

V4 – Analyse von Genomsequenzen

- **Gene identifizieren**

 - Intrinsische und Extrinsische Verfahren:
Homologie bzw. Hidden Markov Modelle

- **Transkriptionsfaktorbindestellen** identifizieren

 - Position Specific Scoring Matrices (PSSM)

- Ganz kurz: finde **Repeat-Sequenzen**

 - Suche nach bekannten Repeat-Motiven

- **Mapping** von **NGS-Daten** auf **Referenzgenom**

- **Alignment zweier Genom-Sequenzen**

 - Suffix Bäume

Leitfragen für V4

Frage1: Wie können wir funktionell wichtige Bereiche in Genom-sequenzen finden?

Ansatz: leite aus bekannten Genen bzw. Transkriptionsfaktorbindestellen allgemeine Prinzipien ab und verwende diese dann zur Vorhersage.

Frage2: Wie können wir funktionell entsprechende Bereiche in anderen Genomsequenzen finden?

Ansatz: finde homologe, nur einmal vorkommende Bereiche in beiden Genomen als Ankerpunkte für das Genom-Alignment.

Identifikation von Genen

Die **einfachste** Methode, DNA Sequenzen zu finden, die für Proteine kodieren, ist nach **offenen Leserahmen (open reading frames oder ORFs)** zu suchen.

In jeder Sequenz gibt es 6 mögliche offene Leserahmen:

3 ORFs starten an den Positionen 1, 2, und 3 und gehen in die 5' 3' Richtung,

3 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 in mRNA transkribiert und die mRNA wird ohne wesentliche Änderungen direkt in einen Aminosäurestrang übersetzt.

Daher ist der längste ORF von dem ersten verfügbaren Met codon (**AUG**) auf der mRNA, das als **Codon** für den **Transkriptionsstart** fungiert, bis zu dem **nächsten Stopcodon** in demselben offenen Leserahmen, gewöhnlich eine gute Vorhersage für die Protein-kodierende Region.

Extrinsische und intrinsische Methoden

Viele Verfahren kombinieren nun

(a) Homologie-Methoden = „**extrinsische Methoden**“ mit

(b) Genvorhersage-Methoden = „**intrinsische Methoden**“

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 Vorhersage-Methoden einsetzen.

Table I: Availability of gene prediction software

| Program | URL | Web interface | Download | Source |
|------------|---|------------------|------------------|------------------|
| Geneid | http://www.limim.es/software/geneid/index.html | yes | yes | yes ^a |
| GlimmerM | http://www.tigr.org/softlab/glimmerm/ | yes | yes ^b | yes ^b |
| GenScan | http://genes.mit.edu/GENSCAN.html | yes | yes ^b | no |
| GenomeScan | http://genes.mit.edu/genomescan/ | yes | no | no |
| MHHgene | http://www.cbs.dtu.dk/services/HMMgene/ | yes | no | no |
| FGENES | http://genomic.sanger.ac.uk/gf/gf.shtml | yes | no | no |
| Genie | http://www.cse.ucsc.edu/~dkulp/cgi-bin/genie | yes ^c | no | no |

^aFree to all under GNU licence.

^bFree only to academic users. Commercial users must purchase a licence.

^cWeb interface to Genie uses an older version of this program, not that one referred to in publications of fruit fly and human genomes or the GASP.

Beispiel: Vergleich von Glimmer und GeneMarkS

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.

^aRefers to the case where both the 5' end and the 3' end predictions match the annotation.

^bRefers 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)

Hidden Markov Modell (HMM)

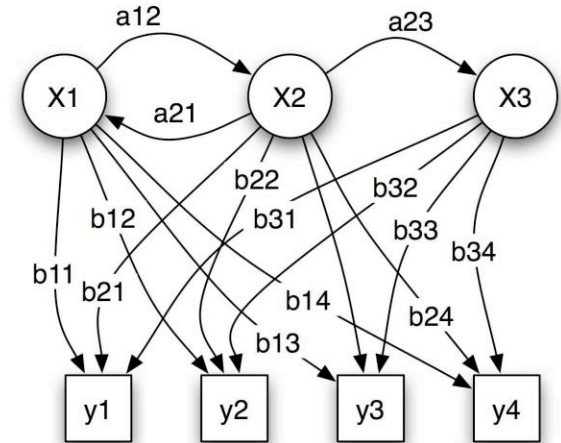
Ein Hidden Markov Modell ist ein Graph, der verschiedene Zustände verbindet.

Im Modell rechts gibt es 3 „verborgene“ Zustände: X1, X2, X3.

Zwischen den Zuständen X1 und X2 und zurück und von X2 nach X3 sind hier Übergänge erlaubt.

Die Übergangswahrscheinlichkeiten hierfür sind a_{12} , a_{21} und a_{23} .

y_1 bis y_4 sind die möglichen Output-Zustände, die aus den verborgenen Zuständen mit den Wahrscheinlichkeiten b_{11} bis b_{34} erzeugt werden.



Die Topologie des Graphen gibt an, zwischen welchen Zuständen Übergänge erlaubt sind. Diese gibt man bei der Spezifikation des HMM vor. Jeder Übergang hängt nur von den beiden Zuständen i und j ab, nicht von früheren Zuständen.

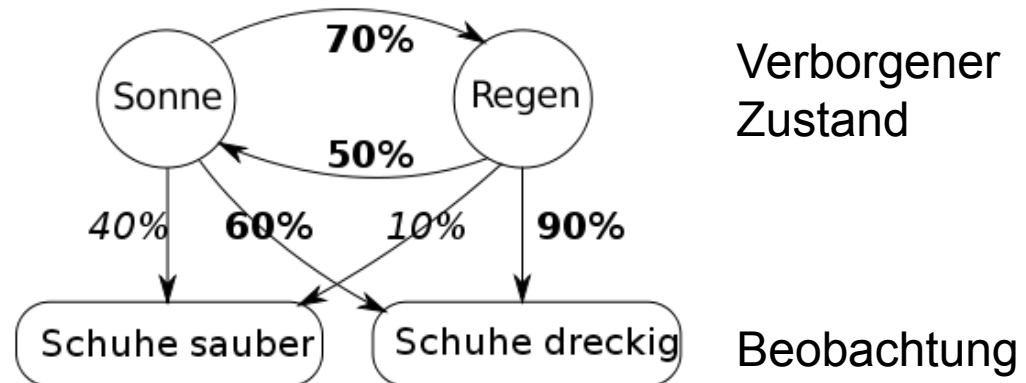
Die Übergangswahrscheinlichkeiten a_{ij} und b_{ij} müssen in der Trainingsphase des HMM hergeleitet werden.

Wettervorhersage mit Hidden Markov Modell

Ein Gefangener im Kerker verliert
möchte das aktuelle Wetter
herausfinden.

Er weiß, dass auf einen sonnigen
Tag zu 70 % ein Regentag folgt und
dass auf einen Regentag zu 50 %
ein Sonnentag folgt.

Weiß er zusätzlich, dass die Schuhe
der Wärter bei Regen zu 90 %
dreckig, bei sonnigem Wetter aber
nur zu 60 % dreckig sind, so kann er
durch Beobachtung der
Wärter Schuhe Rückschlüsse über
das Wetter ziehen.



Generkennung mit Hidden Markov Modellen

Bei der Generkennung möchte man bestimmen, wo in einem Genom Exons (E) und Introns (I) sind.

Der Output ist die bekannte Genomsequenz.

Aus dieser soll jedem Basenpaar der günstigste verborgene Zustand (E/I) zugeordnet werden.

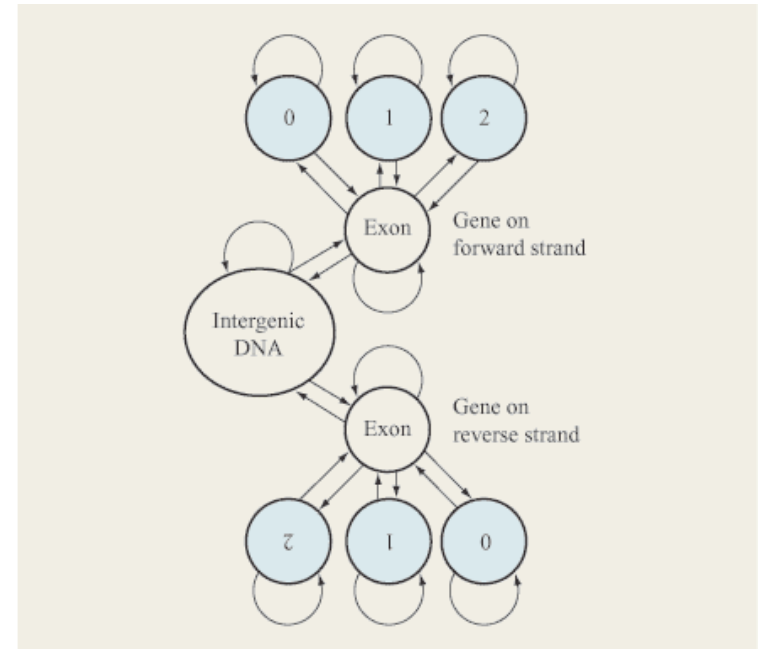


Figure 2

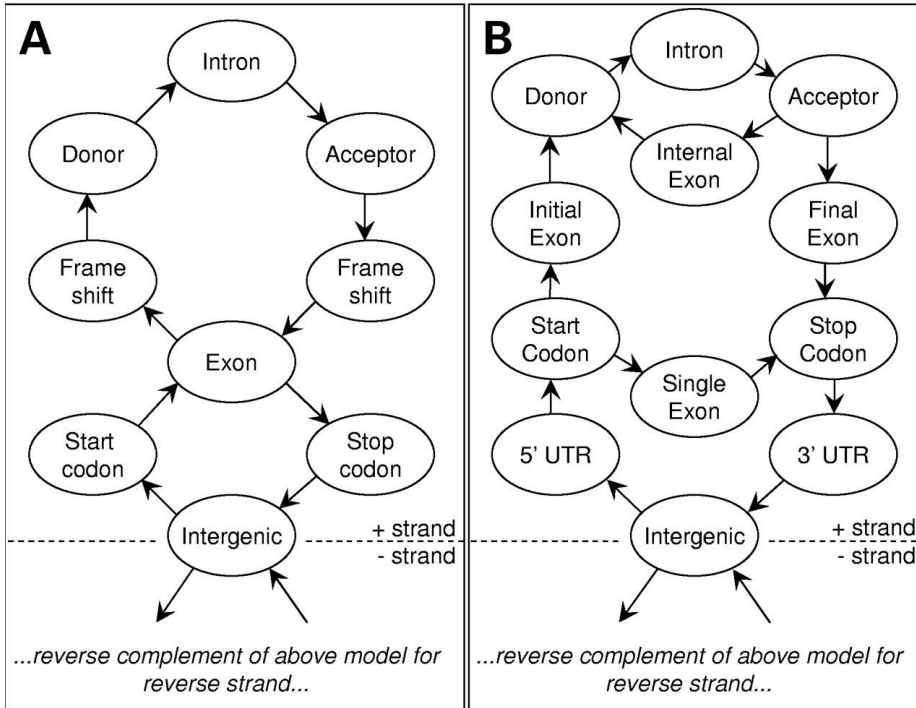
Abbreviated gene HMM model. The HMM is split into two symmetrical parts: genes on the forward or reverse strand of the DNA sequence (DNA sequence can be read in two directions). Each gene model contains a central exon state which has an emission of nucleotides tuned to recognize protein coding regions. 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.

TIGR: GlimmerM, Economy und Unveil

Topologien von
Unveil
 283-Zustands-HMM

Economy
 23-Zustands-GHMM

Für Markov-Modelle gilt:
 Zustand des i -ten Buchstaben
 hängt nur von seinem direkten
 Vorgänger, dem $(i-1)$ -ten
 Buchstaben ab.



Man kann jedoch auch ein
 sliding window einer bestimmten
 Breite benutzen, um der zentralen
 Residue des Fensters z.B. die
 mittlere Hydrophobizität in
 diesem Fenster zuzuordnen.

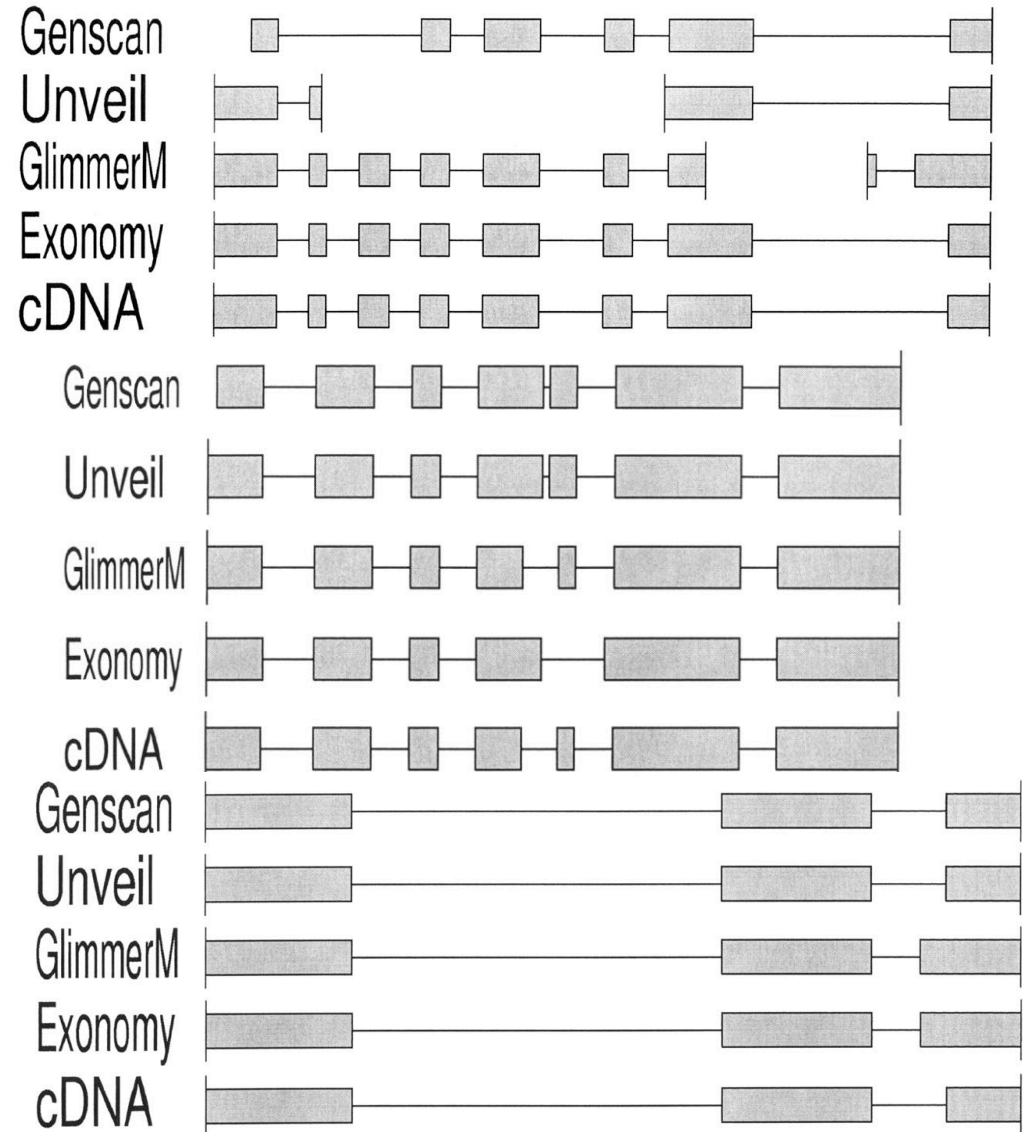
Majoros et al. Nucl. Acids. Res. 31, 3601 (2003)

Methoden funktionieren nicht überall

Ein Beispiel, in dem Exonomy die Gene richtig erkennt.

Ein Beispiel, in dem GlimmerM die Gene richtig erkennt.

Ein Beispiel, in dem Unveil die Gene richtig erkennt (auch Genscan).



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

Zusammenfassung - Genvorhersage

Die Resultate der intrinsischen Genvorhersage werden zuverlässiger; dennoch sollte man sie stets 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 dessen biologischen Funktion zu finden bzw. zu beweisen.

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

Neue Arbeiten wenden sich nun ebenfalls RNA-kodierenden Genen zu.

Mathé et al. Nucl. Acids. Res. 30, 4103 (2002)

Promotervorhersage in *E.coli*

Um *E.coli* Promoter zu analysieren kann man eine Menge von Promotersequenzen bzgl. der Position alignieren, die den bekannten **Transkriptionsstart** markiert und in den Sequenzen nach konservierten Regionen suchen.

→ *E.coli* Promotoren enthalten 3 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

Machbarkeit der Motivsuche mit dem Computer?

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

→ Das Problem lässt sich daher schwer eingrenzen

- so ist die Länge des gesuchten Motivs vorher nicht bekannt
- das Motiv braucht zwischen verschiedenen Promotern nicht stark konserviert sein.
- die Sequenzen, mit denen man nach dem Motiv sucht, brauchen nicht notwendigerweise dem gesamten Promoter entsprechen
- die zu untersuchenden Promotoren verschiedener Gene wurden oft durch einen Clusteralgorithmus in eine Gruppe eingeteilt, der ebenfalls Beschränkungen unterliegt.

Strategie 1

Wird seit der Verfügbarkeit von Microarray Gen-Expressionsdaten eingesetzt.

Durch Clustern erhält man Gruppen von Genen mit ähnlichen Expressionsprofilen (z.B. solche, die zur selben Zeit im Zellzyklus aktiviert sind)

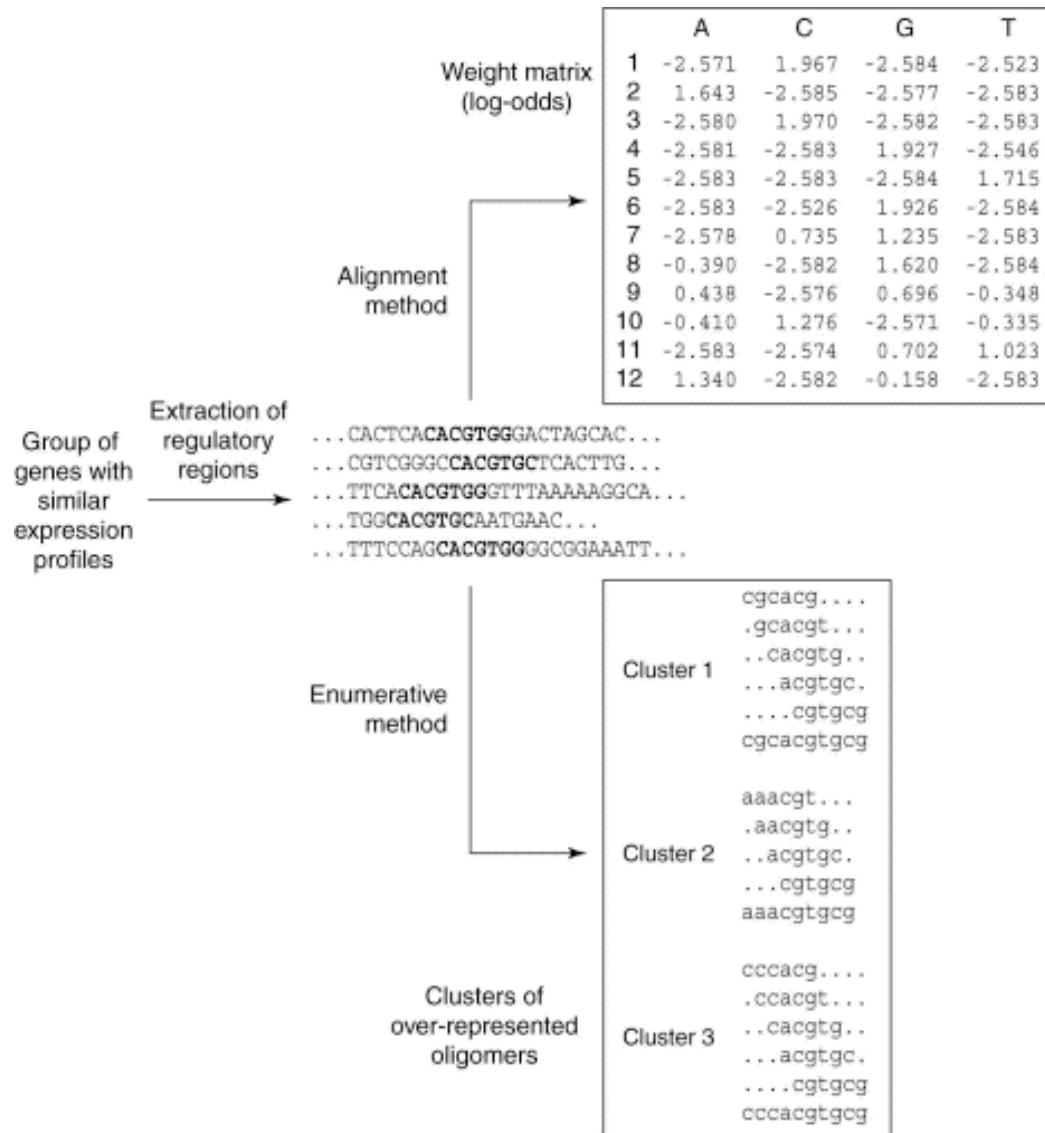
→ Hypothese, dass dieses Profil, zumindest teilweise, durch eine ähnliche Struktur der für die transkriptionelle Regulation verantwortlichen cis-regulatorischen Regionen verursacht wird.

Suche daher nach gemeinsamen Motiven in < 1000 Basen upstream Region.

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

Besser: suche nach dem gleichzeitigen Auftreten von 2 oder mehr Stellen in einem vorgegebenen Abstand! Dadurch wird die Suche empfindlicher.

Motif-Identifizierung



Ohler, Niemann
Trends Gen 17, 2 (2001)

Strategie 2: Erschöpfende Motivsuche in upstream-Regionen

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

→ Suche in der upstream-Region nach überrepräsentierten Motiven

- (1) Ordne Gene nach den überrepräsentierten Motiven.
- (2) Analysiere Gruppen von Genen, die Motive für Ko-Regulation in Microarray-Experimenten gemeinsam haben.
- (3) Betrachte überrepräsentierte Motive, die Gruppen von koregulierten Genen als mögliche Bindungsstellen markieren.

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

Erschöpfende Motivsuche in upstream-Regionen

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)

Positions-spezifische Gewichtsmatrix

Populäres Verfahren wenn es eine Liste von Genen gibt, die ein TF-Bindungsmotiv gemeinsam haben. Bedingung: gute MSAs müssen vorhanden sein.

Alignment-Matrix: wie häufig treten die verschiedenen Buchstaben an jeder Position im Alignment auf?

a) Alignment Matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| | A | A | T | T | G | A |
| | A | G | G | T | C | C |
| | A | G | G | A | T | G |
| | A | G | G | C | G | T |
| | 1 | 2 | 3 | 4 | 5 | 6 |
| A | 4 | 1 | 0 | 1 | 0 | 1 |
| C | 0 | 0 | 0 | 1 | 1 | 1 |
| G | 0 | 3 | 3 | 0 | 2 | 1 |
| T | 0 | 0 | 1 | 2 | 1 | 1 |

consensus: A G G T G N

$$\ln \frac{(n_{i,j} + p_i) / (N + 1)}{p_i} \approx \ln \frac{f_{i,j}}{p_i}$$

b) Weight Matrix

| | | | | | | | |
|---|------|------|------|------|------|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| A | 1.2 | 0 | -1.6 | 0 | -1.6 | 0 | |
| C | -1.6 | -1.6 | -1.6 | 0 | 0 | 0 | |
| G | -1.6 | .96 | .96 | -1.6 | .59 | 0 | |
| T | -1.6 | -1.6 | 0 | .59 | 0 | 0 | |

test sequence: A G G T G C

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.

Positions-spezifische Gewichtsmatrix

Beispiele für Matrizen, die von YRSA verwendet werden:

```
A [11 0 0 10 0 2 11 3 0 4 ]
C [ 0 0 11 0 6 3 0 8 0 0 ]
G [ 0 0 0 1 0 0 0 0 11 7 ]
T [ 0 11 0 0 5 6 0 0 0 0 ]
ID: MY0001
NAME: ABF1
SOURCE: Church Lab: ABF1.mot
LINK: http://atlas.med.harvard.edu/motifs/ABF1.mot
---
```

```
A [4 5 5 0 4 0 9 0 0 0 6 1 4 0 0 ]
C [0 0 0 0 0 9 0 8 9 8 0 1 1 1 3 ]
G [0 2 1 0 4 0 0 0 0 0 0 1 4 1 0 0 ]
T [5 2 3 9 1 0 0 1 0 1 2 3 3 8 6 ]
ID: MY0002
NAME: AFT1
SOURCE: Church Lab: AFT1.mot
LINK: http://atlas.med.harvard.edu/motifs/AFT1.mot
---
```

```
A [ 7 9 0 2 1 0 10 0 0 0 0 ]
C [ 2 0 0 1 0 10 0 10 0 0 0 ]
G [ 0 0 2 4 1 0 0 0 10 0 10 ]
T [ 1 1 8 3 8 0 0 0 0 10 0 ]
ID: MY0003
NAME: CBF1
SOURCE: Church Lab: CBF1.mot
LINK: http://atlas.med.harvard.edu/motifs/CBF1.mot
---
```

<http://forkhead.cgb.ki.se/YRSA/matrixlist.html>

Datenbank für eukaryotische Transkriptionsfaktoren: TRANSFAC

BIOBase / TU Braunschweig / GBF



Relationelle Datenbank

6 Dateien:

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

TRANSFAC[®] 6.0 - Public

TRANSFAC[®] is the database on eukaryotic transcription factors, their genomic binding sites and DNA-binding profiles.

- ♦ [Search](#)
- ♦ [Classification](#)
- ♦ [Documentation](#)
- ♦ [Fungal Metabolic](#)
- ♦ [Pax factors in TRANSFAC[®]](#)
- ♦ [The green site of TRANSFAC[®]](#)
- ♦ [Quality Management in TRANSFAC[®]](#)
- ♦ [TfBlast: Search Tool for Sequence Search in the TRANSFAC[®] Factor Table](#)

Wingender et al. (1998) J Mol Biol 284,241

Datenbank für eukaryotische Transkriptionsfaktoren: TRANSFAC

BIOBase / TU Braunschweig / GBF

```
gatctgtgga accdagggaa tctgcdtagt gctgtctcca agcaccttgg ccatgatgta aaccagaga aattagcadc tccatctcct
      NF-kappaB, q=0,833

      Ets
tccttattcc ccacccaaaa gtcattttcct cttagttcat tacctgggat tttgatgtct atgttccctc ctcgttattg atacacacac
      Ets, q=0,971
      NFAT, q=0,973

      NF-kappaB
agagagagac aaacaaaaaa ggaactttt gaaattcccc cagaaggttt tgagagttgt tttcaatggt gcaacaagtc agtttctagt
      Ets, q=0,973      NF-kappaB, q=0,875

      C/EBP
      C/EBP, q=0,926

      TSS
      TATA-box
      TATA-box,
      q=0,834
      Ets, q=0,971
      exon1
ttaagtttcc atcagaaagg agtagagtat ataagttcca gtaccagcaa cagcaggcaga agaaacaaca tctgtttcag ggccattgga
```

Figure 1. Application of specific profiles provided by the MatchTM program. Potential binding sites found by MatchTM are boxed, name of the transcription factor and score for the match are given under the sequence. (A) The immune-specific profile (with modified cut-offs) is applied to find potential binding sites within the promoter sequence of the human IL-12 p40 gene (EMBL accession no. AY008847, positions 2101 to 2460). Known binding sites for transcription factors are shown in bold, the name of the transcription factor is given above the sequence. The transcription start site (TSS) is indicated by an arrow.

Matys et al. (2003) Nucl Acid Res 31,374

Identifizierung von Repeats: RepeatMasker

RepeatMasker: durchsucht DNA Sequenzen auf

- eingefügte Abschnitte, die **bekannten Repeat-Motiven** entsprechen (dazu wird eine lange Tabelle mit bekannten Motiven verwendet)
- und
- auf **Regionen geringer Komplexität** (z.B. lange Abschnitt AAAAAAAAAA).

Output:

- detaillierte Liste, wo die Repeats in der Sequenz auftauchen und
- eine modifizierte Version der Input-Sequenz, in der die Repeats „**maskiert**“ sind, z.B. durch N's ersetzt sind.

Für die Sequenzvergleiche wird eine effiziente Implementation des Smith-Waterman-Gotoh Algorithmus verwendet.

<http://www.gene-regulation.com>

Zusammenfassung

Es gibt große Datenbanken (z.B. TRANSFAC) mit Informationen über Promoterstellen. Diese Informationen sind experimentell überprüft.

Microarray-Daten erlauben es, nach gemeinsamen Motiven von ko-regulierten Genen zu suchen.

Auch möglich: gemeinsame Annotation in der Gene Ontology etc.

TF-Bindungsmotive sind oft überrepräsentiert in der 1000 bp-Region upstream. Die klare Funktion dieser Bindungsmotive ist oft unbekannt.

Allgemein gilt:

- relativ wenige TFs regulieren eine große Anzahl an Genen
- es gibt globale und lokale TFs
- Gene werden üblicherweise durch mehr als einen TF reguliert

<http://www.gene-regulation.com>

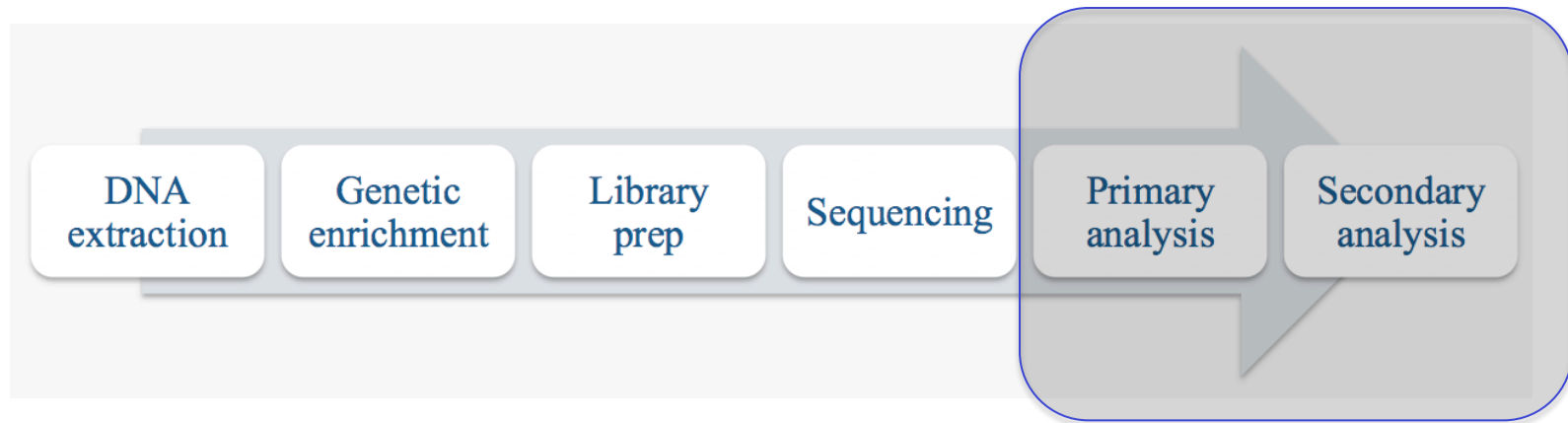
Prozessierung von NGS-Daten

- Whole Genome Sequencing (WGS)
- Anwendung von WGS für mikrobielle Isolate
- Qualitätskontrolle der Sequenzierungs-reads
- Alignment
- SNP calling
- Genomvisualisierung
- Genom Assemblierung

Hier wird dies Thema nur grob vorgestellt,
NGS-Prozessierung wird genauer in Vorlesungen von
Prof. Keller, Prof. Marschall und Dr. Schulz behandelt.

Danksagung für Folien: Mohamed Hamed

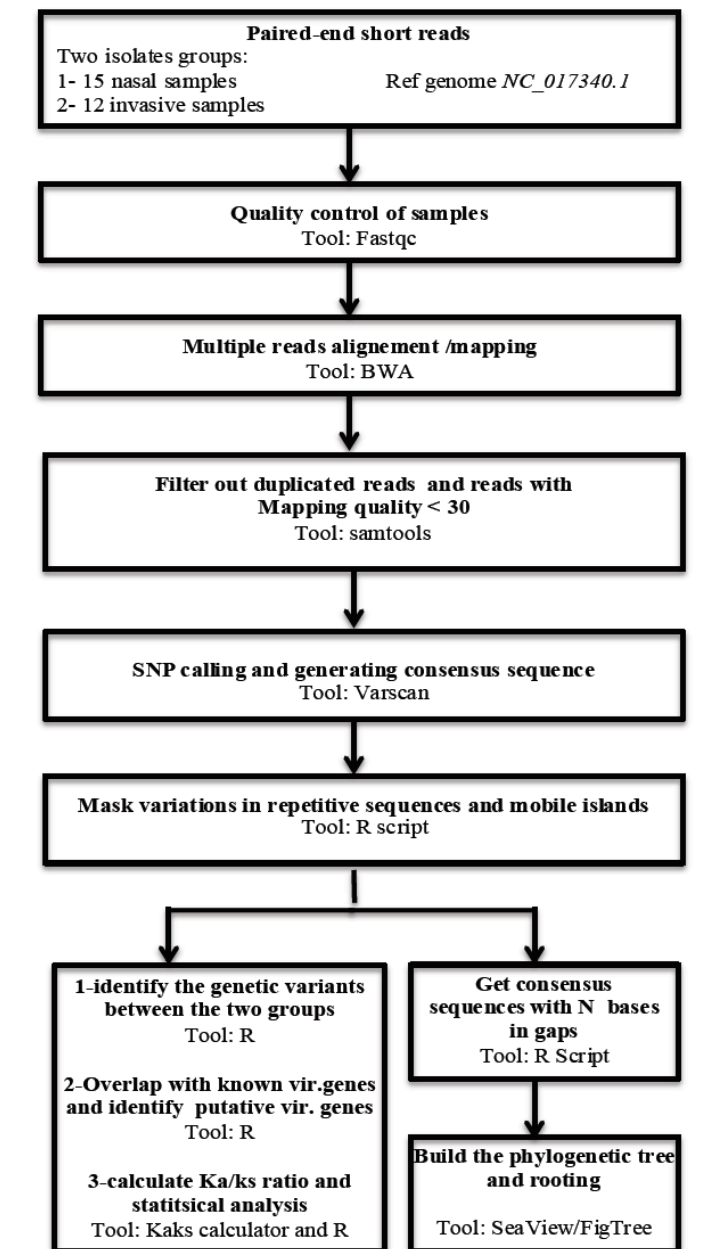
NGS Pipeline im Überblick



1. Extraktion der DNA aus biologischer Probe
2. Genetic enrichment: Manchmal soll nur eine kleine Region des Genoms sequenziert werden (einzelne Gene bzw. nur die Exons bei Sequenzierung von eukaryot. Genomen). Die Extraktion dieser Regionen aus dem Genome nennt man Anreicherung (enrichment).
3. Vorbereitung der Bibliothek (Library prep): Für viele Sequenziermaschinen muss die DNA für die Sequenzierung vorbereitet werden.
4. Die eigentliche Sequenzierung
5. Rohanalyse (primary analysis): Alignment / Assemblierung, SNP calling
6. Eigentliche Analyse (secondary analysis): Identifizierung von kausalen SNPs variants, phänotypische Charakterisierung (z.B. Virulenzfaktoren)

Wir konzentrieren uns auf die Schritte 5 und 6

WGS Pipeline für bakterielle Phylotypisierung



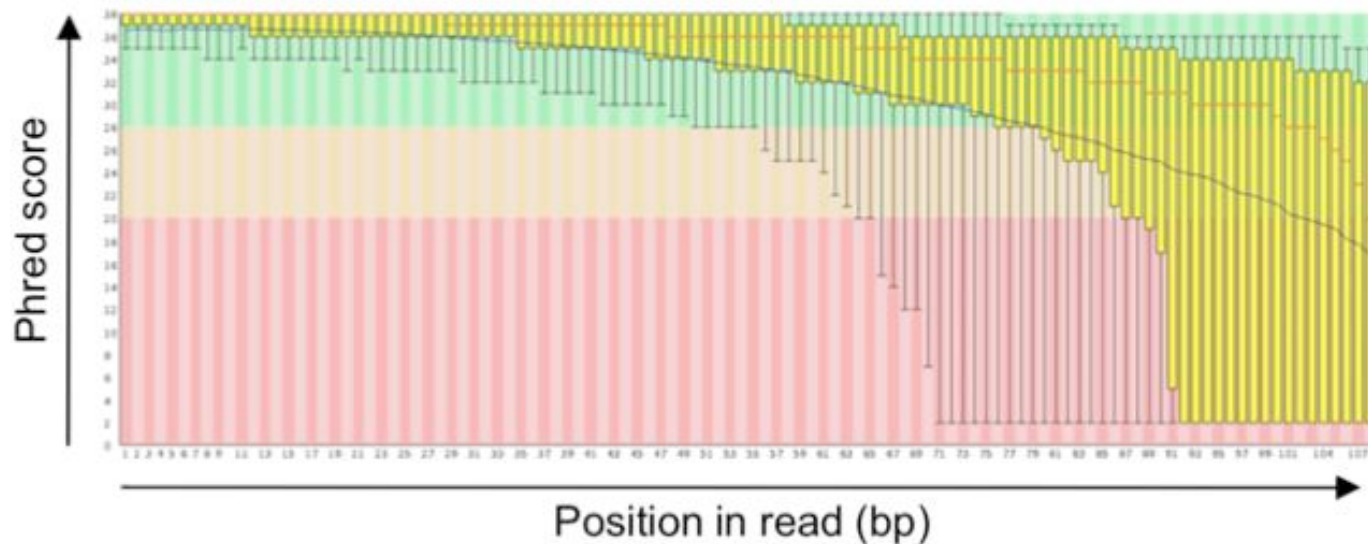
Quality (Phred) score

Phred Score (Q):

$$Q = - 10 \times \log_{10} P$$

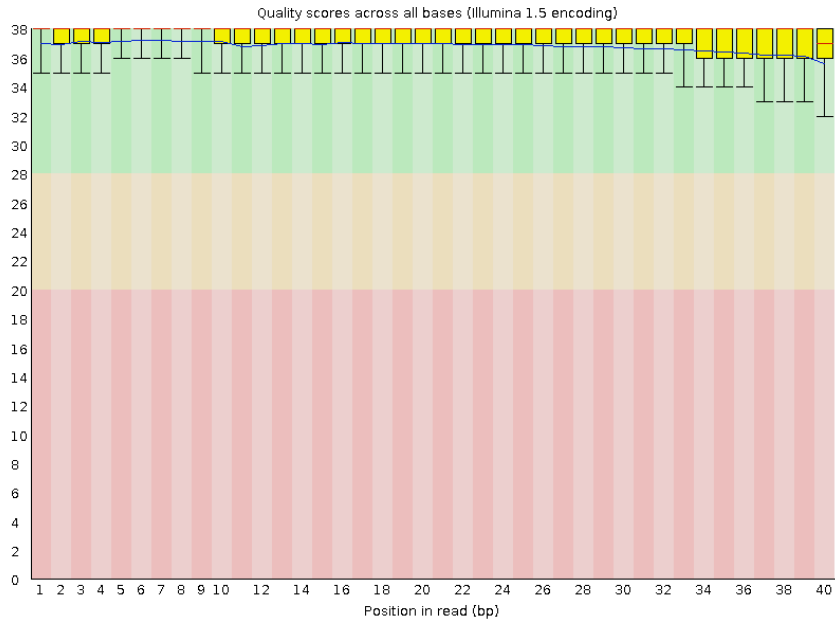
P ist eine Abschätzung für den Fehler des Base-calling aus den Rohdaten der Sequenzierung. D.h. ein Fehler von 0.1% (10^{-3}) ergibt $Q = 30$.

Base Qualitäts-scores nehmen üblicherweise am Ende der reads ab
Deshalb werden die reads vor oder während des Alignment-Schritts „getrimmt“,
d.h. gekürzt.

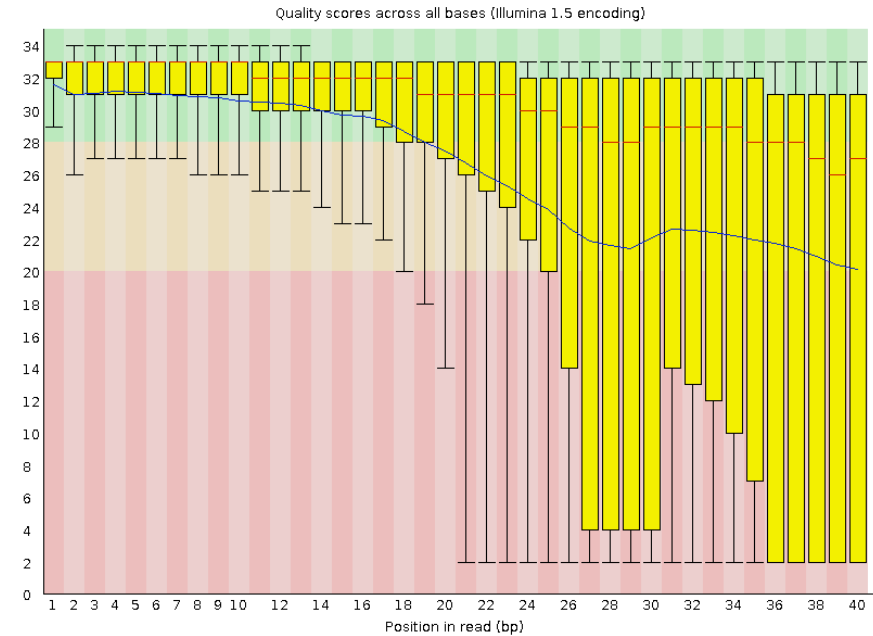


Qualitätskontrolle - FASTQC

✔ Per base sequence quality



✘ Per base sequence quality



Die Sequenzqualität pro Base ist im linken Beispiel durchgehend sehr hoch (d.h. $Q > 30$),

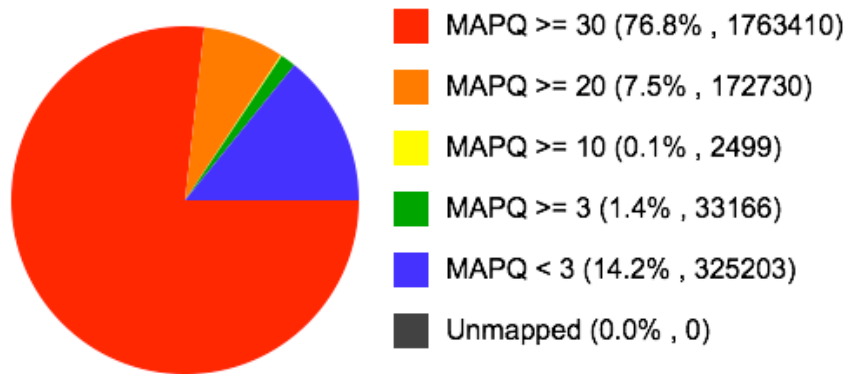
im rechten Beispiel etwa ab Position 20 aber unzuverlässig.

Qualitätskontrolle im Alignment-Schritt

Auch bei der Alignierung mit dem Referenz-Genom muss bewertet werden, ob den Reads zweifelsfreie Positionen zugeordnet werden können.

SAMStat: monitoring biases in next generation sequencing data

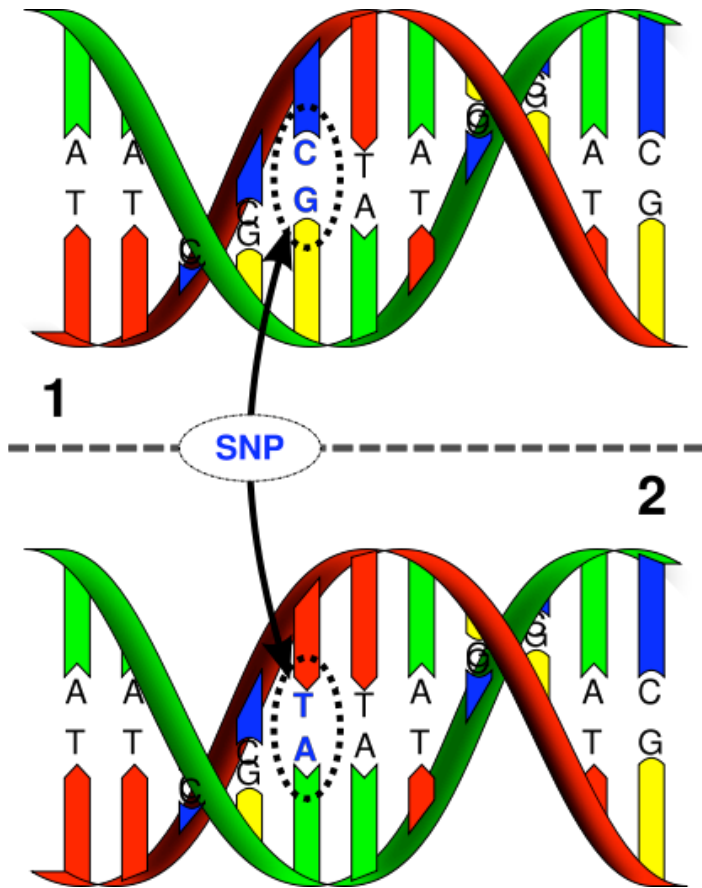
Timo Lassmann^{*}, Yoshihide Hayashizaki and Carsten O. Daub



Verteilung der Mapping
Qualitätsscores

- Reads werden entfernt, deren Mapping-Qualität geringer als 30 ist, d.h. die Fehlerwahrscheinlichkeit, dass der read auf eine andere Region gemappt werden sollte, ist 0.1% und höher.
- Entfernung von duplizierten reads, da diese die Qualität des SNP-Calling beeinflussen.

Biologie von SNP-Mutationen



Verschiedene menschliche Genome unterscheiden sich etwa an jeder 1000ten Base.

Die meisten Variationen sind Unterschiede einzelner Basen.

Polymorphismus: vererbter Unterschied

Somatische Mutation: erworbener Unterschied

http://www.science.marshall.edu/murraye/341/Images/416px-Dna-SNP_svg.png

Mögliche Gründe für Abweichungen in Alignments

- Ein wahrer SNP
- Experimenteller Fehler
 - Fehler bei Präparierung der Bibliothek oder bei der PCR
 - Base calling Fehler während Analyse von Rohdaten
- Fehler beim Alignment oder beim Mapping-Schritt
- Fehler in der Sequenz des Referenzgenoms

- Gebräuchliche Software Tools:
 - Samtools/bcftools
 - Gatk
 - Varscan
 - Snp-mix
- Die Ausgabe des Alignments ist im VCF Format (Variant Call Format)

2-Integrative Genome Viewer (IGV)

<http://www.broadinstitute.org/igv/home>

The screenshot displays the IGV interface with the following components:

- Search Bar:** MRSA017340 | gj|387149188|ref|NC_017340.1 | :_017340.1|:2,039,212-2,039,312 Go
- Navigation Pane (Right):** id0 region, ID=id0, Dbxref=taxon:703339, ls_circular=true, gbkey=Src, genome=chromosome, mol_type=genomic DNA, strain=04-02981, region.
- Tracks:**
 - somatic.uni.vcf
 - SAMPLE
 - S4sfr.bam Coverage (0-36)
 - S4sfr.bam (Sequence: G A G G T T)
 - S5sfr.bam Coverage (0-48)
 - S5sfr.bam
 - S6sfr.bam Coverage (0-34)
 - S6sfr.bam
 - R17sfr.bam Coverage (0-34)
 - R17sfr.bam
 - R18sfr.bam Coverage (0-188)
 - R18sfr.bam (Sequence: A)
 - R19sfr.bam Coverage (0-315)
 - R19sfr.bam (Sequence: C)
 - Sequence (A T A G G A T T T A A A T T A T T T A T T T A T T T A T G A A G G G T G T G T A T T G T T A T T T A C C A)
 - Gene
- Central Panel (Right):**
 - mol_type=genomic DNA, strain=04-02981
 - gi|387149188|ref|NC_017340.1|:1-2821452
 - id73 region
 - Note=phage phiN315 is also present in Staphylococcus au ...
 - gbkey=mobile_element
 - id73 region
 - Note=phage phiN315 is also present in Staphylococcus au ...
 - gbkey=mobile_element
 - gi|387149188|ref|NC_017340.1|:2038751-2082550
 - id74 region
 - id74 region
 - gbkey=mobile_element
 - SA2981_1900 gene1961 gene
 - ID=SA2981_1900, Name=SA2981_1900, Dbxref=GeneID:12864764, gbkey=Gene, locus_tag=SA2981_1900
 - gene1961
 - ID=SA2981_1900, Name=SA2981_1900, Dbxref=GeneID:12864764, gbkey=Gene, locus_tag=SA2981_1900
 - gene
 - gi|387149188|ref|NC_017340.1|:2039245-2039505

Softwarewerkzeuge

32

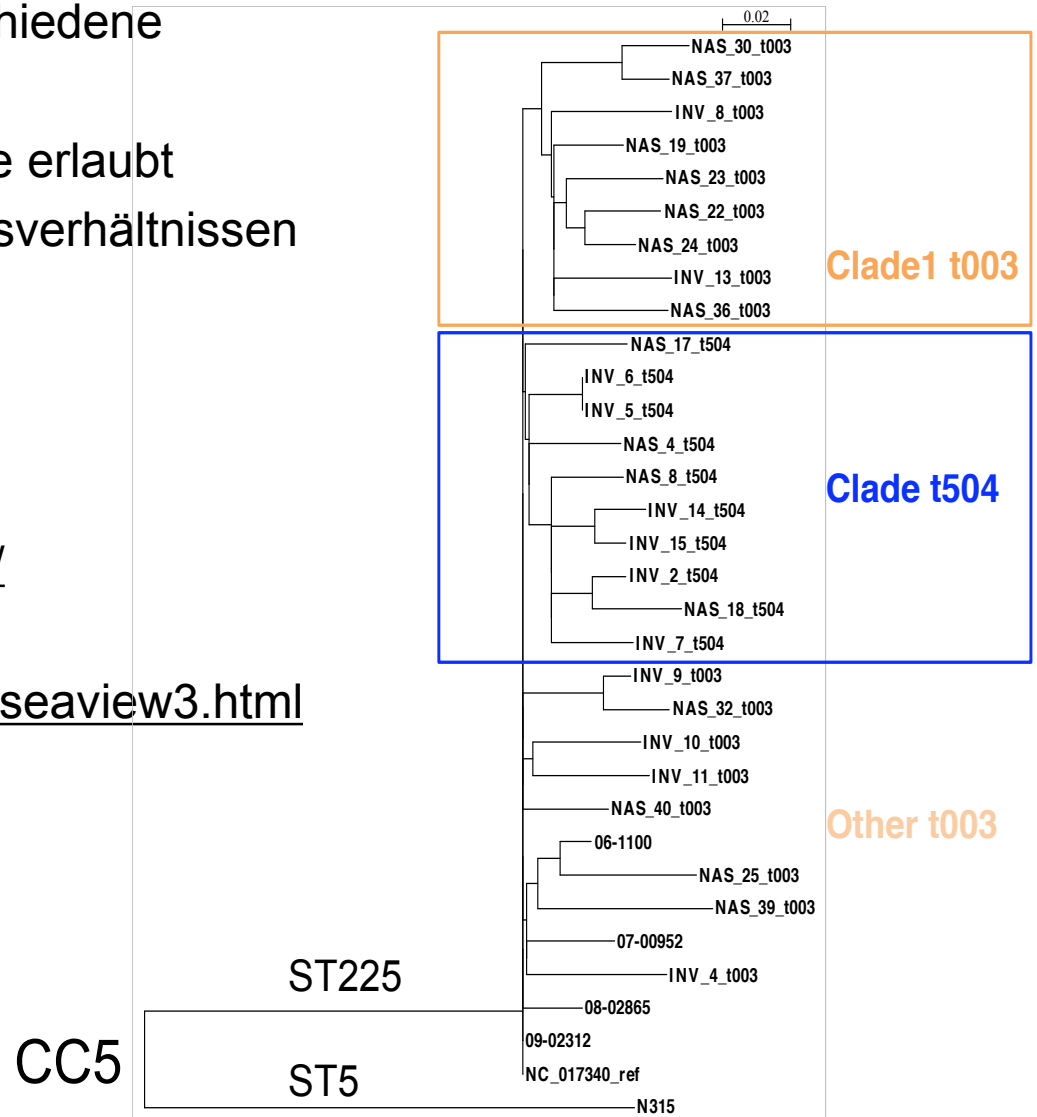
4. Vorlesung WS 2014/15

Phylogenetischer Baum

Beispiel: WGS-Sequenzen für verschiedene *Staphylococcus aureus* Stämme.

Vergleich von SNPs im core-genome erlaubt die Konstruktion von Verwandtschaftsverhältnissen zwischen den Stämmen.

- Tools
 - FigTree
<http://tree.bio.ed.ac.uk/software/>
 - SeaView
<http://pbil.univ-lyon1.fr/software/seaview3.html>
- Eingabe
 - Core genome SNP als Fasta-Datei
- Ausgabe
 - Phylogenetischer Baum



Whole Genome Alignment (WGA)

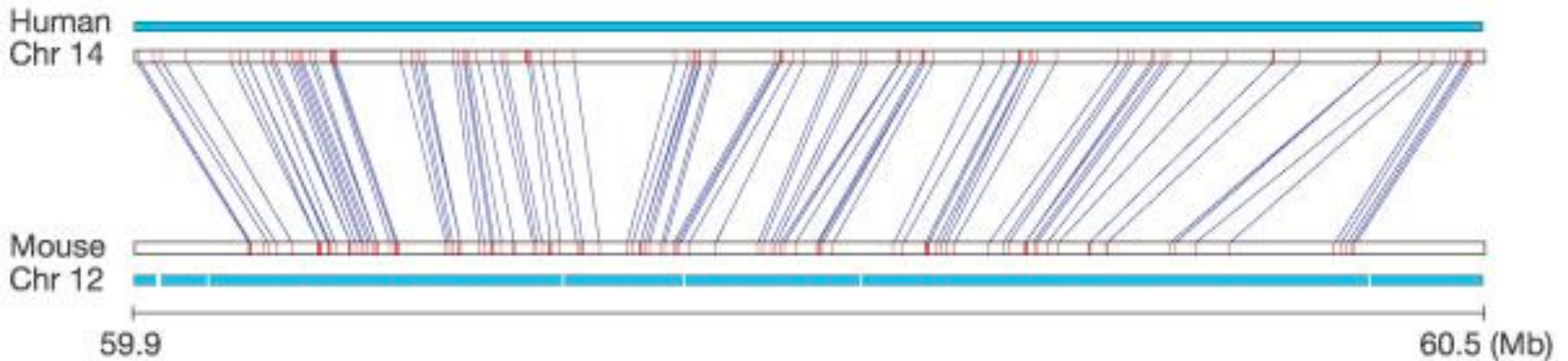
Wenn die genomische DNA-Sequenz eng verwandter Organismen verfügbar wird, ist die erste Frage, wie das Alignment zweier Genome aussieht.

Globale Genom-Alignments machen nur für eng verwandte Organismen Sinn.

Im anderen Fall muss man zuerst die genomischen Rearrangements betrachten.

Dann kann man die **systemischen Regionen** (Regionen, in denen Gen-Reihenfolge des nächsten gemeinsamen Vorfahrens in beiden Spezies konserviert blieb) betrachten und **lokale Genom-Alignments** dieser Regionen produzieren.

Konservierung von Syntenie zwischen Mensch und Maus



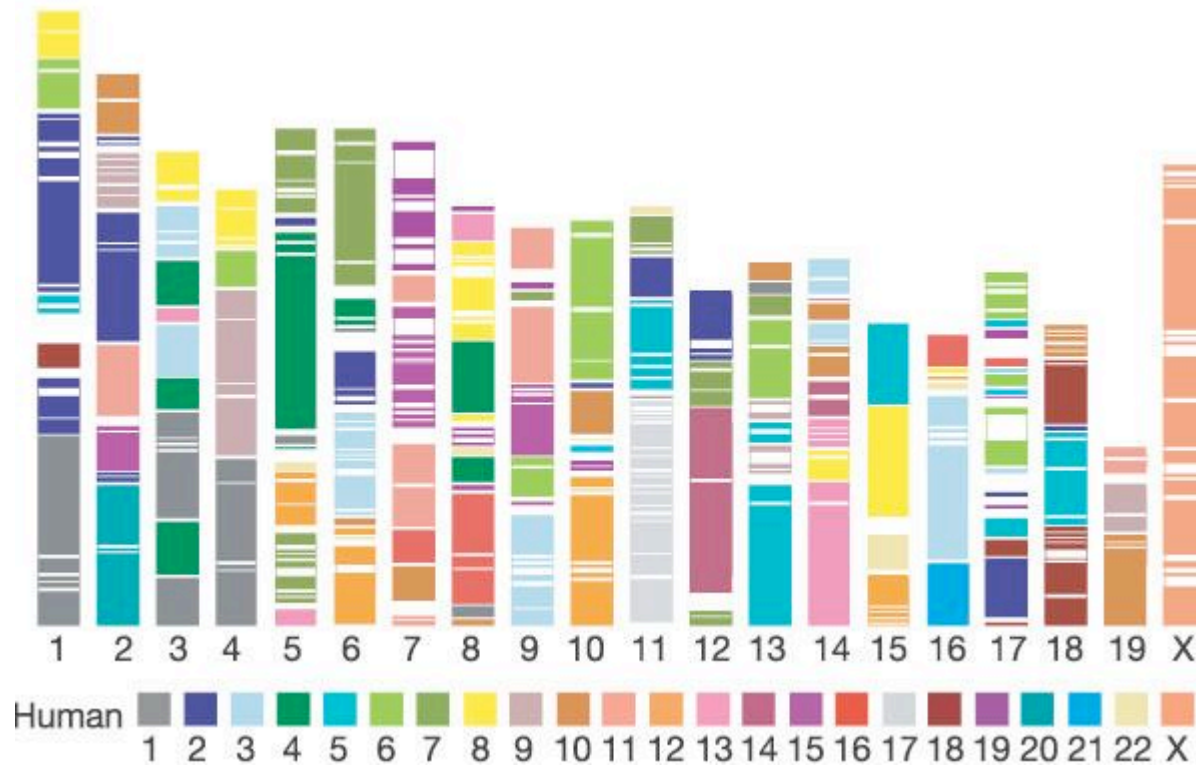
Ein typisches 510-kb Segment des Maus-Chromosoms 12, das mit einem 600-kb Stück des menschlichen Chromosom 14 verwandt ist.

Blaue Linien: reziprok eindeutige Treffer in beiden Genomen.

Rote Markierungen kennzeichnen die Länge der passenden Regionen.

Die Abstände zwischen diesen „Landmarks“ sind im Maus-Genom kleiner als im Mensch, was mit der 14% kürzeren Gesamtlänge des Genoms übereinstimmt.

Entsprechung syntenischer Regionen



342 Segmente und 217 Blöcke >300 kb mit konservierter Syntenie im Mensch sind im Maus-Genom markiert.

Jede Farbe entspricht einem bestimmten menschlichen Chromosom.

Sensitivität

Im globalen Mensch:Maus Alignment sind mehr als eine Millionen Regionen stärker als 70% konserviert (auf 100-bp Level)
– diese Regionen decken > 200 Million bp ab.

Nur 62% von ihnen werden von (lokalen) BLAT-Treffern abgedeckt.

Dies bedeutet, daß man 38% der konservierten Abschnitte nur durch das globale Alignment finden kann!

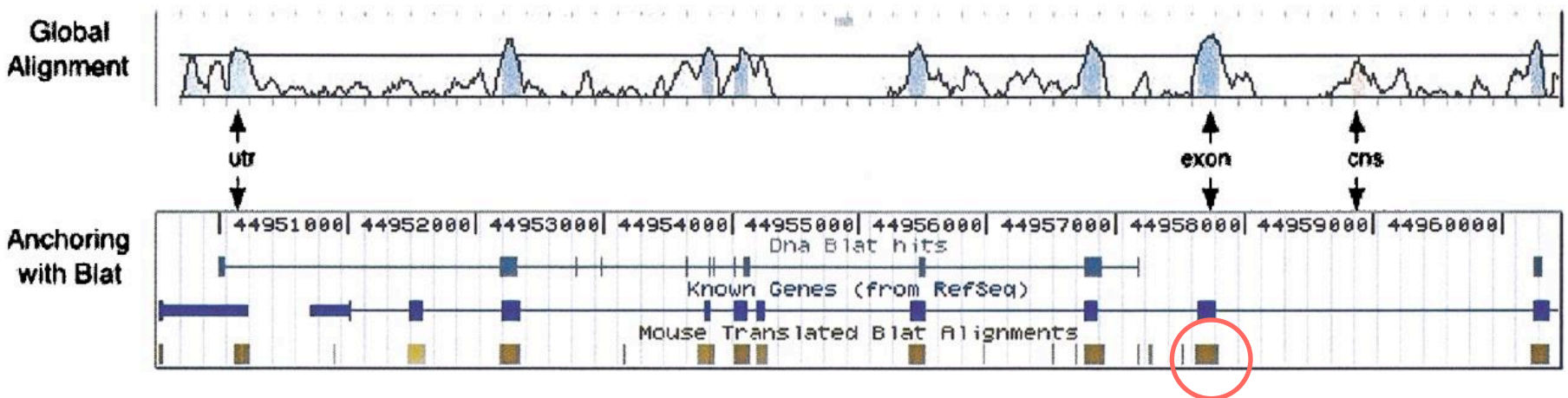
Idee: lokales Alignment soll als **Anker-Verfahren** für anschliessendes globales Alignment dienen.

Dadurch hofft man, viele zusätzliche konservierte Regionen ausserhalb der Anker-Regionen zu finden.

Couronne, ..., Dubchak, Genome Res. 13, 73 (2003)

hohe Sensitivität von globalen Alignments

Beispiel: das globale Alignment der mouse finished sequence NT_002570 gegen die Region, die mit BLAT-Ankern gefunden wurde, zeigt konservierte kodierende und nicht-kodierende Elemente, die mit BLAT nicht gefunden wurden.



Couronne, ..., Dubchak, Genome Res. 13, 73 (2003)

Ankerbasierte Methoden für WGA

Diese Methoden versuchen, sich entsprechende Teile der Buchstabenfolgen der betrachteten Sequenzen zu finden, die wahrscheinlich zu einem globalen Alignment gehören werden.

(Diese teilweisen Treffer können durch lokale Alignments gefunden werden).
Sie bilden „Anker“ in den beiden zu alignierenden Sequenzen.

In diesen Methoden werden zuerst die Ankerpunkte aligniert und dann die Lücken dazwischen geschlossen.

MUMmer ist eine sehr erfolgreiche Implementation dieser Strategie für das Alignment zweier genomischer Sequenzen.

Was ist MUMmer?

- A.L. Delcher *et al.* 1999, 2002 Nucleic Acids Res.
- <http://www.tigr.org/tigr-scripts/CMR2/webmum/mumplot>
- nimm an, dass zwei Sequenzen eng verwandt sind (sehr ähnlich)
- MUMmer kann zwei bakterielle Genome in weniger als 1 Minute alignieren
- nutzt **Suffix-Bäume** um Maximal Unique Matches zu finden
- Definition eines Maximal Unique Matches (MUM):
 - Eine Subsequenz, die in beiden Sequenzen genau einmal ohne Abweichungen vorkommt und in keine Richtung verlängert werden kann.
- Grundidee: ein MUM ausreichender Länge wird sicher Teil eines globalen Alignments sein.

```
Genome A:  tcgatcGACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAAcgactta  
Genome B:  gcattaGACGATCGCGCCGTAGATCGAATAACGAGAGAGCATAAtccagag
```

A maximal unique matching subsequence (MUM) of 39 nt (shown in uppercase) shared by Genome A and Genome B. Any extension of the MUM will result in a mismatch.

By definition, an MUM does not occur anywhere else in either genome.

Delcher et al. Nucleic Acids Res 27, 2369 (1999)

MUMmer: wichtige Schritte

- Erkenne MUMs (Länge wird vom Benutzer festgelegt)

ACTGATTACGTGAACTGGATCCA
ACTCTAGGTGAAGTGATCCA



ACTGATTAC**GTGAA**CTGGAT**TCCA**
ACTCTAG**GTGAA**GTGAT**TCCA**



1 10 20
ACTGATTAC**GTGAA**CTGGAT**TCCA**

1 10 20
ACTC--TAG**GTGAA**GTG-A**TCCA**

Definition von MUMmers

- Für zwei Strings $S1$ und $S2$ und einen Parameter l
- Der Substring u ist eine MUM Sequenz wenn gilt:
 - $|u| > l$
 - u kommt genau einmal in $S1$ und genau einmal in $S2$ (Eindeutigkeit) vor
 - Für jeden Buchstaben a kommt weder ua noch au sowohl in $S1$ als auch in $S2$ vor (Maximalität)

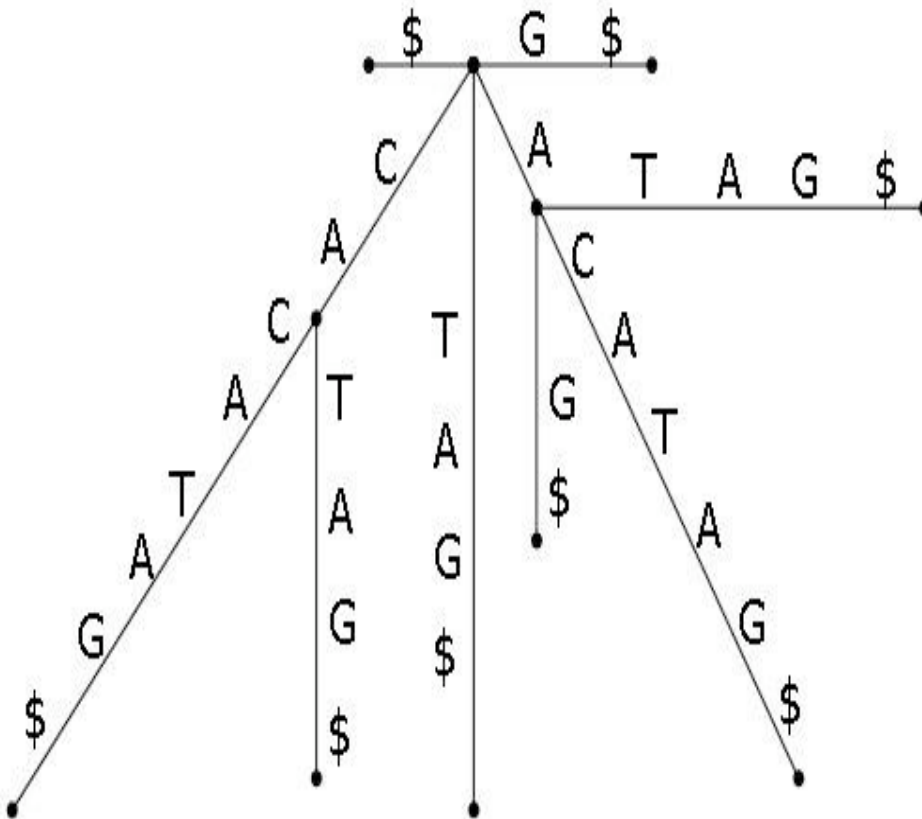
Wie findet man MUMs?

- Naiver Ansatz
 - Vergleiche alle Teilsequenzen von A mit allen Teilsequenzen von B.
Dies dauert $O(n^n)$
- verwende Suffix-Bäume als Datenstruktur
 - ein naiver Ansatz, einen Suffix-Baum zu konstruieren hat eine quadratische Komplexität in der Rechenzeit und dem Speicherplatz
 - durch cleverere Benutzung von Pointern gibt es lineare Algorithmen in Rechenzeit und Speicherplatz wie den Algorithmus von McCreight

Suffix-Bäume

Suffix-Bäume sind seit über 30 Jahren wohl etabliert.

CACATAG\$



Einige ihrer Eigenschaften:

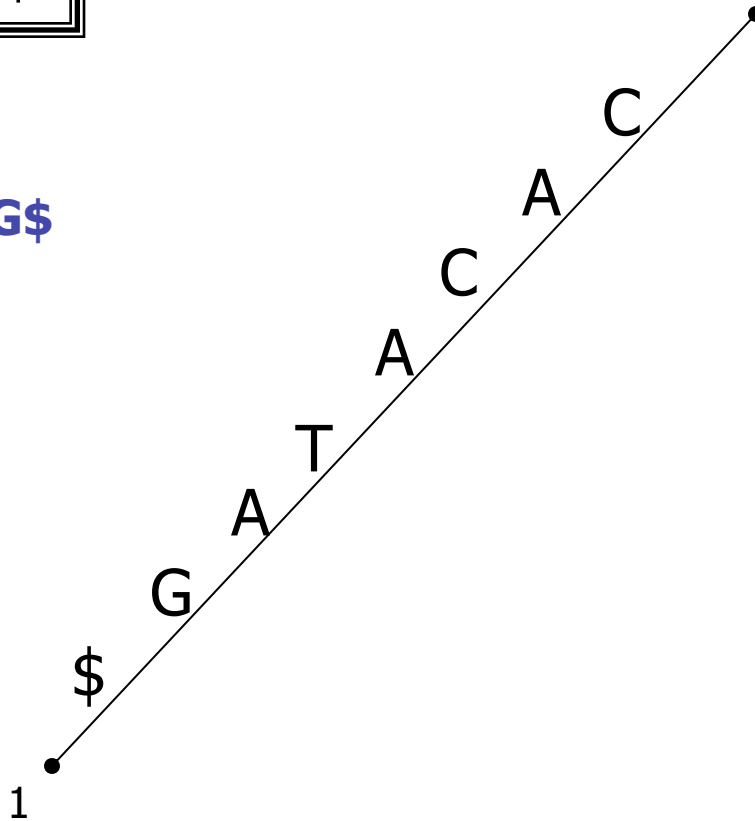
- ein "Suffix" beginnt an jeder Position i der Sequenz und reicht bis zu ihrem Ende.
- Eine Sequenz der Länge N hat N Suffixes.
- Es gibt N Blätter.
- Jeder interne Knoten hat mindest zwei Kinder.
- 2 Kanten aus dem selben Knoten können nicht mit dem selben Buchstaben beginnen.
- Am Ende wird \$ angefügt

Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$

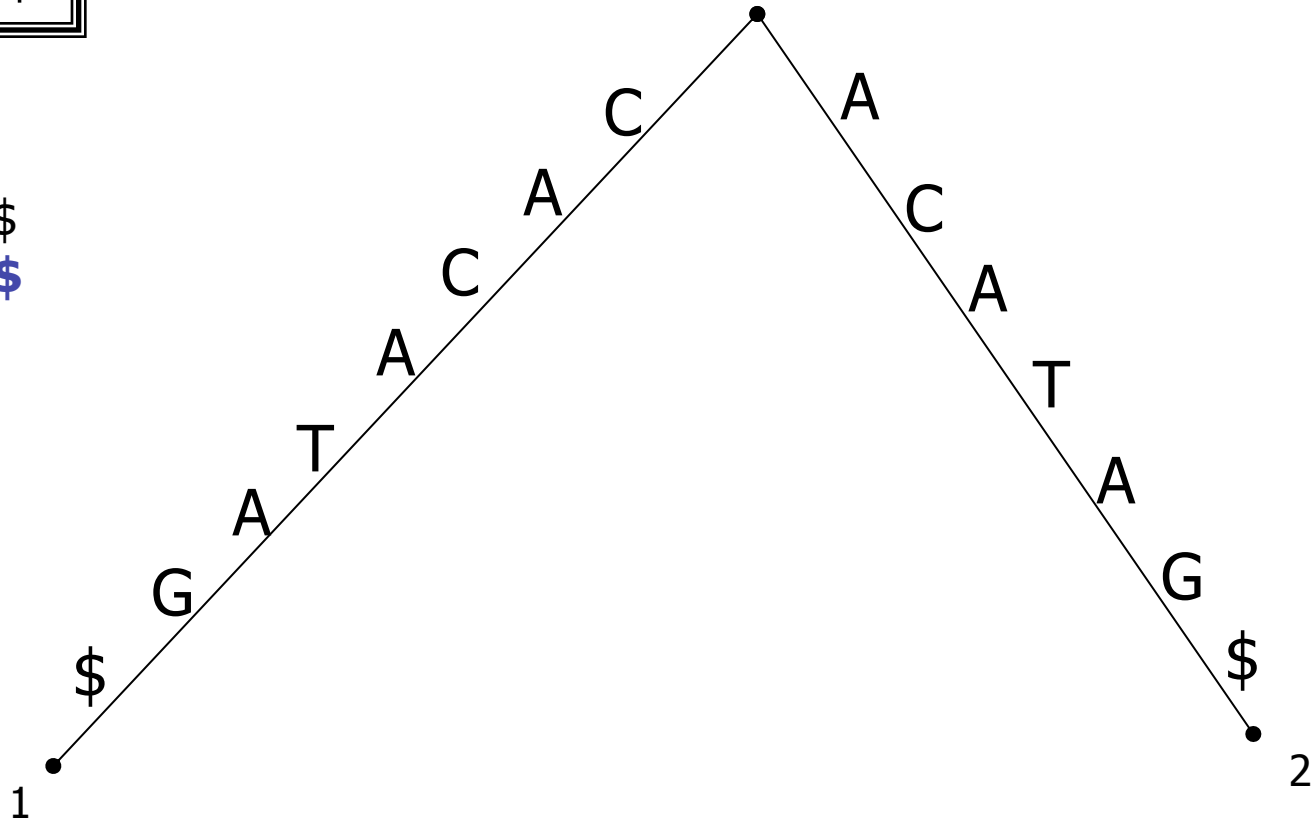


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$

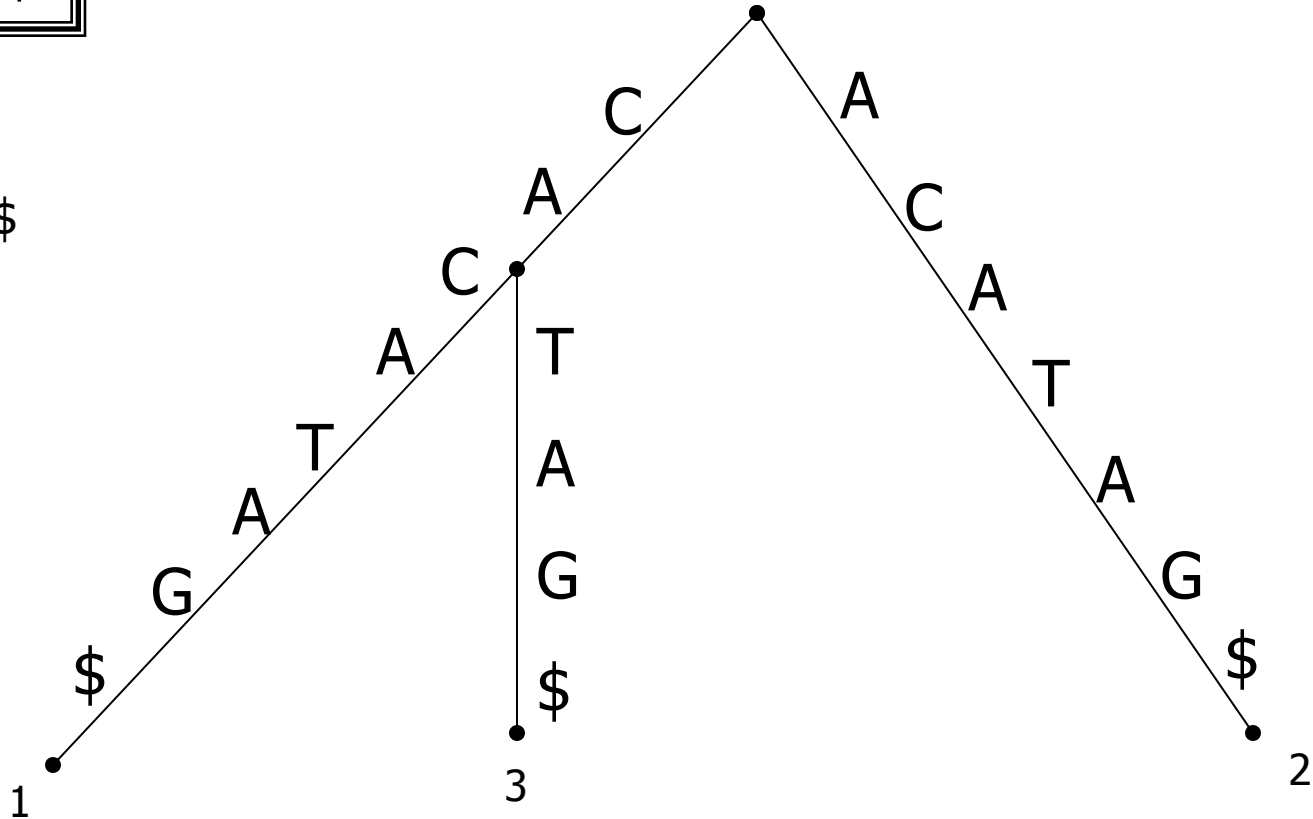


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$
3. **CATAG\$**

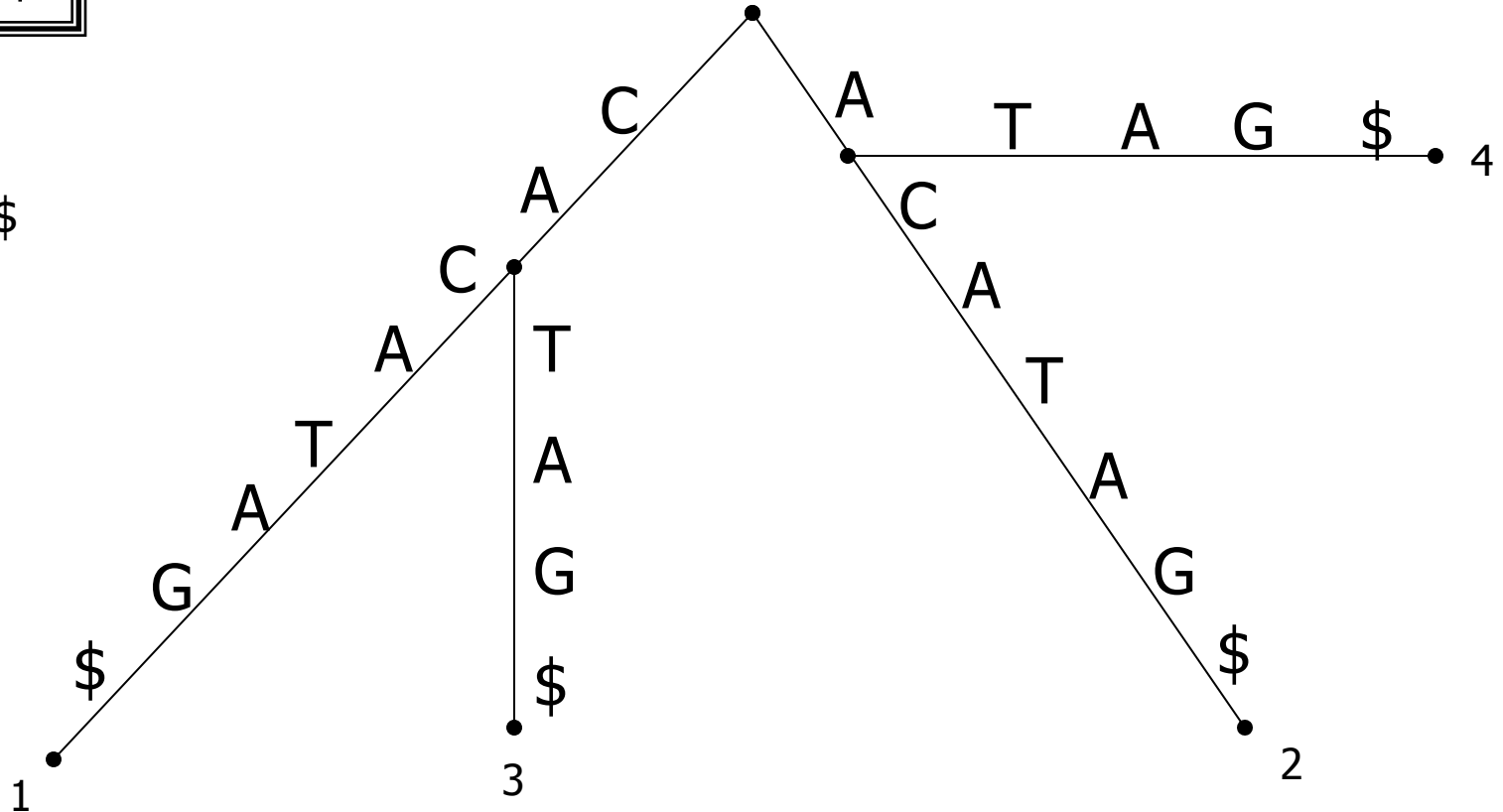


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$
3. CATAG\$
4. **ATAG\$**

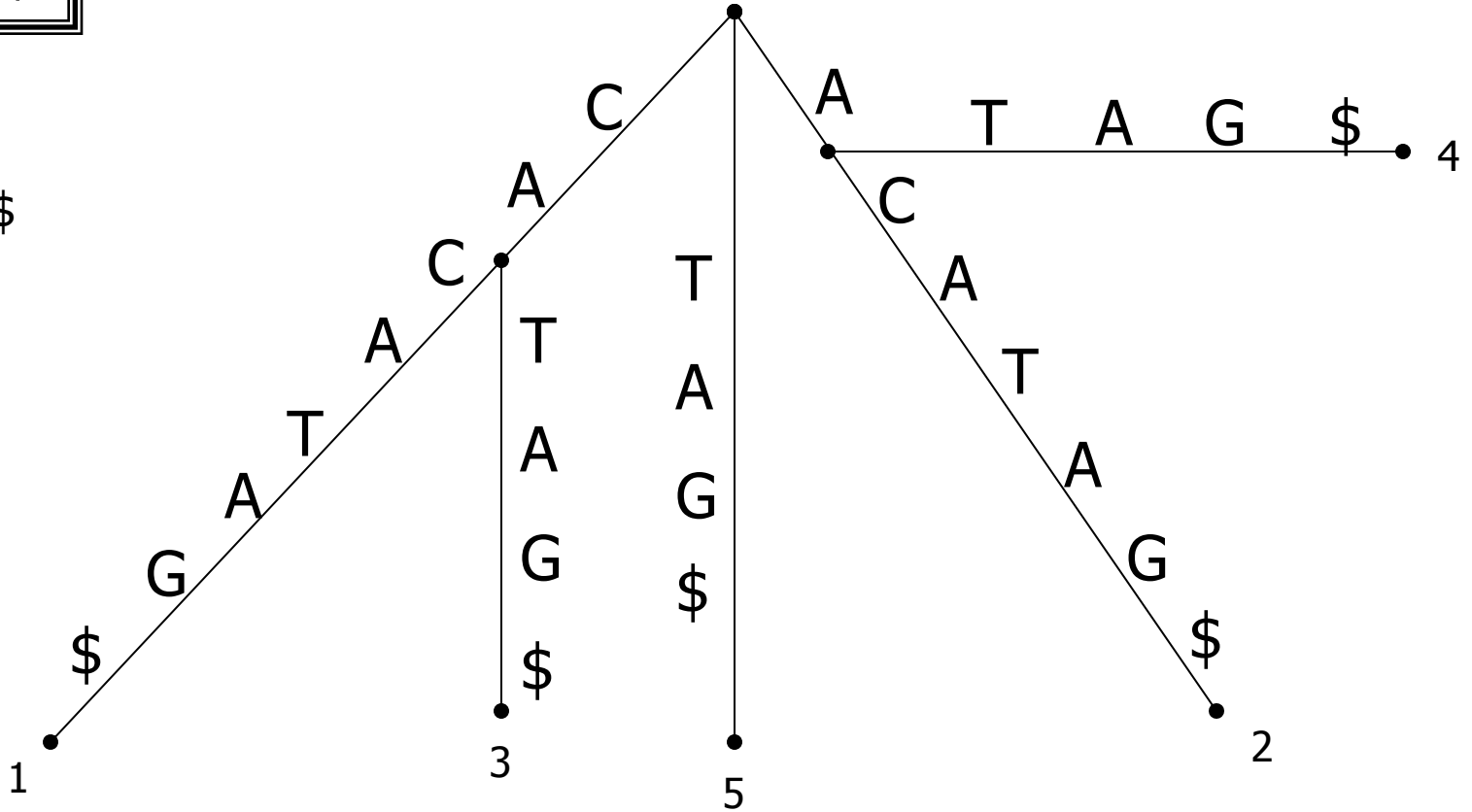


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$
3. CATAG\$
4. ATAG\$
- 5. TAG\$**

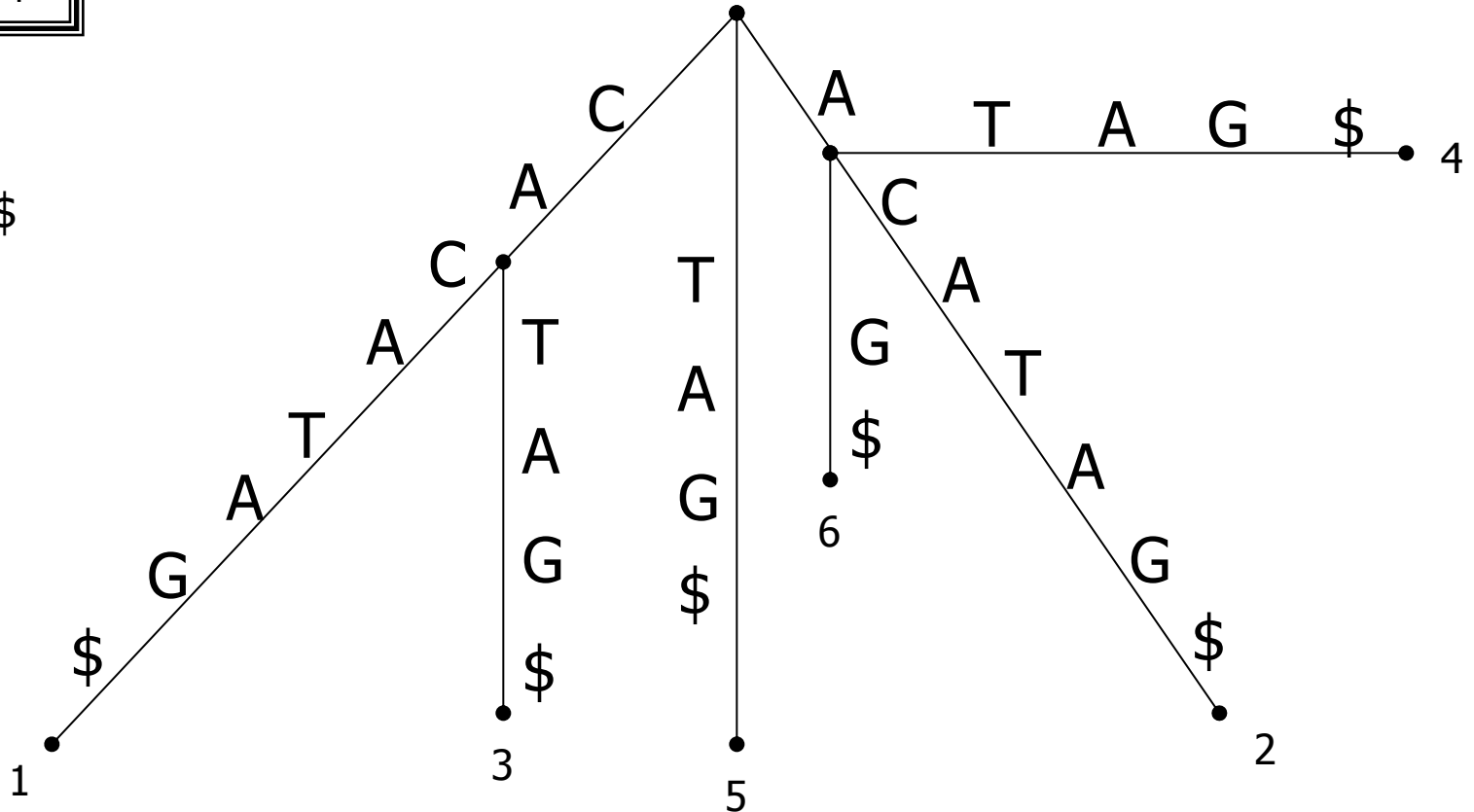


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$
3. CATAG\$
4. ATAG\$
5. TAG\$
- 6. AG\$**

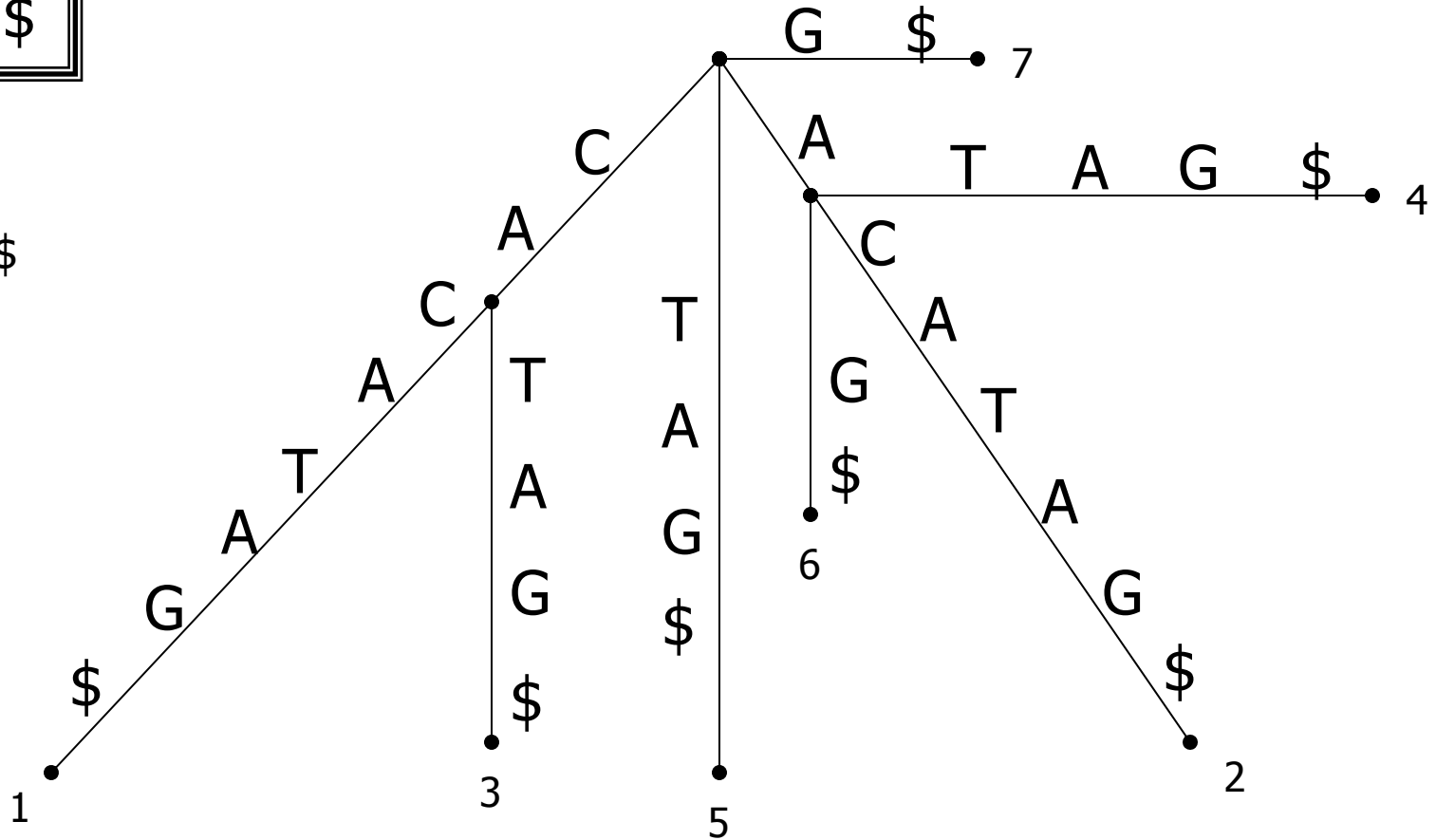


Konstruktion eines Suffix-Baums

CACATAG\$

Suffixes:

1. CACATAG\$
2. ACATAG\$
3. CATAG\$
4. ATAG\$
5. TAG\$
6. AG\$
7. **G\$**

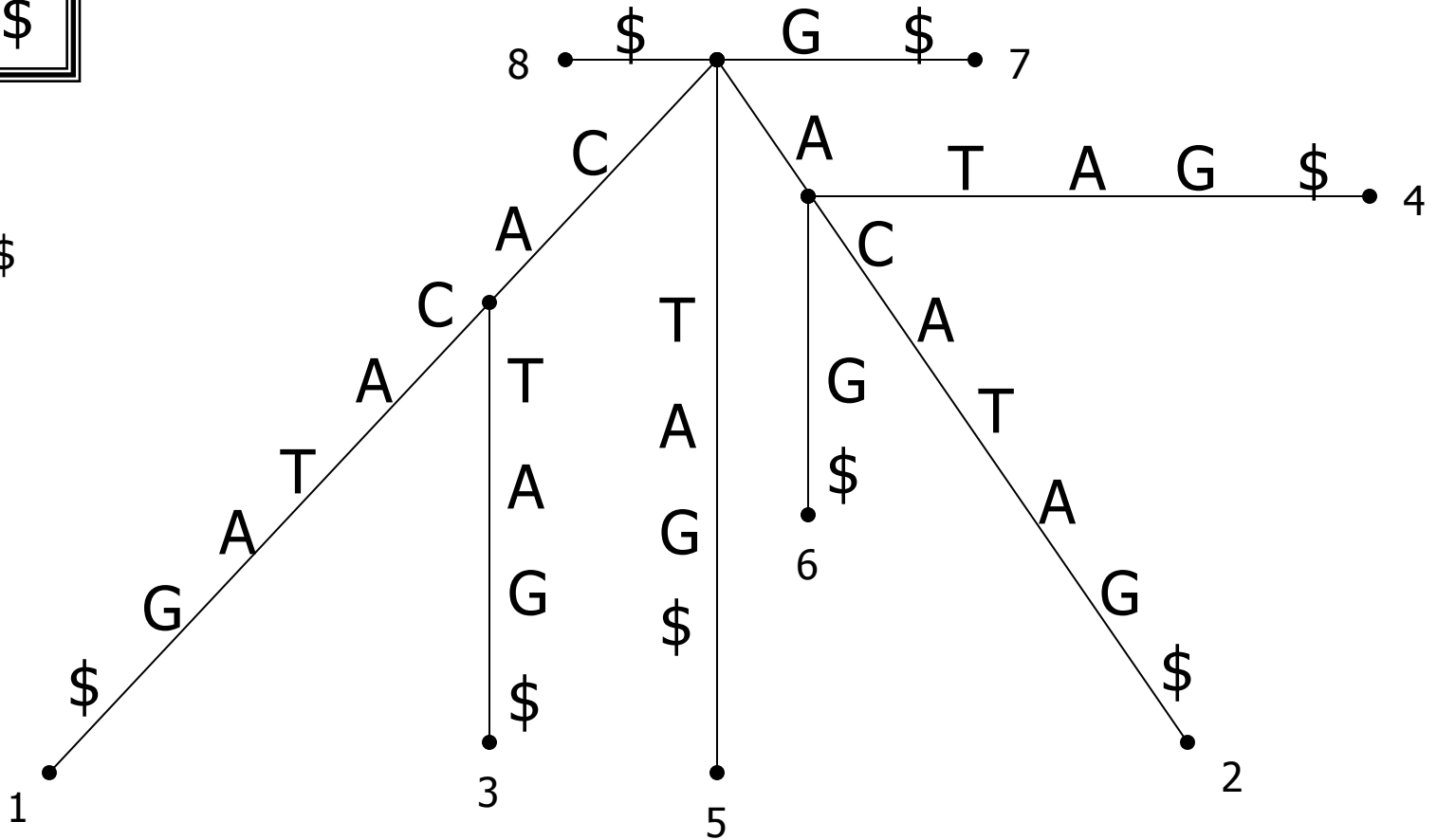


Konstruktion eines Suffix-Baums

CACATAG\$

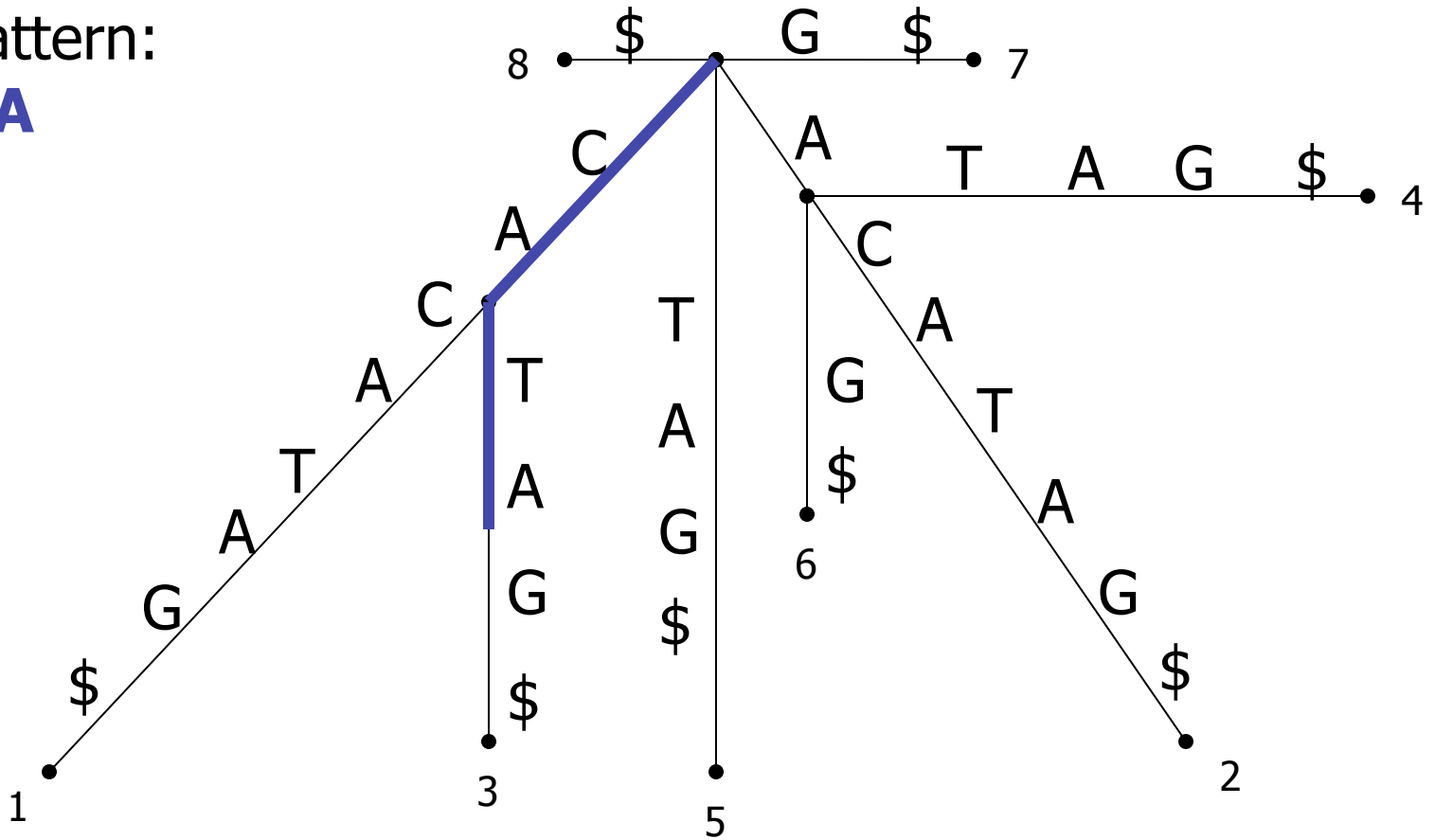
Suffixes:

1. CACATAG\$
2. ACATAG\$
3. CATAG\$
4. ATAG\$
5. TAG\$
6. AG\$
7. G\$
8. \$



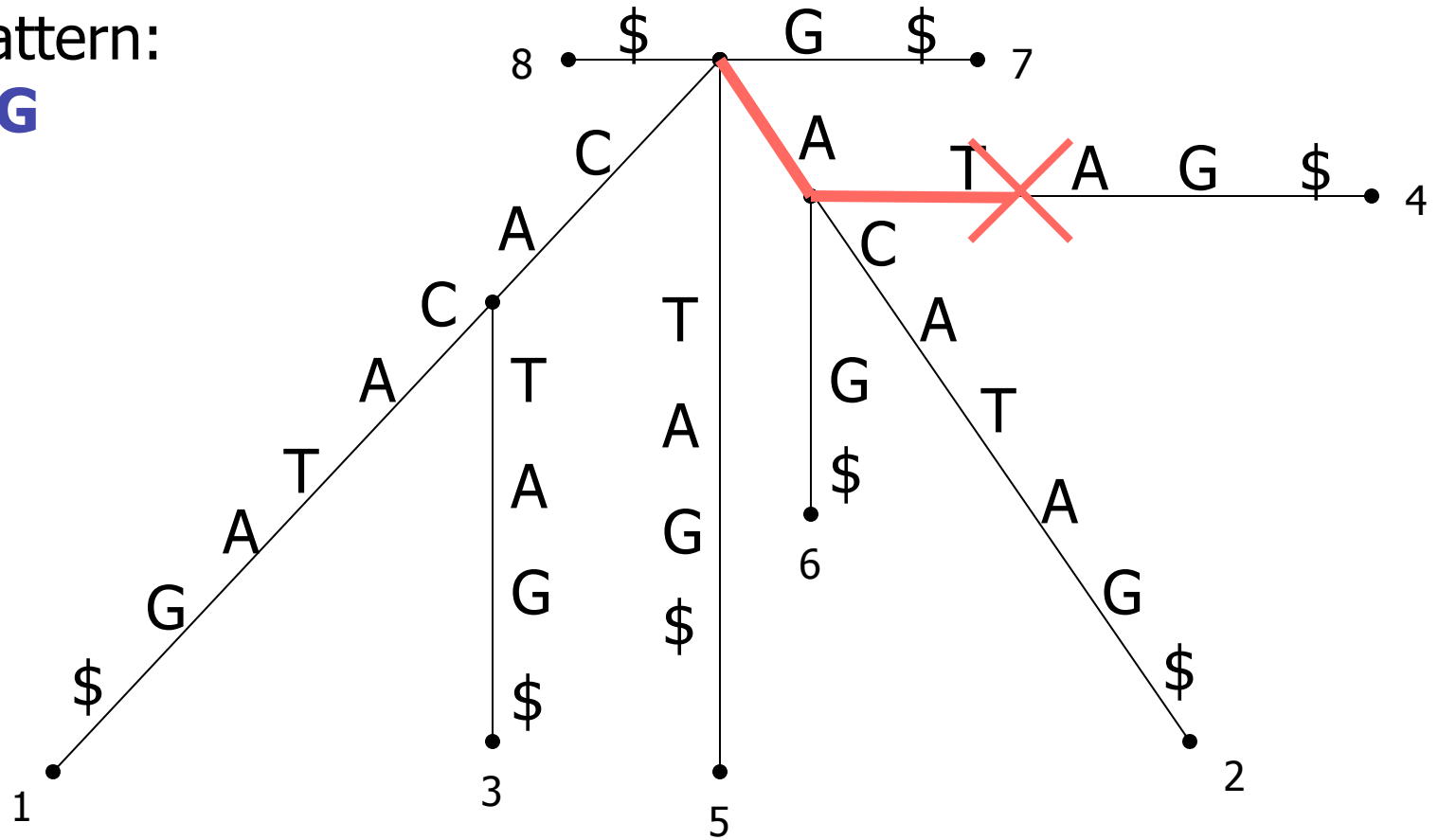
Suchen in einem Suffix-Baum

Search Pattern:
CATA



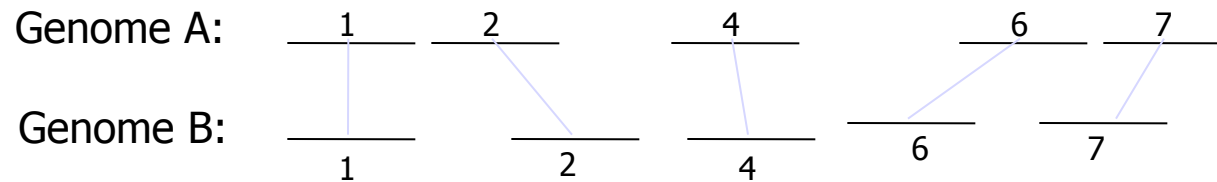
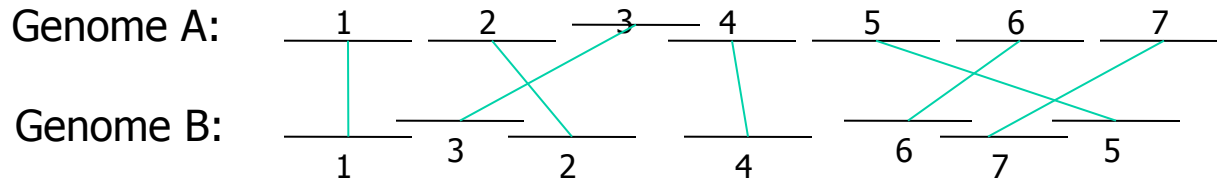
Suchen in einem Suffix-Baum

Search Pattern:
ATCG



Sortieren der MUMs

- MUMs werden nach ihren Positionen in Genom A sortiert



Jeder MUM ist nur mit seiner Nummer gekennzeichnet, ohne Berücksichtigung seiner Länge.

Das obere Alignment zeigt alle MUMs.

Die Verschiebung von MUM 5 in Genom B zeigt eine Transposition an.

Die Verschiebung von MUM 3 könnte ein Zufallstreffer oder Teil einer inexakten Repeat-Sequenz sein.

Unteres Alignment: suche in beiden Genomen die längste gemeinsam ansteigende Folge an Subsequenzen

Beispiel: Alignment zweier Mikroorganismen

Das Genom von *M.genitalium* ist nur etwa 2/3 so lang wie das von *M.pneumoniae*.

Obere Abbildung: FASTA-Alignment von *M.genitalium* und *M.pneumoniae*.

Mitte: Alignment mit 25mers

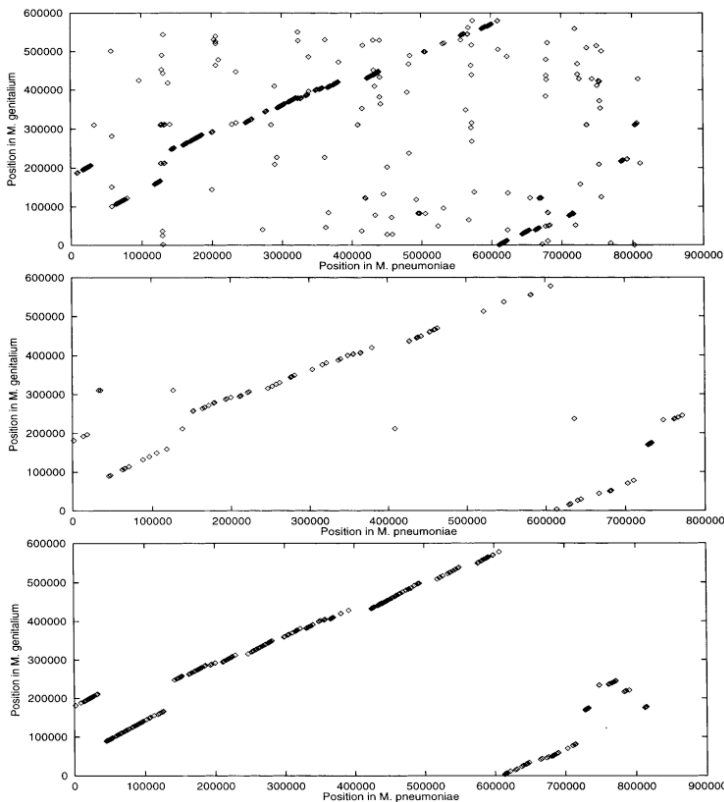
Unten: Alignment mit MUMs. 5 Translokationen.

Ein Punkt bedeutet jeweils einen Treffer zwischen den Genomen.

FASTA-Plot: ähnliche Gene

25-mer-Plot: 25-Basen-Sequenz, die in beiden Sequenzen genau einmal vorkommt.

MUM-Plot: MUM-Treffer.



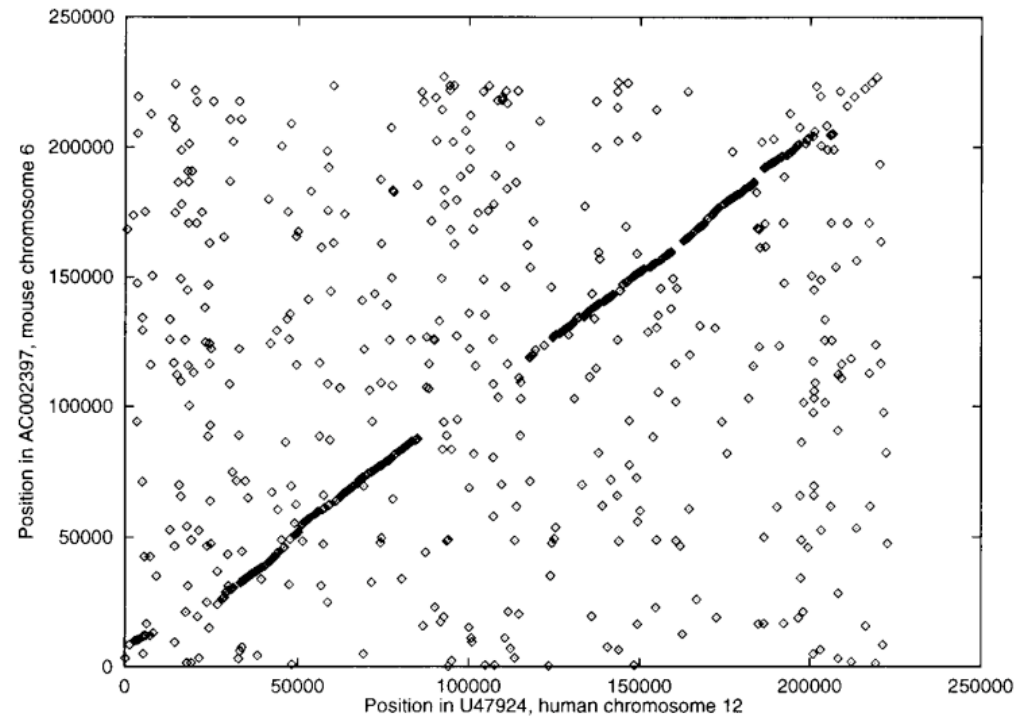
Delcher et al. Nucleic Acids Res 27, 2369 (1999)

Beispiel: Alignment Mensch:Maus

Alignment von weiter entfernt
liegenden Spezies:
Mensch gegen Maus.

Hier: Alignment einer 222 930 bp
Teilsequenz auf dem mensch-
lichen Chromosom 12, accession
no. U47924, gegen eine 227 538
bp lange Teilsequenz des Maus-
chromosoms 6.

Jeder Punkt des Plots entspricht
einem MUM von [ge]15 bp.



Delcher et al. Nucleic Acids Res 27, 2369 (1999)

Zusammenfassung

- Die Anwendung der Suffix-Bäume war ein Durchbruch für die Alignierung ganzer Genome
- MUMmer 2 besitzt zusätzliche Verbesserung für die Rechenzeit und den Speicherplatz
 - die Verwendung von Suffix-Arrays anstatt von Suffix-Bäumen gibt eine verbesserte Datenstruktur (→ Stefan Kurtz, Hamburg)
 - es wird nun möglich, mehr als zwei Genome zu alignieren (implementiert in MGA)