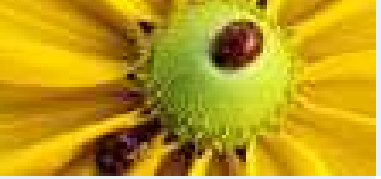
Conditional Random Fields für die Genvorhersage

Mario Stanke

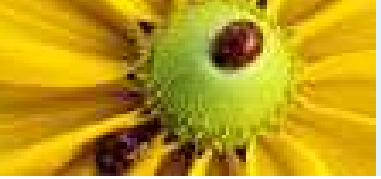
Department of Bioinformatics, University of Göttingen

mstanke@gwdg.de

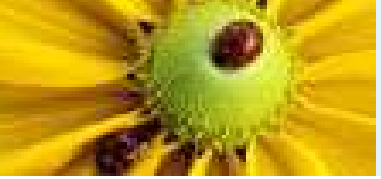


Überblick

1. biologischer Hintergrund: eukaryotische Gene
2. Semi-Markow CRFs
3. ein Semi-Markow CRF für die Genvorhersage
4. ein Online Large-Margin Training Algorithmus für CRFs
5. proteinfamilienbasierte Genvorhersage
6. ein zweidimensionales Semi-Conditional-Random-Field

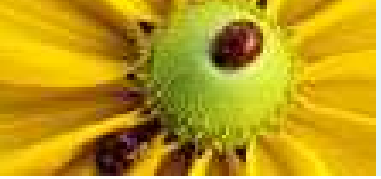


biologischer Hintergrund: eukaryotische Gene



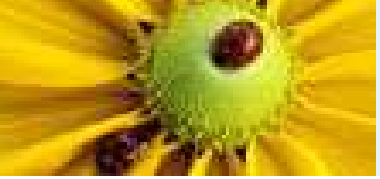
Das eukaryotische Protein-kodierende Gen

DNA ...actaatagacatctatttcgagtcaaggtgtaggcaatgtccttttttctagtcatggttggcaaacagtgggatcctgagagtcagataattgaattggctctgcctttaattatttgttcaagcaagcccctgtcccttaggtgggaatatgtatgagggaccatatttggggttctggtagctccacagggatgcggtgatgagcgctgaatttatgacgtactag...

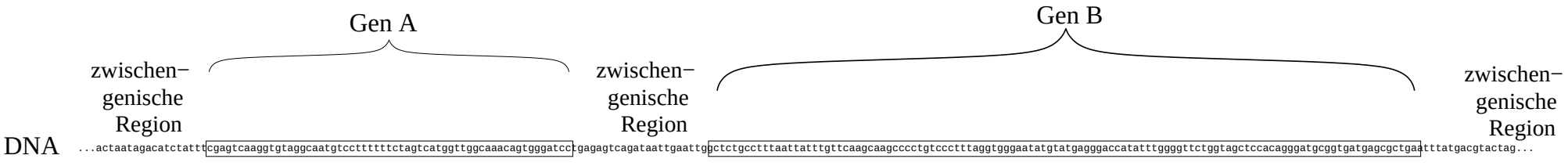


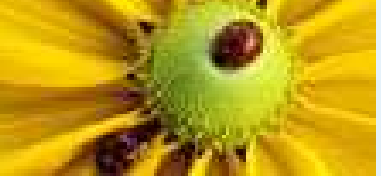
Das eukaryotische Protein-kodierende Gen

DNA ...actaatagacatctatttcgagtcaaggtgtaggcaatgtccttttttctagtcatggttggcaaacagtg gaattggc aattatttgttcaagcaagcccctgtcccttaggtgggaatatgtatgagggaccatatttggggttctggtagctccacagggatgcggtgatgagcgctgaatttatgacgtactag...

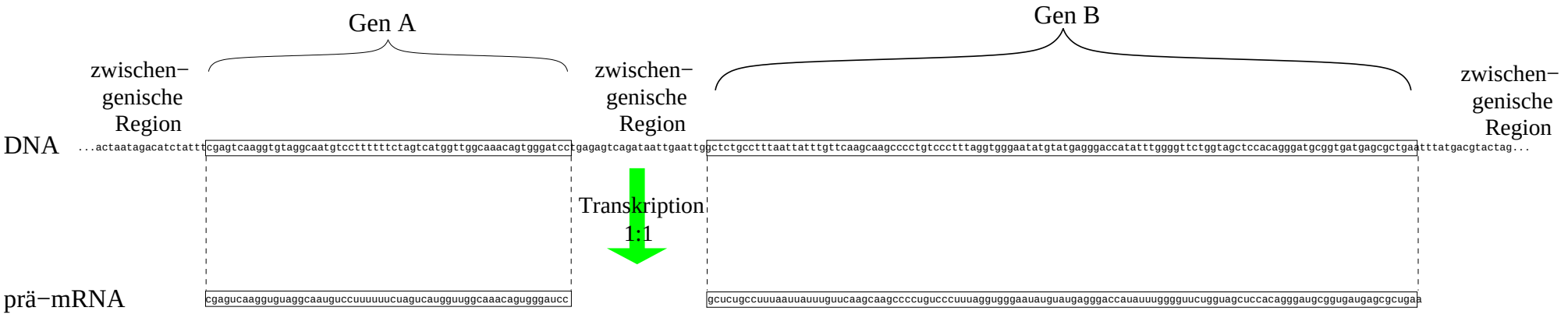


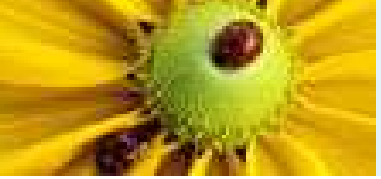
Das eukaryotische Protein-kodierende Gen



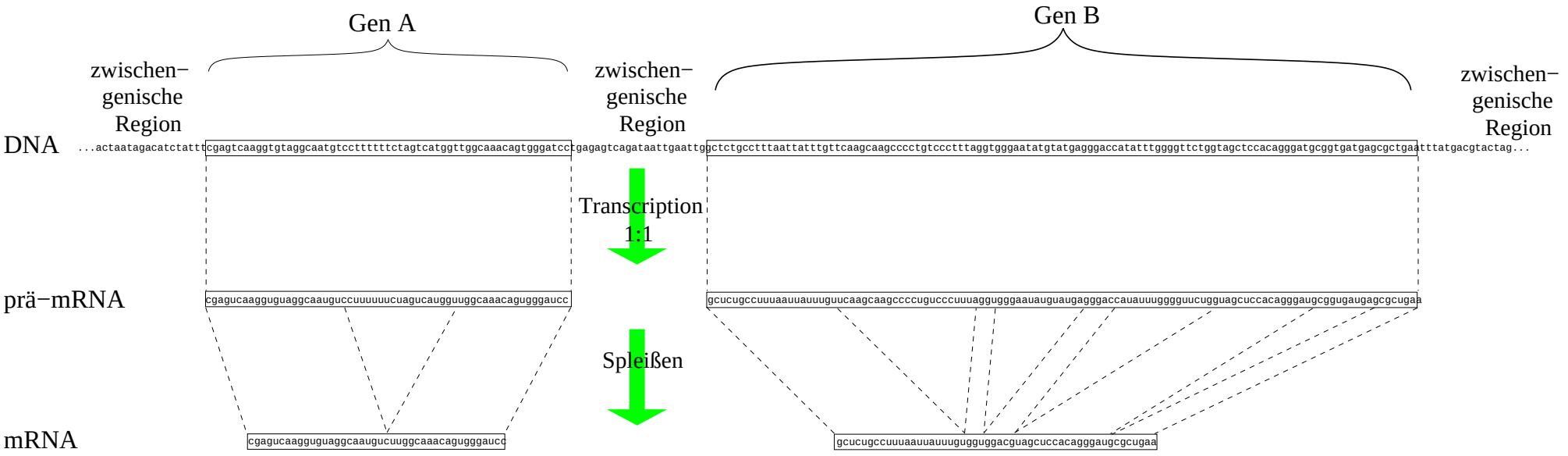


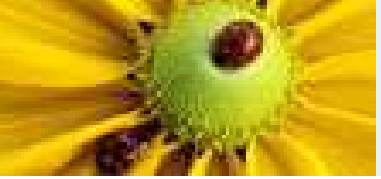
Das eukaryotische Protein-kodierende Gen



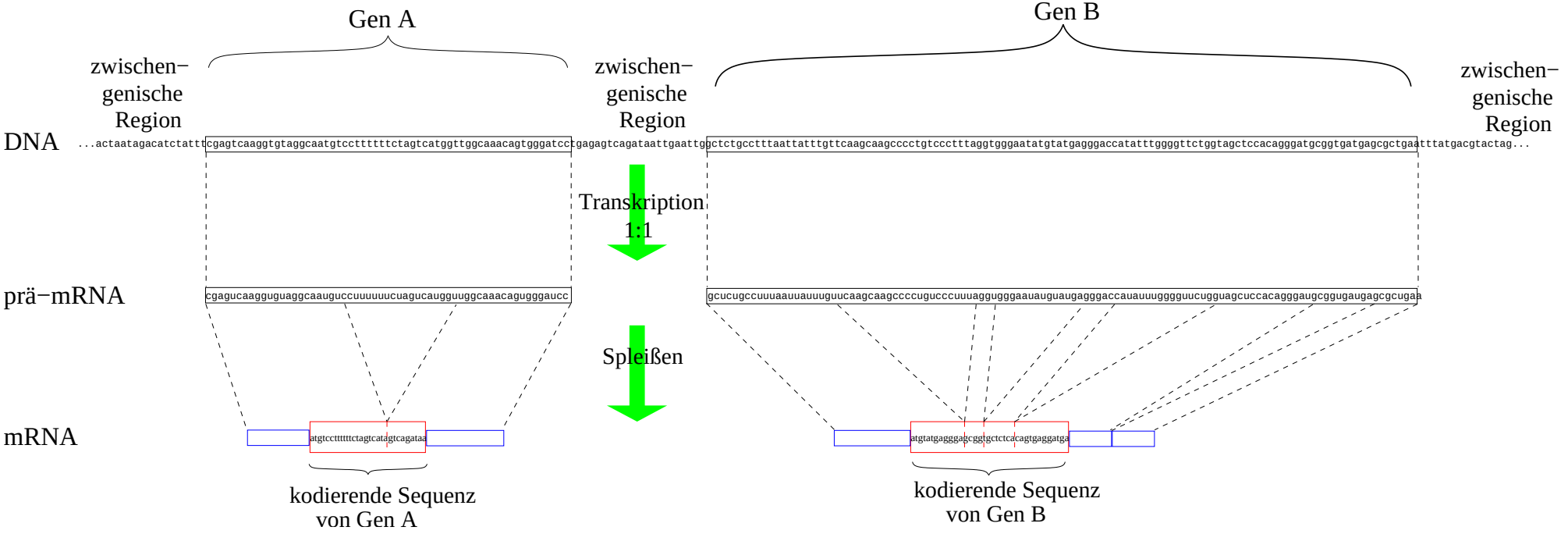


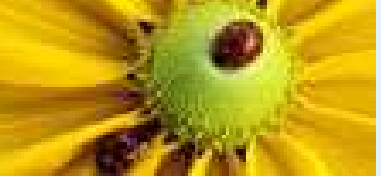
Das eukaryotische Protein-kodierende Gen



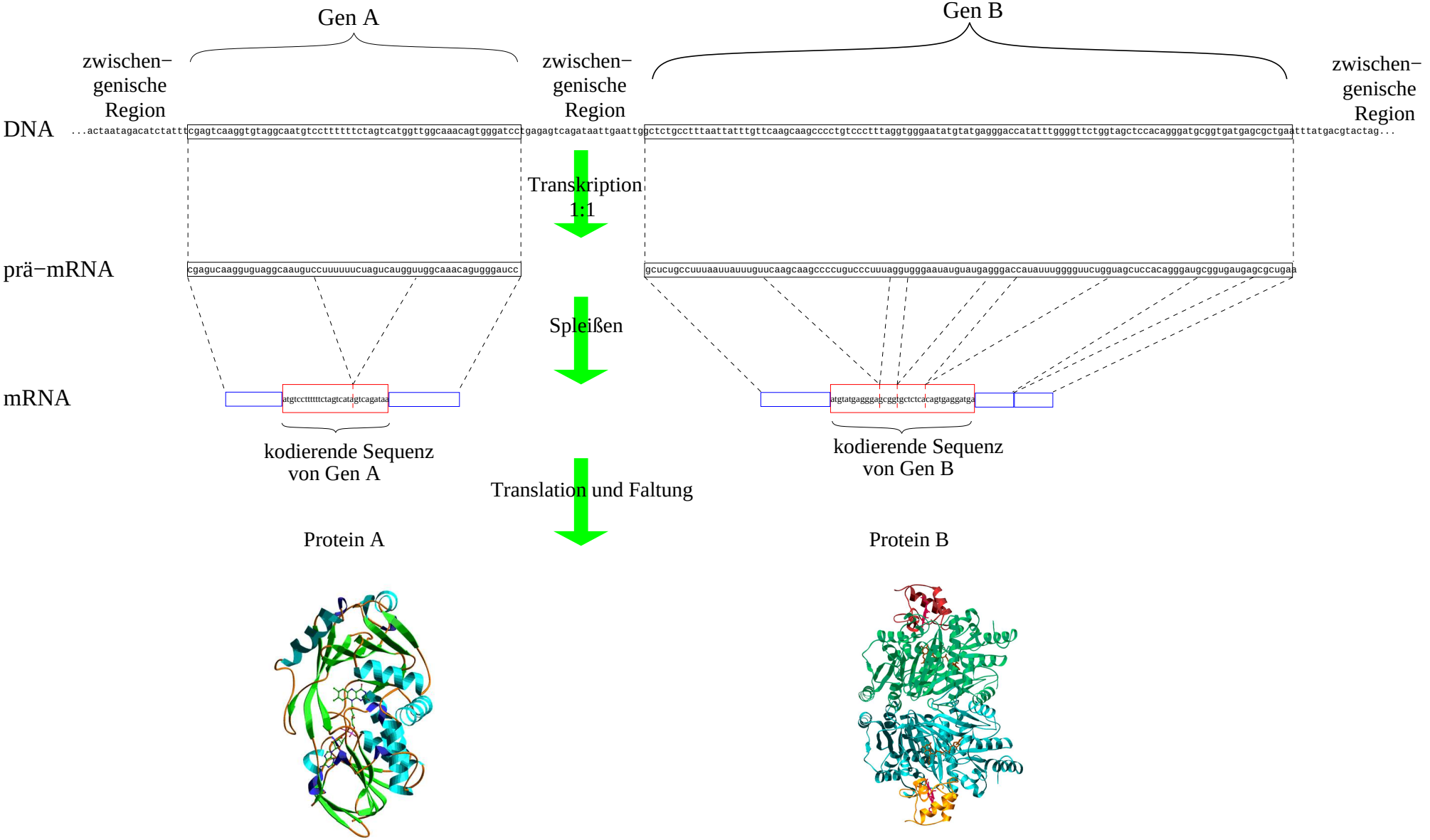


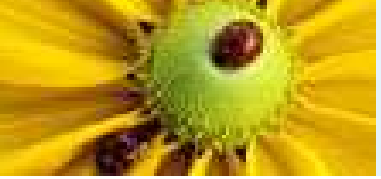
Das eukaryotische Protein-kodierende Gen



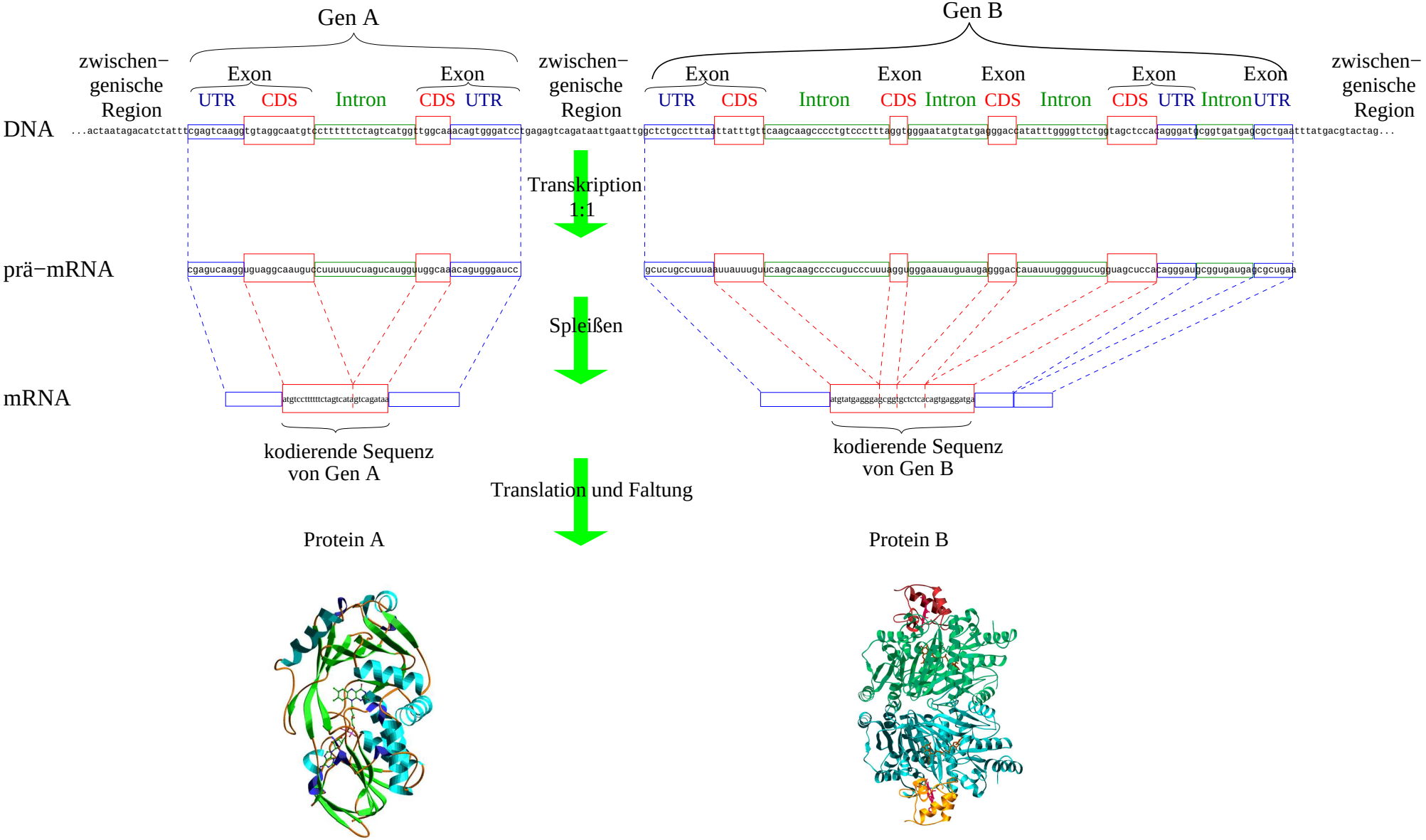


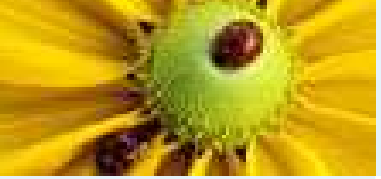
Das eukaryotische Protein-kodierende Gen



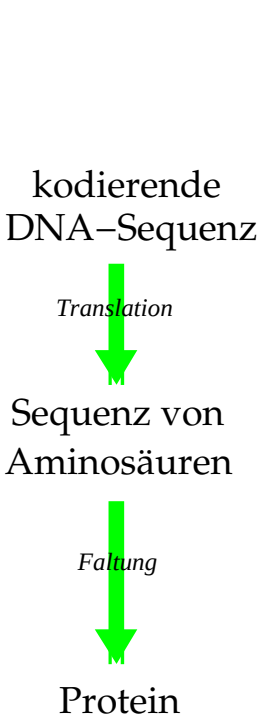


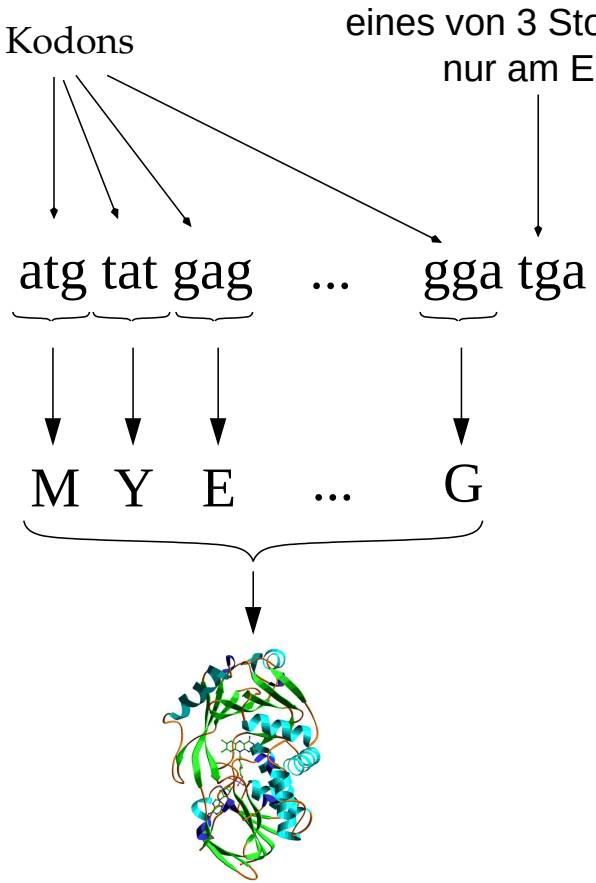
Das eukaryotische Protein-kodierende Gen





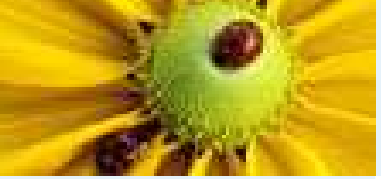
Translation



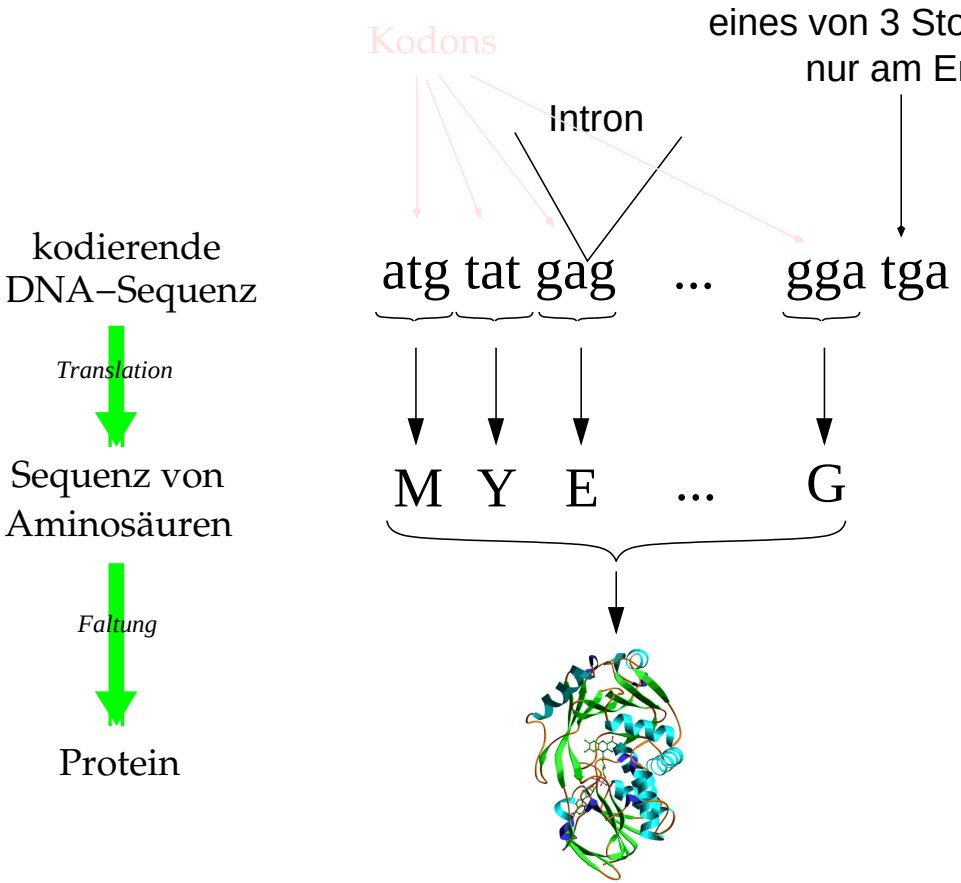


"universeller" genetischer Code

Kodon (DNA)		Amino-säure
aaa	↦	K
aac	↦	N
aag	↦	K
aat	↦	N
.		.
.		.
.		.
atg	↦	M
.		.
.		.
.		.
61 Kodons		20 Amino-säuren

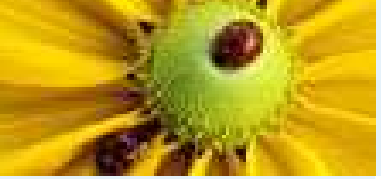


Translation

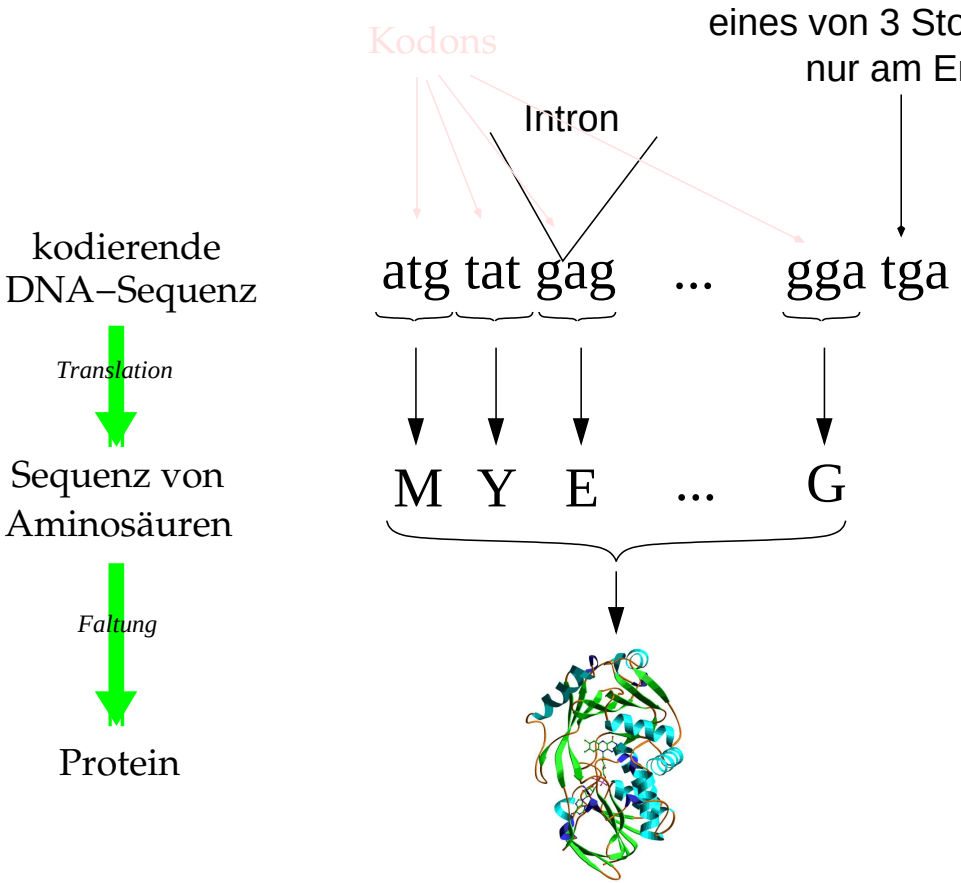


"universeller" genetischer Code

Kodon (DNA)		Amino-säure
aaa	↦	K
aac	↦	N
aag	↦	K
aat	↦	N
.		.
.		.
.		.
atg	↦	M
.		.
.		.
.		.
61 Kodons		20 Amino-säuren

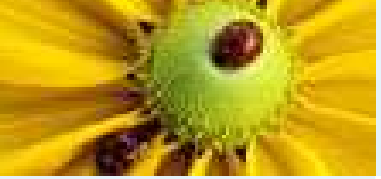


Translation

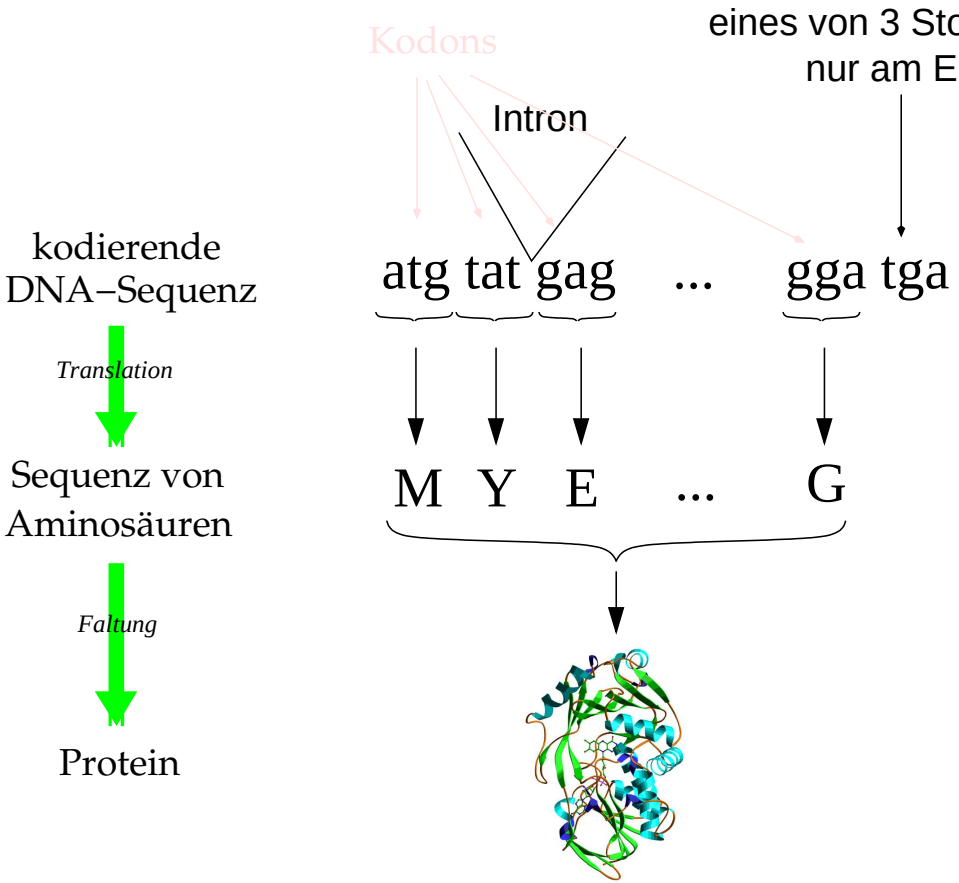


"universeller" genetischer Code

Kodon (DNA)		Amino-säure
aaa	↦	K
aac	↦	N
aag	↦	K
aat	↦	N
.		.
.		.
.		.
atg	↦	M
.		.
.		.
.		.
61 Kodons		20 Amino-säuren

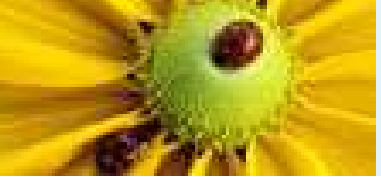


Translation



"universeller" genetischer Code

Kodon (DNA)		Amino-säure
aaa	↦	K
aac	↦	N
aag	↦	K
aat	↦	N
.		.
.		.
.		.
atg	↦	M
.		.
.		.
.		.
61 Kodons		20 Amino-säuren



Problemstellung, praktisch

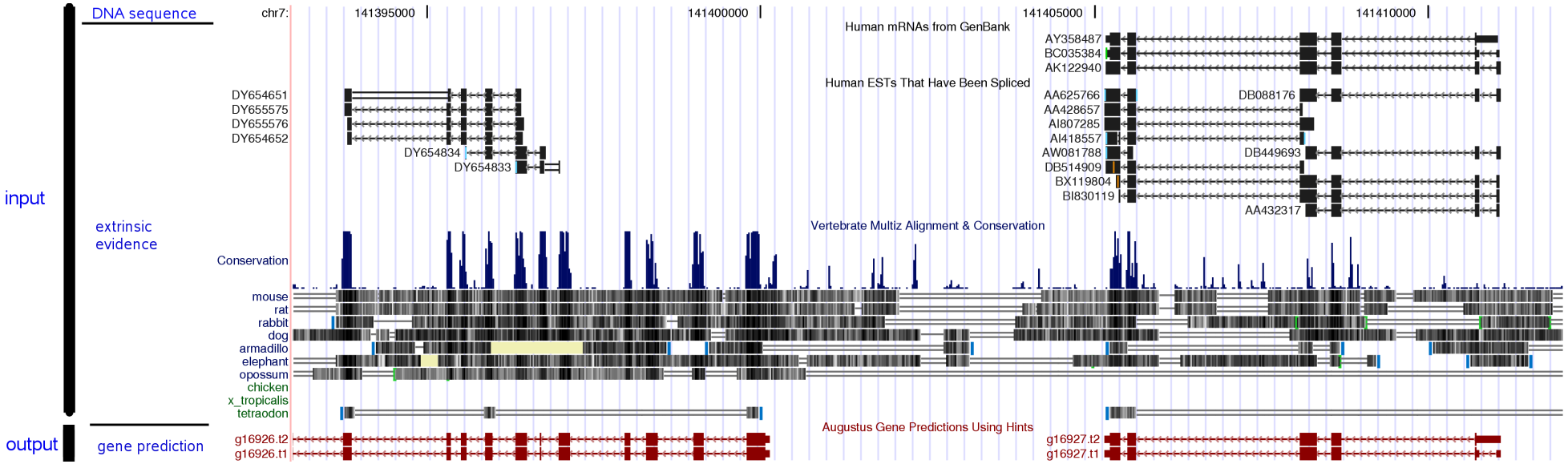
Eingabe:

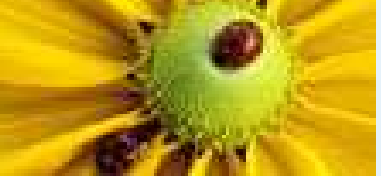
- Ziel-Genomsequenz, z.B. ein Chromosom
- extrinsische Hinweise, z.B. von
  - Stücken sequenzierter mRNA (ESTs) oder durch
  - Vergleich mit verwandten Genomen

Ausgabe:

- Start- und Endpositionen von Genen, kodierenden Regionen, Exons und Introns
- vorhergesagte Proteinsequenz

Beispiel: 19000 Basenpaare des menschlichen Chromosoms 7





Problemstellung, formal

Eingabe: Beobachtung $Y = (\sigma, H)$ oder $Y = \sigma$,
wobei σ die DNA-Sequenz der Länge ℓ ist und H die Menge der Hinweise.

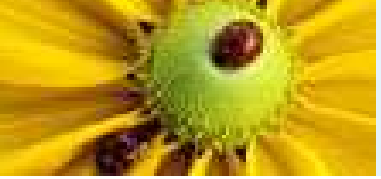
Ausgabe: Markierte (gelabelte) Segmentierung der Eingabe-DNA-Sequenz (Parse):

$$X = (X_1, X_2, \ldots, X_n),$$

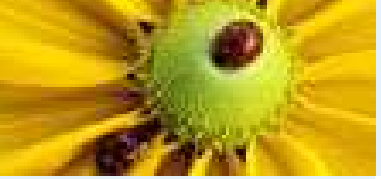
wobei $X_i = (q_i, u_i, v_i)$ das i-te Segment $(u_i, v_i]$ von σ ist und mit einer Markierung (Label) aus $q_i \in Q$ markiert ist.
$u_1 := 0 \leq v_1 = u_2 \leq v_2 \leq \cdots \leq v_n = \ell$.
Q ist hinreichend reichhaltig um Genstruktur eindeutig zu beschreiben, z.B.

$Q = \{\text{CDS}, \text{intergenische Region}, \text{Intron}, \text{UTRintron5}, \text{UTRintron3}, \text{UTR5}, \text{UTR3}\}$

(Tafel)



Semi-Markow CRFs



Bezeichungen

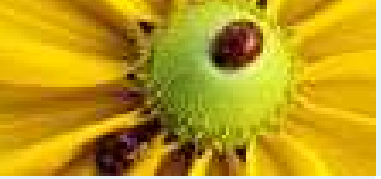
Im Folgenden bezeichne Y eine zufällige Beobachtung aus einer beliebigen Menge.
y sei eine tatsächlich beobachtete Realisierung von Y und ℓ sei eine (sich aus y ergebende) Länge (in unserer Anwendung die Länge der DNA-Eingabesequenz).
x bezeichne Parses der Länge n der folgenden Form:

$$x = ((q_1, u_1, v_1), \ldots, (q_t, u_t, v_n)),$$

wobei die Endpunkte der Segmente $(u_i, v_i]$ die Bedingung

$$u_1 := 0 \leq v_1 = u_2 \leq v_2 \leq \cdots \leq v_n = \ell$$

erfüllen.
$q_i \in Q$ sind Markierungen.
q_0 sei eine spezielle Startmarkierung zur Vereinfachung der Notation.
Markierungen und Label entsprechen den Zuständen beim HMM.



Semi-Markow CRF

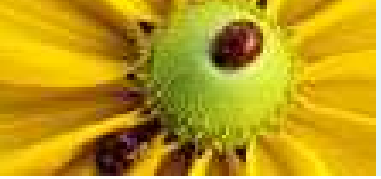
Definition: Ein Semi-Markow-Conditional-Random-Field (Semi-CRF) ist eine bedingte Verteilung von Parses X gegeben die Beobachtung Y der folgenden Form

$$P(X = x \mid Y = y) = \frac{1}{Z(y)} \exp \sum_{j=1}^{n} f(q_{j-1}, q_j, u_j, v_j, y). \qquad (1)$$

$f \in \mathbb{R} \cup \{-\infty\}$ heißt die Merkmalsfunktion (Featurefunktion) und

$$Z(y) = \sum_{x} \exp \sum_{j=1}^{n} f(q_{j-1}, q_j, u_j, v_j, y)$$

ist so gewählt, dass $P(X = x \mid Y = y)$ tatsächlich eine Wkeitsverteilung ist.



Unabhängigkeitseigenschaft Semi-CRF

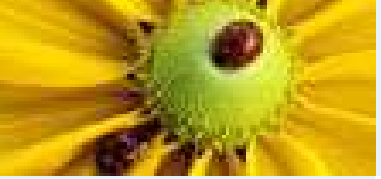
Behauptung:

$$P(X_i \mid X_{\setminus i}, Y) = P(X_i \mid X_{i-1}, X_{i+1}, Y) \tag{2}$$

Erläuterung: $X_{\setminus i}$ ist eine abkürzende Schreibw. für $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$.

Beweis (trivial): Sei y eine beliebige Beobachtung und sei x ein beliebiger Parse mit den Bezeichungen wie oben mit $P(x \mid y) > 0$.

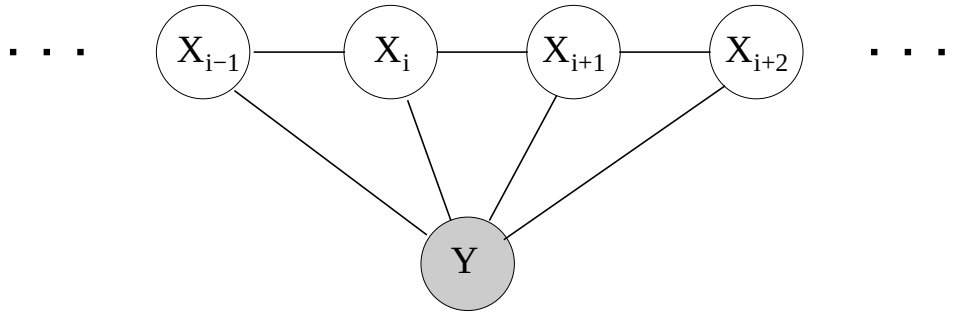
$$\begin{aligned}
& P(X_i = x_i \mid X_{\setminus i} = x_{\setminus i}, Y = y) \\
= \quad & \frac{P(X = x \mid Y = y)}{P(X_{\setminus i} = x_{\setminus i} \mid Y = y)} \\
= \quad & \frac{\prod_{j=1}^{n} \exp f(q_{j-1}, q_j, u_j, v_j, y)}{\prod\limits_{j \notin \{i, i+1\}} \exp f(..) \cdot \sum\limits_{q_i' \in Q} \exp f(q_{i-1}, q_i', u_i, v_i, y) \exp f(q_i', q_{i+1}, u_{i+1}, v_{i+1}, y)} \\
= \quad & \frac{\exp f(q_{i-1}, q_i, u_i, v_i, y) \exp f(q_i, q_{i+1}, u_{i+1}, v_{i+1}, y)}{\sum\limits_{q_i' \in Q} \exp f(q_{i-1}, q_i', u_i, v_i, y) \exp f(q_i', q_{i+1}, u_{i+1}, v_{i+1}, y)}
\end{aligned} \tag{3}$$



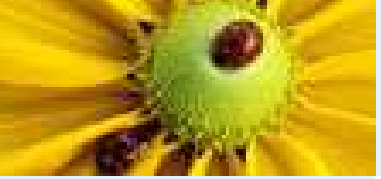
Unabhängigkeitseigenschaft Semi-CRF

Der Ausdruck 3 ist konstant in Bezug auf $x_{\setminus i-1,i,i+1}$, da diese Variablen gar nicht mehr in ihm vorkommen (rausgekürzt). Das bedeutet, da die x_i beliebig waren, dass X_i tatsächlich bedingt unabhängig von $X_1, \ldots, X_{i-2}, X_{i+2}, \ldots, X_n$ ist, gegeben die beiden Nachbarn X_{i-1}, X_{i+1} und die Beobachtung. □

Als graphisches Modell dargestellt hat unser Semi-CRF also folgende Abhängigkeitsstruktur:



Ein Semi-CRF ist deshalb ein CRF im Sinne der Originalliteratur von Lafferty, McCallum und Pereira (2001) mit einer linearen Kette als Graph G auf der Labelmenge $Q \times \mathbb{N} \times \mathbb{N}$ aller markierten Segmente.



Inferenz bei Semi-CRFs

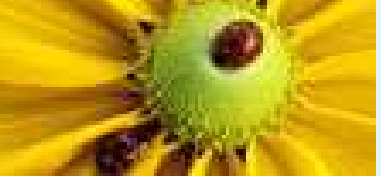
Die Verallgemeinerung auf das Markieren ganzer Segmente geschieht ganz analog wie die entsprechende Verallgemeinerung bei den GHMMs.

$$x^* \in \operatorname*{argmax}_x P(x \mid y)$$

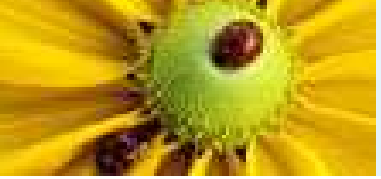
kann wieder mit folgender dynamischer Programmierung gefunden werden, die dem Viterbi-Algorithmus entspricht:

$$\gamma_{q,t} = \max_{\substack{q' \in Q, \\ 0 \leq t' \leq t}} \gamma_{q',t'} + f(q', q, t', t, y) \qquad (q \in Q, 1 \leq t \leq \ell) \quad (4)$$

Speicherplatz: $O(\ell \cdot |Q|)$
Laufzeit:$O(\ell \cdot |Q|^2 \cdot M)$
Hierbei sei
$M = \max \{t - t' \mid f(q', q, t, t', y) \neq -\infty, q, q' \in Q, 0 \leq t' \leq t \leq \ell\}$
wieder die maximale Länge eines Segments.



Ein Semi-Markow CRF für die Genvorhersage



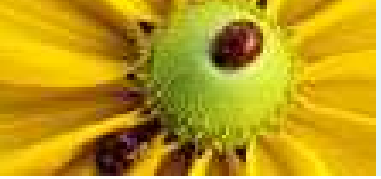
Merkmale aus vorhandenem GHMM

Unter der Merkmalsfunktion $f(q_{j-1}, q_j, u_j, v_j, y)$ kann man sich zunächst die Summe von lokalen Basismerkmalen vorstellen, die den Übergangs-und Emissionswahrscheinlichkeiten bei GHMMs entsprechen. Angenommen wir haben bereits ein GHMM trainiert mit Übergangswkeiten $a_{q',q}$ und Emissionswkeiten $e_q(\sigma(u_j, v_j])$.

Wählt man

$$f(q_{j-1}, q_j, u_j, v_j, y) := \log a_{q_{j-1},q_j} + \log e_{q_j}(\sigma(u_j, v_j]) \tag{5}$$

...

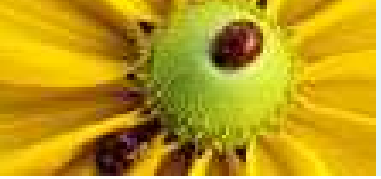


GHMM Spezialfall von Semi-CRF

... so bekommt man

$$
\begin{aligned}
P(x \mid y) &= \frac{1}{Z(y)} \exp \sum_{j=1}^{n} f(q_{j-1}, q_j, u_j, v_j, y) \\
&= \frac{1}{Z(y)} \prod_{j=1}^{n} \exp f(q_{j-1}, q_j, u_j, v_j, y) \\
&= \frac{1}{Z(y)} \prod_{j=1}^{n} a_{q',q} \cdot e_q(\sigma(u_j, v_j]) \\
&= \frac{1}{P_{\text{HMM}}(y)} \prod_{j=1}^{n} a_{q',q} \cdot e_q(\sigma(u_j, v_j]) \qquad (6) \\
&= P_{\text{HMM}}(x, y) / P_{\text{HMM}}(y) \\
&= P_{\text{HMM}}(x \mid y)
\end{aligned}
$$

6 gilt, da $Z(y) = \sum_x \prod_{j=1}^{n} a_{q',q} e_q(\sigma(u_j, v_j]) = P_{\text{HMM}}(y)$. Das bedeutet, das so gewählte Semi-CRF liefert exakt die gleiche Verteilung $P(x \mid y)$ wie das GHMM.



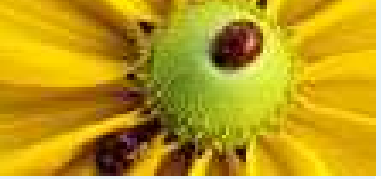
Semi-CRF: mehr Freiheit bei Parametern/Merkmalen

GHMM: Jede Übergangsverteilung aus jedem Zustand ist normiert:
$\sum_q a_{q',q} = 1 \quad \forall q'$
Jede Emissionsverteilung in jedem Zustand ist normiert:
$\sum_w e_q(w) = 1 \quad \forall q$

Semi-CRF: Mit $Z(y)$ nur einmal global normiert.

Die Merkmalsfunktion f gibt uns also wesentlich mehr Freiheiten. Könnten z.B.

- mit GHMM starten und Emissions- und Übergangswkeiten unter Ignorierung der Normierung variieren
- neue, allgemeinere Merkmale verwenden, z.B. dürfen “Emissionen” von der “Zukunft” abhängen



Merkmalsfunktion als gewichtete Summe

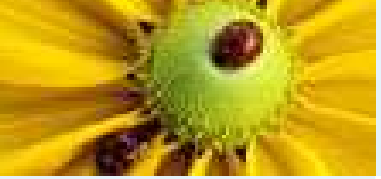
Eine typische Wahl für die Merkmalsfunktion ist folgende:

f wird als gewichtete Summe von lokalen Basismerkmalen dargestellt, d.h.

$$\begin{aligned} f(q_{j-1}, q_j, u_j, v_j, y) &= \langle \mathbf{w}, \mathbf{g}(q_{j-1}, q_j, u_j, v_j, y) \rangle \\ &= \sum_{i=1}^{k} w_i \cdot g_i(q_{j-1}, q_j, u_j, v_j, y) \end{aligned}$$

Dabei ist jede Komponente g_i des Vektors $\mathbf{g}$ ein lokales Basismerkmal. $\mathbf{w}$ ist ein Vektor von reellen Gewichten.
k ist die Anzahl der Basismerkmale.

Wir definieren $f(q_{j-1}, q_j, u_j, v_j, y) := -\infty$, falls $g_i = -\infty$ für ein i. Auf diese Weise können Parses ausgeschlossen werden.



Basismerkmale für die Genvorhersage: Übergänge

vereinfachtes Beispiel (Tafel)

Für jeden erlaubten Übergang $a \to b$ ($a, b \in Q$) verwende ein lokales Basismerkmal

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q'=a, q=b\}}$$

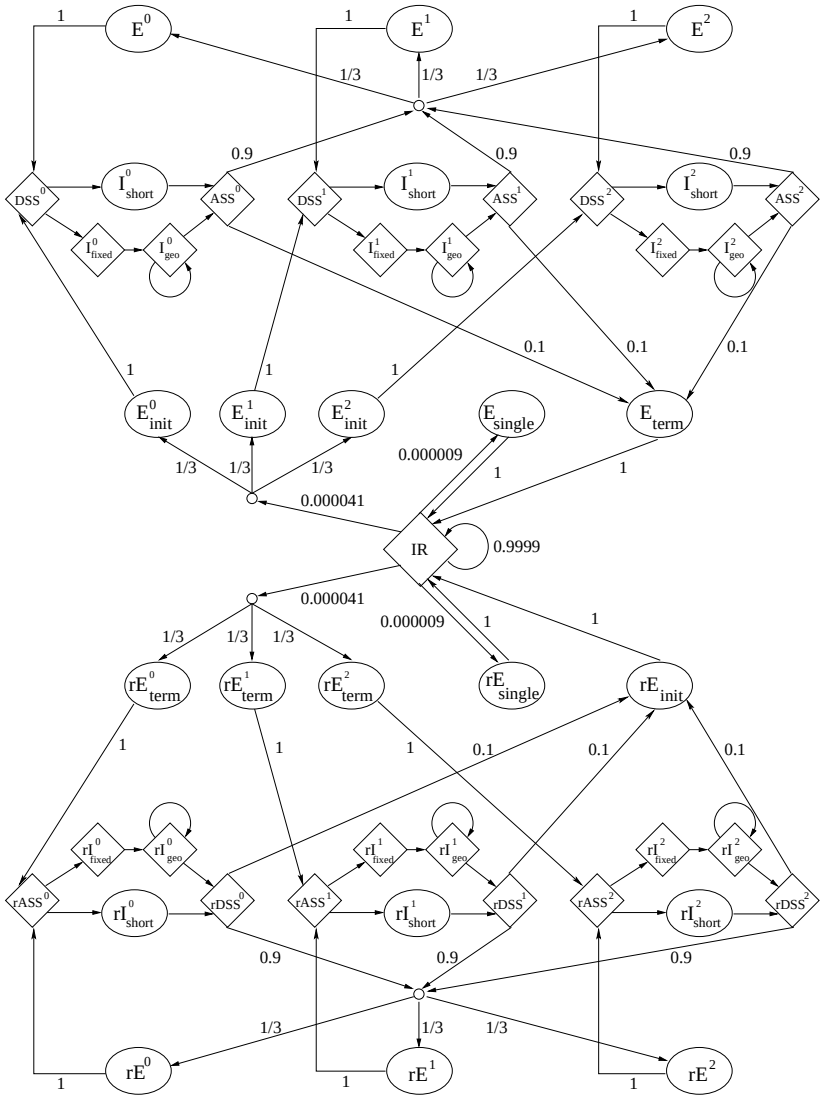
Verwende ein weiteres lokales Basismerkmal um unerlaubte Übergänge auszuschliessen:

$$g_i(q', q, u, v, y) = \begin{cases} -\infty & \text{, falls Übergang } q' \to q, \text{ nicht erlaubt} \\ 0 & \text{, sonst.} \end{cases}$$

Bemerkung: Der Index i an g steht hier nur um anzudeuten, dass es sich um eine (irgendeine) Komponente des Vektors handelt. Die Reihenfolge der Komponenten ist egal.

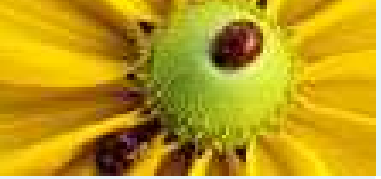
Basismerkmale für die Genvorhersage: Übergänge

realistisches Beispiel:



Labelmenge Q berücksichtigt:

- nur wenige Übergänge $q' \rightarrow q$ machen biologisch Sinn
- brauchen manchmal 3 Zustände um uns Leserahmen zu merken
- DNA Doppelstrang

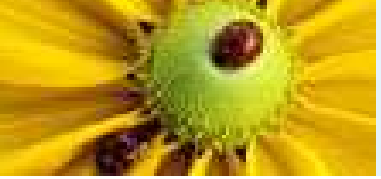


Basismerkmale für die Genvorhersage: Kompositionsmerkmale

Beispiel: Für jedes 5-mer $p \in \{a, c, g, t\}^5$ und jede Intronmarkierung I verwende ein Basismerkmal

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q=I\}} \cdot \sum_{j=u+1}^{v} \mathbf{1}_{\{p=\sigma[j,j+5)\}}$$

Ebenso für die Markierungen der zwischengenischen Region und der UTRs.



Basismerkmale für die Genvorhersage: Kompositionsmerkmale

In der proteinkodierenden Region eines Exons (CDS), wollen wir verwenden, dass die Musterhäufigkeiten abhängen von ihrer Position relativ zu den Kodongrenzen.

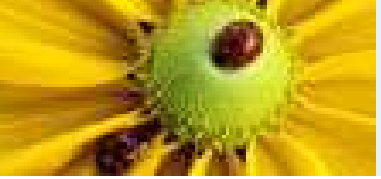
Für jedes 5-mer $p \in \{a, c, g, t\}^5$ und jede Exonmarkierung E_r im Leserahmen r verwende ein Basismerkmal

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q=E_r\}} \cdot \sum_{\substack{j=u+1 \\ j \equiv v-r \text{(mod 3)}}}^{v} \mathbf{1}_{\{p=\sigma[j,j+5)\}}$$

Um zu erzwingen, dass in der Kodonsequenz eines Exons keine Stoppkodons vorkommen ausser ganz am Ende, nehmen wir das Basismerkmal hinzu:

$$g_i(q', q, u, v, y)$$

$$= \begin{cases} -\infty & \text{, falls } \exists r \in \{0,1,2\}, j < v-3 : \sigma[j, j+3) \text{ ist Stoppkodon}, j \equiv v - r \text{(mod 3)} \\ 0 & \text{, sonst.} \end{cases}$$



Basismerkmale für die Genvorhersage: Längen der Segmente

Können auch die Längen der Segmente einer Markierung $a \in Q$ modellieren.
Z.B.

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q=a\}}(v-u)/\bar{\ell}_a$$

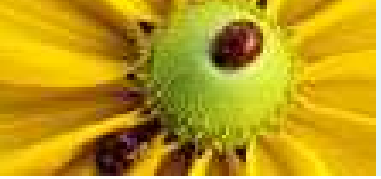
Konstante $\bar{\ell}_a$: geschätzte mittlere Länge einer Markierung a (zur Normierung).
Wenn das entsprechende Gewicht w_i negativ ist, entspricht dies einer geometrischen Verteilung beim HMM. (Warum?)

Können auch nichtparametrisch eine empirisch ermittelte Längenverteilung modellieren (Tafel):

Wählen eine "Eimergröße" d = Intervallbreite im Histogramm der Längen.
Haben ein Basismerkmal für jedes Label $a \in Q$ und jeden "Eimer"
$I_e := (ed, (e+1)d]$ $(e = 0, 1, \ldots, \lfloor M/d \rfloor)$ (engl.: *binning*)

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q=a, v-u \in I_e\}}$$

Die Gewichte w_i entsprechen dann den (logarithmierten) Balkenhöhen im Histogramm.



Basismerkmale für die Genvorhersage: extrinsische Hinweise

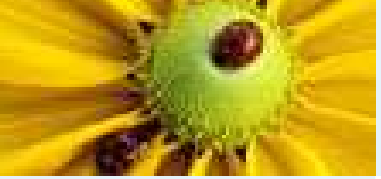
Sei $Y = (\sigma, H)$, wobei H extrinsiche Hinweise sind. Angenommen, H enthielte Hinweise über die wahrscheinliche Lage von Introns.

Bemerkung: Solche Hinweise können aus den Alignments von Stücken sequenzierter mRNA (sogenannte ESTs) mit dem Genom gewonnen werden. (Tafel)

Sei I die Menge der Intronmarkierungen. Verwende Basismerkmale

$$g_i(q', q, u, v, y) = \mathbf{1}_{\{q \in I, H \text{ enthält Hinweis auf Intron von } u+1 \text{ bis } v\}}$$

Entsprechend können andere Typen von Hinweisen (auf Exons, Transkriptionsstarts, etc.) berücksichtigt werden.



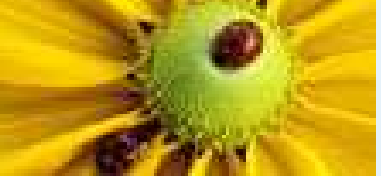
Basismerkmale für die Genvorhersage: externer Score

Manchmal existiert anstatt lokaler Basismerkmale ein externer Score s, z.B. von einem anderen Modell maschinellen Lernens. Z.B. wenn man eine Support Vector Machine hat um Spleißstellen in der DNA vorherzusagen. Spleißstellen heissen die Übergänge zwischen Exons und Introns. Wir nehmen an, dass

$$s = s(q', q, u, v, y) \in \mathbb{R}$$

beliebig ist. Wir nehmen an, dass s informativ dafür ist, ob der Parse korrekt ist, aber der Beitrag von s nicht unbedingt linear ist.

Wir wollen jetzt eine Menge von N Basismerkmalen konstruieren, die es erlauben, den Beitrag von s zu trainieren.



Basismerkmale für die Genvorhersage: externer Score

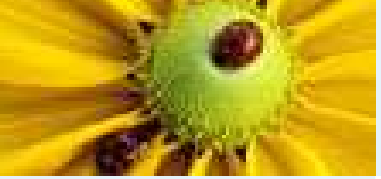
Seien $c_1 < c_2 < \cdots < c_N$ Stützstellen ($N \geq 1$). Eine sinnvolle Wahl wäre hier für c_i das $i/(N+1)$-Quantil der empirischen Verteilung von s zu nehmen. (Tafel)

Sei

$$\text{bin}(s) := \begin{cases} \min\{j \geq 0 \,|\, c_{j+1} > s\} & \text{, falls } c_N \geq s \\ N & \text{, sonst} \end{cases} .$$

Wir definieren Basismerkmale $g_1, g_2, \ldots, g_N$:

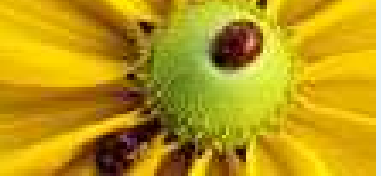
$$g_i(s) := \begin{cases} 1 & \text{, falls } i = 1, \text{bin}(s) = 0 \text{ oder } i = N, \text{bin}(s) = N \\ \frac{c_{i+1} - s}{c_{i+1} - c_i} & \text{, falls } \text{bin}(s) = i, 1 \leq i < N \\ \frac{s - c_{i-1}}{c_i - c_{i-1}} & \text{, falls } \text{bin}(s) = i-1, 1 < i <= N \\ 0 & \text{, sonst.} \end{cases}$$



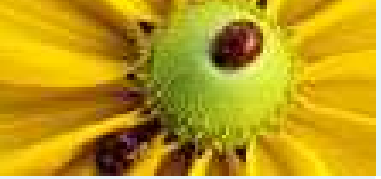
Basismerkmale für die Genvorhersage: externer Score

(Tafel)

Der Beitrag $\sum_{i=1}^{N} w_i g_i$ von s zur Merkmalsfunktion f ist dann eine stetige, in s stückweise lineare Funktion mit den Werten w_i an den Stützstellen c_i. Die w_i müssen gelernt werden – evtl. unter der Monotonienebenbedingung, dass $w_i \leq w_{i+1}$.



Online Large-Margin Training Algorithmus für CRFs



Training

Eingabe: Haben m Paare aus Beobachtung und bekanntem dazugehörigen Parse:

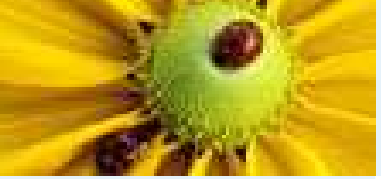
$$\{(x^{(t)}, y^{(t)})\}_{i=1}^{m} \qquad \text{(Trainingsmenge).}$$

Unabhängige Stichprobe aus der Verteilung von (X, Y).

Gesucht:

1. eine geeignete Form für die Merkmalsfunktion f (Basismerkmale g_i auswählen)
   Die Merkmalsfunktion (die g_i) wird für gewöhnlich manuell gewählt unter Berücksichtigung anwendungsspezifischen Wissens.
2. die Parameter von f (Gewichtsvektor $\mathbf{w}$)
   Die Parameter von f (die Gewichte $\mathbf{w}$) werden automatisch *trainiert*, d.h. gemäss eines Algorithmus aus den Trainingsdaten geschätzt.

Nehmen an, dass Basismerkmale $\mathbf{g}$ bereits gewählt sind und wollen nun $\mathbf{w}$ wählen.

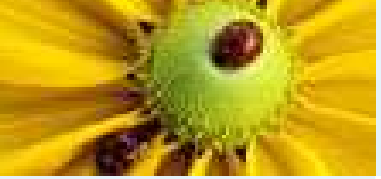


Online Large-Margin Training Algorithmus

Stelle nun Trainingsalgorithmus aus

Bernal, Crammer, Hatzigeorgiou, Pereira, "Global Discriminative Learning for Higher Accuracy Computational Gene Prediction", PLOS Computational Biolog (2007)

vor. Diese Autoren erzielen mit ihrem Algorithmus momentan die besten Ergebnisse bei der *ab initio* Genvorhersage beim Menschen.



Scorefunktion

Der Viterbi-Algorithmus maximiert $P(x \mid y)$ bzgl. x. Da $Z(y)$ konstant und die Exponentialfunktion monoton ist, ist dies äquivalent dazu, folgende Scorefunktion zu maximieren

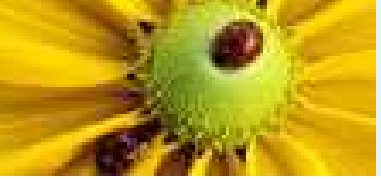
$$S_w(x, y) := \sum_{j=1}^{n} \langle \mathbf{w}, \mathbf{g}(q_{j-1}, q_j, u_j, v_j, y) \rangle \tag{7}$$

Wollen nun $\mathbf{w}$ finden, so dass bei den Trainingsdaten der Viterbi-Parse der i-ten Beobachtung

$$x^* \in \operatorname{argmax} S_w(x, y^{(i)})$$

“nah dran” ist am richtigen Parse $x^{(i)}$, ($i = 1, 2, \ldots, m$).

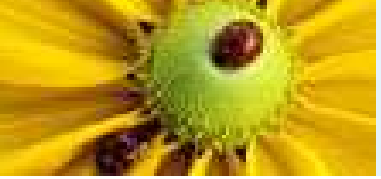
Definieren Verlustfunktion $L(x^*, x^{(i)}) \geq 0$ um den Abstand bzw. Fehler zu messen. $L = 0$, falls $x^* = x^{(i)}$ und L ist umso grösser je weiter der vorhergesagte Parse vom richtigen ist (z.B. L=1-Korrelationskoeffizient).



Online Large-Margin Training Algorithmus

1) initialisiere: $\mathbf{w} = \mathbf{0}$, $\mathbf{v} = \mathbf{0}$
2) wiederhole N Runden:
3) für $i = 1..m$ führe aus:
4) wähle $x^* \in \text{argmax}_x \, S_w(x, y^{(i)})$
5) $\hat{\mathbf{w}} \leftarrow \text{argmin}_{\mathbf{w}'} \, ||\mathbf{w}' - \mathbf{w}||_2$ unter der Nebenbedingung:
$S_{w'}(x^{(i)}, y^{(i)}) - S_{w'}(x^*, y^{(i)}) \geq L(x^*, y^{(i)})$
6) $\mathbf{w} \leftarrow \hat{\mathbf{w}}$
7) $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{w}$
8) $\mathbf{w} \leftarrow \mathbf{v}/(Nm)$

Erläuterungen: 5) Falls die Vorhersage beim i-ten Trainingsbeispiel falsch ist, wird $\mathbf{w}$ so verändert, dass mit dem neuen Gewichtsvektor $\hat{\mathbf{w}}$ der richtige Parse $x^{(i)}$ einen höheren Score (mit Abstand L) hat als der alte Viterbi-Parse x^*.
$||\mathbf{w}' - \mathbf{w}||_2 = \langle \mathbf{w}' - \mathbf{w}, \mathbf{w}' - \mathbf{w} \rangle$
8) Am Ende wird der Durchschnitt aller Gewichtsvektoren verwendet, um die Gefahr von Overfitting zu reduzieren.



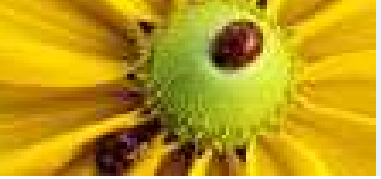
Online Large-Margin Training Algorithmus

Lösen nun das Optimierungsproblem von Zeile 5).

$$\text{OP}: \begin{cases} ||\mathbf{w}' - \mathbf{w}||_2 \rightarrow \min \\ \text{unter NB: } S_{w'}(x^{(i)}, y^{(i)}) - S_{w'}(x^*, y^{(i)}) \geq L(x^*, y^{(i)}) \end{cases}$$

Seien nun

$$\begin{aligned} \mathbf{h} &:= \mathbf{w}' - \mathbf{w}, \\ x^{(i)} &= ((q_1, u_1, v_1), \ldots, (q_n, u_n, v_n)) \\ x^* &= ((q_1^*, u_1^*, v_1^*), \ldots, (q_n^*, u_n^*, v_{n^*}^*)) \qquad \text{(Viterbi-Parse)} \\ L &:= L(x^*, y^{(i)}) \\ s_j &:= (q_{j-1}, q_j, u_j, v_j, y^{(i)}) \\ s_j^* &:= (q_{j-1}^*, q_j^*, u_j^*, v_j^*, y^{(i)}) \end{aligned}$$



Online Large-Margin Training Algorithmus

$$\text{OP}: \begin{cases} ||\mathbf{h}||_2 \rightarrow \min \\ \text{unter NB:} \quad \sum_{j=1}^{n} \langle \mathbf{w}+\mathbf{h}, \mathbf{g}(s_j) \rangle - \sum_{j=1}^{n^*} \langle \mathbf{w}+\mathbf{h}, \mathbf{g}(s_j^*) \rangle \geq L \end{cases}$$

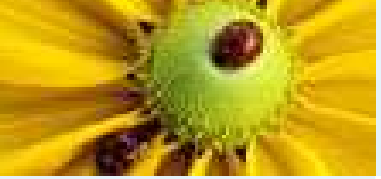
$$\Leftrightarrow \begin{cases} ||\mathbf{h}||_2 \rightarrow \min \\ \text{unter NB:} \quad \sum_{j=1}^{n} \langle \mathbf{h}, \mathbf{g}(s_j) \rangle - \sum_{j=1}^{n^*} \langle \mathbf{h}, \mathbf{g}(s_j^*) \rangle \\ \qquad + \sum_{j=1}^{n} \langle \mathbf{w}, \mathbf{g}(s_j) \rangle - \sum_{j=1}^{n^*} \langle \mathbf{w}, \mathbf{g}(s_j^*) \rangle \geq L \end{cases}$$

$$\Leftrightarrow \begin{cases} ||\mathbf{h}||_2 \rightarrow \min \\ \text{unter NB:} \quad \langle \mathbf{h}, \mathbf{a} \rangle \geq b, \end{cases} \tag{8}$$

wobei

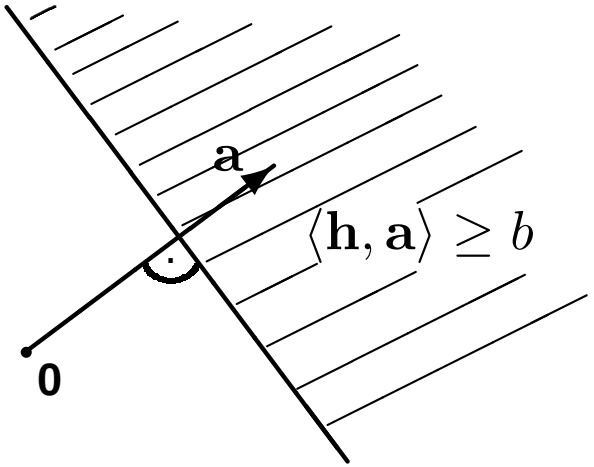
$$\mathbf{a} := \sum_{j=1}^{n} \mathbf{g}(s_j) - \sum_{j=1}^{n^*} \mathbf{g}(s_j^*)$$

$$\begin{aligned} b &:= L - \sum_{j=1}^{n} \langle \mathbf{w}, \mathbf{g}(s_j) \rangle + \sum_{j=1}^{n^*} \langle \mathbf{w}, \mathbf{g}(s_j^*) \rangle \\ &= L - S_w(x^{(i)}, y^{(i)}) + S_w(x^*, y^{(i)}) \geq 0 \end{aligned}$$



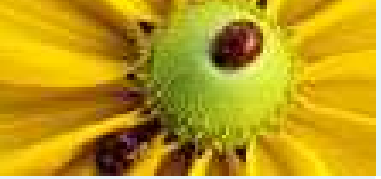
Online Large-Margin Training Algorithmus

8 ist jetzt in einer einfach zu lösenden Form. Wir suchen in dem Halbraum $\langle \mathbf{h}, \mathbf{a} \rangle \geq b$ den Punkt, der am nächsten zum Ursprung ist.



Lösung von OP:

$$\hat{\mathbf{h}} = \frac{b}{\langle \mathbf{a}, \mathbf{a} \rangle} \cdot \mathbf{a} \tag{9}$$



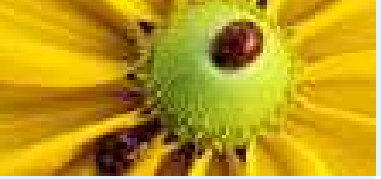
Interpretation

Angenommen in einem Schritt war der Viterbi-Parse falsch und $L > 0$. Der Gewichtsvektor $\mathbf{w}$ wird dann in Richtung $\mathbf{a}$ verändert. Und zwar soweit, dass gerade

$$S_{\hat{w}}(x^{(i)}, y^{(i)}) - S_{\hat{w}}(x^*, y^{(i)}) = L(x^*, y^{(i)})$$

gilt.
Dann ist also unter dem neuen Gewichtsvektor $\hat{\mathbf{w}}$ der Score des richtigen Parses $x^{(i)}$ um L grösser als der Score des Viterbi-Parses x^*. Für dieses Trainingsbeispiel würde also mit dem neuen Gewichtsvektor nicht derselbe Fehler gemacht.

Wiederholte man auf dem i-ten Trainingsbeispiel die Viterbi-Vorhersage mit $\hat{\mathbf{w}}$, so würde ein anderer Parse als x^* vorhergesagt. Es ist naheliegend – aber nicht notwendig – dass dann $x^{(i)}$ den grössten Score hat, also dass die Vorhersage danach richtig ist.



Interpretation von $\mathbf{h}$

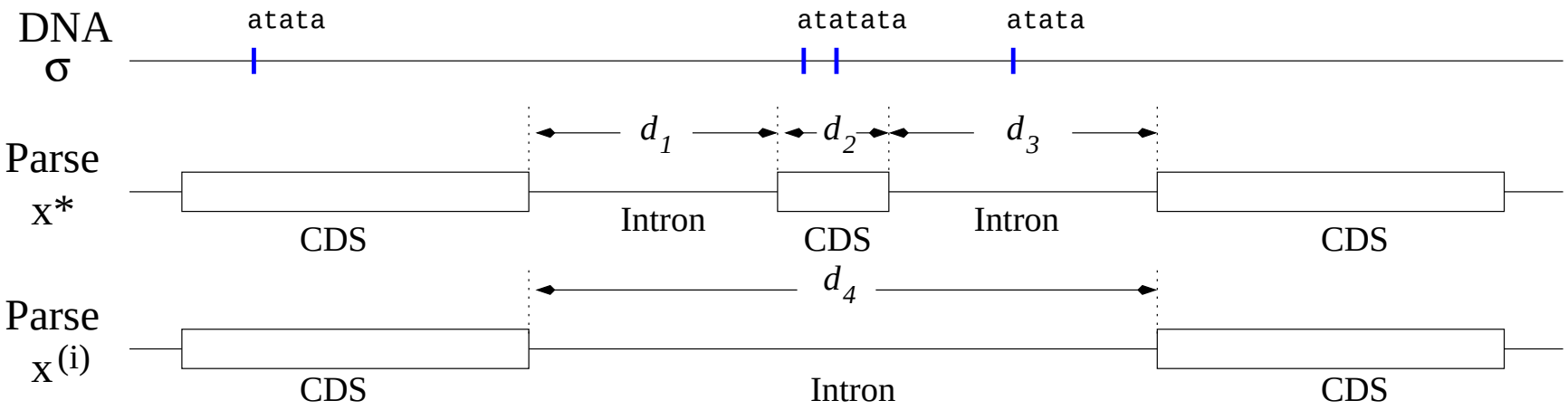
Die Veränderung $\mathbf{h}$ geht in Richtung von

$$\begin{aligned}\mathbf{a} &= \sum_{j=1}^{n} \mathbf{g}(s_j) - \sum_{j=1}^{n^*} \mathbf{g}(s_j^*) \\ &= \sum_{j=1}^{n} \mathbf{g}(q_{j-1}, q_j, u_j, v_j, y^{(i)}) - \sum_{j=1}^{n^*} \mathbf{g}(q_{j-1}^*, q_j^*, u_j^*, v_j^*, y^{(i)})\end{aligned}$$

Das bedeutet, dass die Gewichte w_i von solchen Basismerkmalen g_i erhöht werden, bei denen g_i beim richtigen Pfad grösser ist als beim Viterbi-Pfad: $h_i, a_i > 0$

Umgekehrt werden die Gewichte w_i von solchen Basismerkmalen g_i verkleinert, bei denen g_i beim richtigen Pfad kleiner ist als beim Viterbi-Pfad. $h_i, a_i < 0$

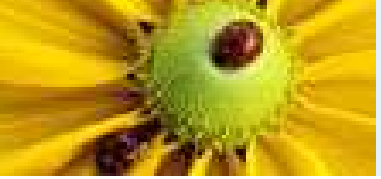
Interpretation von h, Beispiel



Erinnerung:

x^* Viterbiparse
$x^{(i)}$ korrekter Parse zum i-ten Trainingsbeispiel

In obigem Beispiel ist also ein zusätzliches, falsches, kleines Exon vorhergesagt worden.



Interpretation von h, Beispiel

Betrachten jetzt exemplarisch nur folgende Basismerkmale $g_i = g_i(q', q, u, v, y)$

$$g_1 := \text{Anzahl der Pentamere } \textit{atata} \text{ im Intron} = \mathbf{1}_{\{q \text{ ist Intron}\}} \sum_{j=u+1}^{v} \mathbf{1}_{\{\sigma[j,j+5)=atata\}}$$

$$g_2 := \text{Anzahl der Pentamere } \textit{atata} \text{ in CDS} = \mathbf{1}_{\{q \text{ ist CDS}\}} \sum_{j=u+1}^{v} \mathbf{1}_{\{\sigma[j,j+5)=atata\}}$$

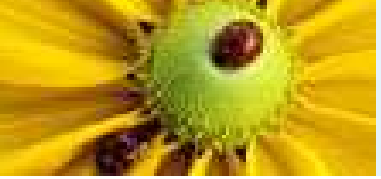
$$g_3 := \text{"Intronlänge} \in I_1\text{"} = \mathbf{1}_{\{q \text{ ist Intron}, v-u \in I_1\}}$$

$$g_4 := \text{"Intronlänge} \in I_3\text{"} = \mathbf{1}_{\{q \text{ ist Intron}, v-u \in I_3\}}$$

$$g_5 := \text{"Intronlänge} \in I_4\text{"} = \mathbf{1}_{\{q \text{ ist Intron}, v-u \in I_4\}}$$

$$g_6 := \text{"Exonlänge} \in I_2\text{"} = \mathbf{1}_{\{q \text{ ist CDS}, v-u \in I_2\}}$$

Dabei seien I_1, I_2, I_3, I_4 Intervalle (Bins) der Längenverteilungen, so dass $d_j \in I_j, \quad (j = 1, 2, 3, 4)$.



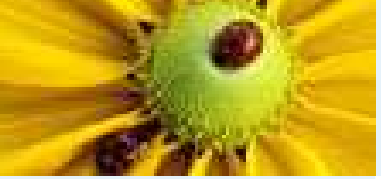
Interpretation von $\mathbf{h}$, Beispiel

In obigen Fall erhalten wir

$$\mathbf{a} = \begin{pmatrix} +2 \\ -2 \\ -1 \\ -1 \\ +1 \\ -1 \\ \vdots \end{pmatrix}$$

Da $\mathbf{h} = \hat{\mathbf{w}} - \mathbf{w}$ proportional zu $\mathbf{a}$ ist werden also die Gewichte der Basismerkmale g_1 und g_5 erhöht und die Gewichte der Basismerkmale g_2, g_3, g_4 und g_6 erniedrigt.

Basismerkmale, die bei beiden Parses gleich sind, werden nicht verändert. Z.B. die Kompositionsmerkmale zu allen Pentameren, die nicht in der Unterschiedsregion vorkommen.



Laufzeit

Laufzeit des Online Large-Margin Training Algorithmus:

Schritte 4) und 5) dominieren die Laufzeit und werden jeweils Nm mal ausgeführt.

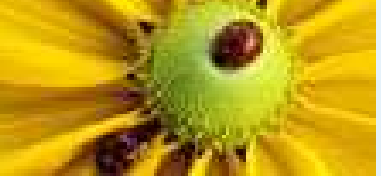
Schritt 4) ist die Durchführung des Viterbi-Algorithmus und benötigt $O(\ell \cdot |Q|^2 \cdot M)$ Zeit. (Annahme dabei: Merkmalsfkt. f kann ammortisiert in konstanter Zeit berechnet werden.)

Schritt 5) erfordert das Berechnen von b und $\mathbf{a}$.

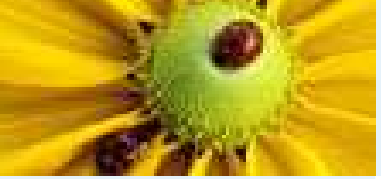
b: Es muss die Lossfunktion und zweimal der Score eines Parses berechnet werden: $O(\ell)$.

$\mathbf{a}$: $O(\ell)$, denn $\mathbf{a}$ kann durch setzen von $w_i = 1, (i = 1, \ldots, k)$ und zweimaliges Berechnen eines Scores berechnet werden. Annahme: $\langle \mathbf{w}, g(q', q, u, v, y) \rangle$ kann in Zeit $O(v - u)$ berechnet werden.

Insgesamt: $O(N \cdot m \cdot \ell \cdot |Q|^2 \cdot M)$



proteinfamilienbasierte Genvorhersage

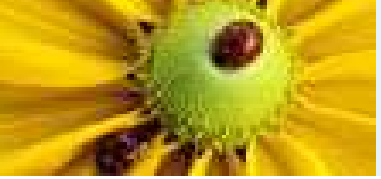


proteinfamilienbasierte Genvorhersage

Proteinfamilien: verwandte Gene mit ähnlicher Funktion und Sequenz

Beispiel: Myosine

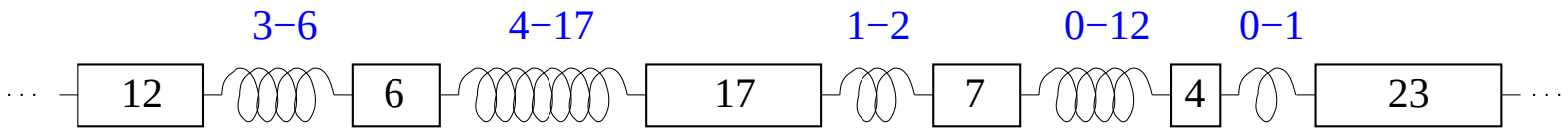
```
HsMhc1_fl       ITGESGAGKTVNTKRVIQYFATIAVTGEKKKEEVTSGKMQGTLEDQIISANPLLEAFGNA
HsMhc2_fl       ITGESGAGKTVNTKRVIQYFATIAVTGEKKKEEITSGKIQGTLEDQIISANPLLEAFGNA
HsMhc4_fl       ITGESGAGKTVNTKRVIQYFATIAVTGEKKKEEPASGKMQGTLEDQIISANPLLEAFGNA
HsMhc8_fl       ITGESGAGKTVNTKRVIQYFATIAVTGEKKKDE--SGKMQGTLEDQIISANPLLEAFGNA
HsMhc3_fl       ITGESGAGKTVNTKRVIQYFATIAATGDLAKKK--DSKMKGTLEDQIISANPLLEAFGNA
HsMhc13_fl      ITGESGAGKTVNTKRVIQYFATIAVTGDKKKETQ-PGKMQGTLEDQIIQANPLLEAFGNA
HsMhc6_fl       ITGESGAGKTVNTKRVIQYFASIAAIGD-RGKKDNANANKGTLEDQIIQANPALEAFGNA
HsMhc7_fl       ITGESGAGKTVNTKRVIQYFAVIAAIGD-RSKKDQS-PGKGTLEDQIIQANPALEAFGNA
HsMhc14_fl      ITGESGAGKTVNTKRVIQYFAIVAALGD------GPGKKAGTLEDQIIEANPAMEAFGNA
HsMhc15_fl      FTGESGAGKTVNSKHIIQYFATIAAMIE-------SRKKQGALEDQIMQANTILEAFGNA
HsMhc20_fl      ITGESGAGKTENTKKVIQYFANIGGTGK------QTTDKKGSLEDQVIQANPVLEAFGNA
HsMhc9_fl       CTGESGAGKTENTKKVIQYLAYVASSHK----SKKDQ---GELERQLLQANPILEAFGNA
HsMhc10_fl      CTGESGAGKTENTKKVIQYLAHVASSHK----GRKDHNIPGELERQLLQANPILESFGNA
HsMhc11_fl      CTGESGAGKTENTKKVIQYLAVVASSHK----GKKDTSITGELEKQLLQANPILEAFGNA
HsMhc16_fl      CTGESGAGKTENTKKVIQYLAHVASSPK----GRKEPGVPGELERQLLQANPILEAFGNA
HsMyo1E_fl      ISGESGAGKTVAAKYIMSYISRVSGGGT----------KVQHVKDIILQSNPLLEAFGNA
HsMyo1F_fl      ISGESGAGKTVAAKYIMGYISKVSGGGE----------KVQHVKDIILQSNPLLEAFGNA
HsMyo7A_fl      ISGESGAGKTESTKLILQFLAAISG------------QHSWIEQQVLEATPILEAFGNA
HsMyo7B_fl      ISGESGAGKTETTKLILQFLATISG------------QHSWIEQQVLEANPILEAFGNA
HsMyo1G_fl      ISGESGAGKTEASKHIMQYIAAVTNPSQ--------RAEVERVKDVLLKSTCVLEAFGNA
HsMyo1D_fl      ISGESGAGKTEASKYIMQYIAAITNPSQ--------RAEVERVKNMLLKSNCVLEAFGNA
HsMyo1C_fl      ISGESGAGKTEATKRLLQFYAETCPAPE----------RGGAVRDRLLQSNPVLEAFGNA
HsMyo1H_fl      ISGESGAGKTEASKKILEYFAVTCPMTQ----------SLQIARDRLLFSNPVLEAFGNA
```

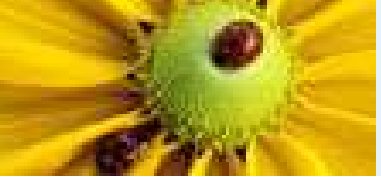


Proteinfamilien

Proteinfamilien: Konservierte Bereiche fester Länge (Blöcke) wechseln sich ab mit variablen Bereichen.

Nutze konservierte Strukturen um gezielt und exakt neue Mitglieder einer gegebenen Proteinfamilie in einem neuen Genom zu finden.

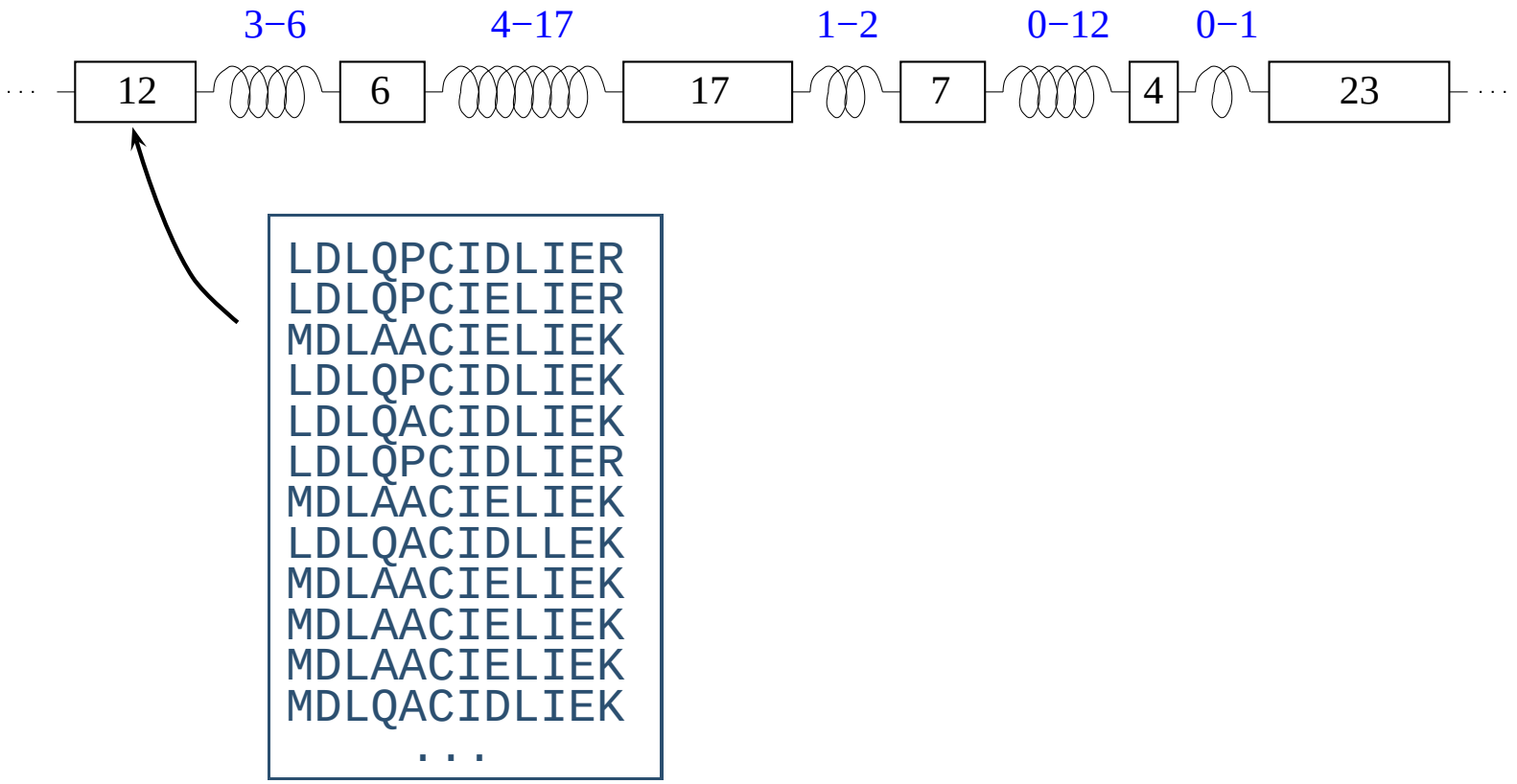


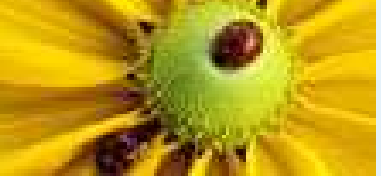


Proteinfamilien

Proteinfamilien: Konservierte Bereiche fester Länge (Blöcke) wechseln sich ab mit variablen Bereichen.

Nutze konservierte Strukturen um gezielt und exakt neue Mitglieder einer gegebenen Proteinfamilie in einem neuen Genom zu finden.

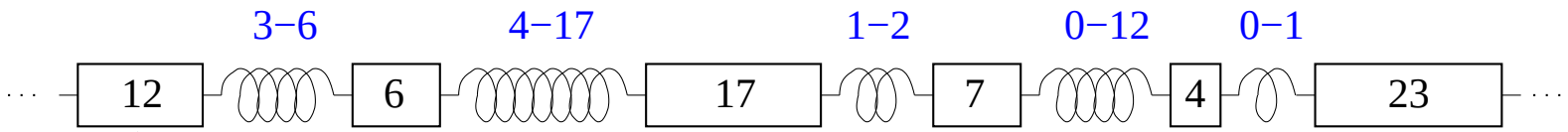




Proteinfamilien

Proteinfamilien: Konservierte Bereiche fester Länge (Blöcke) wechseln sich ab mit variablen Bereichen.

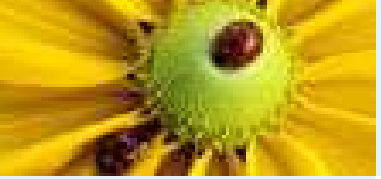
Nutze konservierte Strukturen um gezielt und exakt neue Mitglieder einer gegebenen Proteinfamilie in einem neuen Genom zu finden.




```
LDLQPCIDLIER
LDLQPCIELIER
MDLAACIELIEK
LDLQPCIDLIEK
LDLQACIDLIEK
LDLQPCIDLIER
MDLAACIELIEK
LDLQACIDLLEK
MDLAACIELIEK
MDLAACIELIEK
MDLAACIELIEK
MDLQACIDLIEK
    ...
```

Problem:

Finde Genstruktur im Genom unter Nebenbedinung, dass vorhergesagte Proteinsequenz zur Familie *passt*.



Semi-CRF schlecht geeignet

Die Modelklasse der Semi-CRFs ist zwar gut geeignet um

$$P(\text{Genstruktur} \mid DNA)$$

zu modellieren aber nicht gut um

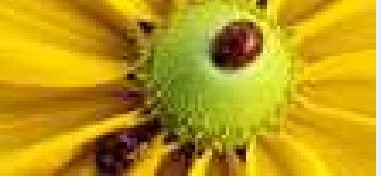
$$P(\text{Genstruktur} \mid \text{DNA}, \text{Proteinfamilie})$$

zu modellieren.

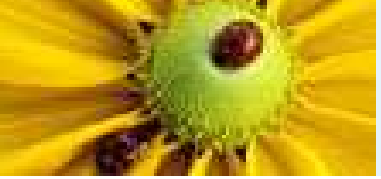
Problem: Zwischen jeweils 2 der drei folgenden Grössen ist die relative Lage zueinander unbekannt.

1. proteinkodierende Exons
2. DNA
3. Proteinfamilienalignment

Bemerkung: Wäre die relative Lage von 2. und 3. zueinander bekannt, z.B. durch ein Alignment A, könnte man gut ein Semi-CRF mit Beobachtung $y = (\text{DNA}, A)$ konstruieren.



Ein zweidimensionales Semi-Conditional-Random-Field



Bezeichungen

σ	Sequenz 1 über Alphabet Σ_1, hier: DNA-Sequenz der Länge ℓ_1
τ	Sequenz 2 über Alphabet Σ_2, hier: Proteinalignment der Länge ℓ_2
Q	Menge von Labels oder Zuständen
$y = (\sigma, \tau)$	Beobachtung

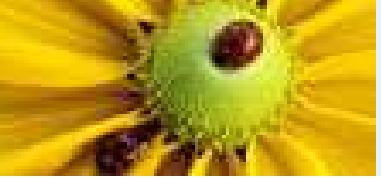
Ein Parse x ist eine gelabelte Segmentierung beider Sequenzen, σ und τ. (Tafel)

$x = (x_1, x_2, \ldots, x_n)$ mit $x_j = (q_j, u_j^{(1)}, v_j^{(1)}, u_j^{(2)}, v_j^{(2)})$

n ist also die Anzahl der Seqmente und in beiden Sequenzen gleich gross.

Dabei ist $q_j \in Q$ die Markierung (das Label), $u_j^{(1)}, v_j^{(1)}$ sind die Start- und Endposition in σ, $u_j^{(2)}, v_j^{(2)}$ sind Start- und Endposition in τ des j-ten Segments.

Es ist $v_{j-1}^{(i)} = u_j^{(i)} \leq v_j^{(i)}$ für $j = 1, \ldots, t, i = 1, 2$. Die Segmentierungen erstrecken sich über die vollständigen Sequenzen, d.h. $v_n^{(i)} = \ell_i$. Ebenso $u_0^{(i)} := 0$ für $i = 1, 2$ zur vereinfachten Notation.



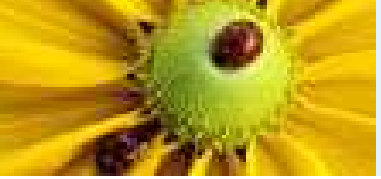
zweidimensionales Semi-Conditional-Random-Field

Definition:
Ein zweidimensionales (Lineare-Kette-)Semi-Conditional-Random-Field ist eine bedingte Verteilung von Parses gegeben die Beobachtung $y = (\sigma, \tau)$ der folgenden Form

$$\mathsf{P}(x \mid y) = \frac{1}{Z(y)} \exp \sum_{j=1}^{n} f(q_{j-1}, q_j, u_j^{(1)}, v_j^{(1)}, u_j^{(2)}, v_j^{(2)}, y).$$

Dabei ist $f \in \mathbb{R} \cup \{-\infty\}$ die Merkmalsfunktion und $Z(y)$ ist wieder so gewählt, dass die Summe über alle Parses 1 ergibt:

$$Z(y) = \sum_{x} \exp \sum_{j=1}^{n} f(q_{j-1}, q_j, u_j^{(1)}, v_j^{(1)}, u_j^{(2)}, v_j^{(2)}, y)$$



2-dim Semi-CRF für proteinfamilienbasierte GV

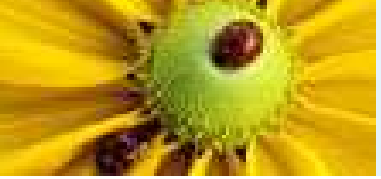
Sei τ eine Sequenz über einem geeigneten Alphabet Σ_2, die das Proteinfamilienalignment repräsentiert (1 Zeichen pro Spalte).
Seit σ eine DNA-Eingabesequenz über dem Alphabet $\Sigma_1 = \{a, c, g, t\}$.

Wir definieren ein zweidimensionales Semi-CRF wie folgt:

$$\begin{aligned} & f(q_{j-1}, q_j, u_j^{(1)}, v_j^{(1)}, u_j^{(2)}, v_j^{(2)}, y) \\ = \quad & f_1(q_{j-1}, q_j, u_j^{(1)}, v_j^{(1)}, y) + s(q, \sigma(u_j^{(1)}, v_j^{(1)}], \tau(u_j^{(2)}, v_j^{(2)}]). \end{aligned}$$

Hierbei sei f_1 eine geeignete Merkmalsfunktion zur normalen Genvorhersage, d.h. die Merkmalsfunktion eines eindimensionalen Semi-CRFs $P(x \mid \sigma)$.

$s(q, \sigma(u_j^{(1)}, v_j^{(1)}], \tau(u_j^{(2)}, v_j^{(2)}])$ bewertet, wie gut ggf. die durch q und $\sigma(u_j^{(1)}, v_j^{(1)}]$ gegebene kodierende DNA-Sequenz zum Segment $\tau(u_j^{(2)}, v_j^{(2)}]$ der Proteinfamilie passt.



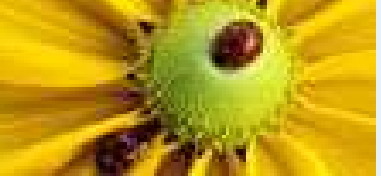
Ähnlichkeitsmaß DNA $\leftrightarrow$ Proteinalignment

DNA $\sigma(u^{(1)}, v^{(1)}]$	ata	agc	atg	atg	gat	ttg	caa	ccc	cat	gcg	tac	ata	tcg	ggg	acc	aat	tat	gaa	ttt	ggg	aca	gcg	gaa	gac	aaa	ccg
transl. DNA	I	S	I	M	D	L	Q	P	H	A	Y	I	S	G	T	N	Y	Z	F	G	T	A	E	D	K	P

$\updownarrow$ aligniert $\updownarrow$

Proteinfamilie $\tau(u^{(2)}, v^{(2)}]$

	L	D	L	Q	P		I	T	G	E		G	T	L	E	D	Q
	L	D	L	Q	P		I	T	G	T		G	T	L	E	D	Q
	M	D	L	A	A		I	T	G	E		L	T	A	E	D	K
	L	D	L	Q	P		I	T	G	E		G	T	L	E	D	Q
	L	D	I	Q	A		I	S	G	E		G	T	L	E	D	Q
	L	D	L	Q	P		I	T	G	E		G	T	L	E	D	Q
	M	D	L	A	A		I	T	G	E		G	T	L	E	D	Q
	L	D	L	Q	A		I	A	G	E		K	T	L	E	D	Q
	M	D	L	A	A		I	T	G	E		G	T	L	E	D	Q
	M	D	I	A	A		F	T	G	E		G	A	V	E	D	K
	M	D	L	A	A		I	T	G	E		G	S	L	E	D	Q
	M	D	L	Q	A		C	T	G	E		G	E	L	E	R	Q



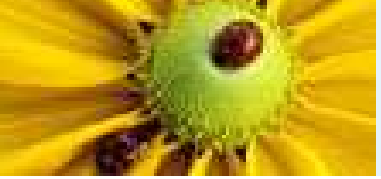
2-dim Semi-CRF für proteinfamilienbasierte GV

Beispiel für den Ansatz von Merkmal s als Ähnlichkeitsmaß zwischen kodierendem DNA-Segment $\sigma(u_j^{(1)}, v_j^{(1)}]$ und Abschnitt $\tau(u_j^{(2)}, v_j^{(2)}])$ im Proteinfamilienalignment

$$
\begin{aligned}
& s(q, \sigma(u_j^{(1)}, v_j^{(1)}], \tau(u_j^{(2)}, v_j^{(2)}]) \\
= & \begin{cases}
\log \prod_{\substack{\text{Position } i \\ \text{in Block}}} \frac{P(\text{AS}(i,\sigma) \mid \tau(i))}{P_0(\text{Kodon}(i,\sigma))} & , \text{ falls } q \in \text{CDS} \\
-\infty & , \text{ falls } q \notin \text{CDS und } u_j^{(2)} < v_j^{(2)} \\
0 & , \text{ sonst.}
\end{cases}
\end{aligned}
$$

Dabei:

CDS = Menge der Markierungen für proteinkodierende Segmente,
$\text{AS}/\text{Kodon}(i, \sigma)$ = Aminosäure/Kodon in σ aligniert zu Position i,
P_0 = Hintergrundwkeit



Aufgabe

Betrachte folgendes CRF “Münze im unehrlichen Casino”:
Zustandsraum $Q = \{\text{f,u}\}$ (fair, unfair), Alphabet $\Sigma = \{\text{K,Z}\}$ (Kopf, Zahl).
Seien $\sigma = y^{(1)} = \text{KZKZKZZZ}$, $x^{(1)} = \text{fffffuuu}$ und

$$P(x \mid \sigma) = \frac{1}{Z(\sigma)} \exp \sum_{i=1}^{8} \langle \mathbf{w}, \mathbf{g} \rangle, \quad \text{mit } \mathbf{g} = \mathbf{g}(x_{i-1}, x_i, \sigma_i) = \begin{pmatrix} \mathbf{1}_{\{x_i=\text{f},\sigma_i=\text{K}\}} \\ \mathbf{1}_{\{x_i=\text{f},\sigma_i=\text{Z}\}} \\ \mathbf{1}_{\{x_i=\text{u},\sigma_i=\text{K}\}} \\ \mathbf{1}_{\{x_i=\text{u},\sigma_i=\text{Z}\}} \\ \mathbf{1}_{\{x_{i-1}=\text{f},x_i=\text{f}\}} \\ \mathbf{1}_{\{x_{i-1}=\text{f},x_i=\text{u}\}} \\ \mathbf{1}_{\{x_{i-1}=\text{u},x_i=\text{f}\}} \\ \mathbf{1}_{\{x_{i-1}=\text{u},x_i=\text{u}\}} \end{pmatrix}$$

gegeben $(x_0 := \text{f})$. Setze $\mathbf{w} = (-1, -1, -1.5, -0.5, -0.5, -2, -2, -0.5)^t$ und führe eine Iteration (Schritte 4 und 5) im Online-Large-Margin-Trainingsalgorithmus durch. Was ist der neue Gewichtsvektor $\hat{\mathbf{w}}$? Nehme die Lossfunktion $L =$ “Anteil falsch klassifizierter Münzwürfe”