Algorithmische Bioinformatik

HMM:
Forward-Backward
Baum-Welch
Anwendung im Sequenzalignment

Ulf Leser

Wissensmanagement in der Bioinformatik

Inhalt der Vorlesung

- Evaluation
  - Globale Betrachtung: Forward-Algorithmus
  - Lokale Betrachtung: Forward-Backward Algorithmus
- Learning
- HMM für Sequenzalignment

Evaluation

- Viterbi berechnet die wahrscheinlichste Zustandsfolge für ein gegebenes HMM
- Aber wie sicher kann man sein, dass man das richtige HMM hat?
- Definition
  Gegeben ein HMM M und eine Sequenz von Zeichen S. Das Evaluationsproblem sucht nach der Wahrscheinlichkeit dafür, dass S von (einer Folge von Zuständen von) M erzeugt wurde.

Forward-Algorithmus

- Gegeben ein HMM, wie wahrscheinlich generiert es eine gegebene Sequenz S?
  - S kann durch verschiedene Zustandsfolgen erzeugt werden
  - Zur Evaluation müssen wir über alle diese aggregieren

$$p(S \mid M) = \sum_{p \in paths} p(S, p)$$

  - Beachte: Viterbi berechnet p*(S|M)
- p(S|M) kann durch eine kleine Variation des Viterbi-Algorithmus berechnet werden - Welche?

Änderung

- Viterbi
  - Sei $v_s(i)$ die Wsk des optimalen Pfad für S[..i], der in Zustand s endet
  - Gesucht: $v_t(i+1)$ für alle t
    - Von s mit a_{st} nach t
    - Dann Emission von S[i+1] mit $e_t(S[i+1])$
- Pfad mit der höchsten Wsk

$$v_t(i+1) = e_t(S[i+1]) * \max_{s \in M}(v_s(i) * a_{st})$$

- Forward-Algorithmus
  - Sei $f_s(i)$ die Gesamtwsk, dass nach i Schritten der Zustand s erreicht ist
    - Egal, über welchen Pfad
  - Gesucht: $f_t(i+1)$ für alle t
    - Von s mit a_{st} nach t
    - Dann Emission von S[i+1] mit $e_t(S[i+1])$
- Gesamtwahrscheinlichkeit

$$f_t(i+1) = e_t(S[i+1]) * \sum_{s \in M}(f_s(i) * a_{st})$$

Komplexität

- Ändert sich nicht (im Vergleich zu Viterbi)
- Sei |S|=n, HMM habe k Zustände
- Tabelle hat n*k Zellen
- Für jede Zelle greifen wir auf k Vorgängerzellen zu
- Zusammen: $O(n*k^2)$

Inhalt der Vorlesung

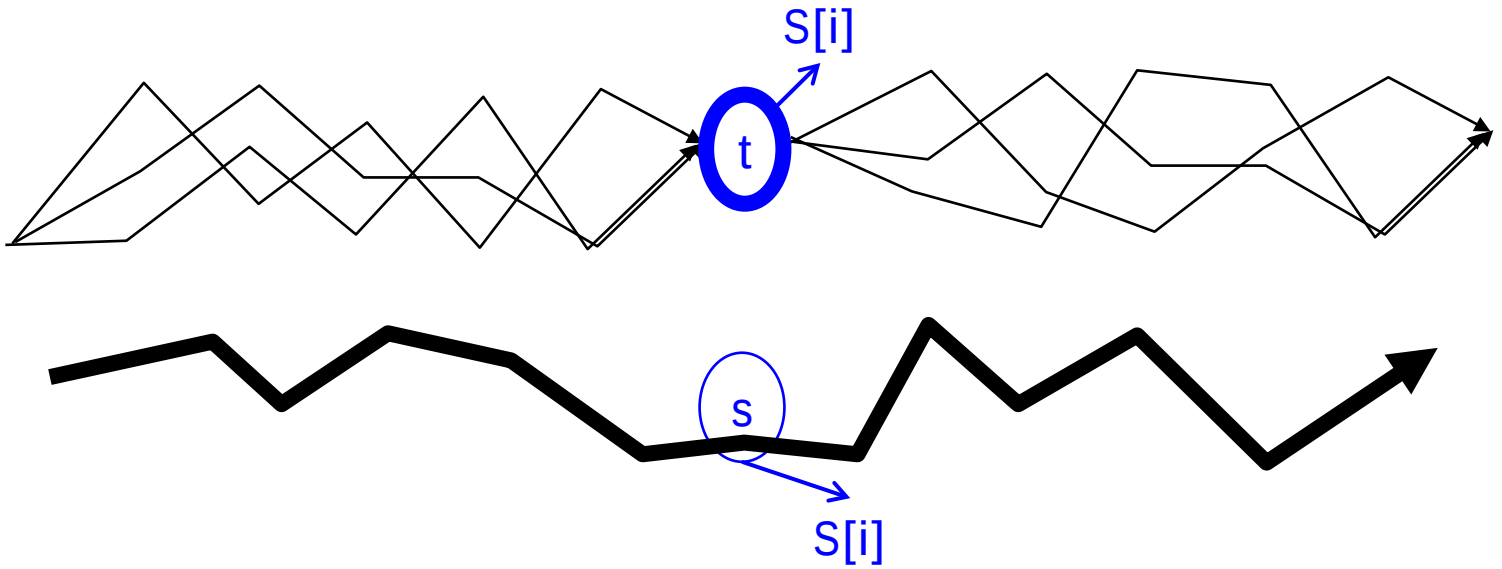
- Evaluation
  - Globale Betrachtung: Forward-Algorithmus
  - Lokale Betrachtung: Forward-Backward Algorithmus
- Parameterschätzung
- HMM für Sequenzalignment

Global / Lokal

- Evaluation hat eine zweite Facette: Welcher Zustand s von M hat am wahrscheinlichsten ein gegebenes Zeichen S[i] in S erzeugt?
  - Dieser Zustand muss nicht auf dem Viterbi-Pfad liegen
  - Das ist auch nicht einfach der Zustand mit höchster Emissionswahrscheinlichkeit für Z
    - Denn der Zustand muss im richtigen Augenblick (Position i) feuern
  - Wir müssen über alle möglichen Pfade aggregieren, in denen s das Zeichen S[i] an Position i emittiert
    - Und das für alle Zustände
- Evaluation?
  - Gute Pfade sollten viele gute Zustände beinhalten

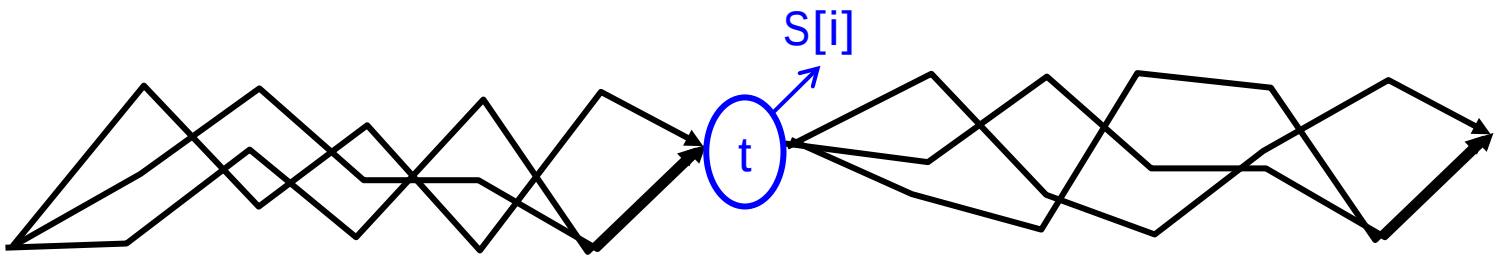
Illustration

Viele Pfade passieren Zustand t an Position i,
aber keiner hat maximale Wahrscheinlichkeit



Pfad mit maximaler Wahrscheinlichkeit
passiert Zustand s an Position i, hat aber
eher geringe Wsk dafür, dass s hier das
Zeichen S[i] emittiert

Problem



- Wir müssen die Wsk aller passenden Pfade summieren
- Bisher berechnen wir aber nur die Wsk wachsender Präfixe – über die Wsk der weiteren Pfade wissen wir an der Stelle i nichts
- Trick: Von vorne und hinten gleichzeitig rechnen
  - Forward / Backward Algorithmus

Berechnung

- Wir benötigen $$p^*(S[i] \mid M) = \max_{s \in M}\left(p(z_i = s \mid S)\right)$$
  - Diesen Term nennt man Posteriori-Wsk, da man ihn erst nach Betrachtung der gesamten Sequenz berechnen kann
  - Eigentlich muss überall ein „ ‚M“ dazu – sparen wir uns meistens
- Die Berechnung erfolgt indirekt
  - Es gilt (für alle s) $$p(z_i = s \mid S) = \frac{p(z_i = s, S)}{p(S)}$$
  - p(S) kann mit Vorwärtsalgorithmus berechnet werden
  - Für den Zähler gilt

HMM erster Ordnung

$$p(z_i = s, S) = p(z_i = s, S[1..i]) * p(S[i+1..n] \mid S[1..i], z_i = s)$$
$$= p(z_i = s, S[1..i]) * p(S[i+1..n] \mid z_i = s)$$

Backward Algorithmus

$$p(z_i = s, S) = p(z_i = s, S[1..i]) * p(S[i+1..n] \mid z_i = s)$$
$$= \quad f_s(i) \quad * \quad b_s(i)$$

- $f_s(i)$ kennen wir: Forward-Algorithmus
- $b_s(i)$ heißt die Backward-Wsk
  - Wsk für die Restsequenz S[i+1..n], gegeben Zustand s an Position i

Backward Algorithmus

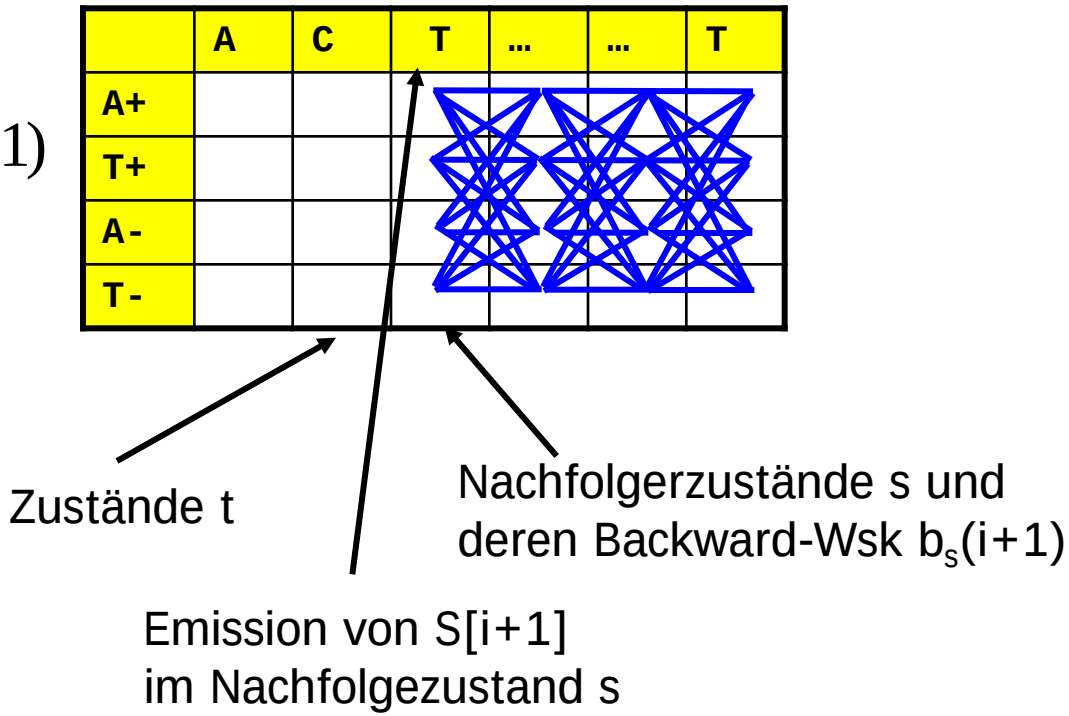
- $b_s(i)$ berechnet man iterativ von n nach 1

$$b_t(i) = \sum_{s \in M} a_{ts} * e_s(S[i+1]) * b_s(i+1)$$

  - In Schritt i+1 kennen wir $b_s(i+1)$, also die Wsk der Restsequenz S[i+1..n] bei Zustand s in Position i+1
  - Berechnung $b_t(i)$
    - So wäre es vorwärts: Gehe von t nach s, emittiere S[i+1], und fahre fort mit den $b_s(i+1)$
    - Wir gehen aber rückwärts: Komme von $b_s(i+1)$, emittieren S[i+1], gehe von s nach t
- Initialisierung: Virtuelle Endzustand mit Wsk 1
  - D.h. $a_{s,n+1}=1$ für alle s

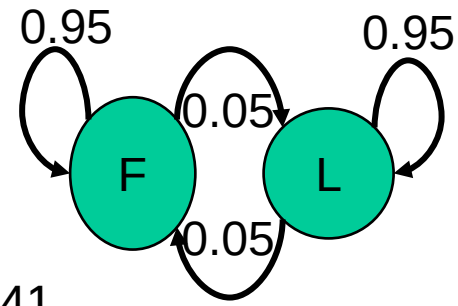
Veranschaulichung

$$b_t(i) = \sum_{s \in M} a_{ts} * e_s(S[i+1]) * b_s(i+1)$$



- Achtung: e(...) ist jetzt in der Klammer (anders als bei Forward-Algorithmus), da das Zeichen nach dem Übergang emittiert wird

Beispiel



12641623562615636612341312265415524 1
12641623562615636612341312265415524 1

- Der wahrscheinlichste Einzelpfad mag nur aus F bestehen
- Aber es gibt vermutlich viele Pfade, die dem blauen Bereich den Zustand L mit höherer Wsk zuordnen als der optimale Pfad ihm den Zustand F zuordnet
  - Z.B. ist die Einzelwsk jeder 6 in L viel höher als in F
- Tritt z.B. auf, wenn
  - jeder Übergang so stark bestraft wird (kleinen Übergangs-Wsken), dass sich Wechsel erst bei langen L-Phasen lohnen
  - Die Unterschiede zwischen den Pfad-Wsk sehr klein sind

Posterior-Pfade

- Kann man aus den Zuständen einer Posterior-Dekodierung einen kompletten Pfad bauen? Wie wahrscheinlich ist der?
  - Natürlich kann man einen Pfad bauen
  - Der kann aber Gesamtwahrscheinlichkeit 0 haben
    - Wenn er z.B. einen Übergang mit Wsk 0 enthält
- Posterior-Dekodierung ist eine lokale Betrachtung

Inhalt der Vorlesung

- Evaluation
- Learning
  - Maximum Likelihood Estimator
  - Baum-Welch Algorithmus
- HMM für Sequenzalignment

Lernen eines HMM

- Wir können
  - Berechnen, wie wahrscheinlich eine Sequenz ist
  - Berechnen, was der wahrscheinlichste Pfad für eine Sequenz ist
  - Berechnen, welches der wahrscheinlichste Zustand für jedes einzelne Zeichen einer Sequenz war
  - ... gegeben ein festes Hidden Markov Model
- Jetzt: Lernen der Parameter eines HMM aus Beispieldaten
  - Parameter sind die Emission-/ Übergangs-Wsk
  - Die Struktur des HMM wird als gegeben vorausgesetzt
- Struktur: Eigentlich nur die Zahl der Zustände
  - Oft setzt man aber auch bestimmte Übergänge fest auf Wsk=0

Einfacher Fall: Zustandssequenz bekannt

- Definition
  Gegeben ein HMM M mit fester Struktur und eine Sequenz S. Das Lernproblem findet die Übergangs-Wsk a_{st} und Emissions-Wsk $e_s(x)$ von M so, dass p(S|M) maximal ist.
- Bemerkung
  - S sollte möglichst lang sein (oder es gibt viele S), um ein möglichst robustes Modell zu lernen (Overfittingproblem)
- Erster Fall
  - Wir kennen zusätzlich zu jedem Zeichen von S den Zustand, der das Zeichen emittiert hat - wir haben gelabelte Trainingsdaten
  - Dann ist ein Maximum Likelihood Estimator möglich
  - Und findet auch das optimale Modell (Beweisen wir nicht)

Maximum Likelihood Estimation für HMM

- Zählen der relativen Häufigkeit a_{st} aller Zustandsübergänge für alle Paare von Zuständen s und t
  - Sei A_{st} die Zahl der Übergänge s→t
  - Dann

$$a_{st} = p(t \mid s) = \frac{A_{st}}{\sum_{t' \in M} A_{st'}}$$

- Zählen der relativen Häufigkeit aller Emission $e_s(x)$ für alle Zustände s und Zeichen x
  - Sei $E_s(x)$ die Häufigkeit, mit der Zustand s Zeichen x emittiert
  - Dann

$$e_s(x) = \frac{E_s(x)}{\sum_{x' \in \Sigma} E_s(x')}$$

Smoothing

- Vorsicht vor Nullen
  - Seltene Übergänge/Emissionen fehlen in kleinen Trainingsmengen
  - Nuller setzen die Wsk von Pfaden auf 0, egal, wie wahrscheinlich der Restpfad ist
- Overfitting: Wir verlassen uns zu sehr auf die Trainings-daten, die aber nur ein Sample der Realität sind
- Smoothing: Artifizielle Höhergewichtung nicht beobachteter Ereignisse
  - Einfachste Lösung: Jeden Übergang / Ausgabe mit einer sehr kleinen Frequenz annehmen („adding 1“)
  - Viele weitere Vorschläge

Inhalt der Vorlesung

- Evaluation
- Learning
  - Maximum Likelihood Estimator
  - Baum-Welch Algorithmus
    - Baum, L. E. and Petrie, T. (1966). "Statistical Inference for Probablistic Functions of Finite State Markov Chains." *Annals of Mathematical Statistics* **37**: 1554-1563.
- HMM für Sequenzalignment

Lernen bei unbekannter Zustandssequenz

- Definition (Wdh)
  Gegeben ein HMM M mit fester Struktur und eine Sequenz S. Das Lernproblem findet die Übergangs-Wsk a_{st} und Emissions-Wsk $e_s(x)$ von M so, dass p(S|M) maximal ist.
- Zweiter Fall: Wir kennen die Zustandssequenz nicht
  - Keine analytische Lösung bekannt
  - Prinzipiell kann man viele Suchheuristiken anwenden
    - Wähle eine Startkonfiguration
    - Bringe viele kleine/große Veränderungen an und betrachte die Veränderung von p(S|M)
    - Übernimm die Veränderungen, die sich positiv auswirken
    - Wiederhole solange, bis ...
  - Findet nur lokale Optima
  - Bekannteste: Baum-Welch Verfahren

Baum-Welch Algorithmus

- Iterative lokale Suchheuristik
- Sei S die (ungelabelte) Trainingssequenz
- Rate Startkonfiguration a_{st} und $e_s(x)$
- Berechne k=p(S|M)
- while (true)
  - Berechne neue Übergangs-Wsk a_{st}
  - Berechne neue Emissions-Wsk $e_s(x)$
  - Berechne k‘=p(S|M) (mit dem neuen Modell)
  - if (k‘-k < t): stop
  - else: k:=k‘
- end while

Anpassung

- Zur Modellanpassung evaluieren wir jede Position i von S
  - Berechnung des maximal wahrscheinlichen Zustands für i
    - Lokale Evaluation
  - Diesen wollen wir in unserem Modell nach Möglichkeit auch gehen
- Sei s der insgesamt wahrscheinlichste Zustand an Position i und t der insgesamt wahrscheinlichste Zustand an Position i+1
- Dann machen wir a_{st} und $e_s(S[i])$ in der nächsten Iteration wahrscheinlicher

Konkret: Neue Übergangs-Wsk a_{st}

- Wsk des Übergangs s→t an Position i im aktuellen Modell

$$p(z_i = s, z_{i+1} = t \mid S) = p(s_i \to t_{i+1} \mid S) = \frac{f_s(i) * a_{st} * e_t(S[i+1]) * b_t(i+1)}{P(S)}$$

  - Alle Pfade bis s, Übergang zu t, t emittiert $S_k[i+1]$, alle Pfade von t bis zum Ende
- Aggregation über alle Positionen in S und alle Ausgaben gibt Erwartungswert für a_{st} gegeben das aktuelle Modell
- Den setzen wir für das neue Modell ein

$$a_{st}^{neu} = \sum_{i=1..n} \frac{p(s_i \to t_{i+1} \mid S)}{p(s_i \mid S)}$$

Neue erwartete Emissions-Wsk

- Berechnung der neuen $e_s(x)$

Gesamtwsk, dass Zustand s an Pos i Zeichen x ausgibt (wobei x=S[i]

$$e_s^{neu}(x) = \frac{1}{\sum_{i=1..n} p(s_i \mid S)} * \sum_{\{i|S[i]=x\}} \frac{f_s(i) * b_s(i)}{P(S)}$$

Normiert auf die Wsk, in Pos i in Zustand s zu sein

Aggregiert über alle Pos i mit S[i]=x

- Auch Werte von f und b müssen aktualisiert werden

Eigenschaften

- Man kann zeigen: Der Baum-Welch Algorithmus konvergiert gegen ein lokale Optimum
  - Konfiguration wird nie schlechter
  - Umgang mit lokalen Optima: Verschiedene Startkonfigurationen
- Wenn mit Real-Zahlen gerechnet wird, terminiert BW ohne Schwellwert (k‘-k=0) i.A. nicht
  - Ewige, infinitesimal kleine Verbesserungen
- BW ist Spezialfall des Expectation Maximization Algorithmus (EM)
  - Siehe Soft-Clustering

Was wir nicht behandeln

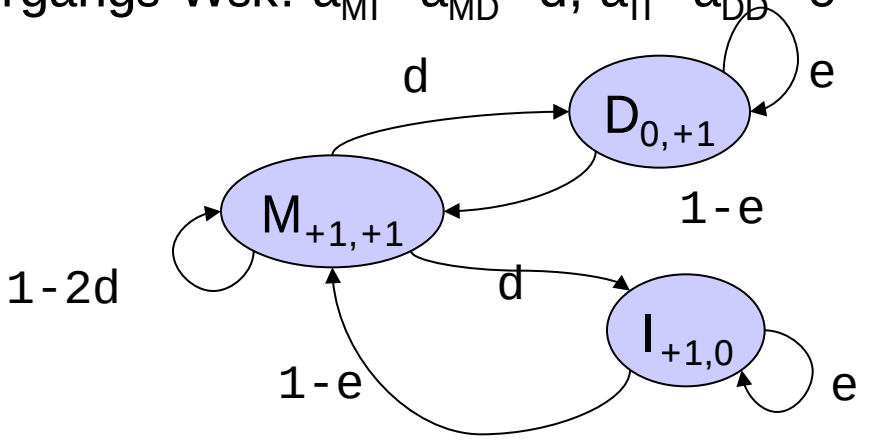
- Wie kommt man zu einer guten Struktur?
  - Trade-Off: Komplexere Strukturen (mehr Zustände) passen sich besser an, aber man muss auch mehr Parameter raten
  - Eine Lsg: Größe des HMM als Parameter in die Optimierung
  - Sonst: „... more of an art [than a science]“
- Wie kommt man zu einer guten Startkonfiguration?
  - BW ist deterministisch – gleiche Startkonfiguration, gleicher Verlauf
  - Startkonfiguration sollte externes Wissen einbeziehen – welche Übergänge sind häufig, welche selten?
  - Diverse Vorschläge in der Literatur
  - Immer eine gute Idee: Verschiedene ausprobieren

Inhalt der Vorlesung

- Evaluation
- Parameterschätzung
- HMM für Sequenzalignment (sketched)

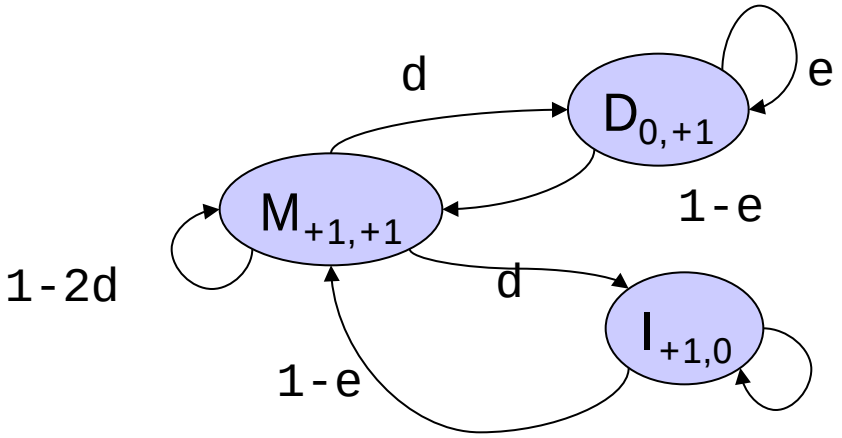
HMMs im Sequenzalignment

- Wir alignieren Sequenzen S und T
- Wir bauen einen Paar-HMM
  - Jede Emission ist eine Spalte des Alignment
  - Die Zustände sind M, I, D
  - Zustand M emittiert Basenpaar (X,Y) an Position i mit Wsk $e_M(X,Y)$
  - Zustand I, D emittiert eine InsDel an Position i mit Wsk p(X)
    - $X \in S$ oder $X \in T$
  - Feste Übergangs-Wsk: $a_{MI}=a_{MD}=d$, $a_{II}=a_{DD}=e$



Emissionswahrscheinlichkeiten nicht gezeigt

Verdeutlichung




```
-AGGCTATCACCTGACCTCCAGGCCGA--TGCCC---
TAG-CTATCAC--GACCGC-GGTCGATTTGCCCGACC
IMMDMMMMMMMDDMMMMMMDMMMMMMMIIMMMMMIII
```

Paar-HMM‘s und Editabstand

- Das optimale Alignment entspricht der optimalen Zustandsfolge durch das Paar-HMM
- Wir können über den Forward-Algorithmus auch die Gesamtwsk berechnen, in wie fern S und T „ähnlich“ unter M sind
  - Damit berechnen wir implizit einen Score über alle möglichen Alignments von S und T
  - Gerade wenn zwei Sequenzen nicht allzu gut alignieren, gibt es meistens viele beste, aber ähnlich schlechte Alignments
  - Der Forward-Algorithmus berechnet einen Wert für die Güte und fasst damit diese Pfade quasi zusammen
- Sehr erfolgreich bei der Berechnung von Multiple Sequence Alignments (siehe später)

Training für HMM-Alignment

- Wo kommen die Parameter her?
- Können aus Trainingsdaten (=alignierte Sequenzen) gelernt werden
- Oder aus Substitutionsmatrizen (PAM / BLOSUM) abgeleitet werden

Literatur

- Gusfield behandelt das Thema nicht
- Literatur
  - Merkl (2015). „Bioinformatik (interaktiv)“, Kapitel 15
  - Rabiner, L. R. (1988). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE* 77(2): 257-286.
  - Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998). "Biological Sequence Analysis: Probablistic Models of Proteins and Nucleic Acids". Cambridge, UK, Cambridge University Press.
  - Anders Krogh (1998). „An Introduction to Hidden Markov Models for Biological Sequences“. http://www.binf.ku.dk/~krogh/publications/ps/Krogh98a.pdf