README

Čauky!

Tenhle soubor jsou moje poznámky z průběžnýho učení na státnice. Neříkám, že jsou kompletní, ale někomu by se mohly hodit :)

Zatím jsou under construction :D

Vycházím ze státnicových otázek na SISu.

U některých zpracovaných otázek jsou odkazy na materiály, ze kterých vycházím (přednášky etc.).

Otázky

Požadavky znalostí k bakalářské státní závěrečné zkoušce z bioinformatiky

Matematika & informatika

1. Matematická analýza
    1. Posloupnosti a řady, konvergence, Cauchyovské posloupnosti.
    2. Reálně funkce jedné proměnné. Limita v bodě a spojitost. Derivace funkcí: definice a základní pravidla, věty o střední hodnotě, derivace vyšších řádů. Extrémy funkcí. Aplikace, např. průběh funkcí, Taylorův polynom.
    3. Integrální počet. Primitivní funkce a Newtonův integrál. Určitý (Riemannův) integrál a jeho použití.
2. Lineární algebra
    1. Soustavy lineárních rovnic, metody řešení.
    2. Matice, operace s maticemi. Hodnost matice, regulární matice a inverzní matice. Odstupňovaný tvar matice.
    3. Základní algebraické struktury: grupy, tělesa, vektorové prostory.
    4. Základní vlastnosti konečně generovaných vektorových prostorů, vektorové podprostory. Báze a dimenze.
    5. Lineární zobrazení. Základní vlastnosti, maticová reprezentace, skládání lineárních zobrazení.
    6. Skalární součin a norma. Vlastnosti v reálném i komplexním případě, Cauchy-Schwarzova nerovnost. Kolmost. Ortogonální doplněk a jeho vlastnosti, ortogonální projekce.
    7. Determinanty. Definice a základní vlastnosti determinantu. Úpravy determinantů, výpočet.
    8. Vlastní čísla a vlastní vektory matic. Výpočet a základní vlastnosti. Diagonální tvar matice, diagonalizovatelnost. Jordanův normální tvaru (v obecném případě).
3. Kombinatorika, pravděpodobnost a statistika
    1. Binární relace, ekvivalence a částečná uspořádání. Kombinatorické počítání: kombinační čísla, binomická věta, princip inkluze a exkluze.

2. Teorie grafů. Základní pojmy teorie grafů: grafy a podgrafy, izomorfismus. Stromy a jejich základní vlastnosti, kostra grafu.
3. Rovinné grafy, barvení grafů. Toky v sítícha aplikace. Souvislost grafů (míra souvislosti), Mengerovy věta.
4. Náhodné jevy, podmíněná pravděpodobnost, nezávislost náhodných jevů. Náhodné veličiny, střední hodnota, linearita střední hodnoty. Bodové odhady a testování hypotéz.

4. Algoritmy a datové struktury
    1. Časová složitost algoritmů. Metoda ,,rozděl a panuj'' - aplikace a analýza složitosti, dynamické programování.
    2. Binární vyhledávací stromy, vyvažování, haldy.
    3. Třídění - sekvenční třídění, porovnávací algoritmy, přihrádkové třídění, třídící sítě.
    4. Grafové algoritmy - prohledávání do hloubky a do šířky, souvislost, topologické třídění, nejkratší cesta, kostra grafu, toky v sítích. Tranzitivní uzávěr.
    5. Algoritmy vyhledávání v textu - Aho-Corasicková, KMP, sufixový strom, sufixové pole. Algebraické algoritmy - DFT, Euklidův algoritmus. RSA. Aproximační algoritmy. Automaty a gramatiky - typy automatů a gramatik, vztahy, příklady.
5. Aplikovaná informatika
    1. Principy a základy implementace objektově orientovaných jazyků - třída, dědičnost, polymorfismus, virtuální funkce, atd. Generické programování a knihovny šablony a generika, kompilační polymorfismus.
    2. Normální formy, referenční integrita. Základy SQL.
    3. Unix - základní pojmy (systém souborů, komunikace mezi procesy), shell (syntaxe, programové konstrukty), základní utility.

Biologie

1. Složení živých buněk - malé molekuly a makromolekuly, jejich interakce, vlastnosti vody a vodných roztoků důležité pro život, kyseliny, zásady a pufry, role vody v živých tělech,
2. Stavba buňky, funkce buněčných kompartmentů, srovnání buněčné stavby pro- a eukaryot, povrchové struktury buněk, význam specifických struktur rostlinných buněk (buněčné stěny, plastidů, vakuol) pro životní strategii rostlin
3. Membrány - stavba, biogeneze a funkce membrán, membránové proteiny, membránový potenciál a transmembránový přenos látek
4. Struktury proteinů a nukleových kyselin - primární, sekundární, terciální a kvartérní struktury, motivy a domény, supramolekulární komplexy (ribosom, spliceosom, proteasom...); princip komplementarity bází, primární a sekundární struktury DNA a RNA
5. Enzymy a jejich vlastnosti - mechanismy katalýzy, regulace enzymové aktivity, názvosloví enzymů
6. Energetický metabolismus - makroergní fosfátové sloučeniny, glykolýza a citrátový cyklus, fermentace, oxidativní fosforylace a transport elektronů, fotosyntéza - celkový přehled, dílčí reakce a komplexy, jejich lokalizace
7. Zpracování genetické informace. Centrální dogma molekulární biologie, struktura virových, pro- a eukaryotických genomů. Vertikální a horizontální přenos dědičné informace. Transpozony, viry, epigenetická dědičnost, priony
8. Základy genetiky - Mendelovy zákony, základní pojmy, různé verze definice genu. Intra- a intergenové interakce, genová vazba, genetické aspekty sexuality, chromozomové

určení pohlaví, pohlavně vázaná dědičnost, mimojaderná dědičnost.
9. Mutace a mutageneze - mutace genové, chromozomové a genomové, molekulární podstata mutací, mutageny, reparace poškozené DNA
10. Exprese genů a její regulace na úrovni transkripční, posttranskripční, translační a posttranslační, genetický kód, syntéza a distribuce proteinů v buňce, folding a účast chaperonů, posttranslační modifikace, regulace stability proteinů
11. Dynamika a funkce buněčných kompartmentů - endoplasmatické retikulum, Golgiho komplex, vezikulární transport, endo- a exocytóza, sekreční dráha a nitrobuněčné adresování proteinů, lyzosom, vakuoly, peroxisom, hydrogenosom
12. Funkční anatomie buněčného jádra - stavba jádra, jaderný obal, organizace genetické informace, chromozomy, chromatin, jadérko
13. Semiautonomní organely - evoluční historie, stavba, funkce, replikace a exprese organelového genomu
14. Cytoskelet - cytoskeletální proteiny, molekulární motory a jiné asociované proteiny, interakce s dalšími buněčnými strukturami, úloha v morfogenezi buňky a buněčném cyklu, růst a pohyb buněk
15. Mezibuněčné spoje a mezibuněčná hmota –napojení buněk na mezibuněčnou hmotu, složení a význam mezibuněčné hmoty; buněčná stěna u prokaryot a eukaryot
16. Buněčný cyklus a programovaná buněčná smrt - porovnání cyklu prokaryotní a eukaryotní buňky, fáze cyklu, replikace DNA, u eukaryot jaderné dělení, mitoza a meioza, rekombinace DNA, cytokineze, apoptóza, buněčná onkogeneze
17. Komunikace uvnitř buněk a mezi buňkami, mezibuněčný a intracelulární přenos signálu, membránové a intracelulární receptory, vybrané příklady signálních drah
18. Principy základních metod molekulární biologie - metody analytické separace makromolekul, PCR, sekvenování, molekulární klonování, genomika, proteomika, transkriptomika. Modelové organismy v molekulární biologii a genetice a jejich krátký popis a srovnání. Nejvýznamnější sekvenační projekty
19. Evoluce, různá její pojetí, významné události v dějinách teorie evoluce.
20. Lamarckismus, darwinismus, neodarwinismus
21. Mechanismy evoluce - drift, draft, evoluční tahy, genový tok, selekce
22. Mutace jako zdroj evolučních novinek, typy mutací, náhodnost mutací co do místa, času a směru
23. Selekce - mechanismus, typy, úrovně
24. Pohlavní výběr - intrasexuální a intersexuální selekce, epigamní znaky, evoluce
25. Speciace: mechanismy a typy speciací
26. Evoluce pohlavního rozmnožování
27. Homologie, analogie, plesiomorfie a apomorfie v evoluci organismů

Bioinformatika

1. definice oboru- historie bioinformatiky – oblasti bioinformatiky- biologická data
2. sekvenční srovnávání - dotplot – substituční tabulky – metody dynamického programování–lokální a globální alignment – parwise versus multiple sequence alignment
3. hledání podobných sekvencí – Blast versus FASTA - statistické zhodnocení významnosti nálezu - profilové metody (PSI-BLAST) – HMM metody

4. hledání domén a motivů – predikce transmembránových proteinů – predikce buněčné lokalizace a postranslačních modifikací
5. databáze – vlastnosti databází – formáty dat- validace dat – významné bioinformatické databáze
6. strukturní srovnávání – hledání podobných struktur
7. predikce struktury makromolekul
8. fylogenetika – stavba stromů – základní metody tvorby stromů (ML, MP, NJ, Bayes) – bootstrap analýza

Výpočet času

Okruh	n otázek	koef. učení	tok/h	hodin
Matematika & informatika	23	1.25	2	57
Biologie - molekulární a buněčná	27	1.00	2	54
Bioinformatika	8	0.9	2	14.4
Celkem	58	-	2	125.4

Poznámky

Co jsem ještě nestihl doprojít

- Hidden Markov Models
  - Bioinformatické algoritmy, přednášky 7,8,9

Bioinformatika

1. Obor "bioinformatika"

> "Bioinformatika je souborem metod, které slouží k třídění, analýze a interpretaci biologických dat (především *in silico*)." (Janet Thornton)

- literatura
  - Evžen - zápisky
    - lokálně
  - Marian - základy bioinformatiky
  - Wiki: Bioinformatics
- definice oboru
  - bioinformatika se zabývá zpracováním biologických dat
    - sběr,
    - archivace,
    - organizace,
    - interpretace
- historie bioinformatiky
  - 1707 - 1778 Carl Linne, první "bioinformatik"
  - 1951 Fred Sanger, sekvence proteinu insulinu
  - 1957 Perutz, Kendrew, první struktura proteinu

    - 1965 Margarett Dayhoff, první sekvenční databáze
    - 1970 Needlman - Wunsch, algoritmus sekvenčního srovnávání
    - 1971 první strukturní databáze
    - 1988 "HUGO project" (HUman Genome Organization)
    - 1990 Altshul, Lipman et al., BLAST
    - 1992 NCBI - GenBank
    - 1995 První osekvenovaný celý genom, Haemophilus influenze
- oblasti bioinformatiky
    - sekvenční (informační biopolymery, proteiny)
    - strukturní (Struktura DNA, RNA, proteinů)
    - signální dráhy v buňkách (stará definice bioinformatiky)
    - regulace exprese genů (signalizace, epigenetika)
- biologická data
    - v základu jakákoli data (délky motýlých křídel)
    - každopádně i další data
        - genomická data
        - sekvence (proteiny, RNA, DNA)
        - interakce
        - příbuzenské vztahy etc.
    - rozsah dat?
        - EBI 2015 - 60PB dat

2. Sekvenční bioinformatika

- literatura
    - essentials
    - Wiki CZ: Dot plot
    - Základy bioinformatiky na Drivu
    - Bioinformatické algoritmy na GDrivu
- dotplot
    - vizuální metoda pro alignment 2 sekvencí
    - postup:
        - jedna sekvence jde do řádku, druhá do sloupce
        - z teček (identita na jednotlivých pozicích) vidíme, jaké vzory se vyskytují
            - čára diagonálně zleva nahoře doprava dolů značí identický úsek
            - kolmo na tuto čáru vidíme inverzi
    - tento postup je možné použít i pro analýzu jedné sekvence
        - dáme do řádku i do sloupce tu stejnou sekvenci
            - samozřejmě bude nepřerušená čára na diagonále (identita)
            - mimo diagonálu uvidíme opakující se subsekvence a kolmo na ně inverzní opakující se subsekvence
    - nevýhoda dot-plotu je, že generuje mnoho šumu (např. pro DNA sekvenci je pravděpodobnost 1/4, že dojde k identitě na pozici u dvou náhodných sekvencí)
- substituční tabulky (Algoritmy - 4)

- při vytváření alignmentu dvou sekvencí je možné použít různé typy skórujících funkcí
    1. Hammingova vzdálenost (Hamming Distance, HD)
    - pro sekvence stejné délky!
    - identita na pozici -> 1
    - neidentita -> 0
    2. Editační (Levensteinova) vzdálenost (ED)
    - minimální Hammingova vzdálenost pro alignment dvou sekvencí
    - je možné ji spolehlivě získat pomocí dynamického programování (DP) (taková ta tabulka dvou sekvencí) a následným backtrackingem
    3. OWED (Operation-Weighted Editation Distance)
    - zobecněná ED
    - jsou zavedeny hodnoty, které mohou být upraveny pro potřeby dané situace
        - d ... penalizace za mezeru
        - e ... skóre, pokud je na pozici identita
        - r ... skóre, pokud na pozici není identita
    - je definovaná rekurzivně, ale počítá se pomocí dynamického programování
    4. AWED (Alphabet-Weighted Edit Distance)
    - OWED upravené pro specifické hodnoty d, e, r pro každou kombinaci znaků
        - znak×mezera,
        - znak sám se sebou,
        - znak×jiný znak
    - z AWEDu vychází většina skórujících tabulek, jen je většinou ještě nastavená jiná hodnota pro první gap a pro následující gapy (otevřít mezeru je "dražší" než ji prodloužit)
- Skórující tabulky
    - není problém vytvořit alignment dvou sekvencí, problém je najít vhodnou skórující tabulku, aby nám nevyšla blbost
    - Probabilistic models
        - počítání skóre na základě pravděpodobnosti, že se budou dané páry znaků ve dvou sekvencích (např. aminokyselin) vyskytovat na stejné pozici
        - dva přístupy
            1. Random Model
            - jde přes všechny kombinace pozic ve dvou sekvencích
            - (jakoby for cyklus přes *i* ve for cyklu přes *j*)
            - předpokládá, že sekvence jsou unrelated
            2. Match Model
            - jde přes shodné indexy
            - (jakoby jen jeden for cyklus přes *i* pro obě sekvence naráz)
            - pravděpodobnost, že na dané pozici pochází daná rezidua ze společného předka
            - match model předpokládá, že sekvence jsou si příbuzné
        - Odds Ratio je podíl výsledku Match modelu a Random modelu

        - podíl pravděpodobnosti, že na pozici je daná kombinace znaků (např. aminokyselin) za předpokladu, že sekvence jsou příbuzné a že sekvence nejsou příbuzné
        - když vyjde vyšší než 1, daný alignment dvou pozic je pravděpodobně nějak evolučně spřízněný
        - když vyjde nižší než 1, pravděpodobně není spřízněný
    - Log Odds Ratio
        - logaritmus Odds Ratio, vyhodí použitelné hodnoty do skórovací tabulky (nespřízněné -> menší než 0, spřízněné -> vyšší než 0, ((log(1)=0)))
    - tenhle přístup má 2 problémy
        1. je potřeba dávat pozor, aby pro nějakou obecnou skórovací matici nebyly vybrány příliš příbuzné sekvence
        2. je potřeba chytře zvolit pravděpodobnost pro Match Model - u sekvencí s evolučně bližším společným předkem bude pravděpodobnost záměny na dané pozici nižší než pro sekvence se vzdálenějším (čas po který sekvence mutovaly)
- Substituční matice PAM (Point/Percent Accepted Mutation (Dayhoff et al.)) a BLOSSUM (BLOckS SUbstitution Matrix (Henikoff et al.))
    - při tvoření matic se využívá pravděpodobnostních modelů viz výše
    - pro evoluční studie jsou běžně používány matice PAM 250 a BLOSSUM 62
    - PAM
        - vytvoří se matice PAM 1 (velmi podobné sekvence)
        - znalost o pravděpodobnostech v PAM 1 se použije pro evolučně vzdálené sekvence (PAM *n*)
        - dvě sekvence jsou 1 PAM vzdálené, pokud série mutací přeměnila jednu sekvenci na druhou pomocí 1 přijaté mutace na 100 aminokyselin (point accepted mutation)
            - přijaté (accepted) znamená, že mutace není letální a není umlčená (silent)
        - dvě sekvence vzdálené 200 PAM mají cca 25% identitu (jsou to proteinové sekvence)
        - pro *ideální* konstrukci PAM *n* matice je potřeba
            1. vzít set n PAM vzdálených sekvencí
            2. manuálně udělat jejich alignment (ještě nemáme skórovací matici, abychom to udělali automaticky)
            3. spočítat pomocí pravděpodobnostního modelu skóre v matici: $PAM_n[i,j] = log(\frac{f(i,j)}{f(i) \times f(j)})$
        - *reálně* se PAM dělá pomocí markovovských modelů
            - z toho vyplývá, že PAM *n* se vyrábí umocněním PAM 1 na *n*-tou
    - BLOSSUM
        - Na bázi PROSITE
            - BLOCKS
                - bloky motivů derivované z PROSITE knihovny

            - podíl spatřených identit a očekávaných identit
            - BLOSSUM n se vytvoří tak, že z BLOCKS se odstraní sekvence s identitou vyšší než n%
- metody dynamického programování
    - optimalizace rekurzivních algoritmů
    - je potřeba zjistit, jaká část předpočítaných výsledků se shoduje
    - nějak chytře je potřeba vytvořit tabulku s těmito shodujícími se tabulkami a začít od začátku, nikoli od konce...
    - příklady: Fibonacciho číslo ... začne se od 0, 1 - > pak se pokračuje v řadě, až se dojde k n-tému číslu
    - v sekvenční bioinformatice ... Needleman-Wunsch algoritmus -> pro dvě sekvence se vytvoří tabulka s dvěma dimenzemi ...
        - definují se hodnoty na začátku, pak si algoritmus na každé pozici vybere minimum z předchozích pozic
        - backtrackingem se zjistí, jak "alignment" postupoval
- lokální a globální alignment
    - jde především o biologický pohled - zarovnání sekvencí tak, aby odpovídaly evoluční příbuznosti
    - je možné vytvořit aligment i ručně s biologickou intuicí
    - automatizace -> informatický pohled
        - jde o to najít alignment s nejlepší skórovací funkcí (AWED)
            - potřeba vhodné skórovací matice
    - globální alignment hledá alignment pro celou sekvenci
    - lokální alignment hledá alignment jednotlivých podsekvencí - předpokládá příbuznost kratších úseků, které mohou být i proházené mezi sebou
    - algoritmus pro globální alignment je možné upravit na lokální tak, že místo záporných hodnot se do tabulky pro dynamické programování vkládají nuly
        - backtracking se pak dělá pro jednotlivé podoblasti
    - algoritmus pro globální alignment (GA) se jmenuje Needleman-Wunsch
    - dtto pro lokální alignment (LA) se jmenuje Smith-Waterman
- parwise versus multiple sequence alignment
    - párový - jednoduše pomocí DP
    - u multiple sequence alignmentu stoupá složitost exponenciálně pro n sekvencí
    - skórování MSA (multiple sequence alignmentu)
        - používá se skórování přes sloupce MSA
        - dvě metody
            - ME (Minimal Entropy) - počítá pravděpodobnost, že je reziduum x na pozici i, skóre je záporně zlogaritmované, aby byla 0 nejvyšší možné skóre
            - SP - Sum of Pairs
                - v podstatě se udělá skórování pomocí PAM / BLOSSUM přes všechny dvojice v každém sloupci MSA
    - spoustu heuristických algoritmů pro MSA
        - Progressive iterative methods
            - Feng&Doolitle

        - ClustalW, Clustal Omega
    - Consistency based
        - T-Coffee
    - Iterative refinement
        - Barton&Sternberg
    - Block-Based
        - DIALIGN
    - Mix
        - MAFFT,
        - MUSCLE

3. Hledání podobných sekvencí v databázích

> hledání podobných sekvencí – Blast versus FASTA - statistické zhodnocení významnosti nálezu - profilové metody (PSI-BLAST) – HMM metody

- literatura
    - Bioinformatické algoritmy na GDrivu, přednášky 6(, 7) a 8
- úvod
    - obří databáze - potřebujeme je efektivně prohledávat
        - v nejhorším případě lineární složitost, spíš lepší
        - optimální algoritmus má kvadratickou složitost pro sekvence
        - je potřeba využít heuristiky
    - využívá se hashování
        - list pro všechny sekvence a k-tice (všechny možné?)
            - kde v sekvenci se daná k-tice nachází
            - pomocí dvou listů, jeden (b) obsahuje pro všechny k-tice jejich první výskyt,
            - druhý (a) pro celou sekvenci dává pointry na tu další k-tici
            - lepší vysvětlení v 6. přednášce na 6. slidu
- Blast versus FASTA
    - FASTA
        - 4 kroky
            1. pro query a databázovou sekvenci se najdou všechny matchující k-tice (v tabulce podobné jako u NW algoritmu)
            2. k-tice se pospojují (za gapy mezi nimi se dává penalizace)
            3. vybere se 10 nejlepších spojených subsekvencí a oskórují se pomocí PAMu nebo BLOSSOM, z nich se započítají ty nejlépe skórující, součtem skóre subsekvencí se ohodnotí celá sekvence
            4. mezi nejlépe skórujícími sekvencemi se udělá Smith-Waterman
    - BLAST (Basic Local Alignment Search Tool)
        - algoritmus má 5 kroků, stejně jako ve FASTA jde o to najít malé množství sekvencí, které pak už budou oskórované pomocí SW algoritmu
            1. pro query sekvenci jsou zjištěny všechny její k-tice (typicky k=3)
            2. pro každou k-tici z kroku 1. jsou vygenerovány všechny teoreticky možné k-tice, které s ní budou skórovat nad určitou hodnotu (T),

skóruje se pomocí PAM, či BLOSSUM
3. pro množinu všech těchto k-tic se udělá "finite state automata (FSA)", kterým se proskenují všechny sekvence v databázi. FSA zaznamená pozici jednotlivých (hledaných) k-tic v sekvencích
4. krok 4 má dvě varianty
    1. ve starší verzi se rozšiřují nalezené k-tice do obou směrů, dokud skóre subsekvence nespadne pod určitý limit (X_u), rozšířená k-tice se nazývá "high scoring pair (HSP)"
    2. v novější verzi BLASTU musí HSP obsahovat na konci rozšiřování alespoň 2 k-tice, čímž se sníží nějaká náhodnost toho procesu
5. je stanovena hodnota S_g, nad kterou dosáhne skóre jen 2% HSP. Sekvence těchto vybraných HSP pak jsou normálně zalignovány pomocí SW algoritmu

- zatímco FASTA spojuje k-tice, které jsou nalezeny v databázových sekvencích a ty se pak oskórují, BLAST pomocí všech k-tic podobných s k-ticemi v zadané sekvenci vytvoří FSA, tím najde všechna stejná místa v databázových sekvencích, tato místa pak nechá rozšířit (HSP) a ty se pak oskórují

- statistické zhodnocení významnosti nálezu
    - E-value
        - E-value rovno 1 znamená, že v prohledávané databázi se dá očekávat pro dané skóre 1 sekvence prostě náhodou
        - čím menší je E-value, tím menší je šance, že by např. vyhledaná sekvence byla vyhledána náhodně.
        - ve velkých databázích by krátké sekvence měly vysoké E-value, i kdyby byly třeba velmi podobné vstupní sekvenci
        - E-value je definovaná pomocí Gumbelova (extrémového) rozdělení
            - E-value je pravděpodobnost, že dobrý alignment nějaké sekvence v databázi má vyšší skóre, než skóre daného hitu
                - Gambelovo rozdělení má CDF definovanou jako $exp(-e^{-\lambda(x-\mu)})$, λ je rozptyl a μ střední hodnota
                - E-value je pak 1-$exp(-e^{-\lambda(x-\mu)})$
                - tohle E-value je trochu jiné, než databázové E-value
                    - FASTA to vypočítané E-value pronásobí počtem sekvencí v databázi
                    - BLAST počítá s tím, že delší sekvence mají větší šanci na to být hitnuté
                        - pronásobí E-value $\times \frac{N}{n}$, kde N je počet reziduí v sekvenci a n je délka nalezené sekvence

- profilové metody (PSI-BLAST) (lecture-09, první polovina)
    - hledají se konzervovaná místa (např. Prosite databáze)
    - přístupy mají různou komplexitu, dávají různé množství informací
    - konsenzus sekvence ("úplně jednoduchý grep"), jen aminokiselina a jeden wild-card - *
    - patterns - grep patterny,

        - má trochu upravenou syntax
            - wildcard ... X
            - [] místo ^ etc.
    - PSSM (Position specific scoring matrix)
        - Hoksza o PSSM říká, že na nich většina studentů u státnic zbytečně pohoří!!! :// Creepy
        - také se nazývá profil
        - pro daný MSA se spočítají frekvence jednotlivých typů aminokyselin na určitých pozicích
        - video přednášky Hokszy od Matěje Zátka je pure gold, na 20:00 kočka Hokszovi ničí mikrofon u sluchátek :D
        - PSSM hodnoty pro jednotlivé aminokyseliny se ještě upraví pomocí pseudo-countů, aby se trochu oslabil vliv toho, co vidíme a měly šanci i jiné aminokyseliny - dává se to kvůli tomu, aby se oslabil vliv toho, že jsem neviděl všechny sekvence z celé té rodiny, v některé sekvenci by třeba nějaká jiná aminokyselina byla.
            - tato hodnota se v PSSM pro danou aminokyselinu na dané pozici se tedy spočítá takto:
                - *n* ... počet reálných výskytů aminokyseliny v alignmentu
                - *k* ... počet sekvencí v alignmentu
                - *a* ... počet aminokyselin (jo jde to určitě i pro nukleotidy?)
                - *ps* ... pseudocount
                - $f_{ij} = \frac{n+ps}{k+a*ps}$
        - pak se to celé ještě prožene log-likelihood ratio nulového modelu
            - $s_{ij} = log(\frac{f_{ij}}{q_i})$
                - kde f_{ij} je ta pseudocountovaná hodnota
                - q_i je pravděpodobnost, že na i-té pozici by daná aminokyselina byla náhodně
        - hodnoty tabulky pak říkají, kolikrát je pravděpodobnější, že hodnota v sekvenci pochází z match modelu než random modelu (pokud je vyšší než 0), nebo kolikrát je pravděpodobnější random model než match model (pokud je nižší než 0)
        - s PSSM se pak jede přes všechny pozice ve zkoumané sekvenci a pro každou pozici se zapisuje skóre
        - nevýhoda, neumožňuje inzerce a delece
        - nechce se mi teď zpětně přepisovat ještě poznámky, ale rád bych líp shrnul postup pro PSSM
            1. vytvoření dobrého MSA sekvencí s motivem
            2. vytvoření PSSM
                1. zaznamenání počtu výskytů jednotlivých reziduí na daných pozicích
                2. volba hodnoty pseudo-countu
                3. spočtení f_{ij} - pseudocountované pravděpodobnosti výskytu reziduí na daných pozicích
                4. log-likelihood ratio (převedení pravděpodobnosti na skóre, využívá se znalosti random a match modelu)

3. použití PSSM na dané sekvenci
    1. oskórování všech pozic pomocí PSSM

- PSI-BLAST
    - je to docela super v tom, že se naleznou vzdálenější homologové než jen pomocí BLASTu, ale může to občas vyhledat něco dost mimo - přidá to do PSSM nějakou nesouvisející sekvenci (profile drift)
    - využívá to metod BLASTu (P) a PSSM
    1. pro zadanou sekvenci se vyhledají pomocí BLASTP podobné sekvence
    2. z nejlepších vyhledaných se udělá MSA
    3. z MSA se vytvoří PSSM
    4. profil (PSSM) se použije pro další vyhledávání v databázi pomocí BLASTP
    5. pokud jsou nové hity, přidat do MSA a vytvořit nový profil
    6. opakovat 4. a 5., dokud jsou nalézány nové hity

- HMM metody (v přednáškách 07, 08, 09)
- HMM metody (lecture-09, druhá část, cca slide 25)
    - PSSM má nevýhodu, že neumí inzerce a delece
    - např. Viterbi algoritmus
        - zjišťuje nejpravděpodobnější variantu (nikoli nejpravdivější), jak by mohla být jakoby zalignovaná zkoumaná sekvence za předpokladu, že patří do příbuzné skupiny k určitému motivu
    - sestrojení FSA, kde jsou tři typy průchodů pro každou pozici
        - *match*
        - *insert*
        - *delete*
    - jde o to najít pravděpodobnosti přechodů
    - ty se vypočítají podobně jako u PSSM z poměrů toho, jak jdou jednotlivé "matche", "inzerce" a "delece" v MSA a plus se to ještě pseudo-countuje

4. Domény a motivy

> hledání domén a motivů – predikce transmembránových proteinů – predikce buněčné lokalizace a postranslačních modifikací