Effiziente Methoden Für Die Berechnung Von Aminosäure Ersetzungsraten

Tobias Gontermann
Johann Wolfgang Goethe-Universität Frankfurt a. M.
12 Juli 2007

Übersicht

Worum geht es ?

- Berechung von Ratenmatritzen Q

Wofür ist das gut?

- Modellierung von Protein-Evolution mit Markovketten
- Hat man die Ratenmatrix Q gegeben, kann man die Übergangsmatritzen P(t) für jede beliebige Zeitspanne t berechnen

Definition: Markov-Eigenschaft

$$Pr[\ X(t)=i \mid X(0) = x_0,\ X(1) = x_1,\ \ldots,\ X_{(t-1)} = j\ ] \quad = \quad Pr[\ X(t) = i \mid x_{t-1} = j\ ]$$

Die Markov-Eigenschaft besagt, dass die Zufallsvariable X_t nur von der Vorangegangenen Zufallsvariable X_{t-1} abhängt.

Struktur der Probability Matrix P(t)

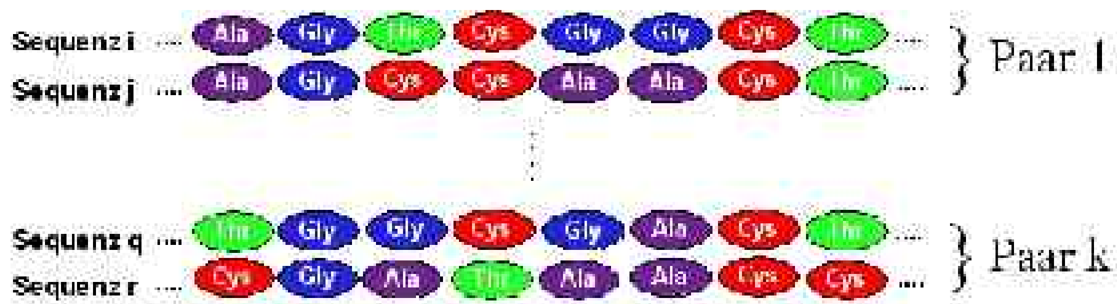
- $P(0) = I$
- $p_{ij}(t) \geq 0$ und $\sum_j q_{ij}(t) = 1$
- $P(s + t) = P(s)P(t)$ für $s, t \geq 0$ Chapman-Kolmogorov-Gleichung

Struktur der Ratenmatrix Q

- $q_{ij} \geq 0$ für alle $i \neq j$
- $q_{ii} \leq 0$ für alle i (Diagonaleinträge Negativ)
- $\sum_j q_{ij} = 0$ für alle i (Summe der Zeilen ist 0)

Eingabe:

- k alignierte Paare von Aminosäuren



- Diese erzeugen k Count-Matritzen $N_1, ..., N_k$
  Deren Einträge bestehen aus den absoluten Ersetzungshäufigkeiten einer Aminosäure durch eine Andere

- Die Markov-Kette beschreibt die Evolution an einer einzelnen Stelle in einem Protein von der Vorgängersequenz zur Nachfahrensequenz

- Da keine direkten Sequenzinformationen von der Vorgängersequenz zur Verfügung stehen, beobachtet man Aminosäurepaare, die einen gemeinsamen Vorgänger besitzen

- Um dennoch Aussagen über den Evolutionsprozess ableiten zu können, nimmt man an, dass die Evolution von der Vorgänger -zur Nachfahrensequenz durch den gleichen Prozess wie der umgekehrt verlaufende beschrieben werden kann

- Die Zeit die zwischen zwei Sequenzen i und j liegt, bezeichnet man als Divergenz und misst sie in PAMs.

- **Definition**: PAM (Percent Accepted Mutation)

  PAM ist eine eingeführte Einheit, um die Anzahl von evolutiven Veränderungen in einer Proteinsequenz zu messen. Die 1.0 PAM Einheit ist die Menge von Veränderungen, die durchschnittlich 1% der Aminosäuren einer Proteinsequenz verändert.

Wichtige Annahmen:

- Die Aminosäuren sind voneinander Unabhängig
- Es gibt eine Stationäre Verteilung π, unabhängig, mit welcher Aminosäure der Markov-Prozess gestartet wurde
- Die Ersetzungen folgen wie bereits erwähnt einem Zeit-Umkehrbaren Markovprozess mit einer 20 x 20 (Ratenmatrix) Intensitätsmatrix Q

Reversibilität

- Sei π_i die Startverteilung von Aminosäure i und Q_{ij} die Wahrscheinlichkeit, dass Aminosäure i durch Aminosäure j ersetzt wird, dann schreibt man formal

  Für alle i,j : $\pi_i Q_{ij} = \pi_j Q_{ji}$

Reversibilität

- Zur Vereinfachung schreibt man nun alle π_i in die Diagonale einer Matrix Π

- Dadurch können wir

$\pi_i Q_{ij} = \pi_j Q_{ji}$ Schreiben als $\Pi Q = Q^T \Pi$

Übersicht

Beziehung zwischen P und Q

- Es gilt : $P(t) = e^{Qt}$

- Dabei ist e^X die Exponentialfunktion für Matritzen

$$e^X = \sum_{i<0} \frac{X^i}{i!}$$

- Aus dieser Gleichung kann man sich drei Aussagen herleiten:

  1.) $\Pi P(t)$ ist Symmetrisch
  2.) $P(t)$ und Q haben die gleichen Eigenvektoren
  3.) λ ist Eigenw. von $Q \Rightarrow e^{\lambda t}$ ist Eigenw. von $P(t)$

Definition: Eigenwert und Eigenvektor

Sei Q eine n x n Matrix, dann nennt man eine reelle Zahl λ Eigenwert zu Q, falls ein Vektor $V \in R^n$, $V \neq 0$ existiert mit $QV = \lambda V$.
V heißt dann Eigenvektor (zum Eigenwert λ).

Definition: Eigenwertzerlegung

Enthält eine Matrix Λ die Eigenwerte von Q in ihrer Diagonalen, und die korrespondierenden Eigenvektoren werden die Spaltenvektoren einer Matrix V_Q, dann schreibt man:

$$QV_Q = V_Q\Lambda$$

wenn V_Q eine invertierbare Matrix ist, erhält man die Eigenwertzerlegung

$$\boxed{Q = V_Q \Lambda V_Q^{-1}}$$

Nutzen der Eigenvektoren

- Da P(t) und Q die gleichen Eigenvektoren haben, gilt folgendes:

$$P(t) = V_Q \, e^{\lambda t} \, V_Q^{-1}$$

- P(t) hat jedoch andere Eigenwerte

Berechnung der Eigenvektoren

- Das Einzige was wir gegeben haben sind die Count-Matritzen N_i und folgender Zusammenhang

$$L_i \Pi P(t_i) = N_i$$

Dabei ist L_i die Länge der Sequenz i

- Wir wissen weiterhin, dass $L_i P(t_i)$ die gleichen Eigenvektoren wie $P(t_i)$ und Q hat

Berechnung der Eigenvektoren

- Durch Normierung der Zeilen von $L_i \Pi P(t_i) = N_i$ zu 1 verändern sich die Eigenvektoren nicht, aber die Faktoren L_i und Π gehen verloren.
- Man erhält also eine Matrix F_i als Approximation von $P(t_i)$
- Um die Symmetrie von $\Pi P(t_i)$ zu gewährleisten, Berechnet man F_i durch $N_i + N_i^T$

Berechnung der Eigenvektoren

- Es gilt weiterhin, dass wenn v ein Eigenvektor von P(t) ist, dann ist er auch ein Eigenvektor von $\sum_i P(t_i)$
- Diesen Zusammenhang nutz man, um eine Art Durchschnittsmatrix S zu konstruieren, aus der man dann die Eigenvektoren V_Q bestimmen kann

$$S = \frac{1}{k} \sum_{k}^{i=1} F_i$$

Auf diesen Grundlagen basieren die folgenden drei Methoden zur Berechnung der Intensitätsmatrix Q

1.) Bayesian Weights (BW)
2.) Müller and Vingron's Resolvent Methode (MVR)
3.) Bayesian Weighting for the Resolvent (BR)

BW-Methode

Übersicht

Idee der Methode

1.) Bestimme $S = \frac{1}{k} \sum_{k}^{i=1} F_i$

2.) Berechnen die Eigenvektoren von S und Speichere sie in V

3.) Nehme eine beliebige Startmatrix Q

4.) Für alle $\tau \in T = \{5, 10, 15, 20, ..., 250\}$
- Approximiert mit Hilfe von Bayesian Weights $P(\tau)$
- Man berechnet $D_\tau := V^{-1} P(\tau) V$

5.) Man bestimmt die Eigenwerte von D_τ und Speichert sie in der Diagonalen von Λ

6.) Man berechnet $Q' := V \Lambda V^{-1}$

7.) Wenn $|Q' – Q| > \varepsilon$ Dann $Q:=Q'$ Gehe zu 4

Bestimmen der Eigenwerte

- Mann nimmt nun an, dass man eine Divergenz τ gegeben hat und $P(\tau)$ bestimmen möchte

- Dafür lässt man alle Sequenzpaare ihre Informationen mit einer gewissen Gewichtung in die Berechnung einfließen

- Dieses Gewicht ist von der a-posteriori Wahrscheinlichkeit abhängig, dass für das entsprechende Sequenzpaar die Divergenz τ vorliegt

- Mit Hilfe dieser Gewicht erhält man folgende Approximation für $P(\tau)$

$$\widetilde{P}_\tau = \frac{\sum_{i=1}^{k} F_i w_Q(N_i, \tau)}{\sum_{i=1}^{k} w_Q(N_i, \tau)}$$

dabei sind $W_Q(N_i,\tau)$ die Gewichte und $Fi \approx N_i$

Bestimmen der Eigenwerte

- Man nimmt an, dass eine a-Priori-Verteilung durch die Wahrscheinlichkeitsdichte $f_{t_i}(\tau)$ gegeben ist, und ein geeignetes Startmodel von Q vorliegt.
- Damit erhält man nach Bayes

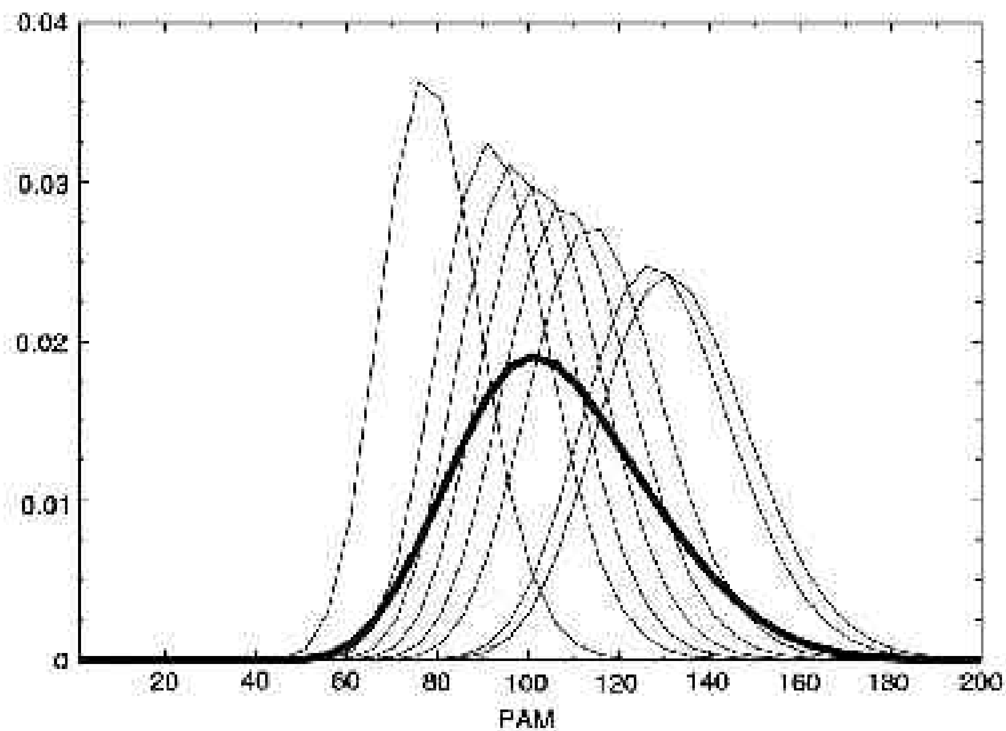
$$\underbrace{f_{t_i|N_i,Q}(\tau)}_{\text{Divergenzverteilung}} = \frac{Pr(N_i|t_i = \tau, Q) f_{t_i}(\tau)}{Pr(N_i|Q)}$$

- Wählt man eine Gleichverteilung von t_i erhält man für die Gewichte

$$\underbrace{w_Q(N_i, \tau)}_{\text{Gewichtung für } N_i \text{ mit Divergenz } \tau} = \frac{Pr(N_i|t_i = \tau, Q)}{Pr(N_i|Q)}$$

Bestimmen der Eigenwerte

- Für 10 Alignments mit 100 Spalten und einer Divergenz von 100 PAM berechnet man ihre a-posteriori-Wahrscheinlichkeiten und zeichnet sie in den Graph (Dünnen Linien)



Bestimmen der Eigenwerte

- Wir haben nun folgende Gleichungen gegeben

$$w_Q(N_i, \tau) = \frac{Pr(N_i | t_i = \tau, Q)}{Pr(N_i | Q)}$$

$$\widetilde{P}_\tau = \frac{\sum_{i=1}^{k} F_i w_Q(N_i, \tau)}{\sum_{i=1}^{k} w_Q(N_i, \tau)}$$

- Damit können wir die Eigenwerte bestimmen

Bestimmen der Eigenwerte

- Wir wenden unsere Gleichung für die Eigenvektorzerlegung

$$P(t) = V_Q\, e^{\lambda t}\, V_Q^{-1}$$

auf das Approximierte $\tilde{P}$ an und erhalten durch Umformen

$$D(\tau) = V_Q^{-1}\, \tilde{P}(\tau)\, V_Q$$

dabei ist $D(\tau)$ eine Diagonalmatrix mit Elementen $d_{\tau,j}$ der Form $e^{\Lambda_j \tau}$ (Eigenwerte)

- Da τ nun als bekannt gilt, kann man die λ_j aus den Diagonalelementen berechnen und Λ erstellen

Nocheinmal die Idee der Methode

✓ 1.) Bestimme $S = \frac{1}{k} \sum_{k}^{i=1} F_i$

✓ 2.) Berechnen die Eigenvektoren von S und Speichere sie in V

✓ 3.) Nehme eine beliebige Startmatrix Q

4.) Für alle $\tau \in T = \{5, 10, 15, 20, ..., 250\}$

- Approximiert mit Hilfe von Bayesian Weights $P(\tau)$
- Man berechnet $D_\tau := V^{-1} P(\tau) V$

5.) Man bestimmt die Eigenwerte $\ln(d_{\tau,j})/\tau$ mit D_τ und Speichert sie in der Diagonalen von Λ

6.) Man berechnet $Q' := V \Lambda V^{-1}$

7.) Wenn $|Q' - Q| > \varepsilon$ Dann $Q:=Q'$ Gehe zu 4

Bestimmen der Eigenwerte

- Durch die Iteration wird Q in jedem Schritt verfeinert

- Man sollte jedoch nicht alle Divergenzen $\tau \in T$ betrachten, da es zu Störungen kommen kann, wenn zu wenig Sequenzen mit dieser Divergenz zur Verfügung stehen.

- Man verwendet nur solche $\tau \in T$, für die die folgende Formel einen bestimmten Wert überschreitet

$$w_Q(\tau) = \sum_{j=1}^{k} w_Q(N_i, \tau)$$

Dabei ist liegt der Schwellenwert z.B. fest bei $w_Q(\tau) \geq 0{,}05$

Bestimmen der Eigenwerte

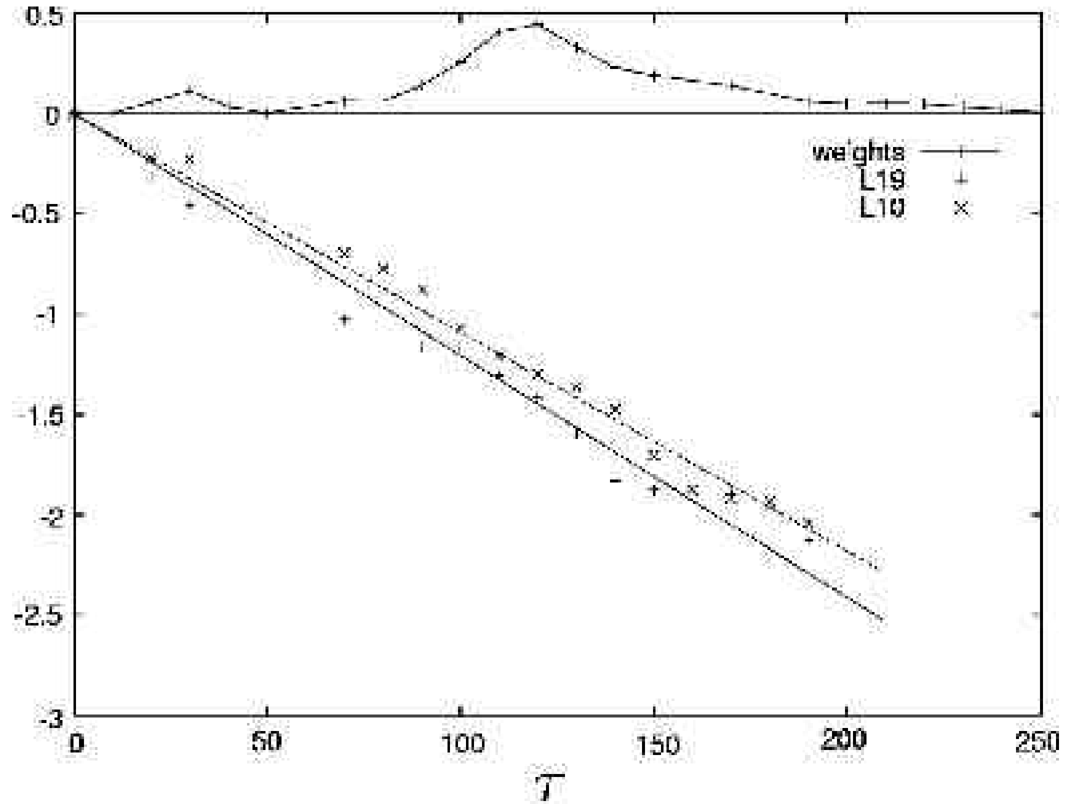
- Ein nützlicher Effekt des Gewichtungs-Schemas ist, dass große Divergenzen, die eher zu Störungen führen, weniger gewichtet werden, als kleinere Divergenzen

- Es kann jedoch vorkommen, dass man für kleine Datenmengen keine einwandfreie Matrix Q' erhält, weil einige Einträge negativ sind

- In diesem Fall weist man einem solchen Eintrag den kleinsten positiven Wert aus Q' zu und berechnet die Diagonalelemente von Q' neu

Bestimmen der Eigenwerte

- Die Folgende Grafik zeigt die Berechnung der Eigenwerte λ_{19} und λ_{20} in der ersten Iteration



Idee der Methode

- Man verwendet die Count-Matritzen N_i um P(t) zu berechnen
- Dann berechnet aus P(t) die Resolvent-Matrix R_α
- Um über folgende Gleichung Q zu bestimmen

$$\boxed{Q = aI - R_\alpha^{-1}}$$

Die Resolvent Matrix

- Die Resolvent-Matrix R_α berechnet sich für für $\alpha > 0$ wie folgt

$$\boxed{R_\alpha = \int_0^\infty e^{-\alpha t} P(t)\, dt} = \int_0^{t_1} e^{-\alpha t} P(t)\, dt + \ldots + R_\alpha = \int_{t_n}^\infty e^{-\alpha t} P(t)\, dt$$

- Die einzelnen Summanden werden wie folgt berechnet.

$$\int_{t_k}^{t_{k+1}} e^{-\alpha t} \left(\tilde{P}(t_k) + \frac{\tilde{P}(t_{k+1}) - \tilde{P}(t_k)}{t_{k+1} - t_k} (t - t_k) \right) dt$$

- Durch die Integrierung kompensiert man einzelne unzureichende Count-Matritzen und erhält eine gute Approximation

Übersicht

Die Resolvent Matrix

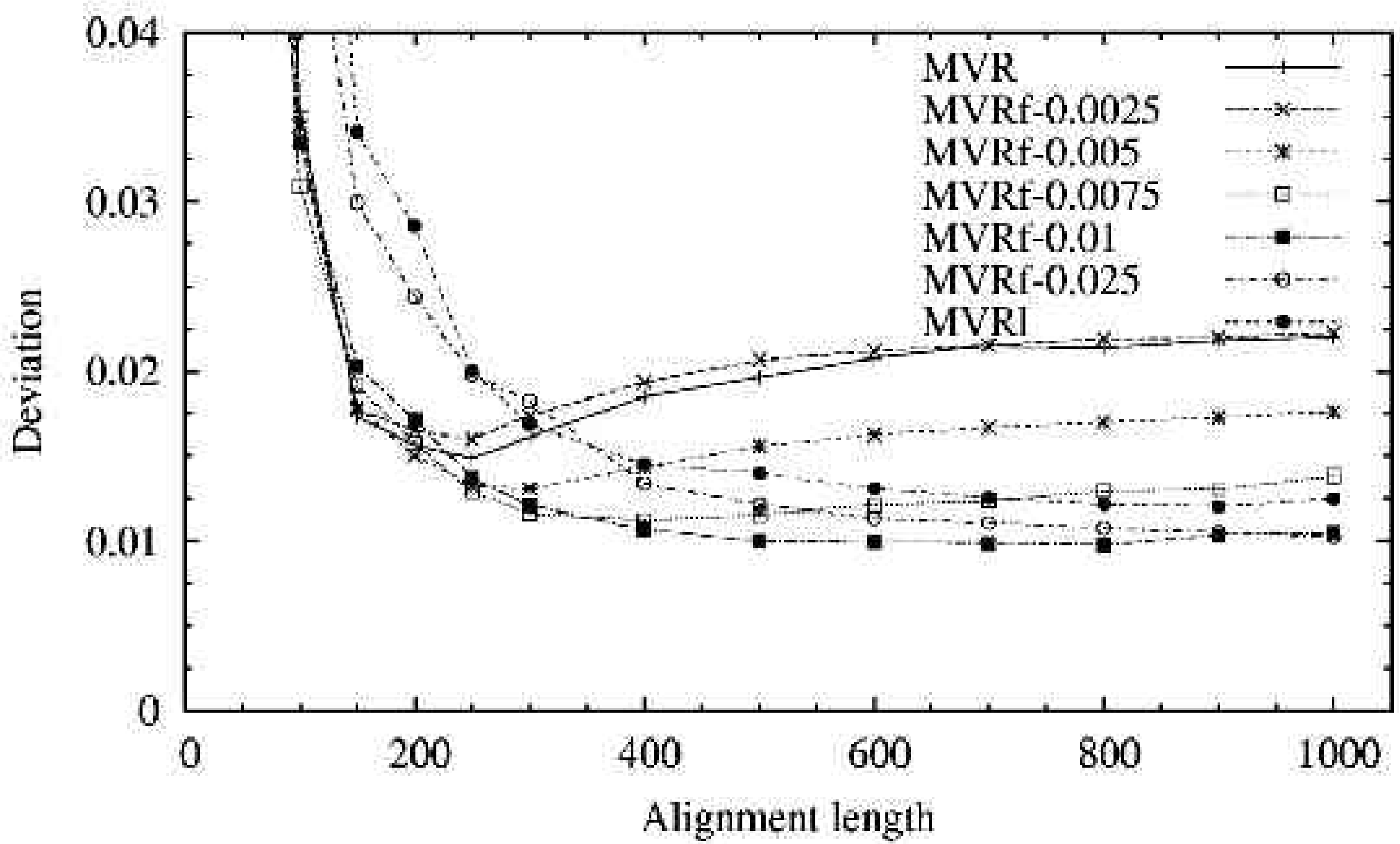
$$R_\alpha = \int_0^\infty e^{\alpha t} P(t)\, dt$$

- Die P(t) werden über die bekannten Frequenzmatritzen F_i approximiert

- Wie wir später noch sehen werden, ist die richtige Wahl von α eine entscheidende Fehlerquelle für die erfolgreiche Bestimmung von Q

Die Richtige Wahl von α

- Beispiel zeigt den Einfluss von α auf das Ergebnis



- Das wird bei MVR α durch Maximum Likelihood bestimmt

Die Idee der Methode

- Wir haben gesehen, dass die Berechnung der Resolvent R_{α} über Integration einer Funktion mit P(t) über t erfolgt.
- Man geht davon aus, dass diese Integration wegen der ungenauen Approximation der P(t) eine weitere Fehlerquelle ist
- Wir haben aber in der BW-Methode eine Möglichkeit gezeigt, um die P(t) mit der Bayesischen Gewichtung gut zu approximieren. Dieses Wissen nutzt man nun aus.

Die Idee der BR-Methode

- BR stellt nun eine Mischform von MVR und BW dar

- Man verwendet den MVR Algorithmus, benutz aber Anstelle der durch die Frequenzmatritzen F_i approximierten P(t), die durch die Subroutine der BW-Methode approximierten P(t)

$$\widetilde{P}_\tau = \frac{\sum_{i=1}^{k} F_i w_Q(N_i, \tau)}{\sum_{i=1}^{k} w_Q(N_i, \tau)}$$

- Das α wird wie bei MVR durch Maximierung der Likelihood bestimmt.

Es wurden nun Tests durchgeführt, um die drei Methoden miteinander zu vergleichen

Test auf synthetischen Daten

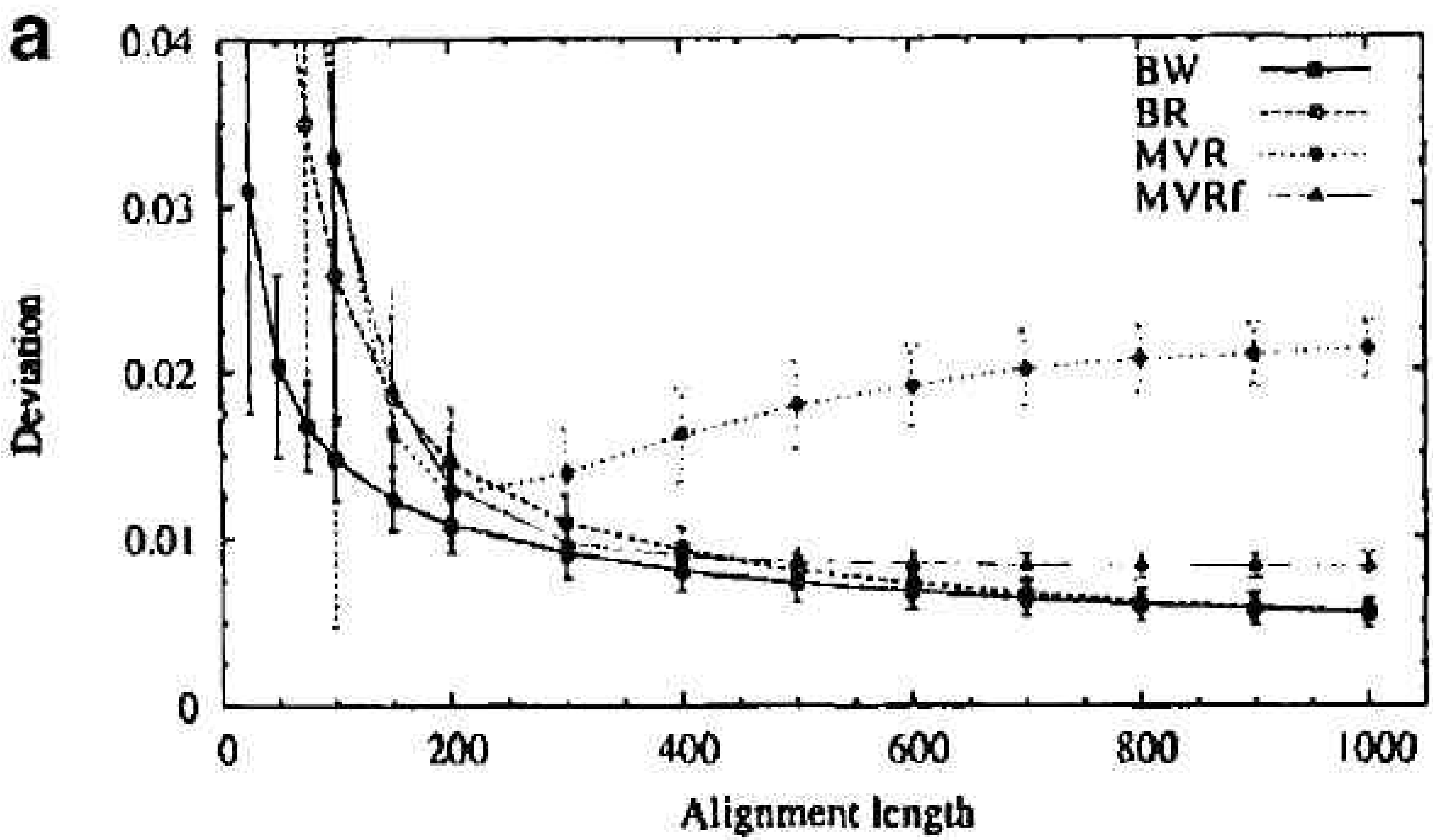
- 30 Alignments der Länge L erzeugt, die Methoden angewendet und jeden Test 1000 mal wiederholt

- Man hat hier zwei Varianten unterschieden:

  a.) Gleichmäßige Verteilung gemäß 10j PAMs für Alignment j
  b.) Normalverteilung im Bereich N(150,90)

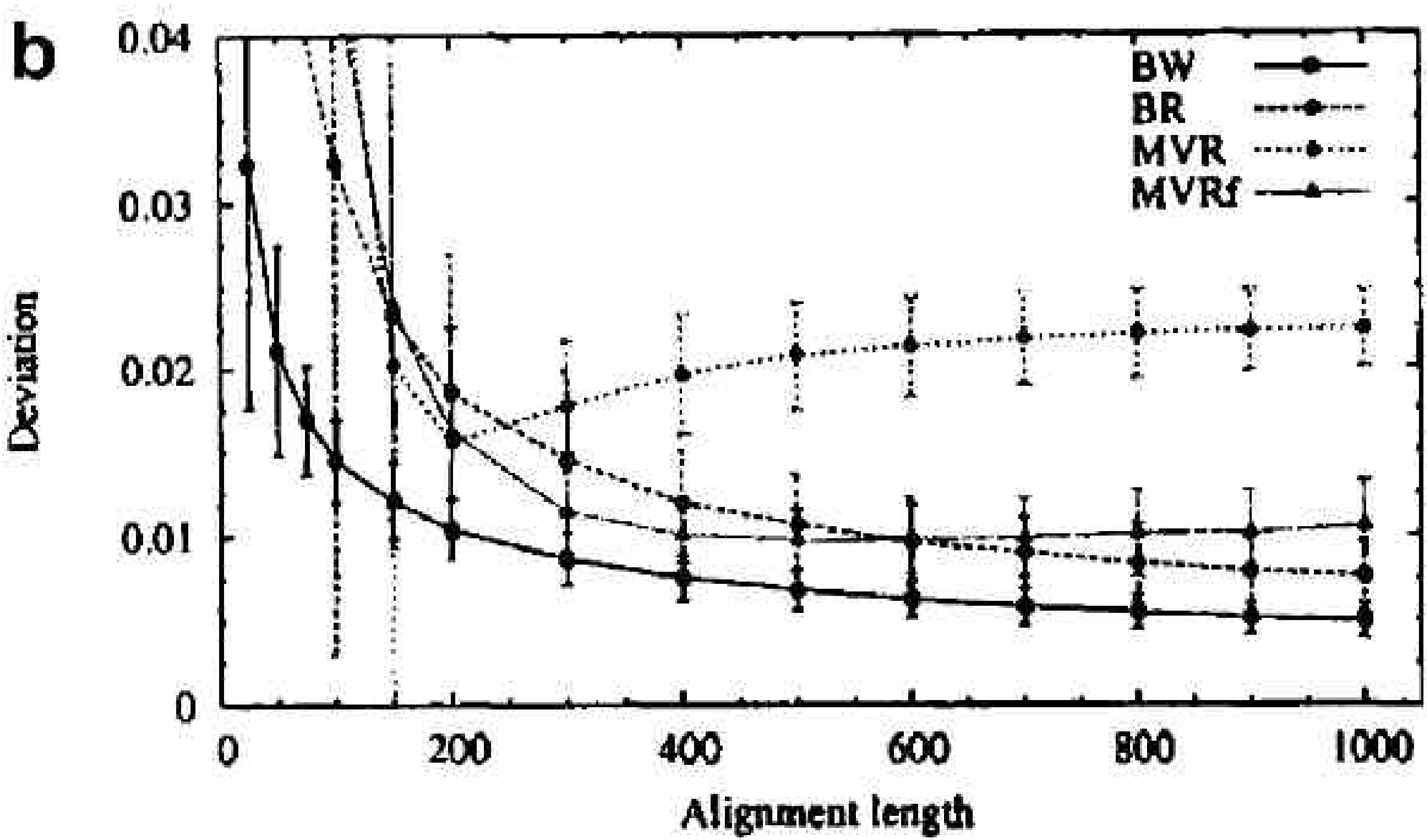
Übersicht

Gleichmäßige Verteilung gemäß 10j PAMs für Align. j



Übersicht

Ergebnisse für die Normalverteilug

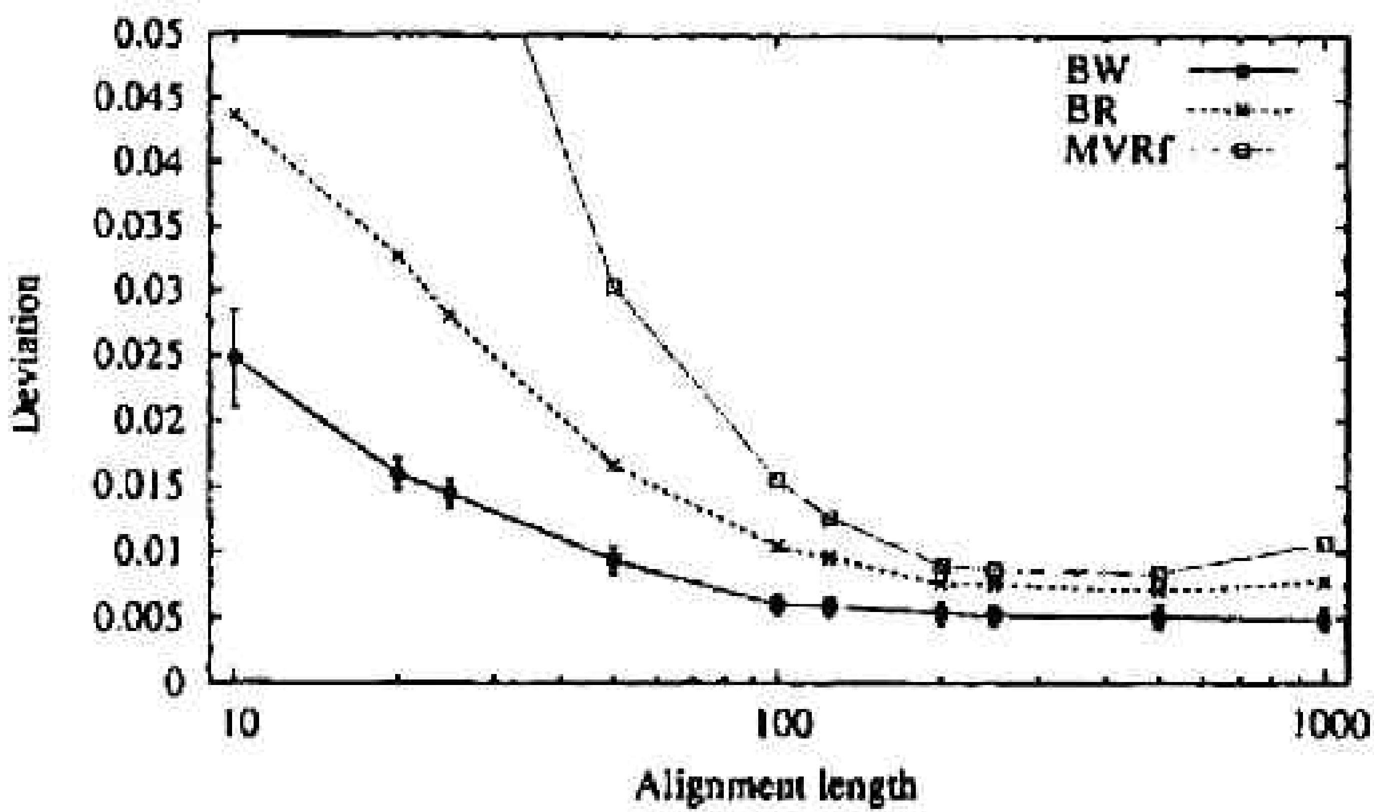


Zusammenfassung

- Die Resolvent Methoden (MVR, BR) brauchen eine gute Verteilung der Divergenzen als Eingabe
- Für die schlechten Ergebnisse für größere L bei MVR gibt es keine gute Erklärung
- Man kann feststellen, dass die MVR-Methoden für wenig Divergente Eingaben gute Ergebnisse bereits für kurze Alignments liefert
- Generell kann man sagen, dass die Genauigkeit von BW und BR mit wachsender Datenmenge zunimmt

Übersicht

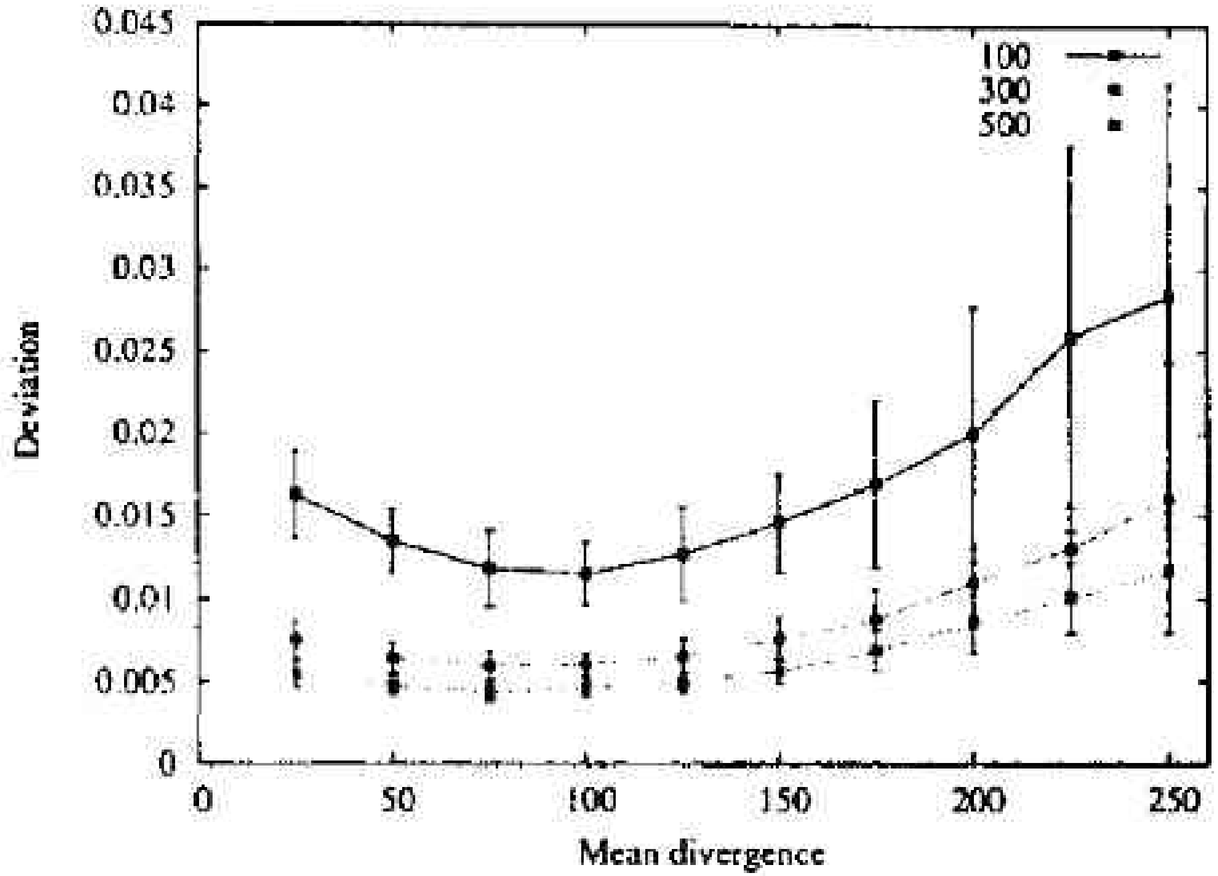
Wie wichtig ist die Alignment-Länge



- Jeder Punkt steht für die durchschnittliche Frobeniusnorm bei 1000 Durchläufen
- Dabei wurde folgende Gleichung Eingehalten: iL|T| = 25000

Übersicht

Wie wichtig ist die Divergenz bei BW



- Für die Alignmentlängen wurden die Werte 100, 300 und 500 gewählt

Endergebnis

- MVR und BR haben einen Nachteil, wenn ihnen eine kleine Anzahl an Sequenzen mit geringen Divergenzen als Eingabe übergeben werden
- BR ist besser als MVR, weil P(t) besser approximiert wird
- In den aktuellen Implementationen ist BW 5X schneller als BR und 8X schneller als MVR
- Auf geringe Verluste der Genauigkeit kann man BW durch kleineres T beschleunigen

Endergebnis

- BR und BW berechnen die Ratenmatritzen besser als MVR.
- **Der klare Sieger ist jedoch BW**

Ende