Identifizierung von bakteriellen Genen und endosymbiontischer DNA

mit Glimmer 3

M. Siebauer

Universität Leipzig

09.07.2007 / Problemseminar

Problemstellung

Ziel: Identifizierung von Genen in mikrobiellen Genomen

Problem:

- Prokariontische Genome gen-reich ($\sim$ 90% codierend)
- Genauer: Welches ORF (*Open Reading Frame*) ist wirkliches Gen?

Lösungen:

- Statische Suche nach homologen Genen (BLAST, FASTA)
- Aber: Nicht immer holomoge Gene bekannt
- Dynamisches Scoring (GeneMark)
- Aber: Benötigt Unmenge an Trainingsdaten

Markov Modell

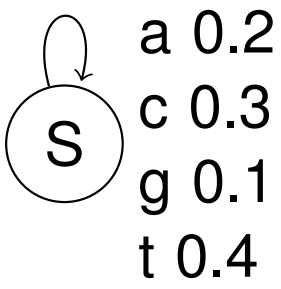


Figure: Sample 1-state Markov model for simple sequence modeling[1]

$$P(aaaaa) = (0.2)^5 = 0.00032$$

Markov Kette

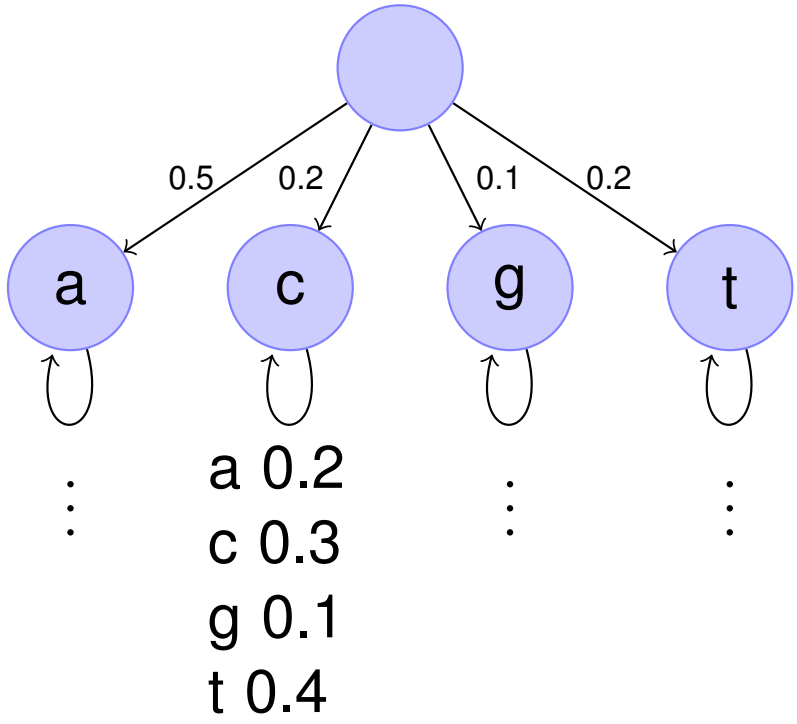


Figure: Markov Kette 1. Grades

Markov Ketten

- Ist Linearkombination von Markov Modellen
- Kette vom Grad k *berechnet* aus k vorherigen Basen die Folgende
- GeneMark nutzt Markov Kette 5. Grades
- Problem:
  - Markov Kette 4. Grades bereits $4^{4+1} = 4096$ Wahrscheinlichkeiten
  - Für stochastisch sichere Vorhersage viele Trainingsdaten nötig

Hier kommt Glimmer ins Spiel!

Interpolierte Markov-Modelle (*IMM*)

- Alle Markov Ketten von Grad 0 bis 8 werden berechnet
- Ketten werden gewichtet nach Häufigkeit in den Trainingsdaten
- Wenn Trainingsdaten für höheren Grad unzureichend, kann auf niedrigeren Grad zurückgefallen werden

$$P(S|M) = \sum_{x=1}^{n} \mathrm{IMM}_8(S_x)$$

$$\mathrm{IMM}_8(S_x) = \underbrace{\lambda_8(S_{x-1})}_{\text{Gewicht}} * \mathrm{P}_8(S_x) + (1 - \lambda_8(S_{x-1})) * \mathrm{IMM}_7(S_x)$$

$$\mathrm{P}_i(S_x) = \mathrm{P}(s_x|S_{x-i}, .., S_{x-1}) = \frac{f(S_{x,i})}{\sum_{b\in\{acgt\}} f(S_{x,i},b)}$$

Bsp: S = AGA..

$$P_2(S_3) = P(A|AG) = \frac{f(AG)}{f(AGA)+f(AGC)+f(AGT)+f(AGG)}$$

Berechnung der Gewichte

Gewicht $\lambda_i(S_x)$

- ist 1.0, falls Vorkommen von $S_{x-i}..S_{x-1}$ in den Trainingsdaten, Schwellwert (*400*) übersteigt

  (Andere Werte konnten experimentell keine höhere Erkennungsrate liefern)

- Andernfalls:
  - Häufigkeit der Basen $f(S_{x,i}, b)$ $b \in \{agct\}$ werden mit den Vorhersagen des nächst kürzeren Model $\mathrm{IMM}_{i-1}(S_{x,i-1}, b)$ verglichen
  - Wenn Unterschiede bestehen (mittels χ^2-Test) wird höheres Gewicht vergeben, genauer:
  - $\lambda_i(S_{x-1}) = \begin{cases} 0.0 & \text{if } c < 0.50 \\ \frac{c}{400} \sum f(s_1 s_2 \ldots s_i b)_{b \in \{acgt\}} & \text{if } c \geq 0.50 \end{cases}$

c := Konfidenz das Vorhersagen des kleineren Modells **nicht** übereinstimmen

Glimmer 1 besteht aus zwei Programmen:

- build-imm Berechnet IMM aus Trainingsdaten
- glimmer Sucht lange *ORF*s und berechnet Score für alle 6 Lesarten

Ablauf

- Glimmer berechnet 7 IMM-Modelle (6 Lesearte + 1 Modell aus nicht kodierenden Abschnitten)
- Glimmer sucht alle *ORF*s und berechnet Score für alle Modelle
- *ORF*s mit ausreichendem Score werden auf Overlaps untersucht Glimmer 1 verwirft *ORF* mit niedrigerem Score
- Glimmer 1 hat noch keine wirkliche Overlap Behandlung

Trainingsdaten

Glimmer benötigt Trainingsdaten als Koordinaten-Datei. Diese kann man bekommen aus:

- NCBI Datenbank (`.ptt`, `.nh`?).
- dem zu untersuchenden Genom selber (`long-orfs`).
- homologen Genen verwandter Organismen.

Vergleich des IMM Modell mit einem Markov Modell [1]

Modell	Gene gefunden	Genes verpasst	Zusätzliche Gene
GLIMMER IMM	1.680 (97,8%)	37	209
Markov 5.Grades	1.575 (91,7%)	143	104

Bezogen auf die 1717 annotierten Gene in *H. influenzae*. Als Trainingsdaten wurden nur *ORF*s $>$ 500 Basen genutzt.

Motivation

Probleme:

- Erkennungsleistung nur bei $\sim$97–98%
- Zu hohe *false-positives rate*

  Obwohl die Autoren im Artikel eher davon ausgehen, dass es neue noch nicht annotierte Gene sind
- Zuviele Gene wegen fehlender Overlap-Behandlung verworfen

Interpolierte Kontext-Modelle (ICM)

Das ICM ist eine Weiterentwicklung des IMM.

Idee:

- Vorhersage einer Base hängt nicht nur von direkten Vorgängern ab
- Position einer Base im Kontext ist ausschlaggebender
- Bei AS-Translation ist Base in der dritten Position meist irrelevant

Interpolierte Kontext-Modelle (ICM)

Die wechselseitige Information I zweier Zufallsvariablen X, Y ist:

$$I(X;Y) = \sum_i \sum_j P(x_i, y_j) * log\left(\frac{P(x_i)P(y_j)}{P(x_i, y_j)}\right)$$

x_i, y_j . . . Werte die X,Y annehmen kann

$P(x_i, y_j)$ . . . Wahrscheinlichkeit das x_i und y_j auftreten

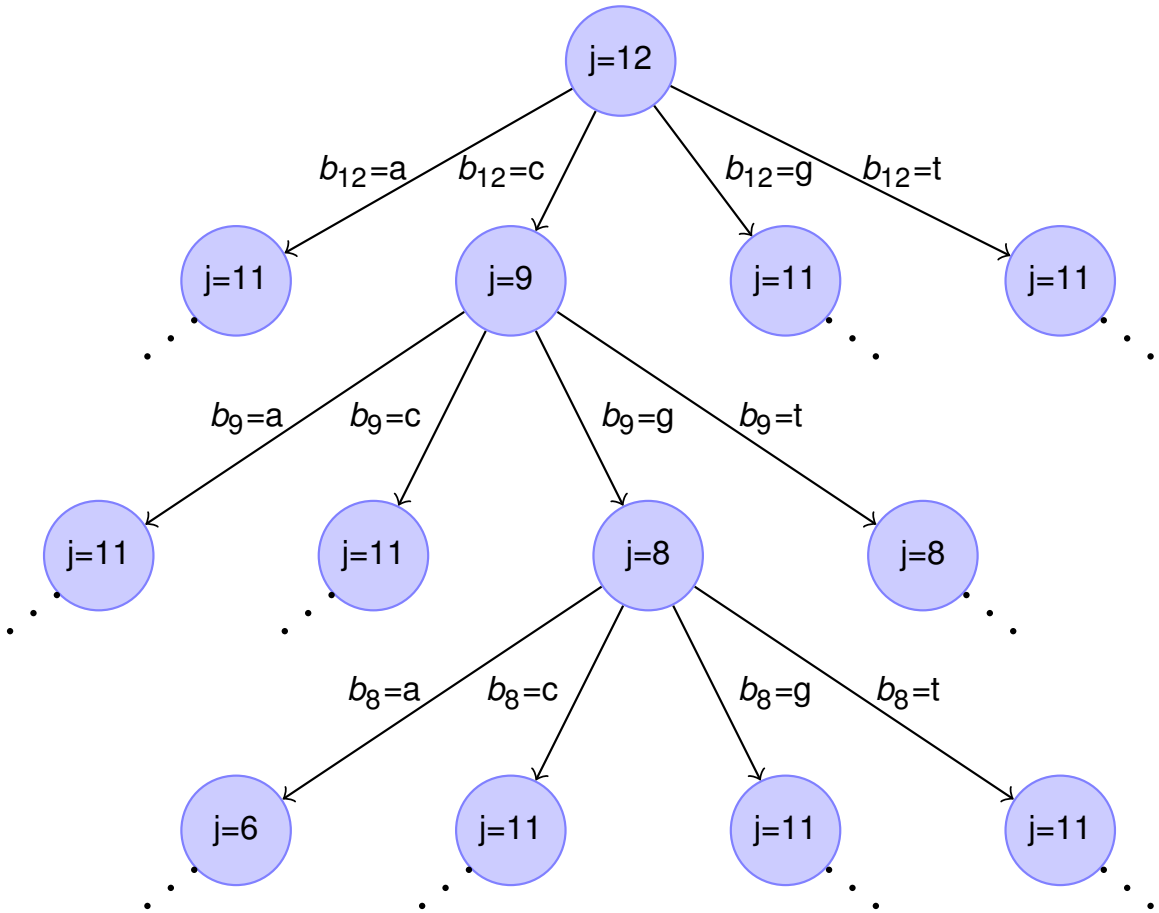
Vorgehensweise

- Sequenz S wird in Fenster der Länge $k + 1$ zerlegt (k = KontextLänge)
- Wechselseitige Informationen (*mutual information*) $I(X_1, X_{k+1}), I(X_2, X_{k+1}), \ldots, I(X_k, X_{k+1}))$ werden berechnet
- Maximum wird gesucht $\Rightarrow I(X_j, X_{k+1})$
- Die Fenstermenge wird in 4 Teilmengen zerlegt, sortiert nach der Base an Position j
- Für jede der 4 Teilmengen beginnt der Algorithmus von vorn

Bis vorgegebene Tiefe erreicht oder Fenstermenge zu klein wird

Es entsteht ein Zerlegungsbaum

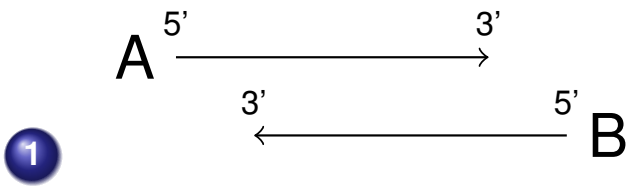
ICM Zerlegungsbaum [2] Fig. 1



- Jeder Knoten im Baum entspricht einer Menge von Fenster
- Wurzel enthält alle Fenster und entspricht einem Markov Model 0. Grades
- Alle andere Knoten geben eine Wahrscheinlichkeitsverteilung für die letzte Basis (*k+1*) unter Voraussetzung das bestimmte Basen an bestimmten Positionen auftreten
- Ist Maximum stets die letzte Base im Fenster, ist der Baum gleich dem IMM aus Glimmer 1

Overlap Behandlung

Glimmer 2 versucht bei Overlaps alternative Start Codon Positionen zu finden.
Angenommen 2 potentielle Gen A, B überlappen; Gen A hat im Moment höheren Score.
Es gibt 4 Möglichkeiten:

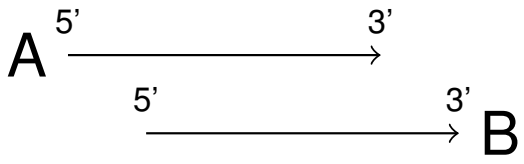


1

Start nicht verschiebbar; Wenn A länger als B wird A genommen, andernfalls Beide (mit einer Anmerkung)

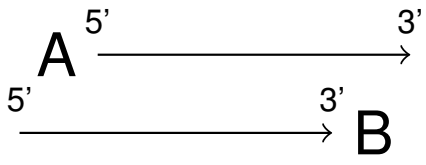
Overlap Behandlung

2



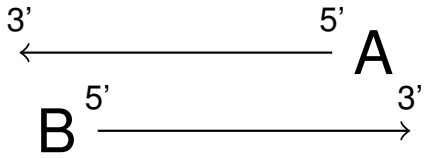
B Start kann verschoben werden; wird verschoben bis Overlap gelöst oder Gene unter minimal Länge

3



A Start kann verschoben werden; Da A höheren Score hat, wird Start nur bei sehr geringem Overlap verschoben; B wird andernfalls verworfen

4



Beide verschiebbar; B Start wird geschoben bis Overlap Region höheren Score als für A liefert, dann wird A Start geschoben bis Score wieder höher füer A. Abwechselnd weiter bis Overlap gelöst oder kein Schritt mehr möglich

Nach jedem “Rauswurf” eines Gens werden alle Overlaps

Vergleich der Genuigkeiten

Glimmer1 vs. Glimmer2 [2]

		Glimmer 1.0		Glimmer 2.0	
Organismus	Gene annotiert	Genes gefunden	Zusätzliche Gene	Genes gefunden	Zusätzliche Gene
H. influenzae	1.738	1.715 (98,7%)	234 (13,5%)	1.720 (99,0%)	242 (13,9%)
M. genitalium	483	479 (99,2%)	78 (16,1%)	480 (99,4%)	82 (17,0%)
M. jannaschii	1.727	1.715 (99,3%)	210 (12,2%)	1.721 (99,7%)	218 (12,6%)
H. pylori	1.545	1.545 (97,2%)	293 (18,4%)	1.550 (97,5%)	322 (20,3%)
E. coli	4.259	4.099 (96,0%)	757 (17,7%)	4.158 (97,4%)	868 (20,3%)
B. subtilis	4.100	4.006 (97,7%)	917 (22,4%)	4.030 (98,3%)	1.022 (24,9%)
A. fulgidus	2.437	2.385 (97,9%)	274 (11,2%)	2.404 (98,6%)	341 (14,0%)
B. burgdorferi	849	845 (99,5%)	67 (7,9%)	843 (99,3%)	62 (7,3%)
T. pallidum	1.039	1.012 (97,4%)	180 (17,3%)	1.014 (97,6%)	250 (24,1%)
T. maritima	1.877	1.849 (98,5%)	190 (10,1%)	1.854 (98,8%)	208 (11,1%)

Quelle: [2] Table 1

Motivation

Ziele:

- Reduzierung der false-positives Gene
- Filterung endosymbiontischer Fremd-DNA

Reverse Scoring

Glimmer 3 berechnet Score rückwärts beginnend mit dem Stop Codon

Gründe

- IMM ist auf reines Gen trainiert → Übergang nicht codierend zu codierend liefert im Kontext Fenster niedrigen Score
- Score wird ständig aufsummiert, so dass beim Erreichen des korrekten Start-Codons ein Maximum erreicht werden wollte (nicht codierende Abschnitte liefern negativen Score)

Glimmer 1/2 bevorzugten längere *ORF*s; Glimmer 3 favorisiert höhere Scores

Reverse Scoring

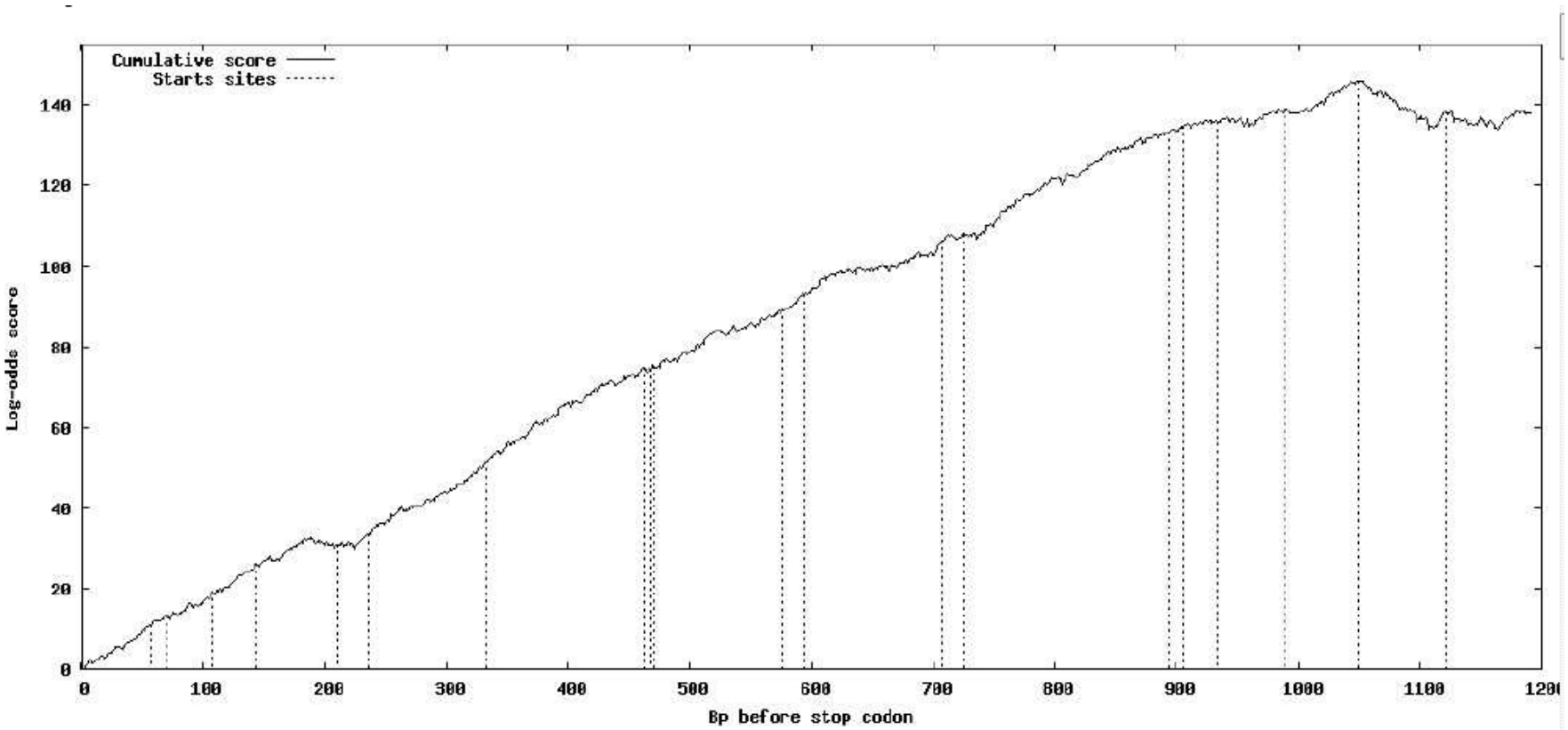


Figure: Quelle: [3]

Ribosomale Bindungsstellen

- Ribsomale Bindungsstelle liefert starken Hinweis auf korrektes Start Codon, wurde aber bisher von Glimmer ignoriert.
- Ursprünglich Postprozessor RBSfinder über Glimmer Ausgabe, jetzt fest integrierter Algorithmus:
  - Open Source Programm ELPH sucht *Motife* aus Menge von Sequenzen
  - ELPH berechnet mittels Gibbs-Sampling Algorithmus eine PWM (*position weight matrix*)
  - Glimmer nutzt die PWM um potentielle *RBS* zu scoren
- Wenn a priori keine Trainingsdaten verfügbar, wird ELPH über Glimmer Vorhersagen geschickt, und dann nochmals Glimmer, mit den PWM Daten, ausgeführt

Overlap Behandlung

Hohe *false-positives* Rate von Glimmer2 primär durch viele Overlaps bei hoch-codierenden Genomen

- Glimmer3 berechnet zunächst **alle** möglichen *ORF*s zwischen allen Start und Stop Codons
- Mittels dynamischem Algorithmus wird versucht Menge von *ORF*s zu einer gültigen Sequenz zusammenzusetzen, bei maximalem Gesamtscore und minimalsten Overlaps:
  - *ORF* werden nach Pos. in Sequenz sortiert
  - bei jedem Start oder Stop Merkmal *f* wird globaler Score bis, und inklusive, *f* berechnet (in allen 6 Lesarten)
  - Da kurze Overlaps erlaubt, wird Gesamtscore nach jedem Schritt neuberechnet, da sonst Overlaps doppelt gescoret würden

Verbessertes Training bei *long-orfs*

Bei Glimmer2 wurden *ORF*s > 500bp gewählt. Glimmer3 bestimmt den Schwellenwert iterativ selbst solange keine Overlaps auftreten.

Problem: Bei hoch codierenden Sequenzen ($\sim >$ 60%) verursacht der Mangel an Stop Codons zu lange *ORF*s!

Lösung: Glimmer 3 hat einen zusätzlichen Filter:

- Nach *Luscombe et al. (2001); Pascal et al. (2005)* gibt es gemeinsame Aminosäure-Verteilung in Genen aller Bakterien
- Glimmer besitzt ein universelles Verteilungsmodell, geschaffen aus sehr großer Anzahl von Genen, sowie ein negatives Modell
- Glimmer berechnet für jedes *ORF* die AS-Verteilung und bestimmt den Abstand zu beiden Modellen

Genauigkeit

Glimmer 3 [3]

Genome			Glimmer3 Vorhersagen				
Organismus	%GC	# Gene	3' passt		5' & 3' passt		Extra
A. fulgidus	49	1.165	1.162	99.7%	841	72.2%	1.308
B. anthracis	35	3.132	3.119	99.6%	2.717	86.7%	2.345
B. subtilits	44	1.576	1.559	98.9%	1.379	87.5%	2.886
C. tepidum	57	1.292	1.284	99.4%	867	67.1%	778
C. perfringens	29	1.504	1.501	99.8%	1.360	90.4%	1.177
E. coli	51	3.603	3.525	97.8%	3.014	83.7%	942
G. sulfurrenducens	61	3.251	2.320	98.7%	1.883	80.1%	1.107
H. pylori	39	915	908	99.2%	785	85.8%	774
P. flourescens	63	4.535	4.484	98.9%	3.412	75.2%	1.896
R. solanacearum	67	2.512	2.468	98.2%	1.922	76.5%	1.091
S. epidermidis	32	1.650	1.646	99.8%	1.496	90.7%	767
T. pallidum	53	575	569	99.0%	397	69.0%	568
U. parvum	26	327	325	99.4%	292	89.3%	297
Averages				99.1%		81.1%	

Quelle: [3] Table 2

Glimmer3 vs. andere GenFinder

Genome		vs. GeneMark.hmm			vs. EasyGene 1.2			vs. GeneMarkS		
Organismus	# Gene	3' passt	5' & 3' passt	Extra	3' passt	5' & 3' passt	Extra	3' passt	5' & 3' passt	Extra
A. fulgidus	1.165	+4	–20	–86	+5	–25	+119	0	+2	–71
B. anthracis	3.132	–2	–48	–134	+13	–63	+175	+1	+412	–142
B. subtilits	1.576	+2	+280	+87	+15	–10	+536	–5	–39	+194
C. tepidum	1.292	+1	+21	+19	+10	+9	+182	+1	–14	+29
C. perfringens	1.504	–2	+177	–120	–2	–8	–21	–3	–14	–139
E. coli	3.603	–25	+18	+188	+60	+44	+407	–25	–29	+190
G. sulfurrenducens	3.251	+13	+215	+34	+5	–1	+60	+14	+41	+66
H. pylori	915	–1	–3	–55	+4	–6	+148	–1	–8	–41
P. flourescens	4.535	+17	+288	+59	NA	NA	NA	+17	+479	+46
R. solanacearum	2.512	+7	+183	+255	+11	+48	+193	–3	+160	+190
S. epidermidis	1.650	+3	–32	–40	NA	NA	NA	+6	+204	–64
T. pallidum	575	+2	–8	+94	+8	+8	+176	–2	–18	+90
Averages		+2	+89	+23	+13	–2	+198	+2	+98	+29

Quelle: [3] Table 4

Andere Anwendungsmöglichkeiten

Durch das flexible IMM Modell kann Glimmer3 leicht auf andere Aufgaben als nur Gen-Identifizierung trainiert werden:

- *P. didemni* ist photosynthetische Mikrobe die nur endosymbiontisch in den Zellen *L. patella* (Seescheide) lebt
- Bei der Shotgun Sequenzierung werden unweigerlich beide Genome vermischt
- Wegen des sehr viel kleineren Bakteriengenoms (5Mbp vs. 160Mbp) ist dessen durchschnittliche Überdeckung (*coverage*) größer
- Jedes längere Fragment (*scaffold*) sollte somit *P. didemni* Sequenzen enthalten
- Fragmente die sich nicht mit anderen *read*s alignen lassen sind vermutlich von *L. patella*

Ergebnisse

- 82.337 Shotgun Fragmente; 36.920 Fragmente > 10Kbp *P. didemni*; 21.276 Fragmente *L. patella*; 24.141 uneindeutig
- Glimmer wurde auf mit beiden Fragment-Klassen trainiert $\Rightarrow$ 2 IMM Modelle
- 22% der uneindeutigen Fragmente als *P. didemni* erkannt
- Genauigkeitsverifikation über Partner Basen: Komplementärer Strang sollte gleiche Vorhersagen erzeugen
- Von 10.500 Basen-partner Fragmenten nur 207 (2%) inkonsistent $\Rightarrow$ 99% Genauigkeit

Zusammenfassung

- Glimmer ist ein Gene finder, der 97–98% aller Gene in einem prokaryontischen Genom ohne menschliche Einfluss findet
- Auch als eukaryontische Version verfügbar (s. Glimmer HMM)
- Durch gezielte Trainingsdaten auswahl können auch andere Aufgabenstellungen realisiert werden
- Online Version von Glimmer 3 verfügbar

Quellen


Salzberg SL, Delcher AL, Kasif S, White O (1998) Microbial gene identification using interpolated Markov models; Nucleic Acids Research (Vol. 26) 2, 544–548
http://www.cbcb.umd.edu/papers/glimmer-nar.pdf


Delcher AL, Harmon D, Kasif S, White O, Salzberg SL (1999) Improved microbial gene identification with GLIMMER; Nucleic Acids Research (Vol. 27) 23, 4636–4641
http://www.cbcb.umd.edu/papers/glimmer2.pdf


Delcher AL, Bratke KA, Powers EC, Salzberg SL (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer; Bioinformatics (Vol. 23) 6, 673–679


Böckenhauer H.-J., Bongartz D. (2003) Algorithmische Grundlagen der Bioinformatik – Modelle, Methoden und Komplexität; Teubner Verlag, 216–223

Quellen

Glimmer is OSI Certified Open Source and available at:

http://cbcb.umd.edu/software/glimmer

Vortragsfolien unter:

http://www.siebauer.com/bioinf/

Kontakt

Michael@Siebauer.com
mai03gxc@uni-leipzig.de

Feierabend!