

Molekularbiologische Datenbanken

Sequenzierung

Ulf Leser

Wissensmanagement in der
Bioinformatik



Formalisierung

- Gegeben

- Menge C von Clones (Länge egal)
- Menge P von Probes
- Matrix

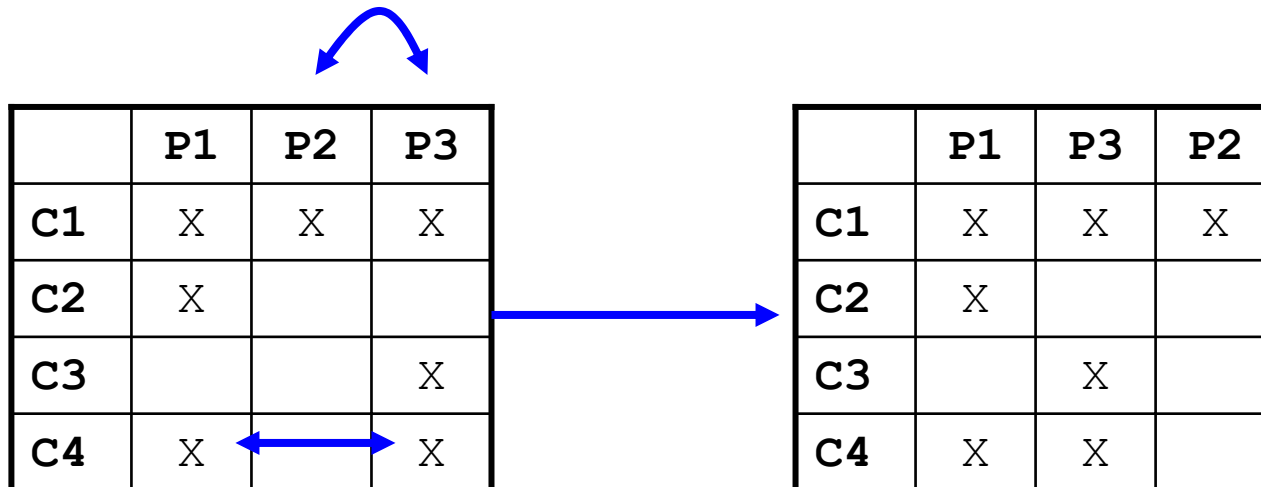
$$M = \begin{cases} 1 & | C_i \text{ matches } P_j \\ 0 & | \text{otherwise} \end{cases}$$

- Definition

- *M hat „consecutive ones property“ (C1P), gdw es eine Permutation π der Spalten von M gibt, so daß in $M' = \pi(M)$ in jeder Zeile alle 1'er in nur einem ununterbrochenen Block liegen*

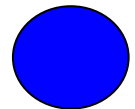
C1P - Ordnung

- Theorem
 - *Eine Matrix M ist C1P gdw. es eine mit M konsistente Ordnung der Clone und Proben gibt*
- Beweis: Literatur



PQ - Baum

- Ziemlich kompliziertes Biest
- Grundidee: Kompakte Repräsentation von Permutationen
- PQ Baum
 - Blätter: Die Proben des Problems
 - P-Knoten
 - Enthält mind. 2 Kinder: PQ-Baum oder Blätter
 - Reihenfolge der Kinder ist nicht festgelegt
 - Q-Knoten
 - Enthält mind. 3 Kinder: PQ-Baum oder Blätter
 - Reihenfolge ist festgelegt: Kinder können in gegebener Ordnung oder umgekehrt vorkommen



Theorem

- Theorem

- *Gegeben eine Matrix M mit n Clonen und m Proben*
- *M ist C1P gdw. das obige Verfahren nacheinander für jeden der Clone eine Transformation findet*
- *Das obige Verfahren findet einen mit M konsistenten PQ Baum, wenn es ihn gibt*
- *Das obige Verfahren läuft in $O(n+m)$*

- Bemerkung

- Das ist nicht selbstverständlich, da i.d.R.
 - viele Regeln matchen, also viele Transformationen möglich sind
 - Die Reihenfolge der Betrachtung der Clone nicht festgelegt ist
- Beweis ist lang und kompliziert

Falsch Positive - Problemformulierung

- Gegeben eine Matrix M
 - die nicht C1P ist, weil sie falsch positive enthält
- Wir suchen die Matrix M' mit den folgenden Eigenschaften
 - M' ist C1P
 - M' kann aus M gewonnen werden durch das beliebige Ersetzen von „1“ durch „0“
 - Also durch Löschen beliebig vieler „positiver“ Signale
 - Unter allen so gewonnenen Matrizen ist M' die Matrix, bei der am **wenigsten Ersetzungen** notwendig sind
 - Also am wenigsten falsch Positive annimmt
 - Und trotzdem C1P ist

Hamming-TSP [AKNW95]

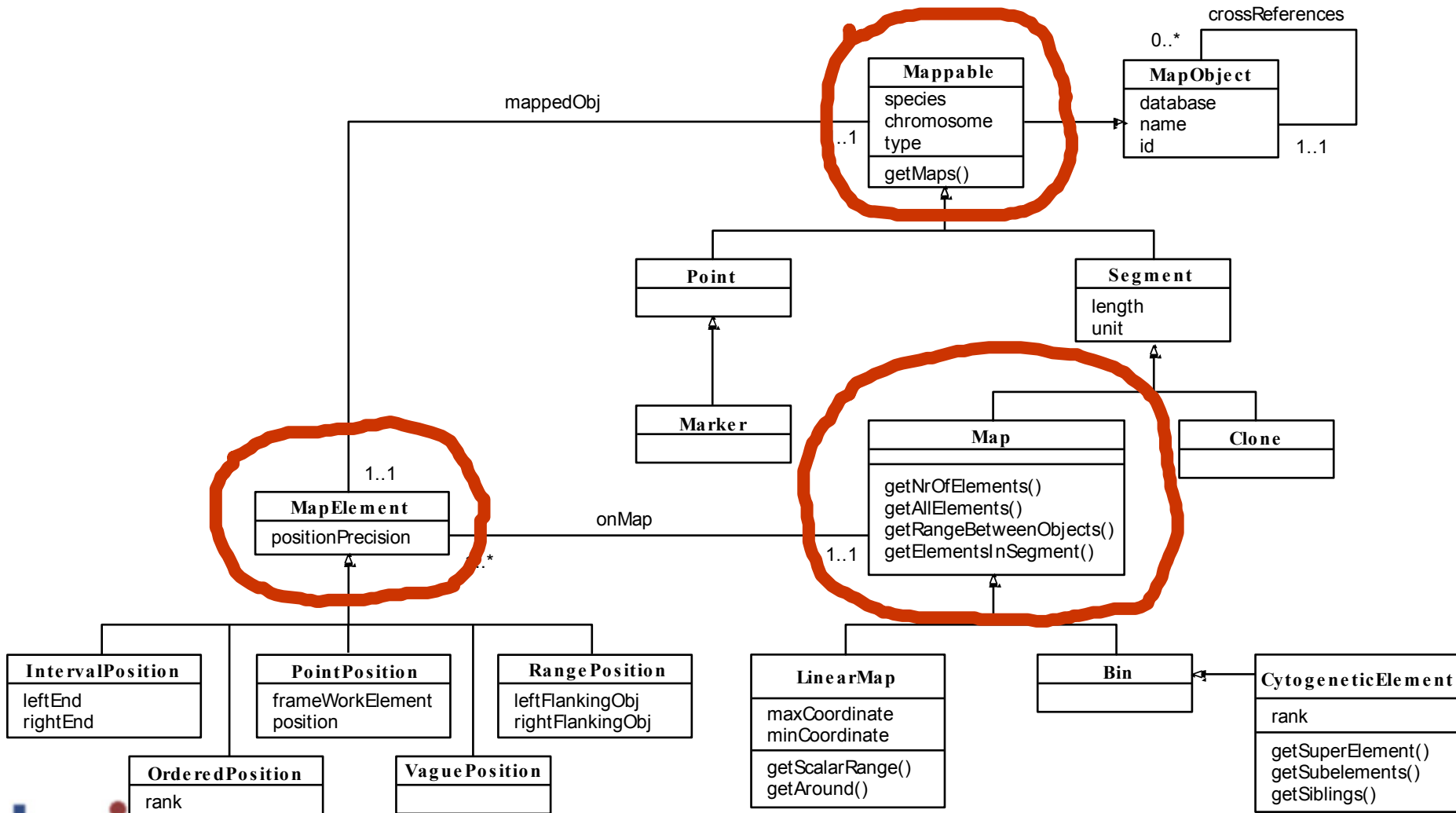
- Theorem

- Geg. eine Clone/Marker Matrix. Konstruiere einen Graphen G wie folgt
 - Ein Knoten für jede Probe
 - Spanne eine Kante von jeder Probe P_1 zu jeder anderen Probe P_2 mit Gewicht $w(P_1, P_2) = \text{hamming}(P_1, P_2)$
- Sei X eine Lösung des Hamming-TSP auf G . Sei M_X die Matrix aus den Spalten von M in der Reihenfolge der Marker in X
- M' kann aus M_X durch Ersetzen der minimalen Menge von „1“ pro Zeile berechnet werden

- Bemerkungen

- Im Allgemeinen gibt es viele X und pro M_X auch viele M' . Alles sind Lösungen im Sinne des Problems
- Lösung ist sehr teuer: **TSP ist NP-vollständig**
 - Und G ist vollständig
- Der letzte Schritt ist einfach – jede Zeile kann einzeln betrachtet werden
- Es werden nur Falsch-Positive beachtet; die anderen drei Fehlerklassen werden ignoriert

OMG Standard „Genome Maps“ [BLL+99]



Inhalt dieser Vorlesung

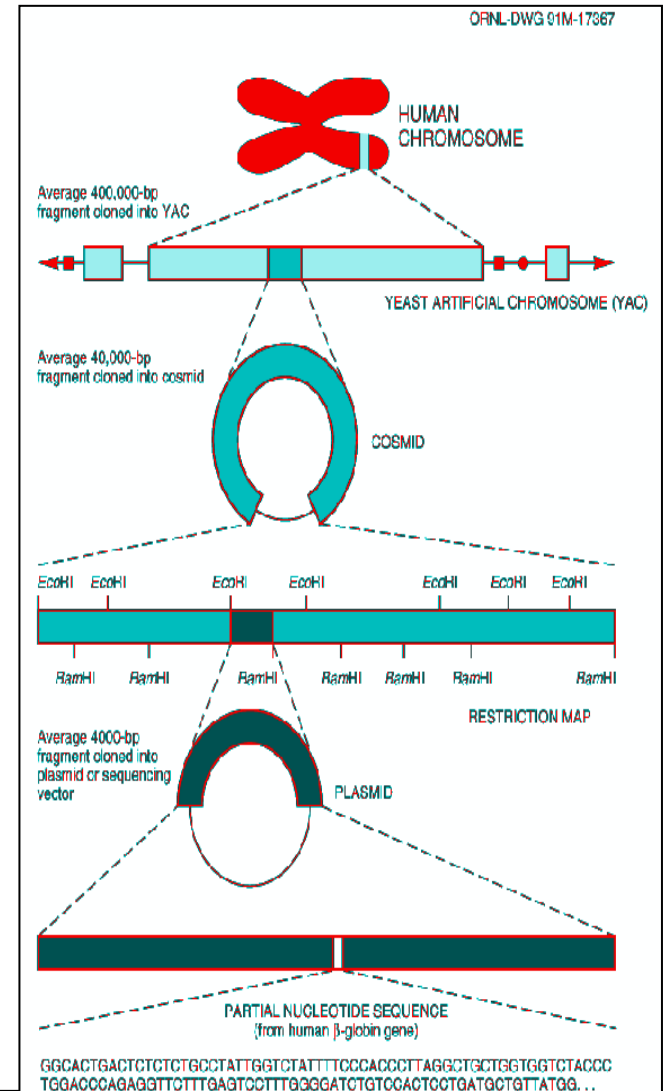
- Strategien zur Sequenzierung
- Sequenzierung
- Datenmodelle
- Datenbanken

Teil I. Strategien zur Sequenzierung

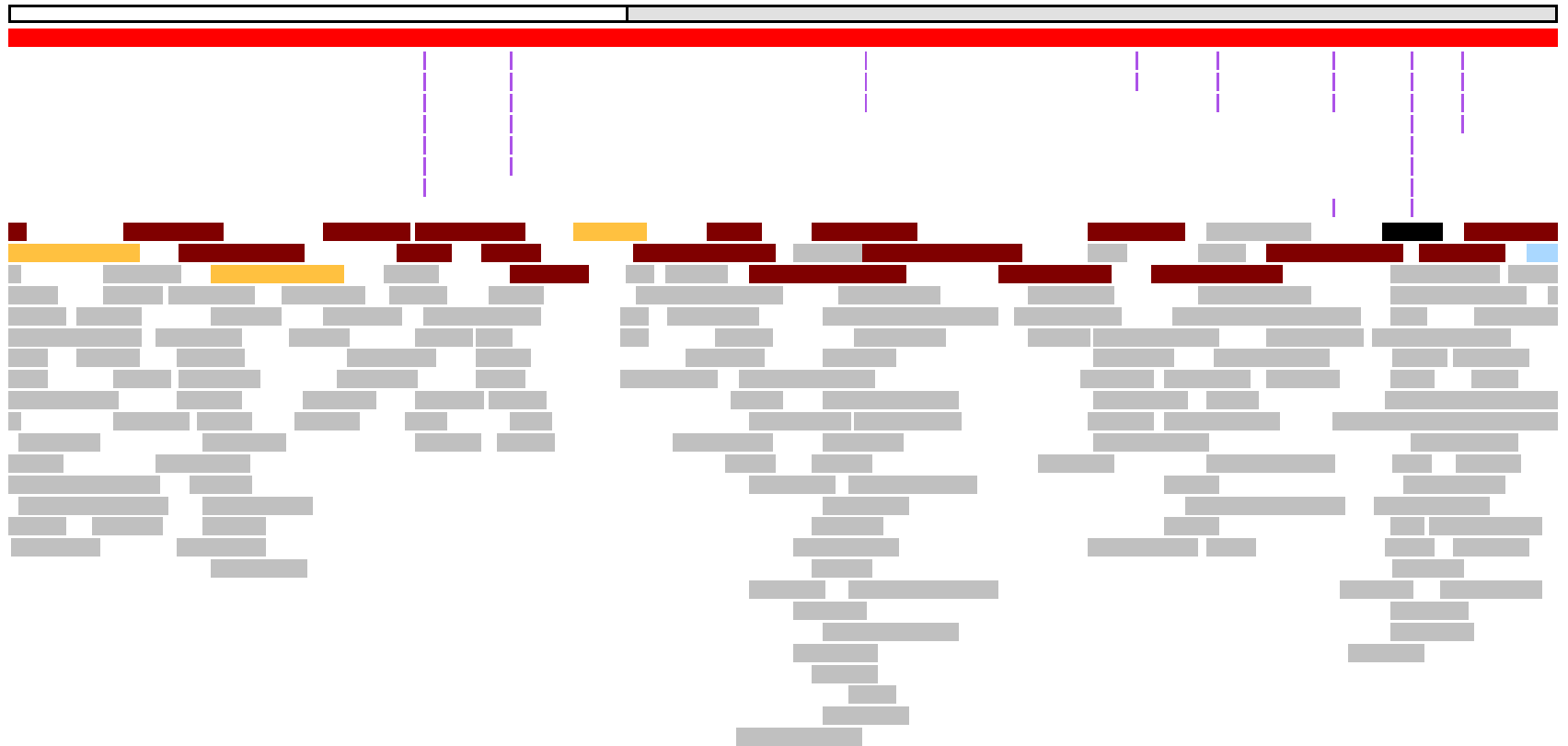
- Mapping und Sequenzierung
- Weg 1: Human Genome Project
- Weg 2: Whole Genome Shotgun Sequencing

Mapping und Sequenzierung

- Chromosome nicht direkt sequenzierbar
 - Zerlegung in Bruchstücke notwendig (Clonierung)
 - Berechnung des Minimum Tiling Paths
 - Sequenzieren der Bruchstücke auf dem MTP
- **Aufwandsminimierung**



BAC Map, Chr.X 1-3 MB



Quelle: <http://www.sanger.ac.uk>

Shotgun Sequenzierung

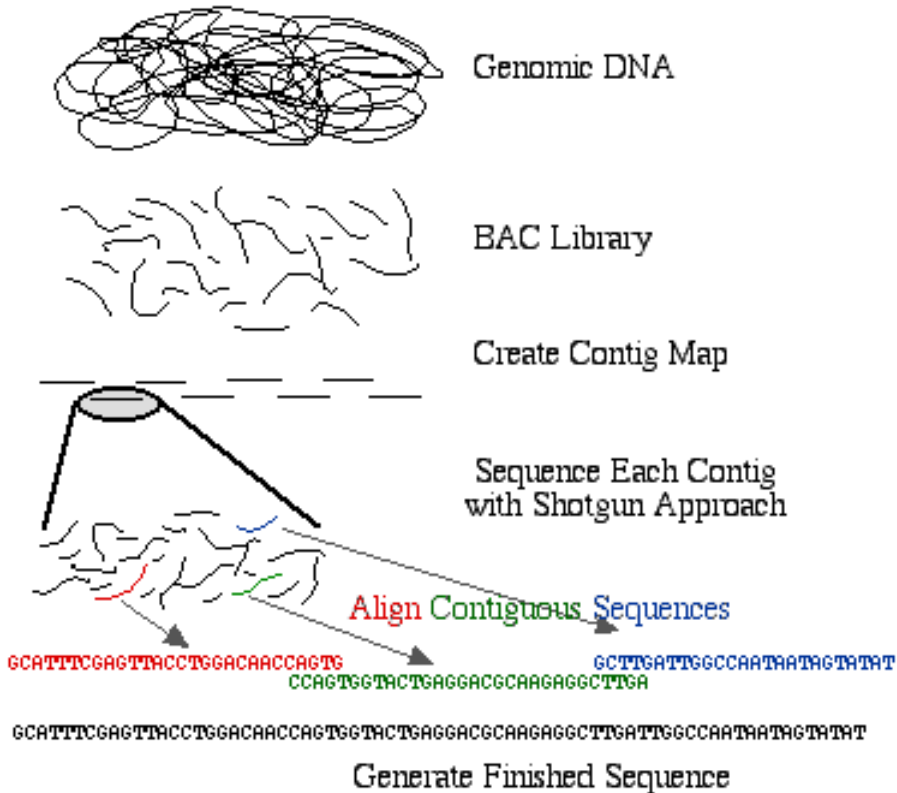
- Auch Cosmids / BACs (30 KB – 300 KB) kann man nicht komplett sequenzieren
- Heutiger Standard: **Shotgun-Sequenzierung**
 - Erneutes Zerbrechen in Stücke \sim 1000 Basen
 - **Sequenzieren aller Stücke** (wie: später)
 - Berechnen der Originalsequenz aus Sequenzüberlappungen (wie: später)
- Warum ?
 - Sequenzierung hochgradig automatisiert
 - Billiger als Feinkartierung in Plasmids
 - Hohe Redundanz verringert Fehlerrate

Redundanz

- Typische notwendige Überdeckung
 - Cosmid (30 KB) : 600 Reads a 500 Basen =
Überdeckungsfaktor 10
 - BAC (120 KB): 2000 Reads a 500 Basen =
Überdeckungsfaktor 8,3

Sequenzierung im HGP

Hierarchical Shotgun Sequencing Method

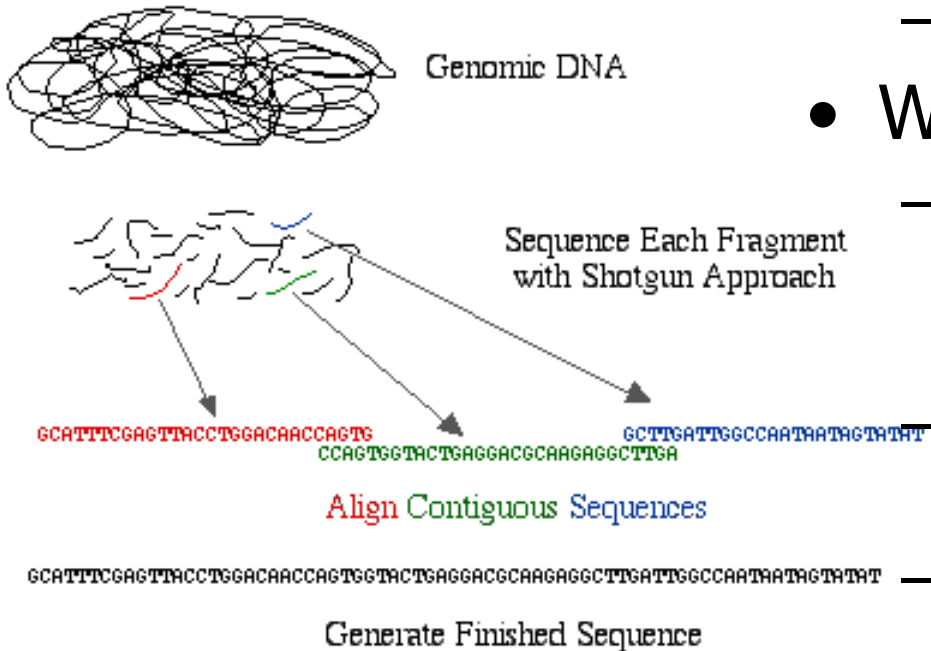


- Mehrstufiges Verfahren
- Kartierung mit YACs und BACs
- Berechnung von BAC Contigs
 - STS Anchoring
- Shotgun - Sequenzieren der BACs

Quelle: <http://www.geneticsplace.com>

Whole Genome Shotgun (WGS)

Whole Genome Shotgun Sequencing Method



- Beobachtung
 - Sequenzierung immer billiger
 - Mapping bleibt teuer
- Whole Genome Shotgun
 - Zerbrechen von **kompletten Genomen** in Stücke 1KB-100KB (keine Kartierung)
 - Alle Bruchstücke (an)sequenzieren
 - Menschliches Genom: 28.000.000 Sequenzen (Celera)

Vergleich

- WGS versus HGP - Ansatz
 - Billiger, schneller
 - Viele Gaps, da Kandidaten zum Gap-Closing nicht vorhanden
 - Mehr Schwierigkeiten mit Repeats (weniger „Scaffolding“)
 - Immenser Aufwand für ausreichend hohe Überdeckung
 - Draftsequenzen
- Kombination beider Methoden

Teil II. Sequenzierungstechnik

- Methode nach Sanger
- Verarbeitungsschritte
 - Base Calling
 - Assembly
 - Finishing

Aufgabe

- Gegeben: Clone unbekannter Sequenz
- Gesucht: Sequenz

- Unmöglich: Ansehen, Messen, Mikroskop, etc.
- Verfahren von Sanger, 1972: „**Radioactive Dideoxy Sequencing**“
- Dissertation 1997 ...

Heute

- Fluoreszente Markierung
- Hochdurchsatz
- Billig

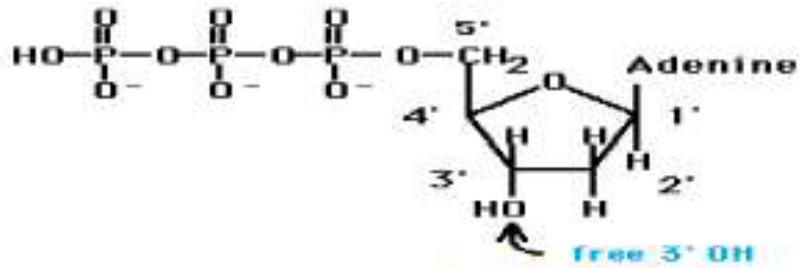


Quelle: <http://www.geneticsplace.com>

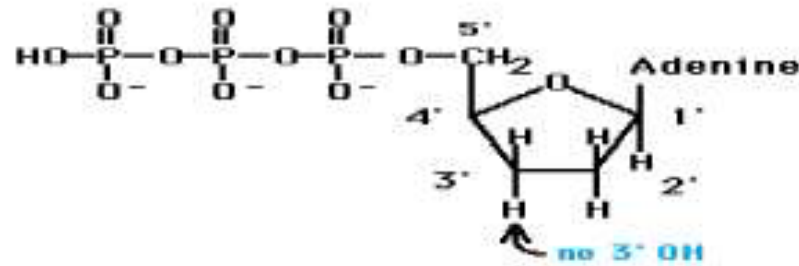
Sequenzierung nach Sanger

- Voraussetzungen
 - Sequenz hat einen definierten Anfang
 - Teil des Clonierungsvektors
 - Dient als Bindungsstelle für Primer und damit als Sequenzierungsanfang
 - Polymerase
 - Bindet an doppelsträngigen Abschnitt
 - Verlängert einsträngige DNA entlang Template
- Deoxy versus Dideoxy Nucleotide
 - DNA besteht aus Deoxy Nucleotiden (dNTP)
 - Einbau von Dideoxy Nucleotiden (ddNTP) möglich
 - ddNTP stoppt Polymerase

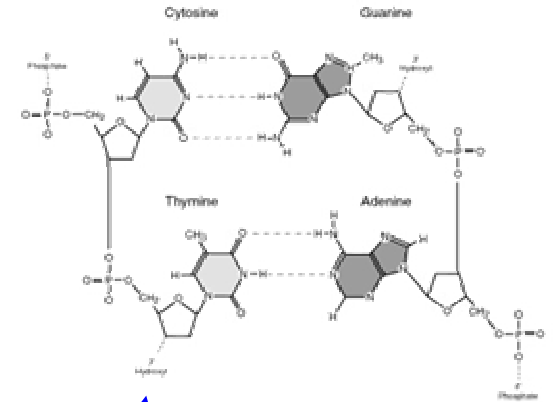
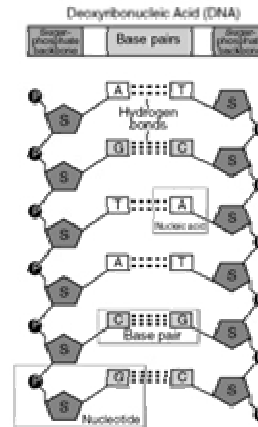
Struktur von dNTP und ddNTP



dNTP



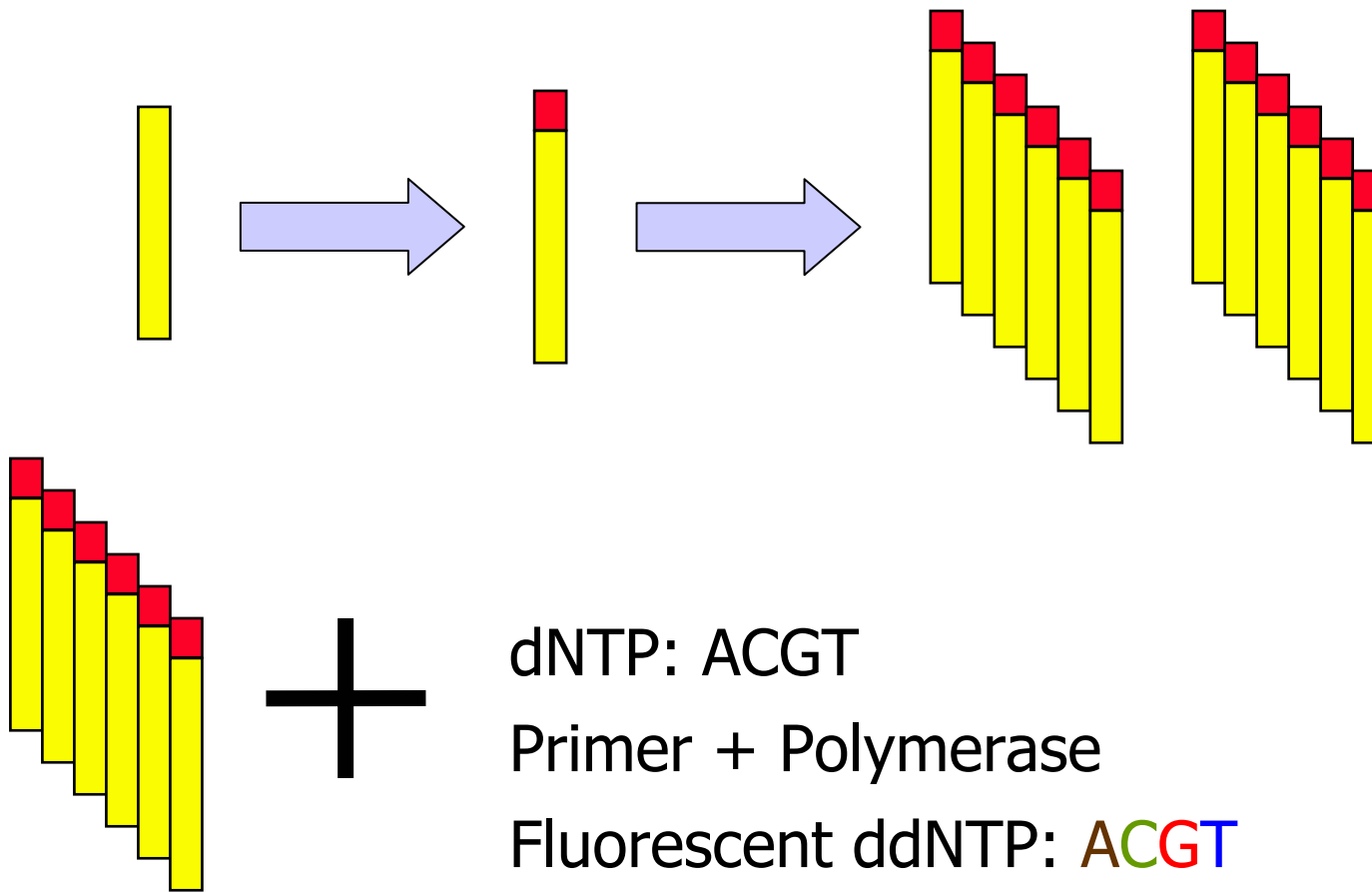
ddNTP



- Dideoxy-Base: keine freie OH Gruppe
- Dideoxy-Base wird eingebaut
- Danach können keine weiteren Basen mehr angehängt werden
- Polymerase fällt ab



Schritt 1 und 2



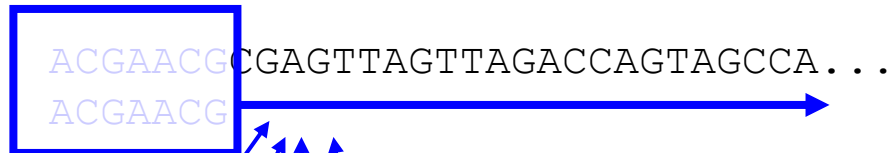
Schritt 3

Primer

Template

ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

Polymerase



ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

ACGAACGCGAGTT**A**

ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

ACGAACGCGA**G**

ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

ACGAACGCGAGTTAGT**T**

ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

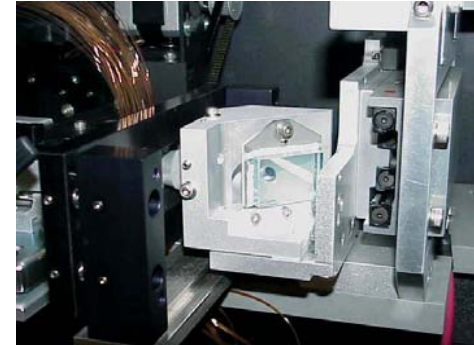
ACGAACGCGAGTTAGTTAG**T**

ACGAACGCGAGTTAGTTAGACCAGTAGCCA...

ACGAACGCG**A**

Schritt 4

Laser & Detektoren



ACGAACGCGAGTT**A**
ACGAACGCGA**G**
ACGAACGCGAGTTAGT**T**
ACGAACGCGAGTTAGTTAG**T**
ACGAACGCG**A**

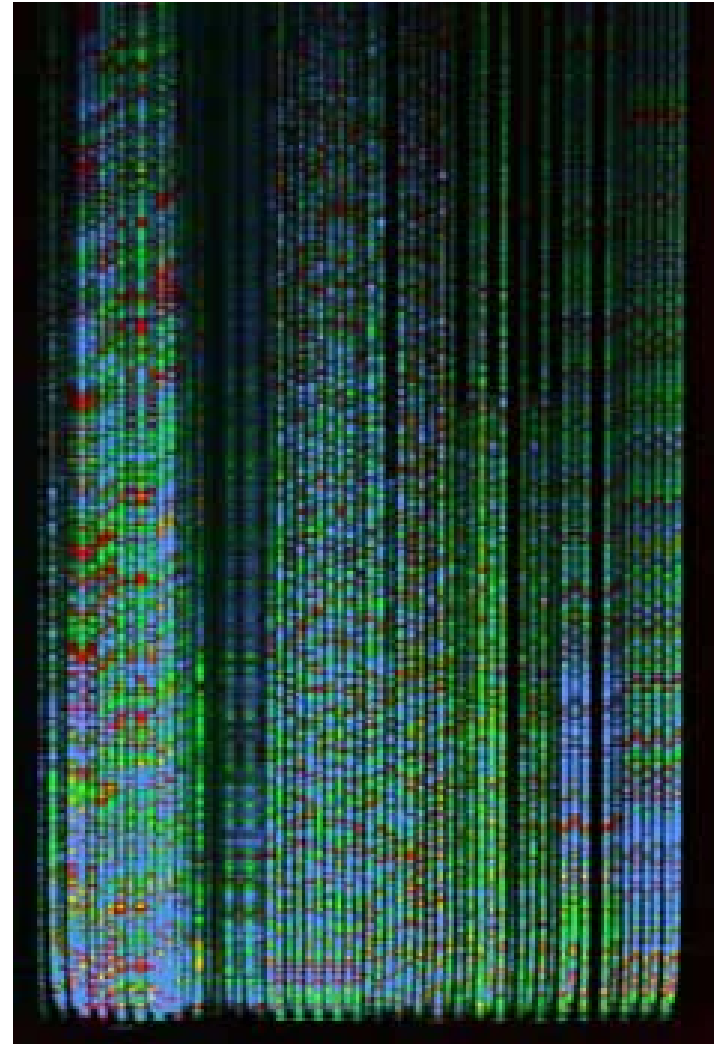
Gel / Kapillar Elektrophorese

ACGAACG**C**
ACGAACGC**G**
ACGAACGCG**A**
ACGAACGCGA**G**
ACGAACGCGAG**T**
ACGAACGCGAGT**T**
ACGAACGCGAGTT**A**
ACGAACGCGAGGTTA**G**



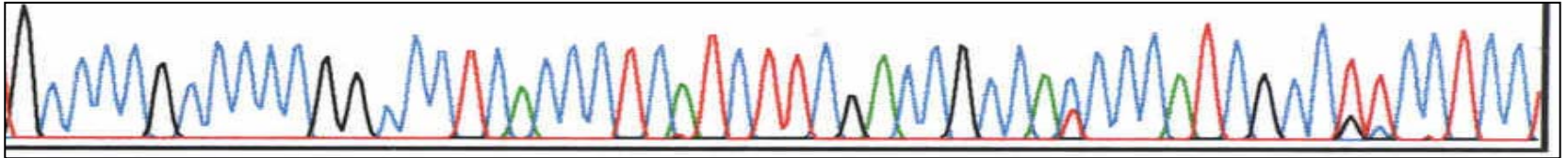
Ergebnis (roh)

- Heutige Geräte
 - >36 Läufe parallel
 - Kapillarelektrophorese (statt Gelen)
 - Direktes Laden von 96 Well Plates
- Sanger
 - Radioaktive Markierung
 - 4 Mischungen (A,G,T,P)
 - 4 Gel - Lanes

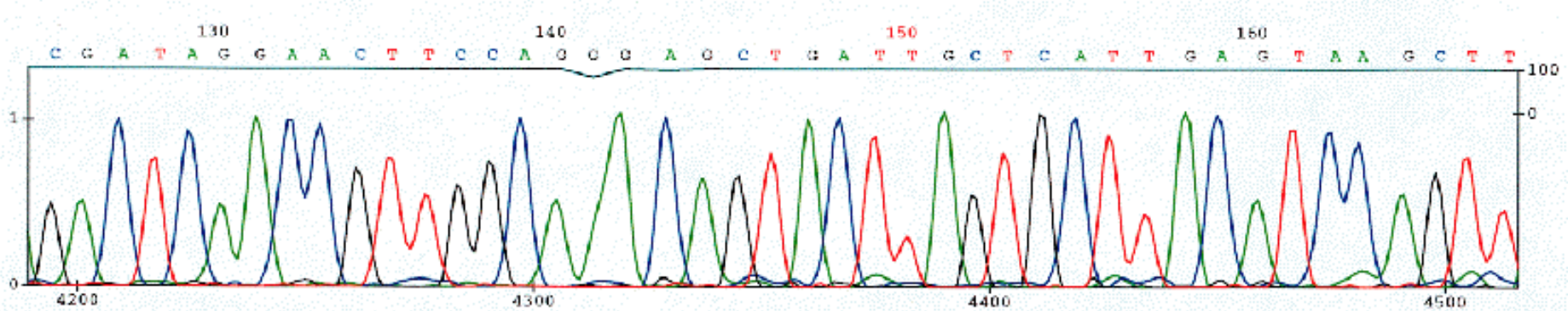
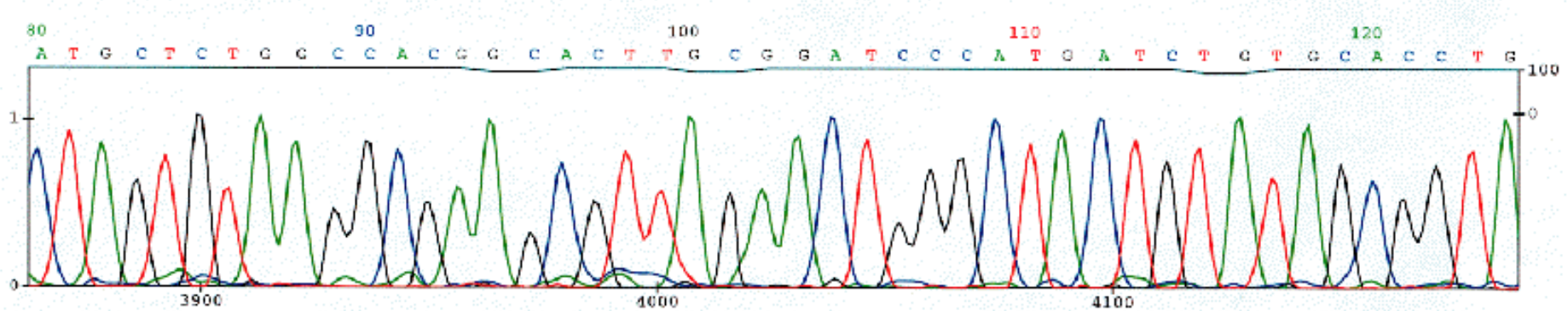
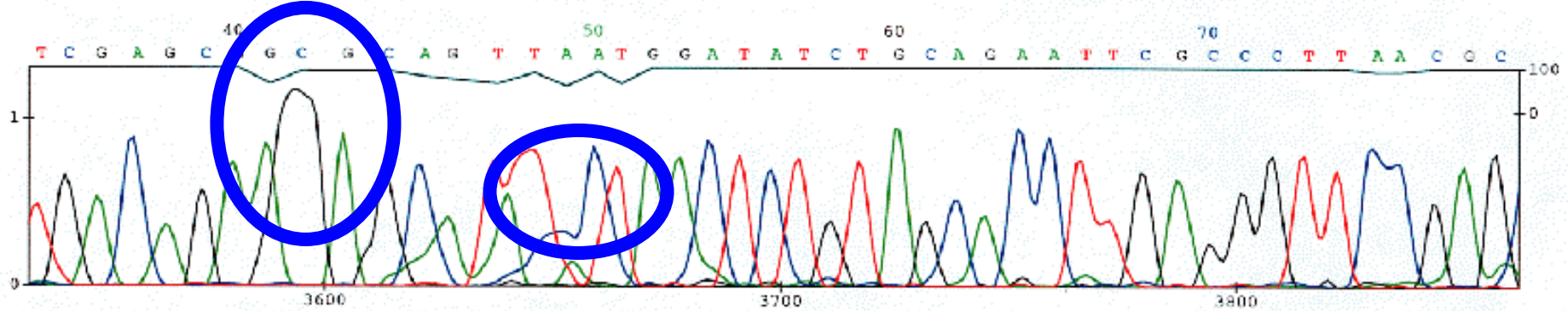
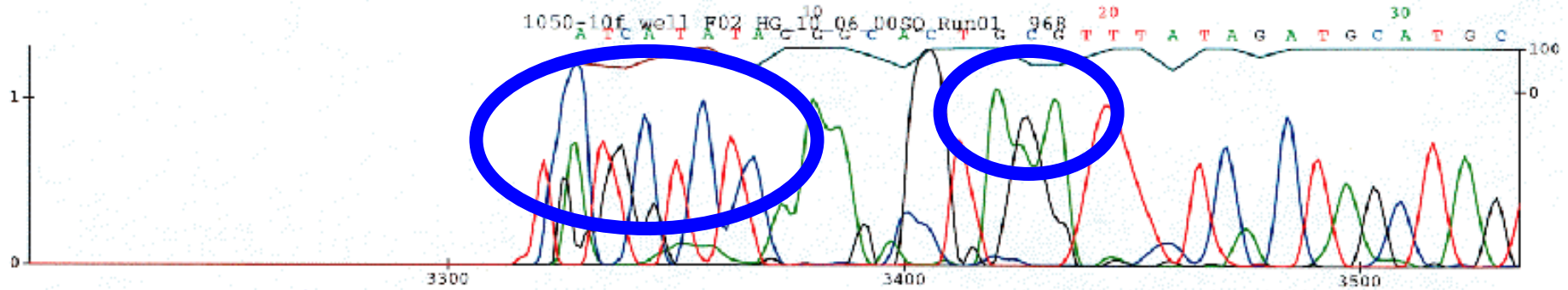


Ergebnis (Zwischenprodukt)

- Signalverarbeitung (Rauschen, ...)



- Übersetzung in Traces
 - 4 Arrays, jedes für eine Farbe
 - Intensitätswerte in regelmäßigen Zeitabschnitten
- Theoretisch
 - Peaks entdecken
 - Immer nur eine Farbe
 - Sequenz zuordnen



Einschub: NCBI Trace Archive

Trace Archive

Search for alt

Trace Archive Querying

The Trace Archive has been developed to store the raw data underlying all of the sequence generated by the genome project. We will be exchanging data regularly with the [Ensembl Trace Server](#).

[Wed Apr 2 11:10:28 2003 EDT] NCBI Trace Archive [Details](#)

presently contains:

Organism	Count
ANOPHELES GAMBIAE	4,888,159
APIS MELLIFERA	2,105,822
ARTIBEUS JAMAICENSIS	8,661
ASPERGILLUS NIDULANS	160,371
ATELERIX ALBIVENTRIS	30,725
BACILLUS ANTHRACIS STRAIN A2012	65,507
BACILLUS CEREUS ATCC 10987	80,024
BOS TAURUS	466,691
BRASSICA OLERACEA	173,145
BRUCELLA SUIS 1330	37,872
BURKHOLDERIA MALLEI ATCC 23344	83,872
CAENORHABDITIS BRIGGSAE	2,354,917

Probleme mit Traces

- Low Quality Bereiche <50, >400-1000
 - Kurze Fragmente laufen unregelmäßig – Anfang schlecht
 - Lange Fragmente sukzessive schlechter (relativer Massenunterschied geringer, weniger Fragmente)
- Typische Probleme
 - Keine klaren Peaks erkennbar
 - Mehrere gleich starke Peaks
 - Periode der Peaks verschoben
- Probleme abhängig von
 - Sequenz (GC Content, Repeats, etc.)
 - Position im Read

Vom Tracefile zur Sequenz

- Tracefiles sind Rohdaten der Sequenzierung
- Verschiedene Verfahren / Tools, um aus Tracefiles Sequenzen zu berechnen
- Drei Schritte
 - **Base Calling:** Abbildung von Traces auf Basensequenzen
 - **Assembly:** Zusammenfügen der Basensequenzen zu Gesamtsequenz
 - **Finishing:** Füllen von Lücken und unklaren Bereichen

Base Calling in PHRED (Sketch) [EHWG98]

- PHRED: Heuristischer, hochgradig getunter Algorithmus
- Schritt 1: Berechnung der erwarteten Peakpositionen
 - Erster Schritt: Finden von vermutlichen Peaks
 - Berechnung der durchschnittlichen Peakabstände
 - Sliding Window von 200 Tracepoints (zum Ausgleich sich kontinuierlich ändernder Perioden)
 - Bestimmung der Periode durch Sinuskurve
 - Erwarteter Peak = Zentrum der Sinuskurve
 - Iteratives Verschiebung des Fensters

PHRED 2

- Schritt 2: Finder von (beobachteten) Peaks im Trace
 - Suche konkave Tracepoints pro Array
 - Mindestbedingungen an Peakhöhe und benachbarte Peaks
- Schritt 3: Zuordnung der erwarteten zu den beobachteten Peaks
 - Phase 1: Finden direkter Entsprechungen (mit min. Fläche und max. Abweichung)
 - Phase 2: Alignierung nicht zugewiesener Peaks (dynamische Programmierung, Minimierung des Gesamtfehlers)
 - Phase 3: Zuordnung des Rests (diverse Heuristiken: Nachbarn, Mindestfläche, etc.)

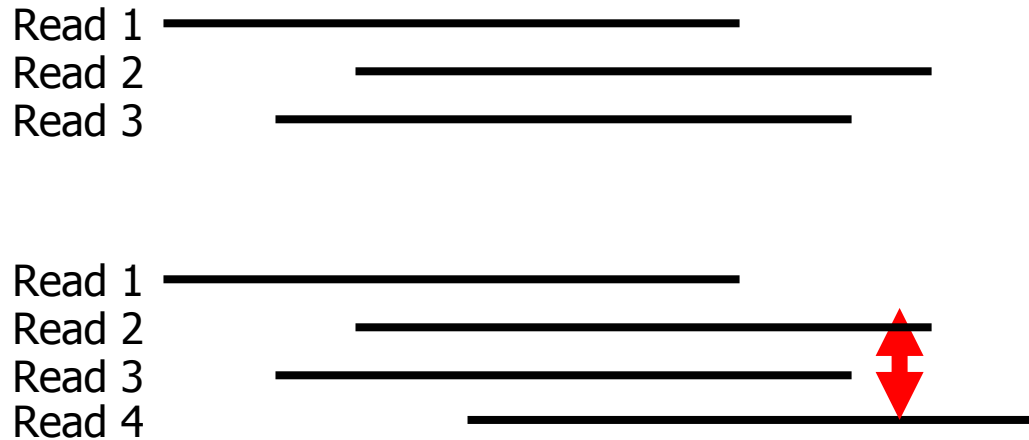
PHRED 3

- Schritt 4: Resteverwertung
 - Einfügung beobachteter, nicht zugewiesener Peaks in Sequenz (z.B. bei starker Periodenverschiebung)
- Vorteile von PHRED
 - Erstmals (1998) bessere Erkennung als ABI Software
 - Besonders gut in den **High-Quality Areas** (wichtig für HGP, nicht wichtig für EST Sequencing)
 - Berechnung aussagekräftiger **Qualitätswerte pro Base** (wichtig für nächste Schritte: Assembly und Finishing)

Assembly

- Szenario: Shotgun Sequenzierung
- Ergebnis des Base Calling: Einzelne Reads
- Gesucht: Gesamtsequenz
- ... bzw.:
 - zusammenhängende Stücke (Contigs)
 - möglichst sichere Sequenz (Redundanz!)
- **Assembly**: Berechnung der „Konsensussequenz“

Assembly



- Überlappung ist gut (Redundanz) und schlecht (potentielle Konflikte)
- **Typische geforderte Überlappung im HGP: 7-10 Reads**
- Alignments immer an mehreren Stellen möglich
- Greedy Ansatz nicht optimal

Assembly – Abstrakte Formulierung

- **SUPERSTRING**

- Geg.: Menge S von Strings

- Ges.: String T so, dass

- (a) $\forall s \in S: s \in T$ (s Substring von T)

- (b) $\forall T',$ für die (a) gilt, gilt: $|T| \leq |T'|$ (T ist minimal)

- **NP-vollständiges Problem**

- **Verschärfungen**

- Fehler in Sequenzen (s „ungefähr Substring“ von T)

- Zwei Orientierungen von s möglich

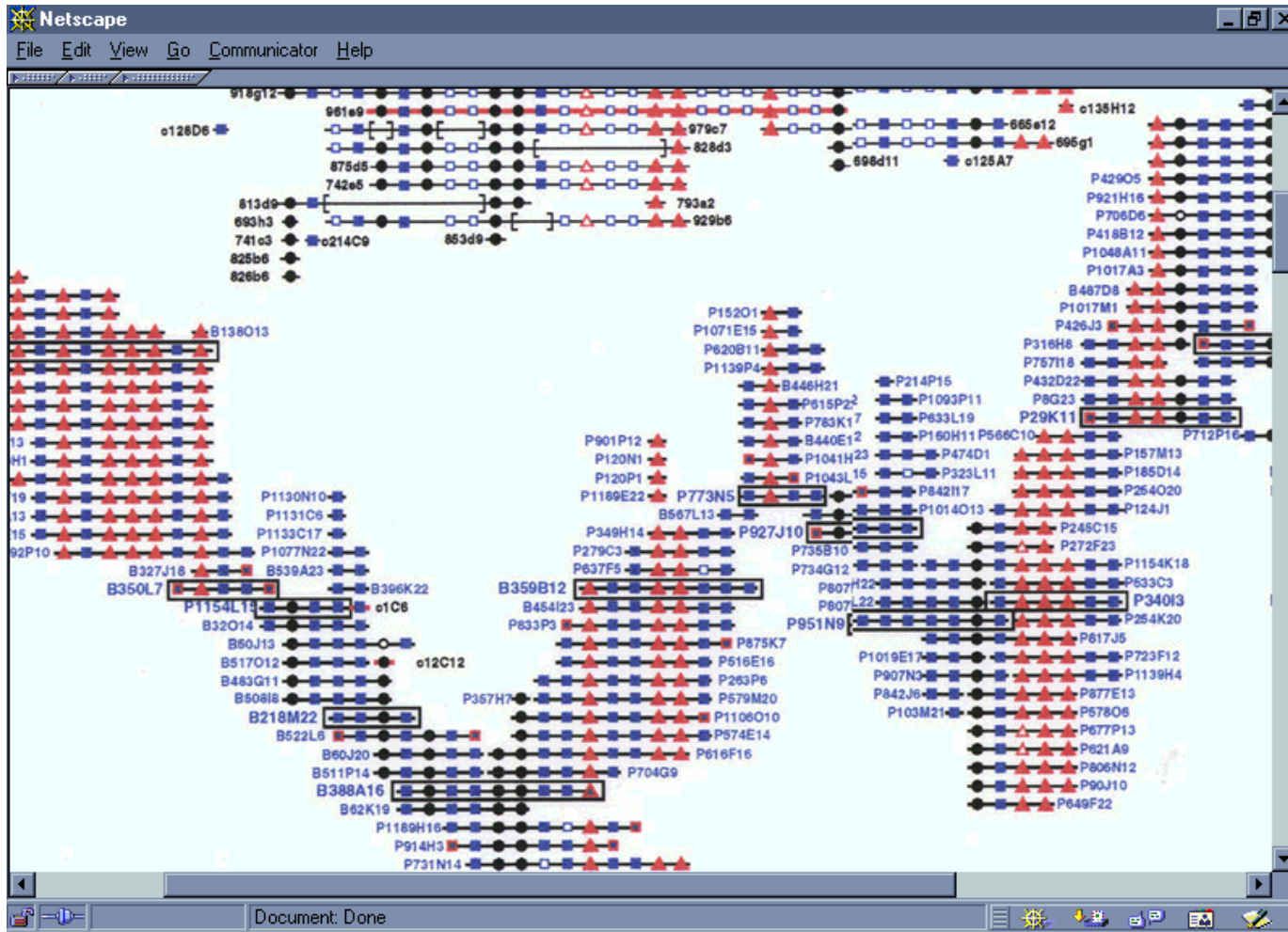
- Gaps

- **Heuristische Verfahren**

Algorithmus [KM95]

- Einer von vielen (PHRAP, CAP3, GAP4, ...)
- 4 Schritte
 - Berechnung von Sequenzüberlappungen unter Zulassung einer festen Fehlerrate. Darstellung als Graph mit Knoten (Fragmente und Komplementäre) und gewichteten Kanten (den Fehlerraten)
 $O(n^2)$
 - Orientierung der Fragmente festlegen. Löschen von Knoten im Graphen so, dass Summe der Kanten möglichst klein wird
NP-vollständig
 - Fragmente anordnen. Auswahl von Kanten so, dass die Summe der Kanten möglichst klein ist
NP-vollständig
 - Konsensussequenz durch Multiple Alignment berechnen.
Je nach Methode auch NP-vollständig
(aber nur in maximalen Zahl überlappender Fragmenten – die "dicksten" Stelle im Layout)

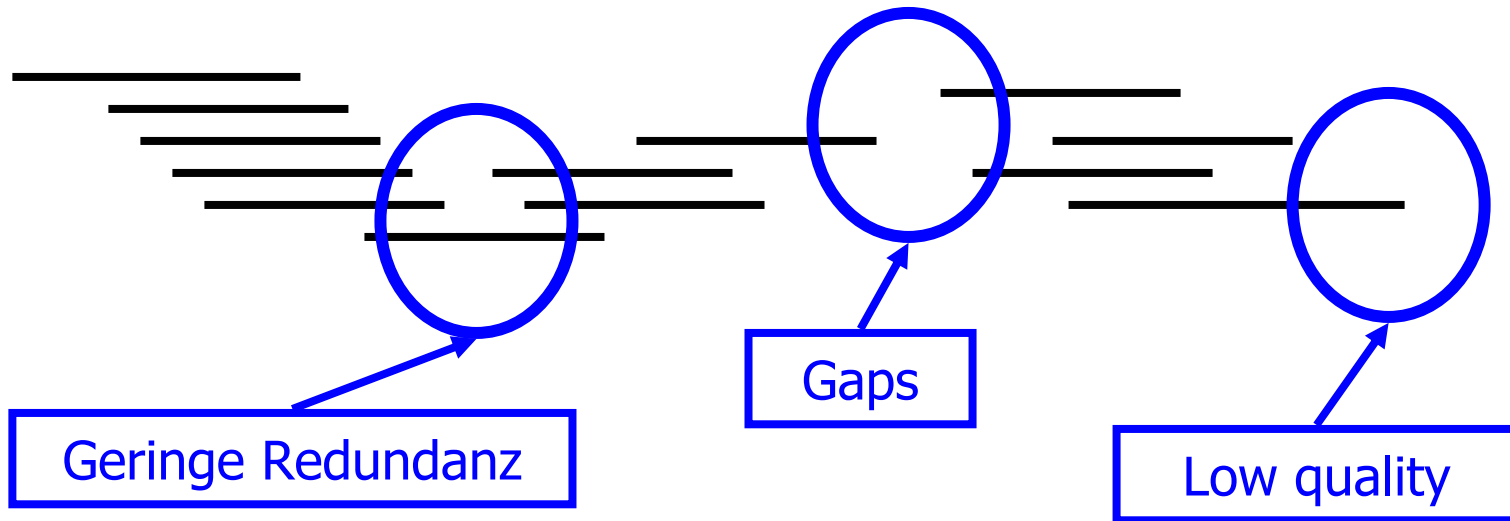
Resultat



Finishing

- Ergebnis des Assembly
 - Menge von Contigs
 - Offenbleibende Probleme
 - Löcher zwischen Contigs (Gaps)
 - Regionen mit geringer Überdeckung
 - Regionen geringer Basenqualität
 - Ungelöste Basen - Mismatches
- **Finishing**
- Auswahl von weiteren Experimenten
 - Automatische/manuelle Korrektur von Mismatches

Finishing – Neue Reads



- Auswahl (möglichst weniger) neuer Reads
 - Design spezifischer Primer
 - Gezielte Selektion von Clonen aus Shotgun–Bibliothek
 - Sequenzierrichtung / - technik ändern
 - Neue Reads – neues Assembly

Finishing - Editing

Read 1: AGAAAGAGA

Read 2: AGAAT-AGA

AGAAAGAGA

AGAAAGAGA

AGAAAGAGA

AGAAAGAGA

AGAAT-AGA

AGAAT-AGA

AGAAT-AGA

AGAAT-AGA

AGAAAGAGA

AGAATAGA

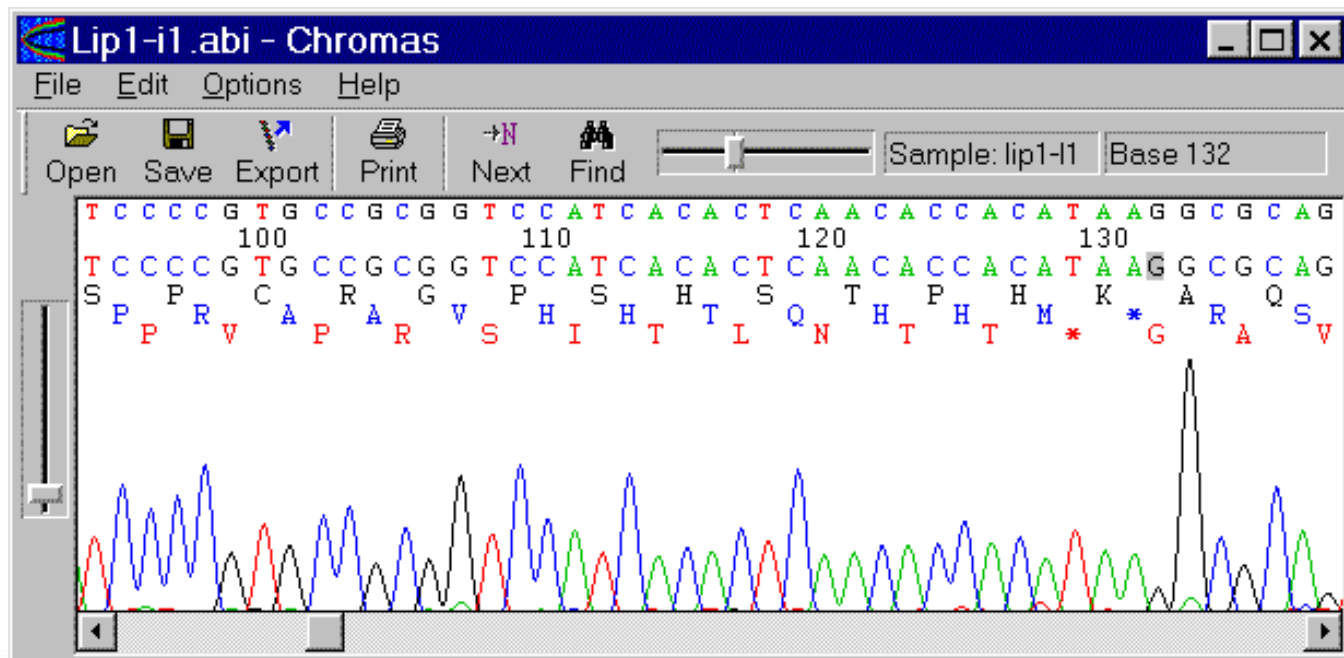
AGAATGAGA

AGAA *GAGA

- * Unknown, - Gap
- **Blau**: Wahrscheinlich falsch, **Rot**: Consensus

Finishing – Ergebnisse

- Finishing nach wie vor teilweise manuelle Arbeit
- Ein paar Lücken bleiben immer
 - Stellen nicht sequenzierbar (GC-reich, Repeats, etc.)
 - Stellen nicht clonierbar (Instabil: Centromere, Interferenz mit Wirtszellen, etc.)



Zusammenfassung

- Basis-Sequenziertechnik: „ Dideoxy Sequencing“
- Viele Teilschritte
 - Cloneerstellung
 - Probenaufbereitung (Shotgun)
 - Sequenzieren
 - Base Calling, Assembly, Finishing
- Extreme Verbilligung in 10 Jahren
 - Kosten von 10 USD auf unter 20 Cent / Base
- Neue Techniken: Kapillare, Massenspektrometer

Teil III. Datenmodelle

- Daten
- Datenmodelle
 - BioSQL
 - EMBL
- Datenbanken
 - EMBL / SRS
 - Genbank / Entrez

Rohdaten

- Daten über den Sequenzierungsprozess
 - Geräterohdaten (Spektren)
 - Trace files
 - Assemblies, Edits, Contigs, Gaps
 - Benutzte Programme
 - Labordaten (Maschinen, Personal, Datum, ...)
- NCBI Trace File Archive
- Viele Sequenzierzentren
 - Sanger
 - University of Washington
 - ...

Sequenzen

- Technische Herkunft
 - Wer, wann, wie, Methode, ...
 - Fehlerrate
- Biologische Herkunft
 - Clone, Organismus, Linie, ...
- Literaturreferenzen
- Sequenz
- Annotation: **Feature der Sequenz**
 - Wird durch Analyse gewonnen
 - Suche nach ähnlichen Sequenzen, Vorhersage von Genen, Berechnung von Repeats oder Verteilungsauffälligkeiten, Erkennen fester Signale (START, STOP), etc.
 - Können stark abweichen
 - Genbank/EMBL: Feature werden vom Autor bestimmt

Feature

- Bestandteile
 - **Location**: Start - Ende, Gaped, Genau - Ungenau
 - **Key**: CDS, Repeat, RNA-Strukturen, Homologe Sequenzen, Promotoren, Marker, Exon/Intron Boundaries, Funktion, Motiv, Polymorphismus, ...
 - **Qualifier**: Ergänzungen, z.B. kodiertes Protein, Splicevariante, Regulationsmechanismen, ...
 - **Label**: Name des Features (Referenzierung)
- EMBL Featuretable:
 - Ca. 50 Typen von Features (keys), hierarchisch organisiert
 - Jeweils 1-20 Qualifier, tw. mit Controlled Vocabularies

EMBL Beispiel

```
FH      Key                Location/Qualifiers FH
FT      source             1..756
FT      /db_xref="taxon:1638"
FT      /organism="Listeria ivanovii"
FT      /strain="ATCC 19119"
FT      RBS                95..100
FT      /gene="sod"
FT      terminator        723..746
FT      /gene="sod"
FT      CDS                109..717
FT      /db_xref="SWISS-PROT:P28763"
FT      /transl_table=11
FT      /gene="sod"
FT      /EC_number="1.15.1.1"
FT      /product="superoxide dismutase"
FT      /protein_id="CAA45406.1"
FT      /translation="MTYELPKLPYTYDALEPNFF
FT      HAELASKPGEELVANLDSVPEEIRGAVRNHGGGH
FT      IESEFGTFDEFKEKFNAAAAARFGSGWAWLVVNN
FT      DVWEHAYYLKFKQNRPEYIDTFWNVINWDERN"
```

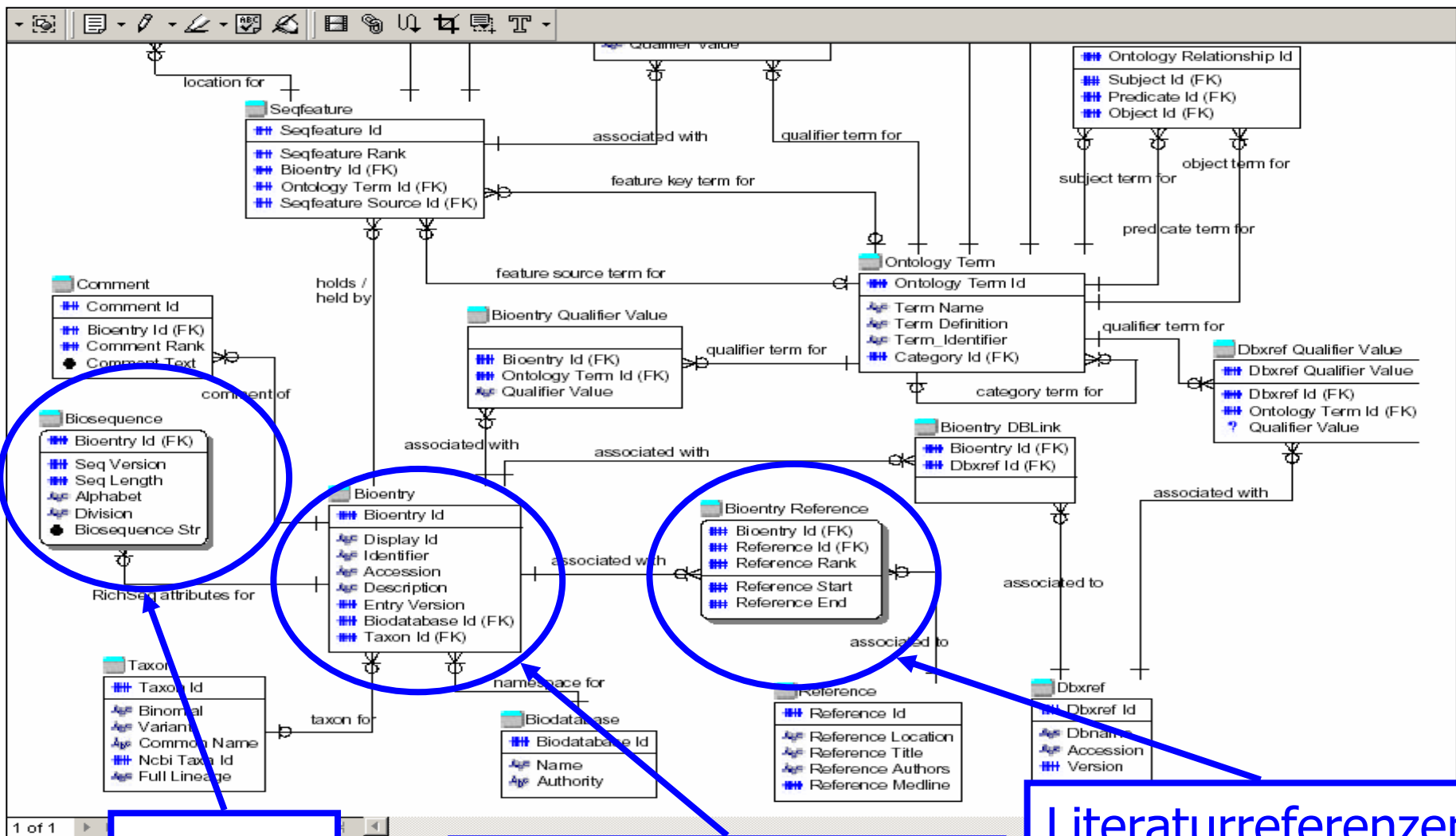
Modellierung

- 2 Möglichkeiten
- **Metamodellierung**
 - Generische Klassen „Feature“, „Term“, „Qualifier“, ...
 - Beispiel: BioSQL
- **Ausmodellieren**
 - Explizite Klassen „RBS“, „Terminator“, „CDS“, ...
 - Beispiel: EMBL Objektmodell

Beispiel: BioSQL

- OpenSource Projekt
- Relationale Schema (Oracle, PostGres, mySQL)
- Parser (Perl)
- EMBL & SwissProt
- Ca. 20 Tabellen

BioSQL Core



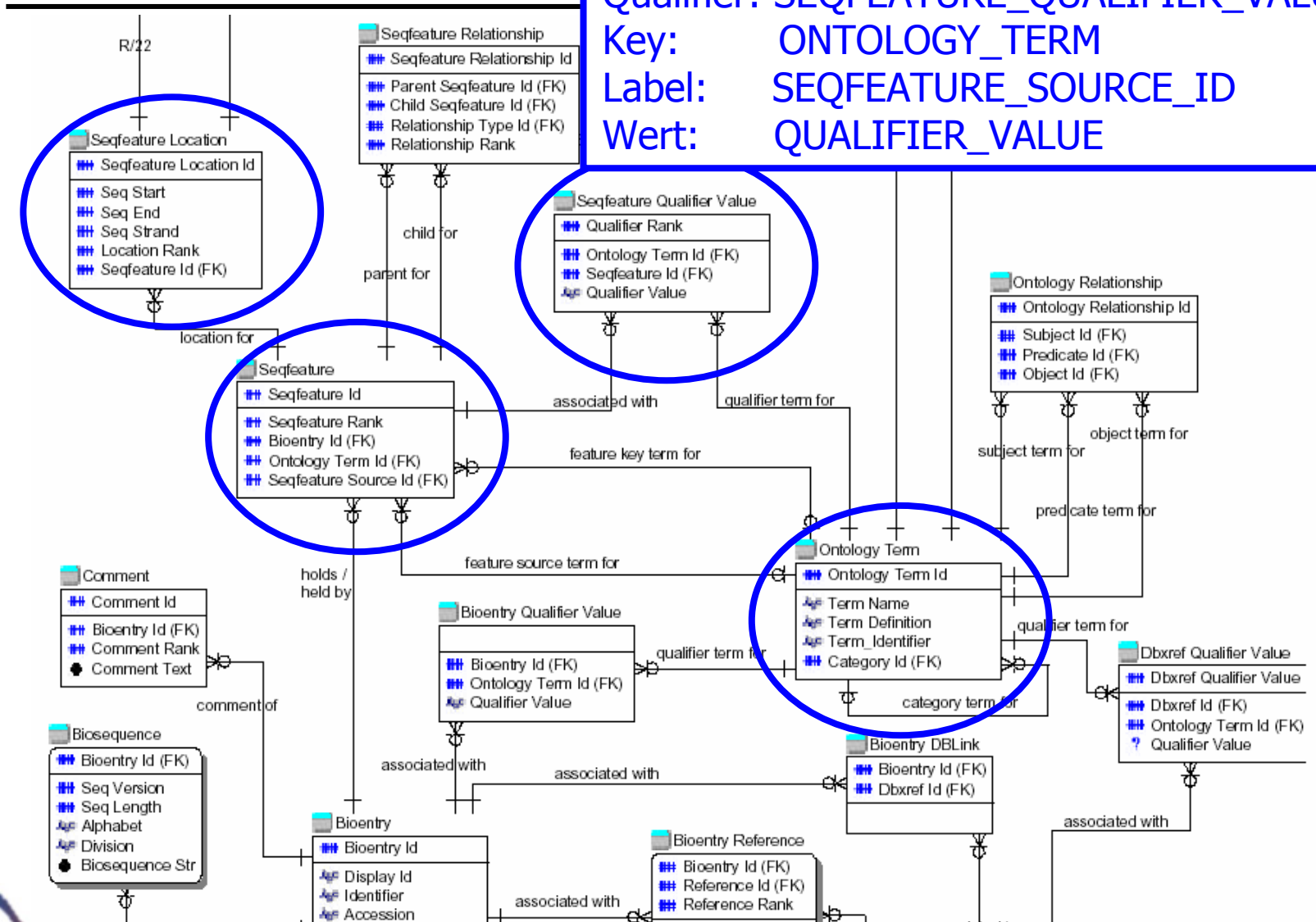
Sequenz

Entry mit IDs / Version

Literaturreferenzen

BioS

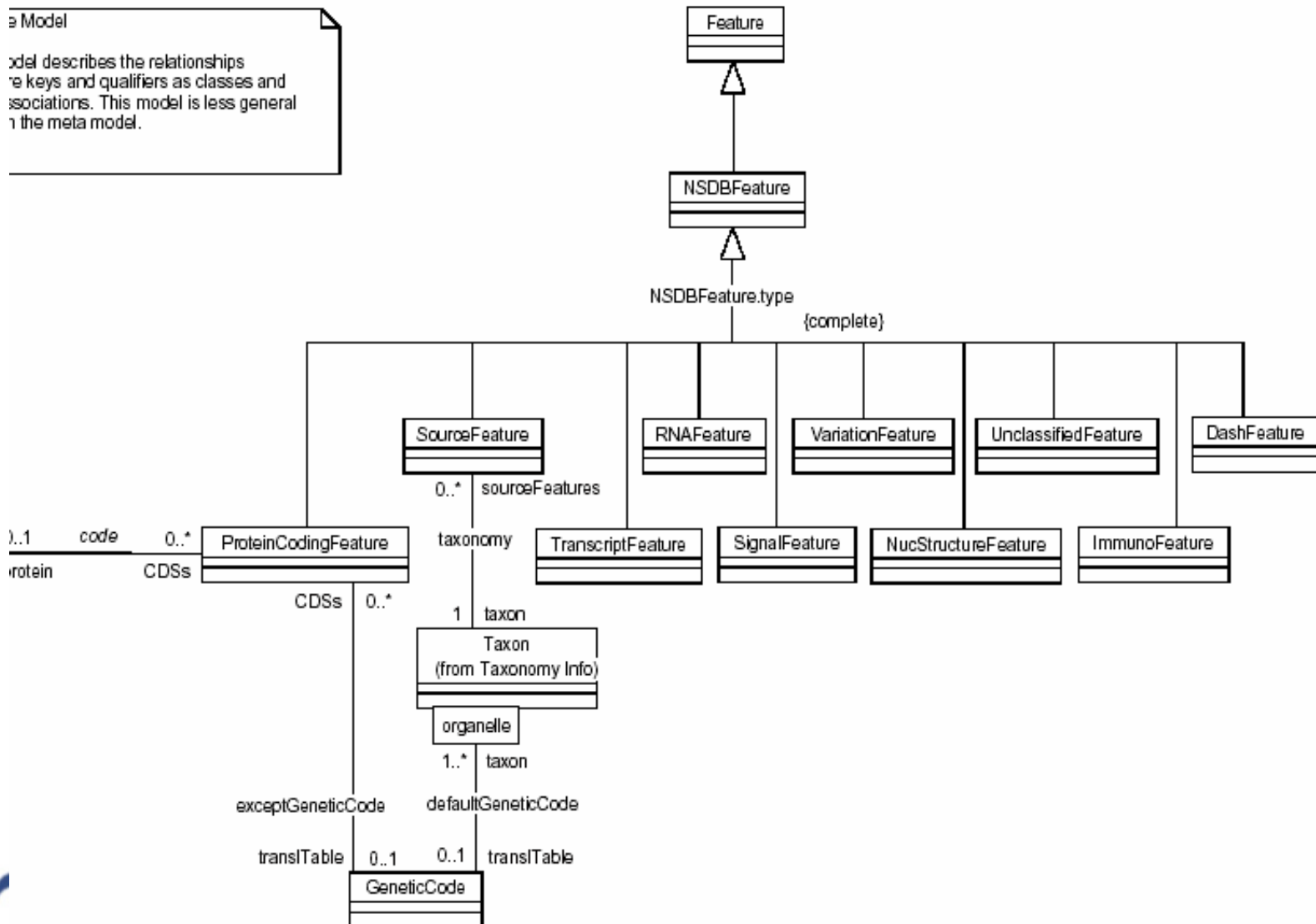
Feature von BIOENTRY_ID
 Location: SEQFEATURE_LOCATION
 Qualifier: SEQFEATURE_QUALIFIER_VALUE
 Key: ONTOLOGY_TERM
 Label: SEQFEATURE_SOURCE_ID
 Wert: QUALIFIER_VALUE



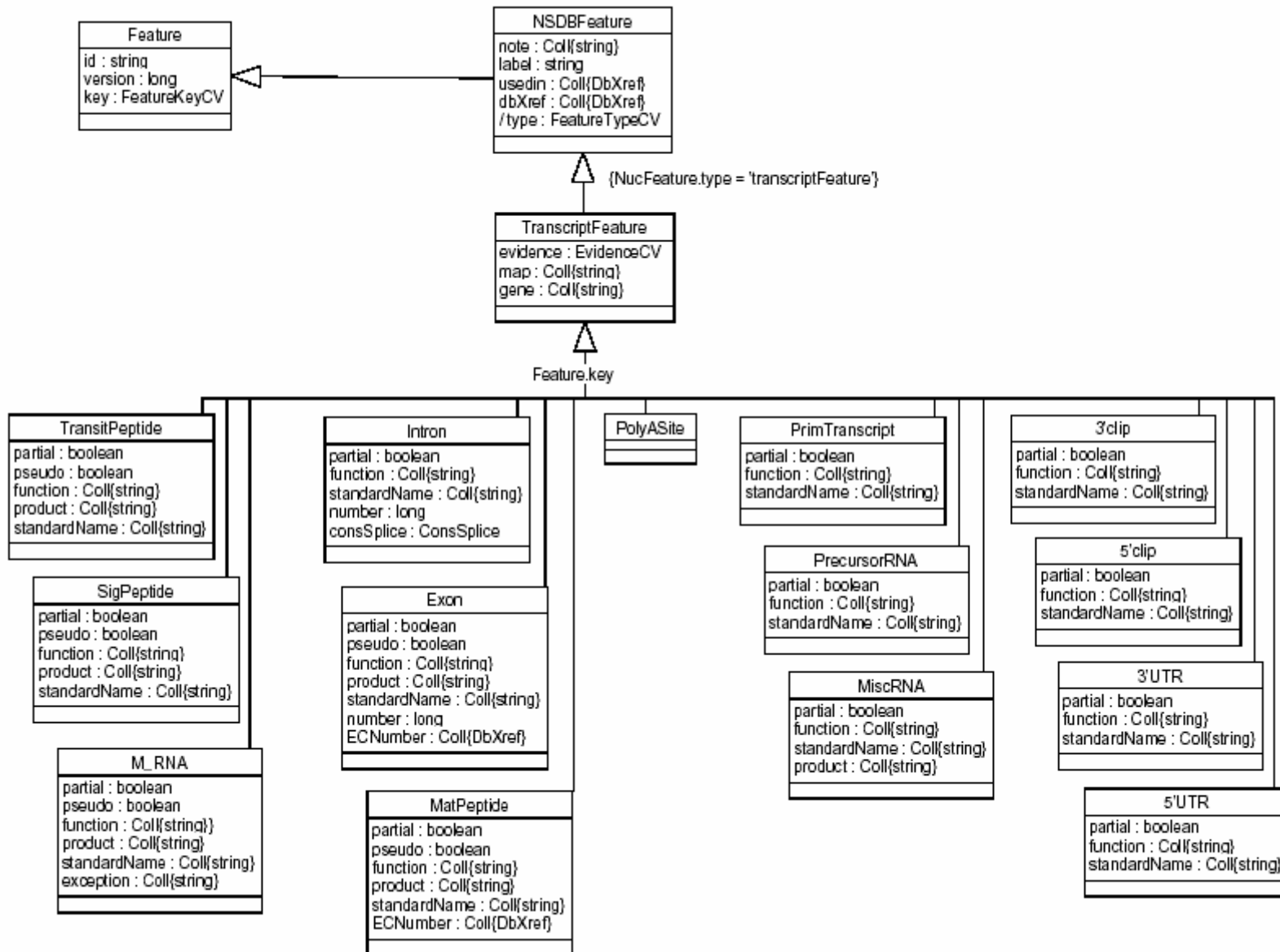
Beispiel: EMBL

Model

Model describes the relationships between keys and qualifiers as classes and associations. This model is less general than the meta model.



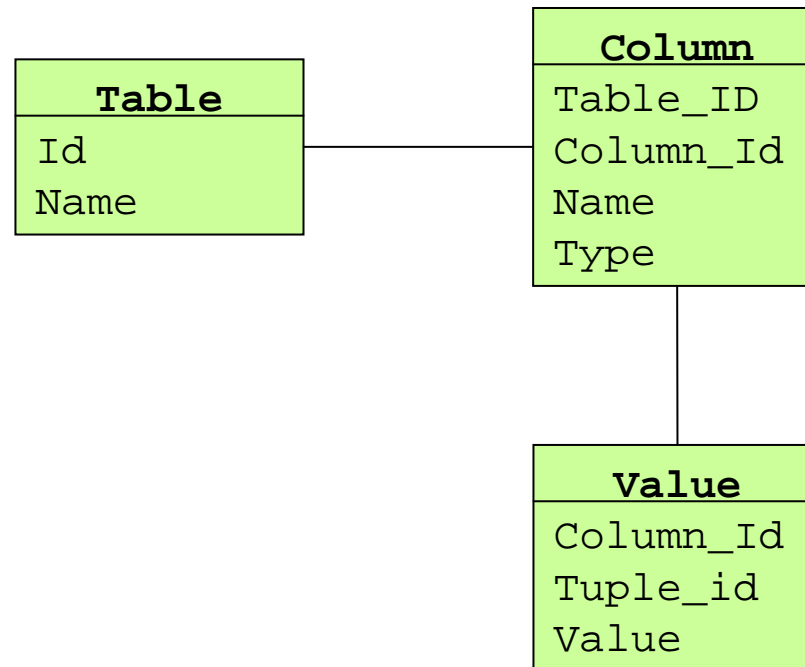
EMBL Feature 2



Zusammenfassung

- Interessanter Teil der Modelle: Feature
- Vorteile Metamodellierung
 - Weniger Tabellen
 - Einfach erweiterbar (INSERT statt CREATE TABLE)
 - Generische Beziehungen
 - Feature-Keys etc. sind Werte – Zugriff über SQL
- Nachteile
 - Langsamer (Größere Tabellen)
 - Unübersichtlich
 - Keine Feature-spezifischen Felder, Datentypen, Constraints
- Modelle sehr ähnlich für Proteinsequenzen
 - SP relationales Schema basiert auf dem von EMBL

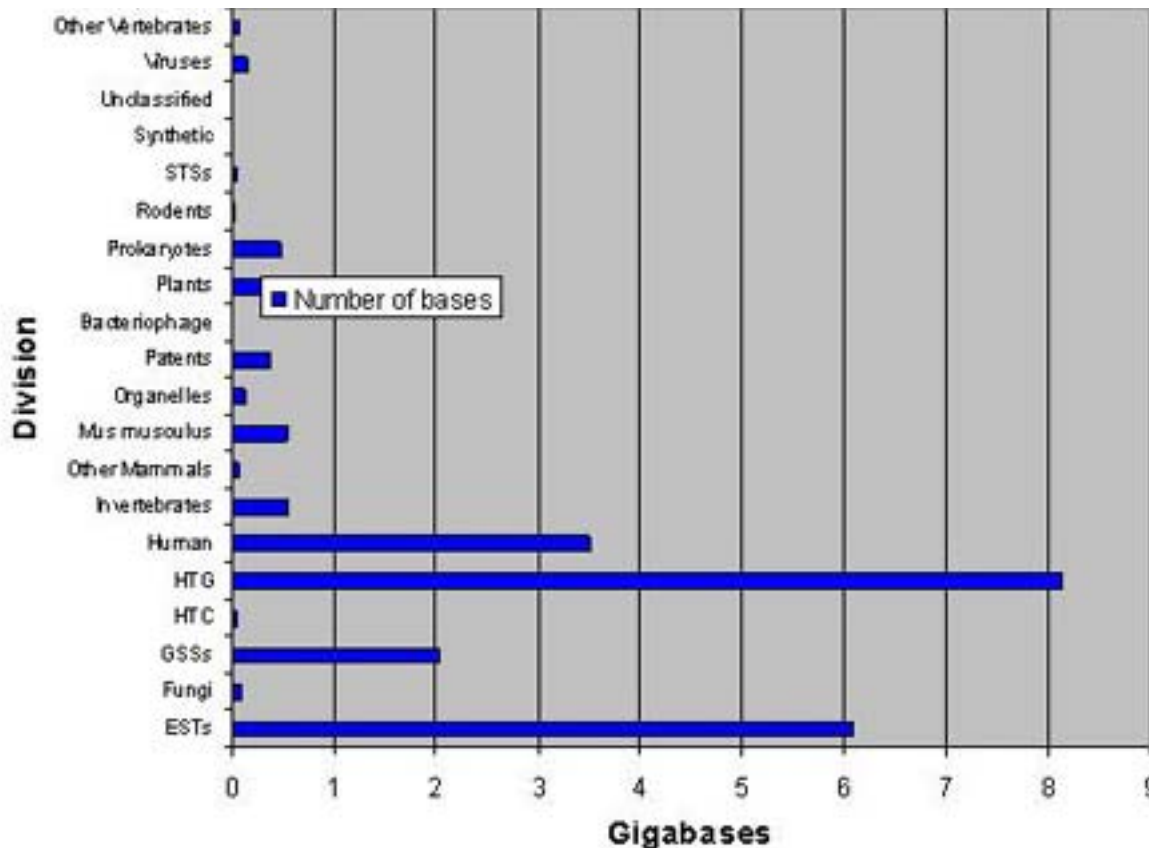
Generischstes relationales Schema



EMBL

- Identischer Datenbestand zu Genbank / DDBJ
 - Nächtlicher Austausch
 - Formatunterschiede
- Submission – based
- Archiv – (fast) keine Bearbeitung der Daten
- Verfügbarkeit
 - Flatfile
 - SRS
 - XML (BSML, ?)
 - Oracle Dump Files

EMBL Divisions



- Tw. Spezies-übergreifend, tw. species-spezifisch
- Exponentielles Wachstum
- Accession-Num und Version-ID

Quelle: <http://www3.ebi.ac.uk/Services/DBStats>, Stand 10.2.2003
EMBL Release 72, Total: 32 GBases

SRS: Sequence Retrieval System

- EMBL – EBI – Lion AG
- Komponenten
 - Flatfile Indexierungssystem
 - WWW Interface
 - Anfragesprache
 - Administrationstools
- Parser für >1000 DB, EBI-Installation: ~150 DBs
- Joins
 - Verfolgen externer Datenbankreferenzen
 - Equi-Joins

Genbank

- Modell in ASN.1
- Datenspeicherung: Relational (?)
- Zugriff: Entrez
 - Ähnlich SRS
 - Keine Joins
 - „Neighbours“ – „Related Documents“
 - Click-And-Browse

Weitere Datenbanken

- Spezielle Auswahl von Sequenzen
 - Organismen (Yeast, Fly, Mouse, HIV, ...)
 - Ribosomen, Immunsystem
 - Transcriptionsfaktoren, Promotoren, ...
- Terminologie
 - GeneOntology (> 7000 Begriffe: Funktion, Prozess, Zelllokation)
 - NCBI Taxonomy Database (119000 Org.)
- ...

Literatur

- [KM95] Kececioglu, J. and Myers, E. W. (1995). "Combinatorial Algorithms for DNA sequence assembly." *Algorithmica* **13**(1/2): 7-51.
- [EHWG98] Ewing, B., Hillier, L., Wendl, M. C. and Green, P. (1998). "Base-Calling of Automated Sequence Traces using PHRED. I. Accuracy Assesment." *Genome Research* **8**: 175-185.
- [PWS02] Pontius, J. U., Wagner, L. and Schuler, G. D. (2002). UniGene: A Unified View of the Transscriptome. Book "UniGene: A Unified View of the Transscriptome", National Library of Medicine, Bethesda
- [BO01] Baxevanis, A. D. and Ouellette, B. F. F., Eds. (2001). "Bioinformatics. A Practical Guide to the Analysis of Genes and Proteins", John Wiley & Sons. (Kapitel 12+13)
- [Sen02] Sensen, C., Ed. (2002). "Essentials of Genomics and Bioinformatics", Wiley-VCH, Weinheim. (Kapitel 2+7)