

- **SPN - Stochastic Petri Net**, bei dem jeder Übergang eine Zeit t braucht, wobei t einer exponentiell verteilten Zufallszahl entspricht.
  $\frac{dN}{dt} = -kN \Rightarrow N(t) = N(0)e^{-kt}$
- **GSPN - Generalized Stochastic Petri Net**, welches sowohl zeitverbrauchende als auch instantane Übergänge besitzt
- **DSPN - Deterministic Stochastic Petri Net**, bei dem die Übergangszeit sowohl exponentiell zufällig als auch fest verteilt sein kann.

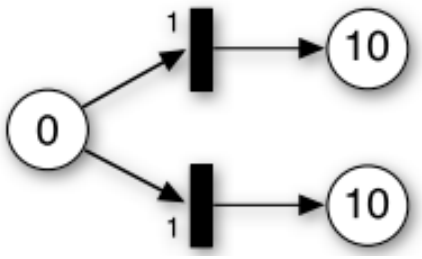
12.1.1 PIPE2 - Softwaretool zur Erstellung und Analyse von Petrinetzen

Ein Softwaretool zur Erstellung solcher Petrinetze ist **PIPE2**, ein open source Programm, das es dem Benutzer Plattform-unabhängig erlaubt, Petrinetze zu erstellen und zu analysieren (http://pipe2.sourceforge.net/, akt. Version 2.5).



Analysiert werden können solche Petrinetze z.B. mit dem sog. **Token Game**, welches einer stochastischen Simulation entspricht.

Beispielhaft soll folgendes Petrinetz als Basis dienen, welches dann über 10 Durchläufe simuliert wird. Dadurch ergeben sich folgende Tokenübergänge mit einem Token-Durchschnittswert von 9.4 für den einen Übergang und 10.6 für den Anderen, es besteht also keine Gleichverteilung:



Lauf	P1	P2
1	10	10
2	15	5
3	11	9
4	9	11
5	13	7
6	7	13
7	7	13
8	5	15
9	9	11
10	8	12
<N>	9.4	10.6
σ	2.8	2.8

Eine weitere Analysemöglichkeit ist die sog. **state-space analysis**, mit der u.a. die mittlere Tokenzahl, Token-Verteilungsdichten als auch Durchsätze ermittelt werden können. Weiterhin kann man die Erreichbarkeit von Markierungen und Zuständen überprüfen und damit das Modell auf dessen Lebendigkeit sowie das Vorhandensein von deadlocks, traps und siphons testen.

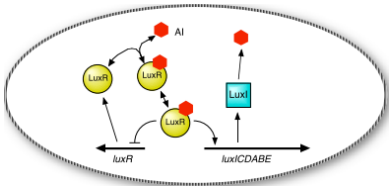
12.2 Boolsche Netze

Oftmals sind biologische Reaktionen durch Bedingungen beschrieben, in denen ein Event von dem Zustand oder dem Vorhandensein eines Metabolit abhängig ist. Solche Aussagen lassen sich sehr gut durch Boolsche Ausdrücke darstellen, wodurch für ein biologisches Modell eine vereinfachte, mathematische Schreibweise möglich ist. Hierbei werden die Dichten der Spezies als diskrete Zustände angenommen, die entweder den Wert 1 oder 0 annehmen, die Zeitentwicklung stellt die Änderung dieser Zustände dar und die Differentialgleichungen eines Modells werden durch sog. **Bedingungstabellen** darstellen.

12.3 Definition eines Boolschen Netzes

Der Zustand eines Systems sei $S_i = \{0, 1, 1, 0, 1, 0, ...\}$, $S_i = \{x_1(i), x_2(i), x_3(i), ...\}$, d.h. ein Vektor diskreter Zustände i. Bei einer festen Anzahl an Spezies mit jeweils endlicher Anzahl an Zuständen ist auch die Anzahl der Zustände des gesamten Systems endlich und es gibt periodische Trajektorien im Zustandsraum. Die periodische Abfolge von Zuständen wird **Attraktor** genannt und alle Zustände, die zu diesem Attraktor führen, **basin of attraction**.

Im Folgenden soll für das Beispiel aus Vorlesung 8, „Quorum sensing“ , ein boolsches Netz aufgebaut werden. Es gelte eine Minimalmenge von Spezies mit den 5 Elementen (LuxR, Al, LuxR:Al, LuxR:Al:Genome, LuxI).



Die Bedingungstabellen lauten dann folgendermaßen, wobei hier nur beispielhaft ein paar der Notwendigen aufgelistet sind:

Lux1I	LuxR:Al:Genome
0	0
1	1

Hier wird festgelegt, inwiefern LuxI von LuxR:Al:Genome abhängig ist.

LuxR:Al:Genome	LuxR:Al
0	0
1	1

Diese Tabelle gibt an, inwieweit LuxR:Al:Genome von LuxR:Al beinflusst wird.

LuxR	LuxR	Al	LuxR:Al:Genome
1	0	0	0
1	1	0	0
0	0	1	0
1	1	1	0
0	0	0	1
1	1	0	1
0	1	1	1
1	1	1	1

LuxR:Al	LuxR	Al	LuxR:Al:Genome
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	0
0	0	0	1
0	1	0	1
0	1	1	1
1	1	1	1

Um nun Attraktoren zu finden, werden die Zustände im hier benutzten Beispiel „Quorum sensing“ auf ganze Zahlen abgebildet, wodurch sich folgende Zuweisung ergibt: {LuxR(LR), LuxR:Al (RA), Al, LuxR:Al:Genome(LAG), LuxI (LI)} = {1, 2, 4, 8, 16}.

Für jeden Attraktor wird dann der periodische Orbit und desse Länge (Periode) bestimmt, sowie der „basin of attraction“ und dessen relativer Anteil am Zustandsraum.

nummer	LR	RA	Al	RAG	LI	-	Zustand
0	.	.	.	.	.	-	0
1	X	.	.	.	.	-	1
2	X	.	.	.	.	-	1

Beginnend bei Zustand 0 ergibt sich z.B. als Attraktor 1 dann ein periodischer Orbit von 1 mit der Länge 1. Die Zustände sind 0 oder 1, wodurch die Größe 2 ergibt und somit einer relativer Anteil am Zustandsraum von 2/32.

Analysiert man nun alle Attraktoren, lassen sich Aussagen über das Verhalten des Systems treffen. Jedoch hängt die Qualität der Ergebnisse stark von der Qualität des Modells ab

und dieses wiederum sehr von den Annahmen, auf denen das Modell basiert. Daher ist es wichtig sich bewusst zu sein, was essentiell für die Modellierung ist und auf welche Annahmen man verzichten kann.

Stichwortverzeichnis