

# Molekularbiologische Datenbanken

Kartierung von Chromsomen II  
Algorithmen und Datenmodelle



Ulf Leser

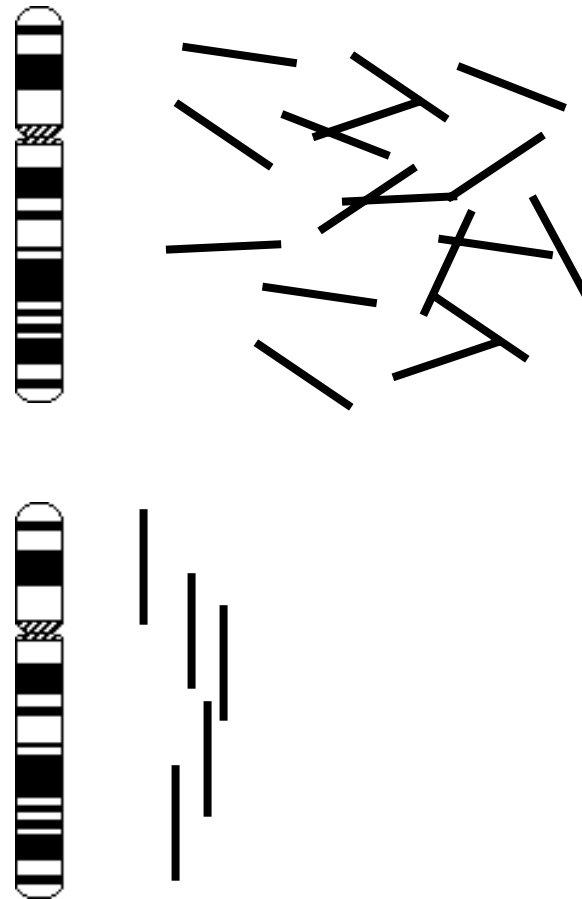
Wissensmanagement in der  
Bioinformatik



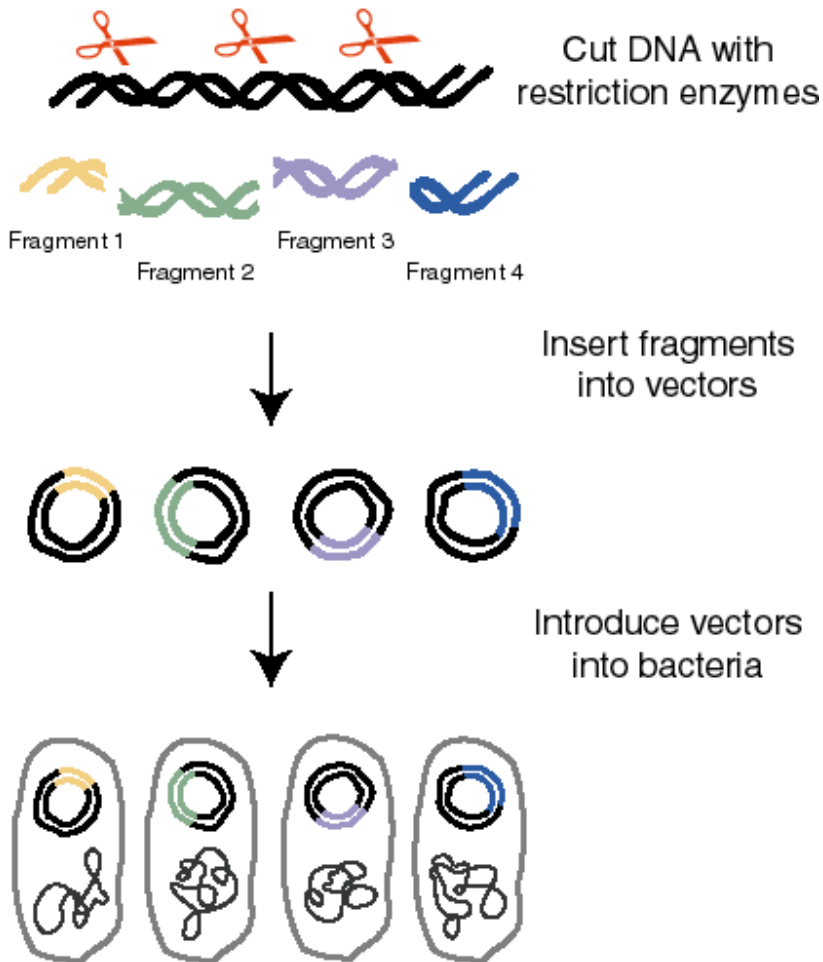
# Grundidee

---

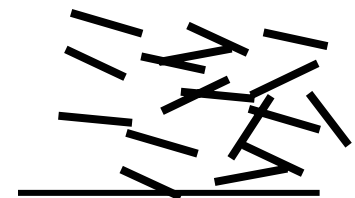
- Chromosom isolieren
- Zufällig in Stücke brechen (Clone)
- **Minimal Tiling Path** ermitteln
- Stücke sequenzieren
- Fertig



# Gesamtübersicht

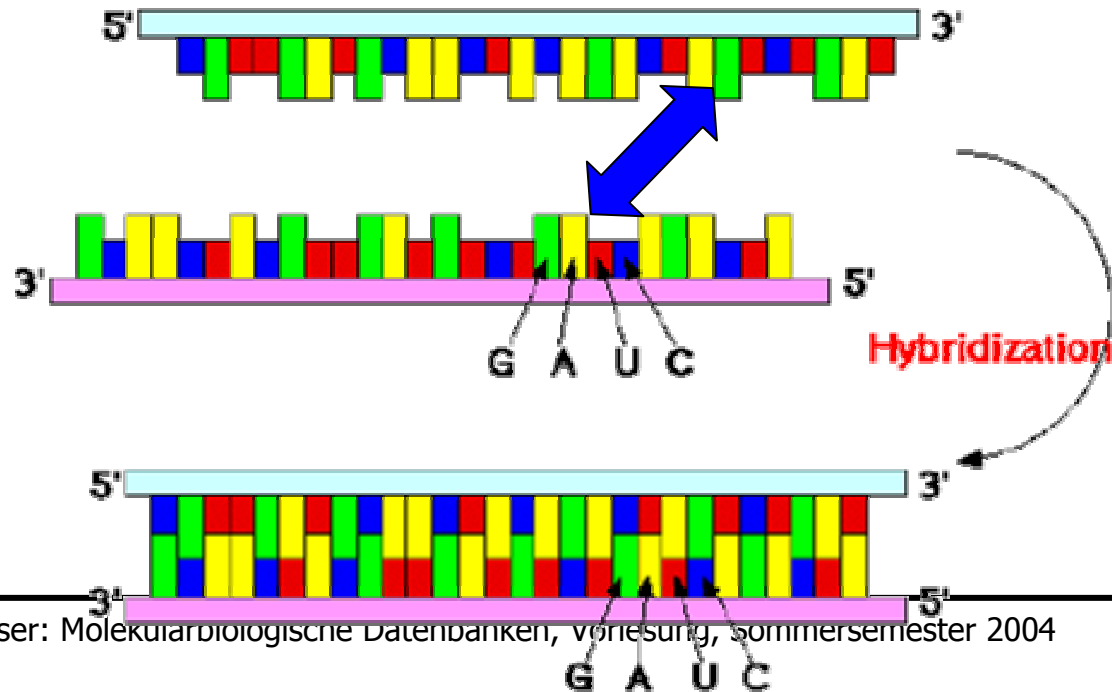


- Schneiden mit Restriktionsenzymen
  - Länge hängt ab von
    - Spezifität des Restriktionsenzym
    - Länge der Behandlung (partieller Verdau)
- Bruchstücke unterschiedlicher Länge
- Auftrennen nach Länge
  - Gelelektrophorese
- Clonierung in Bakterien
  - Vervielfältigung
- Ergebnis:



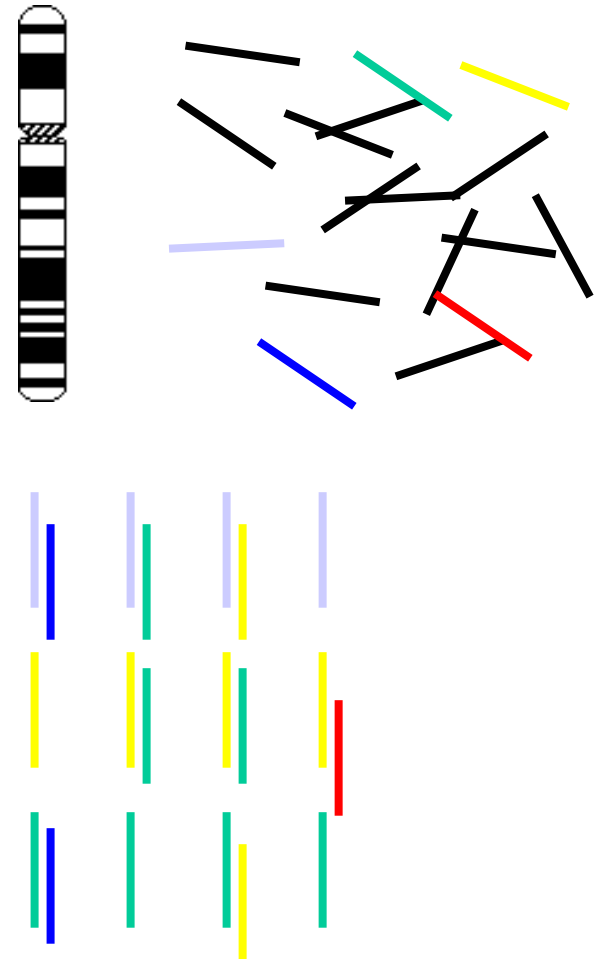
# Hybridisierung (Sketch)

- Ein Clone (Probe, Sonde) wird markiert (radioaktiv, fluoreszierend)
- Probe wird mit Bibliothek auf Filter zusammengebracht
- Clone in Probe hybridisiert mit den Clonen in der Bibliothek, die komplementäre DNA tragen
- Nicht hybridisierte Clone werden abgewaschen

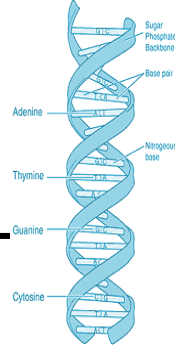


# Abstraktes Ergebnis

- Gegeben
  - Bibliothek
  - Menge von Proben
  - Hybridisierungsergebnisse
- Folgerungen
  - Clone-Überlappungen
  - Clone-Nicht-Überlappungen
- Mapping - Aufgabe
  - Konsistente Reihenfolge der Clone finden
  - Algorithmen später



# PCR Zyklen



Heat  
→

Cool  
↙

Heat  
→

Cool  
↙

Heat  
→

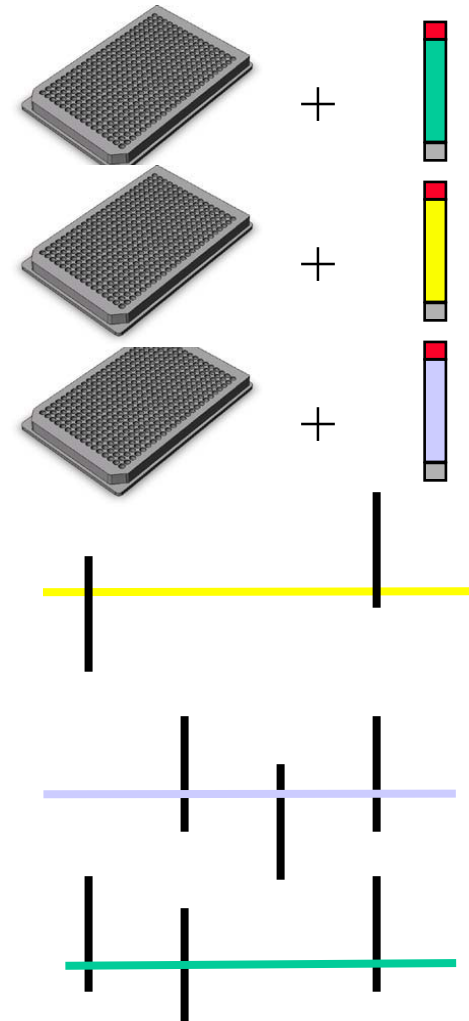
Cool  
↙

Heat  
→

etc.

# STS Content Mapping

- Gegeben
  - Menge von Clonen
  - Menge von STS
- Gesucht
  - Welcher der Clone enthält welche STS ?
- Verfahren
  - PCR mit jedem STS
- Ergebnis
  - „Anchoring“ der Clone
  - Algorithmen später



# Inhalt dieser Vorlesung

---

- Algorithmisches Problem
- Matrixdarstellung
- Algorithmen für STS Content Mapping
  - PQ Trees
  - Hamming TSP
- Datenmodelle
- Konkrete Kartierungsdatenbanken

# Abstrakte Problemdarstellung

---

- „Clone-Mapping“ Problem

- Geg.:

- Menge C von Clonen, Menge P von Proben, Menge S von Überlappung ( $P \subseteq C$ )
- $(c_i, c_j) \in S$ :  $c_i \in C$  und  $c_j \in P$  überlappen (Hybridisierung)
- $(c_i, c_j) \notin S$ :  $c_i$  und  $c_j$  überlappen nicht

- Gesucht: **Ordnung von  $c \in C$ , die mit S konsistent ist**

- „STS Content-Mapping“ Problem

- Geg.:

- Menge C von Clonen, Menge P von Markern, Menge S von Treffern
- $(p_i, c_j) \in S$ :  $p_i$  ist in  $c_j$  enthalten (PCR)
- $(p_i, c_j) \notin S$ :  $p_i$  ist nicht in  $c_j$  enthalten

- Gesucht: **Ordnung von  $p \in M$ , die mit S konsistent ist**

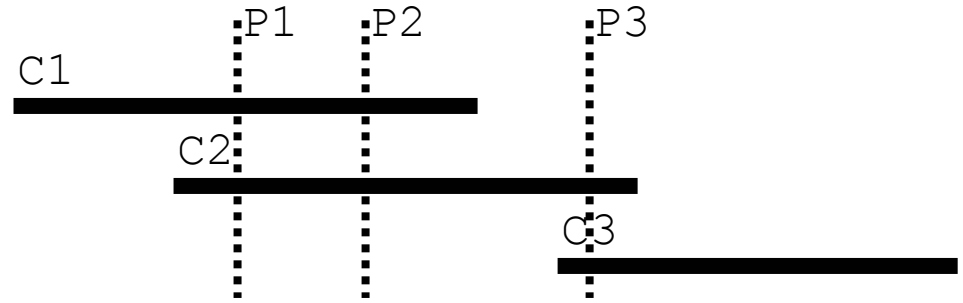
# Matrixdarstellung

---

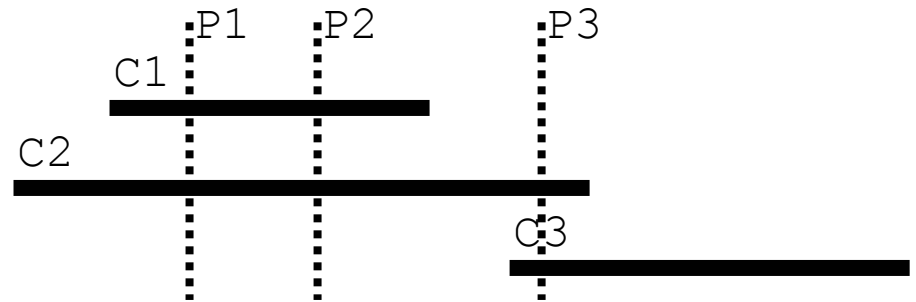
- Darstellung der experimentellen Ergebnisse in einer Matrix  $M$
- Zeilen: Clone aus  $C$
- Spalten: Marker aus  $P$
- Zellen
  - 1:  $p_i$  ist in  $c_j$
  - 0: sonst

# Beispiel: Marker-Mapping

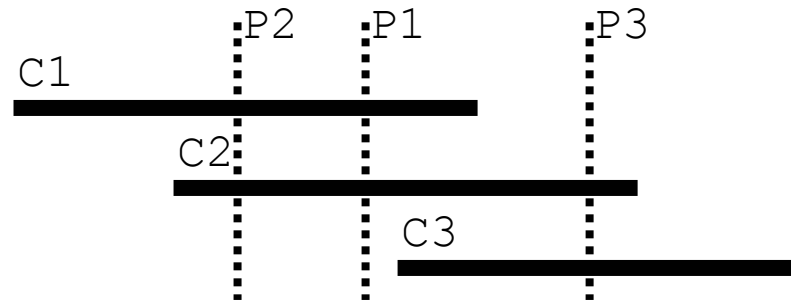
	P1	P2	P3
C1	X	X	
C2	X	X	X
C3			X



Oder ?

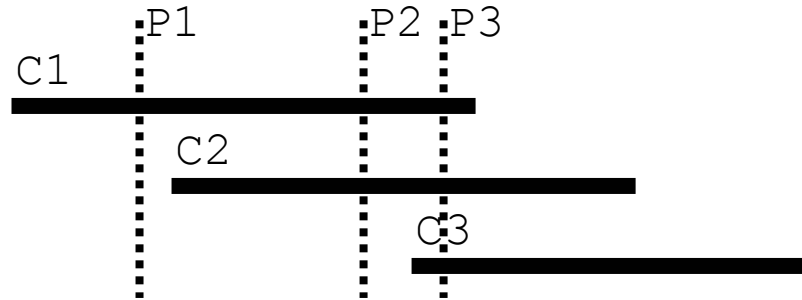


Oder ?

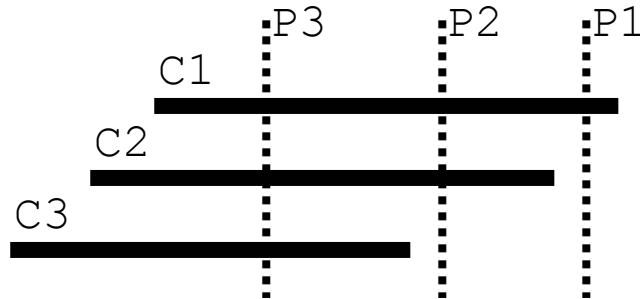


# Beispiel 2

	P1	P2	P3
C1	X	X	X
C2		X	X
C3			X



Oder ?

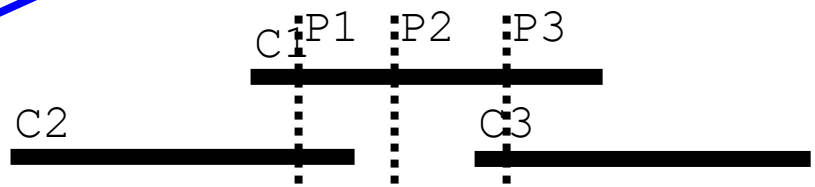


Richtung nicht bestimmbar

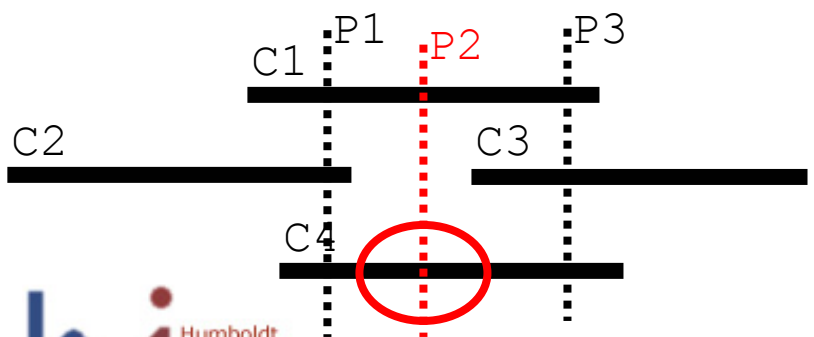
# Marker-Mapping mit Fehler

	P1	P2	P3
C1	X	X	X
C2	X		
C3			X
C4	X		X

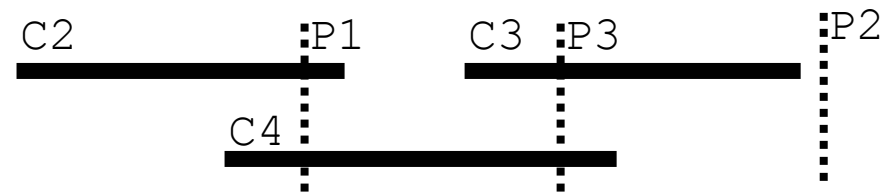
C1, C2, C3



C1, C2, C3, C4

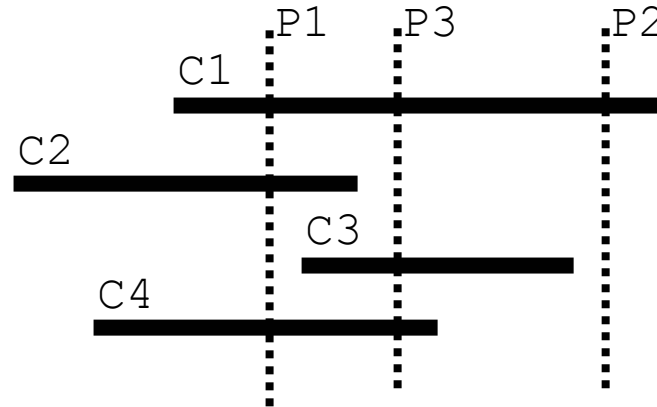


C2, C3, C4



# Tatsächlicher Fehler ?

	P1	P2	P3
C1	X	X	X
C2	X		
C3			X
C4	X		X



- Lösung vorhanden, wenn Clone unterschiedliche Länge haben können
- Aber: Clone einer Bibliotheken haben i.d.R. gleiche Länge

# Formalisierung

---

- Gegeben

- Menge C von Clones (Länge egal)

- Menge P von Probes

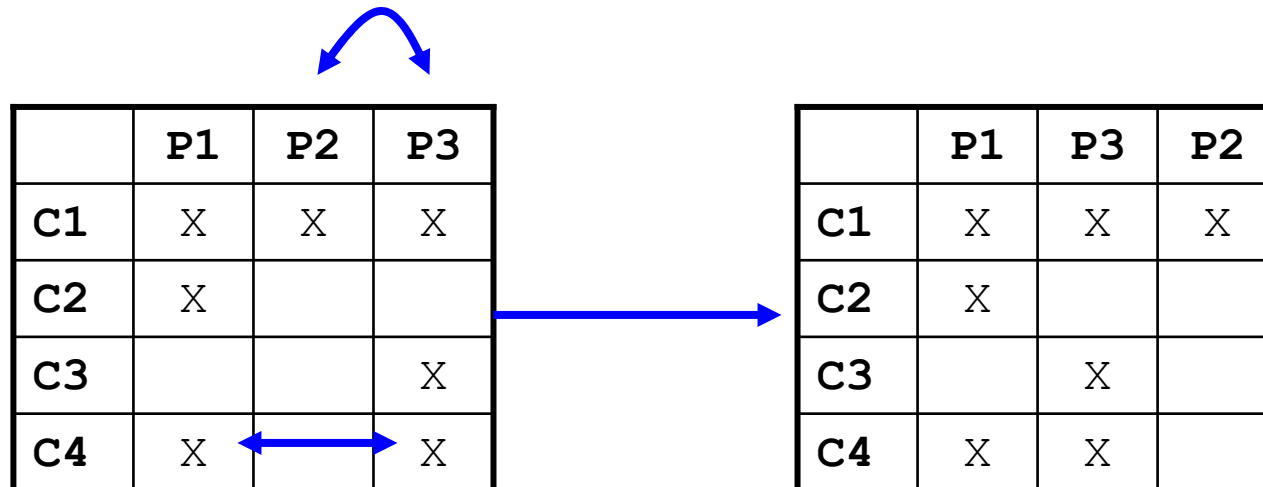
- Matrix

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


- Def: M hat „*consecutive ones property*“ (C1P),  
gdw es eine Permutation  $\pi$  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

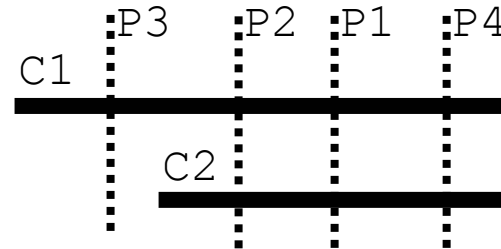
- Theorem: Eine Matrix M ist C1P gdw. es eine mit M konsistente Ordnung der Clone und Proben gibt
- Beweis: Literatur



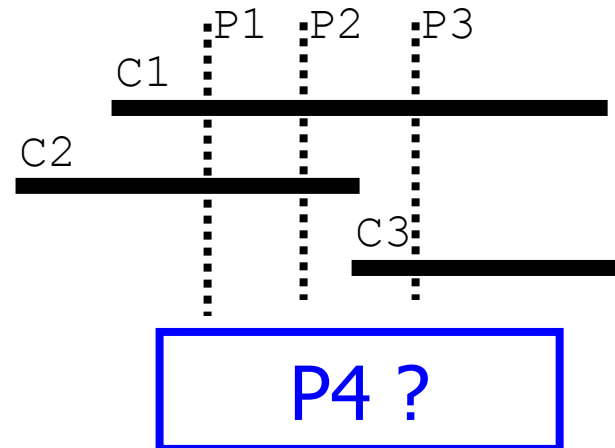
# Weitere Beispiele



	P1	P2	P3	P4
C1	X	X	X	
C2	X	X		X



	P1	P2	P3	P4
C1	X	X	X	
C2	X	X		X
C3			X	X



# Eigenschaften

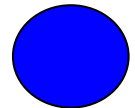
---

- Nicht jede Matrix ist C1P
  - Ergebnisse der Experimente haben immer Fehler
  - False negatives, false positives, chimerism, ...
- Wenn eine Matrix C1P ist, kann es viele Permutationen geben
  - Viele Lösungen möglich
  - Experiment ist unterdeterminiert
- Feststellen der C1P Eigenschaft ?
  - [PQ-Baum](#); linearer Algorithmus (Booth & Luecker)

# PQ - Baum

---

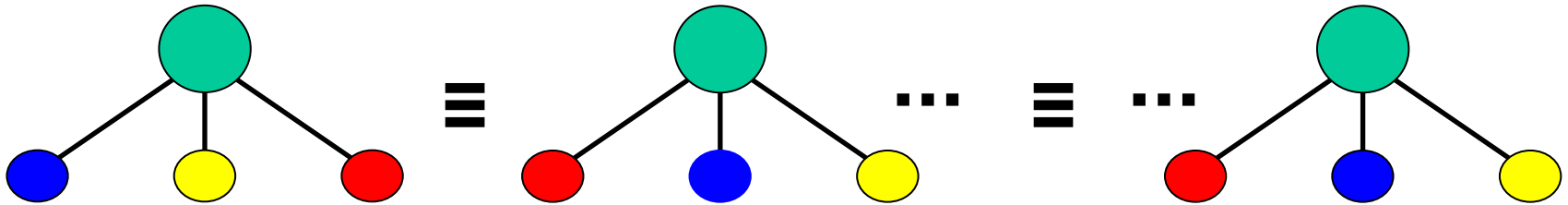
- Ziemlich kompliziertes Biest
- Grundidee: Kompakte Repräsentation von Permutationen
- PQ Baum
  - Blätter: Die Proben des Problems
  - P-Knoten
    - Enthält mind. 2 Kinder: PQ-Baum oder Blätter
    - Reihenfolge der Kinder ist nicht festgelegt
  - Q-Knoten
    - Enthält mind. 3 Kinder: PQ-Baum oder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



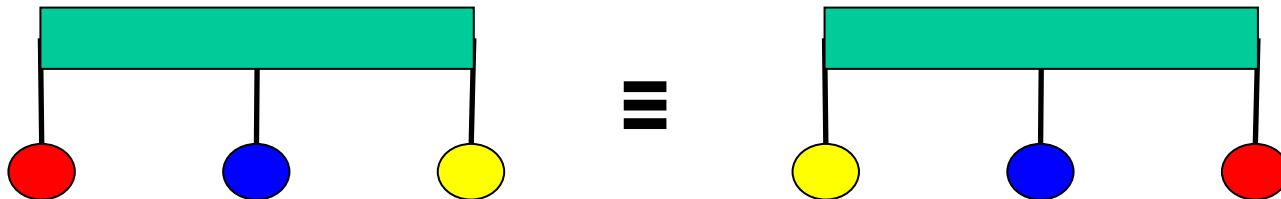
# PQ-Baum: Knotentypen

---

- P-Knoten



- Q Knoten



# Äquivalenz

---

- Definition

- Eine *Frontier* eines PQ Baums ist eine Ordnung der Blätter des Baumes von links nach rechts so, dass die Bedingungen der P bzw. Q Knoten erfüllt sind
- Zwei PQ Bäume  $T_1$  und  $T_2$  mit identischer Blättermenge heißen *äquivalent*, wenn es eine Ordnung der Blätter so gibt, dass diese Ordnung sowohl eine Frontier von  $T_1$  als auch von  $T_2$  ist

- Bemerkung

- Die Frontier eines PQ Baumes ist in der Regel nicht eindeutig

# Verfahren

---

- Verfahren
  - Starte mit Baum bestehend aus einem P-Knoten mit allen Proben als Blätter
  - Betrachte nacheinander Clone  $C_i$  mit jeweils positiven Proben  $P_1, \dots, P_n$ 
    - Negative Informationen werden implizit mit berücksichtigt
  - **Versuche, T zu äquivalentem  $T'$  so zu transformieren**, dass  $T'$  eine Frontier besitzt mit der Teilfolge  $P_1, \dots, P_n$ 
    - Dazu werden 9 Transformationsregeln aufgestellt
    - Regeln bestehen aus einem **Pattern**, das mit einem Ausschnitt des Baumes matchen muss, und einem **Replacement**, mit dem das Pattern ersetzt wird
- Bemerkung
  - Das geht nicht immer
  - Reihenfolge des Matching ist wichtig

# 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  $O(n+m)$*

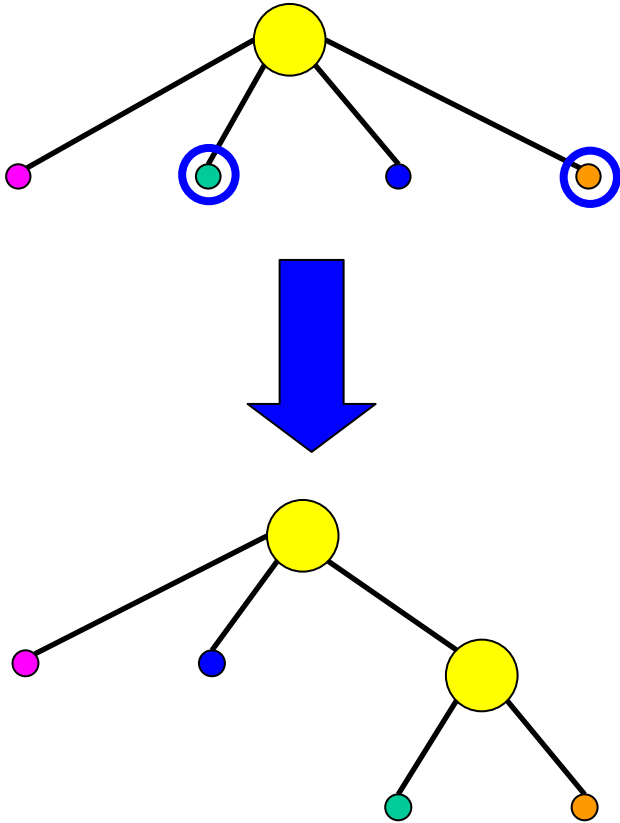
- Bemerkung

- 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





# Beispiel

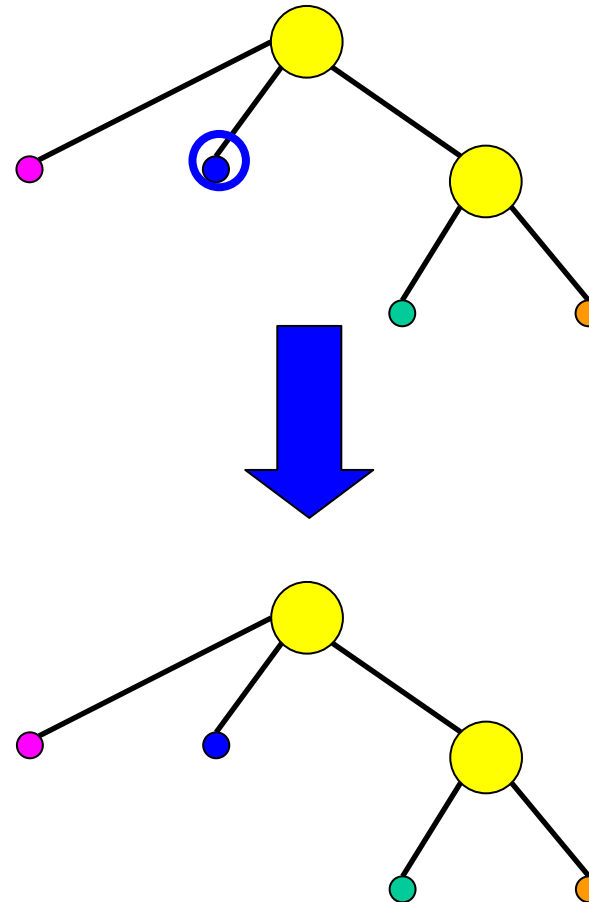
	 P1	 P2	 P3	 P4
C1		X		X
C2			X	
C3	X		X	
C4		X	X	X

	P1	P2	P3	P4
<b>C1</b>		X		X
C2			X	
C3	X		X	
C4		X	X	X



# Beispiel -2-

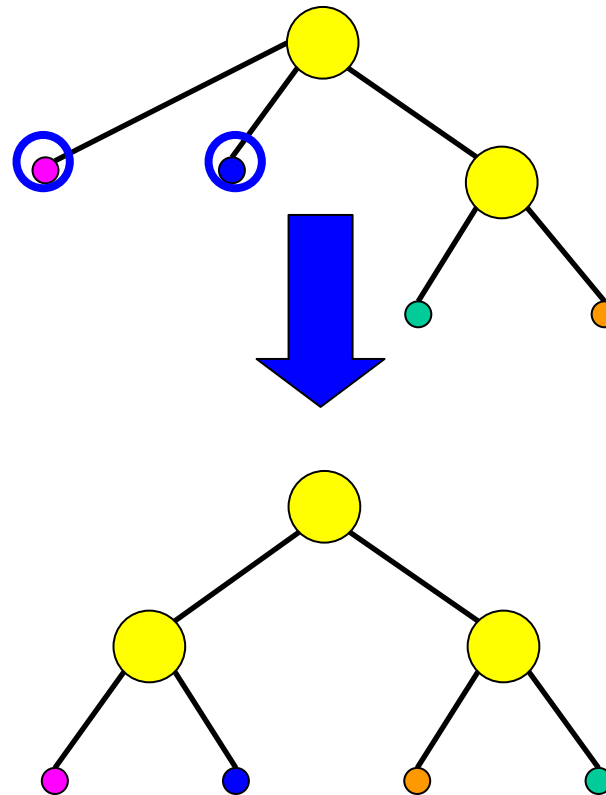
	 P1	 P2	 P3	 P4
C1		X		X
C2			X	
C3	X		X	
C4		X	X	X



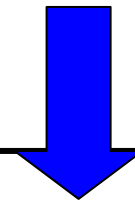
# Beispiel -3-





● ● ● ●

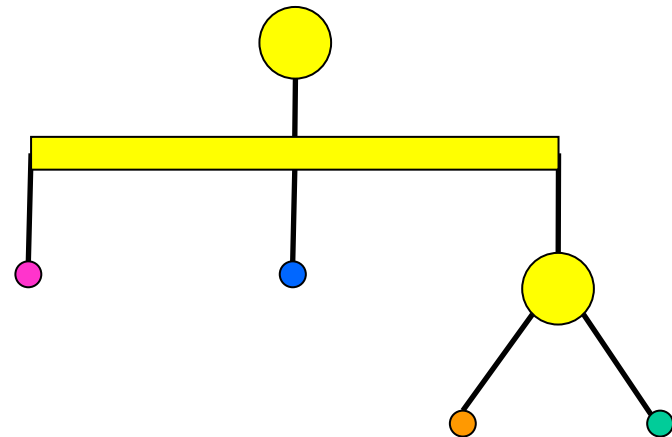
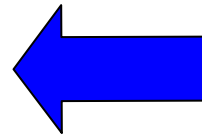
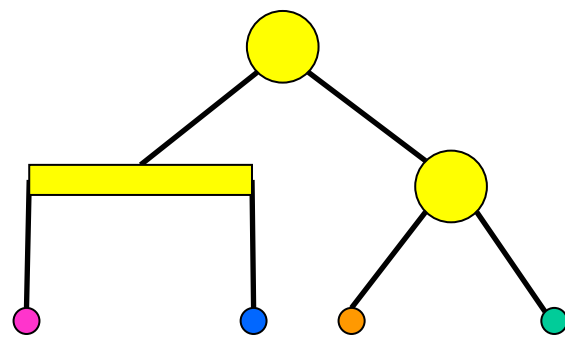
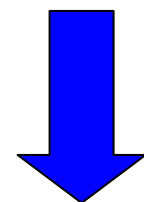
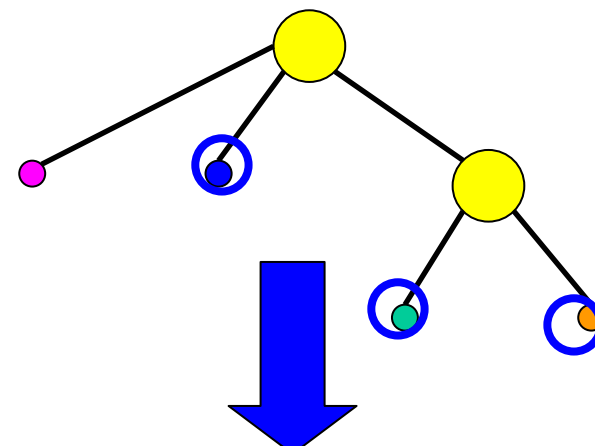
	P1	P2	P3	P4
C1		X		X
C2			X	
C3	X		X	
C4		X	X	X



# Beispiel -4-

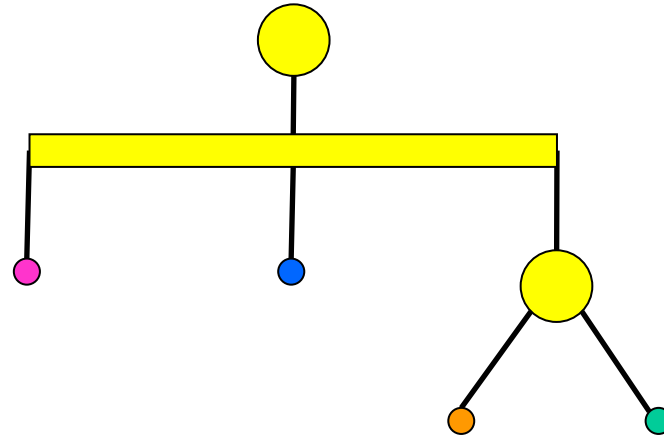


	 P1	 P2	 P3	 P4
C1		X		X
C2			X	
C3	X		X	
C4		X	X	X



# Ergebnis -5-

	<span style="color: magenta;">●</span> P1	<span style="color: cyan;">●</span> P2	<span style="color: blue;">●</span> P3	<span style="color: orange;">●</span> P4
C1		X		X
C2			X	
C3	X		X	
C4		X	X	X



	<span style="color: magenta;">●</span> P1	<span style="color: blue;">●</span> P3	<span style="color: orange;">●</span> P4	<span style="color: cyan;">●</span> P2
C1			X	X
C2		X		
C3	X	X		
C4		X	X	X

	<span style="color: magenta;">●</span> P1	<span style="color: blue;">●</span> P3	<span style="color: cyan;">●</span> P2	<span style="color: orange;">●</span> P4
C1			X	X
C2		X		
C3	X	X		
C4		X	X	X

	<span style="color: cyan;">●</span> P2	<span style="color: orange;">●</span> P4	<span style="color: blue;">●</span> P3	<span style="color: magenta;">●</span> P1
C1	X	X		
C2			X	
C3			X	X
C4	X	X	X	

Siehe auch: <http://www.comp.nus.edu.sg/~liujian1/hyp/>

# Zusammenfassung

---

- Finden der entsprechenden Transformationen nicht trivial, aber linear
- Keine Erweiterung bekannt für „noisy“ data
- Keine Verwendung als Mapping Algorithmus für real life data

# Back to real life

---

- Echte Daten sind nicht so hübsch
  - Falsche positive Signale
  - Falsche negative Signale
  - Chimeric Clones
- Matrix nicht C1P
- Problem: Finde die wahrscheinlichste Lösung
  - Möglichst wenig „1“ einfügen (false negatives)
  - Möglichst wenig „1“ löschen (false positives)
  - Möglichst wenig Clone entfernen/auftrennen (chimerism)
  - Möglichst wenig Proben entfernen (Non-Unique)

# Verfahren

---

- Diverse Verfahren sind veröffentlicht
  - Maximum Likelihood Berechnung
  - Bayes'sche Wahrscheinlichkeiten
  - ...
- Alle sehr teuer, deshalb Näherungsverfahren
  - Simulated Annealing
  - Greedy
  - ...
- Im Folgenden
  - Mapping Problem nur mit False Positives
  - Problem wird sofort NP (ohne False Positives: Linear)

# Falsch Positive - Problemformulierung

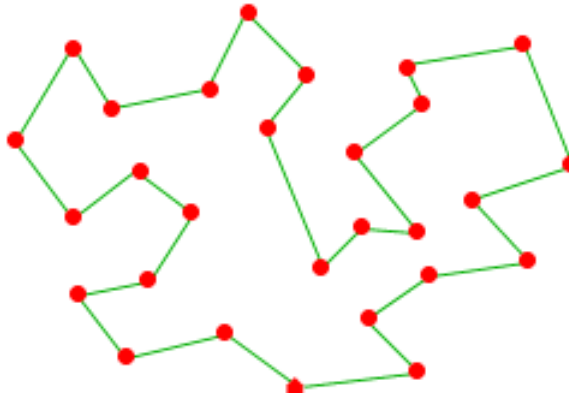
---

- 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

# Voraussetzung

---

- Hamming-Abstand zweier Spalten
  - Anzahl ungleicher Werte
- Traveling Salesman Problem
  - Geg.: Graph mit gewichteten Kanten
  - Finde Weg so, dass jeder Knoten genau einmal besucht wird und die Summe der Kantengewichte minimal ist



# Hamming-TSP [AKNW95]

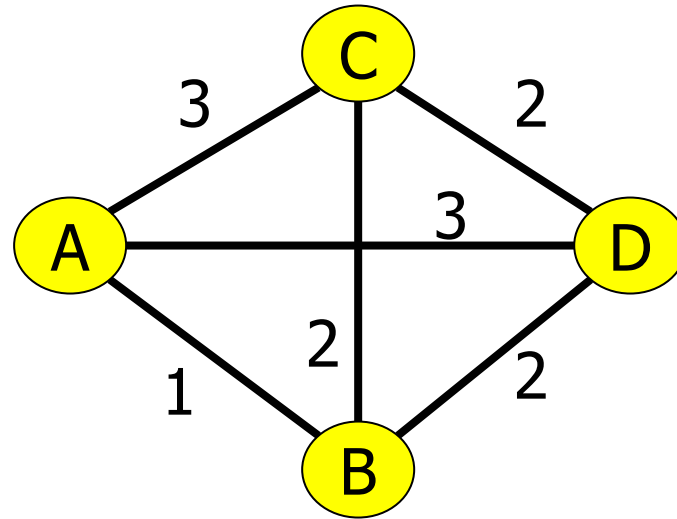
---

- Theorem
  - Löse das TSP Problem für folgenden Graphen
    - Jede Probe ein Knoten
    - $w(P_1, P_2) = \text{hamming}(P_1, P_2)$
  - Der Pfad der Lösung induziert  $M'$
- Problem
  - TSP ist NP vollständig
  - ... und der Graph ist vollständig

# Beispiel 1

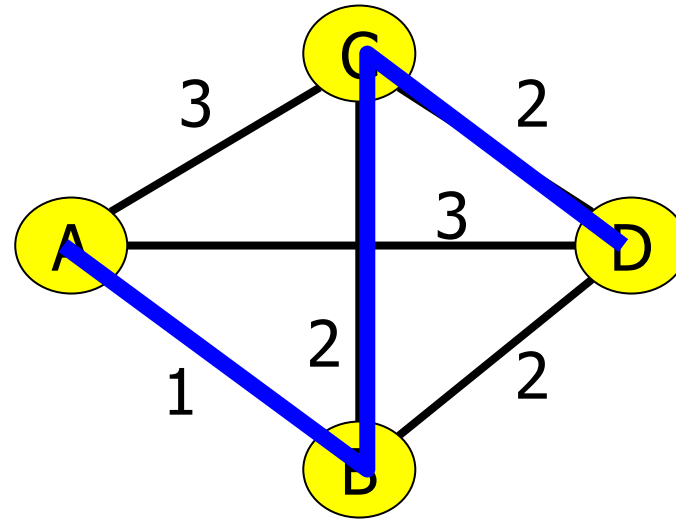
War nicht lösbar

	A	B	C	D
C1	X	X	X	
C2	X	X		X
C3			X	X
C4		X	X	X



# Beispiel 2 - Optimal

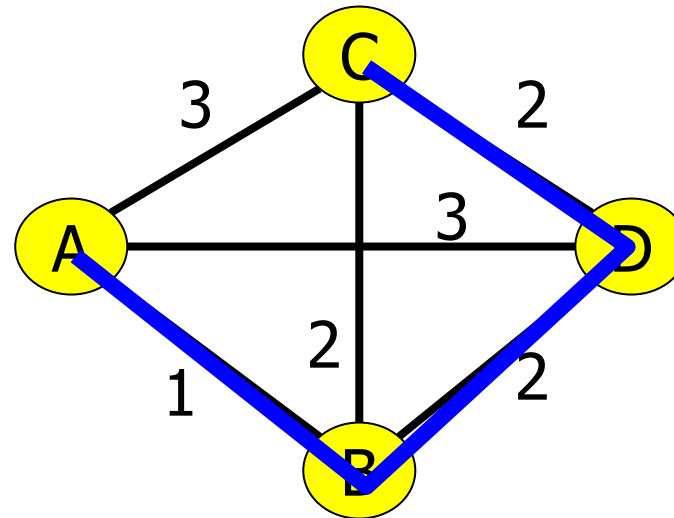
	A	B	C	D
C1	X	X	X	
C2	X	X		X
C3			X	X
C4		X	X	X



	A	B	C	D
C1	X	X	X	
C2	X	X		X
C3			X	X
C4		X	X	X

# Beispiel 3 - Optimal

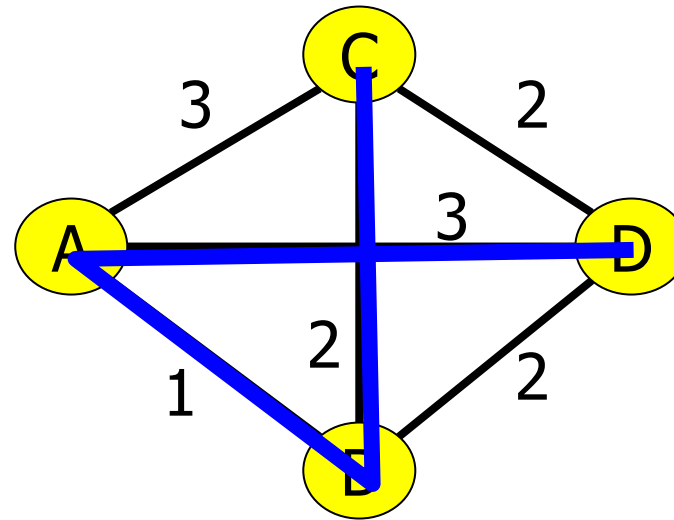
	A	B	C	D
C1	X	X	X	
C2	X	X		X
C3			X	X
C4		X	X	X



	A	B	D	C
C1	X	X		X
C2	X	X	X	
C3			X	X
C4		X	X	X

# Beispiel 4 – Nicht optimal

	A	B	C	D
C1	X	X	X	
C2	X	X		X
C3			X	X
C4		X	X	X



	D	A	B	C
C1		X	X	X
C2	X	X	X	
C3	X			X
C4	X		X	X

# Teil II. Datenmodelle

---

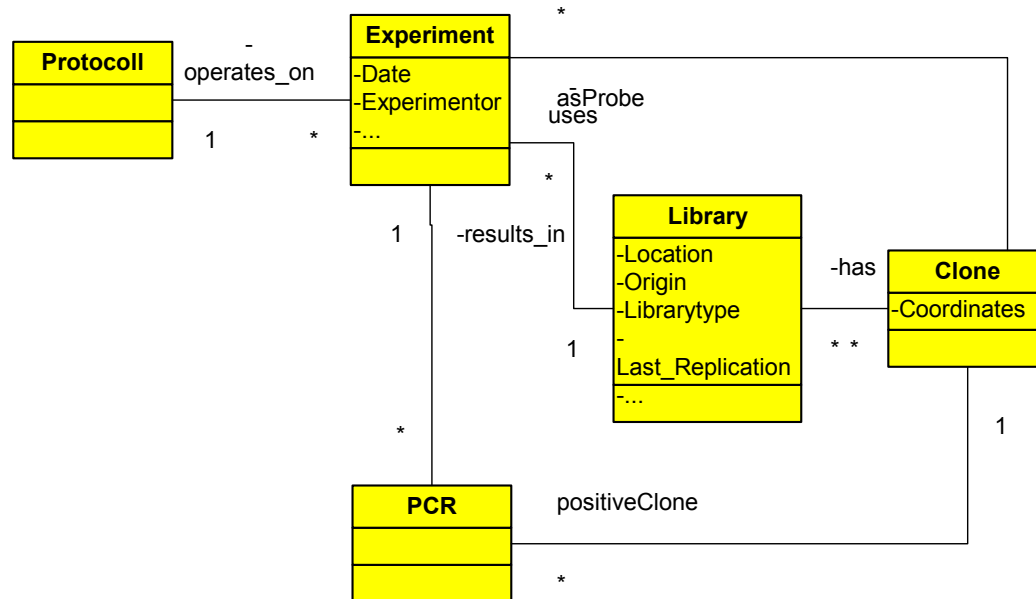
- Modellierung von experimentellen Daten
- Modellierung von Karten (OMG Standard)

# Experimentelle Daten

---

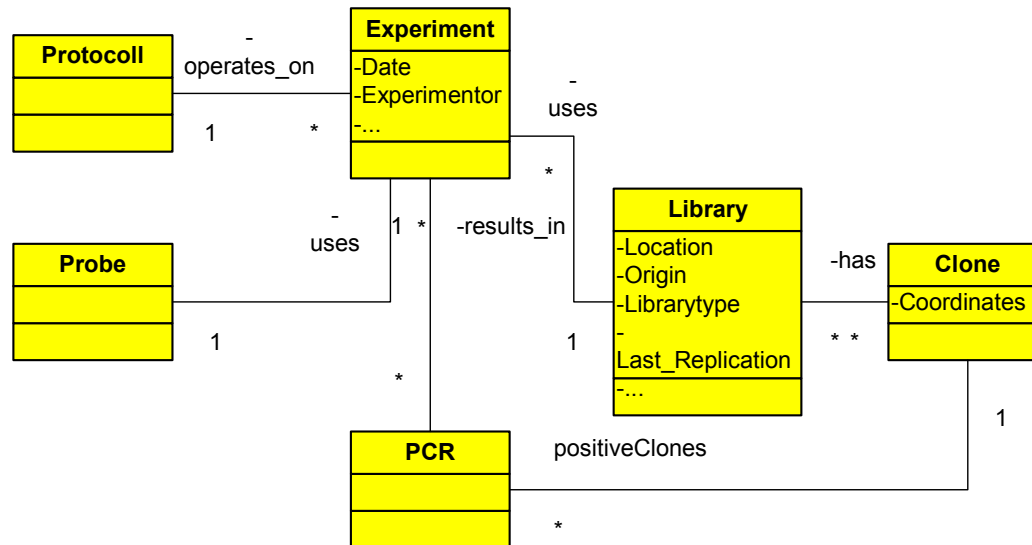
- Rohdaten der Kartierung sind Experimente
- Metadaten
  - Durchführender, Datum, Roboter/Geräte, angewandtes Protokoll (Temperaturen, Farbstoffe, Enzyme, ...)
- Experimentdaten (**physikalische Karten**)
  - Clone mit Koordinaten (in Bibliothek)
  - Probe
- Ergebnisdaten
  - Clone-Mapping: Überlappungen
  - Marker-Mapping: Part\_of

# Beispielhaftes Modell - Hybridisierung



- Pro Experiment ist ein Clone die Probe

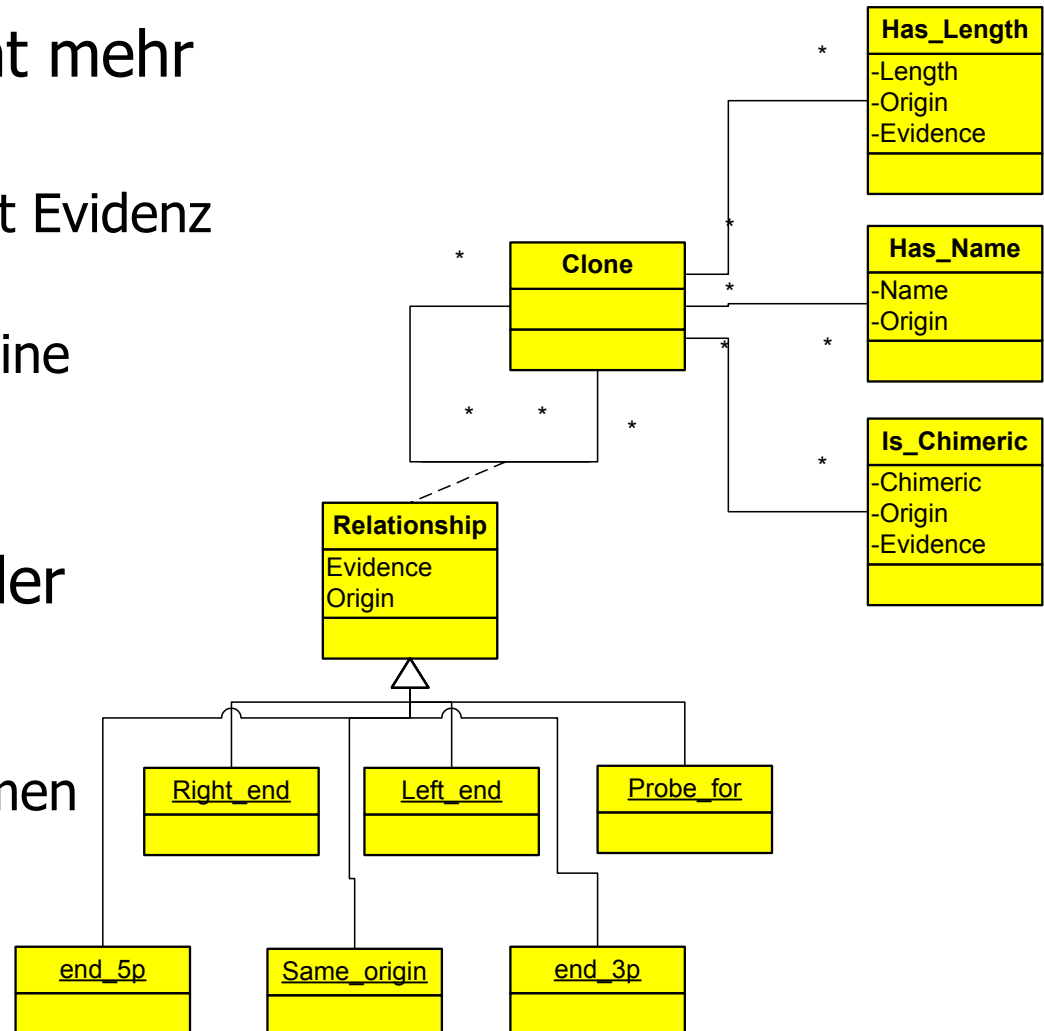
# Beispielhaftes Modell - PCR



- Bibliothek – Clone: m:n

# Integrierte Karten

- Objektattribute nicht mehr eindeutig
  - Jede Eigenschaft mit Evidenz und Ursprung
  - Auch ein Name ist eine Eigenschaft
- Objekte stehen in Beziehung zueinander
  - Als zusätzliche Informationen für Kartierungsalgorithmen nutzbar



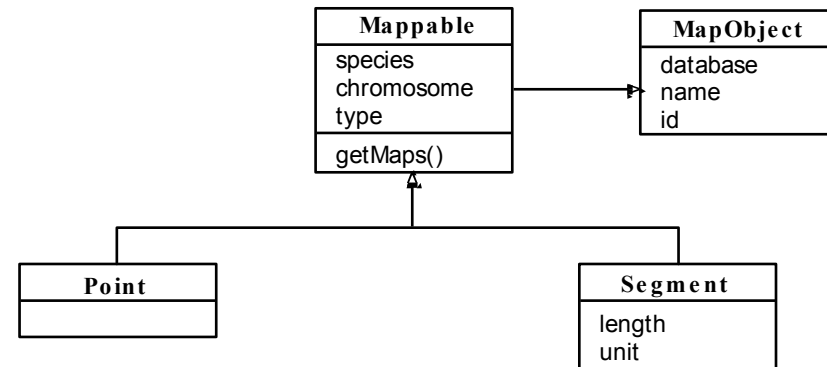
# Modellierung von Genomkarten

---

- Abstraktion von technischen Details der Experimente
- Elemente von Genomkarten
  - Kartentyp
    - Genetisch, physikalisch, cytogenetisch, ...
  - Kartierte Objekte
    - Clone, Marker, **Unterkarten, Contigs**, ...
  - Beziehungen zwischen kartierten Objekten
    - Enthalten, Überlappt, liegt\_vor, liegt\_nach, linkes\_ende, ...
  - Koordinaten
    - Ordnung, Halbordnung, ungeordnet
    - Numerische Koordinaten oder nicht
- **Hierarchische Beziehungen**: Karten enthalten Karten



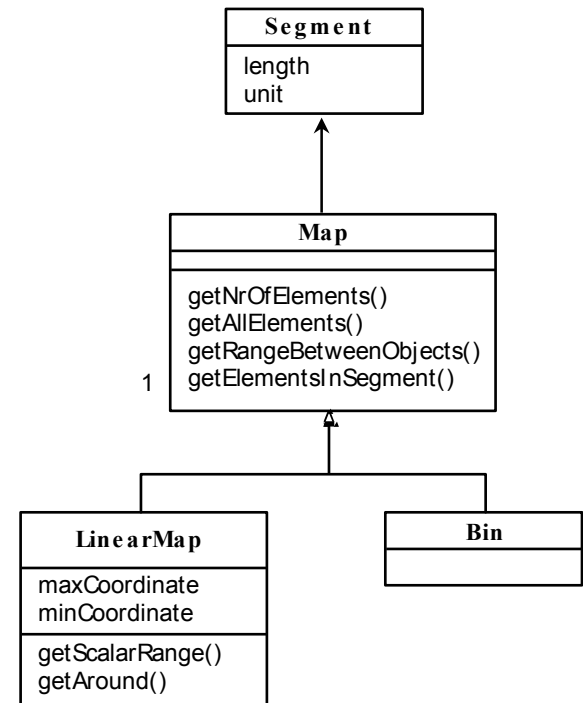
# „Mappable“



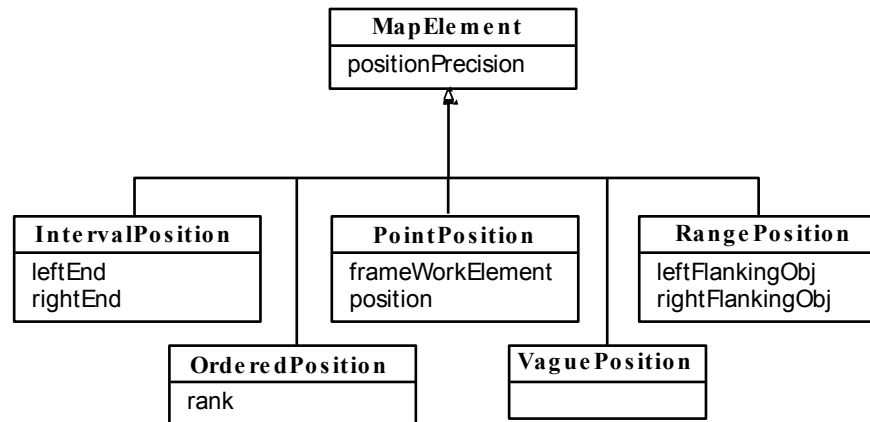
- **Mappable**: Alle Objekte, die man auf einer Karte platzieren kann
- **Segments**: Haben Ausdehnung: Clone, Chrombands, Karten, ...
- **Points**: Keine Ausdehnung: Marker, EST, STS, ...

# „Map“

- **Maps:** Sind selber Segmente
- Karten können Karten enthalten
- Zwei Arten von Karten
  - **Linear maps:** Mit Koordinatensystem:
    - Physikalische Karten
    - Genetische Karten
  - **Bin maps:** Geordnete/ halbgeordnete/ ungeordnete Kollektionen
    - Radiation-hybrid Karten
    - Chromsomenbänder



# „MapElement“



- **MapElement**: Zuweisung eines **Mappable** zu einer **Map** mit einer **Position** in einem **Coordinatesystem**
- **Position**: vor, nach, zwischen, in, ...
- n:m Relationship zwischen **Map** und **Mappable**

# Bewertung

---

- Sehr mächtiges und generisches Modell
- Orientierung an „veröffentlichten“ Karten
  - Keine Rohdaten, sondern Koordinaten
  - Exakte Koordinaten nicht überbewerten
- Viele Constraints nicht repräsentiert
  - Points dürfen keine IntervalPosition haben
  - BinMaps dürfen keine Koordinaten-Positions haben
  - Humane Karten können keine Mausclone enthalten
  - ...
- Keine Behandlung von Synonymen, Homonymen, etc.
  - OMG Philosophie

# Teil III. Datenbanken

---

- The Genome Database (GDB)
- eGenome
- LocusLink



- Jahrelang Standarddatenbank für HGP Mapping Daten
- Anzahl Objekte
  - 4.000.000 Clone (BAC und PAC)
  - 14.000 Gene mit Position
  - 150.000 Marker
- Verfahren der Integration
  - Submission based (CTL – Format)
  - Idee der „Community Curation“
  - GDB Consensus Maps: heuristisches Verfahren, ohne Rohdaten (<http://www.gdb.org/gdb/uCoordTech.html>)
  - Chromosome Editors
- Technische Implementierung
  - OPM, Sybase
  - OPM-Modell mit ca. 75 Klassen, ca. 140 Tabellen

# GDB

Display Results As: Table

**Names:**

Name

**Library Addresses:**

Library	Plate location	Plate Row position	Plate Column position	Location Type
?				Original Replated

**Cytogenetic Localization:**

Chromosome	Left Marker	Right Marker
?	?	?

**All Localizations:**

Chromosome	Left Marker	Right Marker
?		

**Nucleic Acid Sequence Links:**

?

**Related Segments:**

Marker

Document: Done (0.852 secs)

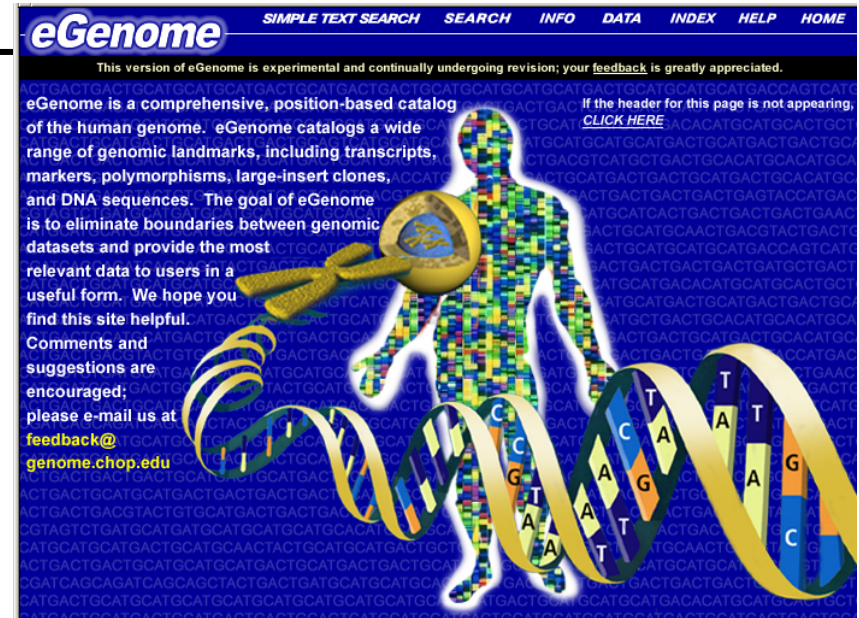
# GDB Bewertung

---

- Sehr technisch orientiert
- Modell ähnlich zu OMG Standard
  - Maps on Maps
  - Coordinate Systems
  - Object – Mapping – Map
- Komplizierte Search-Forms kaum benutzt
- Community Curation kaum benutzt
- Langsam
  - Durch vertikales OR-Mapping – viele Joins
  - Langsame Webinterfaces werden nicht akzeptiert

# eGenome [http://egenome.chop.edu]

- Projekt, org. Chromosom 1
- Z.Z. ca.
  - 72.000 Marker
  - 116.000 Clones
  - Ein Marker alle 45 KB
- Integrationsverfahren
  - Konservatives Verfahren
  - CEPH Genetic maps – Radiation Hybrid maps - UniGene Cluster -
  - Benutzung der Konfidenzvalues
  - I.d.R: Berechnung von Bins (Bundels), nicht Koordinaten
- Technische Realisierung
  - 4Dimension (3GL Datenbank + Entwicklungsumgebung)
  - Export als Flatfiles verfügbar



# eGenome

Welcome to eGenome

**eGenome** SIMPLE TEXT SEARCH  
Position Description

**Sequence position**  
Base pairs 103,530 to 103,688 from Xpter ([UCSC](#))

**RH map position**  
[WI-4757](#) to [DXS999](#)  
43.6 to 63 cR from Xpter  
RH positions 4 to 6

**Cytogenetic position**  
Xp22.3 to Xp22.1

**RH score**  
[Genebridge4](#)  
0200202100 2200001010 0000000010 001000000  
00R0000201 0011000002 0101000002 000100000

**RHdb entry**  
[RH36003](#)

**Primer sequences**  
TTATTACCTTAAGTAGCAGAA  
TACCTGGCAGCTATATTC

**Search for LOC51611 in:**  
[UCSC](#) [NCBI MapViewer](#) [GDB](#)  
[Ensembl](#) [GenBank](#) [GeneCards](#)

Document: Done (0.792 secs)

**GDB Mapview 3.0**  
File Edit View Maps Align Detail Help

Selected:

Zoom:

eGenome Chr X (Chr. X)

Markers...  Intervaled RH markers

APXL -33.0 cR  
AFMa0522c1, stS646149, stS647635, TMSB4X  
stS648747, stS651739, DXS7005E, WI-5813  
WI-5700, PRPS2, KIAA1280, AFMb321xg5  
ARHGAP6, AMELX, DXS1224, CLCN4  
TRIM28, WI-3796, SHGC-34710, WI-17390  
OA1, TIGR-A004U24, DXS8051, WI-2992  
emb1-T67560, MSL3L1, TLR7, A003X02

WI-4757 -45.0

RH67763 -53.0  
- RH68864, stS613075

DXS999 -61.0  
\_PIR, stS663643, PRO2325, ACE2, PIGA, AP1S2, DXS1053, AFM205yd12  
stS635971, WI-13794, SHGC-6624, REP52, WIAF-2131, RBBP7, WI-1819

SCML2 -69.0  
▼DXS1226, DXS989

DXS1223, WI-5439  
DXS1060, DXS6807  
WI-3094, AFM212zb12  
stS630779, stS631919  
stS629320, WI-13308  
STS, HCCS  
stS635934, DXF68S1E  
TIGR-A008T30, DXS7031E  
HADHB, MID1  
RH38768, DXS1283E  
stS615225, RH38708

GLRA2  
RH68032  
stS632112

RH38775, SH3KBP1  
RH38701, PPEF1  
RH38703, stS650968  
TADA3L, PHKA2  
RAI2, stS652924

stS630363, G1  
RH38751, WI-6  
BMX, RH28491  
SCML1, stS62  
stS62851, AR:  
stS631973, DI  
GY62, DXS70  
PDHA1, FIGF,  
stS630601, S  
DRAP1

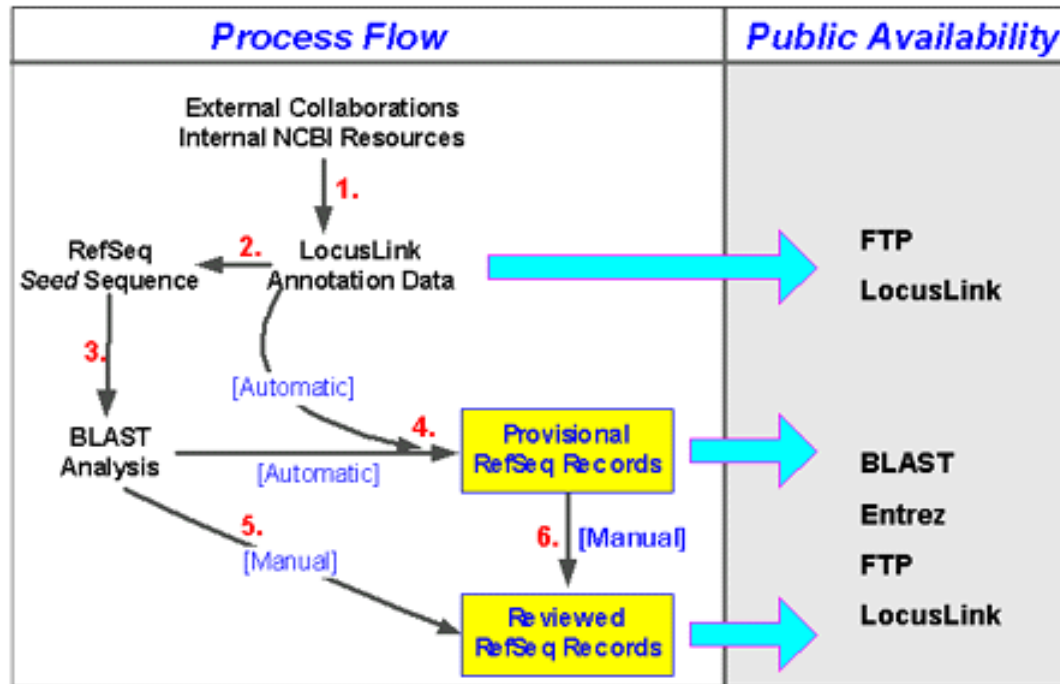
Welcome to Mapview  
Jump Scroll:     Horiz. Increment:  Vert.:

Java Applet Window

## Einfache Textsuche & „Region Search“



# LocusLink Integrationsworkflow



Quelle: [www.ncbi.nlm.nih.gov/LocusLink/build.html](http://www.ncbi.nlm.nih.gov/LocusLink/build.html)

- Mischung aus manueller und automatischer Bearbeitung
- Objektstatus: Provisional - Reviewed
- Kein Releasekonzept, keine Versionierung

# Weitere

---

- Human Genome Browser
- Chromosomenspezifische Datenbanken
  - Jedem Chromosom seine Datenbank
- Die grossen Mappingprojekte
  - Whitehead, Washington, Ceph, ...
- Die grossen Sequenzierprojekte
  - Sanger, Celera, ...
- ...
- Übersicht z.B. in [BO01], Kap. 6

# Zusammenfassung

---

- Diverse Algorithmen zur Berechnung von Karten bekannt
  - Graphbasiert, PQ-Trees, statistisch
- Realistische Algorithmen berechnen nur „bestmögliche“ Anordnungen
  - Näherungstechniken und Heuristiken
  - Lösungen i.d.R. nicht eindeutig
- Kleine Fehler – große Wirkung (Chimeric Clones)
- Real life
  - Kombination von Daten aus unterschiedlichen Quellen / Verfahren
  - Crossvalidierung
  - Manuelle Nachbearbeitung
- Datenmodelle i.d.R. mit rekursiver Komponenten
  - Objekte sind „geschachtelt“

# Literatur

---

- [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 6
- [Sen02] Sensen, C., Ed. (2002). "Essentials of Genomics and Bioinformatics", Wiley-VCH, Weinheim., Kapitel 2, 6
- [Pri96] Primrose, S. B. (1996). "Genomanalyse". Heidelberg, Spectrum; Akademischer Verlag.
- [GI95] Greenberg, D. S. and Istrail, S. (1995). "Physical Mapping by STS Hybridisation: Algorithmic Strategies and the Challenge of Software Evaluation." *Journal of Computational Biology* **2**(2): 219-274.
- [AKNW95] Alizadeh, F., Karp, R. M., Newberg, L. A. and Weissner, D. K. (1995). "Physical Mapping of Chromosomes: A Combinatorial Problem in Molecular Biology." *Algorithmica* **13**(1/2): 52-76.
- [BLL+99] Barillot, E., Leser, U., et al. (1999) "A Proposal for a Standard CORBA Interface for Genome Maps", *Bioinformatics* 15(2): 157-169.