

Rozpoznávání genů

Bioinformatika

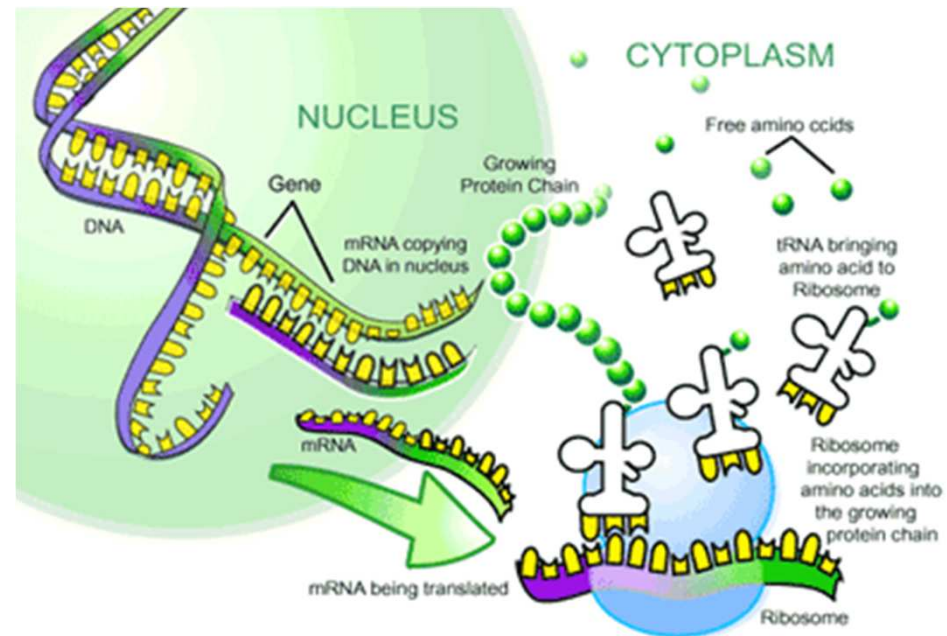
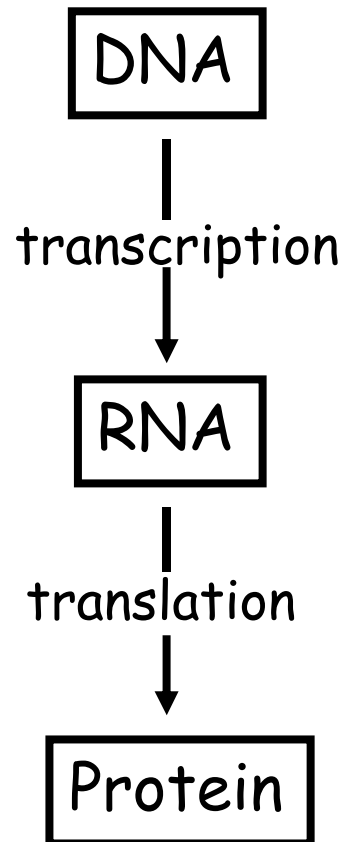
Tomáš Martínek

martinto@fit.vutbr.cz

Osnova

- Úvod
- Statistické metody
- Prokaryota vs. eukaryota
- Skryté Markovovy modely (HMM)
- Shrnutí

Úvod – Centrální dogma



Transkripce a Translace

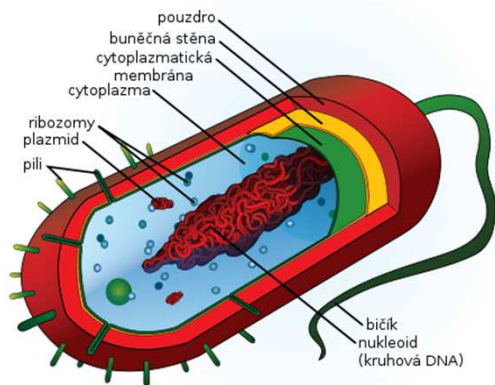


Reprezentace proteinu

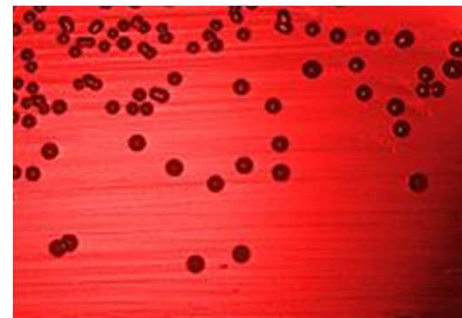
Detekce genů je nezbytná pro
predikci 3D struktury proteinu a
pochopení jeho funkce

Úvod

- Studium genomů bakteriálních organismů (prokaryotů) ukázalo zajímavé poznatky o minimálních požadavcích na život:
 - schopnost replikovat svou DNA (cca 32 genů)
 - schopnost tvořit proteiny (cca 100 až 150 genů)
 - schopnost uchovat energii (cca 30 genů)
- Jedním s příkladů jednoho z nejmenších prokaryotických organismů je bakterie *Haemophilus influenzae* (256-300 genů)



Prokaryotní buňka



Haemophilus influenzae

Objev kodonů

- V roce 1961, Sydney Brenner a Francis Crick objevili **mutaci posunu rámce**
- Systematicky odstraňovali nukleotidy z DNA, která kódovala protein a zjistili:
 - Odstranění jednoho nebo dvou znaků výrazně změnilo strukturu výsledného proteinu
 - Odstranění trojic nukleotidů mělo jen minimální efekt na výsledný protein
- **Závěr:**
 - každý triplet nukleotidů tzv. **kodon**, kóduje právě jednu aminokyselinu výsledného proteinu

Objev kodonů

- Příklad: analogie objevu kodonů
 - V následujícím řetězci
THE SLY FOX AND THE SHY DOG
 - Odstraň 1, 2, a 3 nukleotidy za prvním 'S':
THE SYF OXA NDT HES HYD OG
THE SFO XAN DTH ESH YDO G
THE SOX AND THE SHY DOG
 - Která z výše uvedených vět dává největší smysl?

Objev kodonů

- Kodon je trojice po sobě jdoucích nukleotidů
- $4^3 = 64$ možných kodonů
- Genetický kód je redundantní
 - Každý gen začíná Start (**AUG**) kodonem a končí některým ze tří Stop kodonů (**UAA, UAG, UGA**)
 - Každá aminokyselina může být kódována více kodony
- Sekvence znaků vyskytující se mezi start a stop kodonem se nazývá tzv. **čtecí rámec (Open Reading Frame - ORF)**

		2. báze			
		U	C	A	G
1. báze	U	UUU (Phe/F)	UCU (Ser/S)	UAU (Tyr/Y)	UGU (Cys/C)
		UUC (Phe/F)	UCC (Ser/S)	UAC (Tyr/Y)	UGC (Cys/C)
		UUA (Leu/L)	UCA (Ser/S)	UAA Ochre	UGA Opal
		UUG (Leu/L)	UCG (Ser/S)	UAG Amber	UGG (Trp/W)
	C	CUU (Leu/L)	CCU (Pro/P)	CAU (His/H)	CGU (Arg/R)
		CUC (Leu/L)	CCC (Pro/P)	CAC (His/H)	CGC (Arg/R)
		CUA (Leu/L)	CCA (Pro/P)	CAA (Gln/Q)	CGA (Arg/R)
		CUG (Leu/L)	CCG (Pro/P)	CAG (Gln/Q)	CGG (Arg/R)
	A	AUU (Ile/I)	ACU (Thr/T)	AAU (Asn/N)	AGU (Ser/S)
		AUC (Ile/I)	ACC (Thr/T)	AAC (Asn/N)	AGC (Ser/S)
		AUA (Ile/I)	ACA (Thr/T)	AAA (Lys/K)	AGA (Arg/R)
		AUG (Met/M)	ACG (Thr/T)	AAG (Lys/K)	AGG (Arg/R)
	G	GUU (Val/V)	GCU (Ala/A)	GAU (Asp/D)	GGU (Gly/G)
		GUC (Val/V)	GCC (Ala/A)	GAC (Asp/D)	GGC (Gly/G)
		GUA (Val/V)	GCA (Ala/A)	GAA (Glu/E)	GGA (Gly/G)
		GUG (Val/V)	GCG (Ala/A)	GAG (Glu/E)	GGG (Gly/G)

Naivní metoda

- **Cíl:**
 - Nalezení všech ORF rámců
- **Postup:**
 - Vyhledej všechny sekvence mezi **Start** a **Stop** kodonem
- **Problém:**
 - Nevím, kde začíná kodon a zda jsem na přímém nebo reverzním vlákně
- **Řešení:**
 - Prohledám všech šest možností

Naivní metoda

- **Příklad:**

CTGCAGACGAAACCTCTTGATGTAGTTGGCCTGACACCGACAATAATGAAGACTACCGTCTTACTAACAC
CTGCAGACGAAACCTCTTGATGTAGTTGGCCTGACACCGACAATAATGAAGACTACCGTCTTACTAACAC
CTGCAGACGAAACCTCTTGATGTAGTTGGCCTGACACCGACAATAATGAAGACTACCGTCTTACTAACAC

→

CTGCAGACGAAACCTCTTGATGTAGTTGGCCTGACACCGACAATAATGAAGACTACCGTCTTACTAACAC
GACGTCTGCTTTGGAGAACTACATCAACCGGACTGTGGCTGTTATTACTTCTGATGGCAGAATGATTGTG

←

GACGTCTGCTTTGGAGAACTACATCAACCGGACTGTGGCTGTTATTACTTCTGATGGCAGAATGATTGTG
GACGTCTGCTTTGGAGAACTACATCAACCGGACTGTGGCTGTTATTACTTCTGATGGCAGAATGATTGTG
GACGTCTGCTTTGGAGAACTACATCAACCGGACTGTGGCTGTTATTACTTCTGATGGCAGAATGATTGTG

- **start kodon** – ATG
- **stop kodony** – TAA, TAG, TGA

Naivní metoda

- **Problém:**
 - Jak zjistím, že jsem nenašel Start a Stop kodon v nekódující oblasti?
 - Pokud se na nekódující oblast podíváme jako na náhodnou sekvenci kodonů, potom v průměru každých $(64/3) \approx 21$ kodonů se vyskytuje jeden stop kodon
 - Avšak geny jsou obvykle mnohem delší
- **Řešení:**
 - Hledám pouze sekvence mezi start a stop kododem delší než 21 kodonů.
- **Poznámky:**
 - První programy pro hledání genu tak skutečně fungovaly, ale není to příliš spolehlivé, protože některé geny jsou kratší než 21 kodonů (např. geny nervového nebo imunitního systému)

Metoda založená na statistice kodonů

- Vylepšená metoda sleduje četnost kodonů v kódujících oblastech
- Aminokyseliny jsou sice kódovány více kodony, ale některé z nich se používají častěji
- Podobně, četnost použití určitých kodonů v kódujících oblastech se liší od nekódujících

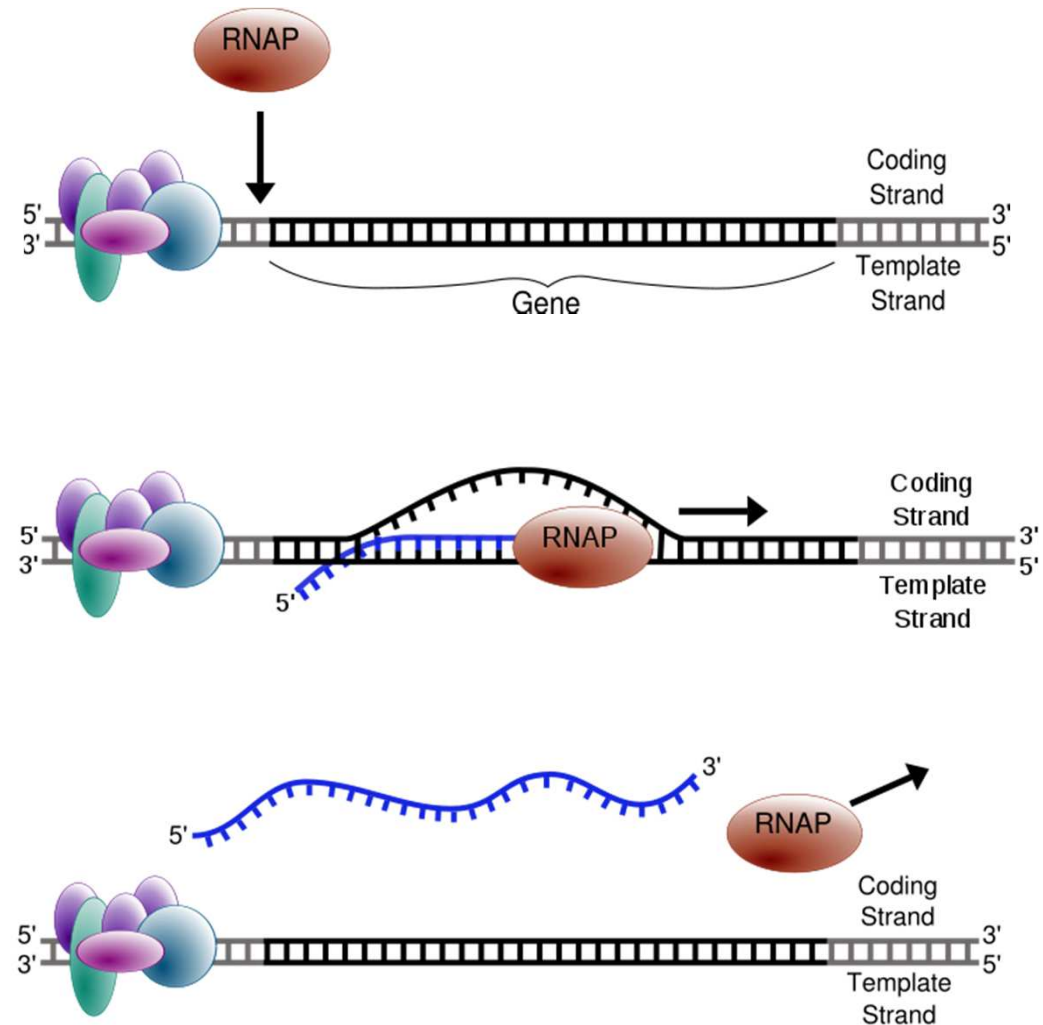
		2. báze			
		U	C	A	G
1. báze	U	UUU (Phe) 57	UCU (Ser) 16	UAU (Tyr) 58	UGU (Cys) 45
		UUC (Phe) 43	UCC (Ser) 15	UAC (Tyr) 42	UGC (Cys) 55
		UUA (Leu) 13	UCA (Ser) 13	UAA (Stp) 62	UGA (Stp) 30
		UUG (Leu) 13	UCG (Ser) 15	UAG (Stp) 8	UGG (Trp) 100
	C	CUU (Leu) 11	CCU (Pro) 17	CAU (His) 57	CGU (Arg) 37
		CUC (Leu) 10	CCC (Pro) 17	CAC (His) 43	CGC (Arg) 38
		CUA (Leu) 4	CCA (Pro) 20	CAA (Gln) 45	CGA (Arg) 7
		CUG (Leu) 49	CCG (Pro) 51	CAG (Gln) 66	CGG (Arg) 10
	A	AUU (Ile) 50	ACU (Thr) 18	AAU (Asn) 46	AGU (Ser) 15
		AUC (Ile) 41	ACC (Thr) 42	AAC (Asn) 54	AGC (Ser) 26
		AUA (Ile) 9	ACA (Thr) 15	AAA (Lys) 75	AGA (Arg) 5
		AUG (Met) 100	ACG (Thr) 26	AAG (Lys) 25	AGG (Arg) 3
	G	GUU (Val) 27	GCU (Ala) 17	GAU (Asp) 63	GGU (Gly) 34
		GUC (Val) 21	GCC (Ala) 27	GAC (Asp) 37	GGC (Gly) 39
		GUA (Val) 16	GCA (Ala) 22	GAA (Glu) 68	GGA (Gly) 12
		GUG (Val) 36	GCG (Ala) 34	GAG (Glu) 32	GGG (Gly) 15

Metoda založená na statistice kodonů

- **Princip:**
 - Posuvné okénko postupně analyzuje sekvenci, vyhodnocuje četnost kodonů a porovnává s tabulkou
 - Dosahuje nohem lepší výsledky, než testováním délky
 - Další vylepšení: **in-frame hexamer counter**: počítání frekvence po sobě jdoucích dvojic kodonů
- **Problémy:**
 - Jak určit velikost okénka?
 - Některé geny jsou příliš krátké pro statisticky významné porovnání
 - => Je potřeba v sekvenci sledovat více signálů, než jen statistiku kodonů

Průběh transkripce

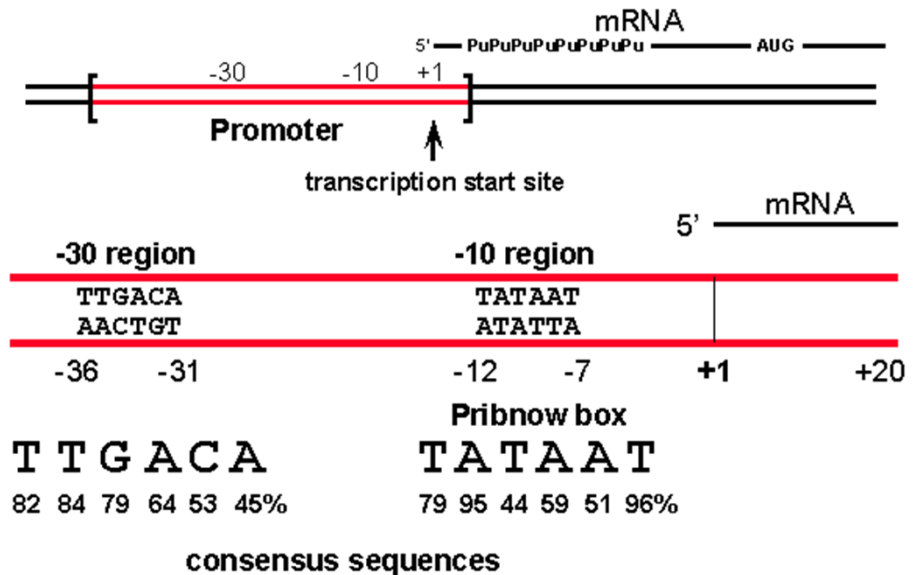
- Hlavní roli při transkripci hraje **RNA Polymeráza**, která nasedá na oblast tzv. **promotoru** – specifická oblast před začátkem genu cca 30 bází
- K transkripci dochází pouze za předpokladu, že jsou v oblasti promotoru přítomny i příslušné **regulátory** – proteiny, které rozhodují, zda k transkripci dojde, či nikoliv
 - **negativní regulátor** – zabraňuje, aby RNAP začala transkripci
 - **pozitivní regulátor** – bez něj ke transkripci nedojde



Startovací signály

- **Struktura promotoru**
 - Transkripce začíná na offsetu 0
 - **Pribnow Box (TATA box)** začíná na offsetu -10
 - **Gilbertův Box** na offsetu -30
- U jednotlivých sekvencí je uvedena procentuální pravděpodobnost výskytu
- Někdy se uvádí také **logo** nebo **konsensus sekvence** (podobná regulárnímu výrazu)

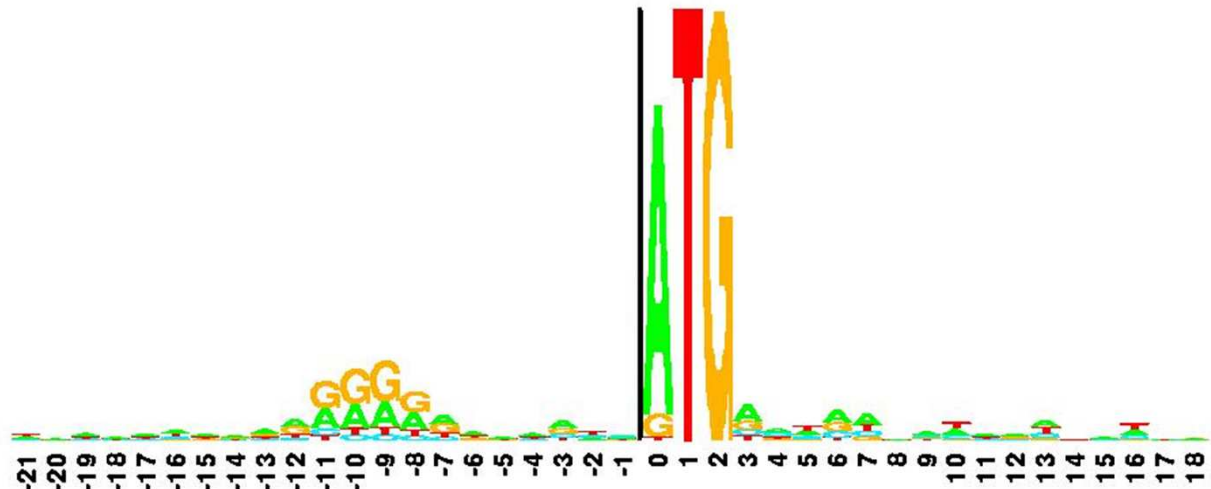
Promoter structure in prokaryotes



Startovací signály

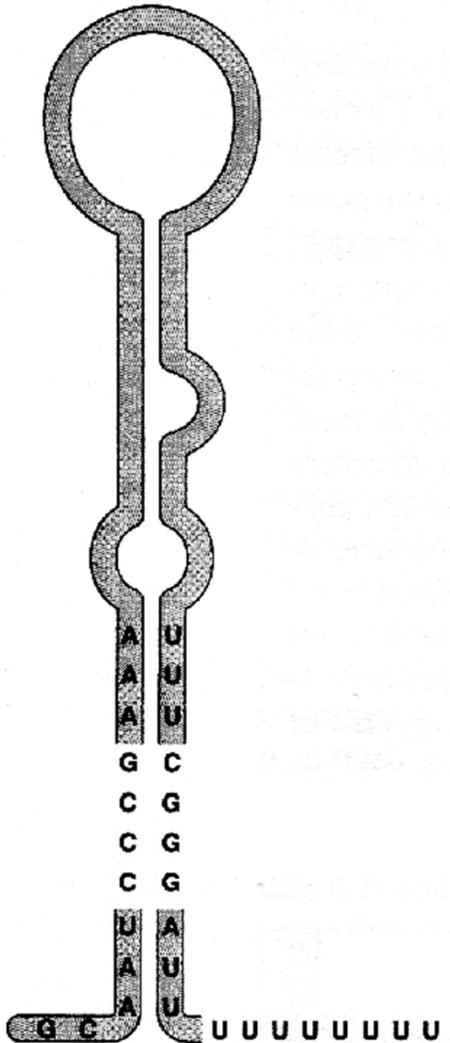
- Charakteristickým znakem prokaryotických genu je, že obsahují tzv. **Shine-Delgarno sekvence**
- Tyto sekvence jsou umístěny **mezi začátkem transkripce a start kodonem** a pomáhají ribozomům začít překlad
- Existuje více typů těchto sekvencí, ale většina z nich obsahuje znaky: **AGGAGGU**

1055 E. coli Ribosome binding sites listed in the Miller book



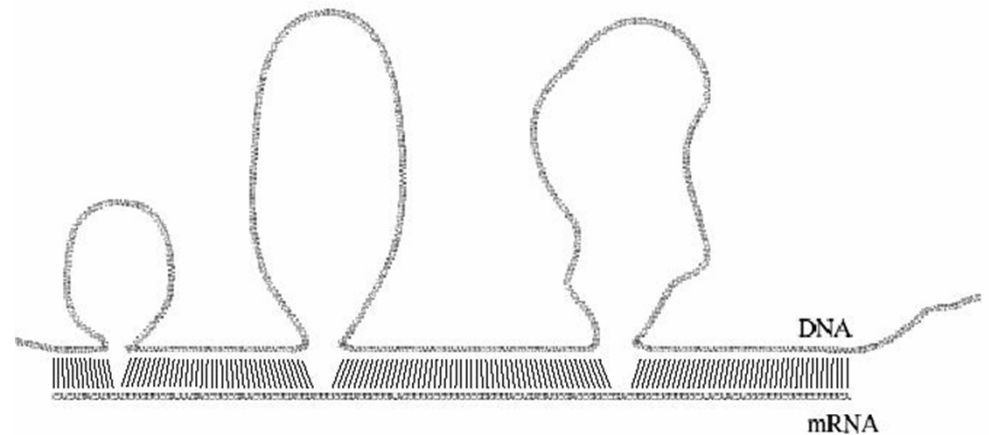
Ukončovací signály

- Naprostá většina genů (>90%) obsahuje také ukončovací sekvence pro transkripci - **terminátory**
- Pro tyto sekvence je charakteristické, že obsahují **palindromickou strukturu** délky **7-20 nukleotidů** následovanou **cca 6-ti uracily**
- Palindromická struktura u RNA vytvoří pevnější vazby mezi GC a AU (viz příklad) a vytvoří tzv. sekundární strukturu RNA
- Experimentálně bylo zjištěno, že pokud RNA polymeráza narazí na tuto strukturu, tak se pozastaví na cca 1 minutu (velké zpoždění vzhledem k běžné rychlosti 100 nukl./sekundu)
- Toto zpomalení způsobí, že vazby sekvence uracilu a potenciálního ademinu už nejsou tak silné a transkripce se ukončí



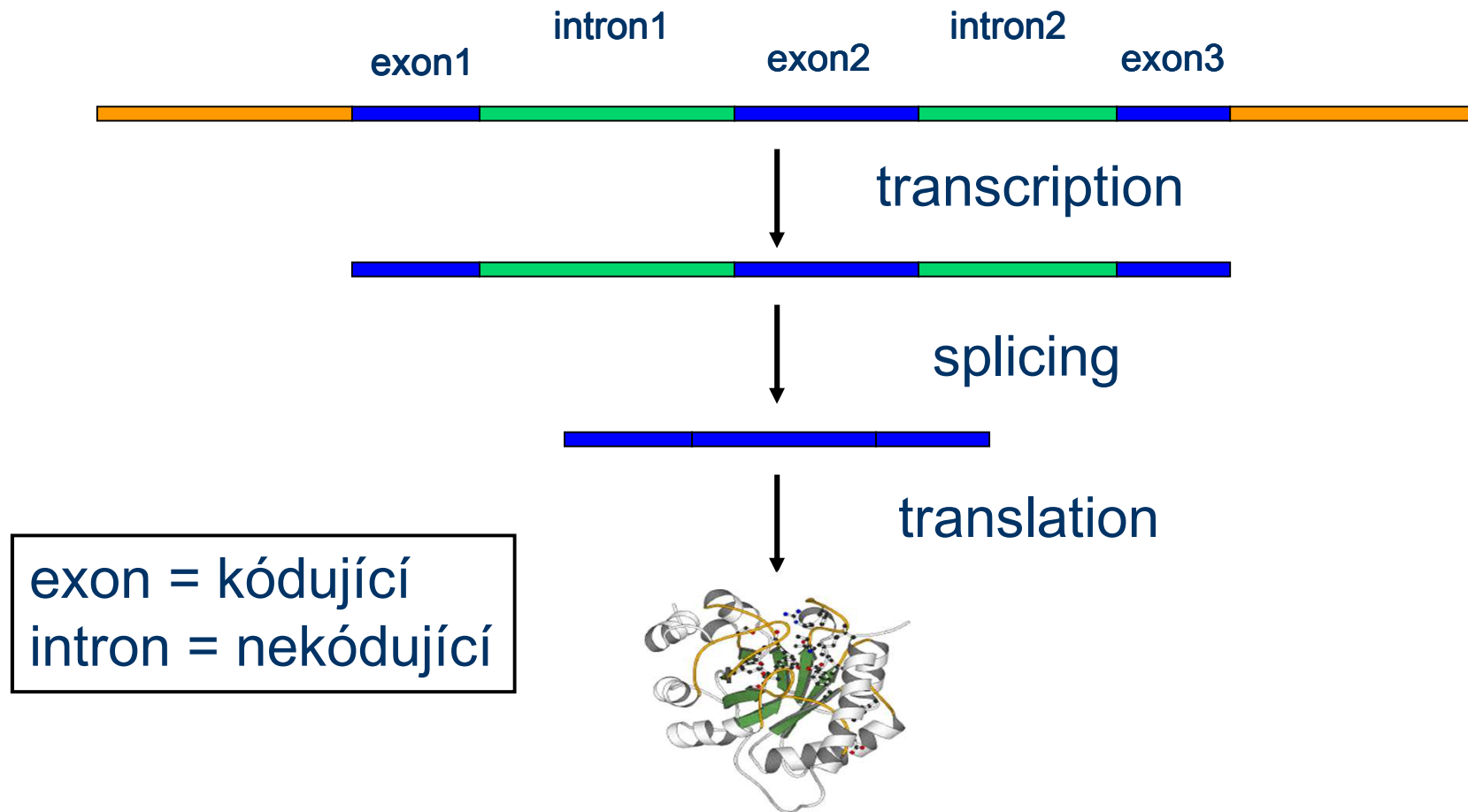
Objevení intronů

- V roce 1977, Phillip Sharp a Richard Roberts experimentovali s mRNA *hexonu* (virový protein):
 - porovnali mRNA viru s jeho ekvivalentní formou v DNA
 - objevili, že mRNA-DNA hybrid tvoří tři kuriózní smyčky namísto spojitého segmentu
- V roce 1978 Walter Gilbert tyto mezery nazval **introny** (časopis Nature paper “Why Genes in Pieces?”)



- V eukaryotech jsou geny složeny z kódujících segmentů (**exonů**), které jsou přerušovány nekódujícími segmenty (**introny**)
- **K tomuto dochází pouze u eukaryotů**

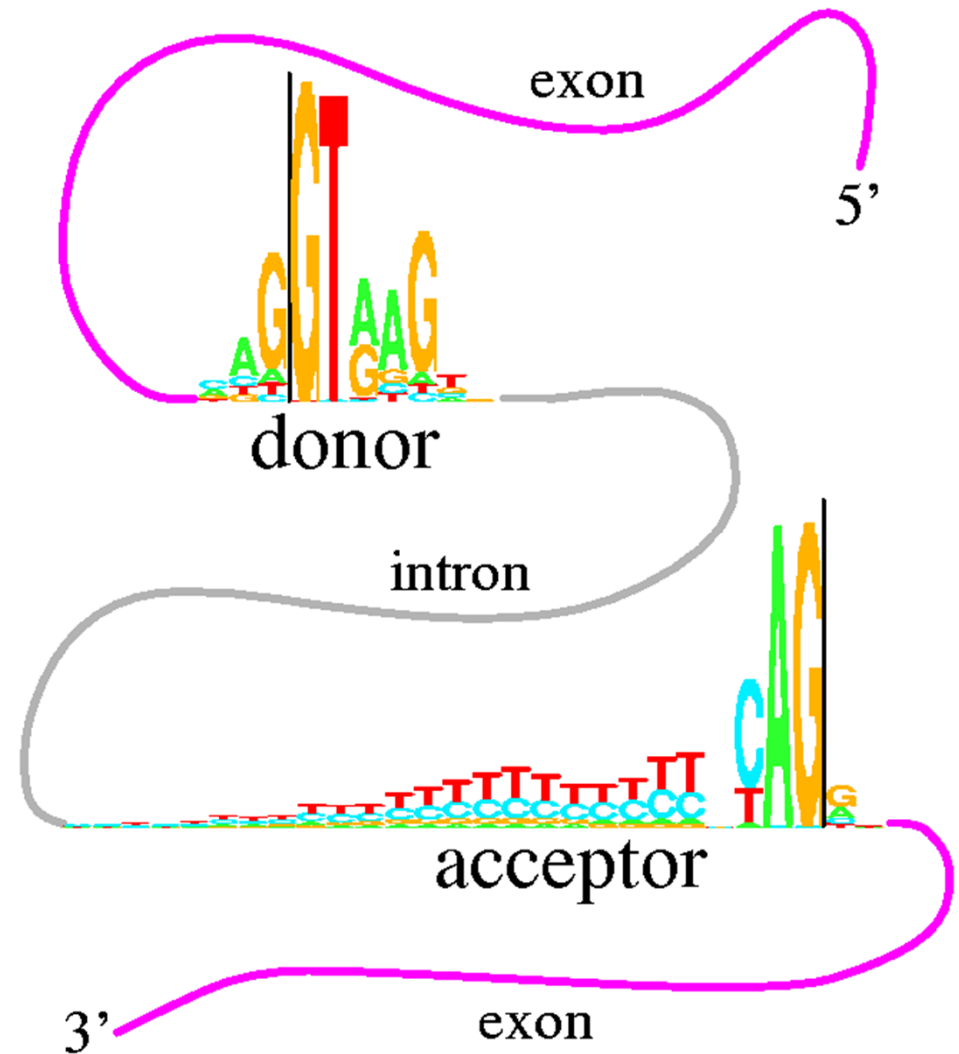
Centrální dogma a splicing



Jak poznat začátek a konec intronu?

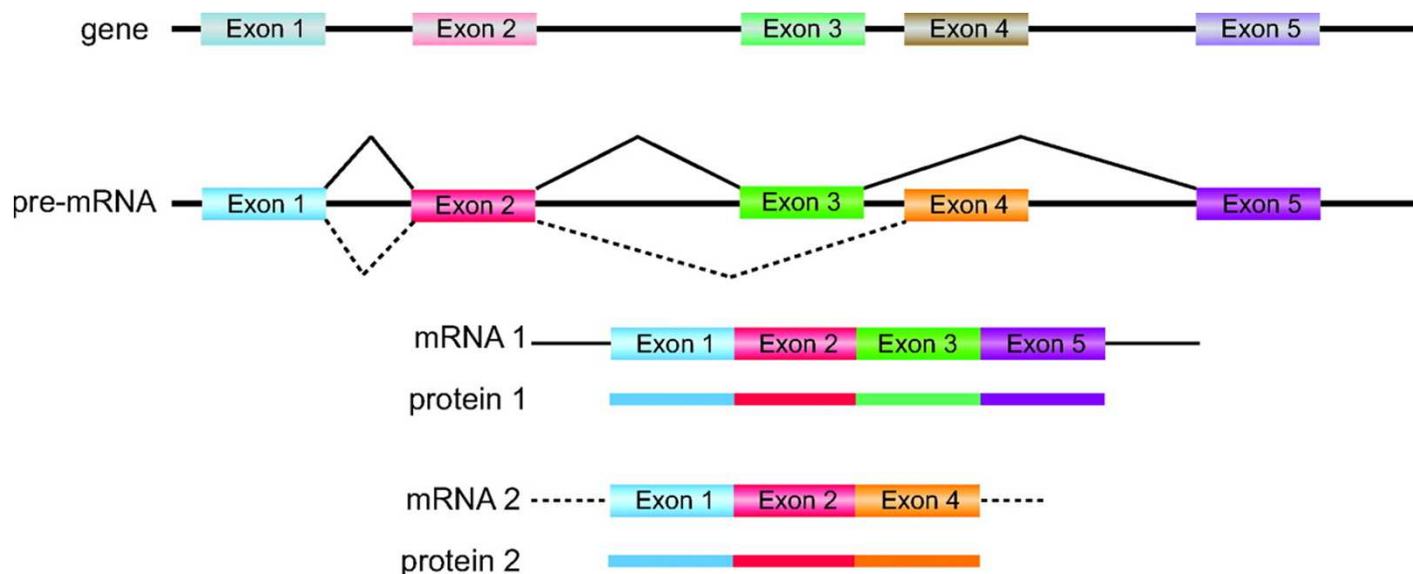
Oddělovací signály

- Existuje cca 8 typů intronů, jeden z nich vyhovuje GT-AG pravidlu (převážně vyskytující se)
- Minimální délka je 60bp (aby bylo možné všechny potřebné signály pro ořezání), maximální délka není omezena
- Rovněž délka exonů je různá cca 100 - 2000 bp nebo více
- Neexistují žádná pravidla pro distribuci intronů
- Příklady:
 - genom kvasinky: cca 6000 genů, celkem 239 intronů
 - lidské genom: některé geny mají až 100 intronů
 - genomy obratlovců: 95% genů má alespoň jeden intron



Alternativní splicing

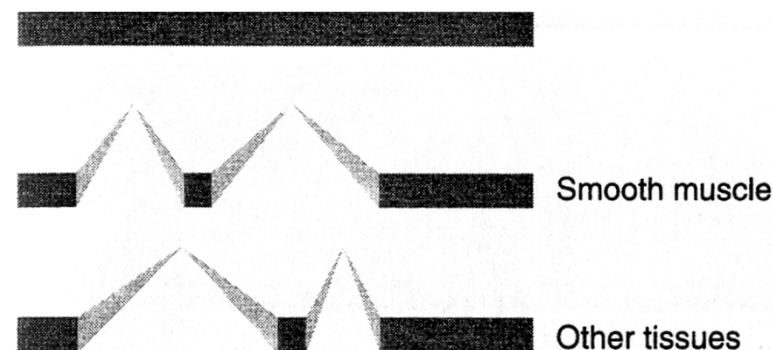
- Bylo vypořádováno, že ze stejných transkriptů genů se tvoří různé varianty mRNA – **alternativní slicing**
- Odhaduje se, že až **20%** lidských genů má tuto vlastnost. Některé z nich mají až **64** různých mRNA variant
- **Objev alternativního splicing-u vyvrátil původní hypotézu, že z jednoho genu se tvoří vždy jeden protein**



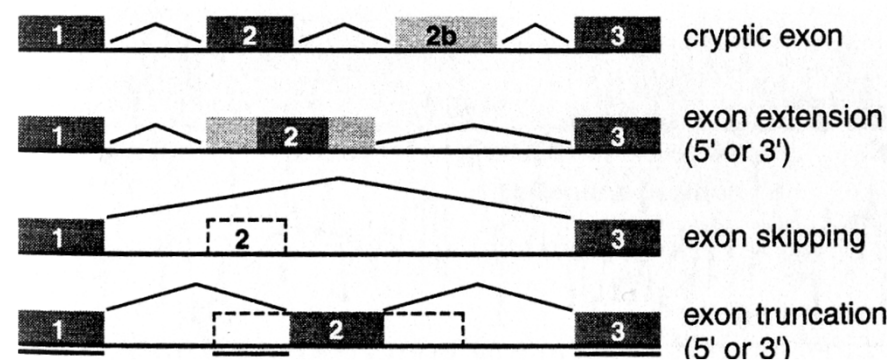
Alternativní splicing

- **Způsob spojování** genů může být ovlivněn typem buněk nebo jinými okolnostmi
 - např.: T gen myši obsahuje exon 2 a 3, které se vzájemně vylučují, v závislosti na typu buněk. Exon 2 je použit v buňkách hladkého svalstva, zatímco exon 3 je použit ve všech ostatních tkáních
- **Spojovací aparát je tvořen z malých jaderných RNA (snRNA)** a několika proteinů v závislosti na typu buněk
- Existují **různé modely spojování exonů**
- **Vytvoření vhodného výpočetního modelu pro spojování genů stále zůstává jedním z hlavních problémů algoritmů pro rozpoznávání genů**

Obr.: T gen myši



Obr.: Různé modely spojování exonů



Prokaryoty vs. Eukaryoty

- **Prokaryoty**

- Vysoká hustota genů (85-88% genomu jsou kódující sekvence)
- Jeden promotor sdílí více genů současně
- 1 typ RNA polymerázi, pouze několik typů regulátorů
- Neobsahují introny

- **Eukaryoty**

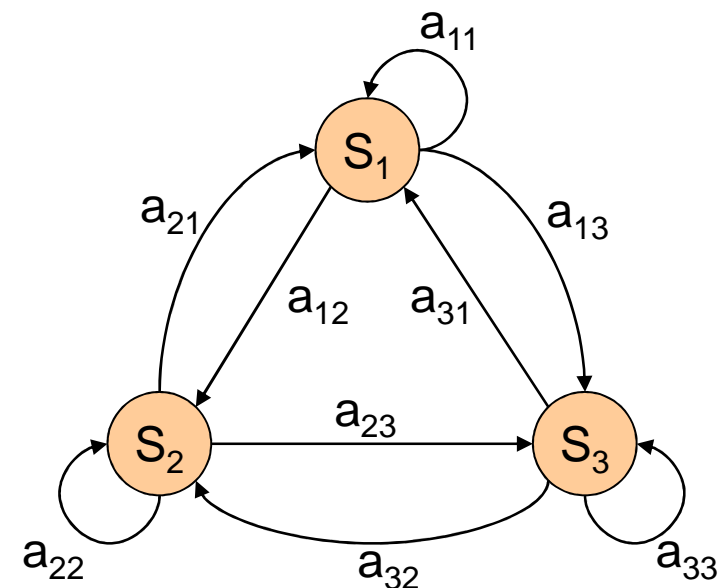
- Nízká hustota genů (např. u člověka pouze 5% tvoří kódující sekvence)
- Každý gen má svůj vlastní promotor
- 3 typy RNA polymerázi, velké množství regulátorů
- Obsahují introny
- Alternativní splicing

Detekce genů v eukaryotech je v současnosti jedna z nejsložitějších výpočetních úloh v oblasti bioinformatiky

Hidden Markov Models (HMM)

- Diskrétní Markovův model
 - model systému, který se může nacházet v jednom z množiny (diskrétních) stavů S_1, S_2, \dots, S_N
 - Přechody mezi stavy i a j jsou dány podmíněnou pravděpodobností
 - $a_{ij} = P(g_t = S_i | g_{t-1} = S_j)$
 - Pro hodnoty pravděpodobností platí:
 - $a_{ij} \geq 0$
 - $\sum_{j=1}^N a_{ij} = 1$

- Příklad:



Fair Bet Casino

- Příklad modelu: „Fair Bet Casino“
 - Cílem hry je předpovědět výsledek hodu mincí
 - Výsledkem hodu mincí může být pouze pana (**Head**) nebo orel (**Tail**)
 - Pokud je mince férová (**F**), potom je pravděpodobnost hodu obou stran shodná $P(H|F) = P(T|F) = \frac{1}{2}$
 - Pokud je ale mince podvržená (**B**), potom se pravděpodobnosti liší např. $P(H|B) = \frac{3}{4}$ a $P(T|B) = \frac{1}{4}$
 - Falešný obchodník, který mincí hází, střídá mezi férovou a podvrženou mincí s pravděpodobností 0,1

Fair Bet Casino

- Sestavení Modelu:

- Rozlišujeme dva stavy:

- Férová mince (F)
 - Podvržená mince (B)

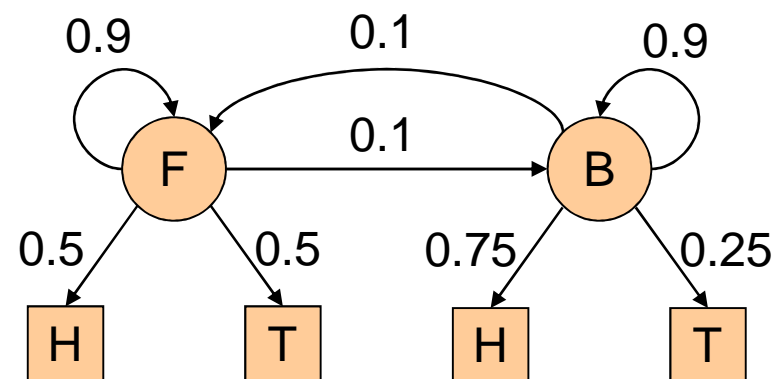
- V každém stavu padá na výstupu buď pana (H) nebo orel (T) s tzv. **emisní pravděpodobnost**

- $e_F(H) = e_F(T) = \frac{1}{2}$
 - $e_B(H) = \frac{3}{4}$ a $e_B(T) = \frac{1}{4}$

- Pravděpodobnost přechodu** mezi stavy je:

- $a_{F,B} = a_{B,F} = 0.1$
 - $a_{F,F} = a_{B,B} = 0.9$

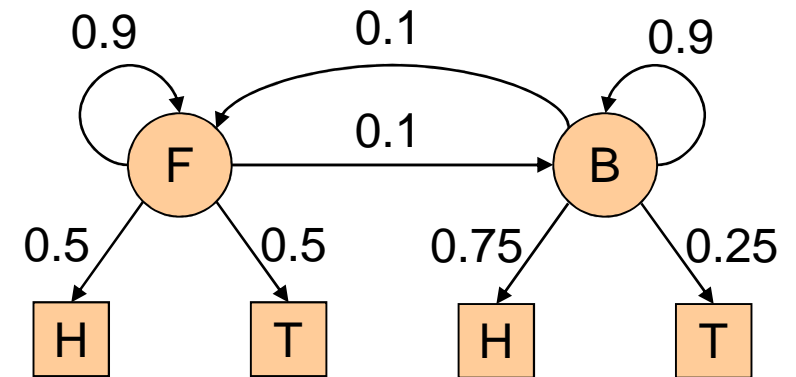
- Markovův model



- Pravděpodobnost, že obchodník na začátku vybral férovou nebo podvrženou minci je:
 - $\pi_F = \pi_B = \frac{1}{2}$

Fair Bet Casino

- Na výstupu pozorujeme sekvenci $x = \text{THTHHHTHTT}$.
- Jaká je pravděpodobnost, že tato výstupní sekvence byla generována průchodem skrze stavy $S = \text{FFB BBBBFFF}$?



Pravděpodobnost, že x_i bylo emitováno ve stavu S_i

x	T	H	T	H	H	H	T	H	T	T	H
S	F	F	F	B	B	B	B	B	F	F	F
$e_{S_i}(x_i)$	$1/2$	$1/2$	$1/2$	$3/4$	$3/4$	$3/4$	$1/4$	$3/4$	$1/2$	$1/2$	$1/2$
a_{S_{i-1}, S_i}	$1/2$	$9/10$	$9/10$	$1/10$	$9/10$	$9/10$	$9/10$	$9/10$	$1/10$	$9/10$	$9/10$

Pravděpodobnost přechodu ze stavu S_{i-1} do stavu S_i

Fair Bet Casino

- Matematický zápis:
 - $P(x, S|\lambda)$: Pravděpodobnost, že sekvence x byla vygenerována cestou S :

$$P(x, S|\lambda) = \pi_{S_0} \cdot \prod_{i=1}^n e_{S_i}(x_i) \cdot a_{S_i, S_{i+1}}$$

- Příklad:
 - $P(\text{THTHHHTHTT}, \text{FFFB BBBBFFFF}|\lambda)$
= $\pi_F \cdot e_F(\text{T}) \cdot a_{F,F} \cdot e_F(\text{H}) \cdot a_{F,F} \cdot e_F(\text{H}) \cdot a_{F,B} \dots$
= $0.5 \cdot 0.5 \cdot 0.9 \cdot 0.5 \cdot 0.9 \cdot 0.5 \cdot 0.1 \dots$

Fair Bet Casino

- Proč Skrytý Markovův Model?
 - Ve skutečnosti nevidíme stavy modelu, ale pouze výstupy (pana vs. orel)
- Abychom v kasinu vydělali musíme se na základě výstupní sekvence ptát: Kterou z mincí (stav modelu) obchodník právě používá?
- Nebo ještě obecněji: Jaká je nejpravděpodobnější sekvence stavů pokud na výstupu pozorujeme sekvenci $x = \text{HTHTTTHTHH}$?

Fair Bet Casino

- **Jednoduché řešení:**

1. Vyzkoušíme všechny možné sekvence stavů S
2. Vyhodnotíme pravděpodobnost $P(x, S|\lambda)$
3. Vybereme sekvenci stavů s nejvyšší pravděpodobností

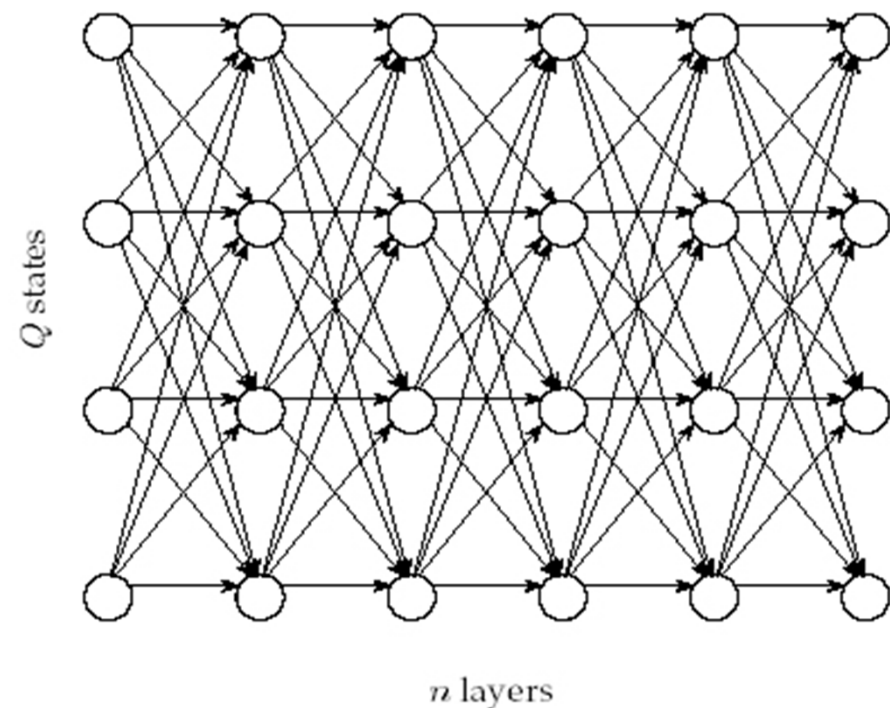
- **Problém:**

- Počet sekvencí je obecně N^T , kde N je počet stavů a T je délka sekvence
- Výpočet hrubou silou je možný cca pro 5 stavů a sekvenci dlouhou 100 znaků

Viterbiho algoritmus

- Založen na technice dynamického programování
- Obecný postup:
 1. Postupně prochází všechny stavy zleva doprava
 2. Pro každý stav S_i vyhodnocuje pravděpodobnost přechodu z předcházejících stavů S_{i-1} do aktuálního stavu
 3. Vybere maximální pravděpodobnost a přiřadí ji uzlu. Toto maximum se v uzlu využije pro výpočet v dalším kroku $i+1$
- Nejlépe lze pozorovat na diagramu všech možných stavů

- Diagram možných stavů



Viterbiho algoritmus - Příklad

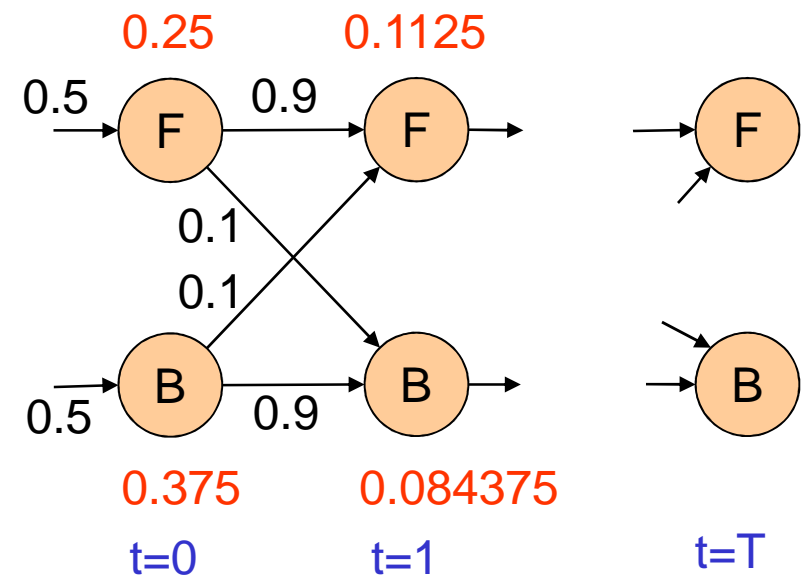
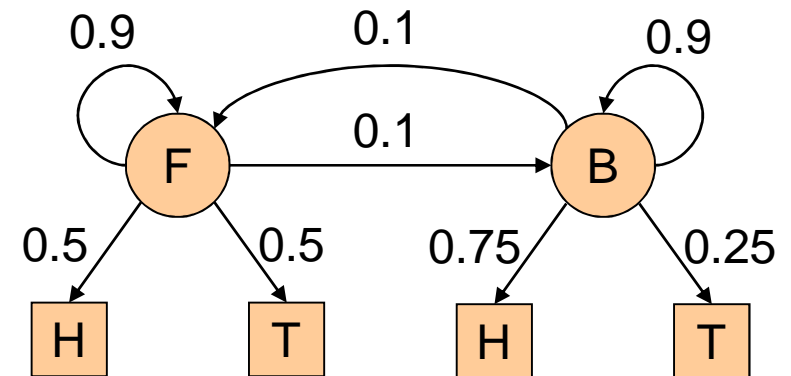
- Fair Bet Casino: Jaká je nejpravděpodobnější sekvence stavů pokud na výstupu pozorujeme sekvenci $x = \text{HTHTTTHTHH}$?
- Výpočet:

$$\delta_0(F) = \pi_F \cdot e_F(H) = 0.5 \cdot 0.5 = \mathbf{0.25}$$

$$\delta_0(B) = \pi_B \cdot e_B(H) = 0.5 \cdot 0.75 = \mathbf{0.375}$$

$$\begin{aligned} \delta_1(F) &= \max[\delta_0(F)a_{F,F}, \delta_0(B)a_{B,F}] \cdot e_F(T) \\ &= \max[0.25 \cdot 0.9, 0.375 \cdot 0.1] \cdot 0.5 \\ &= 0.225 \cdot 0.5 = \mathbf{0.1125} \end{aligned}$$

$$\begin{aligned} \delta_1(B) &= \max[\delta_0(F)a_{F,B}, \delta_0(B)a_{B,B}] \cdot e_B(T) \\ &= \max[0.25 \cdot 0.1, 0.375 \cdot 0.9] \cdot 0.25 \\ &= 0.3375 \cdot 0.25 = \mathbf{0.084375} \end{aligned}$$



Viterbiho algoritmus

- V každém stavu si musím uchovávat informaci o tom, která z cest byla nejvýhodnější
 - $\psi(i) = \operatorname{argmax}[\delta_{t-1}(i)a_{i,j}]$
- Zpětný průchod:
 1. V posledním kroku vyber stav s hodnotu nejvyšší hodnotou pravděpodobnosti
 2. S pomocí $\psi(i)$ rekonstruuji cestu k nejvyšší pravděpodobnosti
- Časová složitost:
 - Diagram stavů má $|Q|^2(n-1)$ hran,
 - Časová složitost je $O(n|Q|^2)$, tj. výrazně lepší než $O(N^T)$

Trénování HMM

- Doposud jsme předpokládali, že známe hodnoty pravděpodobností modelu, tj, jak **přechodů**, tak **emisní pravděpodobnosti**
- Obvykle tyto hodnoty neznáme a zjišťujeme je až metodou trénování na známé množině vzorků výstupních sekvencí + sekvenci stavů = **trénovací množina**
- Existuje celá řada metod pro trénování HMM, nejčastěji se používá **Baum-Welchův** přístup
- **Avšak s ohledem na jeho složitost, uvedeme pouze obecný princip trénování**

Baum-Welchova metoda trénování

- **Iterační přístup**, každou iterací se zpřesňují parametry HMM modelu
- **Postup (pouze náznak):**
 1. Nastav počáteční parametry modelu
 2. Pro zadanou výstupní sekvenci + sekvenci stavů vypočti:
 - $\gamma_t(i)$ - pravděpodobnost, že v kroku t je systém ve stavu i
 - $\epsilon_t(i,j)$ - pravděpodobnost, že v kroku t došlo k přechodu a_{ij}
 - $\sum \gamma(i)$ = očekávaný počet přechodů ze stavu S_i
 - $\sum \epsilon_t(i,j)$ = očekávaný počet přechodu z S_i do S_j
 3. Pravděpodobnost přechodů
 - $a_{ij} = \sum \epsilon_t(i,j) / \sum \gamma(i)$
 4. Emisní pravděpodobnost
 - $e_t(k) = \sum \gamma(i,k) / \sum \gamma(i)$
 5. Opakuj od bodu 2. dokud dochází ke zlepšení

Aplikace HMM na rozpoznání genů

1. Sestavení modelu

- Ruční identifikace jednotlivých stavů a přechodů na základě znalosti o struktuře genů

2. Trénování modelu

- Emisní a přechodové pravděpodobnosti určíme na základě sekvencí se známými geny

3. Analýza neznáme sekvence

- Pro neznámou sekvenci S najdi nejpravděpodobnější rozdělení sekvence na exony, introny a mezigenovou oblast pomocí Viterbiho algoritmu

HMM 1 – Jednoduchý příklad

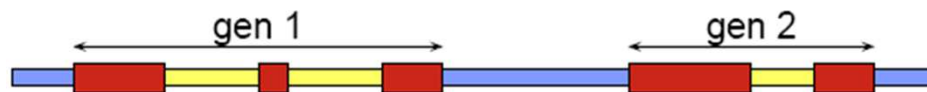
- Rozlišujeme pouze tři stavy:

- Exon, Intron, Mezi genový úsek

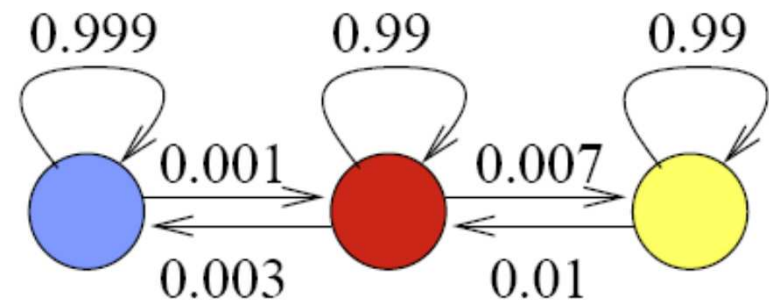
$$A_i \in \{\text{red}, \text{yellow}, \text{blue}\}$$

- Emitované znaky jsou báze DNA

$$S_i \in \{a, c, g, t\}$$



Pozn.: Přejchod mezi intronem na mezigenovým úsekem není možný



a			
	0.99	0.007	0.003
	0.01	0.99	0
	0.001	0	0.999

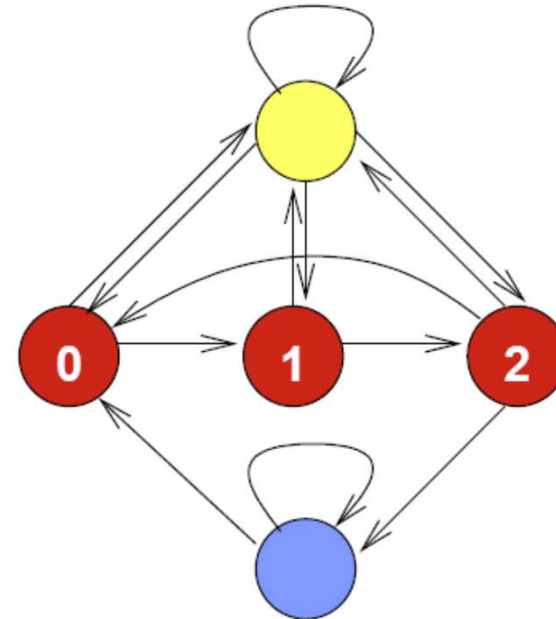
$\Pr(A_i|A_{i-1})$

e	a	c	g	t
	0.24	0.27	0.28	0.21
	0.26	0.22	0.22	0.30
	0.27	0.23	0.23	0.27

$\Pr(S_i|S_i)$

HMM 2 – respektování kodonů

- Přidáme další informace:
 - každý kodon je tvořen trojicí znaků
 - některé přechody HMM nejsou povoleny
 - změna přechodových i emisních pravděpodobností



A	0	1	2	Yellow	Blue
0	0		0		0
1	0	0			0
2		0	0		
Yellow					0
Blue		0	0		0

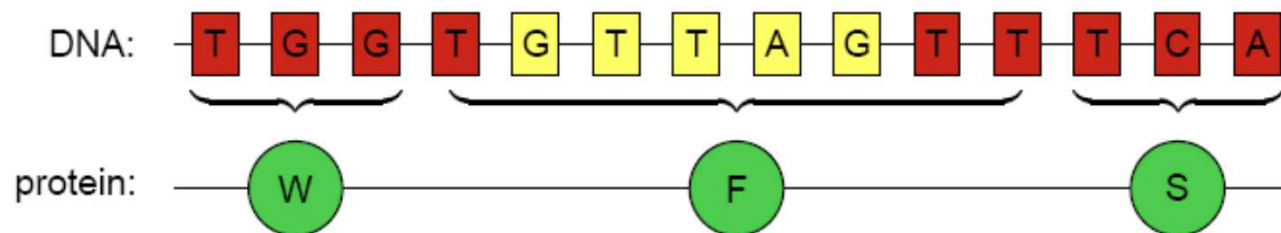
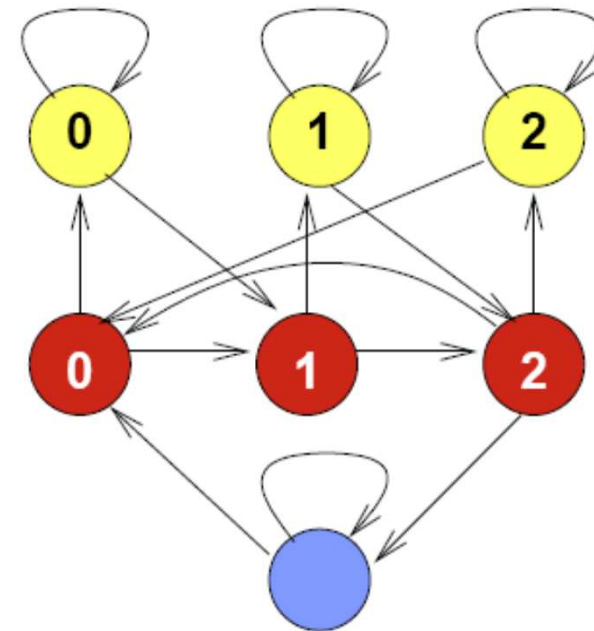
$\text{Pr}(A_i|A_{i-1})$

η	a	c	g	t
0	0.26	0.26	0.32	0.16
1	0.30	0.24	0.20	0.26
2	0.17	0.32	0.31	0.20
Yellow	0.26	0.22	0.22	0.30
Blue	0.27	0.23	0.23	0.27

HMM 3 – přerušení kodonu intronem

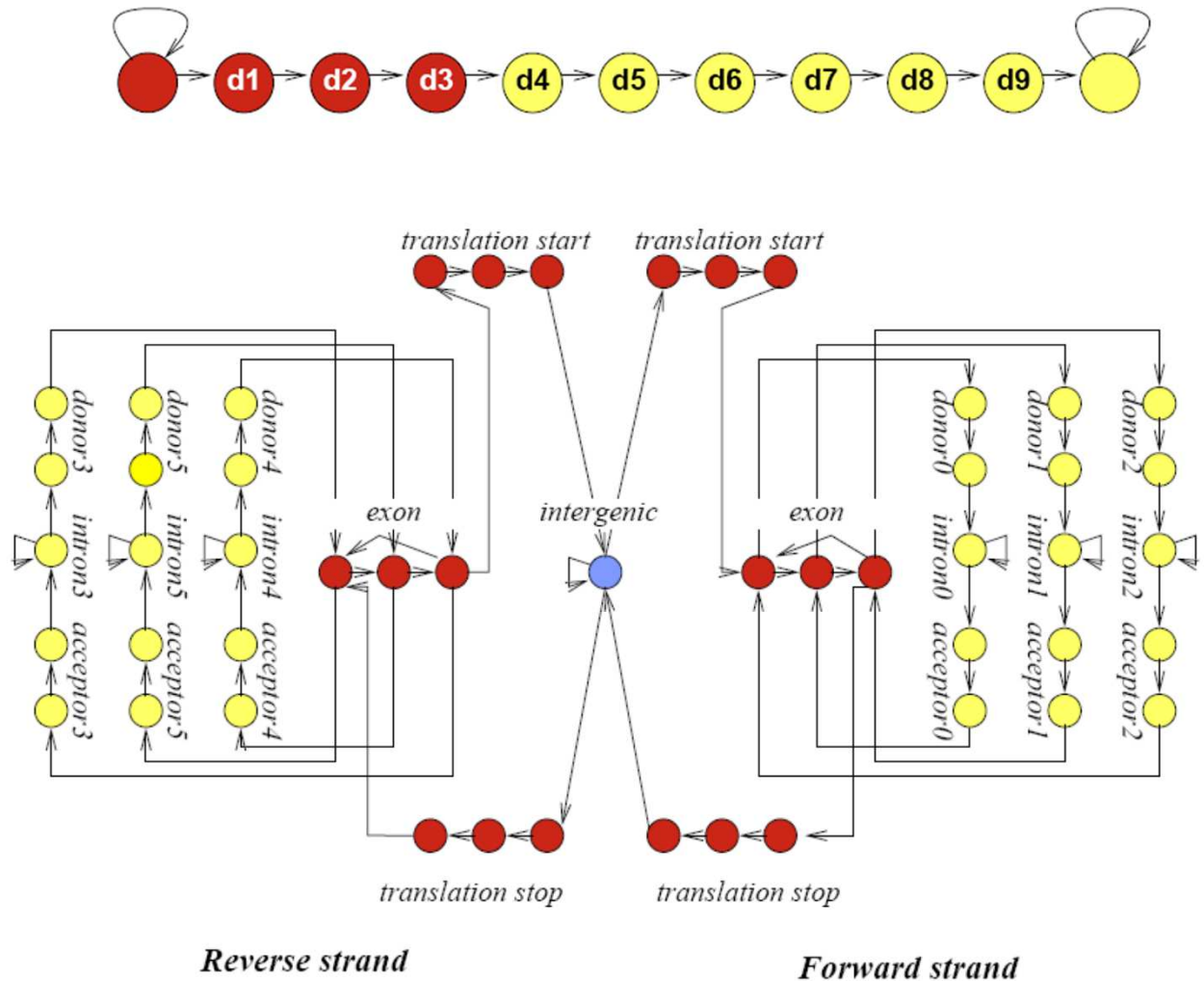
- Přerušení kodonu

- ve skutečnosti může být trojice kodonu přerušena intronem
- z jakékoliv pozice kodonu můžeme přejít do intronu
- při přechodu z intronu je nutné pokračovat od rozpracované pozice



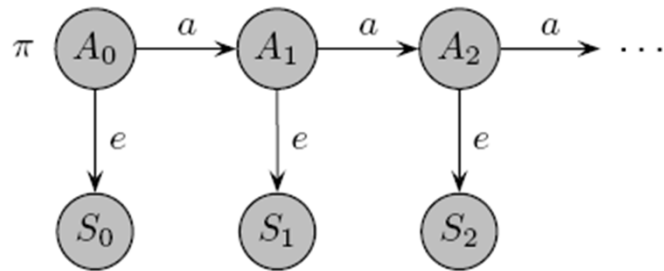
HMM 4 – detekce signálů

- Je potřeba respektovat různé skupiny signálů (sekvencí) na
 - začátku transkripce a translace
 - konce transkripce a translace
 - začátku intronu (donor)
 - konce intronu (akceptor)



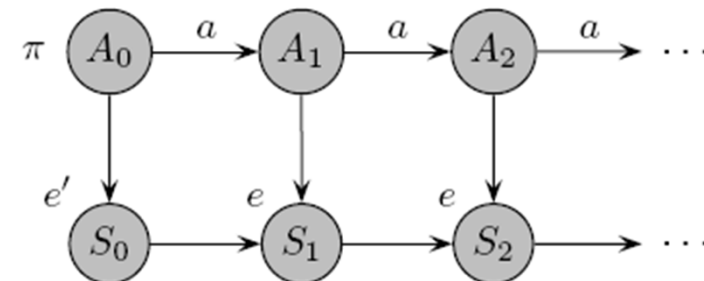
HMM vyšších řádů

- Doposud jsme předpokládali, že pravděpodobnost emise znaků je závislá pouze na stavu (HMM řádu 0)





- Ve skutečnosti může být závislá i na předchozím emitovaném znaku (HMM řádu 1, 2, ...)
- Reálné modely pro predikci genů jsou 4-5 řádu

- HMM řádu 1



e určuje $\Pr(s_i | a_i, s_{i-1})$:

a_t	s_{t-1}	a	c	g	t
	a	0.24	0.23	0.34	0.19
	c	0.30	0.31	0.13	0.26
	g	0.27	0.28	0.28	0.17
	t	0.13	0.28	0.38	0.21
	a	0.30	0.18	0.27	0.25
	c	0.32	0.28	0.06	0.35
	g	0.27	0.22	0.27	0.24
	t	0.20	0.21	0.26	0.33
...					

Příklady programů na hledání genů

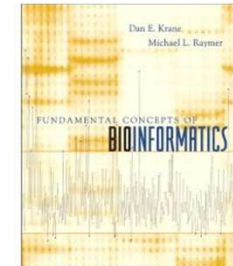
- Programy
 - GeneMark, Glimmer (Prokaryoty)
 - HMMGene, GeneScan, ExonHunter
 - Exonyphi, N-SCAN, GeneZilla, apod
 - Většina založená na HMM + doplňující informace
- Časté omezení programů
 - Alternativní splicing
 - Překrývající se geny resp. geny v intronech
 - Netypické geny (neobvyklé signály, velmi krátké nebo dlouhé exony a nebo introny)

Shrnutí

- Rozpoznávání genů **nelze** spolehlivě vyřešit **jednoduchými statistickými metodami**
- Při konstrukci algoritmu je nezbytné uvažovat řadu pomocných signálů jako jsou např. **začátek/konec transkripce, začátek/konec translace**
- Velkou **komplikací vnáší** do problematiky **výskyt intronů** uvnitř kódujících oblastí a **alternativní splicing**
- Všechny signály se navíc vyskytují v několika **variantách** a jen s určitou **pravděpodobností**
- Velmi zajímavý přístup s poměrně vysokou úspěšností nabízejí **Skryté Markovovy modely**

Literatura

- Dan K. Krane, Michael L. Raymer: ***Fundamental Concepts of Bioinformatics***, ISBN: 0-8053-4633-3, Benjamin Cummings 2003.
- Neil C. Jones, Pavel A. Pevzner, ***An Introduction to Bioinformatics Algorithms***, ISBN-10: 0262101068, The MIT Press, 2004
- Broňa Brejová, Tomáš Vinař, ***Letná škola bioinformatiky***, Bratislava, 2007



Konec

Děkuji za pozornost