Sequenzvergleiche ohne Alignments durch lokales Dekodieren von Sequenzen

Christian Otto

(16. Juli 2007)

Inhalt

Alignments

- beruhend auf Editieroperation (Substitution, Deletion, Insertion), schlechte Einschätzung von Deletionen
- Löschen von mehrdeutigen Regionen bei multiplen Alignments
  $\longrightarrow$ Informationsverlust
- oft Vernachlässigung von Permutationen und Inversionen
- Verwendung von manuelles Alignment (Expert Editing) zum Berücksichtigen von mehrdeutige Regionen
  $\longrightarrow$ großer Aufwand, Problem der Reproduzierbarkeit

Idee der Sequenzvergleiche ohne Alignments:

- intuitiver Ansatz: Nukleotidhäufigkeiten vergleichen, da Sequenzen mit signifikant unterschiedlichen Nukleotidzusammensetzungen nicht nah verwandt sein können
  $\longrightarrow$ aber: nicht nah verwandte Sequenzen können gleiche Nukleotidhäufigkeiten besitzen

Idee der Sequenzvergleiche ohne Alignments:

- intuitiver Ansatz: Nukleotidhäufigkeiten vergleichen, da Sequenzen mit signifikant unterschiedlichen Nukleotidzusammensetzungen nicht nah verwandt sein können
  $\longrightarrow$ aber: nicht nah verwandte Sequenzen können gleiche Nukleotidhäufigkeiten besitzen
- nächster Schritt: N-Wörter (Teilsequenzen der Länge N) zwischen Sequenzen vergleichen, um daraus Ähnlichkeiten zu berechnen
  aber: N-Wörter entweder identisch oder unterschiedlich

Idee der Sequenzvergleiche ohne Alignments:

- intuitiver Ansatz: Nukleotidhäufigkeiten vergleichen, da Sequenzen mit signifikant unterschiedlichen Nukleotidzusammensetzungen nicht nah verwandt sein können
  $\longrightarrow$ aber: nicht nah verwandte Sequenzen können gleiche Nukleotidhäufigkeiten besitzen
- nächster Schritt: N-Wörter (Teilsequenzen der Länge N) zwischen Sequenzen vergleichen, um daraus Ähnlichkeiten zu berechnen
  aber: N-Wörter entweder identisch oder unterschiedlich
- Kompensierung durch Betrachtung von nicht perfekt identischen N-Wörtern (die Modelle und Algorithmen von diesem Vortrag berücksichtigen dies noch nicht)

Um evolutionäre Informationen aus mehrdeutigen Regionen zu gewinnnen und nicht auf manuelles Alignment angewiesen zu sein, wurde dann die Idee vom lokalen Dekodieren vom Grad N von Sequenzen entwickelt:

- beruhend auf Vergleichen zwischen überlappenden Wörtern der Länge N
- Übereinstimmungen von solchen Teilwörter zwischen verschiedenen Sequenzen entsprechen konzeptionell homologen Blöcken zwischen diesen Sequenzen
  $\longrightarrow$ Sequenzvergleiche und somit Erstellung eines Verwandtschaftsbaumes basierend auf dieser Idee

Definition der Relation $\overset{k}{\smile}_N$

Sei s eine Sequenz und $N \in \mathbb{N}, N > 0$.

Dann ist die Relation $(\overset{k}{\smile}_N)_{-1 \leq k < N}$ für alle Paare von Positionen (i, j) von s definiert durch:

- $i \overset{-1}{\smile}_N j \Leftrightarrow i = j$
- Für $0 \leq k < N$:
  $i \overset{k}{\smile}_N j \Leftrightarrow \exists l, 0 \leq l \leq k$: $s_{[i-l, i-l+N-1]} = s_{[j-l, j-l+N-1]}$.

Für alle Zahlen $-1 \leq k < N$ sei dann $\overset{k}{\smile}_N$ die transitive Hülle der Relation $\overset{k}{\smile}_N$.

Aus der Relation $\underset{\smile}{\overset{k}{—}}_N$ folgende Aussagen:

1. $i \underset{\smile}{\overset{k-1}{—}}_N j \Rightarrow i \underset{\smile}{\overset{k}{—}}_N j$

Aus der Relation $\underset{\smile}{\overset{k}{\frown}}_N$ folgende Aussagen:

1. $i \underset{\smile}{\overset{k-1}{\frown}}_N j \Rightarrow i \underset{\smile}{\overset{k}{\frown}}_N j$
2. $i \underset{\smile}{\overset{k-1}{\frown}}_N j \Rightarrow i+1 \underset{\smile}{\overset{k}{\frown}}_N j+1$

Aus der Relation $\underset{\smile}{\overset{k}{=}}_N$ folgende Aussagen:

1. $i \underset{\smile}{\overset{k-1}{=}}_N j \Rightarrow i \underset{\smile}{\overset{k}{=}}_N j$
2. $i \underset{\smile}{\overset{k-1}{=}}_N j \Rightarrow i+1 \underset{\smile}{\overset{k}{=}}_N j+1$
3. $i \underset{\smile}{\overset{k}{=}}_N j \Rightarrow s_{[i,i-k+N-1]} = s_{[j,j-k+N-1]}$

Aus der Relation $\overset{k}{\underset{\sim}{}}_N$ folgende Aussagen:

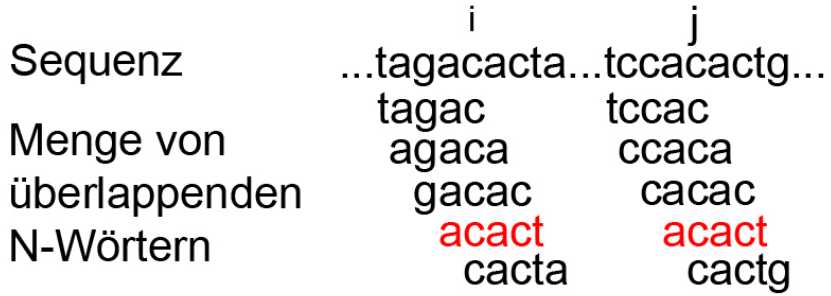
1. $i \overset{k-1}{\underset{\sim}{}}_N j \Rightarrow i \overset{k}{\underset{\sim}{}}_N j$
2. $i \overset{k-1}{\underset{\sim}{}}_N j \Rightarrow i+1 \overset{k}{\underset{\sim}{}}_N j+1$
3. $i \overset{k}{\underset{\sim}{}}_N j \Rightarrow s_{[i,i-k+N-1]} = s_{[j,j-k+N-1]}$

Für alle $-1 \leq k < N$ ist die Relation $\overset{k}{\underset{\sim}{}}_N$ eine Äquivalenzrelation (unter der Annahme, dass $|s| \geq N$ gilt).

Δ_k sei die Menge der zur Relation $\overset{k}{\underset{\sim}{}}_N$ gehörigen Äquivalenzklassen.

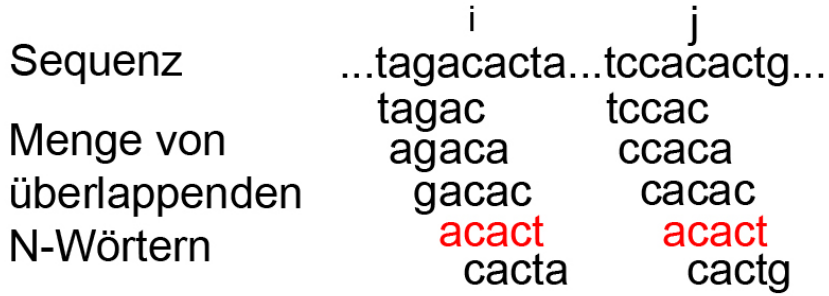
Verwandtschaft zwischen Sequenzpositionen

- Es gilt $i \overset{N-1}{\smile}_N j$, wenn ein überlappendes N-Wort über Position i und j übereinstimmt, wobei i und j in diesem Wort jeweils an der gleichen Position steht.
  Beispiel:



Verwandtschaft zwischen Sequenzpositionen

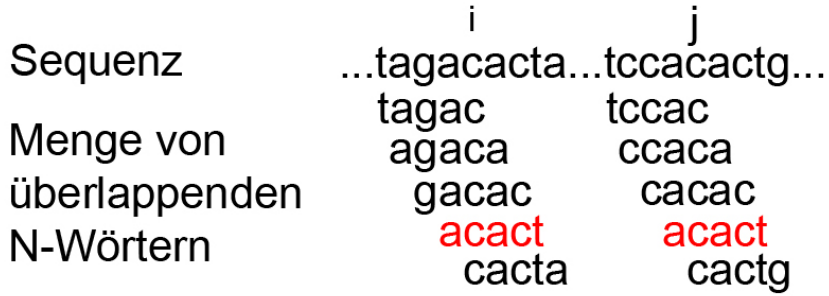
- Es gilt $i \overset{N-1}{\smile}_N j$, wenn ein überlappendes N-Wort über Position i und j übereinstimmt, wobei i und j in diesem Wort jeweils an der gleichen Position steht.
  Beispiel:



- Zwei Sequenzpositionen a, b sind genau dann verwandt, wenn $a \overset{N-1}{\smile}_N b$ gilt

Verwandtschaft zwischen Sequenzpositionen

- Es gilt $i \overset{N-1}{\smile}_N j$, wenn ein überlappendes N-Wort über Position i und j übereinstimmt, wobei i und j in diesem Wort jeweils an der gleichen Position steht.
  Beispiel:



- Zwei Sequenzpositionen a, b sind genau dann verwandt, wenn $a \overset{N-1}{\smile}_N b$ gilt
  d.h. es gibt Positionen $k_0, \ldots, k_n$ mit $k_0 = a$, $k_n = b$ und $\forall i \in \{0, \ldots, n-1\} : k_i \overset{N-1}{\smile}_N k_{i+1}$

Ermittlung der Äquivalenzklassen

a) Menge der N-Wörter jeder Sequenzposition ermitteln und miteinander vergleichen

$\Rightarrow$ Finden von Relationen $i \overset{N-1}{\smile}_N j$

Ermittlung der Äquivalenzklassen

a) Menge der N-Wörter jeder Sequenzposition ermitteln und miteinander vergleichen

   $\Rightarrow$ Finden von Relationen $i \overset{N-1}{\underset{\smile}{\frown}}_N j$

b) Transitiven Abschluss der Relation $\overset{N-1}{\underset{\smile}{\frown}}_N$ und Äquivalenzklassen bilden

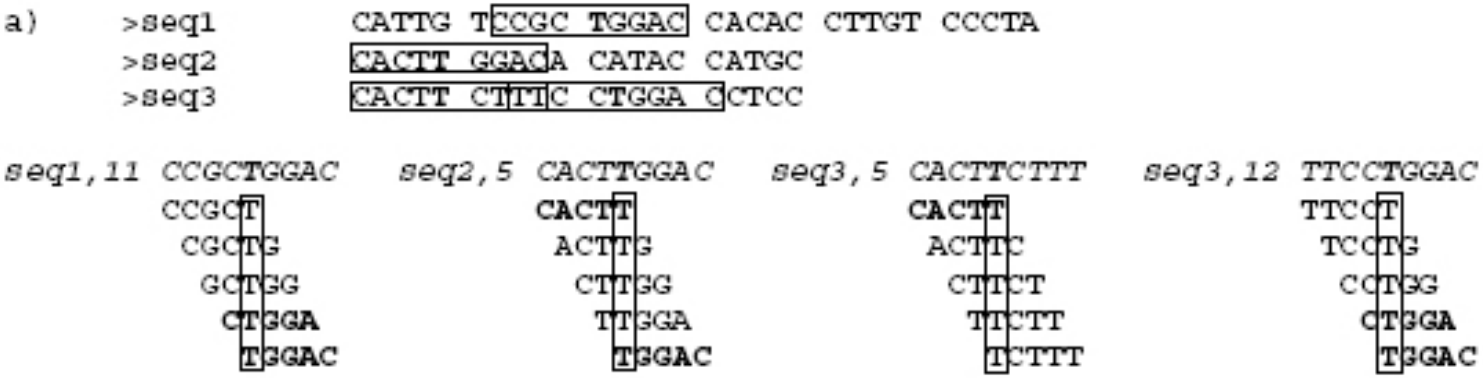
Ermittlung der Äquivalenzklassen

a) Menge der N-Wörter jeder Sequenzposition ermitteln und miteinander vergleichen

   $\Rightarrow$ Finden von Relationen $i \overset{N-1}{\smile}_N j$

b) Transitiven Abschluss der Relation $\overset{N-1}{\smile}_N$ und Äquivalenzklassen bilden

Beispiel:



b)

```
class_1 : T0
      seq1   3    CATTGTC
      seq1  22 CACCTTGTC
class_2 : T1
      seq1   4   CATTGTCC
      seq1  23 ACCTTGTCC
class_3 : G0
      seq1   5 CATTGTCCG
      seq1  24 CCTTGTCCC
class_4 : T2
      seq1   6 ATTGTCCGC
      seq1  25 CTTGTCCCT
class_5 : C0
      seq1   7 TTGTCCGCT
      seq1  26 TTGTCCCTA
class_6 : C1
      seq1   8 TGTCCGCTG
      seq1  27 TGTCCCTA
class_7 : C2
      seq1  10 TCCGCTGGA
      seq3  11 TTTCCTGGA
class_8 : T3
      seq1  11 CCGCTGGAC
      seq2   5 CACTTGGAC
      seq3   5 CACTTCTTT
      seq3  12 TTCCTGGAC
class_9 : G1
      seq1  12 CGCTGGACC
      seq2   6 ACTTGGACA
      seq3  13 TCCTGGACC
```

```
class_10 : G2
      seq1  13 GCTGGACCA
      seq2   7 CTTGGACAC
      seq3  14 CCTGGACCT
class_11 : A0
      seq1  14 CTGGACCAC
      seq2   8 TTGGACACA
      seq3  15 CTGGACCTC
class_12 : C3
      seq1  15 TGGACCACA
      seq2   9 TGGACACAT
      seq3  16 TGGACCTCC
class_13 : C4
      seq1  16 GGACCACAC
      seq3  17 GGACCTCC
class_14 : C5
      seq2   1     CACTT
      seq3   1     CACTT
class_15 : A1
      seq2   2    CACTTG
      seq3   2    CACTTC
class_16 : C6
      seq2   3   CACTTGG
      seq3   3   CACTTCT
class_17 : T4
      seq2   4  CACTTGGA
      seq3   4  CACTTCTT
```

Lokales N-Dekodieren

c) Äquivalenzklassen auf Zustände abbilden
(Positionen, die in einelementigen Klassen auftauchen, werden auf dem identischen Symbol abgebildet und nicht weiter betrachtet.)
$\Rightarrow$ Dies wird als lokales N-Dekodieren der Sequenz bezeichnet (analog zu Hidden Markov Modell).

Lokales N-Dekodieren

c) Äquivalenzklassen auf Zustände abbilden
(Positionen, die in einelementigen Klassen auftauchen, werden auf dem identischen Symbol abgebildet und nicht weiter betrachtet.)
$\Rightarrow$ Dies wird als lokales N-Dekodieren der Sequenz bezeichnet (analog zu Hidden Markov Modell).

Beispiel:

c) >seq1 C A $T_0T_1G_0T_2C_0C_1G$ $C_2T_3G_1G_2A_0C_3C_4A$ C A C C $T_0T_1G_0T_2C_0C_1C$ T A
>seq2 $C_5A_1C_6T_4T_3G_1G_2A_0C_3A$ C A T A C C A T G C
>seq3 $C_5A_1C_6T_4T_3C$ T T T C $C_2T_3G_1G_2A_0C_3C_4T$ C C

d) Erstellen einer Liste mit Anzahl Vorkommen jedes Zustandes (der einer Äquivalenzklasse entspricht) je Sequenz

d) Erstellen einer Liste mit Anzahl Vorkommen jedes Zustandes (der einer Äquivalenzklasse entspricht) je Sequenz

Beispiel:

d)	seq1	seq2	seq3
T3	1	1	2
T4	0	1	1
A0	1	1	1
A1	0	1	1
G1	1	1	1
G2	1	1	1
C2	1	0	1
C3	1	1	1
C4	1	0	1
C5	0	1	1
C6	0	1	1

(T0, T1, T2, G0, C0 and C1 are only repeated in seq1, two times each)

Berechnung der Distanzmatrix

Seien Zustände $x_1, \ldots, x_n$ mit $n \in \mathbb{N}$ gegeben.
Nun sei $|u|_{x_i}$ die Anzahl von Vorkommen des Zustandes x_i in der Sequenz u und $|u|$ die Länge der Sequenz u.

Berechnung der Distanzmatrix

Seien Zustände $x_1, \ldots, x_n$ mit $n \in \mathbb{N}$ gegeben.
Nun sei $|u|_{x_i}$ die Anzahl von Vorkommen des Zustandes x_i in der Sequenz u und $|u|$ die Länge der Sequenz u.

e) Berechnen der Ähnlichkeiten $\mathrm{sim(u, v)}$ zwischen Sequenzen u und v durch:

$$\mathrm{sim(u, v)} = \sum_{i=1}^{n} \min\left(|u|_{x_i}, |v|_{x_i}\right)$$

Berechnung der Distanzmatrix

Seien Zustände $x_1, \ldots, x_n$ mit $n \in \mathbb{N}$ gegeben.
Nun sei $|u|_{x_i}$ die Anzahl von Vorkommen des Zustandes x_i in der Sequenz u und $|u|$ die Länge der Sequenz u.

e) Berechnen der Ähnlichkeiten $\mathrm{sim(u, v)}$ zwischen Sequenzen u und v durch:

$$\mathrm{sim(u, v)} = \sum_{i=1}^{n} \min\left(|u|_{x_i}, |v|_{x_i}\right)$$

Normalisierung und Berechnung der Distanzen durch:

$$\mathrm{dist(u, v)} = 1 - \frac{\mathrm{sim(u, v)}}{\min\left(|u|, |v|\right)}$$

Beispiel:

d)

	seq1	seq2	seq3
T3	1	1	2
T4	0	1	1
A0	1	1	1
A1	0	1	1
G1	1	1	1
G2	1	1	1
C2	1	0	1
C3	1	1	1
C4	1	0	1
C5	0	1	1
C6	0	1	1

e) Similarities

	seq2	seq3
seq1	5	7
seq2		9

Distances

	seq2	seq3
seq1	0.75	0.65
seq2		0.55

Beispiel:

d)

	seq1	seq2	seq3
T3	1	1	2
T4	0	1	1
A0	1	1	1
A1	0	1	1
G1	1	1	1
G2	1	1	1
C2	1	0	1
C3	1	1	1
C4	1	0	1
C5	0	1	1
C6	0	1	1

e) Similarities

	seq2	seq3
seq1	5	7
seq2		9

Distances

	seq2	seq3
seq1	0.75	0.65
seq2		0.55

$\Rightarrow$ Erstellung eines Verwandtschaftsbaumes möglich (z.B. Neighbor-Joining-Tree)

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren
- Idee: Vergleich mit Distanzwerten von randomisierten Sequenzen $\Rightarrow$ Bootstrap

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren
- Idee: Vergleich mit Distanzwerten von randomisierten Sequenzen $\Rightarrow$ Bootstrap

Allgemeine Vorgehensweise bei Bootstrap

1. Erstellen von Bootstrapsamples aus einer Stichprobe (Resampling)

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren
- Idee: Vergleich mit Distanzwerten von randomisierten Sequenzen $\Rightarrow$ Bootstrap

Allgemeine Vorgehensweise bei Bootstrap

1. Erstellen von Bootstrapsamples aus einer Stichprobe (Resampling)
2. Berechnung einer bestimmten Größe aus dem Sample

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren
- Idee: Vergleich mit Distanzwerten von randomisierten Sequenzen $\Rightarrow$ Bootstrap

Allgemeine Vorgehensweise bei Bootstrap

1. Erstellen von Bootstrapsamples aus einer Stichprobe (Resampling)
2. Berechnung einer bestimmten Größe aus dem Sample
3. Iteration dieser Schritte
   $\Rightarrow$ Ergebnis: Verteilung der Größe aus den Samples

Bootstrap

- Distanzmaß ist zunächst nicht aussagekräftig bei neuem Verfahren
- Idee: Vergleich mit Distanzwerten von randomisierten Sequenzen $\Rightarrow$ Bootstrap

Allgemeine Vorgehensweise bei Bootstrap

1. Erstellen von Bootstrapsamples aus einer Stichprobe (Resampling)
2. Berechnung einer bestimmten Größe aus dem Sample
3. Iteration dieser Schritte
   $\Rightarrow$ Ergebnis: Verteilung der Größe aus den Samples

$\Rightarrow$ p-Wert Schätzung möglich

Anwendung des Bootstraps auf diesen Sachverhalt

Sei Distanz $\mathrm{dist}(\mathrm{u}, \mathrm{v})$ der Sequenzen u und v gegeben.

1. u und v resamplen (mit Zurücklegen aus der alten Sequenz ziehen und neue Sequenz gleicher Länge erstellen)
   $\Rightarrow$ neue Sequenzen u^*, v^*

Anwendung des Bootstraps auf diesen Sachverhalt

Sei Distanz $\mathrm{dist}(\mathrm{u}, \mathrm{v})$ der Sequenzen u und v gegeben.

1. u und v resamplen (mit Zurücklegen aus der alten Sequenz ziehen und neue Sequenz gleicher Länge erstellen)
   $\Rightarrow$ neue Sequenzen u^*, v^*
2. $\mathrm{dist}(\mathrm{u}^*, \mathrm{v}^*)$ über N-lokales Dekodieren berechnen

Anwendung des Bootstraps auf diesen Sachverhalt

Sei Distanz $\mathrm{dist}(\mathrm{u}, \mathrm{v})$ der Sequenzen u und v gegeben.

1. u und v resamplen (mit Zurücklegen aus der alten Sequenz ziehen und neue Sequenz gleicher Länge erstellen)
   $\Rightarrow$ neue Sequenzen u^*, v^*
2. $\mathrm{dist}(\mathrm{u}^*, \mathrm{v}^*)$ über N-lokales Dekodieren berechnen
3. mehrfache Iteration $\Rightarrow$ Verteilung von dist

Anwendung des Bootstraps auf diesen Sachverhalt

Sei Distanz $\mathrm{dist}(\mathrm{u}, \mathrm{v})$ der Sequenzen u und v gegeben.

1. u und v resamplen (mit Zurücklegen aus der alten Sequenz ziehen und neue Sequenz gleicher Länge erstellen)
   $\Rightarrow$ neue Sequenzen u^*, v^*
2. $\mathrm{dist}(\mathrm{u}^*, \mathrm{v}^*)$ über N-lokales Dekodieren berechnen
3. mehrfache Iteration $\Rightarrow$ Verteilung von dist

Wenn $dist(u, v)$ nun extrem in der Verteilung liegt
(z.B. in den 2.5% niedrigsten oder höchsten Werten),
dann deutet dieser Distanzwert auf keinen Zufallsbefund hin
sondern besitzt Aussagekraft (z.B. starke Verwandtschaft).

Naiver Algorithmus ist von hoher Komplexität in CPU-Zeit.
Problem ist die Berechnung der Äquivalenzklassen Δ_{N-1}.
$\Rightarrow$ Verbesserung auf lineare Komplexität in Zeit und Speicher möglich

Naiver Algorithmus ist von hoher Komplexität in CPU-Zeit.
Problem ist die Berechnung der Äquivalenzklassen Δ_{N-1}.
$\Rightarrow$ Verbesserung auf lineare Komplexität in Zeit und Speicher möglich
Dazu weitere Definitionen:

- $\nabla_k \subset \Delta_k$ ist definiert durch:

  $\delta \in \nabla_k \Leftrightarrow \delta \in \Delta_k, \exists$ i,j $\in \delta$ mit $(i+1) \overset{k}{\not\sim}_N (j+1)$

Naiver Algorithmus ist von hoher Komplexität in CPU-Zeit.
Problem ist die Berechnung der Äquivalenzklassen Δ_{N-1}.
$\Rightarrow$ Verbesserung auf lineare Komplexität in Zeit und Speicher möglich
Dazu weitere Definitionen:

- $\nabla_k \subset \Delta_k$ ist definiert durch:

  $\delta \in \nabla_k \Leftrightarrow \delta \in \Delta_k, \exists$ i,j $\in \delta$ mit $(i+1) \overset{k}{\not\sim}_N (j+1)$
- Die Abbildung F_k:$\Delta_k \rightarrow \Delta_{k+1}$ ist definiert durch:
  $F_k(\gamma) = \delta \Leftrightarrow \gamma \in \Delta_k, \delta \in \Delta_{k+1}, \forall i \in \gamma : i+1 \in \delta$
  Man nennt δ dann k-Nachfolger von γ.

Naiver Algorithmus ist von hoher Komplexität in CPU-Zeit.
Problem ist die Berechnung der Äquivalenzklassen Δ_{N-1}.
$\Rightarrow$ Verbesserung auf lineare Komplexität in Zeit und Speicher möglich
Dazu weitere Definitionen:

- $\nabla_k \subset \Delta_k$ ist definiert durch:
  $\delta \in \nabla_k \Leftrightarrow \delta \in \Delta_k, \exists$ i,j $\in \delta$ mit $(i+1) \overset{k}{\not\sim}_N (j+1)$
- Die Abbildung F_k:$\Delta_k \rightarrow \Delta_{k+1}$ ist definiert durch:
  $F_k(\gamma) = \delta \Leftrightarrow \gamma \in \Delta_k, \delta \in \Delta_{k+1}, \forall i \in \gamma : i+1 \in \delta$
  Man nennt δ dann k-Nachfolger von γ.

Funktionsweise des verbesserten Algorithmus

- Ermittlung von Δ_{-1}, Δ_0, ∇_0 und F_{-1}
- Iterative Berechnung von Δ_k, ∇_k und F_{k-1} aus Δ_{k-1}, ∇_{k-1} und F_{k-2} für $0 < k < N-1$
  $\Rightarrow \Delta_{N-1}$

Ermittlung von Δ_0, ∇_0 und F_{-1}

Gegeben sei die Sequenz s.

Beispiel: (N=2)

positions :	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s =	a	g	a	c	c	a	a	g	c	a	a	g	c	c

Ermittlung von Δ_0, ∇_0 und F_{-1}

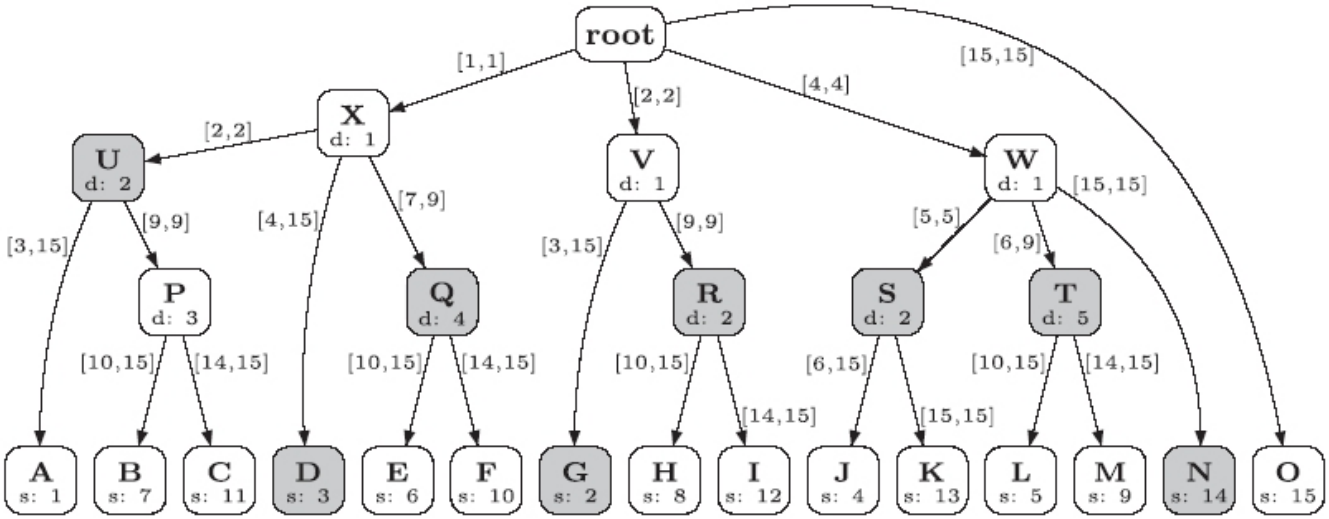
Gegeben sei die Sequenz s.

Beispiel: (N=2)

positions :	1	2	3	4	5	6	7	8	9	10	11	12	13	14
s =	a	g	a	c	c	a	a	g	c	a	a	g	c	c

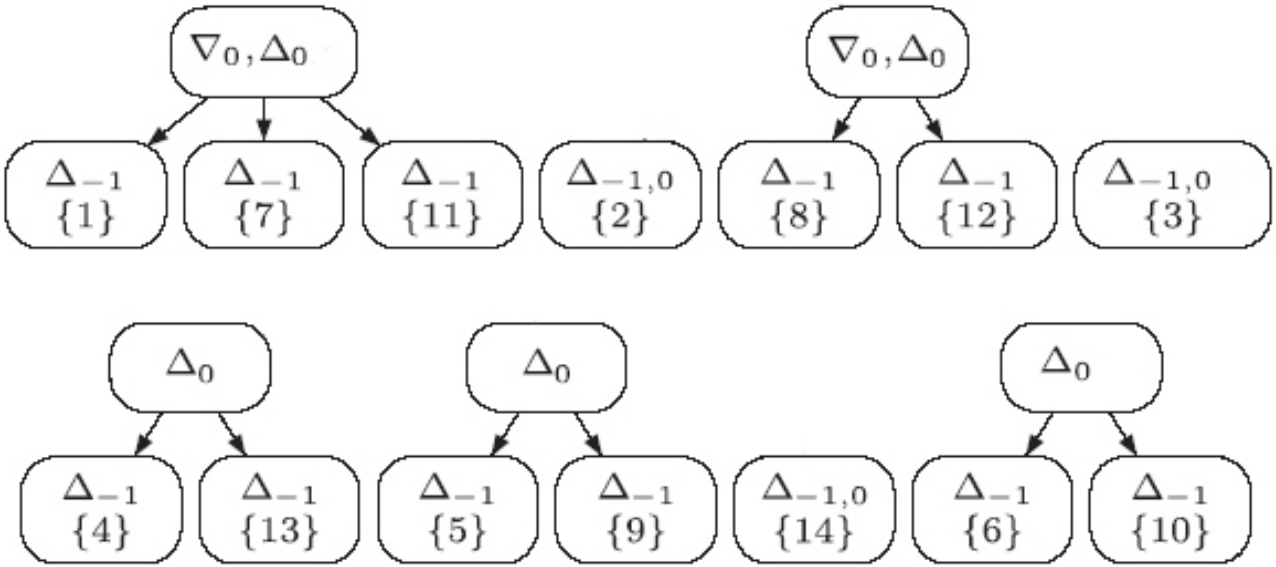
1. Erstellung des Suffixbaumes aus der Sequenz s

Beispiel:



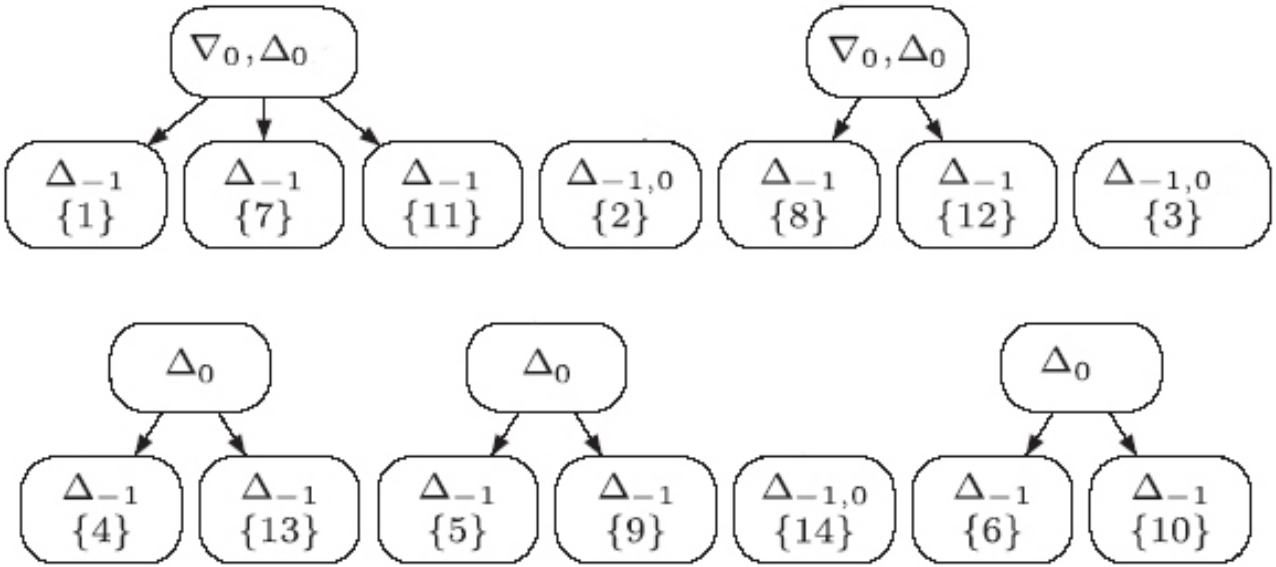
2. Ablesen von Δ_0 und ∇_0

Beispiel:



2. Ablesen von Δ_0 und ∇_0

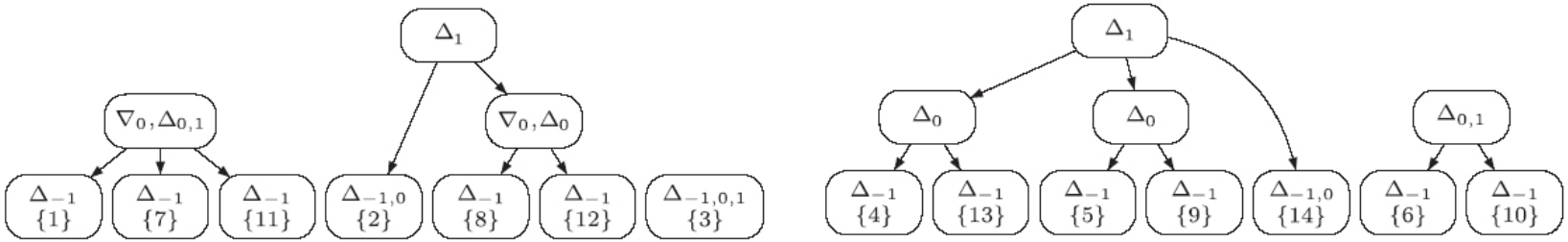
Beispiel:



3. Erstellen der Funktion F_{-1}

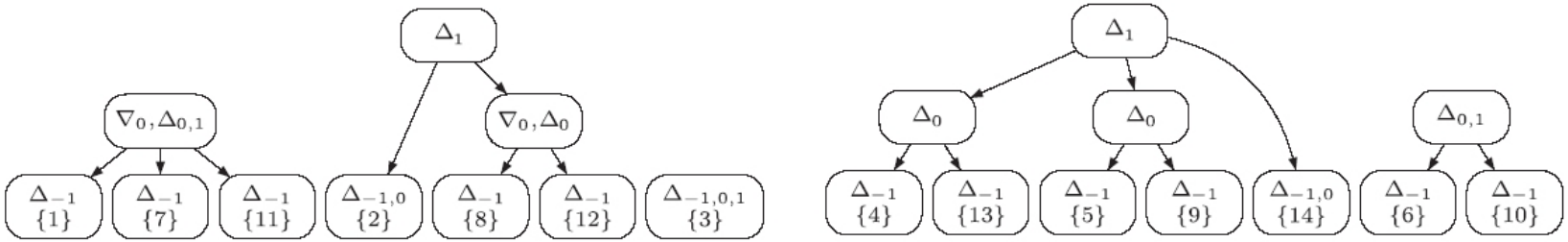
4. Iteration (detailierter siehe Handout):
   Für jedes Element aus ∇_{k-1} wird ein neues Element in ∇_k erstellt, falls kein Nachfolger gesetzt ist.
   Kinderelemente und Nachfolgerverweise werden dann entsprechend gesetzt.

Beispiel:



4. Iteration (detailierter siehe Handout):
   Für jedes Element aus ∇_{k-1} wird ein neues Element in ∇_k erstellt, falls kein Nachfolger gesetzt ist.
   Kinderelemente und Nachfolgerverweise werden dann entsprechend gesetzt.

Beispiel:



Bemerkungen zum Algorithmus:

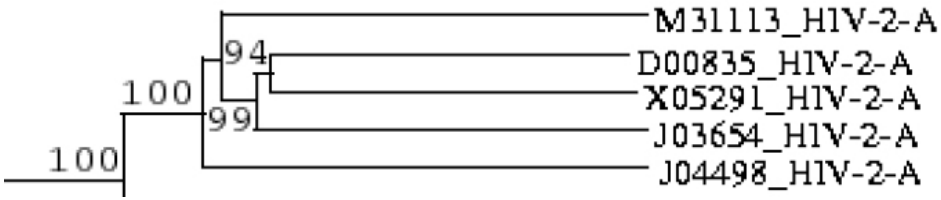
- beruhend auf einigen theoretischen Sätzen zu Δ, ∇ und F [2]
- Anwendung auf multiple Alignments durch Aneinanderketten der Sequenzen zu einer Sequenz mit einzigartigen voneinander unterschiedlichen Trennsymbolen zwischen den einzelnen Sequenzen

Optionen zur Erweiterung

- Möglichkeit auch nicht perfekte Übereinstimmungen zu berücksichtigen
- Inversionen berücksichtigen:
  Invertierte Strings in Menge von überlappenden N-Wörtern übernehmen
- weitere Möglichkeiten wären denkbar
  (ggf. Permutationen betrachten),
  aber: Rechenzeit wird dadurch erhöht

- Anwendung des N-lokales Dekodierens auf 70 HIV/SIV Genomen (davon 4 unvollständige)
- sehr gute Übereinstimmung der Topologie des resultierenden Baumes mit existierenden Wissen
- Baum beinhaltet bekannte evolutionär signifikanten HIV/SIV Ereignisse
- genaue Topologie abhängig von Wahl von N
  identische Topologie für N=13-29 mit HIV/SIV Sequenz Compendium 2000 [3]

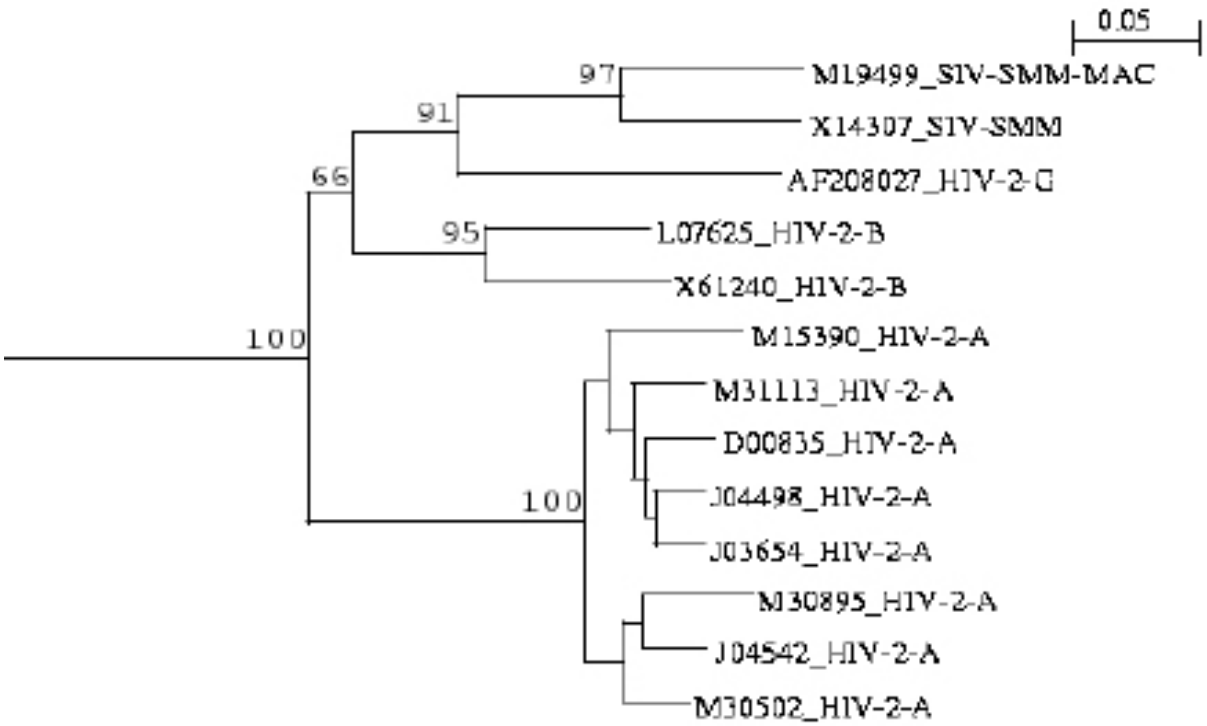
Ausschnitt: (mit N=15)



- Gene *gag*, *pol* und *env* codieren Informationen über die strukturellen Proteine für neue Viruspartikel,
  nef codiert Protein für effiziente Vermehrung des Virus
- Gensequenzen wurden aus 66 HIV/SIV Genomen verwendet
  $\rightarrow$ Distanzmatrizen des N-lokales Dekodieren von *gag*, *pol* und *env* stimmt gut mit etablierten HIV/SIV Phylogeniebäumen für diese Regionen überein
- Einige Unterschiede bei *nef*, wobei diese möglicherweise darauf zurückzuführen sind, dass beim N-lokalen Dekodieren mehrdeutige Regionen berücksichtigt werden (im Gegensatz zu multiplen Alignments)
- Empirisch gute Werte für N in den Regionen:
  gag - N=11-23; *pol* - N=11-30; *env* - N=12-24 und *nef* - N=11-20

- LTR ist RNA-Sequenz am Ende von jedem HIV/SIV Strang, dienen als Schalter zur Kontrolle neuer Viren
- Sequenzvergleich von 43 non-coding LTR-Regionen (enthalten viele Duplikationen/Insertionen/Deletionen)
  $\longrightarrow$ mehrdeutige Regionen bei multiplen Alignment
- keine Referenzbäume vorhanden
  $\longrightarrow$ Versuch der Erzeugung mit Hilfe von CLUSTAL-W, DIALIGN-2
- Abgleich der gewonnenen Bäume mit umfangreichen Wissen über HIV/SIV-Subtypen:
  CLUSTAL-W führte schlechte Trennung zwischen HIV-1, HIV-2 und SIV-Sequenzen durch, DIALIGN trennte Subtypen von HIV-1 nicht gut auf (z.B. HIV-1 N ist innerhalb von HIV-1 M)
  $\longrightarrow$ erwartungsgemäß schlechte Performance von Alignment-basierten Methoden

- Gute Trennung der HIV/SIV Subtypen durch N-lokales Dekodieren, bei N=10-21
- Problem: fehlende Referenzbäume zur genauen Einschätzung des erstellten Baumes



- Erweiterung auf komplexeres evolutionäres Modell wäre sinnvoll
- Wahl von Parameter N entscheidend für sinnvolle Ergebnisse:
  N=13-20 sind gute empirische Standardwerte,
  $\longrightarrow$ weitere Untersuchungen notwendig
- Berücksichtigung auch von mehrdeutigen Regionen (z.B. LTR),
  aber: Problem der Validierung
- Implementation des verbesserten Algorithmus verfügbar [4]

Literatur


Didier G, Debomy L, Pupin M, Laprevotte I,
Comparing sequences without using alignments: application to HIV/SIV subtyping, BMC Bioinformatics. 2007 Jan 02


Didier G, Laprevotte I, Pupin M, Hénaut A,
Local decoding of sequences and alignment-free comparison,
J Comp Biol. 2006, 13: 1465-1476


The 2000 HIV Sequence Compendium,
http://www.hiv.lanl.gov/content/hiv-db/COMPENDIUM/2000/HIV12SIVcomplete.pdf


SCeNE - Prototypimplementierung des N-lokalen Dekodierens,
http://iml.univ-mrs.fr/∼didier/laprevot/